USP – Universidade de São Paulo

ICMC – Instituto de Ciências Matemáticas e de Computação

Computação Gráfica

Notas Didáticas - *Viewing*

Aluno: Marcio Kassouf Crocomo

Professora Doutora Rosane Minghim

São Carlos

Maio 2010

Sumário

USP – Universidade de São Paulo..1
Resumo..3
1. Introdução..4
2 Coordenadas de Viewing..6
2.1 Transformação de coordenadas..8
3. Projeções...9
3.1 Projeção Perspectiva...10
3.2 Projeção paralela..13
4. Volume de visão..15
4.1 Transformação geral de projeção paralela..16
4.1 Transformação geral de projeção perspectiva...17
5. Recorte ou Clipping..19
Bibliografia..22

Resumo

Este documento explica o processo de *viewing* em três dimensões, utilizado na Computação Gráfica. Através de uma analogia deste processo com uma câmera posicionada no espaço para produção de uma foto, este texto aborda os seguintes tópicos: (1) transformações de coordenadas do mundo para coordenadas de *viewing*, (2) projeções paralelas e perspectivas, (3) volumes de visão, (4) transformações de projeção e (5) c*lipping* (recorte).

1. Introdução

Em Computação Gráfica, o processo de *viewing* é o responsável por determinar o que será exibido na tela, e como. Com o objetivo de melhor compreender este processo, podemos realizar uma analogia com o mundo real e uma câmera fotográfica. Para saber o que será exibido em uma foto, existem vários parâmetros que devem ser levados em consideração. Como exemplo, podemos citar:

1. Posicionamento da câmera no espaço. (sobre uma mesa, no chão, encostada na parede, etc.)
2. Direção para onde a câmera aponta. (aponta para a região do espaço em que se encontra um vaso de flor? Para o teto? Para uma pessoa?)
3. Orientação da câmera. (a câmera está de cabeça para baixo? De lado?)

Após um botão ser pressionado, uma parte de todo o cenário é representada no filme da máquina (a foto). Esta analogia com a máquina fotográfica, ajuda a entender a etapa de *viewing* em computação gráfica. A Figura 1 ilustra um exemplo do posicionamento de uma câmera no espaço, assim como sua direção e orientação.

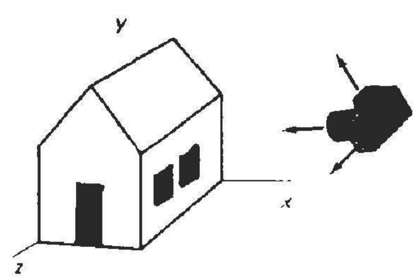

Figura 1 – Posicionamento, direção e orientação de uma câmera no espaço. Fonte : Computer Graphics: C version. Cap. 12, Hearn & Baker

Após termos um modelo 3D, devemos definir como o mesmo será observado pelo usuário. A Figura 2 mostra a sequência das transformações realizadas, representadas pelas setas entre os sistemas de coordenadas. Inicialmente temos nossas coordenadas do modelo 3D que são transformadas para as coordenadas do mundo (onde todos os modelos são posicionados, resultando em uma cena). Em seguida, é realizada uma transformação para coordenadas de *viewing*, que define um sistema de coordenadas a partir do observador da cena (a máquina fotográfica, em nossa analogia). Feito isso, é feita uma transformação para que a imagem observada seja projetada em um plano de visão (a foto, em nossa analogia). A última etapa mostrada pela imagem, é a transformação da imagem para ser exibida em nosso ambiente de trabalho, como por exemplo, uma região do monitor.

Figura 2 – Transformações realizadas entre os sistemas de coordenadas, representadas pelas setas.

Os capítulos a seguir estão organizados da seguinte maneira:

- **2. Coordenadas de *Viewing*** – É explicado como as coordenadas de *viewing* são definidas, e como as transformações são realizadas a partir das coordenadas do mundo real.
- **3. Projeções** – Neste capítulo, é realizada uma introdução aos tipos de projeção existentes (paralela e perspectiva), e em seguida, são explicadas as transformações realizadas para obter esses tipos de projeção.
- **4. Volumes de Visão** – Nesta sessão, é discutida qual é a área potencialmente visível em uma cena. Também são mostradas transformações de projeção que resultam em volumes de visão em um formato que torne mais simples as operações de recorte do próximo capítulo.
- **5. Recorte ou *Clipping*** – São discutidos os procedimentos padrões utilizados para detecção e exclusão dos elementos fora da área de visão da cena, e em seguida, são explicadas técnicas eficientes para executar tais procedimentos.

