

Pontifícia Universidade Católica do Rio de Janeiro — PUC-Rio

TRANSPARÊNCIAS:

*TRANSFORMAÇÕES GEOMÉTRICAS
PARA
VISUALIZAÇÃO EM 3D*

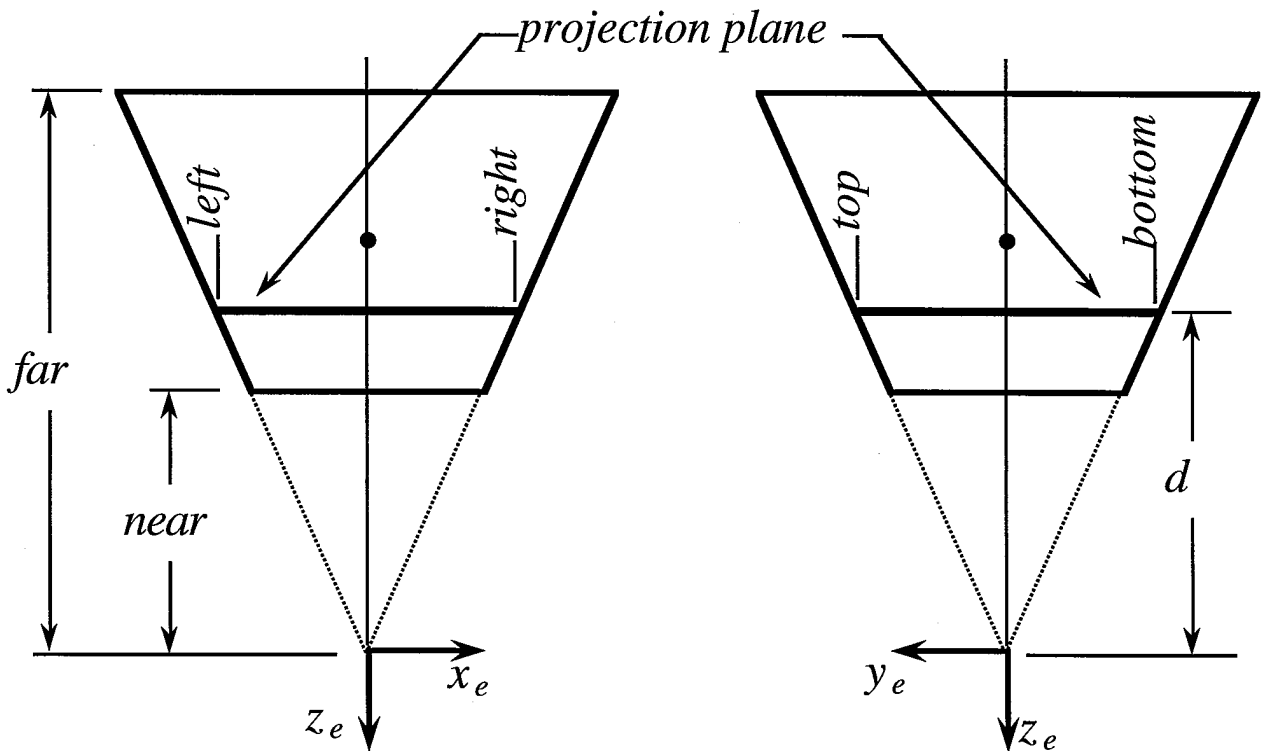
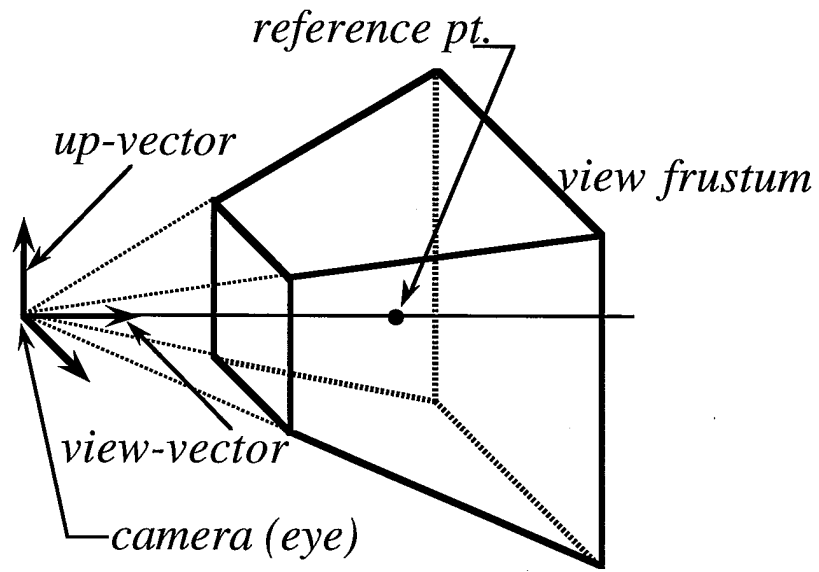
Luiz Fernando Martha

Referências

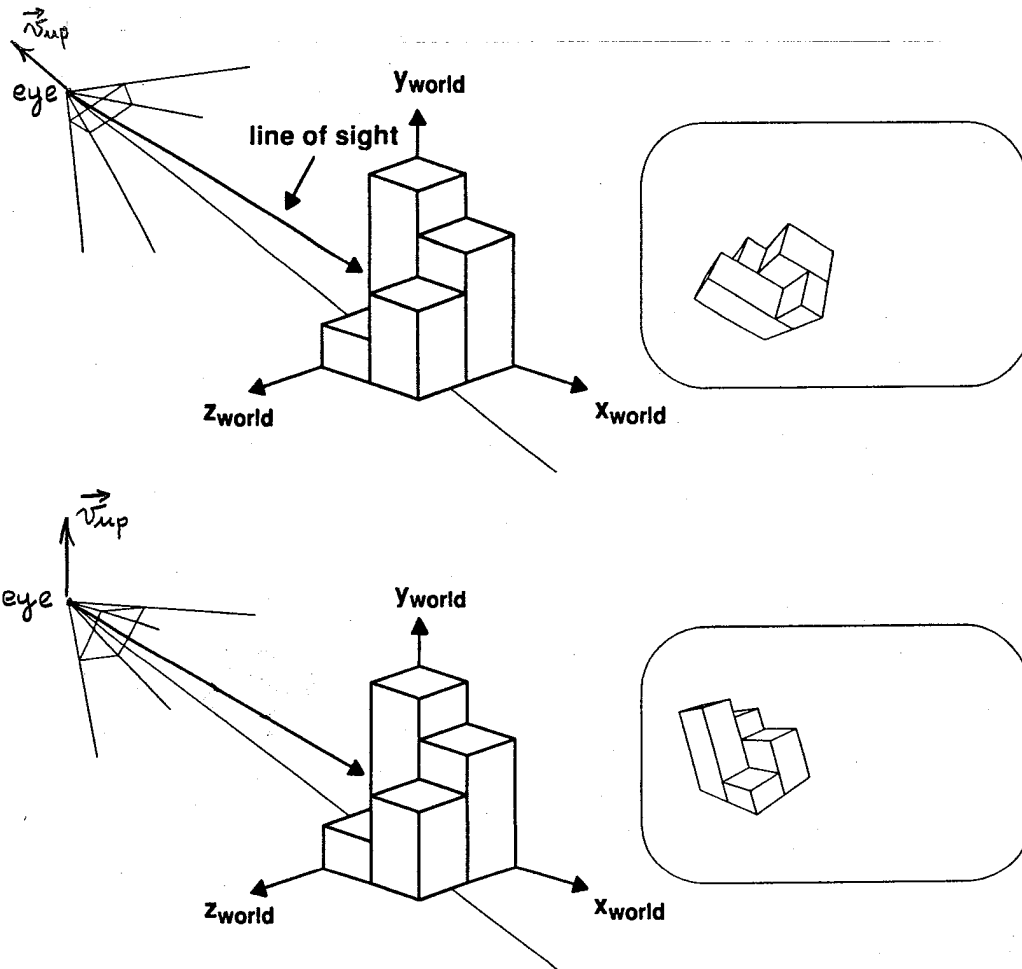
1. “Planar Geometric Projections and Viewing Transformations” – Carlbom, I.; Paciorek, J. – *ACM Computing Surveys*, Vol. 10, No. 4, Dec. 1978.
2. *OpenGL Programming Guide – The Official Guide to Learning OpenGL*, Release 1 – Neider, J.; Davis, T.; Woo, M. – Addison-Wesley Publishing Company, Reading, Massachusetts, 1993.
3. Manual IBM AIX Version 3 for RISC System/6000 – *Graphics Programming Concepts*, March 1990.

