

Viewing Pipeline 3D

Maria Cristina F. de Oliveira
Rosane Minghim
2010

Viewing (onde estamos no pipeline)

- Pipeline
 - Geração dos modelos do objetos
Modelagem (*modeling*)
(geometria + topologia)
 - Posicionamento na Cena e Descrição da Cena (*Viewing*)
 - Rendering
(Geração da Imagem)

Viewing

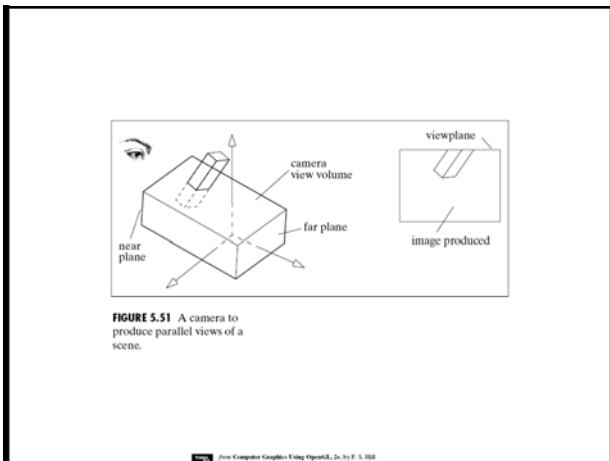
- 2D: Modelo da Cena 2D/Janela 2D/recorte 2D/viewport
- 3D: Posicionamento da câmera, volume de visualização, recorte 3D, projeção, *viewport*
 - outros elementos, como iluminação, remoção de superfícies ocultas, *depth cueing*
 - *Depth cueing* = percepção de profundidade: uso de *shading*, *textura*, *cor*, *fog*, etc.) para dar uma indicação da distância de um objeto ou superfície em relação ao observador
 - <http://www.vis.uni-stuttgart.de/depthcue/>

Viewing Pipeline (OpenGL)

- Todos os vértices, de todos os objetos, entram no 'pipeline gráfico'
- Cada vértice é processado por três matrizes:
 - Modelview
 - Projeção
 - Viewport

FIGURE 5.52 The OpenGL pipeline (slightly simplified).

FIGURE 5.50 Simple viewing used in OpenGL for 2D drawing.



Viewing Pipeline 3D

- No caso 3D, o *pipeline* requer:
 - A definição de um volume de interesse na cena 3D (SRV)
 - O mapeamento de seu conteúdo para o SRV (transformação de visualização)
 - A projeção do conteúdo do volume de interesse em um plano (transformação de projeção)
 - Mapeamento da janela resultante na *viewport* normalizada e depois para coordenadas do dispositivo

8

Viewing Pipeline 3D: Analogia Câmera

- Imaginamos um observador que vê a cena através das lentes de uma câmera virtual
 - "fotógrafo" pode definir a posição da câmera, sua orientação e ponto focal, abertura da lente...
 - câmera real obtém uma projeção de parte da cena em um plano de imagem 2D (o filme)
- Analogamente, a imagem obtida da cena sintética depende de vários parâmetros que determinam como esta é projetada para formar a imagem 2D no monitor
 - posição da câmera, orientação e ponto focal, tipo de projeção, posição dos "planos de recorte" (*clipping planes*), ...

9

Viewing Pipeline 3D: Analogia Câmera

- Três parâmetros definem completamente a câmera
 - Posição: aonde a câmera está
 - Ponto focal: para onde ela está apontando
 - Orientação: controlada pela posição, ponto focal, e um vetor denominado *view up*
- Outros parâmetros
 - Direção de projeção: vetor que vai da posição da câmera ao ponto focal
 - Plano da imagem: plano no qual a cena será projetada, contém o ponto focal e, tipicamente, é perpendicular ao vetor direção de projeção

