Viewing 3D

Profa. Ana Luísa D. Martins Lemos

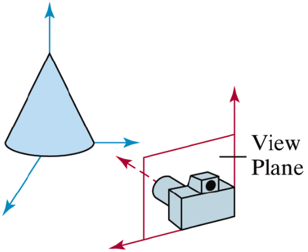
May 9, 2018

Visão Geral

- Viewing Pipeline 3D
  - As funções de viewing processam a descrição de objetos por meio de um conjunto de procedimentos a fim de projetar uma visão específica desses na superfície do dispositivo de saída
  - Alguns desses procedimentos são similares aos do *Viewing Pipeline 2D*
    - Rotinas de recorte
  - Mas outros são específicos do 3D
    - Rotinas de projeção
    - Identificação de partes visíveis da cena
    - Efeitos de luz

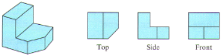
Visualizando uma Cena 3D

- Para se obter uma visão de uma cena 3D descrita por *coordenadas do mundo*, primeiro é necessário definir um sistema de referência para os parâmetros de visão (ou câmera)
  - Define a posição e orientação do **plano de visão** (ou **plano de projeção**) - plano do filme da câmera

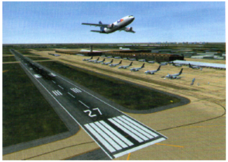


Projeções

- É possível escolher diferentes métodos para projetar uma cena no plano de visão
  - **Projeção Paralela**: projeta os pontos de um objeto ao longo de linhas paralelas - usado para desenhos arquitetônicos e de engenharia
  - **Projeção Perspectiva**: projeta os pontos de um objeto ao longo de caminhos convergentes - cenas mais realísticas (objetos longe da posição de visão são mostrados menores)



(a) Projeção Paralela



(b) Projeção Perspectiva

Depth Cueing (Profundidade)

- Com raras exceções, informação de profundidade é importante para a composição de uma cena 3D
  - Identificar a parte da frente e de trás dos objetos

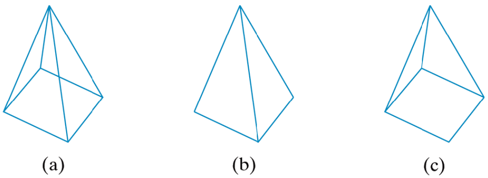
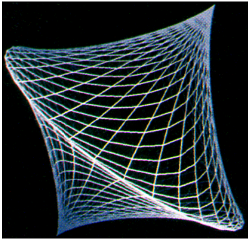


Figure : Problemas de ambiguidade pela falta de informação de profundidade em (a), que pode ser interpretada como (b) ou (c)

Depth Cueing (Profundidade)

- Depth Cueing
  - Forma simples de indicar profundidade de objetos aramados (*wire-frame*) variando o brilho das linhas
  - Linhas mais próximas da posição de visão são mais brilhantes



Identificando Linhas e Superfícies Visíveis

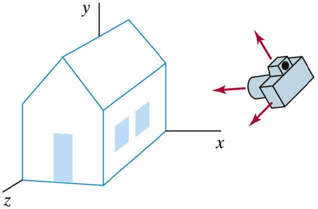
- Cenas Wire-Frame
  - Existem outras formas de tornar visualizações de *wire-frame* mais realísticas
    - Colorir as linhas visíveis de forma diferente das não-visíveis
    - Mostrar as linhas não visíveis como linhas pontilhadas
    - Remover as linhas não visíveis - também remove informação sobre a forma do objeto
  - Cenas Realísticas
    - Para cenas realísticas, as partes não visíveis de um objeto são completamente eliminadas, e somente as visíveis são mostradas
    - Os pixels da tela terão informação apenas das cores das superfícies da frente

Rendering de Superfície

- Efeitos realísticos são alcançados usando efeitos de iluminação
  - Define-se a luz ambiente
  - Especifica-se a localização e cor das diferentes fontes de luz
- As características dos materiais são também especificadas
  - Transparente, opaco, rugoso, etc.

Viewing Pipeline 3D

- O processo para se criar uma imagem de computação gráfica de uma cena 3D é parecido com o processo de se tirar uma foto
  - É necessário escolher a posição de visão, onde será posta a câmera
  - É preciso definir a orientação da câmera
    - Como apontar a câmera a partir da posição de visão
    - Como a câmera vai estar rotacionada, definindo a posição *up*

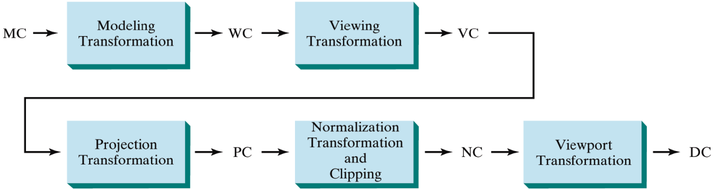


Viewing Pipeline 3D

- Algumas das operações do *Viewing Pipeline 3D* são semelhantes ao 2D
  - Uma **viewport 2D** é usada para posicionar a visão projetada no dispositivo de saída
  - Uma **janela de recorte 2D** é usada para selecionar uma visão que será mapeada na *viewport*
  - Uma janela de saída é definida em coordenadas da tela
- Porém, outras são diferentes
  - A janela de recorte é posicionada sobre um plano de visão, e a cena é recortada considerando um volume no espaço (**volume de recorte**) usando planos de recorte

