

## 7. Projeções Geométricas e Visualização 3D

Aprendemos a criar e transformar geometricamente objetos 3D, no entanto, nossa janela de visualização é apenas bi-dimensional. Assim, necessitamos desenvolver procedimentos que nos permitam representar objetos 3D nesse espaço 2D, ou seja, no sistema de coordenadas de vídeo. A solução para este problema encontra-se nos métodos de projeção.

Um outro aspecto igualmente fundamental é desenvolvermos a capacidade de criar vistas dos objetos 3D, a partir de qualquer ângulo, ou posição no espaço. A solução para este problema encontra-se na definição dos parâmetros e transformações de visualização.

- **Sistemas de Coordenadas de Visualização**

As transformações de visualização, no seu conjunto, objetivam mapear as superfícies dos objetos em cena para o suporte de exibição. A sua principal finalidade é propiciar a execução mais eficiente das diversas etapas do processo de visualização. Para atingir este objetivo, os objetos devem ser transformados para sistemas de coordenadas (ou espaços de referência) nos quais se torne mais natural e conveniente a realização das tarefas inerentes a cada etapa do processo. Os principais sistemas de coordenadas (ou espaços) envolvidos neste processo são:

**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).

A figura 7.1 mostra a relação entre estes espaços.

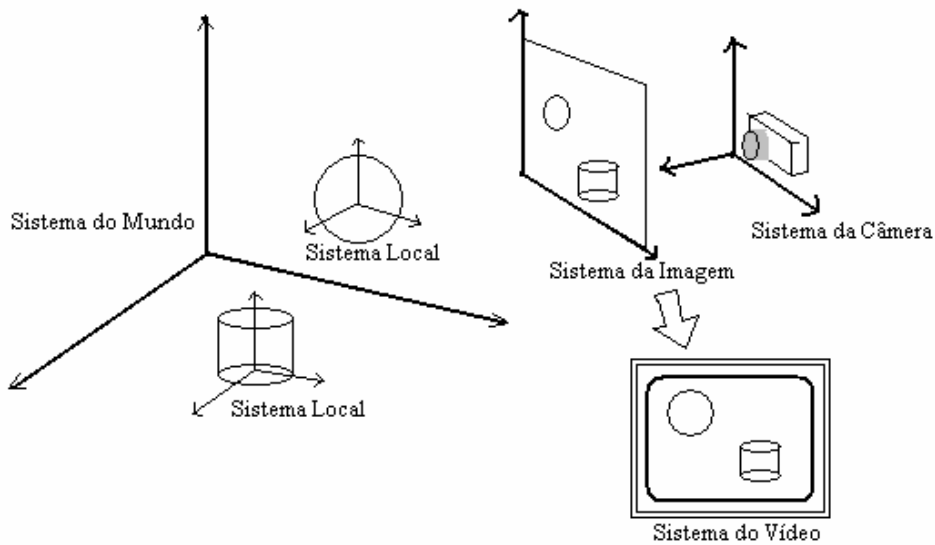


Fig. 7.1 Sistemas de coordenadas.

- Sistema da Câmera e a Projeção Geométrica Planar

As imagens dos objetos distribuídos na cena serão formadas sobre o *plano de projeção* (Sistema da Imagem). Para formar esta imagem são aplicadas operações de *projeção planares*. As figuras 7.2 e 7.3 mostram dois métodos diferentes de projeção de um mesmo objeto, no caso, um segmento de reta. Como nas demais transformações, somente os pontos extremos são projetados e depois conectados. Esta classe de projeção é chamada de *projeção geométrica planar*, em virtude de projetar sobre superfícies planas utilizando-se de projetores (curvas) retilíneos.

As projeções planares podem ser divididas em dois tipos básicos: *perspectiva* e *paralela*. A distinção está na relação entre o centro de projeção e o plano de projeção. Se a distância de um ao outro é finita, então a projeção é perspectiva, se a distância é infinita, então é Paralela.