Informações mais detalhadas sobre partes específicas deste texto podem ser encontradas na bibliografia, após os capítulos listados acima, sendo [1] a principal referência utilizada para a escrita deste documento.

2 Coordenadas de *Viewing*

Considerando que temos nosso sistema de coordenadas do mundo, devemos agora obter nosso sistema de coordenadas de *viewing.* Para isso, devemos definir 3 parâmetros, seguindo nossa analogia com a câmera:

1. A origem de nosso sistema de coordenadas de *viewing*, dado por um ponto que define a posição da câmera no sistema de coordenadas do mundo (P_0).
2. Um ponto focal, que define a direção para onde a câmera aponta.
3. Orientação, dada através de um vetor chamado “view up”.

A partir destes parâmetros, podemos obter os eixos que compõem o sistema de coordenadas de *viewing* da seguinte forma: Obtemos o vetor de projeção (***N***), dado pelo vetor que vai da direção da câmera até o ponto focal, este vetor define nosso eixo Z_v. Em seguida, chamamos Y_v o eixo definido pelo vetor “view up” (***V***). O terceiro eixo, chamado de X_v é dado pela linha perpendicular aos eixos Y_v e Z_v, podendo ser obtido pelo produto vetorial entre ***N*** e ***V*** (vetor ***U***). A orientação do sistema de coordenadas é normalmente calculada utilizando a regra da mão direita, sendo esta regra a mesma utilizada para o sistema de coordenadas do mundo. A Figura 3 mostra o sistema de coordenadas do mundo, definida pelos eixos X_w, Y_w e Z_w, e o sistema de coordenadas de viewing, com os eixos X_v, Y_v e Z_v.

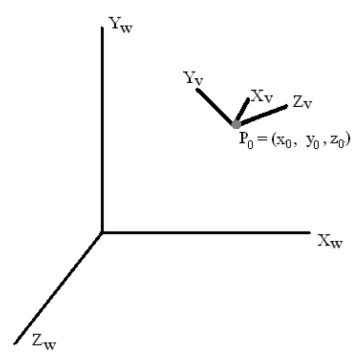

Figura 3 – Sistemas de coordenadas do mundo (X_w, Y_w e Z_w) e sistemas de coordenada de *viewing* (X_v, Y_v e Z_v)

Manipulação da câmera

A câmera pode ser manipulada através da alteração dos parâmetros acima descritos, resultando em operações ilustradas pelas Figuras 4 e 5 [2].

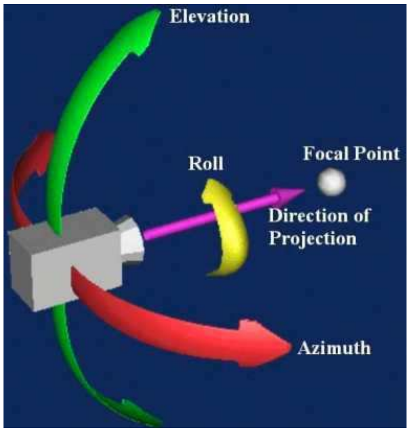

Figura 4 – Funções de Manipulação da Câmera, parte 1. Fonte: Schröeder, The Visualization Toolkit, 1998

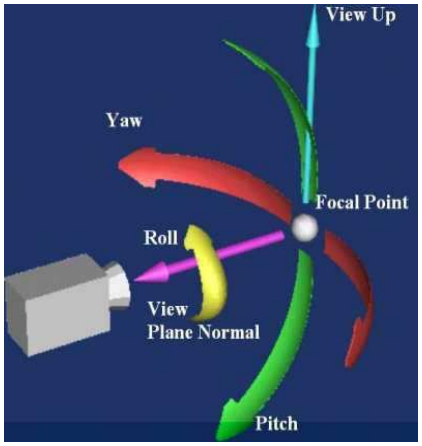

Figura 5 – Funções de Manipulação da Câmera, parte 2. Fonte: Schröeder, The Visualization Toolkit, 1998

• **Azimuth -** rotaciona a posição da câmera ao redor do vetor "*view up*", utilizando o ponto focal como centro da transformação.

• **Elevation -** rotaciona a posição da câmera ao redor do vetor dado pelo produto vetorial entre vetor "view up" e o vetor direção de projeção, utilizando o ponto focal como centro da transformação.

• **Roll (Twist) -** rotaciona o vetor "*view up*" em torno do vetor normal ao plano de projeção.

• **Yaw -** rotaciona o ponto focal da câmera em torno do vetor "*view up*", utilizando a câmera como centro da transformação.