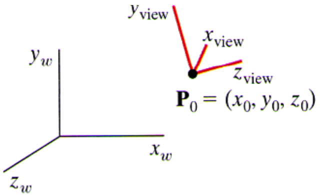
Viewing Pipeline 3D

- A **posição de visão**, o **plano de visão**, a **janela de recorte** e os **planos de recorte** são especificados dentro do sistema de **coordenadas do mundo**



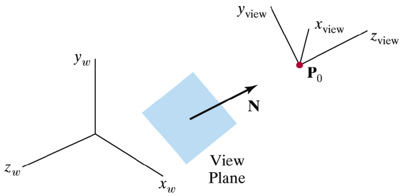
Parâmetros de Coordenadas de Visão 3D

- Para estabelecer um sistema de coordenadas de visão 3D é preciso definir
  - A origem do sistema $P_0 = (x_0, y_0, z_0)$, chamada de **ponto de visão** (também referido como posição do olho ou da câmera)
  - O **vetor view-up** V, que define a direção y_{view}
  - Uma segunda direção para um dos eixos coordenados restantes, normalmente o z_{view}, com a direção de visão ao longo desse eixo



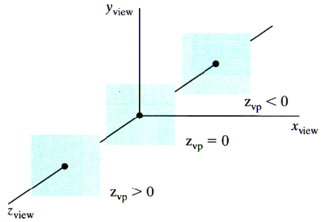
Vetor Normal ao Plano de Visão

- Como a direção de visão em geral é definida sobre o eixo z_{view}, o plano de projeção é normalmente assumido ser perpendicular a esse eixo
    - Assim, a orientação do plano de projeção e a direção positiva do eixo z_{view} podem ser definidas com um **vetor normal N ao plano de projeção**



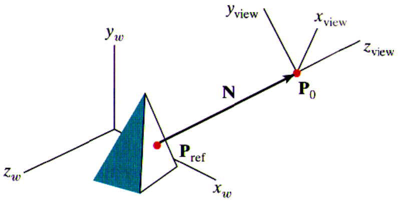
Vetor Normal ao Plano de Visão

- Um valor escalar é usado para definir a posição do plano de projeção em alguma coordenada z_{vp} ao longo do eixo z_{view}
  - Especifica a distância da origem da visão ao longo da direção de visão, normalmente na direção negativa de z_{view}
  - Portanto, o plano de projeção é sempre paralelo ao plano $x_{view}y_{view}$



Vetor Normal ao Plano de Visão

- O vetor normal N pode ser especificado de várias formas
  - A direção de N pode ser definida ao longo da linha partindo de um ponto de referência P_{ref} até a origem de visão P_0 (o contrário também é válido)
    - Nesse caso, esse ponto de referência é denominado **look-at point**, com a direção de visão oposta à de N



O Vetor View-Up

- Uma vez estabelecida a direção normal ao plano de projeção N, podemos estabelecer o **vetor view-up** V que dá a direção do eixo y_{view}
- Usualmente V é definido selecionando uma posição relativa à origem do sistema de coordenadas do mundo

O Vetor View-Up

- V pode ser definido em qualquer direção, desde que não paralela a N
  - Uma forma conveniente seria definir uma direção paralela ao eixo y_w, $V = (0, 1, 0)$
  - Se esse não for exatamente perpendicular a N, as rotinas de visão podem ajustá-lo projetando-o em um plano perpendicular a N

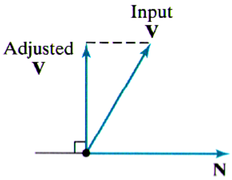


Figure : Ajuste do vetor *view-up* V para torná-lo perpendicular a N

Sistema de Coordenadas de Visão **uvn**

- Como a normal N define a orientação z_{view}, e o vetor *view-up* V é usado para definir y_{view}, só é necessário definir a direção positiva de x_{view}
  - Essa direção é representada pelo vetor U, calculada como o produto vetorial de N e V
  - O produto vetorial de N e U também pode ser usado para ajustar o valor de V ao longo do eixo y_{view}

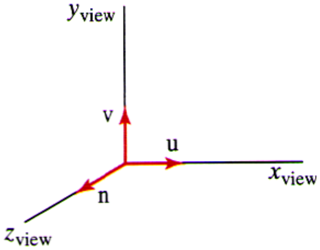
Sistema de Coordenadas de Visão **uvn**

- Para se obter o sistema de coordenadas **uvn** fazemos

$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

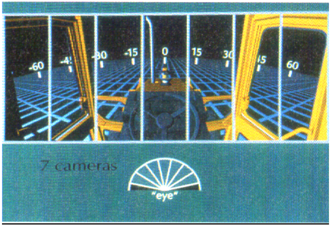
$$u = \frac{V \times n}{|V \times n|} = (u_x, u_y, u_z)$$

$$v = n \times u = (v_x, v_y, v_z)$$



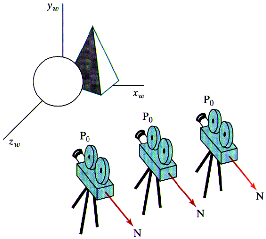
Gerando Efeitos de Visão 3D

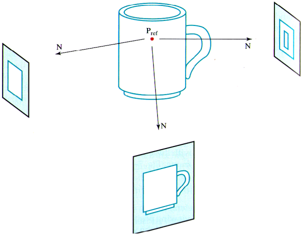
- Variando-se os parâmetros de visão é possível obter diferentes efeitos
  - De uma posição de visão fixa é possível mudar N para mostrar objetos em posições ao redor da origem
  - Variar N para criar uma cena composta de múltiplas visões de uma posição fixa da câmera
    - Quando alterar N não esqueça de mudar os outros eixos para se manter o sistema orientado pela mão direita