10

Viewing Pipeline 3D: Analogia Câmera

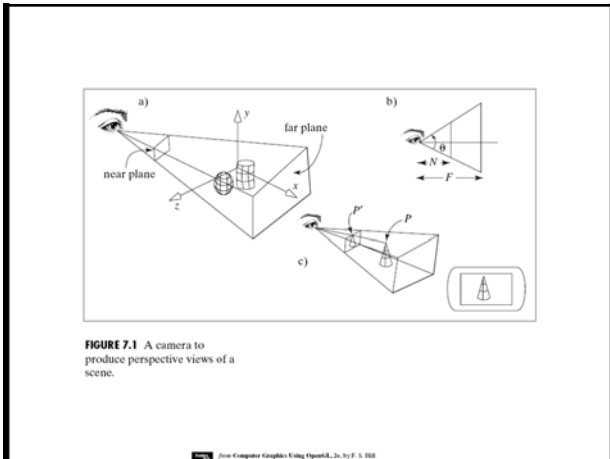
	With a Camera	With a Computer
Observação:	posiciona câmera	posiciona volume de observação
Cena:	posiciona modelo	posiciona modelo
Projeção:	escolhe lentes	escolhe formato vv
Viewport:	escolhe tamanho foto	escolhe porção da tela

fonte: curso CG Arizona State University, Dianne Hansford

Viewing Pipeline 3D: Analogia Câmera

Fonte Figura: Schröder, The Visualization Toolkit, 1998

12



Viewing Pipeline 3D: Analogia Câmera

- O método de projeção controla como os objetos da cena (atores) são mapeados no plano de imagem
 - Projeção ortográfica, ou paralela: processo de mapeamento assume a câmera no infinito, i.e., os raios de luz que atingem a câmera são paralelos ao vetor de projeção
 - Projeção perspectiva: os raios convergem para o ponto de observação, ou centro da projeção. Nesse caso, é necessário determinar o ângulo de visão da câmera
 - Os planos de recorte delimitam a região de interesse na cena
 - Anterior (near plane): elimina objetos muito próximos da câmera
 - Posterior (far plane): elimina objetos muito distantes

14

Viewing Pipeline 3D

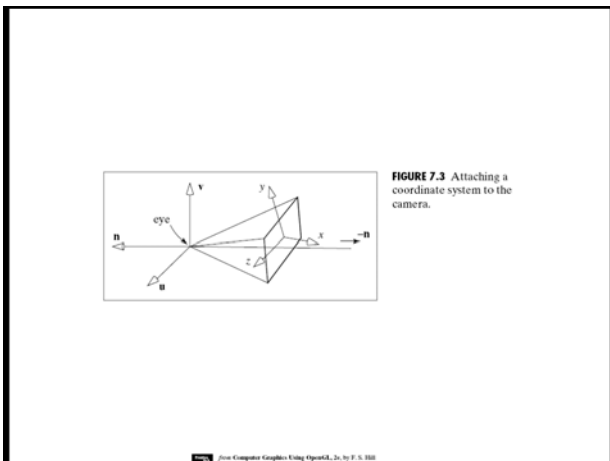
- Retomando: o *pipeline* requer a transformação da cena especificada no SRU para o SRV (ou VCS)
 - O SRV descreve a cena como vista pela câmera...
 - O primeiro passo nesse processo consiste em especificar o SRV. Como?
 - Necessário especificar origem e os três eixos de referência...

15

Especificação do SRV

- Origem do sistema
 - Posição da câmera (VRP: View Reference Point, ou PRO)
- Associados à câmera:
 - Vetor direção de projeção (**N**), que dá a direção do ponto focal, e vetor view-up (**V**), que indica o 'lado de cima' da câmera (ambos devem ser perpendiculares entre si!)
 - Plano de imagem, no qual a cena 3D será projetada, perpendicular ao vetor direção de projeção
- Eixos:
 - eixo z associado ao vetor direção de projeção, eixo y associado ao vetor view-up, eixo x...

