Pontifícia Universidade Católica do Rio de Janeiro — PUC-Rio

TRANSPARÊNCIAS:

TRANSFORMAÇÕES GEOMÉTRICAS PARA CONTROLE DE VISUALIZAÇÃO 3D

Luiz Fernando Martha

Referências

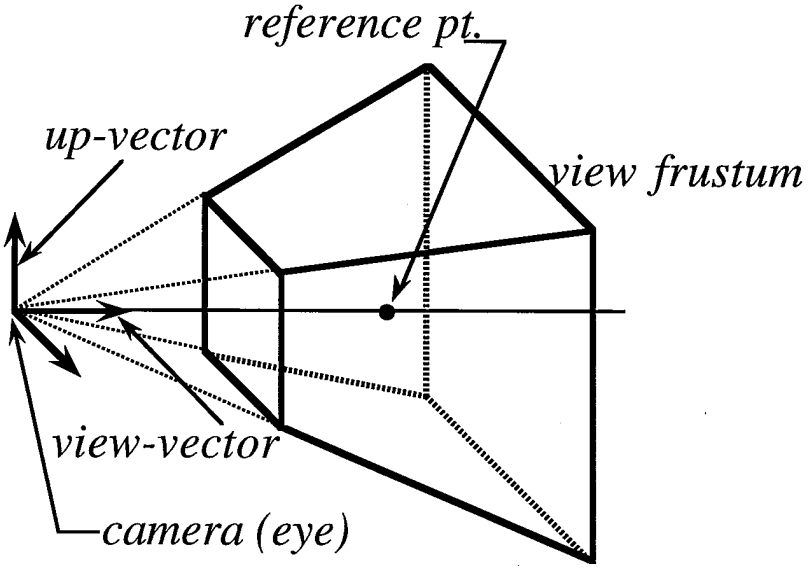
1. "Planar Geometric Projections and Viewing Transformations" – Carlbom, I.; Paciorek, J. – *ACM Computing Surveys*, Vol. 10, No. 4, Dec. 1978.

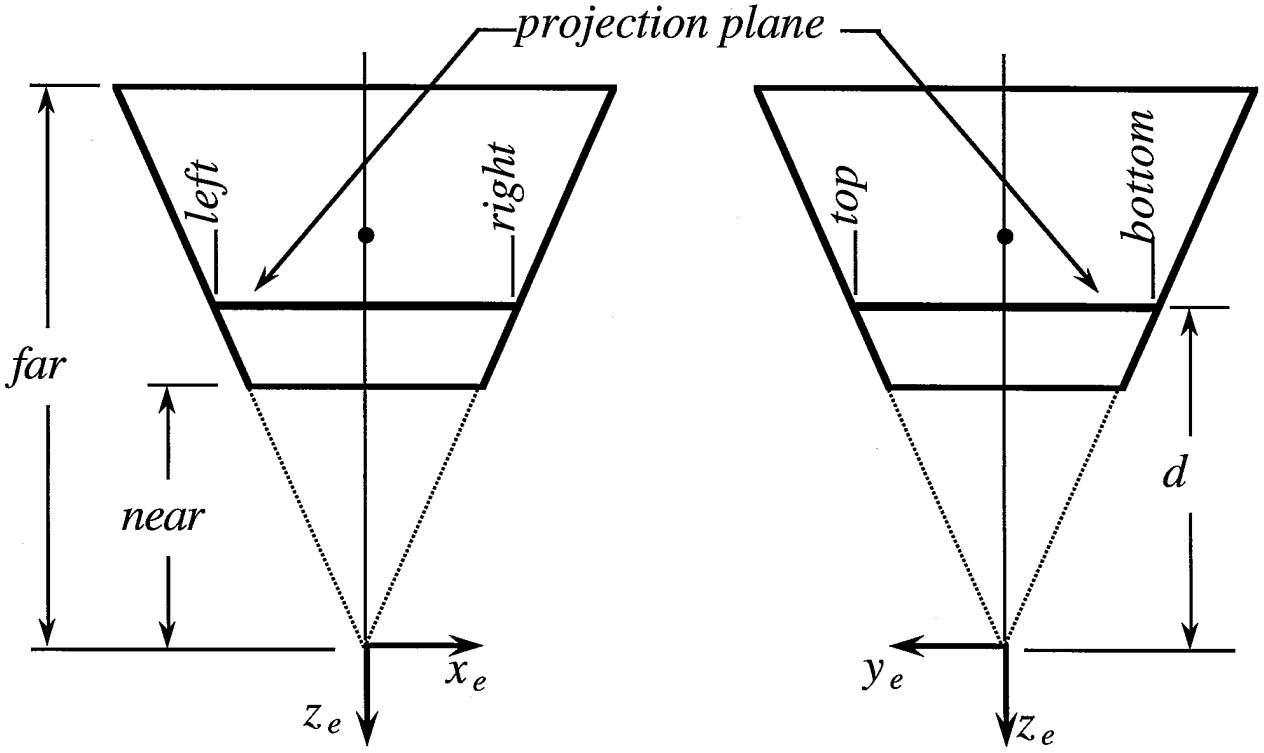
2. *OpenGL Programming Guide – The Official Guide to Learning OpenGL*, Release 1 – Neider, J.; Davis, T.; Woo, M. – Addison-Wesley Publishing Company, Reading, Massachusetts, 1993.