• **Pitch -** rotaciona o ponto focal ao redor do vetor dado pelo produto vetorial entre o vetor *“view up”* e o vetor direção de projeção, utilizando a câmera como centro da transformação.

• **Dolly -** (*in*, *out*) move a posição da câmera ao longo da direção de projeção.

• **Zoom -** altera o ângulo de visão da câmera, de modo que uma região maior ou menor da cena fique na região visível.

Considere agora um plano de visão, perpendicular ao vetor de projeção. Este plano pode ser posicionado em várias posições ao longo do eixo Z_v, sendo comum seu posicionamento no ponto focal já definido. Pode-se pensar neste plano de visão, como sendo o filme da máquina fotográfica, onde a imagem final será projetada. No entanto, antes de projetarmos a cena no plano de visão, devemos realizar a transformação de coordenadas do mundo, para coordenadas de *viewing.*

2.1 Transformação de coordenadas

Queremos 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*. Para tanto é suficiente obter as matrizes de transformações que alinhem o sistema de coordenadas de *viewing* ao sistema de coordenadas do mundo, composto por uma matriz de translação (T), e três matrizes de rotação (uma para cada eixo: R_z, R_x e R_y). Desta forma, nossa matriz seria dada por: $M_{wc,wv} = R_x * R_z * R_y * T$.

Outra forma de calcular a matriz $M_{wc,wv}$, é calculando vetores unitários ($\boldsymbol{u}$,$\boldsymbol{v}$,$\boldsymbol{n}$) que representem o sistema de coordenadas de *viewing*. Com estes vetores, é possível calcular a matriz de rotação direta (R). A partir dos vetores $\boldsymbol{N}$ e $\boldsymbol{V}$ explicados anteriormente, os vetores unitários $\boldsymbol{u}$, $\boldsymbol{v}$ e $\boldsymbol{n}$ podem ser calculados da seguinte forma[1]:

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

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

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

A partir dos vetores unitários acima, a matriz de rotação direta é:

1 Para informações sobre como calcular a norma de um vetor, produto vetorial, e divisão de um vetor por um número escalar, consultar [3].

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

Desta forma, encontramos a matriz de transformação $M_{wc,wv} = R * T$. Lembrando que a matriz de translação utilizada é a que leva a origem do sistema de coordenadas de viewing (x_0, y_0, z_0) até a origem do sistema de coordenadas do mundo, dada por:

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

Portanto, nossa matriz de transformação final é dada por:

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

3. *Projeções*

Após as coordenadas do objeto da cena terem sido transformadas para coordenadas de *viewing*, podemos pensar na projeção dos objetos 3D para o plano de visão, resultando em uma imagem 2D da cena. Existem dois tipos básicos de projeção:

- Projeção paralela
- Projeção perspectiva

Na projeção paralela, as coordenadas do objeto são transformadas para o plano de visão através de retas paralelas. Já a projeção perspectiva, utiliza retas que se encontram em um ponto chamado “centro de projeção”. A Figura 6 mostra um exemplo de projeção paralela e outro de projeção perspectiva.

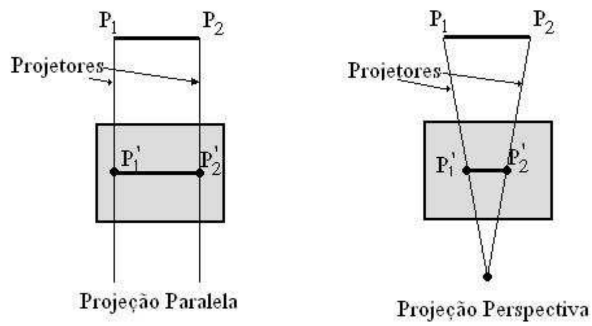

Figura 6 – Projeção paralela e projeção perspectiva.

A projeção paralela deve ser usando quando desejamos preservar as proporções relativas do objeto, no entanto, a projeção perspectiva apresenta um realismo maior. Para melhor entender esta afirmação, compare abaixo a imagem de um cubo, criada usando projeção paralela e projeção perspectiva (Figura 7).

