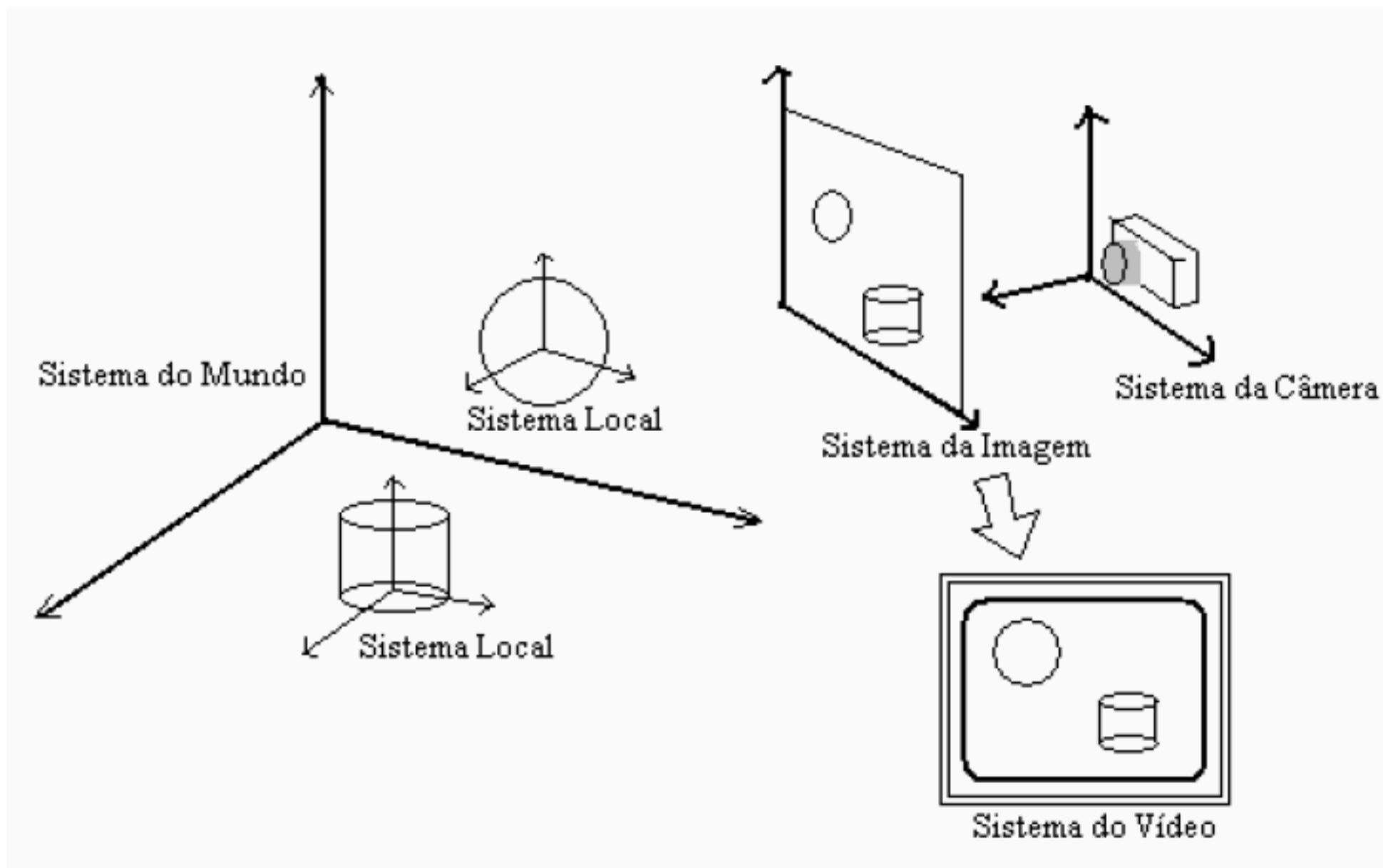


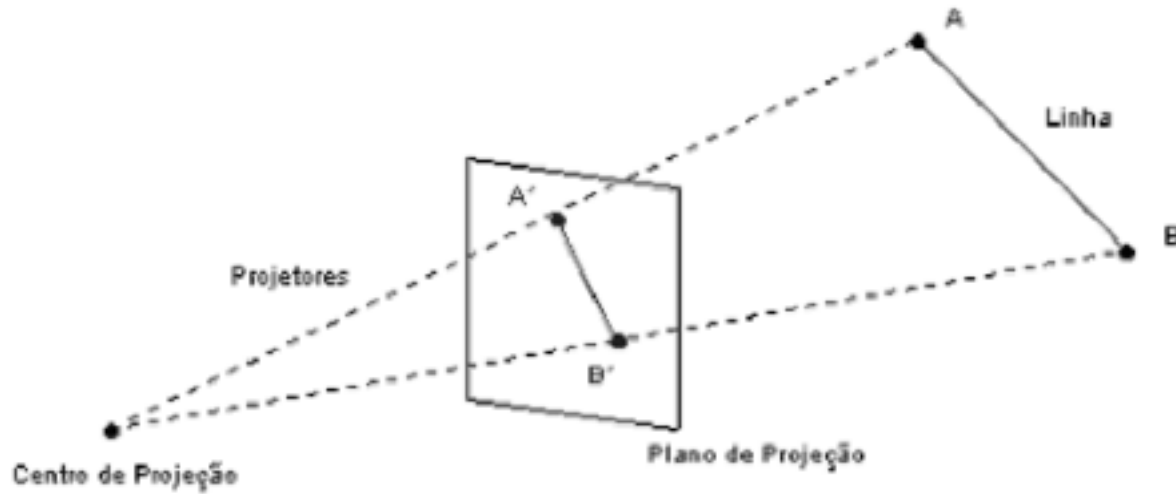
Visualização 3D

Sistemas de coordenadas

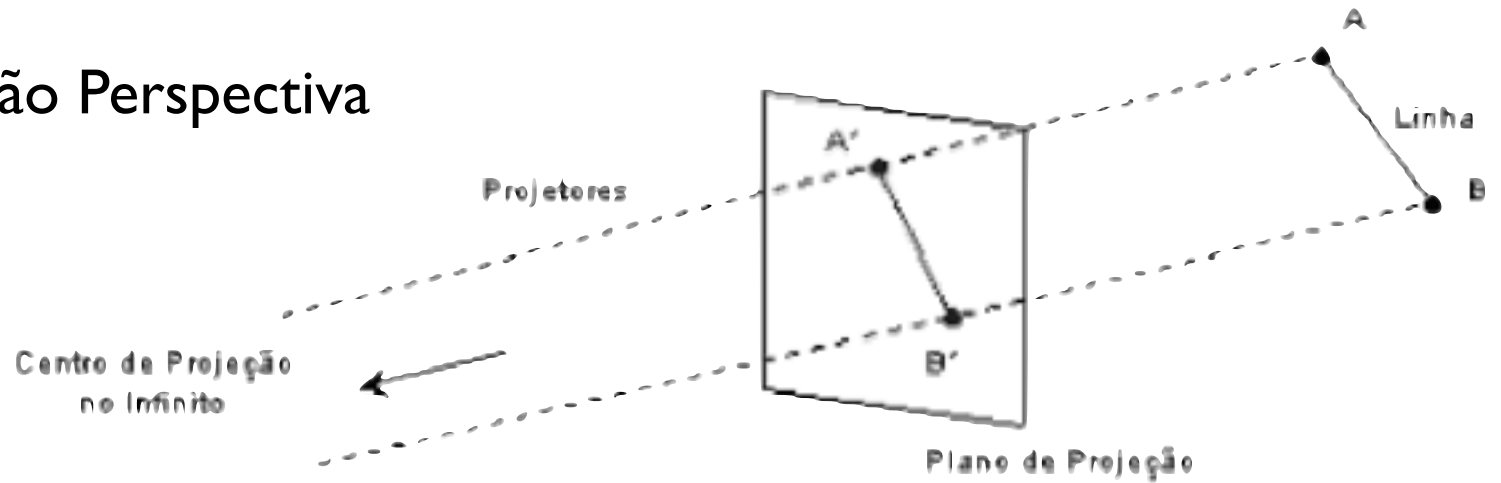
- **Espaço do Objeto:** Sistema de coordenadas local associado a cada objeto. Visa facilitar a modelagem de cada objeto.
- **Espaço da Cena ou do Mundo:** Sistema de coordenadas global. Nele estão incluídos todos os objetos da cena, inclusive a câmera virtual. Neste espaço é possível saber a relação espacial entre todos os objetos do cenário.
- **Espaço da Câmera:** Espaço definido por um sistema de coordenadas associado à câmera. Pode ser associado à uma projeção perspectiva ou a uma projeção paralela. Este sistema é utilizado para definir os parâmetros da câmera em relação ao mundo: posição, orientação, distância focal, etc.
- **Espaço da Imagem:** Espaço definido por um sistema de coordenadas 2D no plano de projeção, onde se localiza a tela virtual.
- **Espaço do Dispositivo:** Espaço associado à superfície de exibição do dispositivo de saída gráfica (vídeo, monitor).