Rio de Janeiro, Novembro de 1994

Modelo de Câmera



Parâmetros de Visualização (Posicionamento da Câmera)



Parâmetros necessários:

- posição do olho \rightarrow (eye_x, eye_y, eye_z)
 - posição do ponto de referência \rightarrow (ref_x, ref_y, ref_z)
 - orientação vertical da câmera \rightarrow (\vec{v}_{up})
-) Direção de visão

Parâmetros de visualização em 3D

Posição da câmera (olho)

$$(eye_x, eye_y, eye_z)$$

Posição do ponto de referência

(um ponto no espaço de modelagem para onde a câmera mira)

$$(ref_x, ref_y, ref_z)$$

Vetor de orientação vertical da câmera

(view up-vector – *vup*)

$$(vup_x, vup_y, vup_z)$$

Sistema de coordenadas do olho

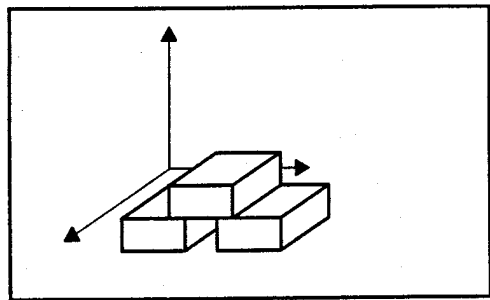
$$\mathbf{view} = (ref_x, ref_y, ref_z) - (eye_x, eye_y, eye_z)$$

$$\mathbf{z}_e = -\mathbf{view} / \|\mathbf{view}\|$$

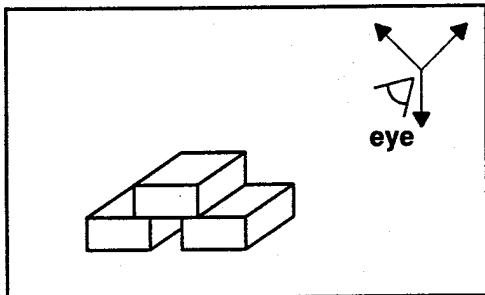
$$\mathbf{x}_e = (\mathbf{vup} \times \mathbf{z}_e) / \|\mathbf{vup} \times \mathbf{z}_e\|$$

$$\mathbf{y}_e = \mathbf{z}_e \times \mathbf{x}_e$$

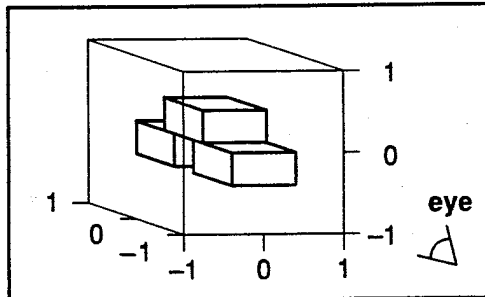
Transformações Geométricas para Visualização em 3D



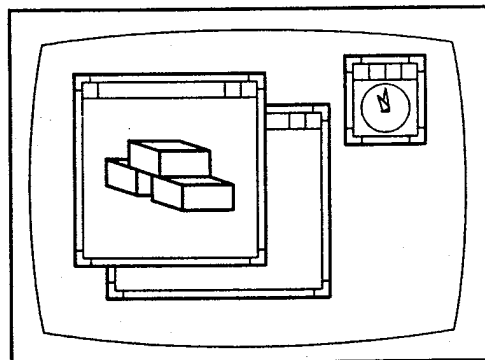
World coordinates



Viewing coordinates



Normalized Device coordinates



Screen coordinates

Operations performed on the matrix stack to transform from World to Viewing coordinates

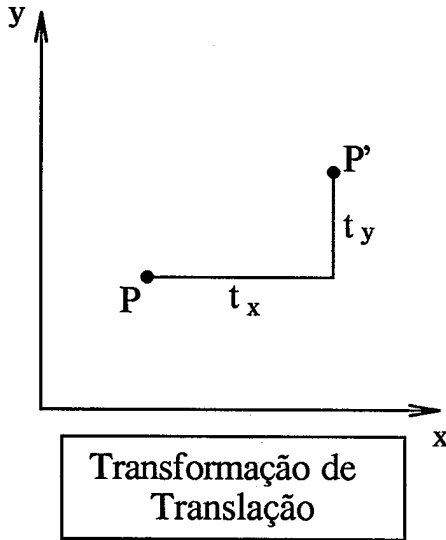
Operations performed on the projection matrix to transform from Viewing to Normalized Device coordinates

Operations performed on the viewport stack to transform from Normalized Device to Screen coordinates

