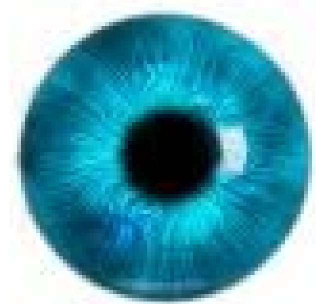
Computação Gráfica

Viewing

Aluno: Marcio Kassouf Crocomo

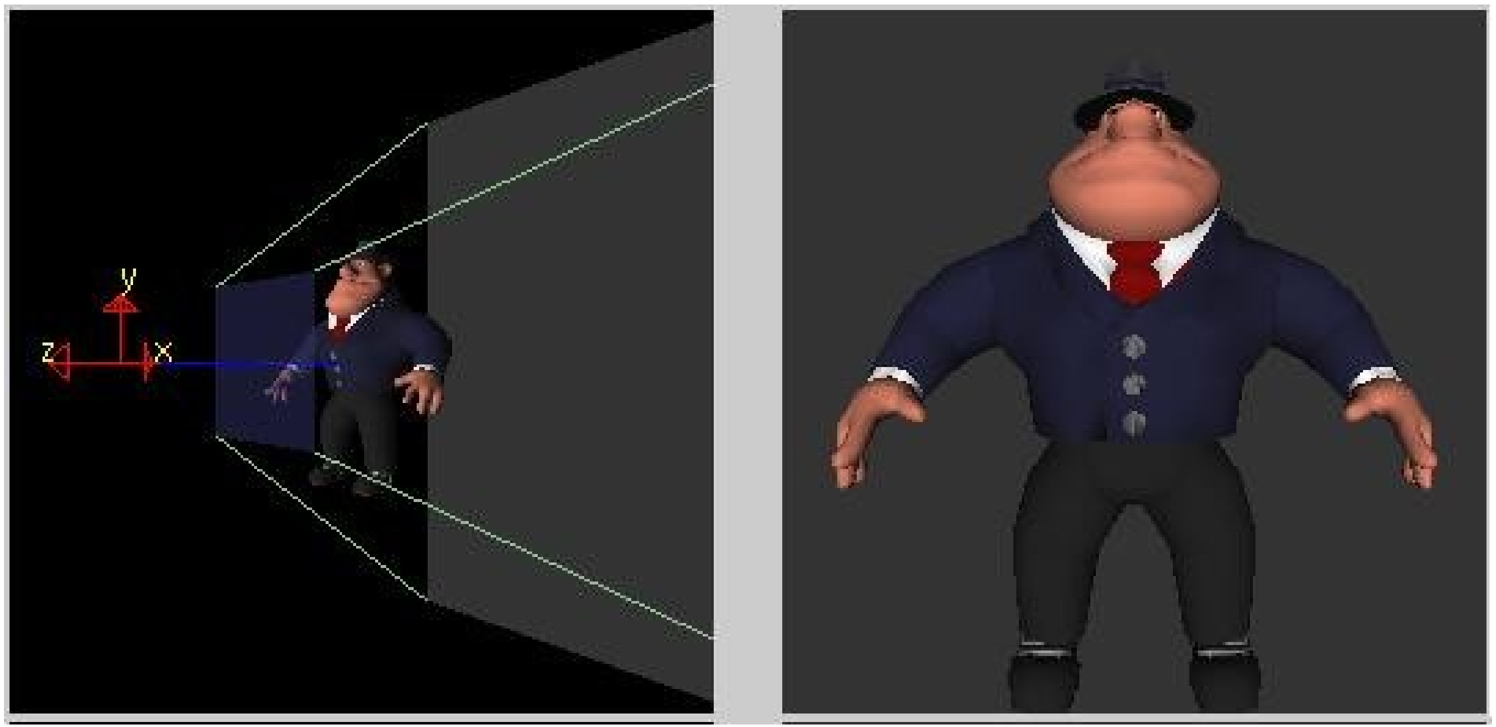
Prof: Rosane Minghim



O que é Viewing?

Processo responsável por determinar

o que será exibido no dispositivo de saída, e como

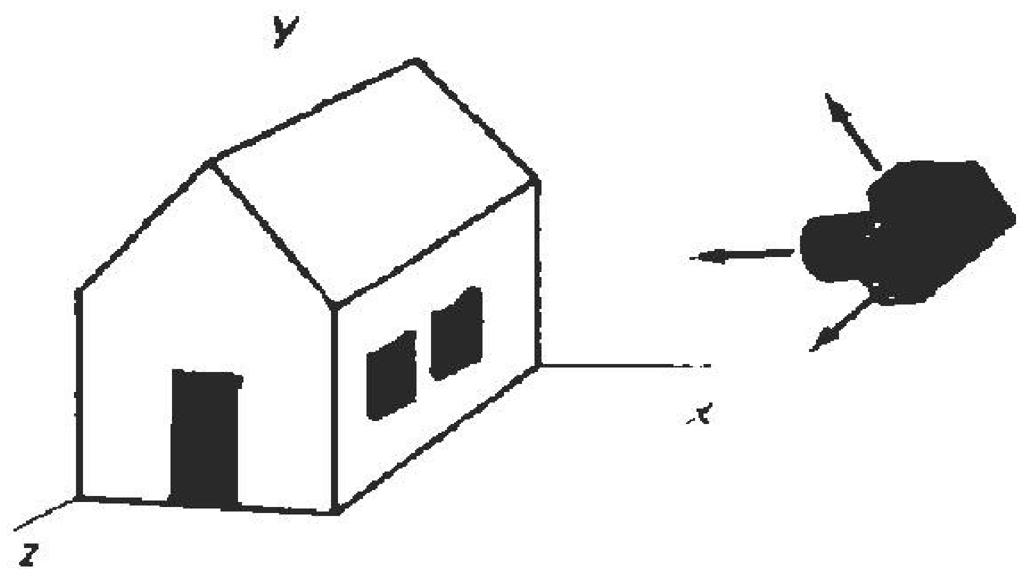


Fonte: Software disponível em: “Nate Robins: Main OpenGL Chronicles Allies”: http://www.xmission.com/~nate/tutors.html

Analogia com uma câmera

Parâmetros a serem considerados ao bater uma foto:

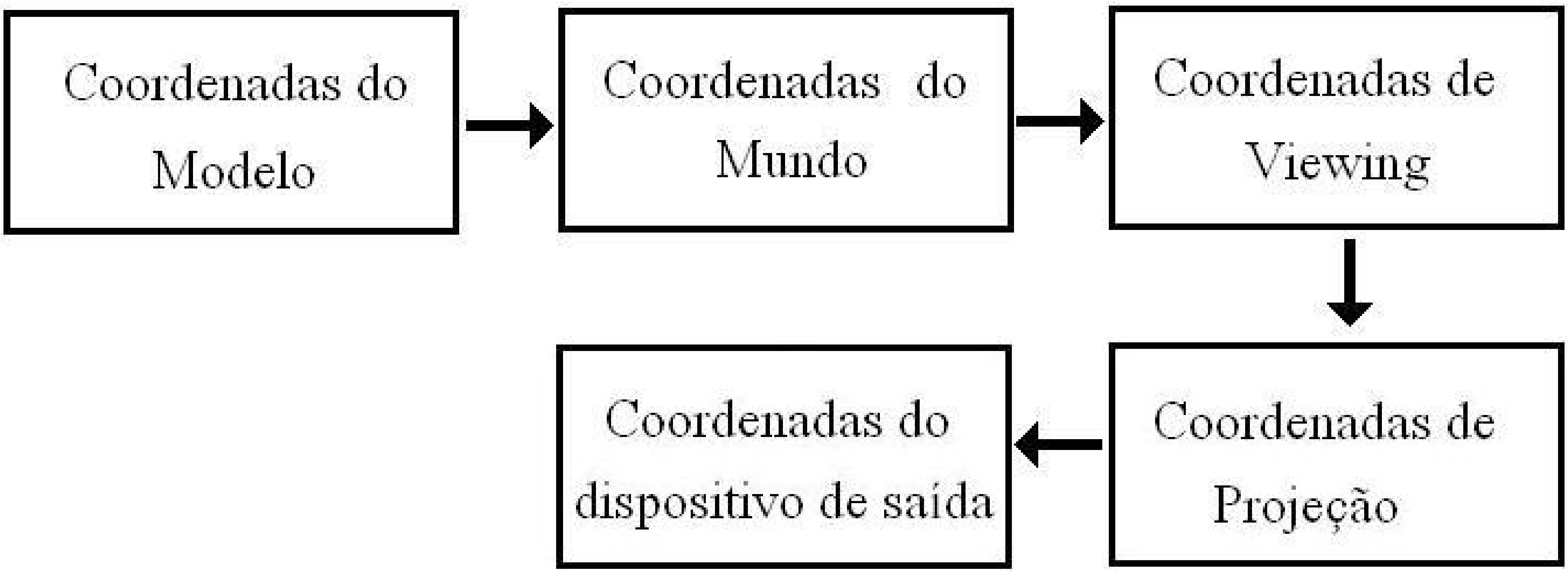
- Posicionamento da câmera no espaço.
- Direção para onde a câmera aponta.
- Orientação da câmera