Gerando Efeitos de Visão 3D

- Efeitos de mover a câmera (pan) podem ser obtidos fixando N e modificando o ponto de visão
- Para mostrar diferentes visões de um objeto (visão lateral, frontal, etc.) podemos mover o ponto de visão ao redor do objeto





Transformação

Sistema de Coordenadas do Mundo para o de Visão

- No *Viewing Pipeline 3D*, o primeiro passo a ser executado após a cena ser construída é transferir a descrição dos objetos ao sistema de coordenadas de visão
  - Essa conversão é equivalente a sobrepor o sistema de referência de visão sobre o sistema de referência do mundo
- Esse mapeamento pode ser feito
  1. Transladando a origem do sistema de coordenadas de visão para a origem do sistema de coordenadas do mundo
  2. Rotacionando para alinhar os eixos x_{view}, y_{view} e z_{view} com os eixos do mundo x_w, y_w e z_w

Transformação

Sistema de Coordenadas do Mundo para o de Visão

- Se a origem do sistema de visão for $P_0 = (x_0, y_0, z_0)$ a matriz de translação será

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

Transformação

Sistema de Coordenadas do Mundo para o de Visão

- A matriz de rotação pode ser obtida direto dos vetores $u = (u_x, u_y, u_z)$, $v = (v_x, v_y, v_z)$ e $n = (n_x, n_y, n_z)$

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

Transformação

Sistema de Coordenadas do Mundo para o de Visão

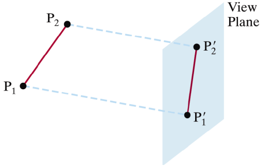
- A matriz de transformação é portanto

$$M_{\mathrm{WC,VC}} = R.T$$

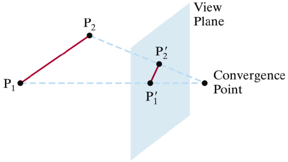
$$M_{\mathrm{WC,VC}} = \begin{bmatrix} u_x & u_y & u_z & -u.P_0 \\ v_x & v_y & v_z & -v.P_0 \\ n_x & n_y & n_z & -n.P_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformações de Projeção

- Após a transformação para as coordenadas de visão, o próximo passo do *Viewing Pipeline 3D* é a projeção no plano de projeção
- Em geral os pacotes gráficos suportam
  - **Projeção Paralela**: as coordenadas são transferidas para o plano de projeção ao longo de linhas paralelas
  - **Projeção Perspectiva**: as coordenadas são transferidas para o plano de projeção ao longo de linhas que convergem a um ponto



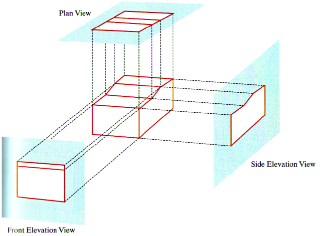
Projeção Paralela



Projeção Perspectiva

Projeções Ortogonais

- Projeção Ortogonal (ou Ortográfica)
  - Transformação da descrição dos objetos a um plano de projeção ao longo de linhas paralelas ao vetor normal N
- Frequentemente usada para produzir a visão frontal, lateral e superior de um objeto
- Preserva tamanhos e ângulos: útil para desenhos arquitetônicos e de engenharia



Projeções Ortogonais

Axonométricas e Isométricas

- Projeção Ortogonal Axonométrica
  - Projeção ortogonal que mostra mais de uma face de um objeto
  - A mais comum é a **isométrica**
    - O plano de projeção é alinhado de forma a intersectar cada eixo coordenado no qual o objeto é definido à mesma distância da origem

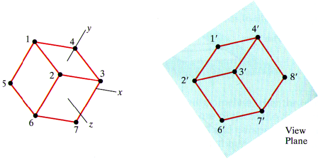


Figure : Essa projeção é obtida alinhando o vetor N com a diagonal do cubo

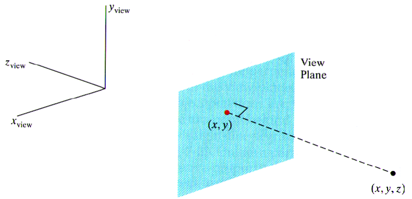
Coordenadas de Projeções Ortogonais

- Com a direção de projeção paralela a z_{view}, as equações para a transformação de projeção ortogonal de uma posição (x, y, z) no sistema de coordenadas de visão são

$$x_p = x$$

$$y_p = y$$

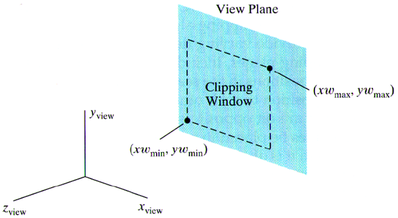
- O valor de z é armazenado para uso futuro nos procedimentos para determinar visibilidade