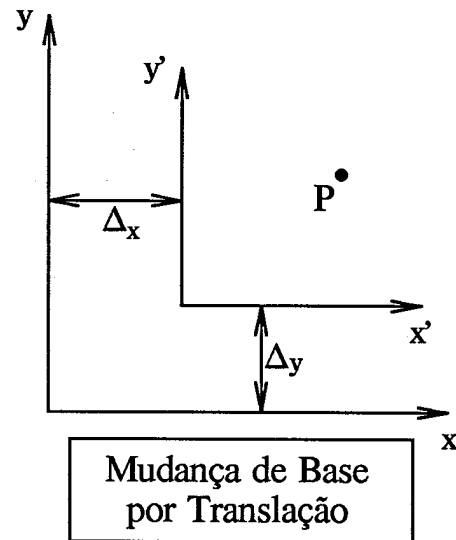
Coordinate Transformations

Transformações vs. Mudança de Base

Translação



$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} + \begin{Bmatrix} t_x \\ t_y \end{Bmatrix}$$

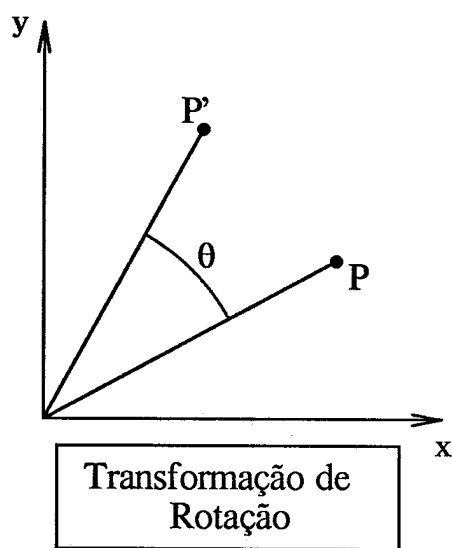


$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} - \begin{Bmatrix} \Delta_x \\ \Delta_y \end{Bmatrix}$$

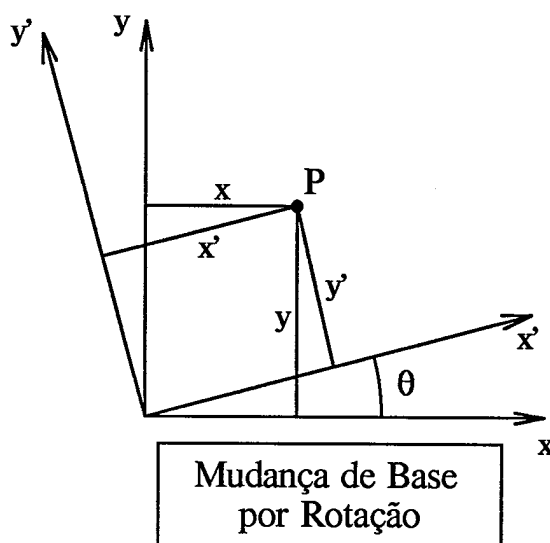
Conclusão

A mudança de base por translação pode ser vista como uma transformação de translação do ponto P de $(-\Delta_x, -\Delta_y)$.

Rotação



$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\text{sen}\theta \\ \text{sen}\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix}$$

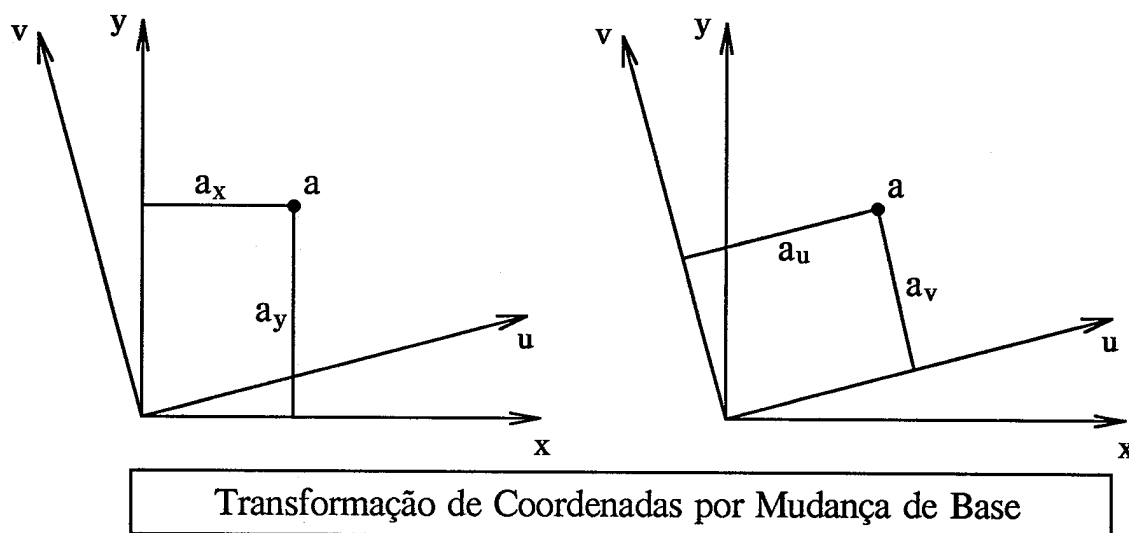


$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos\theta & \text{sen}\theta \\ -\text{sen}\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix}$$

Conclusão

A mudança de base por rotação pode ser vista como uma transformação de rotação do ponto P de $(-\theta)$.

Mais sobre mudança de base por rotação



$$M: \mathbf{R}^2 \rightarrow \mathbf{R}^2$$

$$\begin{Bmatrix} a_x \\ a_y \end{Bmatrix} \rightarrow \begin{Bmatrix} a_u \\ a_v \end{Bmatrix} = [M] \begin{Bmatrix} a_x \\ a_y \end{Bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{Bmatrix} a_x \\ a_y \end{Bmatrix}$$

Objetivo

Tentar escrever os vetores que inicialmente se referem à base “xy” na base “uv”. Ou seja, deseja-se encontrar a matriz $[M]$.

São conhecidos

$$\begin{Bmatrix} \hat{u} \\ \hat{v} \end{Bmatrix} \quad \text{e} \quad \begin{Bmatrix} \hat{u} \\ \hat{v} \end{Bmatrix} = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \begin{Bmatrix} \hat{i} \\ \hat{j} \end{Bmatrix}$$

Vamos usar o fato

que $[M]$ é ortogonal, isto é, $[M]^{-1} = [M]^T$

“É mais fácil achar $[M]^{-1}$ do que $[M]$ porque nós conhecemos \hat{u} e \hat{v} em função de \hat{i} e \hat{j} e não o inverso”

Assim

$$\begin{Bmatrix} a_u \\ a_v \end{Bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{Bmatrix} a_x \\ a_y \end{Bmatrix} \quad \text{e} \quad \begin{Bmatrix} a_x \\ a_y \end{Bmatrix} = \begin{bmatrix} m_{11} & m_{21} \\ m_{12} & m_{22} \end{bmatrix} \begin{Bmatrix} a_u \\ a_v \end{Bmatrix}$$