Projeções planares

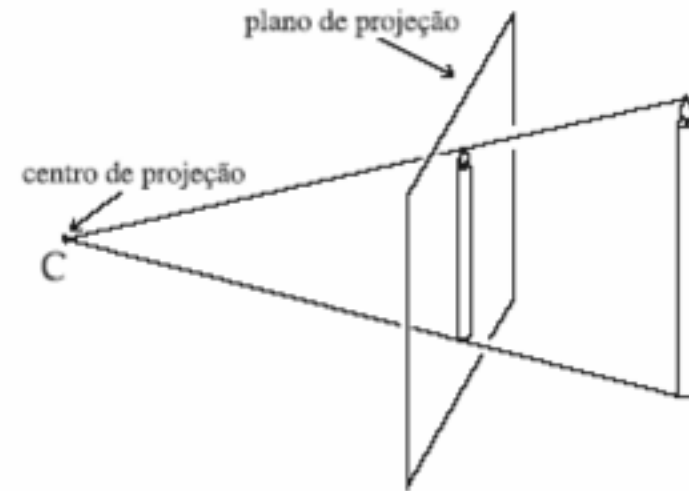
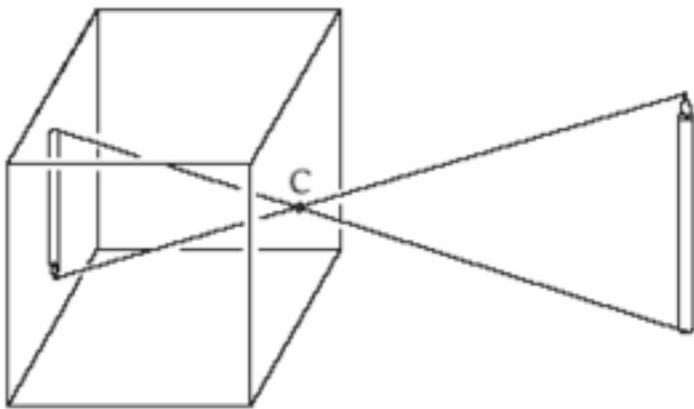