Projeções Ortogonais

Janela de Recorte e Volume de Projeção Ortogonal

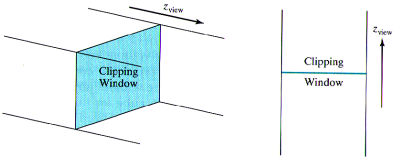
- Para de determinar o quanto de uma cena 3D será transferido para o plano de projeção, uma **Janela de Recorte** é utilizada
  - É necessário determinar os limites dessa janela sobre o plano de projeção com as arestas paralelas aos eixos x_{view} e y_{view}



Projeções Ortogonais

Janela de Recorte e Volume de Projeção Ortogonal

- As arestas da *Janela de Recorte* especificam os limites x e y da parte da cena que será mostrada, formando o **volume de visão de projeção ortogonal**



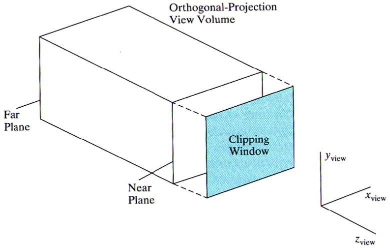
Side View
(a)

Top View
(b)

Projeções Ortogonais

Janela de Recorte e Volume de Projeção Ortogonal

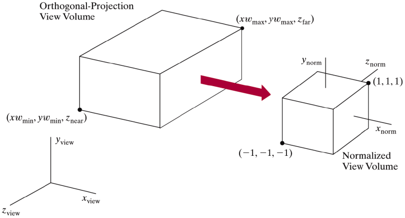
- Para se limitar a extensão desse volume na direção z_{view}, dois planos de fronteira, chamados **planos de recorte near/far** são considerados, paralelos ao plano de visão
  - Permite eliminar objetos que estão na frente ou atrás de uma parte da cena
  - Com a direção de visão ao longo do eixo negativo z_{view}, temos $z_{\text{far}} < z_{\text{near}}$



Projeções Ortogonais

Transformação de Normalização

- Como qualquer posição (x, y, z) em uma projeção ortogonal é mapeada para (x, y), as coordenadas dentro do volume de visão são as coordenadas de projeção, assim essas podem ser mapeadas para um **volume de visão normalizado** sem precisarem ser reprojetadas



Projeções Ortogonais

Transformação de Normalização

- Essa transformação de normalização é semelhante à obtida em 2D, com a adição dos valores da coordenada z sendo normalizados do intervalo z_{near} a z_{far} para -1 a 1

$$M_{\text{orto,norm}} = \begin{bmatrix} \frac{2}{xw_{\max} - xw_{\min}} & 0 & 0 & -\frac{xw_{\max} + xw_{\min}}{xw_{\max} - xw_{\min}} \\ 0 & \frac{2}{yw_{\max} - yw_{\min}} & 0 & -\frac{yw_{\max} + yw_{\min}}{yw_{\max} - yw_{\min}} \\ 0 & 0 & -\frac{2}{z_{\text{near}} - z_{\text{far}}} & \frac{z_{\text{near}} + z_{\text{far}}}{z_{\text{near}} - z_{\text{far}}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projeções Ortogonais

Transformação de Normalização

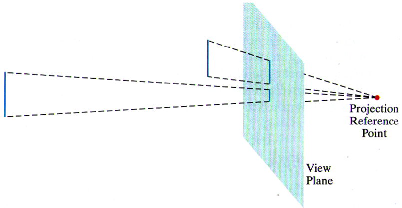
- A multiplicação dessa matriz pela matriz que transforma as coordenadas do mundo em coordenadas de visão produz a transformação completa para se obter as coordenadas normalizadas da projeção ortogonal

$$M_{\mathrm{orto,norm}} \cdot M_{WC,VC}$$

- Outras operações, como recorte, identificação de superfícies visíveis, etc. podem ser executadas de forma mais eficiente

Projeções Perspectivas

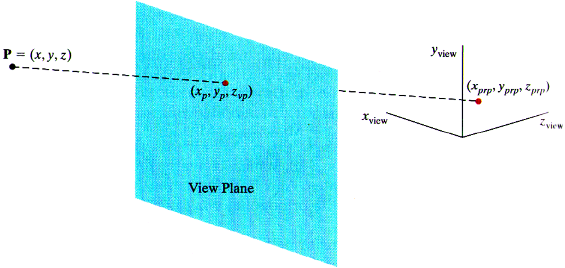
- Para conseguir maior realismo que o obtido nas projeções paralelas, temos que considerar que os raios de luz refletidos na cena seguem caminhos convergentes
- Isso pode ser aproximado projetando objetos ao plano de visão ao longo de caminhos convergentes a uma posição chamada **ponto de referência de projeção** (ou **centro de projeção**)



Projeções Perspectivas

Transformação de coordenadas

- Algumas bibliotecas gráficas permitem que se escolha o ponto de referência de projeção $(x_{\text{prp}}, y_{\text{prp}}, z_{\text{prp}})$



Projeções Perspectivas

Transformação de coordenadas

- Considerando que a projeção do ponto (x, y, z) intersecte o plano de projeção na posição (x_p, y_p, z_p) podemos descrever qualquer ponto ao longo dessa linha de projeção como

$$x' = x + u(x_{prp} - x)$$

$$y' = y + u(y_{prp} - y)$$

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

$$0 \leq u \leq 1$$

- No plano de visão $z' = z_{vp}$, então podemos encontrar u

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

Projeções Perspectivas

Transformação de coordenadas

- Substituindo esse valor de u para as equações de x' e y' obtemos as equações de projeção perspectiva