16



Especificação do SRV

- Dados os vetores **N** e **V**, os vetores unitários podem ser calculados como indicado ao lado

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = (n_1, n_2, n_3)$$

$$\mathbf{u} = \frac{\mathbf{V} \times \mathbf{N}}{|\mathbf{V} \times \mathbf{N}|} = (u_1, u_2, u_3)$$

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} = (v_1, v_2, v_3)$$

18

Conversão SRU->SRV

- Temos 2 espaços vetoriais (sist. coordenadas) em \mathbb{R}^3 , definidos por duas bases ortonormais
 - SRU, espaço x_w, y_w, z_w (i, j, k)
 - SRV, espaço x_v, y_v, z_v (u, v, n)
- Para transformar a descrição geométrica dos objetos do SRU para o SRV: aplicamos a transformação que alinha os eixos do SRV com os eixos do SRU

19

Conversão SRU->SRV

- A origem do SRV está em $P = (x_0, y_0, z_0)$ no SRU
 - translada para coincidir as origens
 - Aplica a matriz de rotação necessária para alinhar os eixos

20

Conversão SRU->SRV

- Matriz de translação (alinha a origem do SRV a do SRU)

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

21

Conversão SRU->SRV

- Matriz de rotação (alinha os eixos do SRV aos do SRU) é dada por

$$\mathbf{R} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

22

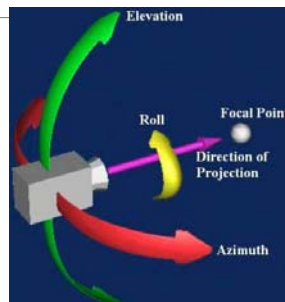
Conversão SRU->SRV

- A matriz completa de transformação é

$$\mathbf{M}_{SRU, SRV} = \mathbf{R} \cdot \mathbf{T} = \begin{bmatrix} u_x & u_y & u_z & -\mathbf{u} \cdot \mathbf{P}_0 \\ v_x & v_y & v_z & -\mathbf{v} \cdot \mathbf{P}_0 \\ n_x & n_y & n_z & -\mathbf{n} \cdot \mathbf{P}_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

23

Manipulação da Câmera



Fonte Figura: Schröder, The Visualization Toolkit, 1998

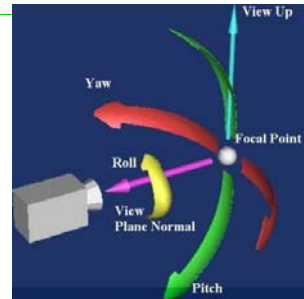
24

Manipulação da Câmera

- *Azimuth*: rotaciona a posição da câmera ao redor do seu vetor *view up*, com centro no ponto focal
- *Elevation*: rotaciona a posição ao redor do vetor dado pelo produto vetorial entre os vetores *view up* e direção de projeção, com centro no ponto focal
- *Roll (Twist)*: rotaciona o vetor *view up* em torno do vetor normal ao plano de projeção

25

Manipulação da Câmera



Fonte Figura: Schröder, The Visualization Toolkit, 1998

26

Manipulação da Câmera

- *Yaw*: rotaciona o ponto focal em torno do vetor *view up*, com centro na posição da câmera
- *Pitch*: rotaciona o ponto focal ao redor do vetor dado pelo produto vetorial entre o vetor *view up* e o vetor direção de projeção, com centro na posição da câmera
- *Dolly (in, out)*: move a posição ao longo da direção de projeção (mais próximo ou mais distante do ponto focal)
- *Zoom (in, out)*: altera o ângulo de visão, de modo que uma região maior ou menor da cena fique potencialmente visível

27

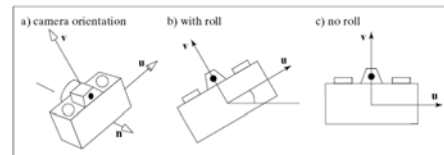
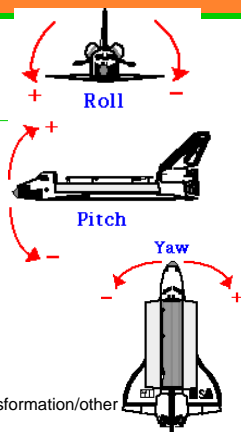
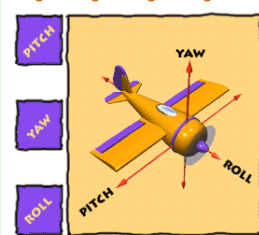


FIGURE 7.6 Various camera orientations.