Projeção Perspectiva



Projeção Paralela

Projeção perspectiva



Modelo de câmera *pinhole*

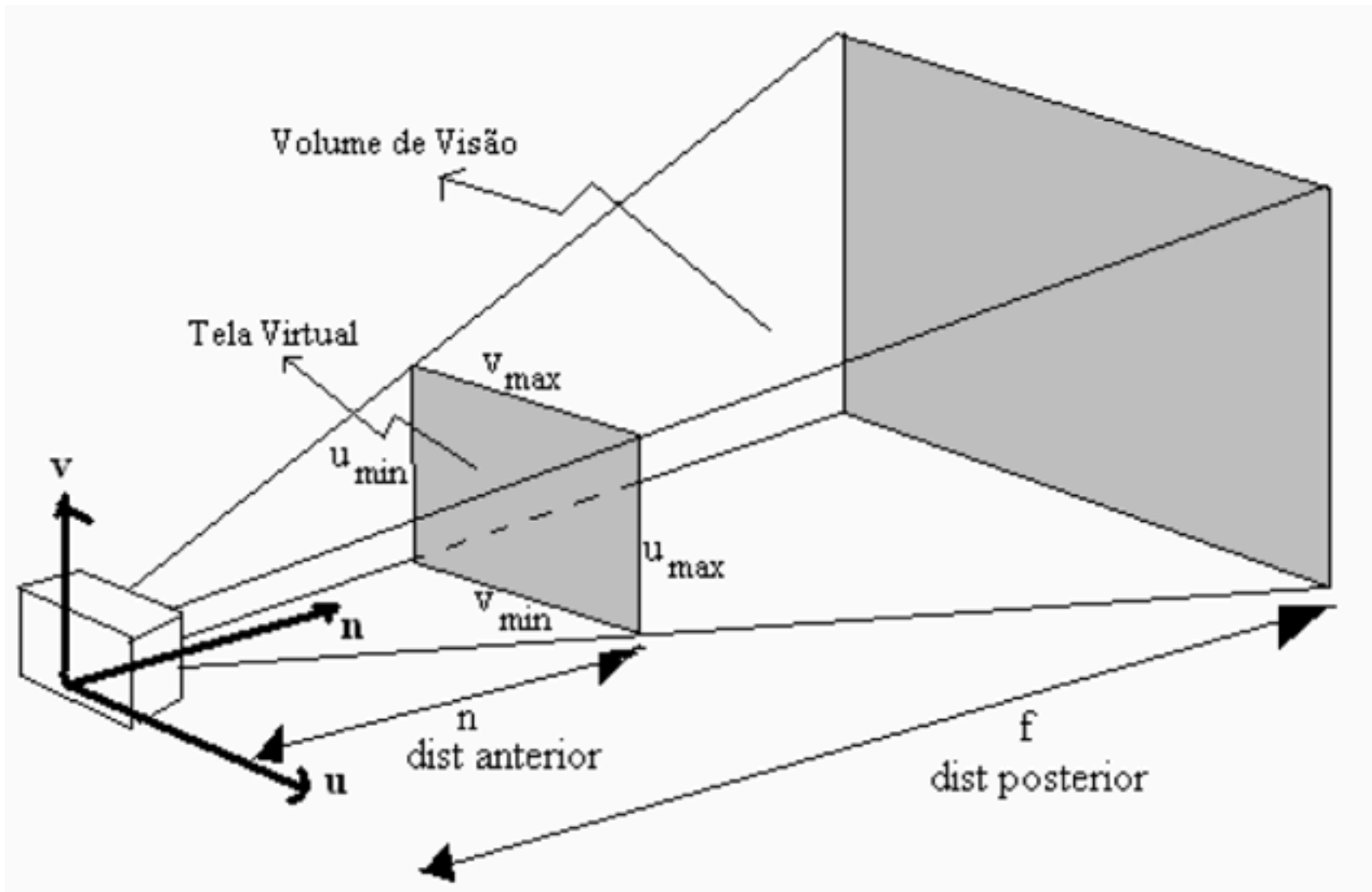
Especificação da câmera virtual

- **Posição da Câmera:** também chamado de centro de projeção é a posição do espaço onde a câmera será colocada. Este parâmetro às vezes é chamado de posição do olho (eye).
- **Orientação da Câmera:** é definida por: uma direção $\mathbf{n}=(n_x,n_y,n_z)$ que indica para onde a câmera está olhando, uma direção de inclinação $\mathbf{v}=(v_x,v_y,v_z)$. O produto vetorial destes dois vetores \mathbf{n} e \mathbf{v} definem um outro vetor \mathbf{u} perpendicular a \mathbf{u} e a \mathbf{n} . Os tres vetores $\mathbf{u},\mathbf{v},\mathbf{n}$ definem o sistema de coordenadas da câmera

Especificação da câmera virtual

- **Tela Virtual:** definida pelos valores (u_{min}, u_{max}) , (v_{min}, v_{max}) , representa a região do plano de projeção onde a imagem será formada.
- **Distância Anterior e Posterior:** a distância anterior n indica onde está o plano de projeção e a distância posterior f limita o volume de visão.