3. Manual IBM AIX Version 3 for RISC System/6000 – *Graphics Programming Concepts*, March 1990.

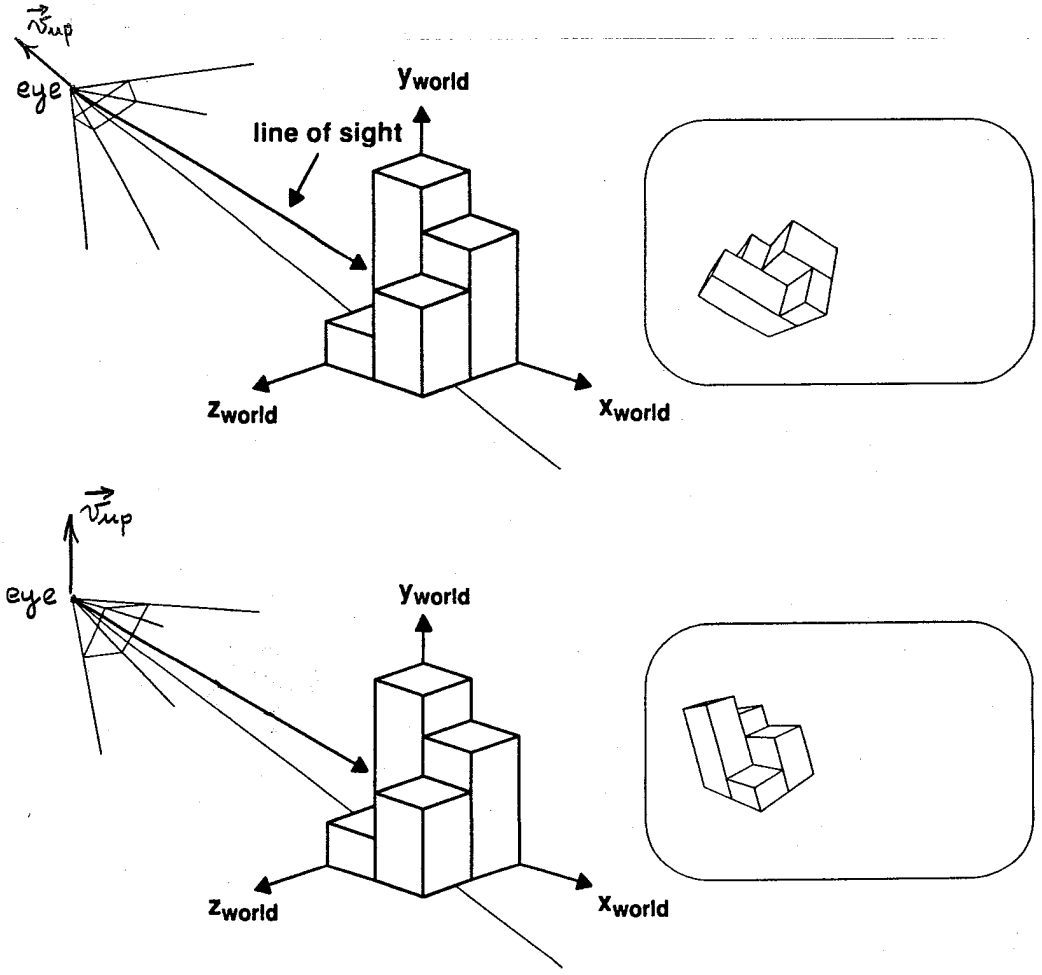
Rio de Janeiro, Novembro de 1994

Modelo de Câmera





Parâmetros de Visualização
(Posicionamento da Câmera)



Parâmetros necessários:

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