From Computer Graphics Using OpenGL, 2e, by F. S. Hill

Outra visão



Fonte:
<http://escience.anu.edu.au/lecture/cg/Transformation/other/Rotation.en.html>

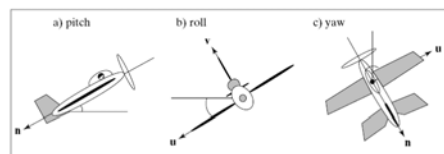


FIGURE 7.4 A plane's orientation relative to the "world."

From Computer Graphics Using OpenGL, 2e, by F. S. Hill

FIGURE 5.52 The OpenGL pipeline (slightly simplified).

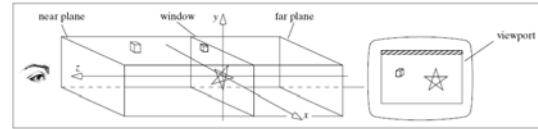
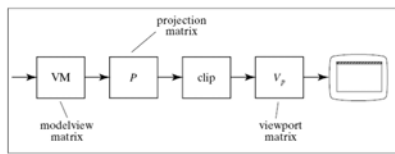


FIGURE 5.50 Simple viewing used in OpenGL for 2D drawing.

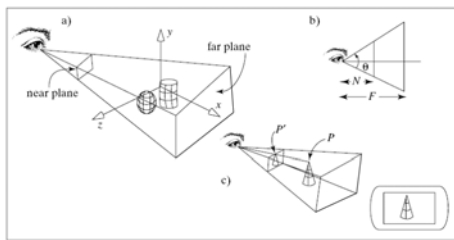


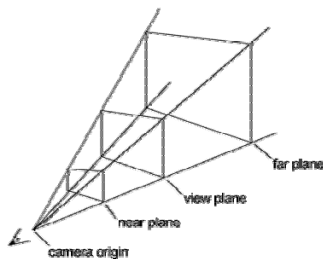
FIGURE 7.1 A camera to produce perspective views of a scene.

Transformação de Projeção

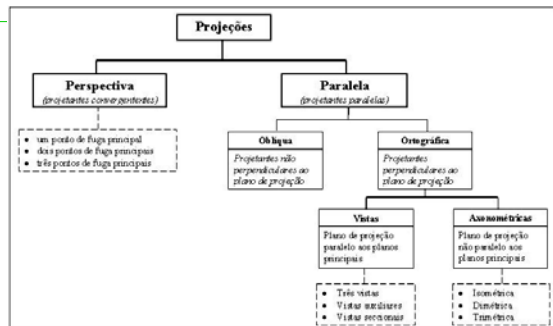
- Tendo a cena descrita no SRV, o próximo passo no *pipeline* consiste em projetar o conteúdo do volume de visualização no plano de imagem
 - Volume de visualização: 'viewing frustum': define a região de interesse na cena
 - Antes da projeção é aplicado um processo de 'recorte' (*clipping*), em que as partes dos objetos que estão fora do VF são descartadas
 - Recorte 3D – em relação aos planos de recorte (*clipping planes*)

Viewing Frustum

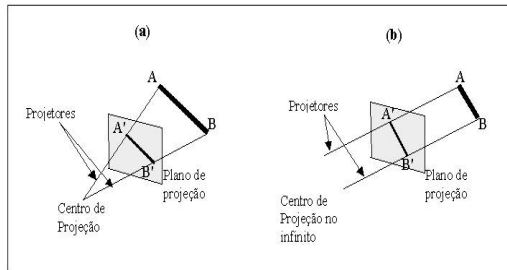
Volume de visualização, projeção perspectiva