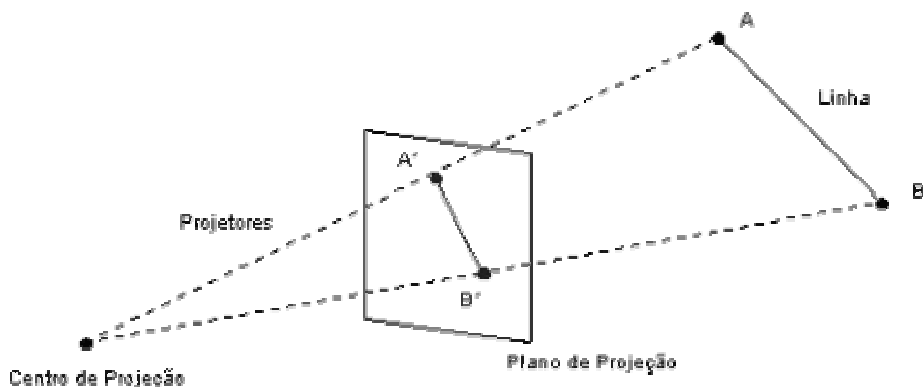


Fig. 7.2 A projeção perspectiva de um segmento de reta.

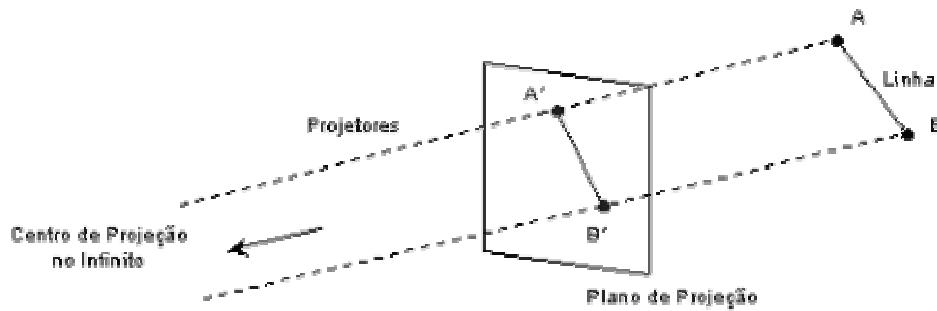


Fig. 7.3 A Projeção paralela de um segmento de linha.

A projeção paralela é assim denominada pelo fato dos projetores serem paralelos, uma vez que o centro de projeção está no infinito. Quando definimos a projeção perspectiva devemos especificar o centro de projeção, já para a projeção paralela, devemos especificar a direção de projeção. A projeção perspectiva é utilizada quando algum grau de realismo visual é desejado, no entanto, não é eficiente quando se deseja manter, ou resguardar, a exatidão das formas e dimensões dos objetos. Na projeção perspectiva, o tamanho da projeção varia inversamente com a distância do objeto ao centro de projeção. Já na projeção paralela, o tamanho da projeção é constante, uma vez que os projetores são retas paralelas.

Estas projeções definem o sistema de coordenadas da câmera, conforme será visto adiante.

- Projeção Perspectiva

O modelo de câmera aqui mostrado é baseado na *câmera de furo* (pinhole camera), mostrado na figura 7.4a: a luz passa através do orifício *O* em um dos lados da caixa, e projeta a imagem do objeto no plano do lado oposto.

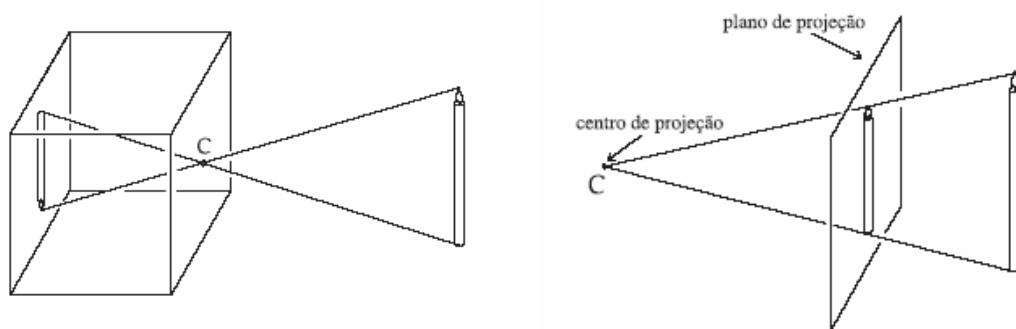


Fig. 7.4 Modelo de câmera pinhole.