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* – $\boldsymbol{vup}$)

(vup_x, vup_y, vup_z)

Sistema de coordenadas do olho

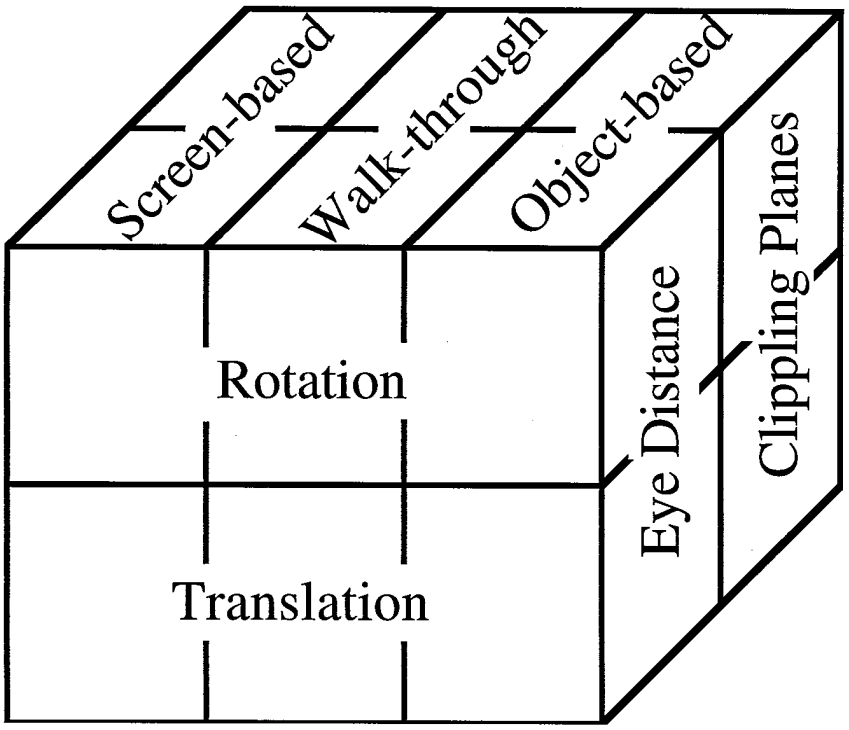
$\boldsymbol{view} = (ref_x, ref_y, ref_z) - (eye_x, eye_y, eye_z)$

$z_e = - \boldsymbol{view} \; / \; \|\boldsymbol{view}\|$

$\boldsymbol{x_e} = (\boldsymbol{vup} \times z_e) \; / \; \|\boldsymbol{vup} \times z_e\|$