Taxonomia das projeções

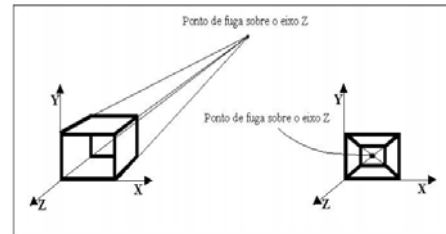


Projeções paralela e perspectiva



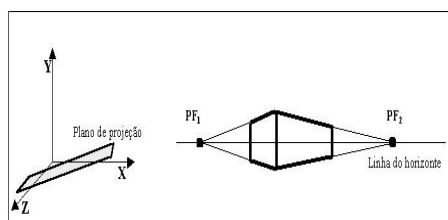
37

Projeção perspectiva um ponto de fuga



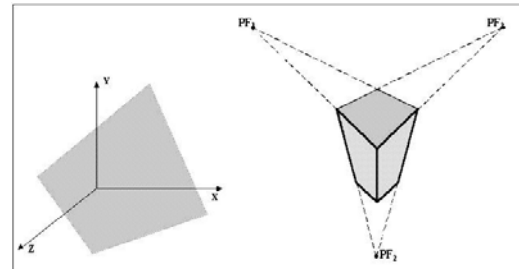
38

Projeção perspectiva dois pontos de fuga



39

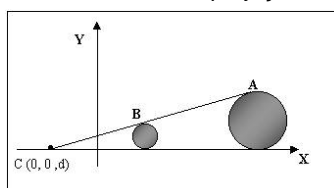
Projeção perspectiva três pontos de fuga



40

Características da Perspectiva

- Encurtamento perspectivo
 - Objetos ficam menores a medida que se distanciam do centro de projeção

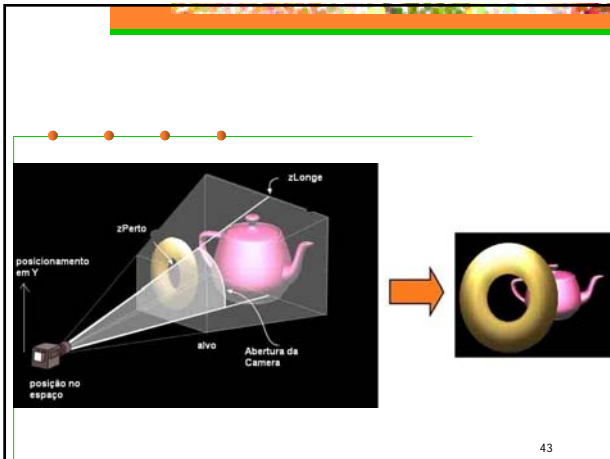


41

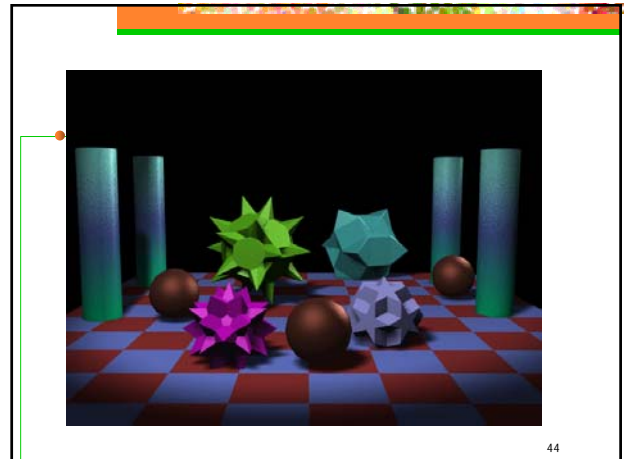
Características da Perspectiva

- Pontos de Fuga
 - Retas não paralelas ao plano de projeção parecem se interceptar em um ponto no horizonte
- Confusão Visual
 - Objetos situados atrás do centro de projeção são projetados de cima para baixo e de trás para a frente
- Distorção Topológica
 - Pontos contidos no plano paralelo ao plano de projeção que contém o centro de projeção são projetados no infinito