Para que a imagem projetada não apareça invertida, o *plano de projeção* será posicionado entre o *centro de projeção*  $C$  e o objeto a ser visualizado (figura 7.4b).

A especificação da câmera virtual requer os seguintes parâmetros:

**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

Outros parâmetros requeridos são:

**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.

Os parâmetros de posição e orientação da câmera servem para situar a câmera no espaço da cena. Já os parâmetros da tela virtual e distâncias anterior e posterior definem o volume de visão da câmera. Este volume limita os objetos que devem ser projetados no plano de projeção para formar a imagem. A figura 7.5 mostra o volume de visão de uma câmera.

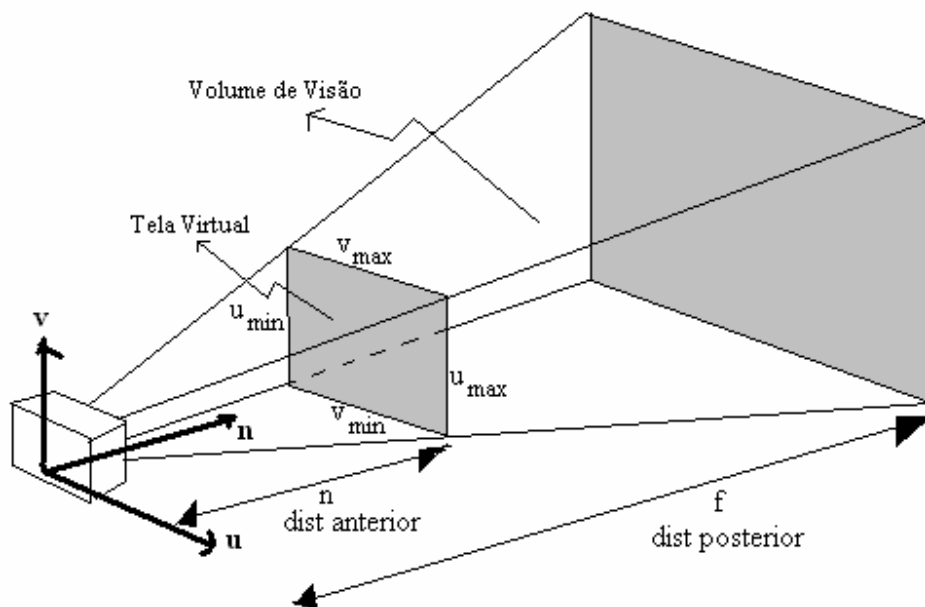


Fig. 7.5 Volume de visão da projeção perspectiva.

Em OpenGL a posição e orientação da câmera são definidos através da função

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

onde (eyex,eyey,eyez) representa a posição da câmera no espaço da cena, (nx,ny,nz) representam a direção para onde a câmera está olhando e (upx,upy,upz) representa a direção de inclinação da câmera.

Já os parâmetros que definem o volume de visão para a projeção perspectiva são especificados pela função

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

onde  $u_{min}$ ,  $u_{max}$ ,  $y_{min}$ ,  $y_{max}$ ,  $n$  e  $f$  são os parâmetros intrínsecos da câmera que definem o volume de visão.

- A Projeção Paralela ou Ortográfica

A forma mais simplista de realizar a projeção paralela seria descartar a coordenada Z. De fato, este é um caso particular do método (projeção ortográfica), onde o plano de projeção é paralelo ao plano XY e os projetores, ou linhas de projeção, são paralelas ao eixo Z.