$\boldsymbol{y_e} = z_e \times \boldsymbol{x_e}$

Uma taxonomia para o controle de visualização em 3D



Controle da visualização através de movimentos do *mouse*

Movimento do *mouse* em *pixels*

$$\boldsymbol{m} = (m_x , m_y)$$

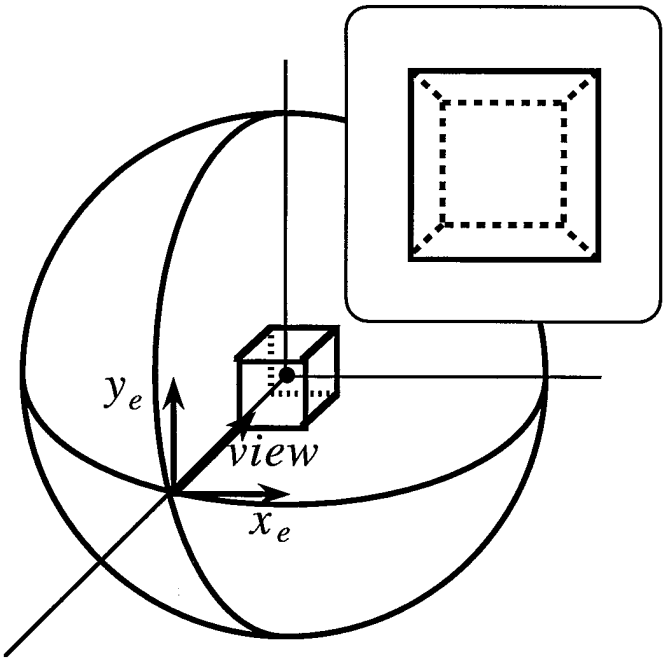
Movimento do *mouse* normalizado para controle de rotações

$$\delta_x = m_x / hsize$$
$$\delta_y = m_y / vsize$$
$$\delta_{\boldsymbol{m}} = (\delta_x , \delta_y)$$

Movimento do *mouse* normalizado para controle de translações

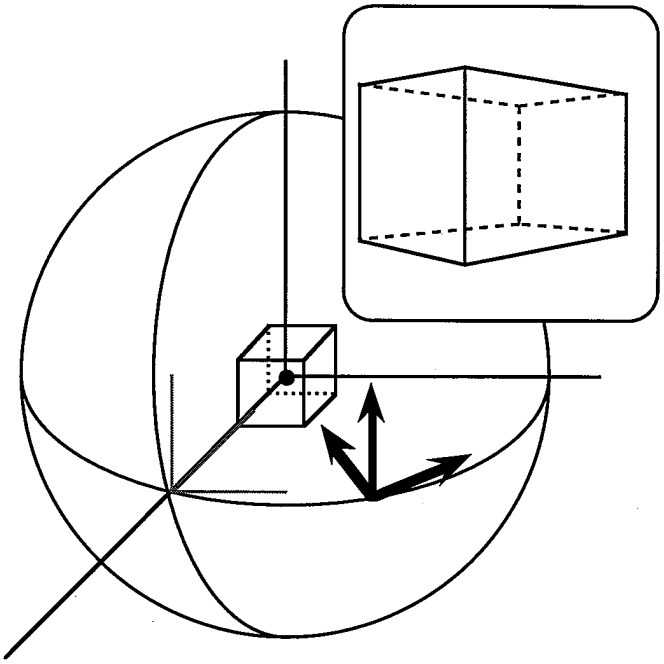
$$\Delta_x = \delta_x \ (right - left)$$
$$\Delta_y = \delta_y \ (top - bottom)$$
$$\Delta_{\boldsymbol{m}} = (\Delta_x , \Delta_y)$$