Fonte Figura: Computer Graphics: C version. Cap. 12, Hearn & Baker

Viewing - Transformações

Transformações entre sistemas de coordenadas:



Organização

- Coordenadas de *Viewing*
- Projeções
- Volume de Visão
- Recorte (*Clipping*)

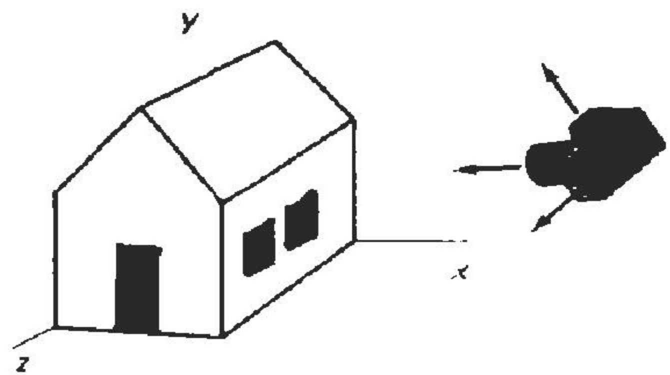
Sistema de Coordenadas de Viewing (VCS)

Para obter um VCS, devemos determinar:

- Origem do sistema de coordenadas P0 = (x_0, y_0, z_0).

- Ponto focal.

– Orientação: Vetor chamado “view up”, ou ***V***



Sistema de Coordenadas de Viewing (VCS)

A partir dos 3 parâmetros definidos, temos os 3 vetores abaixo, que nos dão os eixos do nosso sistema de coordenadas de Viewing (X_v, Y_v e Z_v):

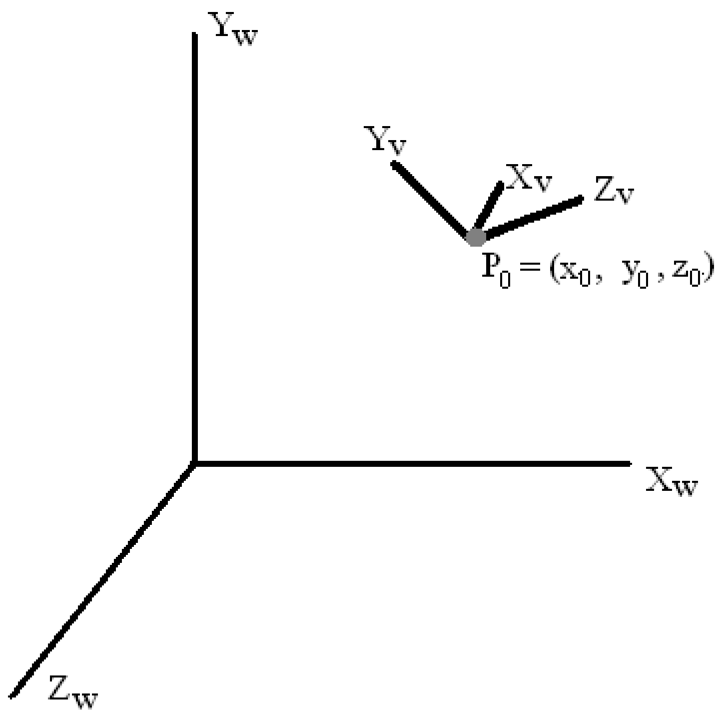
- ***V*** – Vetor *“View Up”*
- ***N*** – Vetor Normal ao plano de visão *.
- ***U*** – Produto vetorial entre ***V*** e ***N*** (***V*** x ***N***)

*Plano de visão: Plano perpendicular à linha que liga P0 ao ponto focal. Normalmente contém o ponto focal.