Câmera Virtual

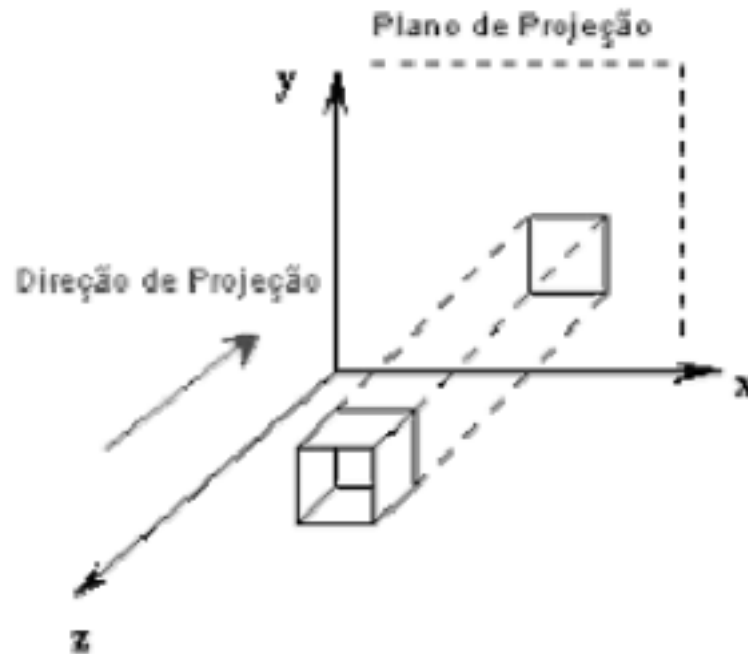


Câmera em *OpenGL*

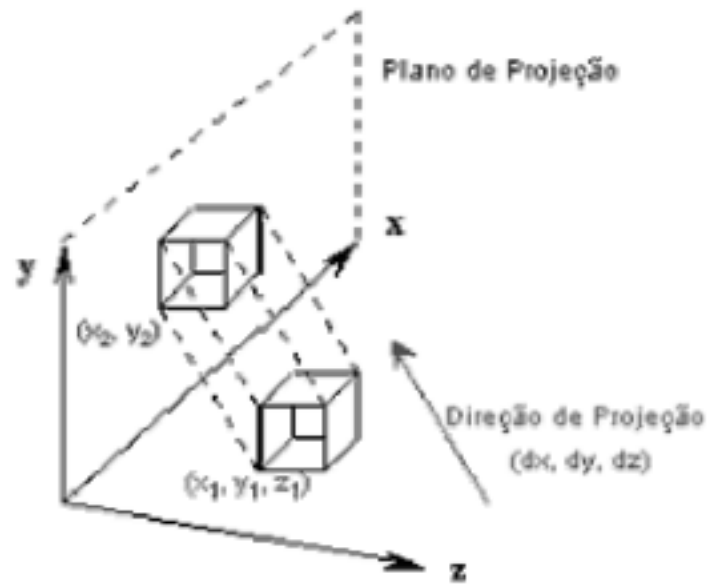
```
void gluLookAt(double eyex, double eyey, double eyez,  
              double nx, double ny, double nz,  
              double upx, double upy, double upz);
```

```
void glFrustum (double u_min, double u_max, double y_min,  
              double y_max, double n, double f);
```