Posição inicial da câmera e imagem resultante

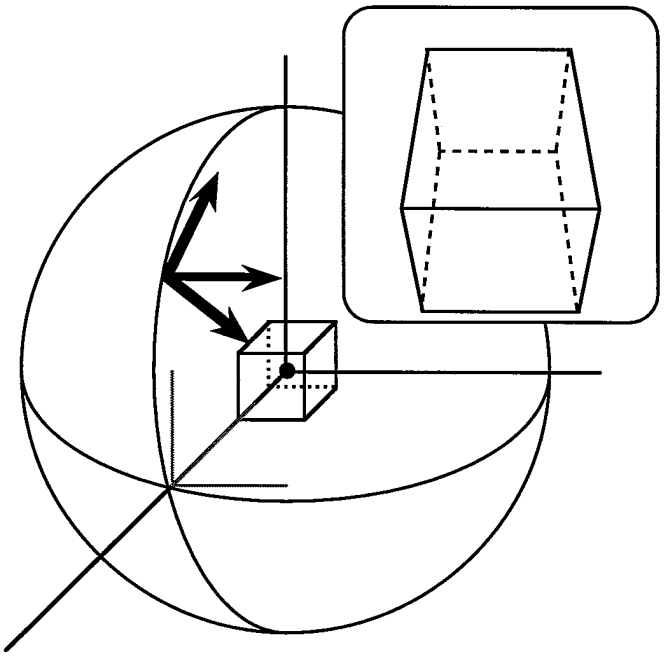


Controle de rotação tipo *Screen-based*

Movimento horizontal do *mouse*



Movimento vertical do *mouse*



Controle de rotação tipo *Screen-based* (cont.)

Eixo de rotação na tela

$$r_m = (-\delta_y , \delta_x)$$

Eixo de rotação no espaço do objeto

$$r = \delta_x \, y_e - \delta_y \, x_e$$

Ângulo de rotação

(movimento do *mouse* de um lado ao outro da janela $\Rightarrow 180°$)

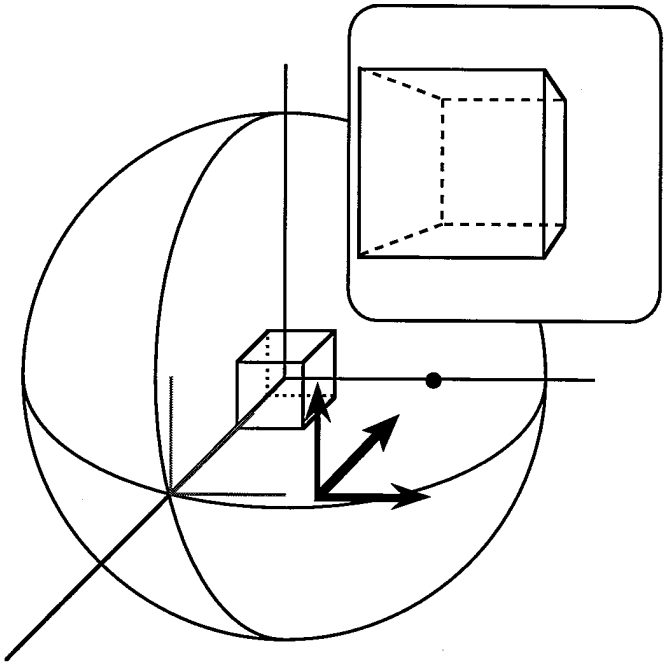
$$angle = -180° \sqrt{\delta_x^2 + \delta_y^2}$$

Atualização dos parâmetros de visualização

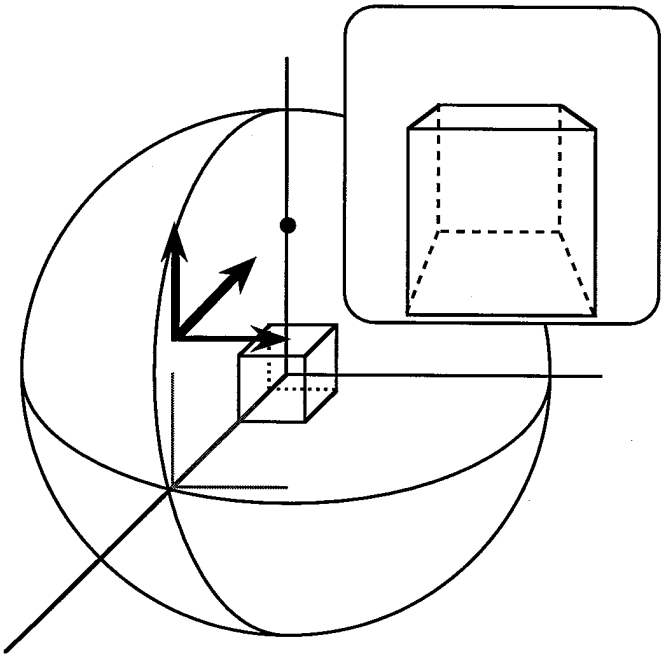
RotateAroundAxisAboutPoint($angle$, r_x, r_y, r_z, ref_x, ref_y, ref_z, eye_x, eye_y, eye_z *)*
RotateAroundAxisAboutOrigin($angle$, r_x, r_y, r_z, vup_x, vup_y, vup_z *)*