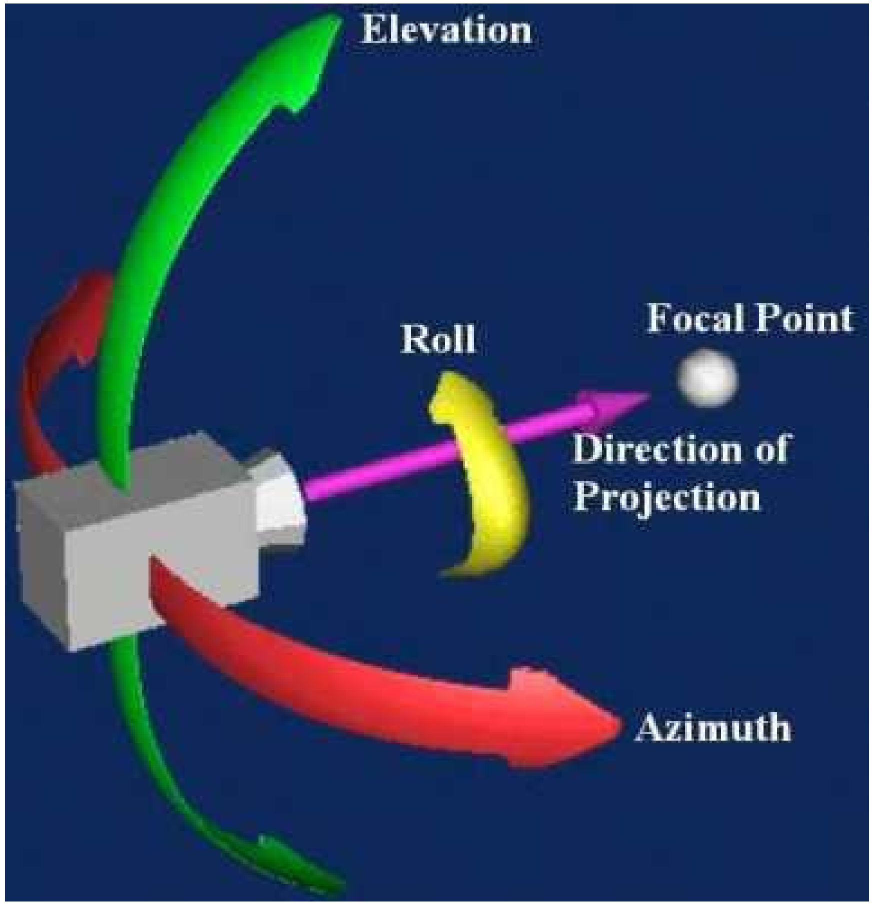
Sistemas de Coordenadas

Sistemas de Coordenadas do Mundo (WCS) – X_w, Y_w, Z_w

Sistemas de Coordenadas de Viewing (VCS) – X_v, Y_v, Z_v

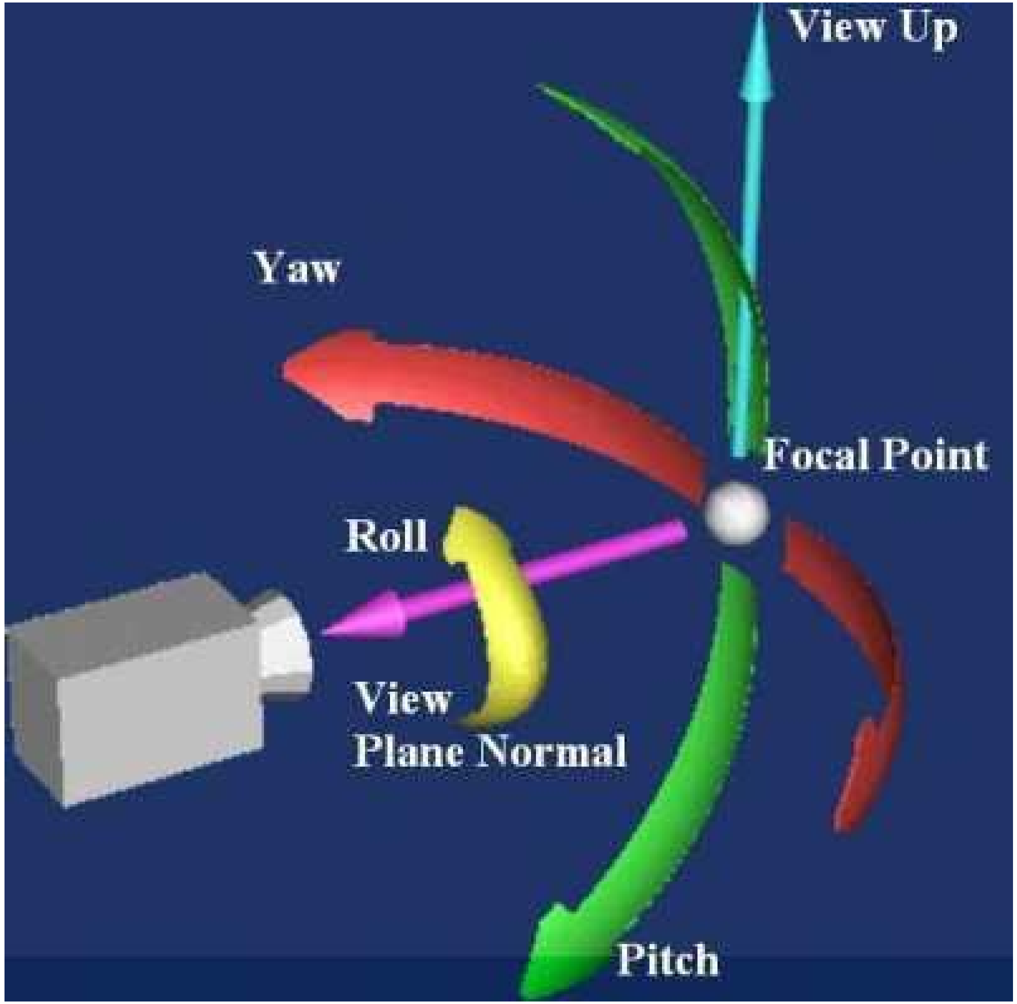


Manipulação da Câmera



Fonte Figura: Schröeder, The Visualization Toolkit, 1998

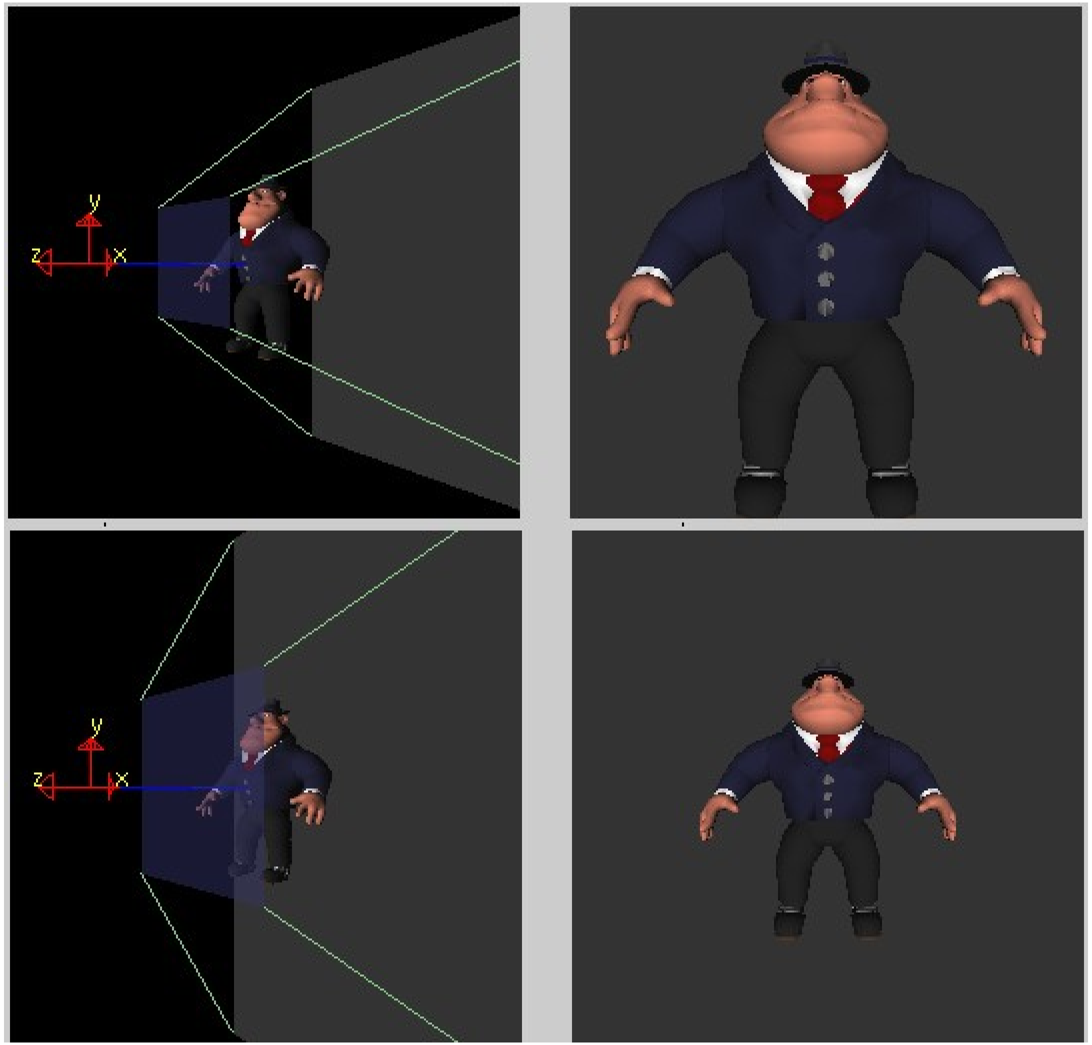
Manipulação da Câmera



Manipulação da Câmera

Zoom in

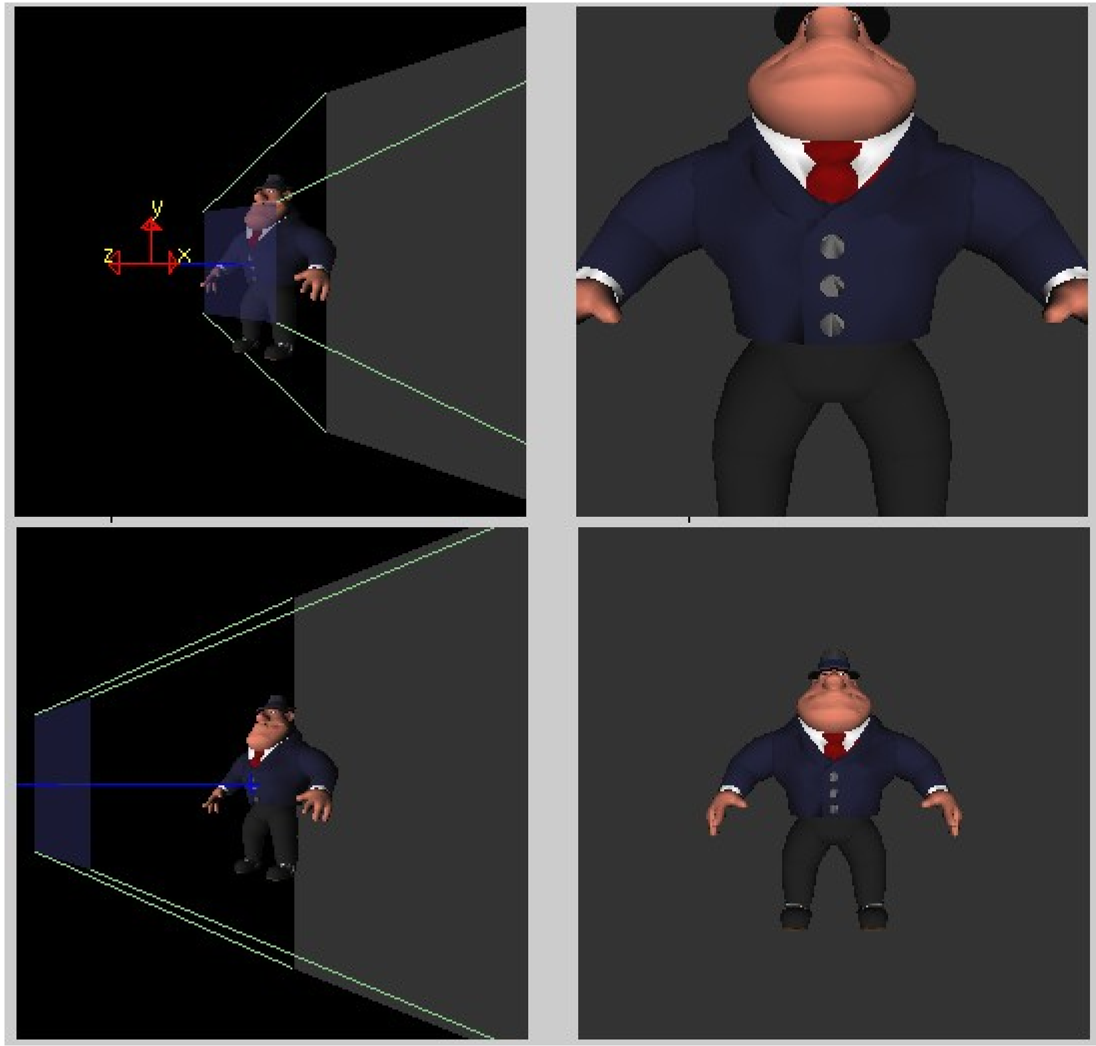
Zoom out



Manipulação da Câmera

Dolly in

Dolly out



Transformação – WCS → VCS

Obter uma matriz de transformação ($M_{wc,wv}$) que leve um ponto que esteja no sistema de coordenadas do mundo para o sistema de coordenadas de viewing.

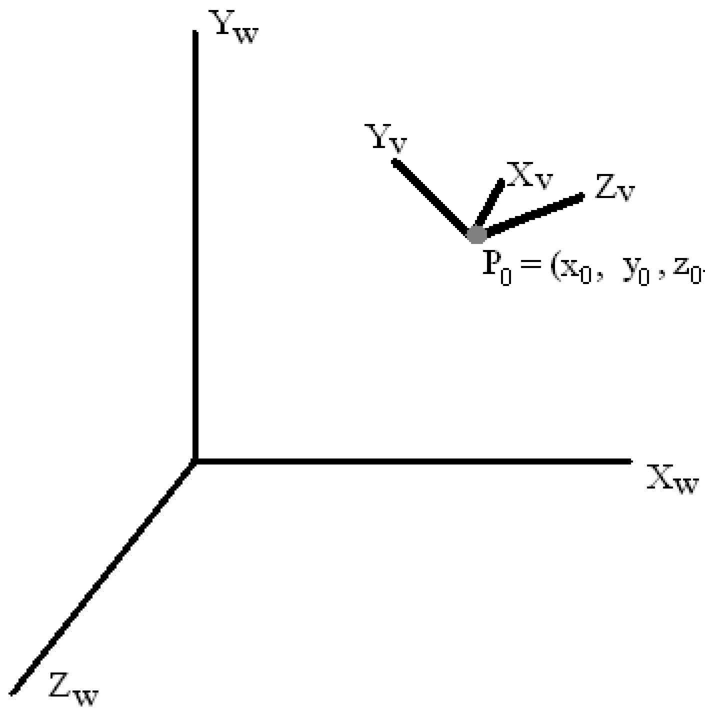
Uma translação, seguida de 3 rotações:

$$M_{wc,wv} = R_x * R_z * R_y * T$$