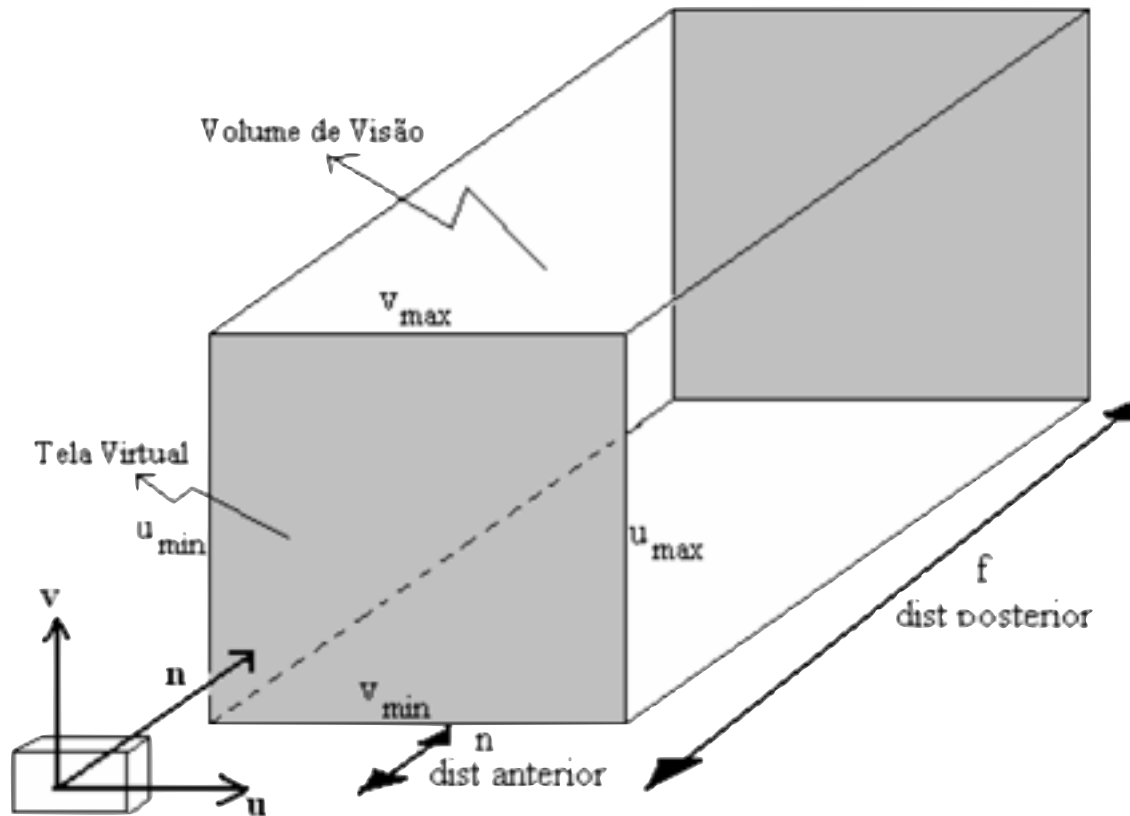
Projeção Ortográfica



Projeção Ortográfica



Projeção Ortográfica em *OpenGL*



```
void glOrtho(double  $u_{min}$ , double  $u_{max}$ , double  $y_{min}$ ,  
             double  $y_{max}$ , double  $n$ , double  $f$ ),
```

gluLookAt revisitado

**void gluLookAt(double eyeX, double eyeY, double eyeZ,
double centerX, double centerY, double centerZ,
double upX, double upY, double upZ);**

$$F = \begin{pmatrix} \text{centerX} - \text{eyeX} \\ \text{centerY} - \text{eyeY} \\ \text{centerZ} - \text{eyeZ} \end{pmatrix}$$