Controle de translação do tipo *Screen-based*

Movimento horizontal do *mouse*



Movimento vertical do *mouse*



Controle de translação tipo *Screen-based* (cont.)

Vetor de translação na tela

$\Delta_m = (\Delta_x , \Delta_y)$

Vetor de translação no espaço do objeto

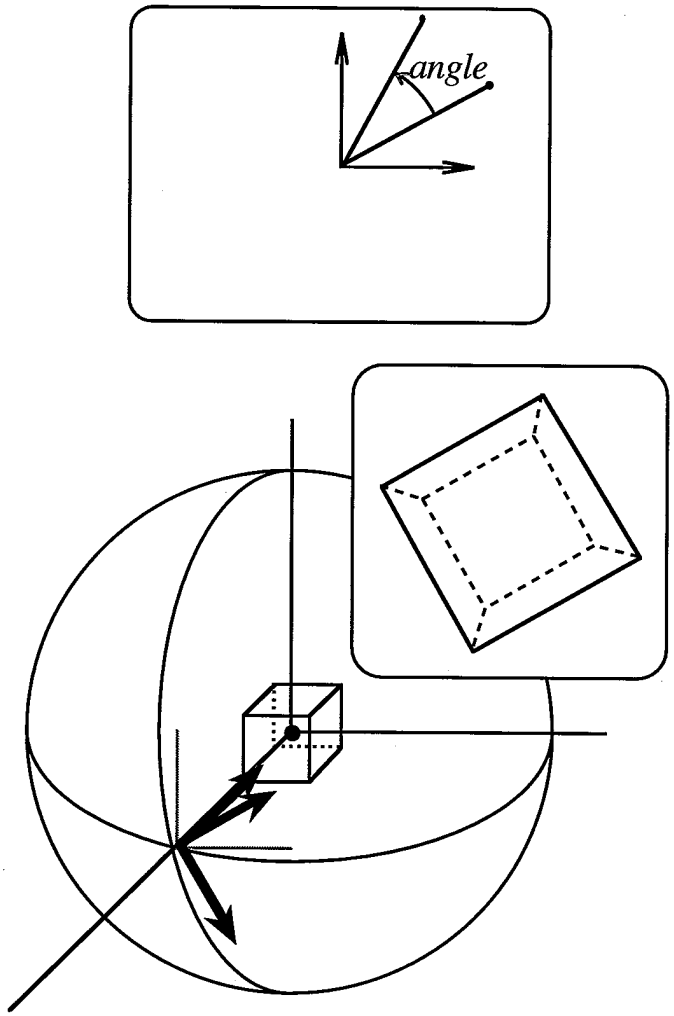
$t = \Delta_x \, x_e + \Delta_y \, y_e$

Atualização dos parâmetros de visualização

Translate($-t_x$, $-t_y$, $-t_z$, eye_x, eye_y, eye_z)
Translate($-t_x$, $-t_y$, $-t_z$, ref_x, ref_y, ref_z)