Transformação – WCS → VCS

Sistemas de Coordenadas do Mundo (WCS) – X_w, Y_w, Z_w

Sistemas de Coordenadas de Viewing (VCS) – X_v, Y_v, Z_v



Matriz de rotação direta

Alternativa: Calcular a matriz de rotação direta: R.

$$M_{wc,wv} = R * T$$

$$n = \frac{N}{|N|} = (n_{1,} n_{2,} n_3)$$

$$u = \frac{V \times N}{|(V \times N)|} = (u_{1,} u_{2,} u_3)$$

$$v = n \times u = (v_{1,} v_{2,} v_3)$$

$$R = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

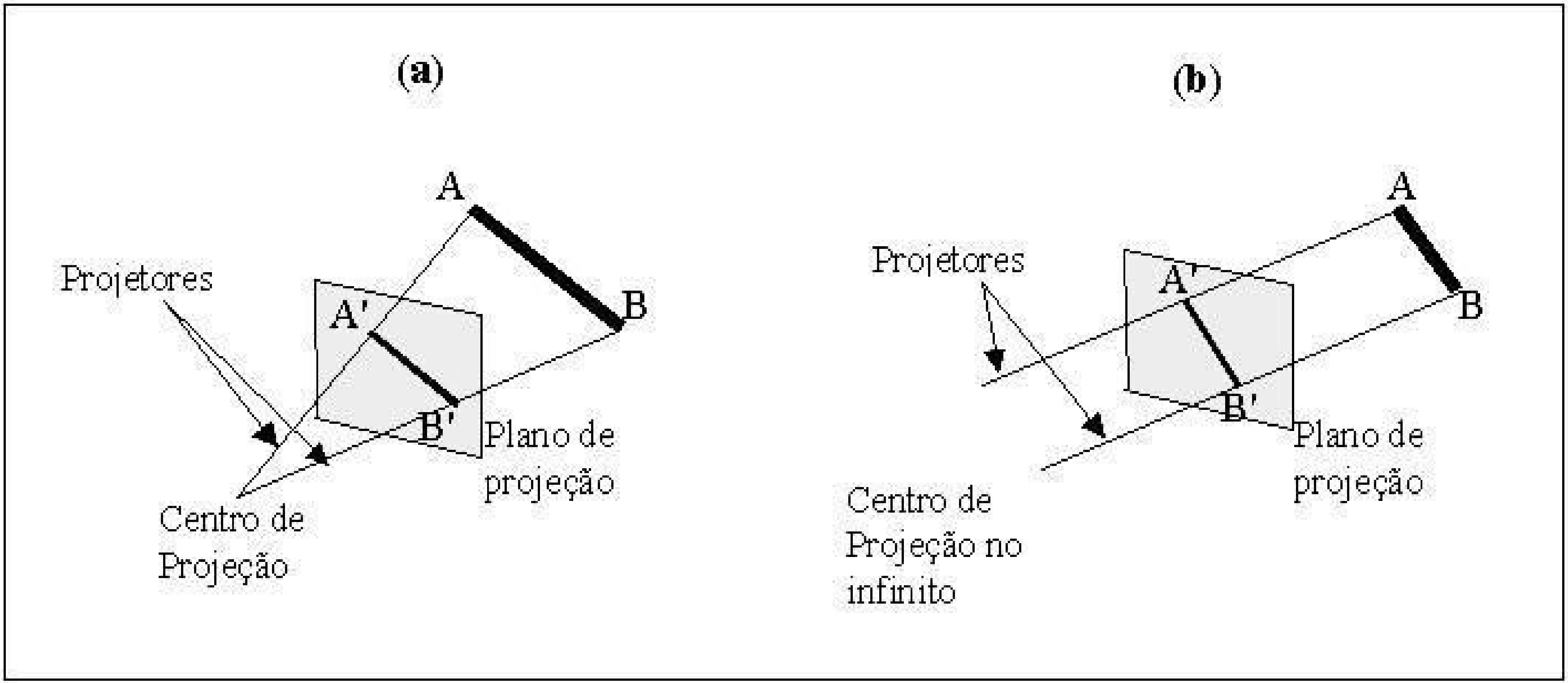
Matriz resultante - *Mwc,wv*

$$R = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{wc,vc} = R \cdot T = \begin{bmatrix} u_1 & u_2 & u_3 & -\boldsymbol{u} \cdot P_0 \\ v_1 & v_2 & v_3 & -\boldsymbol{v} \cdot P_0 \\ n_1 & n_2 & n_3 & -\boldsymbol{n} \cdot P_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projeção