Se for considerado que

$$\vec{a} = a_x \hat{i} + a_y \hat{j} = \{ a_x \ a_y \} \begin{Bmatrix} \hat{i} \\ \hat{j} \end{Bmatrix} \quad \text{e} \quad \vec{a} = a_u \hat{u} + a_v \hat{v} = \{ a_u \ a_v \} \begin{Bmatrix} \hat{u} \\ \hat{v} \end{Bmatrix}$$

Então

$$\vec{a} = \{ a_u \ a_v \} \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \begin{Bmatrix} \hat{i} \\ \hat{j} \end{Bmatrix}$$

Portanto

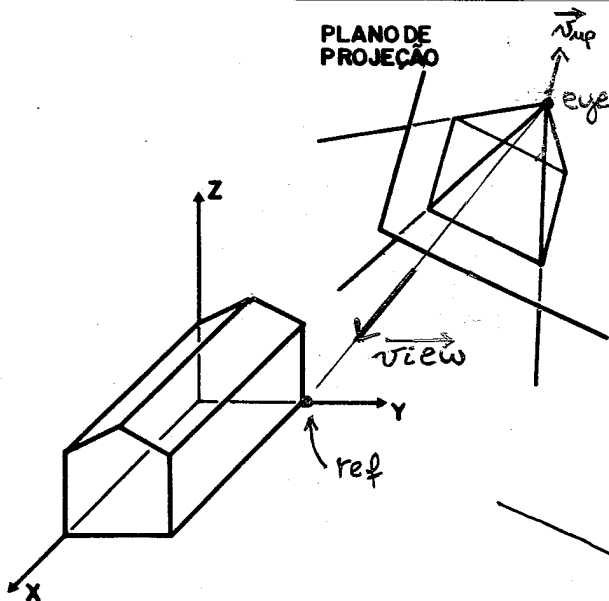
$$\{ a_x \ a_y \} = \{ a_u \ a_v \} \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \quad \text{ou melhor} \quad \begin{Bmatrix} a_x \\ a_y \end{Bmatrix} = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix} \begin{Bmatrix} a_u \\ a_v \end{Bmatrix}$$

Vê-se que $[M]^{-1} = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix}$

Finalmente $[M] = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix}$

“A matriz que faz a transformação (por rotação) da mudança de uma base para outra é construída por linhas, sendo que em cada linha coloca-se os cossenos diretores dos versores da nova base descritos na base antiga”

Mudança de Base $XYZ \rightarrow X_e Y_e Z_e$



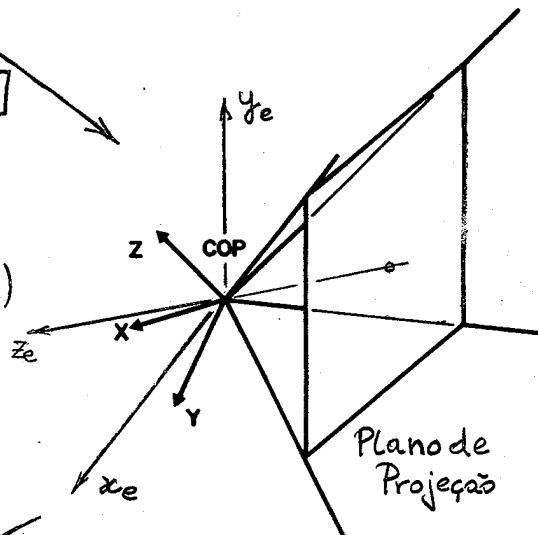
Translação

$$[T] = \begin{bmatrix} 1 & 0 & 0 & -eye_x \\ 0 & 1 & 0 & -eye_y \\ 0 & 0 & 1 & -eye_z \\ \hline 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação da Base

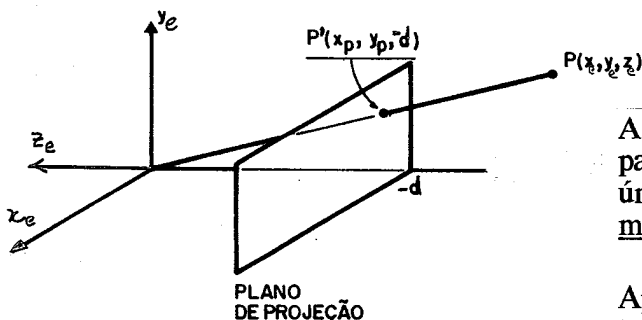
$$\vec{view} = (ref_x - eye_x, ref_y - eye_y, ref_z - eye_z)$$

$$\begin{cases} \hat{z}_e = \frac{-\vec{view}}{\|\vec{view}\|} \\ \hat{x}_e = \frac{\vec{v}_{up} \times \hat{z}_e}{\|\vec{v}_{up} \times \hat{z}_e\|} \\ \hat{y}_e = \hat{z}_e \times \hat{x}_e \end{cases}$$



[R]

$$[R] = \begin{bmatrix} x_{ex} & x_{ey} & x_{ez} & 0 \\ y_{ex} & y_{ey} & y_{ez} & 0 \\ z_{ex} & z_{ey} & z_{ez} & 0 \\ \hline 0 & 0 & 0 & 1 \end{bmatrix}$$

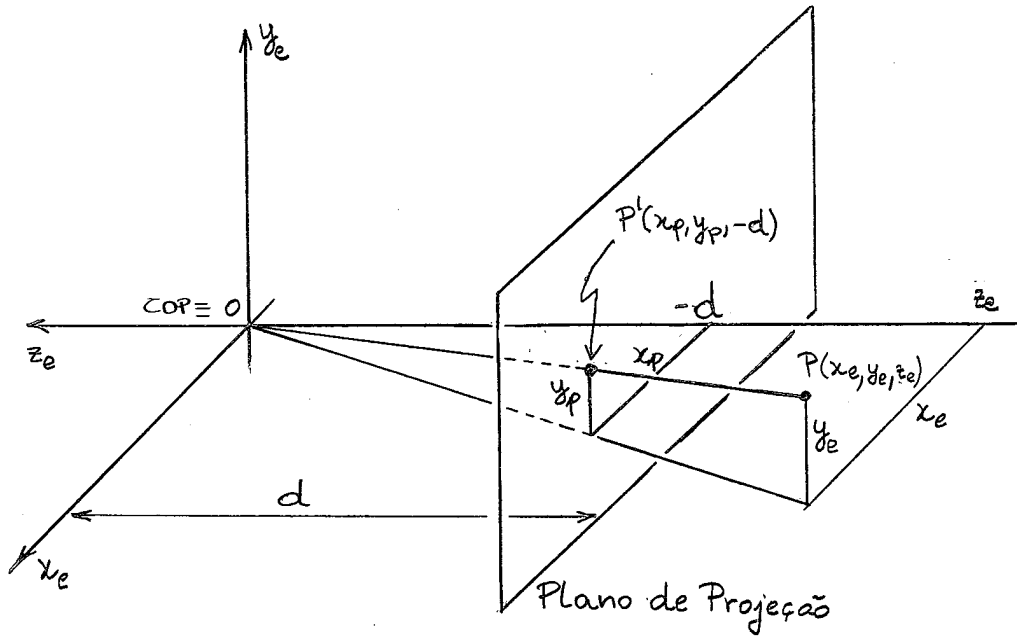


A mudança de base do espaço de modelagem para o espaço do olho pode ser grupada em uma única matriz de transformação, definida como matriz de posicionamento de câmera:

$$[C] = [R][T]$$

Após a translação [T] e a rotação [R], pronto para projetar no plano de projeção, fazendo uma projeção cônica (perspectiva) ou ortográfica.