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

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

Projeções Perspectivas

Casos Especiais das Equações de Projeção Perspectiva

- É possível restringir os parâmetros de projeção para facilitar os cálculos
- Se o centro de projeção estiver sobre o eixo z_{view}, então $x_{\text{prp}} = y_{\text{prp}} = 0$

$$x_p = x\left(\frac{z_{\text{prp}} - z_{\text{vp}}}{z_{\text{prp}} - z}\right),\ y_p = y\left(\frac{z_{\text{prp}} - z_{\text{vp}}}{z_{\text{prp}} - z}\right)$$

- Se o centro de projeção for fixado na origem, então $(x_{\text{prp}}, y_{\text{prp}}, z_{\text{prp}}) = (0, 0, 0)$

$$x_p = x\left(\frac{z_{\text{vp}}}{z}\right),\ y_p = y\left(\frac{z_{\text{vp}}}{z}\right)$$

Projeções Perspectivas

Casos Especiais das Equações de Projeção Perspectiva

- É possível restringir os parâmetros de projeção para facilitar os cálculos
- Se o plano de projeção for o plano *uv* e o centro de projeção estiver sobre o eixo z_{view}, então $x_{\text{prp}} = y_{\text{prp}} = z_{\text{vp}} = 0$

$$x_p = x\left(\frac{z_{\text{prp}}}{z_{\text{prp}} - z}\right), y_p = y\left(\frac{z_{\text{prp}}}{z_{\text{prp}} - z}\right)$$

Projeções Perspectivas

Casos Especiais das Equações de Projeção Perspectiva

- Em geral o plano de projeção está entre o centro de projeção e a cena, mas outras posições são possíveis (menos sobre o plano de projeção)

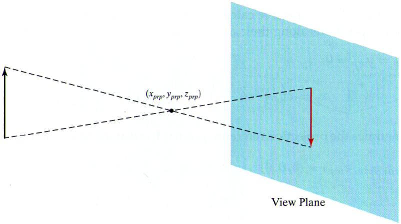


Figure : Os objetos são invertidos se o ponto de referência está entre o plano de visão e a cena

Projeções Perspectivas

Casos Especiais das Equações de Projeção Perspectiva

- Os efeitos de perspectiva também dependem da distância entre o centro de projeção e o plano de visão

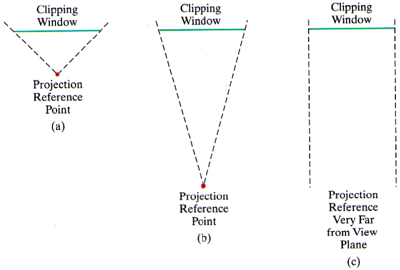
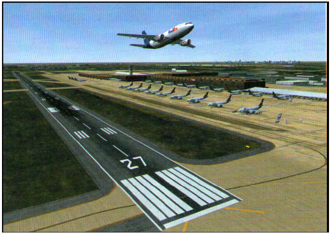


Figure : Se o centro de projeção está próximo ao plano de projeção, objetos mais próximos ao plano aparecerão maiores do que os distantes

Projeções Perspectivas

Pontos de Fuga para Projeções Perspectivas

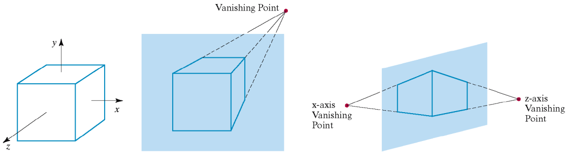
- Na projeção perspectiva
  - Linhas paralelas entre si e ao plano de projeção são projetadas como linhas paralelas
  - Linhas paralelas que não são paralelas ao plano de projeção são projetadas em linhas convergentes
- O ponto para o qual aparentemente as linhas convergem é chamado **ponto de fuga**



Projeções Perspectivas

Pontos de Fuga para Projeções Perspectivas

- Conjuntos de linhas que são paralelas a algum dos eixos principais de um objeto levam à composição dos **pontos de fuga principais**
  - É possível controlar o número desses pontos de fuga (um, dois ou três) por meio da orientação dos plano de projeção
  - O número de pontos de fuga principais é igual à quantidade de eixos principais que intersectam o plano de projeção



(a) Principle Axes for Cube

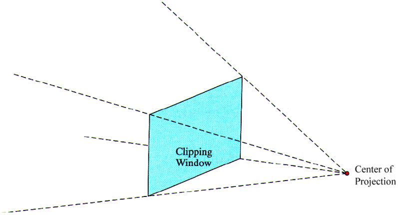
(b) One-Point Perspective Projection

(c) Two-Point Perspective Projection

Projeções Perspectivas

Volume de Projeção Perspectiva

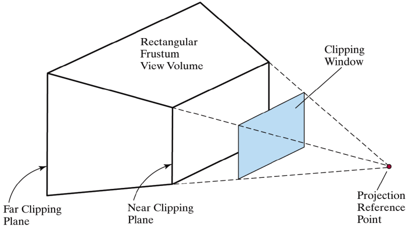
- Em uma projeção perspectiva, o volume de visão definido é uma pirâmide infinita com seu ápice no centro de projeção, normalmente chamada de **pirâmide de visão**
  - Objetos fora dessa pirâmide são eliminados pelas rotinas de recorte