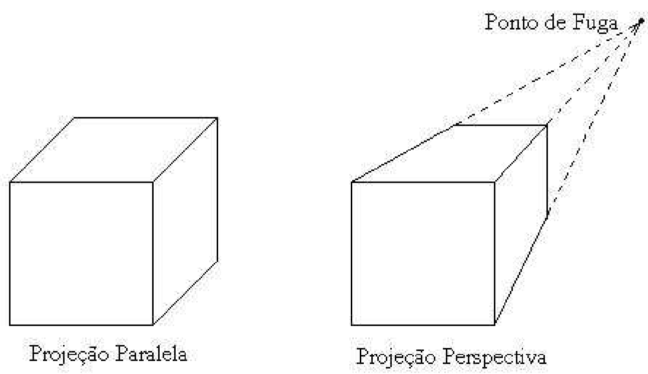

Figura 7 – Imagens resultantes de projeção paralela (esquerda) e projeção perspectiva (direita)

Na projeção paralela, todas as retas do cubo que são paralelas entre si, continuam paralelas na projeção em duas dimensões. Já na projeção perspectiva, as únicas retas do cubo que continuam paralelas na imagem final, são as paralelas ao plano de projeção.

3.1 Projeção Perspectiva

Na projeção perspectiva, as retas não paralelas ao plano de projeção convergem em chamados “pontos de fuga”. Uma imagem pode ter até 3 pontos de fuga (um para cada coordenada), dependendo do seu posicionamento com relação ao plano de visão. A Figura 7 Ilustra uma imagem projetada contendo apenas um ponto de fuga, as Figuras 8 e 9 abaixo mostram imagens contendo 2 e 3 pontos de fuga respectivamente.

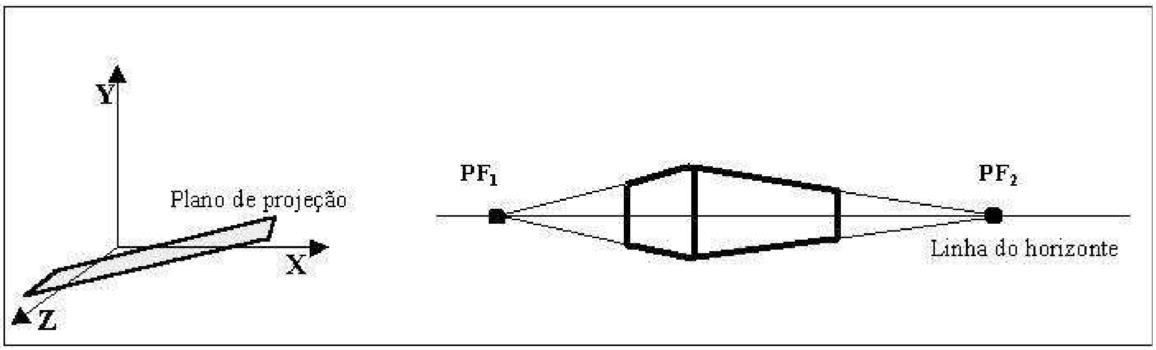

Figura 8 – Projeção perspectiva com dois Pontos de Fuga (PF)

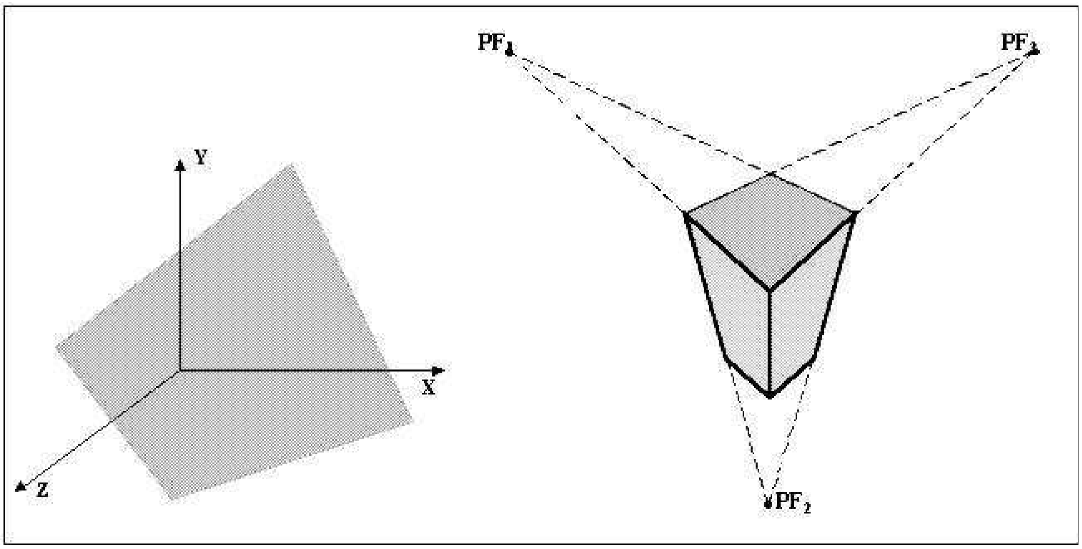

Figura 9 – Projeção perspectiva com três Pontos de Fuga (PF). Fonte: Computer Graphics: C version. Cap. 12, Hearn & Baker

Outras características da projeção perspectiva são:

- Objetos ficam menores a medida que se afastam do plano de projeção.
- 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.