42



43



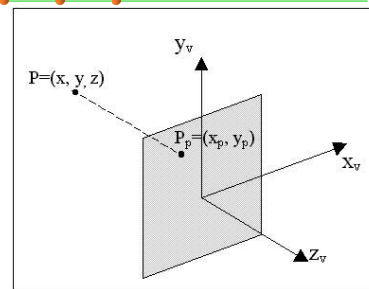
44

Transformação de Projeção

- PRP: *Projection Reference Point*
 - o centro de projeção...
 - Alguns sistemas assumem que coincide com a posição da câmera (a origem do SRV)
- Problema
 - determinar as coordenadas (x_p, y_p, z_p) de um ponto $P = (x, y, z)$ projetado no plano de projeção (Figura)

45

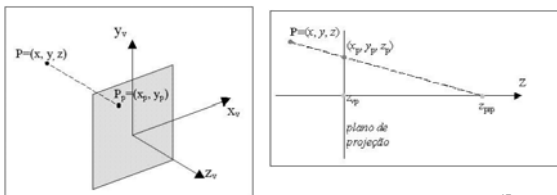
Transformação de Projeção



46

Transformação de Projeção

- Suponha o centro de projeção posicionado em z_{prp} , um ponto no eixo z_v , e que o plano de projeção, normal ao eixo z_v , está posicionado em z_{vp} , um ponto que intercepta o eixo z_v



47

Transformação de Projeção



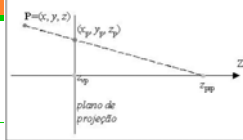
- Coordenadas projetadas (x_p, y_p, z_p) de um ponto (x, y, z) ao longo da linha de projeção

$$\begin{aligned} x_p &= x - x^*u \\ y_p &= y - y^*u \\ z_p &= z - (z - z_{prp})^*u, u \in [0, 1] \end{aligned}$$

- Para $u = 0$ estamos em $P = (x, y, z)$, para $u = 1$ temos o centro de projeção $(0, 0, z_{prp})$.
- No plano de projeção: $z_p = z_{vp}$. Podemos resolver z_p para obter o valor de u nessa posição...

48

Transformação de Projeção



- Valor de u no plano de projeção:

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

- Substituir nas eqs. de x_p e y_p
- Seja d_p : distância do centro de projeção ao plano de projeção, i.e.,

$$d_p = z_{prp} - z_{vp}$$

49

Transformação de Projeção

- Substituindo nas eqs. de x_p e y_p (e w_p)

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

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

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

50

- Fator homogêneo:

$$h = \frac{z_{prp} - z}{d_p}$$

- Normalizar em relação a $w = 1$ (dividir por h) para obter as coordenadas projetadas no plano:

$$x_p = \frac{x_h}{h}, \quad y_p = \frac{y_h}{h}$$

51

Transformação de Projeção

- Na forma matricial homogênea

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

52

Transformação de Projeção

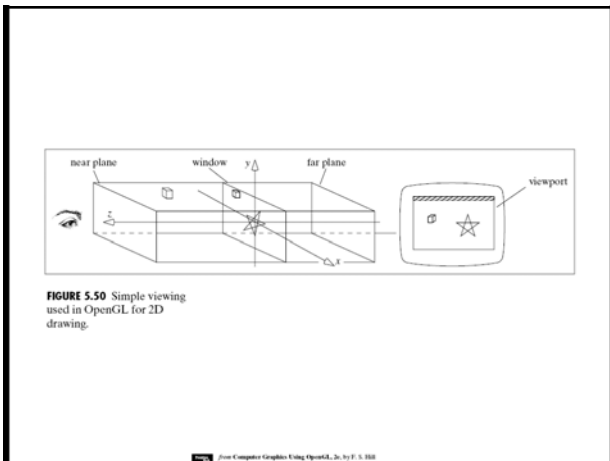
- Observações

- Valor original da coordenada z (no VCS) deve ser mantido para uso posterior por algoritmos de remoção de superfícies ocultas
- Centro de projeção não precisa necessariamente estar posicionado ao longo do eixo z_v . Eqs. podem ser obtidas considerando o centro um ponto qualquer
- Alguns pacotes gráficos assumem $z_{prp} = 0$, i.e., centro de projeção coincide com origem do VCS
- Outro caso especial é $z_{prp} \neq 0$ e o plano de projeção coincidente com o plano $x_v y_v$, i.e., $z_{vp} = 0$ (e $d_p = z_{prp}$)