Projeções Perspectivas

Volume de Projeção Perspectiva

- Adicionando os planos de recorte *near*/*far* perpendiculares ao eixo z_{view} essa pirâmide é truncada resultando em um tronco de pirâmide (**frustum**)



Projeções Perspectivas

Matriz de Transformação de Projeção Perspectiva

- Não é possível a partir das equações derivadas anteriormente definir uma matriz de transformação perspectiva - os denominadores dos coeficientes de x e y são função de z

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

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

- Essa limitação pode ser superada usando coordenadas homogêneas

$$x_{\text{p}} = \frac{x_h}{h}, y_{\text{p}} = \frac{y_h}{h}$$

- Onde o parâmetro homogêneo é

$$h = z_{\text{prp}} - z$$

Projeções Perspectivas

Matriz de Transformação de Projeção Perspectiva

- Os numeradores permanecem os mesmos

$$x_h = x(z_{prp} - z_{vp}) + x_{prp}(z_{vp} - z)$$

$$y_h = y(z_{prp} - z_{vp}) + y_{prp}(z_{vp} - z)$$

- E o fator homogêneo é

$$h = z_{prp} - z$$

Projeções Perspectivas

Matriz de Transformação de Projeção Perspectiva

- Então pode-se definir uma matriz de transformação que converte uma posição espacial em coordenadas homogêneas
  - Inicialmente calcula-se as coordenadas homogêneas $P_h = (x_h, y_h, z_h, h)$ de um ponto $P = (x, y, z, 1)$ usando a matriz de projeção perspectiva

$$P_h = M_{\text{pers}}.P$$

- Então essas são divididas por h para se obter as coordenadas das posições transformadas

Projeções Perspectivas

Matriz de Transformação de Projeção Perspectiva

- Definir a matriz para encontrar x_h e y_h é de forma direta, mas informação de profundidade deve ser introduzida para o parâmetro homogêneo h não distorcer z
- Isso pode ser feito definindo os valores para a transformação de z de forma a normalizar as coordenadas z_p da projeção
    - Isso pode ser feito de várias formas, uma delas

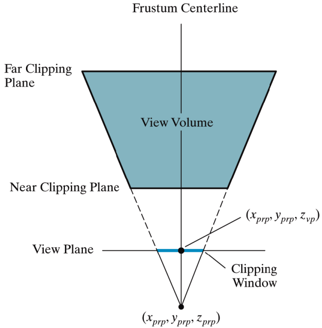
$$M_{\text{pers}} = \begin{bmatrix} z_{\text{prp}} - z_{\text{vp}} & 0 & -x_{\text{prp}} & x_{\text{prp}} z_{\text{prp}} \\ 0 & z_{\text{prp}} - z_{\text{vp}} & -y_{\text{prp}} & y_{\text{prp}} z_{\text{prp}} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{\text{prp}} \end{bmatrix}$$

- Onde s_z e t_z são fatores de escala e translação para a normalização das coordenadas z

Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

- A linha do centro de projeção através do centro da janela de recorte e do volume de visão é a linha central do *frustum* de projeção perspectiva
  - Se essa for perpendicular ao plano de visão temos um **frustum simétrico**



Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

- Como a linha central do *frustum* intersecta o plano em $(x_{prp}, y_{prp}, z_{prp})$ é possível expressar os cantos da janela de recorte em termos das dimensões da janela

$$xw_{min} = x_{prp} - \frac{\text{width}}{2}, xw_{max} = x_{prp} + \frac{\text{width}}{2}$$

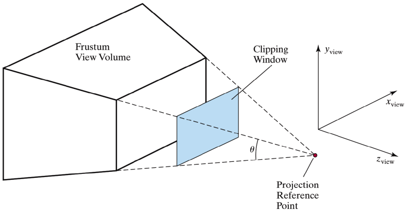
$$yw_{min} = y_{prp} - \frac{\text{height}}{2}, yw_{max} = y_{prp} + \frac{\text{height}}{2}$$

- Assim é possível especificar uma projeção perspectiva simétrica usando a largura e a altura da janela de recorte ao invés das coordenadas da janela

Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

- Uma projeção perspectiva pode também ser aproximada considerando o **cone de visão**, definido pelo **ângulo do campo de visão**, de uma câmera
  - Grandes ângulos de campo de visão correspondem a lentes grandes-angulares



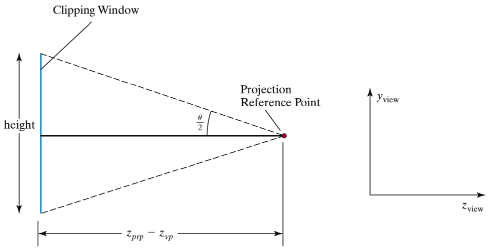
- De forma geral, o ângulo do campo de visão é definido entre o plano de recorte superior e o inferior do *frustum*

Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

- Dado o ponto de referência e a posição do plano de visão, o ângulo do campo de visão determina a altura da janela de recorte

$$\tan\left(\frac{\theta}{2}\right) = \frac{\text{height}/2}{z_{\text{prp}} - z_{\text{vp}}}$$



Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

- Para definir a largura é necessário considerar um parâmetro adicional que poderia ser a largura da janela ou a razão de aspecto

$$\text{aspect} = \left(\frac{\text{width}}{\text{height}}\right)$$

Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

- Assim podemos substituir os elementos $(z_{prp} - z_{vp})$ da diagonal da matriz M_{pers} por

$$\text{height} = 2(z_{prp} - z_{vp}) \tan\left(\frac{\theta}{2}\right)$$

$$z_{prp} - z_{vp} = \frac{\text{height}}{2} \cot\left(\frac{\theta}{2}\right)$$

- Ou por

$$z_{prp} - z_{vp} = \frac{\text{width.} \cot(\theta/2)}{2.\text{aspect}}$$

Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

- Assim temos

$$M_{\text{pers}} = \begin{bmatrix} \frac{\text{height}}{2}\cot\left(\frac{\theta}{2}\right) & 0 & -x_{\text{prp}} & x_{\text{prp}}z_{\text{prp}} \\ 0 & \frac{\text{height}}{2}\cot\left(\frac{\theta}{2}\right) & -y_{\text{prp}} & y_{\text{prp}}z_{\text{prp}} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{\text{prp}} \end{bmatrix}$$

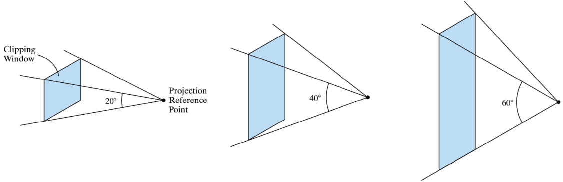
- Ou

$$M_{\text{pers}} = \begin{bmatrix} \frac{\text{width.}\cot(\theta/2)}{2.\text{aspect}} & 0 & -x_{prp} & x_{prp}z_{prp} \\ 0 & \frac{\text{width.}\cot(\theta/2)}{2.\text{aspect}} & -y_{prp} & y_{prp}z_{prp} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{\text{prp}} \end{bmatrix}$$

Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

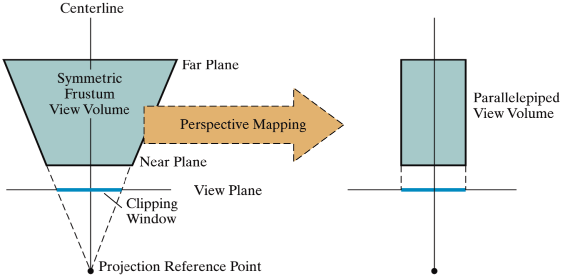
- Diminuir o ângulo do campo de visão diminui a janela de recorte
  - Mover o ponto de projeção para longe do plano de visão
  - Zoom-in de uma pequena região da cena
- Aumentar o ângulo do campo de visão aumenta a janela de recorte
  - Mover o ponto de projeção para próximo do plano de visão
  - Zoom-out da cena



Projeções Perspectivas

Frustum Simétrico de Projeção Perspectiva

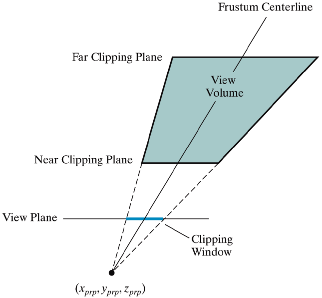
- Observação Importante: Para um *frustum* simétrico, a transformação perspectiva mapeia localizações dentro do *frustum* a coordenadas de projeção ortogonais dentro de um paralelepípedo retangular



Projeções Perspectivas

Frustum Oblíquo de Projeção Perspectiva

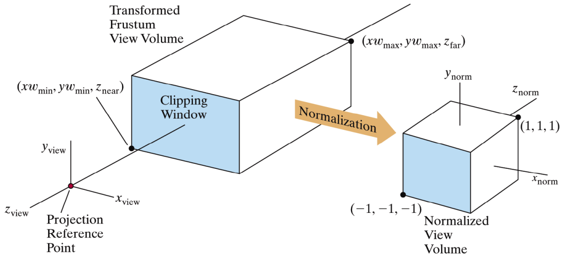
- Se a linha central do volume de visão não é perpendicular ao plano de visão, temos um **frustum oblíquo**
  - Primeiro transformar esse em um *frustum* simétrico



Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- O último passo da projeção perspectiva é mapear o paralelepípedo obtido para um **volume de visão normalizado**
  - É aplicado o mesmo procedimento da projeção paralela



Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- Os parâmetros para normalização da coordenada z já estão incluídos na matriz de projeção perspectiva, mas esses ainda precisam ser definidos
- Também são necessários os parâmetros para a normalização das coordenadas x e y
  - Não é preciso translação porque a linha central da paralelepípedo retangular é z_{view}, só é preciso uma transformação de escala com relação à origem

$$M_{xy\text{scale}} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- Se considerarmos que o ponto de referência de projeção está na origem do sistema de coordenadas de visão, $(x_{prp}, y_{prp}, z_{prp}) = (0, 0, 0)$, e o plano de visão está sobre o plano de recorte *near*, $z_{vp} = z_{near}$, a matriz de projeção perspectiva é simplificada para

$$M_{pers} = \begin{bmatrix} -z_{near} & 0 & 0 & 0 \\ 0 & -z_{near} & 0 & 0 \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- Concatenando essa matriz de escala com a matriz de projeção perspectiva temos

$$M_{\text{norm pers}} = M_{xy\text{scale}} \cdot M_{\text{pers}}$$

$$= \begin{bmatrix} -z_{\text{near}} \cdot s_x & 0 & s_x \frac{xw_{\min} + xw_{\max}}{2} & 0 \\ 0 & -z_{\text{near}} \cdot s_x & s_y \frac{yw_{\min} + yw_{\max}}{2} & 0 \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- Desta transformação obtemos as coordenadas homogêneas