Perspectiva Canônica



$$\frac{x_p}{x_e} = \frac{d}{-z_e} \Rightarrow x_p = \frac{d}{-z_e} \cdot x_e$$

$$\frac{y_p}{y_e} = \frac{d}{-z_e} \Rightarrow y_p = \frac{d}{-z_e} \cdot y_e$$

$$z_p = -d$$

[P]

$$\begin{Bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{Bmatrix} \triangleq \begin{bmatrix} d & 0 & 0 & | & 0 \\ 0 & d & 0 & | & 0 \\ 0 & 0 & d & | & 0 \\ \hline 0 & 0 & -1 & | & 0 \end{bmatrix} \begin{Bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{Bmatrix} = \begin{Bmatrix} d x_e \\ d y_e \\ d z_e \\ -z_e \end{Bmatrix} \triangleq \begin{Bmatrix} (d/-z_e) \cdot x_e \\ (d/-z_e) \cdot y_e \\ -d \\ 1 \end{Bmatrix}$$

ω $\div \omega$

Sistema de Coordenadas da Tela ($X_s Y_s Z_s$)

As coordenadas $(x_p, y_p, -d)$ são as coordenadas de um ponto (x, y, z) do objeto transformadas e projetadas no plano de projeção da tela. No entanto, não existe nenhuma informação sobre a profundidade do ponto para o interior da tela. Esta profundidade é uma informação importante pois possibilita a correta identificação dos objetos que estão mais próximos do plano de projeção e que, por isso, são os objetos visíveis. É, então, criado um espaço de coordenadas (x_s, y_s, z_s) , chamado de:

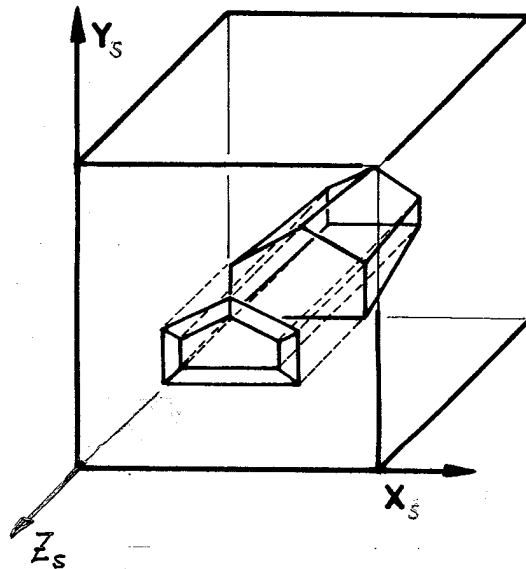
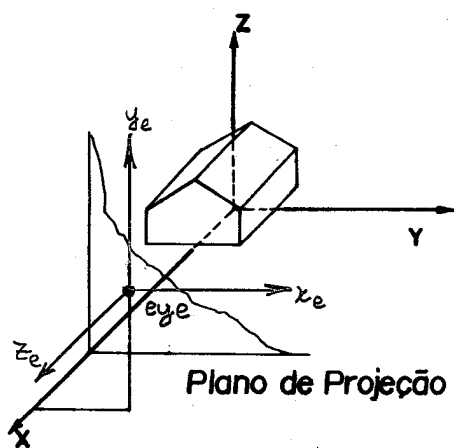
Sistema de coordenadas da tela:

$$x_s = x_p = \frac{d}{-z_e} x_e$$

$$y_s = y_p = \frac{d}{-z_e} y_e$$

$$z_s = \text{profundidade}$$

Na verdade, para desenhar na tela um ponto (x_s, y_s, z_s) , nós precisamos apenas usar as coordenadas x_s e y_s e ignorar a coordenada z_s . O desenho é uma projeção ortográfica do objeto transformado para o sistema de coordenadas da tela. Isto é, a transformação do objeto para o sistema de coordenadas da tela o distorce de tal maneira que o resultado da projeção ortográfica no plano $x_s y_s$ é igual à transformação de projeção de perspectiva [P] sobre o objeto.



Determinação da Profundidade Z_s

A transformação do objeto para o espaço da tela é conveniente pois faz com que o processo de remoção de linhas e superfícies ocultas de uma imagem seja feito com base em retas perpendiculares ao plano de projeção (projeção ortográfica). Isto é, para saber se um ponto de uma linha ou superfície é obscurecido por outro ponto basta comparar a profundidade z_s dos dois pontos: o que tiver a menor profundidade é o ponto que aparecerá na tela.

Entretanto, para que o resultado da transformação do objeto para o sistema de coordenadas da tela seja útil para a eliminação de linhas e superfícies escondidas, é necessário que se calcule a profundidade de uma linha, não somente para os seus pontos extremos, mas também para qualquer ponto intermediário. Para que a interpolação de um ponto intermediário no espaço da tela seja simples, é preciso que:

- Linhas retas no sistema de coordenadas do olho sejam transformadas para linhas retas no sistema de coordenadas da tela.
- Planos no sistema de coordenadas do olho sejam transformados para planos no sistema de coordenadas da tela.

Assim:

$$a x_e + b y_e + c z_e + q = 0 \quad (1)$$

$$a' x_s + b' y_s + c' z_s + q' = 0 \quad (2)$$

$$x_s = \frac{d}{-z_e} x_e \quad \rightarrow \quad x_e = \frac{-z_e}{d} x_s$$

$$y_s = \frac{d}{-z_e} y_e \quad \rightarrow \quad y_e = \frac{-z_e}{d} y_s$$

Substituindo x_e e y_e na equação (1) e dividindo por $-z_e$, tem-se:

$$\frac{a}{d} x_s + \frac{b}{d} y_s - c - \frac{q}{z_e} = 0 \quad (3)$$

Comparando (3) com (2), vê-se que:

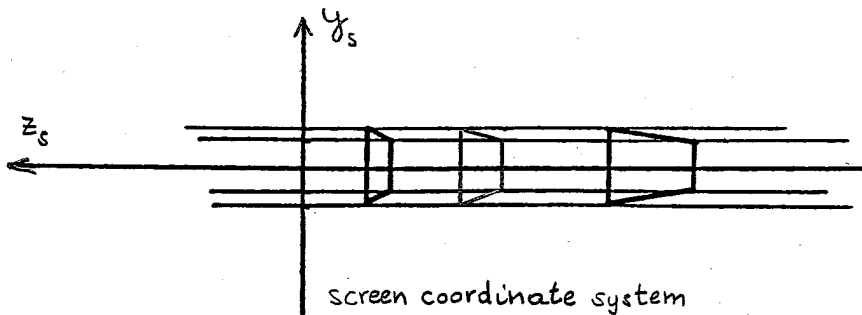
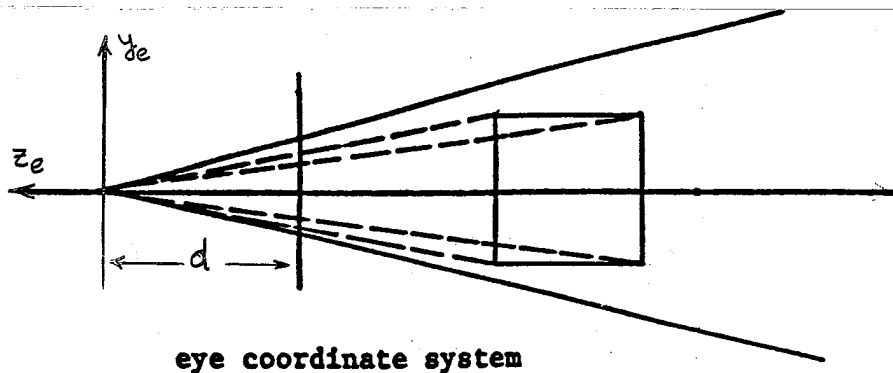
$$c' z_s + q' = -c - \frac{q}{z_e} \quad \rightarrow$$

$$z_s = \alpha + \frac{\beta}{z_e}$$

Escolha Conveniente dos Parâmetros da Profundidade Z_s

Existem infinitos valores de α e β que satisfazem a imposição de planaridade $z_s = \alpha + \beta/z_e$. Isto é, dados dois valores quaisquer para α e β , tem-se que qualquer reta ou plano no sistema de coordenadas do olho vai se transformar em um plano ou reta no sistema de coordenadas da tela.

De fato, o coeficiente α pode ser considerado uma translação de corpo rígido e β um coeficiente influenciando a quantidade de distorção por perspectiva.



A figura mostra o espaço distorcido da imagem (sistema de coordenadas da tela). Todos os objetos distorcidos mostrados satisfazem as condições de planaridade e resultam em soluções idênticas para valores de projeção ortográfica x_s e y_s . A transformação de perspectiva canônica fica assim:

$$\begin{Bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{Bmatrix} \triangleq \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & -\alpha & -\beta \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{Bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{Bmatrix} = \begin{Bmatrix} d \cdot x_e \\ d \cdot y_e \\ -\alpha \cdot z_e - \beta \\ -z_e \end{Bmatrix} \triangleq \begin{Bmatrix} (d/z_e) x_e \\ (d/z_e) y_e \\ \alpha + (\beta/z_e) \\ 1 \end{Bmatrix}$$

↙ [P]
↘ $\div (-z_e)$

Escolha Conveniente dos Parâmetros da Profundidade Z_s (cont.)

Os valores de α e β podem ser escolhidos de tal forma que todos os valores z_e que estão dentro do frustum de visão no sistema de coordenadas do olho mapeiem para valores convenientes de z_s no sistema de coordenadas da tela.

Uma possível solução é manter os valores da coordenada z_e para os planos frontal e posterior de cerceamento no espaço da tela. Isto é, associa-se uma coordenada $z_s = -near$ para pontos com $z_e = -near$ e associa-se $z_s = -far$ para pontos com $z_e = -far$. O resultado disso é:

$$\alpha = - (far + near) \quad \beta = - (far \ near)$$

$$z_s = - (far + near) - \frac{far \ near}{z_e}$$

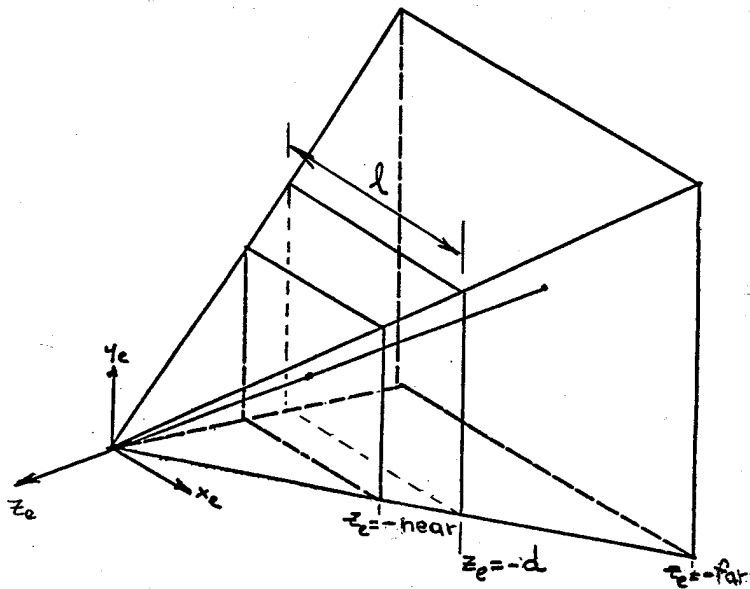
A matriz [P] com estes valores para α e β fica assim:

$$[P] = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & (far + near) & (far \ near) \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Distorção do Frustum de Visão para o Espaço da Tela

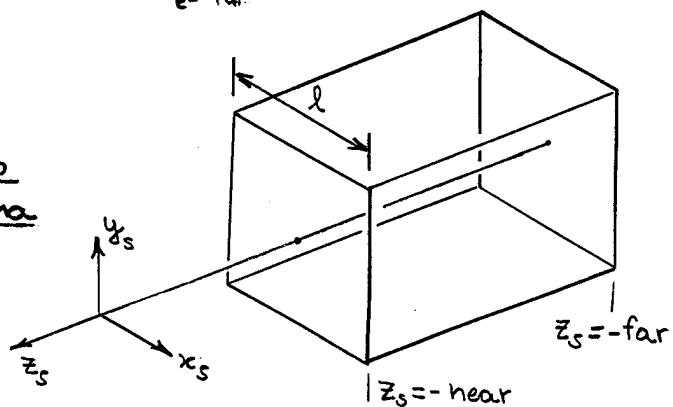
A transformação [P] distorce o frustum de visão de um tronco de pirâmide para um prisma, onde as coordenadas máximas e mínimas z_e nos planos de cerceamento anterior e posterior são preservadas.

As dimensões laterais deste prisma correspondem às dimensões da janela de visão que é definida pela interseção do plano de projeção com o frustum de visão no espaço do olho.