Calculando a matriz de transformação para projeção perspectiva

Considere um centro de projeção no eixo $z_{v,}$ representado por z_{prp}, e um plano de projeção, perpendicular ao eixo z_v, posicionado em z_{vp} como mostra a Figura 10 abaixo.

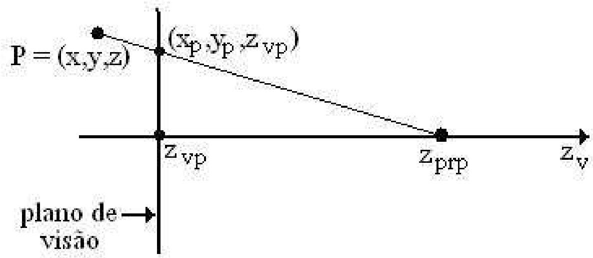

Figura 10 – Projeção perspectiva do ponto P no plano de visão

Queremos calcular a projeção do ponto (x,y,z) no plano de visão, isto é, queremos calcular os valores de x_p e y_p. Para isso, podemos escrever as equações que descrevem as coordenadas dos pontos da linha de projeção:

$$x' = x - xu$$

$$y' = y - yu$$

$$z' = z - (z - z_{prp})u$$

O parâmetro u varia no intervalo de 0 a 1. Quando u=0, temos a posição do ponto sendo projetado (ponto P), e quando u=1, temos a posição do centro de projeção (0,0,z_{prp}). Para encontrar o ponto projetado (x_p,y_p,z_{vp}), devemos substituir o valor conhecido z_{vp} na última equação acima, encontrando assim o valor de u, e substituir u nas equações para x' e y'.

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

Substituindo nas equações para x' e y' 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 $d_p = z_{prp} - z_{vp}$ a distância do plano de visão ao centro de projeção. Usando coordenadas homogêneas, podemos representar a matriz de transformação que representa as equações acima da seguinte forma:

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

Sendo h, o fator homogêneo $h = \frac{z_{prp} - z}{d_p}$. Os pontos x_p e y_p podem ser obtidos dividindo x_h e y_h pelo fator homogêneo:

$$x_p = x_h / h$$

$$y_p = y_h / h$$

3.2 Projeção paralela

Existem dois tipos de projeção paralela: projeção ortogonal e projeção oblíqua. Podemos especificar uma projeção paralela a partir de um vetor de projeção, que nos informa a direção das linhas de projeção. Quando este vetor é perpendicular ao plano de visão, temos a projeção ortogonal, caso contrário, temos uma projeção oblíqua. A Figura 11 mostra exemplo de uma projeção oblíqua, e de uma projeção ortogonal.

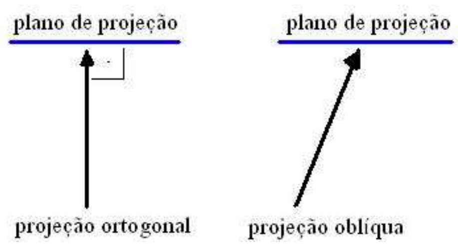

Figura 11 – Vetores indicando uma projeção paralela ortogonal (esquerda) e paralela obíqua (direita)

A transformação de projeção utilizada para projeção ortogonal, é trivial, e dada pela matriz de projeção abaixo:

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

Para calcularmos a matriz de projeção de uma projeção oblíqua, vamos considerar o caso mostrado na Figura 12 abaixo, onde o ponto (x,y,z) está sendo projetado no ponto (x_p,y_p), em nosso plano de visão. Notamos que dois ângulos descrevem esta projeção: o ângulo α, entre o vetor de projeção e o plano de visão, e o ângulo Φ, entre a linha que liga (x,y) a (x_p,y_p) e o eixo x_v.