Controle de rotação axial (*spin*) tipo *Screen-based*

Ângulo de giro do *mouse* em relação ao centro da janela



Eixo de rotação no espaço do objeto

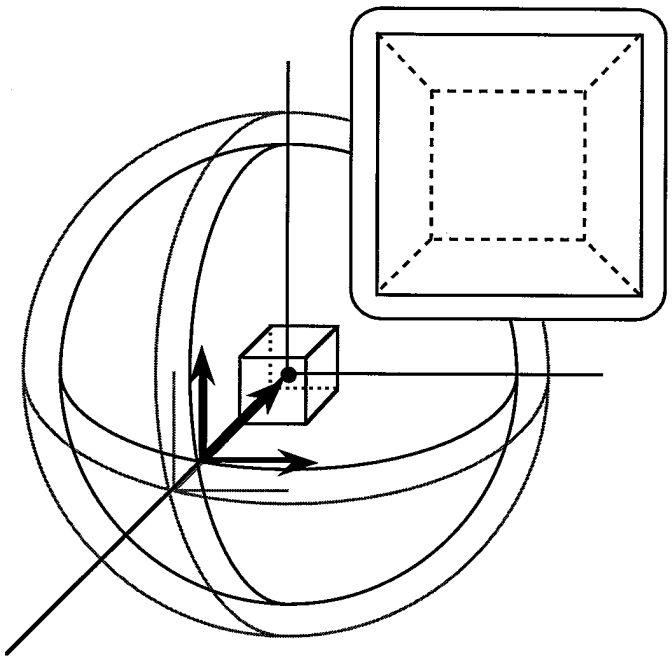
$r = z_e$

Atualização dos parâmetros de visualização

RotateAroundAxisAboutOrigin($-angle$, r_x, r_y, r_z, vup_x, vup_y, vup_z)

Controle de translação radial tipo *Screen-based*

Movimento Δ do *mouse* na janela (usado como um potenciômetro)



Vetor de translação no espaço do objeto

$t = \Delta z_e$

Atualização dos parâmetros de visualização

Translate($-t_x$, $-t_y$, $-t_z$, eye_x, eye_y, eye_z)

Manipulação centralizada na câmera (*Walk-through*)

(É sempre o inverso da manipulação do tipo *Screen-based*)

Rotação horizontal e vertical

$\boldsymbol{r_m} = (-\delta_y , \delta_x)$

$\boldsymbol{r} = \delta_x \boldsymbol{y_e} - \delta_y \boldsymbol{x_e}$

$angle = -180° \sqrt{\delta_x^2 + \delta_y^2}$

RotateAroundAxisAboutPoint($angle, r_x, r_y, r_z, eye_x, eye_y, eye_z, ref_x, ref_y, ref_z$ *)*

RotateAroundAxisAboutOrigin($angle, r_x, r_y, r_z, vup_x, vup_y, vup_z$ *)*

Translação horizontal e vertical

$\boldsymbol{\Delta_m} = (\Delta_x , \Delta_y)$

$\boldsymbol{t} = \Delta_x \boldsymbol{x_e} + \Delta_y \boldsymbol{y_e}$

Translate($t_x, t_y, t_z, eye_x, eye_y, eye_z$ *)*

Translate($t_x, t_y, t_z, ref_x, ref_y, ref_z$ *)*

Rotação axial

(ângulo computado pelo giro do *mouse* em relação ao centro da janela)

$\boldsymbol{r} = \boldsymbol{z_e}$

RotateAroundAxisAboutOrigin($angle, r_x, r_y, r_z, vup_x, vup_y, vup_z$ *)*

Translação radial

(movimento Δ do mouse na janela usado como um potenciômetro)

$\boldsymbol{t} = \Delta \boldsymbol{z_e}$

Translate($t_x, t_y, t_z, eye_x, eye_y, eye_z$ *)*

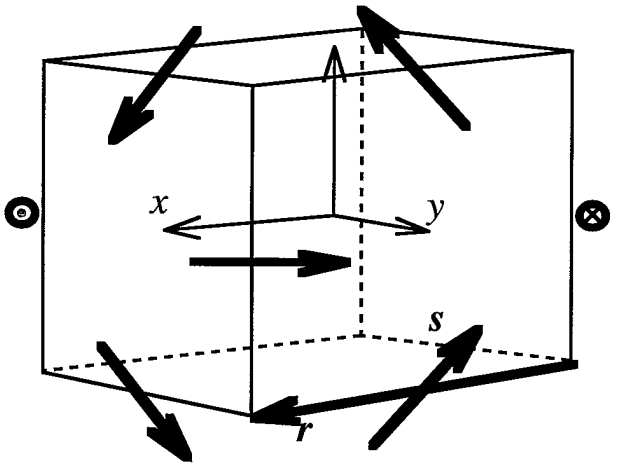
Translate($t_x, t_y, t_z, ref_x, ref_y, ref_z$ *)*