Projeção Perspectiva x Projeção Paralela



Fonte Figura: Computer Graphics: C version. Cap. 12, Hearn & Baker

Projeção Paralela x Perspectiva

Projeção Perspectiva

- Objetos ficam menores a medida que se afastam do plano de projeção.
- Retas paralelas que não são paralelas com o plano de visão, se encontram em um Ponto de Fuga (PF)
- Visão mais realista

Projeção Paralela

- Tamanhos dos objetos são mantidos independente da distância.
- Retas paralelas continuam paralelas quando projetadas.
- Visão menos realista

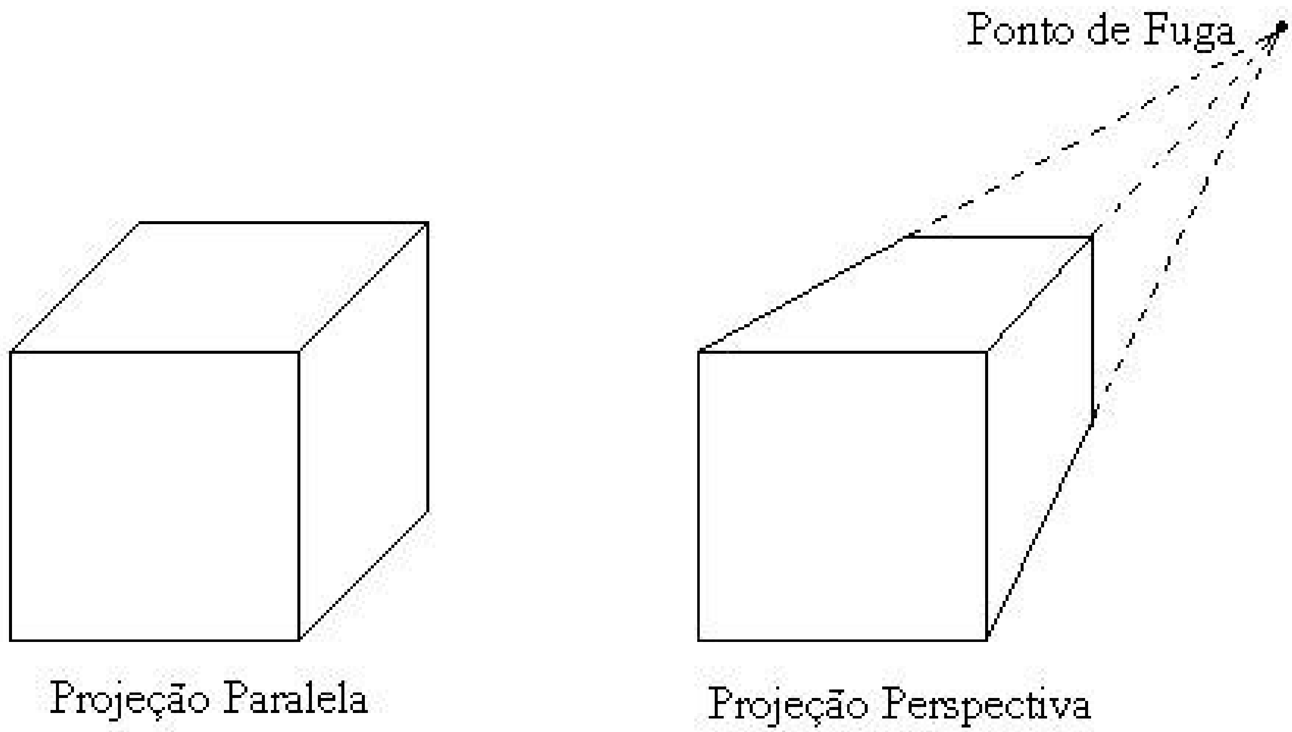
Projeção Perspectiva

Outras características:

•Objetos posicionados após o centro de projeção, são exibidos invertidos na horizontal e na vertical.

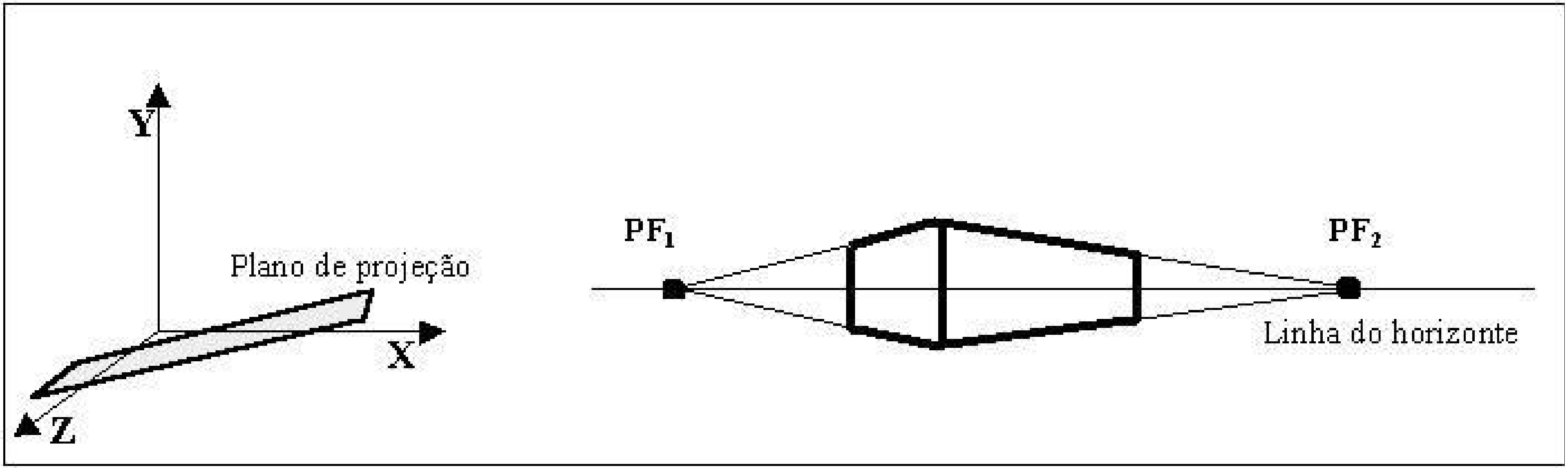
•Pontos contidos no plano que contém o centro de projeção, e que é paralelo ao plano de visão, são projetados no infinito.