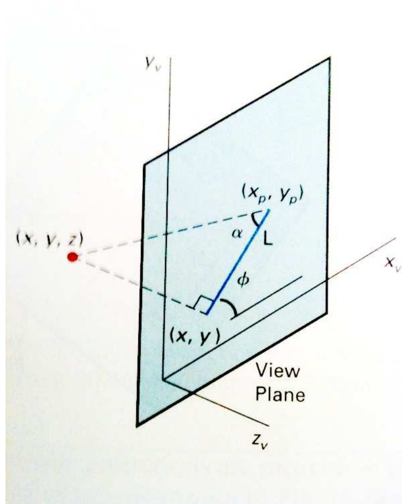

Figura 12 – Projeção do ponto (x,y,z) no plano de visão, em uma projeção paralela oblíqua. Fonte: Computer Graphics: C version. Cap. 12, Hearn & Baker

Calculando a matriz de transformação para projeção oblíqua

Podemos calcular a posição do ponto (x_p,y_p) conforme as equações abaixo:

$$x_p = x + L\cos(\phi)$$

$$y_p = y + L\sin(\phi)$$

sendo:

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

Reescrevendo então as equações para x_p e y_p, temos:

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

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

O que nos dá a matriz de projeção:

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

4. Volume de visão

Na projeção paralela, a área visível é limitada nos eixos *x* e *y* pelos limites da janela do plano de visão, já na projeção perspectiva, os limites em *x* e *y* para a área visível são dados por ângulos verticais e horizontais, representando a abertura da lente da câmera, em nossa analogia. A Figura 12 ilustra diferentes ângulos de abertura, exibidos no sistema de coordenadas do mundo (a esquerda), e a respectiva imagem projetada no plano de visão (a direita).

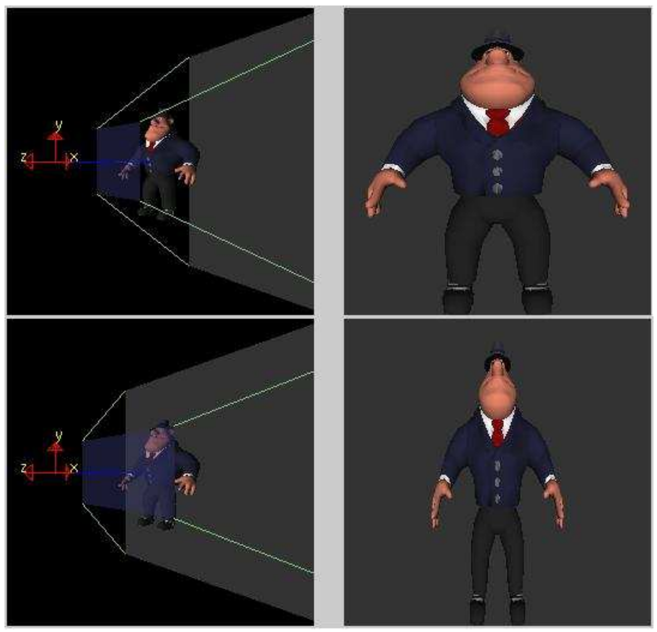

Figura 12 – Ângulos de abertura em uma projeção perspectiva. Figura obtida através do software obtido em [4].

No entanto, se considerarmos apenas os limites em *x* e em *y*, a área visível ainda é infinita. Para limitar a área visível, precisamos definir dois planos de recorte no eixo *z*, um frontal e um traseiro, ambos paralelos ao plano de visão. A Figura 13 mostra áreas visíveis, limitadas por estes planos. Enquanto o volume de visão em uma projeção paralela é um paralelepípedo, o volume limitado pela projeção perspectiva é um tronco de base paralelas (ou *frustum*, em inglês). O plano de visão pode estar em qualquer lugar entre os planos de recorte, desde que o centro de projeção não esteja entre estes planos.

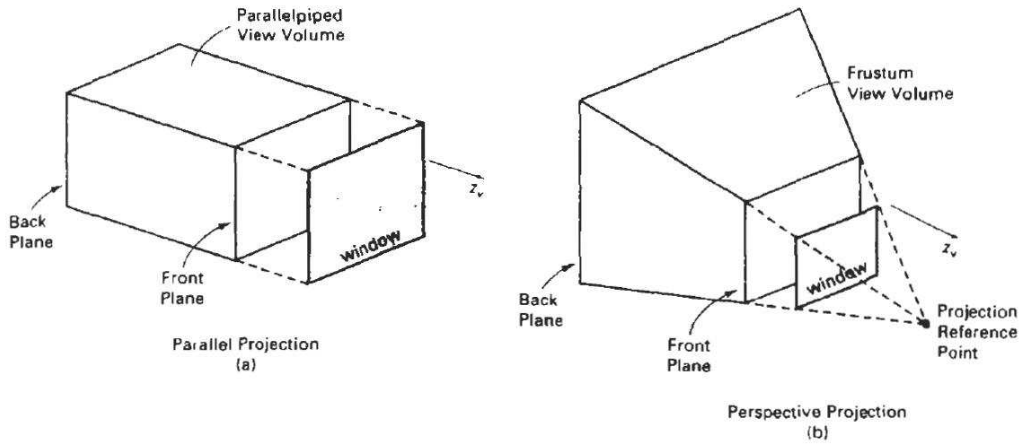

Figura 13 – Volume de visão em uma projeção paralela ortogonal (esquerda) e em uma projeção perspectiva (direita)

Para determinar os objetos que serão ou não exibidos, é mais simples realizar comparações com um volume de visão na forma de um paralelepípedo regular, mas para isso, as transformações vistas até o momento não são suficientes. As sessões a seguir mostram como obter transformações que convertem o volume de visão para um paralelepípedo regular, em projeções oblíquas e perspectivas.