53

Projeção Paralela Ortográfica

54



Projeções Paralelas

- No caso de projeções ortográficas, matrizes de transformação são triviais
- Ex. projeção no plano x_v, y_v (VCS):

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

56

Paralela vs. Perspectiva

- Projeção perspectiva
 - Tamanho varia inversamente com distância: aparência 'realista'
 - Distâncias e ângulos não são preservados
 - Linhas paralelas não são preservadas
- Projeção paralela
 - Boa para medidas exatas
 - Linhas paralelas são preservadas
 - Ângulos não são preservados
 - Aparência menos realista

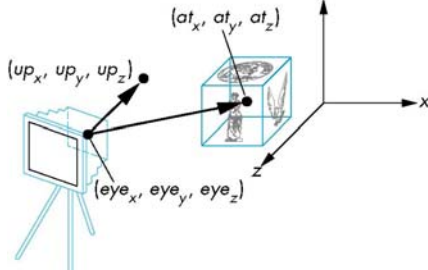
57

No OPENGL

- `glOrtho(left, right, bottom, top, near, far);`
- `glFrustum(left, right, bottom, top, near, far);`
- `gluPerspective(angle, aspect, near, far);`
- `gluLookAt(eyex, eyey, eyez, atx, aty, atz, upx, upy, upz);`

gluLookAt

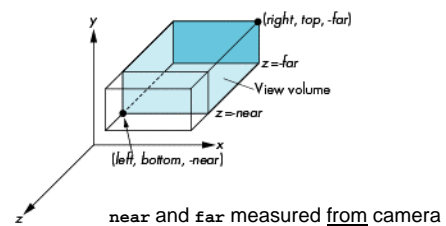
`gluLookAt(eyex, eyey, eyez, atx, aty, atz, upx, upy, upz)`



Fonte: E. Angel, Interactive Computer Graphics, 4a. Ed., Addison-Wesley 2005

OpenGL Orthogonal Viewing

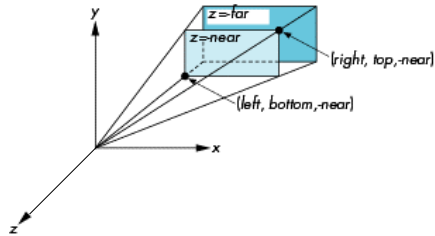
`glOrtho(left, right, bottom, top, near, far)`



Fonte: E. Angel, Interactive Computer Graphics, 4a. Ed., Addison-Wesley 2005

OpenGL Perspective

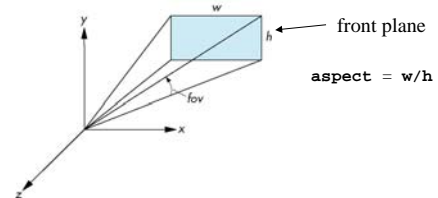
```
glFrustum(left, right, bottom, top, near, far)
```



Fonte: E. Angel, Interactive Computer Graphics, 4a. Ed., Addison-Wesley 2005

Using Field of View

```
gluPerspective(fovy, aspect, near, far)
```



Fonte: E. Angel, Interactive Computer Graphics, 4a. Ed., Addison-Wesley 2005

Bibliografia

- Capitulo 6 da apostila
- Cap. 7 Hearn & Baker
- Cap. 2 Conci e Azevedo
- <http://escience.anu.edu.au/lecture/cg/Transformation/index.en.html>
- Curso CG da ACM (link na pág. GBDI)
- ...