Frustum de visão no sistema de coordenadas do olho.

Frustum de visão, distorcido para um prisma, no sistema de coordenadas da tela.

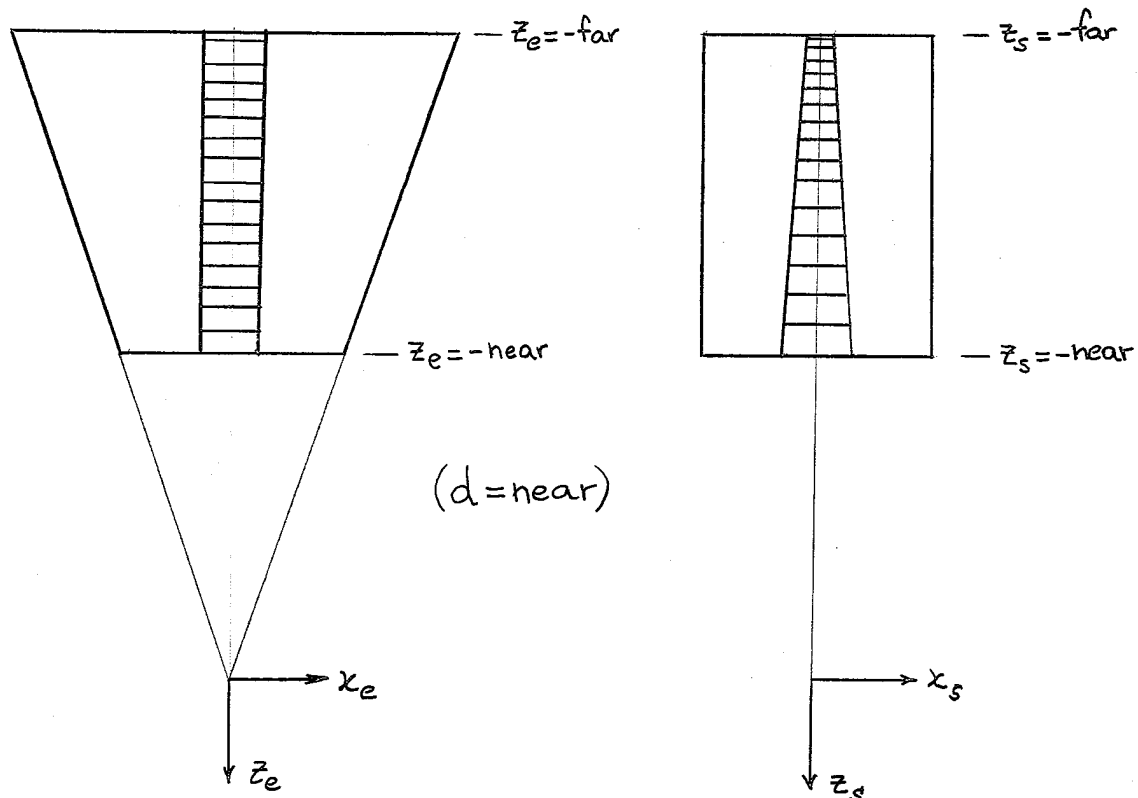


Não-linearidade na Distorção para o Espaço da Tela

A condição de planaridade $z_s = \alpha + \beta / z_e$, associada à projeção de perspectiva canônica, garante que linhas retas no sistema de coordenadas do olho sejam transformadas para linhas retas no sistema de coordenadas da tela.

Entretanto, a distorção introduzida por esta condição não preserva a métrica das linhas retas, isto é, pontos igualmente espaçados ao longo de uma reta no espaço do olho, quando transformados para o espaço da tela, não são igualmente espaçados, embora pertençam a uma mesma reta. Pode-se mostrar que eles só se mantêm igualmente espaçados quando a reta é paralela ao plano de projeção.

Esta não linearidade da distorção do espaço do olho para o espaço da tela pode ser entendida através do exemplo do trilho do trem mostrado abaixo.