4.1 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. Seja o vetor de projeção $\boldsymbol{V_p} = (p_x, p_y, p_z)$, devemos encontrar a matriz de cisalhamento que alinhe este vetor com o vetor normal ao plano de visão. Esta transformação é representada pela equação abaixo:

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

Onde $M_{paralela}$ é similar a matriz encontrada no capítulo 3.2, e representa um cisalhamento no eixo z:

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

Para obter os parâmetros a e b, podemos substituir $M_{paralela}$ na equação anterior, obtendo as seguintes equações:

$$0 = p_x + ap_z$$

$$0 = p_y + bp_z$$

Resultando em:

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

Observando a figura 12 do capítulo 3.2, podemos encontrar os valores de a e b por semelhança de triângulos, dado por:

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

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

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 $M_{wc,wv}$ encontrada no capítulo 2.

4.1 Transformação geral de projeção perspectiva

Para obter um volume de visão em forma de um paralelepípedo regular usando a projeção perspectiva, devemos aplicar duas transformações:

1. Cisalhamento do volume de visão, para que a linha central de projeção fique perpendicular ao plano de visão.
2. Escala, com um fator de 1/z, para tornar menor objetos mais distantes.

Para obter a matriz do primeiro passo, devemos combinar uma matriz de cisalhamento com uma matriz de translação, para que a linha central de projeção fique perpendicular ao plano de visão, também passando pelo ponto central da janela de visão. Como mostrado na Figura 14.

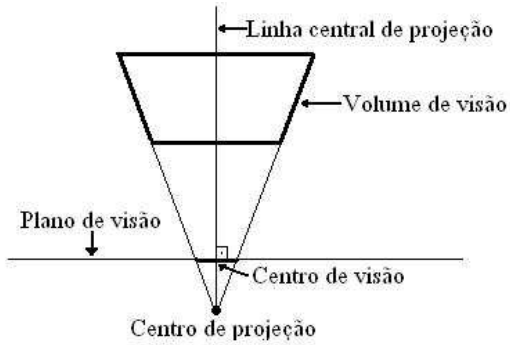

Figura 14 – Projeção perspectiva com linha central de projeção perpendicular ao plano de projeção

A matriz responsável por esta transformação é dada abaixo:

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

Os parâmetros de cisalhamento são dados pelas equações:

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

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

Devemos agora aplicar a transformação de escala, dado pelo item 2 citado anteriormente para produzir um paralelepípedo regular. A transformação para essa conversão é dada pelas equações:

$$x''=x'\left(\frac{z_{prp}-z_{vp}}{z_{prp}-z}\right)+x_{prp}\left(\frac{z_{vp}-z}{z_{prp}-z}\right)$$

$$y''=y'\left(\frac{z_{prp}-z_{vp}}{z_{prp}-z}\right)+y_{prp}\left(\frac{z_{vp}-z}{z_{prp}-z}\right)$$

Sendo a representação na forma de matriz homogênea dada abaixo:

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

A matriz geral de transformação perspectiva é obtida através da composição das matrizes de escala e de cisalhamento: $M_{perspectiva}=M_{escala}\cdot M_{cisalhamento}$. E a matriz de transformação direta de coordenadas do mundo para coordenadas de projeção perspectiva é dada pela concatenação da matriz $M_{perspectiva}$, com a matriz $M_{wc,wv}$ encontrada no capítulo 2.

5. Recorte ou *Clipping*

Para recortar um segmento de reta considerando o volume de visão, precisamos testar a posição da linha usando os as equações dos planos delimitadores do volume. substituindo os pontos extremos do segmento da linha nestas equações. Podemos desta forma, determinar se o ponto está fora ou dentro do volume de visão. 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. Sendo A, B, C e D os parâmetros que constituem o plano.

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. A intersecção de uma linha com um plano é encontrada quando a equação $Ax + By + Cz + D = 0$ é satisfeita. Para recortar a superfície de um polígono, podemos recortar individualmente cada uma de suas arestas, calculando os pontos de intersecção quando necessário.