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = M_{\text{norm pers}} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Que resulta nas coordenadas de projeção

$$x_p = \frac{x_h}{h} = \frac{-z_{\text{near}}.s_x.x + s_x(xw_{\text{min}} + xw_{\text{max}})/2}{-z}$$

$$y_p = \frac{y_h}{h} = \frac{-z_{\text{near}}.s_y.y + s_y(yw_{\text{min}} + yw_{\text{max}})/2}{-z}$$

$$z_p = \frac{z_h}{h} = \frac{s_z.z + t_z}{-z}$$

Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- Para normalizar essa projeção queremos que $(x_p, y_p, z_p) = (-1, -1, -1)$ quando as coordenadas de entrada forem $(xw_{\text{min}}, yw_{\text{min}}, z_{\text{near}})$ e $(x_p, y_p, z_p) = (1, 1, 1)$ quando essas forem $(xw_{\text{max}}, yw_{\text{max}}, z_{\text{far}})$
- Assim, resolvendo as equações anteriores com essas restrições obtemos

$$s_x = \frac{2}{xw_{\text{max}} - xw_{\text{min}}}, s_y = \frac{2}{yw_{\text{max}} - yw_{\text{min}}}$$

$$s_z = \frac{z_{\text{near}} + z_{\text{far}}}{z_{\text{near}} - z_{\text{far}}}, t_z = \frac{2.z_{\text{near}}.z_{\text{far}}}{z_{\text{near}} - z_{\text{far}}}$$

Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- Resultando na seguinte matriz de projeção perspectiva normalizada

$$M_{\text{norm pers}} =$$

$$\begin{bmatrix} \frac{-2.z_{\text{near}}}{xw_{\max}-xw_{\min}} & 0 & \frac{xw_{\max}+xw_{\min}}{xw_{\max}-xw_{\min}} & 0 \\ 0 & \frac{-2.z_{\text{near}}}{yw_{\max}-yw_{\min}} & \frac{yw_{\max}+yw_{\min}}{yw_{\max}-yw_{\min}} & 0 \\ 0 & 0 & \frac{z_{\text{near}}+z_{\text{far}}}{z_{\text{near}}-z_{\text{far}}} & -\frac{2.z_{\text{near}}.z_{\text{far}}}{z_{\text{near}}-z_{\text{far}}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

UNIFESP
UNIVERSIDADE FEDERAL DE SÃO PAULO
1933

- Se o volume de projeção perspectiva for definido como simétrico, é possível expressar os elementos da projeção perspectiva normalizada em termos do ângulo do campo de visão e das dimensões da janela de recorte
  - Com o ponto de referência da projeção na origem e o plano de visão sobre o plano de recorte *near*, temos

$$M_{\text{norm pers}} = \begin{bmatrix} \frac{\cot(\theta/2)}{\text{aspect}} & 0 & 0 & 0 \\ 0 & \cot(\theta/2) & 0 & 0 \\ 0 & 0 & \frac{z_{\text{near}}+z_{\text{far}}}{z_{\text{near}}-z_{\text{far}}} & -\frac{2.z_{\text{near}}.z_{\text{far}}}{z_{\text{near}}-z_{\text{far}}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Projeções Perspectivas

Transformação de Projeção Perspectiva Normalizada

- A matriz completa de transformação das coordenadas do mundo para projeção perspectiva normalizada é a composição

$$M_{\text{norm view vol}} = M_{\text{norm pers}}.R.T = M_{\text{norm pers}}.M_{\text{WC,VC}}$$

- Após essa matriz ser aplicada, as rotinas de recorte posem ser executadas

Transformação de Viewport

- Após o conteúdo do volume de visão ter sido definido, esse pode ser transferido para coordenadas da tela
- Esse é um processo semelhante ao 2D, porém informação de profundidade é preservada para teste de visibilidade e *rendering*
  - A variável z é normalizada entre 0 e 1, sendo que $z = 0$ temos a altura da tela

$$M_{\text{norm view vol, 3D screen}} =$$

$$\begin{bmatrix} \frac{xv_{\max}-xv_{\min}}{2} & 0 & 0 & \frac{xv_{\max}+xv_{\min}}{2} \\ 0 & \frac{yv_{\max}-yv_{\min}}{2} & 0 & \frac{yv_{\max}+yv_{\min}}{2} \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformação de Viewport

- Nas coordenadas normalizadas, $z_{\text{norm}} = -1$ é mapeada para a coordenada na tela $z_{\text{screen}} = 0$
  - *Viewport* é definida como $(xv_{\text{min}}, yv_{\text{min}}, 0)$ e $(xv_{\text{max}}, yv_{\text{max}}, 1)$
- As posições x e y são enviadas para o **frame buffer** (informação de cor para os pontos na tela)
- Os valores de z são enviados para o **depth buffer** para serem usados em rotinas para determinação de cor e visibilidade