$$f = \frac{F}{\|F\|}$$

$$UP'' = \frac{UP}{\|UP\|}$$

$$s = f \times UP$$

$$u = s \times f$$

gluLookAt revisitado

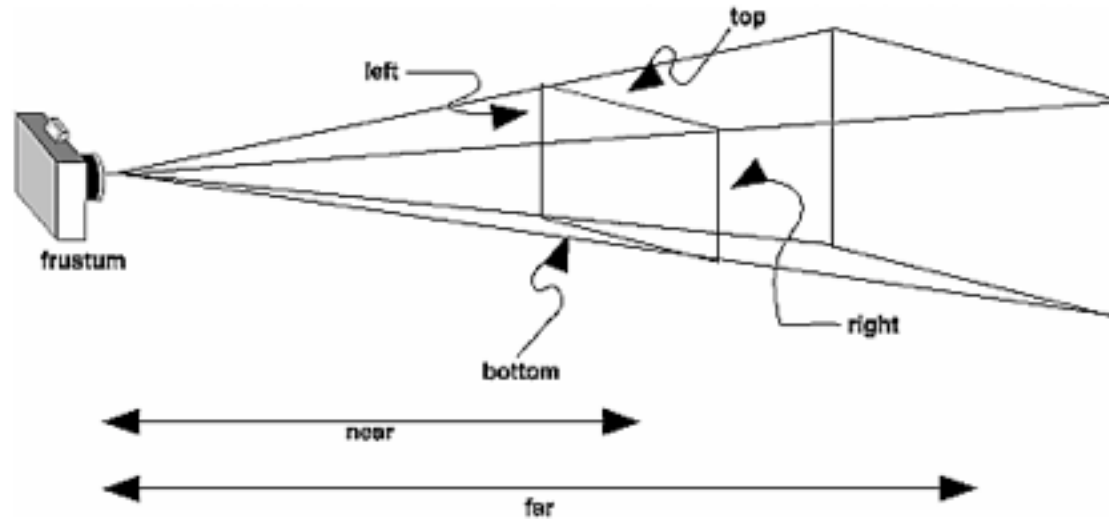
$$M = \begin{pmatrix} s[0] & s[1] & s[2] & 0 \\ u[0] & u[1] & u[2] & 0 \\ -f[0] & -f[1] & -f[2] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
glMultMatrixf(M);  
glTranslated(-eyex, -eyey, -eyez);
```

Espaço do Mundo para Espaço da Câmera

glFrustum revisitado

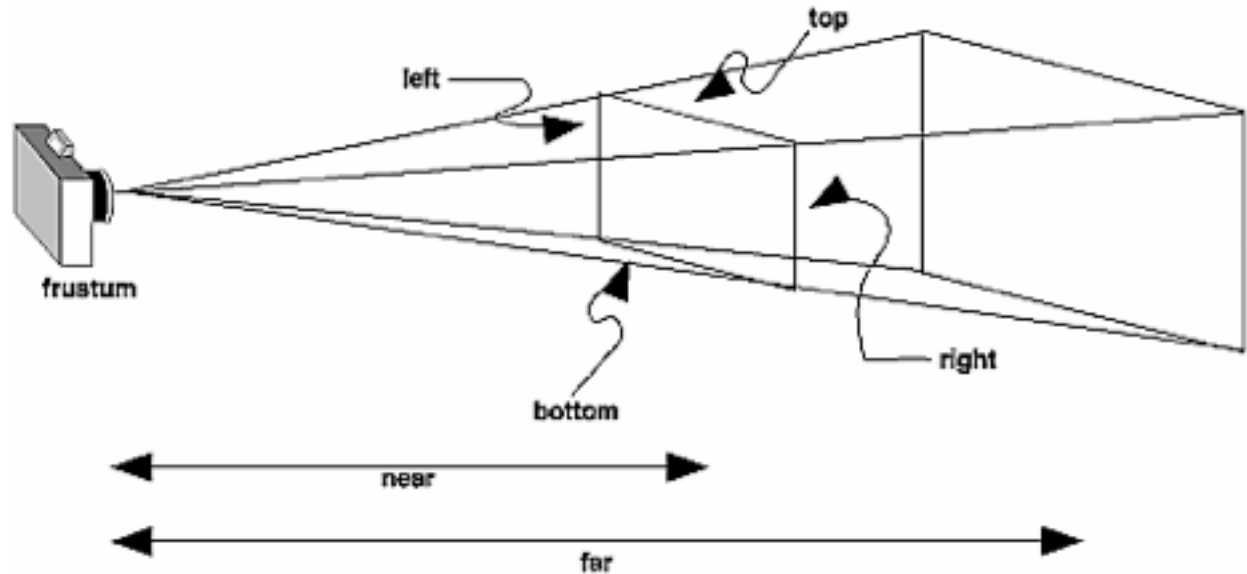
```
void glFrustum (double  $u\_min$ , double  $u\_max$ , double  $y\_min$ ,  
double  $y\_max$ , double  $n$ , double  $f$ );
```



glFrustum revisitado

1. Recorte (Clipping)

2. Projeção



$$\begin{bmatrix} \frac{2nearVal}{right - left} & 0 & A & 0 \\ 0 & \frac{2nearVal}{top - bottom} & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$A = \frac{right + left}{right - left} \quad C = -\frac{farVal + nearVal}{farVal - nearVal}$$