Caso o volume de visão seja um paralelepípedo regular, a verificação necessária para o recorte torna-se mais simples. Quando os planos delimitadores são paralelos aos eixos do sistema de coordenadas, basta verificar se cada coordenada está dentro dos limites impostos pelos planos. Algumas APIs como o PHIGS, adicionam uma etapa adicional de transformações, para obter um volume de visão normalizado, onde os planos que limitam a região visível são x=0, x=1, y=0, y=1, z=0 e z=1. As vantagens trazidas por esta normalização do volume de visão incluem a separação do tamanho dos objetos de qualquer característica especifica dos dispositivos de saída, e a simplificação e padronização dos procedimentos de recorte.

Recorte da Janela de Visualização

Podemos utilizar 6 bits para representar a posição de um ponto com relação ao volume de visão. Cada um dos bits é associado a um plano delimitador, e possui valor 1 quando o ponto encontra-se fora da região limitada pelo plano em particular. Sendo 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). Cada um dos bits possui o valor 0, ou 1 nos casos abaixo:

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 $z > zv_{max}$

Desta forma, qualquer sequência de bits diferente de 000000 indica um ponto que está fora da região de visão. Como exemplo, podemos citar a sequência 100100, que indica um ponto que está atrás e abaixo do volume de visão. Um segmento de reta está dentro da região visível, caso ambos os pontos extremos estejam associados a uma sequência 000000. Quando pelo menos um dos pontos não possui uma sequência 000000, aplica-se o operador lógico “and” sobre as sequências dos dois pontos, se o resultado for diferente de 0, temos que ambos os pontos estão fora de um mesmo limitador (por exemplo: ambos acima da região visível). Neste caso, o segmento de reta é descartado. 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. Admitindo um segmento de reta com pontos extremos em $P_1 = (x_1,y_1,z_1)$ e $P2=(x_2,y_2,z_2)$, podemos escrever as equações paramétricas das linhas da seguinte forma:

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

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

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

Sendo $0 \leqslant u \leqslant 1$.

Para encontrar a intersecção de uma linha com um plano limite do volume de visão, substitui-se o valor da coordenada daquela plano na equação apropriada. Como exemplo, se estivermos testando uma linha com o plano zv_{min}, encontramos o valor de u utilizando a última equação dada acima.

$$u = \frac{zv_{min} - z_1}{z_2 - z_1}$$

Utilizamos então o valor de u encontrado, para encontrar as coordenas x e y, utilizando as outras duas equações. Se x ou y não estiverem dentro dos limites permitidos, então esta linha intercepta o plano zv_{min} fora dos limites do volume de visão.

Coordenadas homogêneas

Devemos lembrar que caso estejamos trabalhando com coordenadas homogêneas, deve-se verificar o parâmetro homogêneo antes de realizar as comparações explicadas neste capítulo. Algumas transformações realizadas podem ter modificado o parâmetro homogêneo h para um valor diferente de um, como é o caso da transformação de projeção perspectiva. Caso isto ocorra, devemos verificar se o ponto obtido (x_h, y_h, z_h, h) encontra-se dentro da região visível, efetuando as comparações abaixo:

$$h\,xv_{min} \leqslant x_h \leqslant h\,xv_{max} \,,\, h\,yv_{min} \leqslant y_h \leqslant h\,yv_{max} \,,\, h\,zv_{min} \leqslant z_h \leqslant h\,zv_{max} \,,\, para\; h > 0$$

$$h\,xv_{max} \leqslant x_h \leqslant h\,xv_{min} \,,\, h\,yv_{max} \leqslant y_h \leqslant h\,yv_{min} \,,\, h\,zv_{max} \leqslant z_h \leqslant h\,zv_{min} \,,\, para\; h < 0$$

Levando isso em consideração, as operações de recorte podem ser realizadas da mesma forma como explicadas anteriormente.

Bibliografia

[1] Hearn, D. D. & Baker, M. P. *Computer Graphics with OpenGL Prentice Hall* Professional Technical Reference, 2003

[2] Schroeder, W.; Martin, K. M. & Lorensen, W. E. *The visualization toolkit (2nd ed.): an object-oriented approach to 3D graphics Prentice-Hall*, Inc., 1998

[3] Boulos, P. & Camargo, I. Geometria Analítica - Um Tratamento Vetorial Pearson Prentice Hall, 1998

[4] Nate Robins Tutor. *Nate Robins Main OPENGL Chronicles allies Tutors*. *http://www.xmission.com/~nate/tutors.html*. Último acesso: 26/05/2010,