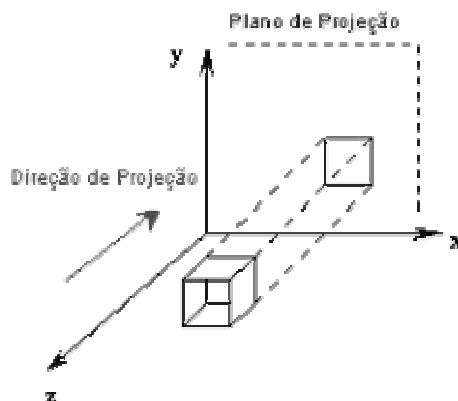


Fig. 7.6 A Projeção paralela ortográfica.

No caso geral de projeção paralela, podemos selecionar qualquer direção para as linhas de projeção, desde que não sejam paralelas ao plano de projeção.

Assim, se a direção de projeção, ou vetor de projeção, é dado por (nx, ny, nz) e o plano de projeção é o plano XY, o nosso problema consiste em determinar a localização (x<sub>2</sub>, y<sub>2</sub>), da projeção de um ponto (x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>) pertencente ao objeto.

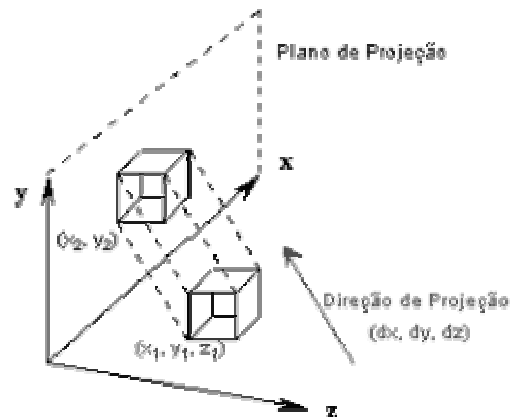


Fig. 7.7 O caso geral de projeção paralela.

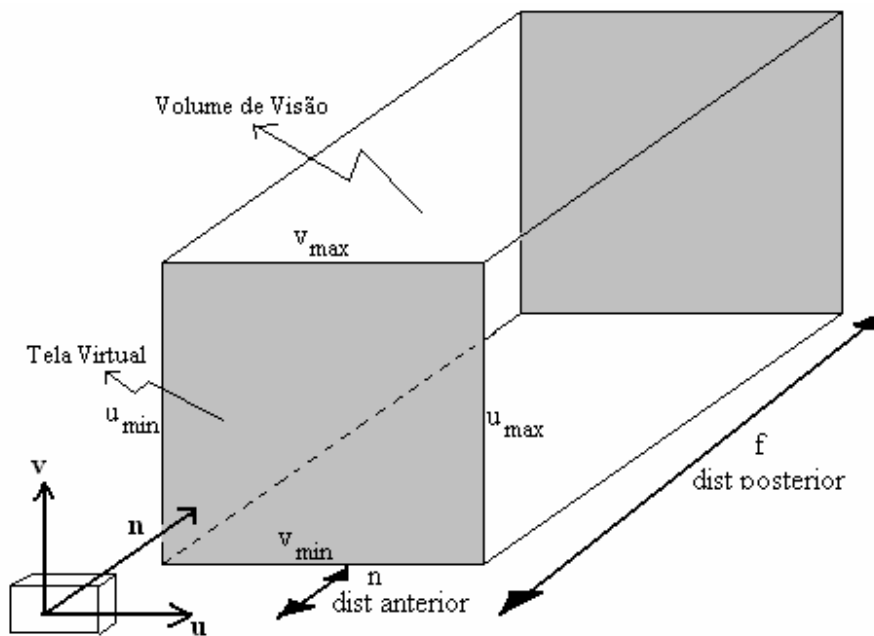


Fig. 7.8 Volume de visão da projeção ortográfica.

Para realizar projeção ortogonal ou paralela em Openg deve usar a função `glOrtho` em vez de `glFrustum`.

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

onde  $u_{min}$ ,  $u_{max}$ ,  $y_{min}$ ,  $y_{max}$ ,  $n$  e  $f$  são os parâmetros intrínsecos da câmera que definem o volume de visão. Perceba que estes parâmetros são os mesmo utilizado em `glFrustum`.