Projeção Paralela x Perspectiva



Projeção Perspectiva

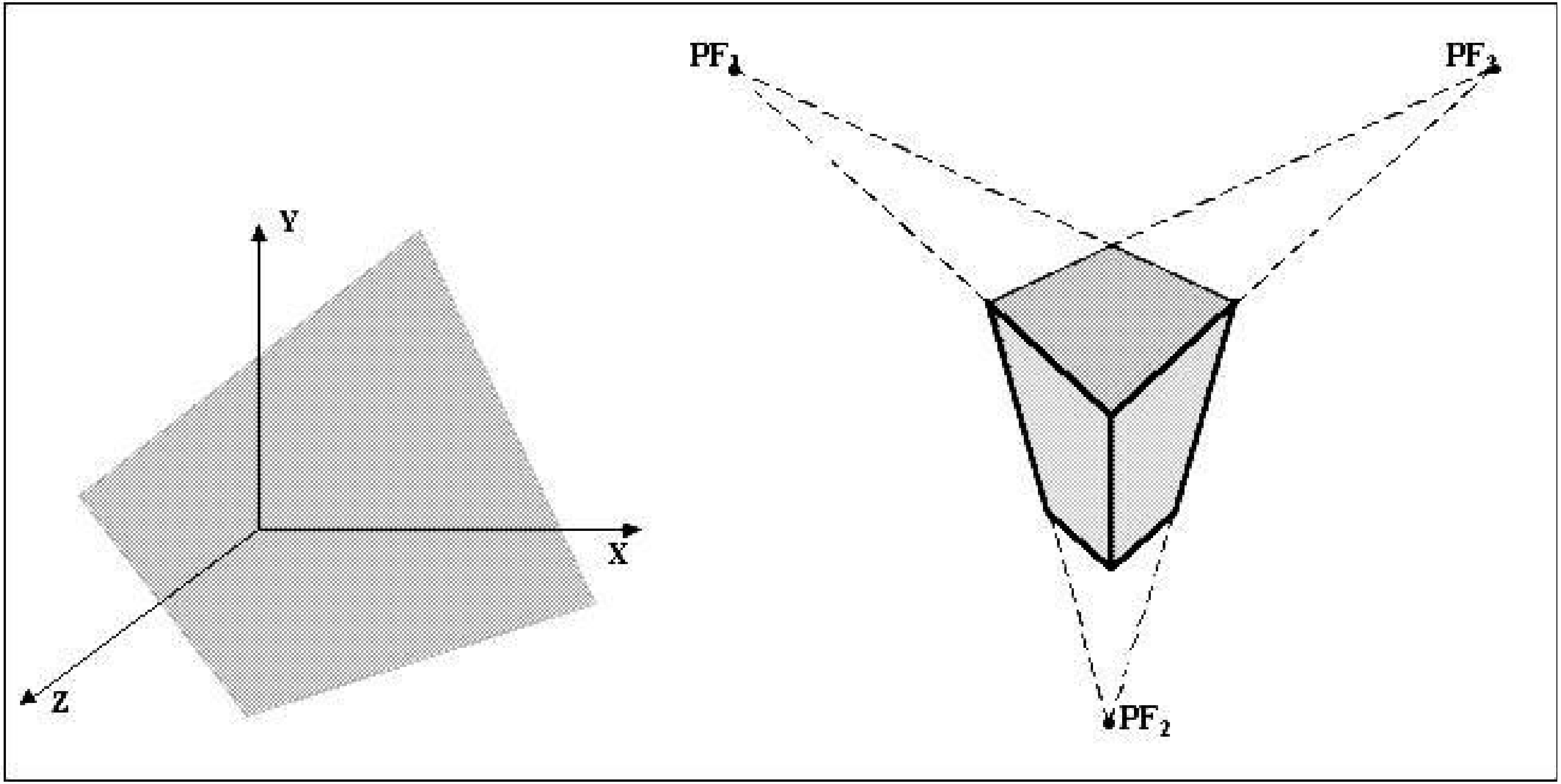
2 pontos de fuga



Fonte Figura: Computer Graphics: C version. Cap. 12, Hearn & Baker

Projeção Perspectiva

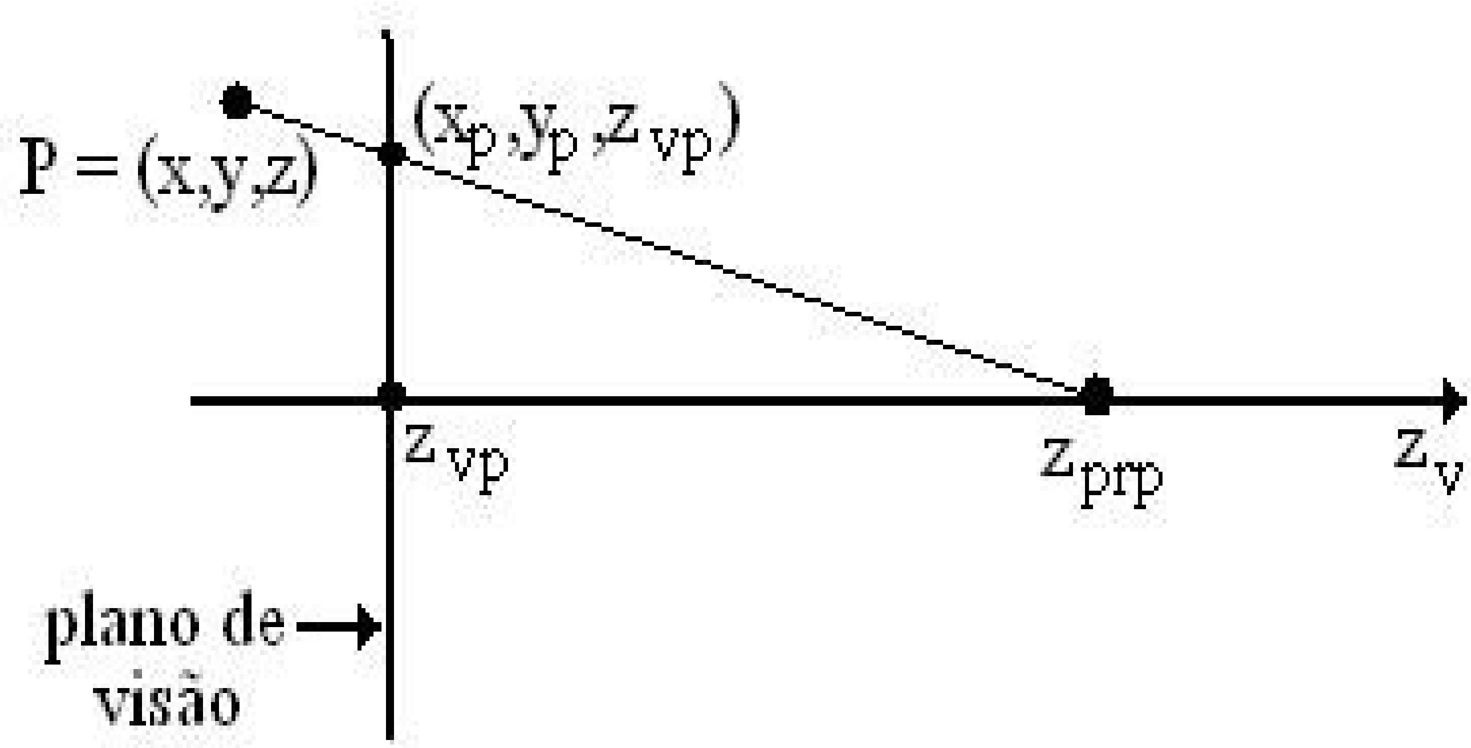
3 pontos de fuga



Fonte Figura: Computer Graphics: C version. Cap. 12, Hearn & Baker

Projeção Perspectiva – Matriz de Transformação

Como determinar as coordenadas em que a reta corta o plano de visão?



Projeção Perspectiva – Matriz de Transformação

Equações paramétricas da reta:

$$x' = x - xu$$
$$y' = y - yu$$
$$z' = z - (z - z_{prp})u$$

A reta corta o plano em $z' = z_{vp}$

Substituindo na última equação, encontramos u:

$$u = \frac{z_{vp} - z}{z_{prp} - z}$$

Projeção Perspectiva – Matriz de Transformação

Substituindo u, temos:

$$x_p = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = x\left(\frac{d_p}{z_{prp} - z}\right)$$

$$y_p = y\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = y\left(\frac{d_p}{z_{prp} - z}\right)$$

Sendo d_p a distância do plano de visão ao centro de projeção: $d_p = z_{prp} - z_{vp}$

Projeção Perspectiva – Matriz de Transformação

Na forma matricial:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/d_p & z_{vp}(z_{prp}/d_p) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Sendo *h* o fator homogêneo: $h = \dfrac{z_{prp} - z}{d_p}$

Projeção Perspectiva – Matriz de Transformação

s pontos xp e yp podem ser obtidos dividindo x_h e y_h pelo fator homogêneo *h*:

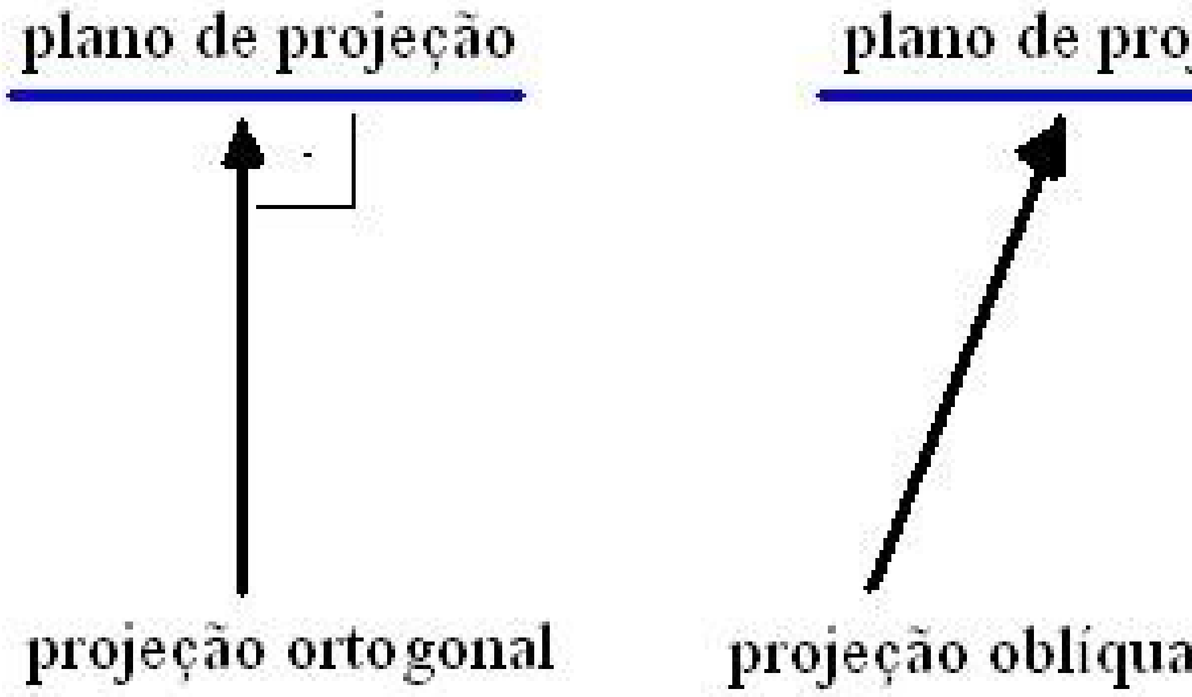
$$x_p = x_h / h$$

$$y_p = y_h / h$$

Sendo h o fator homogêneo: $h = \dfrac{z_{prp} - z}{d_p}$

Projeção Paralela

2 tipos:

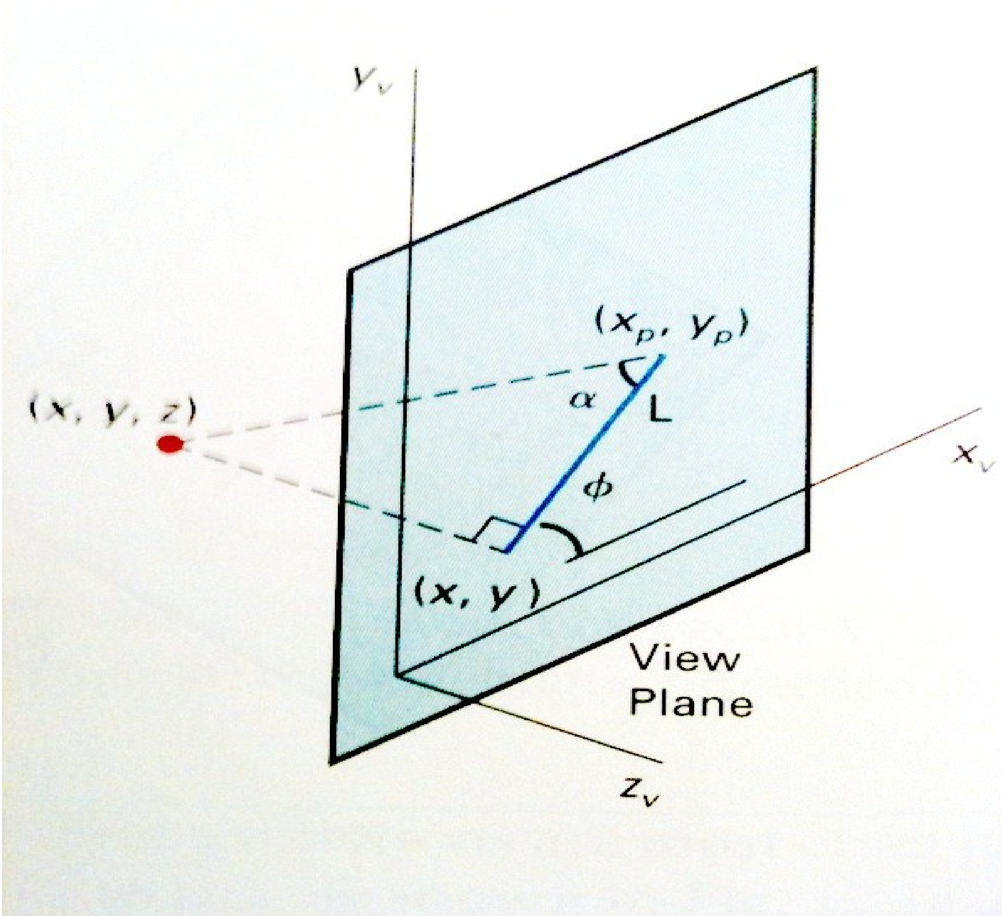


Projeção Ortogonal – Matriz de Transformação

Os pontos no plano de visão são dados pelas coordenadas X_v e Y_v:

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

Projeção Oblíqua



Fonte Figura: Computer Graphics: C version. Cap. 12, Hearn & Baker

Projeção Oblíqua

Podemos calcular (x_p, y_p)

$$x_p = x + L\cos(\phi)$$
$$y_p = y + L\sin(\phi)$$

Sendo:

$$L = \frac{z}{\tan(\alpha)}$$

Projeção Oblíqua – Matriz de Projeção

Reescrevendo para x_p e y_p:

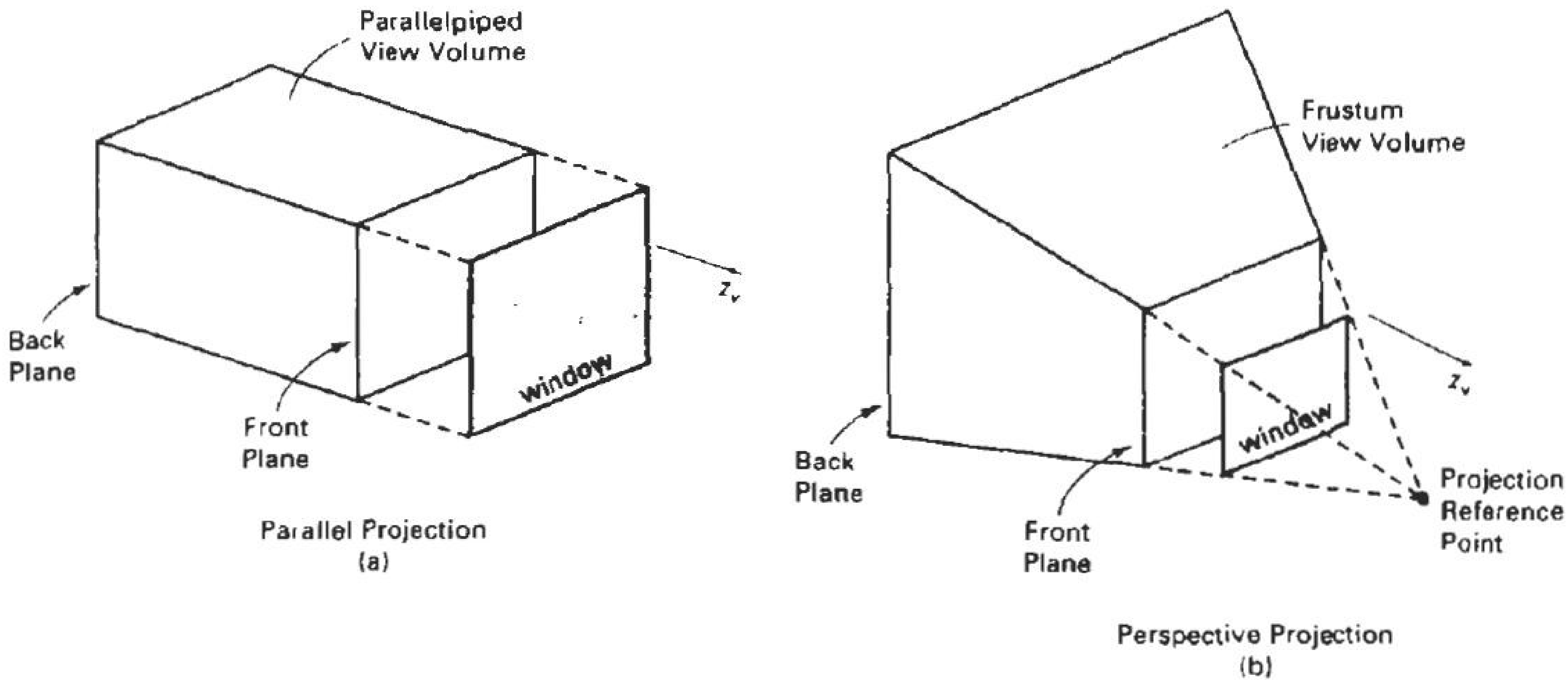
$$x_p = x + \frac{\cos(\phi)}{\tan(\alpha)}$$

$$y_p = y + \frac{\sin(\phi)}{\tan(\alpha)}$$

$$M_{paralela} = \begin{bmatrix} 1 & 0 & \frac{\cos(\phi)}{\tan(\alpha)} & 0 \\ 0 & 1 & \frac{\sin(\phi)}{\tan(\alpha)} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Volume de Visão

Área Visível: Limitada nos eixos *x* e *y* pelo plano de visão, e no eixo *z* por dois planos de recorte:



Fonte Figura: Computer Graphics: C version. Cap. 12, Hearn & Baker

Volume de Visão

A área em forma de um paralelepipedo regular trás vantagens na hora de realizar o recorte.