$$B = \frac{top + bottom}{top - bottom} \quad D = -\frac{2 farVal nearVal}{farVal - nearVal}$$

Espaço da Câmera para Espaço da Imagem

Câmera Yaw, Pitch e Roll

α : Ângulo de rotação anti-horária ao redor do eixo z

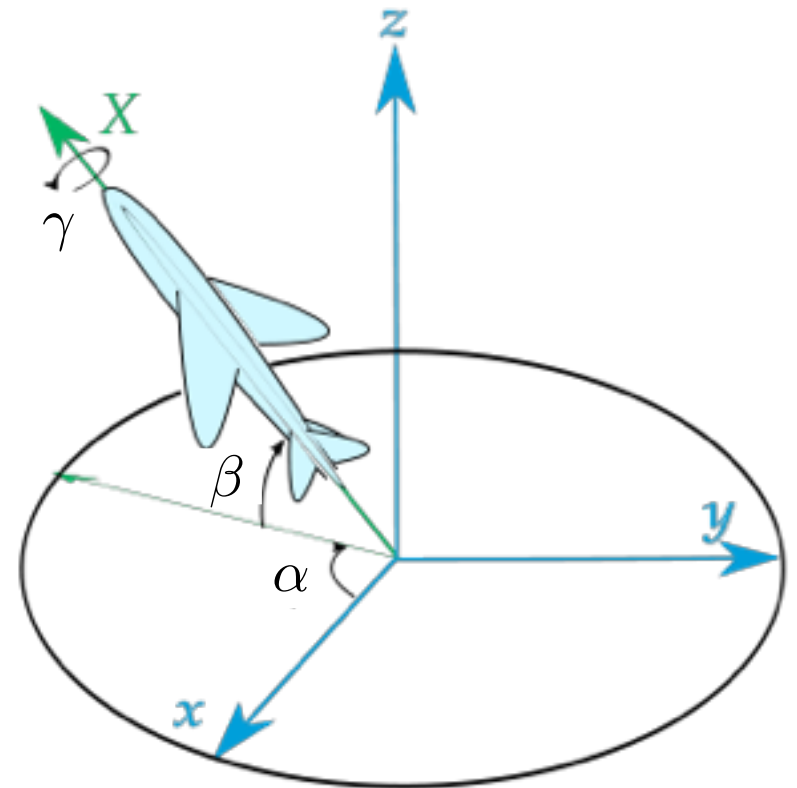
$$R_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

β : Ângulo de rotação anti-horária ao redor do eixo y

$$R_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}.$$

γ : Ângulo de rotação anti-horária ao redor do eixo x

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}.$$



Câmera Yaw, Pitch e Roll

$$R(\alpha, \beta, \gamma) = R_z(\alpha) R_y(\beta) R_x(\gamma) =$$

$$\begin{pmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{pmatrix}.$$