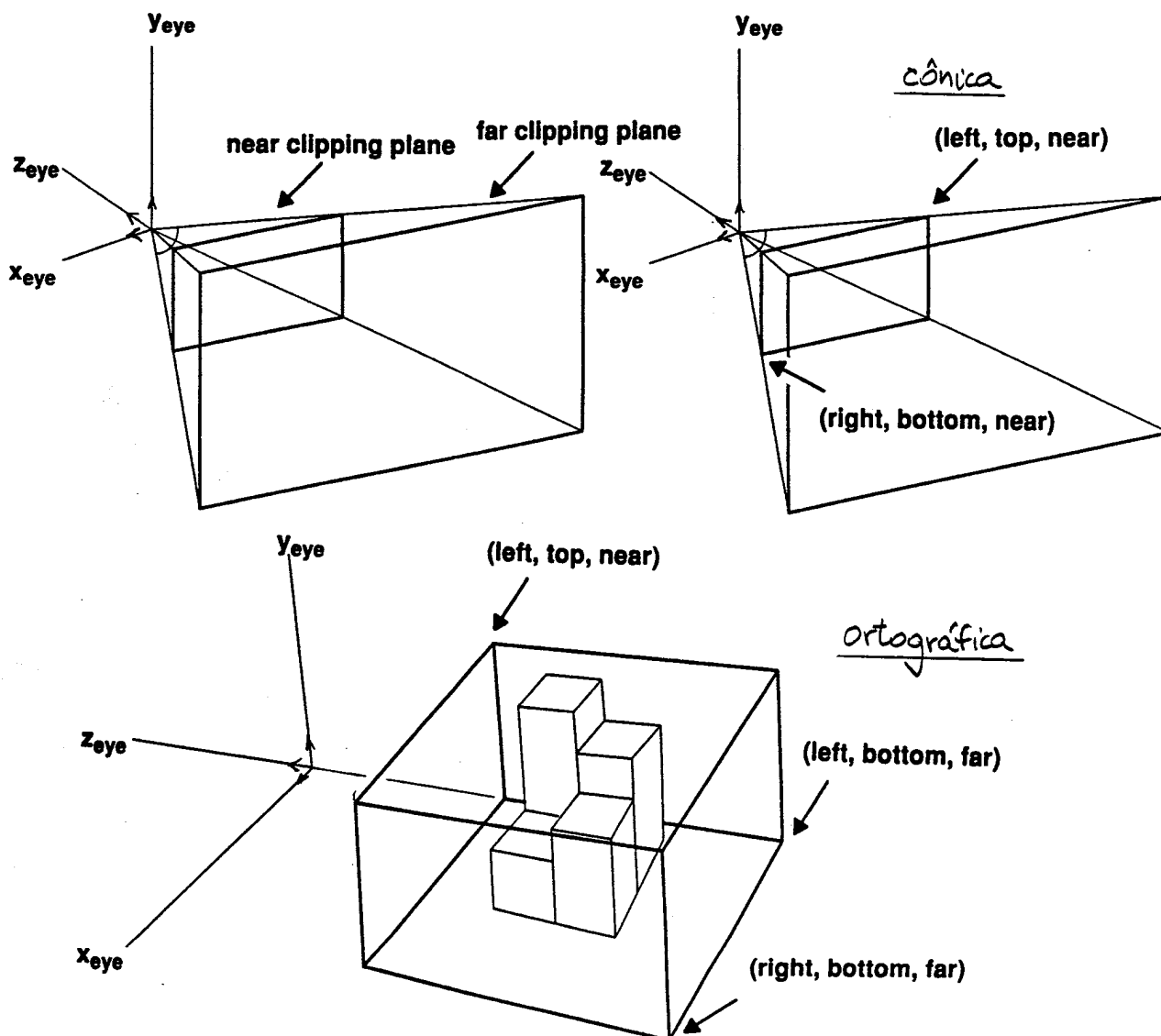
Volume de Visão

Além das transformações geométricas para visualização em 3D, é preciso definir o volume de visão, que é o volume correspondente ao frustum de visão. As porções dos objetos fora deste volume são cerceadas ("clipped"), isto é, não são desenhadas.

O volume de visão é definido, no sistema de coordenadas do olho, pelos dois planos de cerceamento anterior (*near*) e posterior (*far*), e pelos limites da janela de visão (*left, right, bottom, top*) definida no plano de projeção. É comum definir o plano de cerceamento anterior como sendo o plano de projeção. Isto é,

$$d = \textit{near}$$

A figura mostra os parâmetros que definem o volume de visão para projeções cônicas no caso de $d = \textit{near}$. Observe que os mesmos parâmetros servem para definir o volume de visão no caso de projeções ortográficas.



Normalização de Coordenadas

O cerceamento (“clipping”) é feito, no caso de projeções cônicas (perspectiva), depois da transformação para o sistema de coordenadas da tela. Alguns sistemas gráficos executam o cerceamento, para projeções cônicas e ortográficas, após uma normalização das coordenadas do volume de visão para um cubo com coordenadas variando de -1 a $+1$.

Nesta normalização de coordenadas, em geral, a menor coordenada $z_s = -1$ corresponde ao plano $z_e = -near$ e a maior coordenada $z_s = +1$ corresponde ao plano $z_e = -far$. Nota-se que é feito um espelhamento em relação ao plano $x_s y_s$. Isto é feito para que pontos no plano anterior de cerceamento tenham a menor profundidade z_s , e pontos no plano posterior de cerceamento tenham a maior profundidade z_s .

Para projeções ortográficas, a matriz de normalização de coordenadas $[N]$ pode ser deduzida a partir de uma transformação de translação do sistema de eixos para o centro do volume de visão, seguida de uma transformação de escala nas três direções, e finalmente de uma transformação de espelhamento em relação $x_s y_s$. Isto resulta na concatenação das duas matrizes mostradas abaixo:

$$[N] = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & -S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\Delta_x \\ 0 & 1 & 0 & -\Delta_y \\ 0 & 0 & 1 & -\Delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{onde } \begin{cases} S_x = \frac{2}{right - left} \\ S_y = \frac{2}{top - bottom} \\ S_z = \frac{2}{far - near} \end{cases} \quad \text{e} \quad \begin{cases} \Delta_x = \frac{right + left}{2} \\ \Delta_y = \frac{top + bottom}{2} \\ \Delta_z = -\frac{far + near}{2} \end{cases}$$

Normalização de Coordenadas (cont.)

A matriz de normalização para projeções ortográficas que resulta é:

$$[N] = \begin{bmatrix} \frac{2}{right - left} & 0 & 0 & -\frac{right + left}{right - left} \\ 0 & \frac{2}{top - bottom} & 0 & -\frac{top + bottom}{top - bottom} \\ 0 & 0 & \frac{-2}{far - near} & -\frac{far + near}{far - near} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para projeções cônicas, pode-se obter uma única matriz de transformação que engloba a perspectiva canônica [P] e a normalização de coordenadas [N]. Esta é a chamada matriz de frustum [F] = [N][P].

A concatenação de [P] e [N] para a obtenção de [F] pode ser entendida da seguinte maneira:

- A primeira transformação [P] a ser aplicada distorce o frustum de visão de um tronco de pirâmide para um prisma, onde as coordenadas máximas e mínimas z_e nos planos de cerceamento anterior e posterior são preservadas. As outras coordenadas máximas e mínimas deste prisma correspondem à janela de visão (*left, right, bottom, top*) definida no plano de projeção.
- A transformação [P] distorceu o frustum de visão para um prisma e fez o problema recair em uma normalização de projeções ortográficas. Portanto, basta aplicar a transformação [N] para normalizar as coordenadas.

A matriz [F] resultante é mostrada abaixo:

$$[F] = \begin{bmatrix} \frac{2d}{right - left} & 0 & \frac{right + left}{right - left} & 0 \\ 0 & \frac{2d}{top - bottom} & \frac{top + bottom}{top - bottom} & 0 \\ 0 & 0 & -\frac{far + near}{far - near} & \frac{-2 far near}{far - near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Resumo das Transformações Geométricas para Visualização em 3D

- 1º Mudança de base por translação da origem do sistema XYZ do espaço de modelagem para a posição do olho (origem do sistema de coordenadas do olho) — [T].
- 2º Mudança de base por rotação do sistema XYZ do espaço de modelagem para o sistema $X_e Y_e Z_e$ do olho — [R].
- 3º Se for projeção cônica, transformação do espaço $X_e Y_e Z_e$ para o sistema de coordenadas da tela $X_s Y_s Z_s$ (isto não é necessário para o caso de projeção ortográfica) — [P].
- 4º Normalização das coordenadas do volume de visão para um cubo com coordenadas variando de -1 a +1 — [N].
- 5º Cerceamento de pontos fora do volume de visão — < clip >. A divisão pela quarta coordenada homogênea é feita somente após o “clip” pois esta divisão projeta pontos atrás do olho no plano de projeção.
- 6º Remoção de pontos de linhas e superfícies ocultos por outros pontos mais próximos do plano de projeção — < hidden >.
- 7º Projeção ortogonal para o plano de projeção — [O]. Esta operação se resume na simples eliminação da coordenada z.
- 8º Transformação bidimensional afim dos pontos projetados na janela de visão (“window”) para a janela do dispositivo (“viewport”) — [V].

$$\begin{Bmatrix} x_v \\ y_v \end{Bmatrix} = [V] [O] \langle \text{hidden} \rangle \langle \text{clip} \rangle [N] [P] [R] [T] \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix}$$

$$\begin{Bmatrix} x_v \\ y_v \end{Bmatrix} = [V] [O] \langle \text{hidden} \rangle \langle \text{clip} \rangle [F] [C] \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix}$$