É possível realizar transformações para que a área seja sempre um paralelepípedo regular, mesmo quando usando projeção oblíqua ou perspectiva.

Transformação geral de projeção paralela

Para transformar o volume de visão obtido através de uma projeção oblíqua em um paralelepípedo regular, é necessário aplicar uma matriz de cisalhamento.

Transformação Geral de Projeção Paralela

Seja o vetor de projeção ***Vp*** = (p_x,p_y,p_z). Devemos encontrar a matriz de cisalhamento que o alinhe com o vetor normal ao plano de visão.

$$V'_p = M_{paralela} \cdot V_p$$

Transformação geral de projeção paralela

$M_{paralela}$ representa um cisalhamento no eixo *z*:

$$M_{paralela} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformação Geral de Projeção Paralela

Obtemos então as seguintes equações:

$$0 = p_x + a p_z$$

$$0 = p_y + a p_z$$

Resultando em:

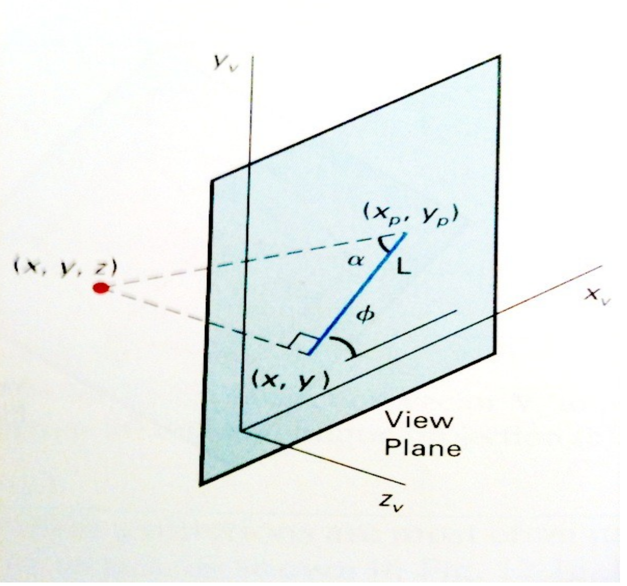
$$a = \frac{-p_x}{p_z} \qquad b = \frac{-p_y}{p_z}$$

Transformação Geral de Projeção Paralela

Por semelhança de triângulos:

$$a = \frac{-p_x}{p_z} = \frac{L\cos\phi}{z}$$

$$b = \frac{-p_y}{p_z} = \frac{L\sin\phi}{z}$$



Podemos obter então a matriz de transformação direta de coordenadas do mundo para coordenadas de projeção, concatenando a matriz $M_{paralela}$ encontrada com a matriz $M_{wc,wv}$

Transformação Geral de Projeção Perspectiva

devemos aplicar duas transformações:

• Cisalhamento do volume de visão, para que a linha central de projeção fique perpendicular ao plano de visão.

• Escala, com um fator de 1/z, para tornar menor objetos mais distantes.

Transformação Geral de Projeção Perspectiva

$$M_{cisalhamento} = \begin{bmatrix} 1 & 0 & a & -az_{prp} \\ 0 & 1 & b & -bz_{prp} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$a = -\frac{x_{prp} - (xw_{min} + xw_{max})/2}{z_{prp}}$$

$$b = -\frac{y_{prp} - (yw_{min} + yw_{max})/2}{z_{prp}}$$

Transformação Geral de Projeção Perspectiva

$$M_{escala} = \begin{bmatrix} 1 & 0 & \dfrac{-x_{prp}}{z_{prp} - z_{vp}} & \dfrac{x_{prp} z_{vp}}{z_{prp} - z_{vp}} \\ 0 & 1 & \dfrac{-y_{prp}}{z_{prp} - z_{vp}} & \dfrac{y_{prp} z_{vp}}{z_{prp} - z_{vp}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \dfrac{-1}{z_{prp} - z_{vp}} & \dfrac{z_{prp}}{z_{prp} - z_{vp}} \end{bmatrix}$$

Podemos obter então a matriz de transformação direta de coordenadas do mundo para coordenadas de projeção, concatenando as matrizes

$M_{cisalhamento}$, M_{escala} e $M_{wc,wv}$

Recorte (*Clipping*)

Um ponto (x,y,z) está fora do limite exercido por um plano se $Ax + By + Cz + D > 0$, e está dentro do limite caso contrário.

Para testar se uma linha está dentro ou fora da região visível, testamos seus dois pontos extremos.

Recorte (*Clipping*)

Caso ambos os pontos extremos de uma linha estejam fora de um plano delimitador, a linha é descartada (está fora do volume de visão). Caso ambos os pontos estejam dentro dos limites de todos os planos delimitadores, a linha é salva.

Recorte (*Clipping*)

A intersecção de uma linha com um plano é encontrada quando a equação $Ax + By + Cz + D = 0$ é satisfeita.

Recorte (*Clipping*)

volume de visão é um paralelepípedo regular → verificação mais simples.

Basta verificar se cada coordenada dos pontos está dentro do intervalo definido pelos planos delimitadores.

Recorte (*Clipping*)

Estratégia: Sejam os planos que limitam o volume de visão, representados pelas coordenadas:

- xv_{min} (esquerda)
- xv_{max} (direita)
- yv_{min} (baixo)
- yv_{max} (cima)
- zv_{min}<(frente)
- zv_{max}(trás)

Recorte (*Clipping*)

Usar 6 bits para identificar a localização de um ponto:

bit 1 = 1, se $x < xv_{min}$

bit 2 = 1, se $x > xv_{max}$

bit 3 = 1, se $y < yv_{min}$

bit 4 = 1, se $y > yv_{max}$

bit 5 = 1, se $z < zv_{min}$

bit 6 = 1, se $z > zv_{max}$

qualquer sequência de bits diferente de 000000 indica um ponto que está fora da região de visão.

Recorte (*Clipping*)

Um segmento de reta está dentro da região visível → pontos extremos → 000000.

Quando pelo menos um dos pontos não possui uma sequência 000000, aplica-se o operador lógico “and”. resultado diferente de 0 → ambos os pontos estão fora de um mesmo limitador

Recorte (*Clipping*)

No caso de não conseguir identificar se o segmento de reta está completamente dentro ou fora do volume, deve-se testar utilizando pontos de intersecção com os planos limitantes.

Recorte (*Clipping*)

Para encontrar os pontos de intersecção com os planos limitantes, substitui-se o valor da coordenada daquela plano na equação apropriada:

$$x = x_1 + (x_2 - x_1)u\,,$$

$$y = y_1 + (y_2 - y_1)u\,,$$

$$z = z_1 + (z_2 - z_1)u\,,$$

Bibliografia

- Capítulo 12 Hearn & Baker
- Nate Robins: Main OpenGL Chronicles Allies: http://www.xmission.com/~nate/tutors.html
- Geometria Analítica: um tratamento Vetorial, Paulo Boulos, Ivan de Camargo
- Schröeder, The Visualization Toolkit, 1998

Agradecimentos

Prof. Rosane Minghim

Danilo Medeiros Eler