(neste caso o ponto de referência também é atualizado)

Manipulação centralizada no objeto proposta

(Ideal para objetos com caixa envolvente natural)

Rotação em torno de um eixo principal ***r*** da caixa envolvente



Vetor tangente ***s*** descrito no sistema de coordenadas do olho (*eye*)

$$\boldsymbol{s_e} = (s_{ex}, s_{ey}, s_{ez})$$

Vetor ***s*** projetado na tela (plano de projeção) e normalizado

$$\boldsymbol{v} = (v_x, v_y)$$
$$v_x = s_{ex} / \sqrt{s_{ex}^2 + s_{ey}^2}$$
$$v_y = s_{ey} / \sqrt{s_{ex}^2 + s_{ey}^2}$$

Ângulo de rotação em torno do centro ***c*** da caixa envolvente

(proporcional ao produto interno entre $\boldsymbol{v}$ e δ_m)

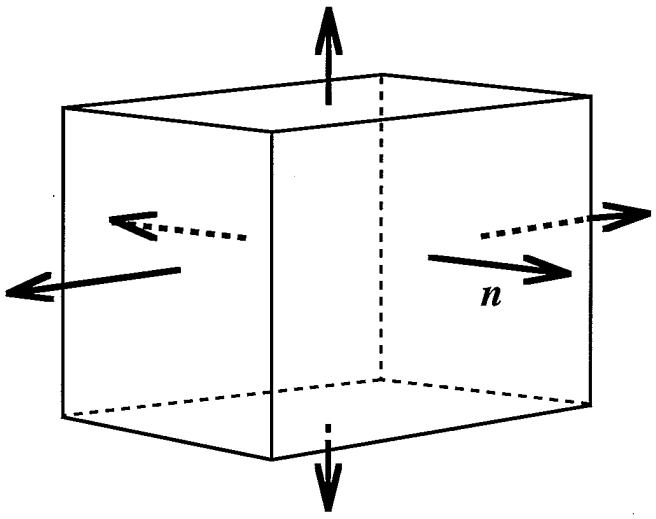
$$angle = -180^\circ \ (\delta_x v_x + \delta_y v_y)$$

Atualização dos parâmetros de visualização

RotateAroundAxisAboutPoint(angle, $r_x, r_y, r_z, c_x, c_y, c_z, eye_x, eye_y, eye_z$)
RotateAroundAxisAboutPoint(angle, $r_x, r_y, r_z, c_x, c_y, c_z, ref_x, ref_y, ref_z$)
RotateAroundAxisAboutOrigin(angle, $r_x, r_y, r_z, vup_x, vup_y, vup_z$)

Manipulação centralizada no objeto proposta (cont.)

Translação na direção de uma normal ***n*** da caixa envolvente



Vetor normal ***n*** descrito no sistema de coordenadas do olho (*eye*)

$\boldsymbol{n_e} = (n_{ex}, n_{ey}, n_{ez})$

Vetor ***n*** projetado na tela (plano de projeção) e normalizado

$\boldsymbol{u} = (u_x, u_y)$

$u_x = n_{ex} / \sqrt{n_{ex}^2 + n_{ey}^2}$

$u_y = n_{ey} / \sqrt{n_{ex}^2 + n_{ey}^2}$

Vetor de translação no espaço do objeto
(proporcional ao produto interno entre ***u*** e Δ_m, na direção de ***n***)

$\boldsymbol{t} = (\Delta_x\, u_x + \Delta_y\, u_y)\, \boldsymbol{n}$

Atualização dos parâmetros de visualização

Translate($-t_x$, $-t_y$, $-t_z$, eye_x, eye_y, eye_z)
Translate($-t_x$, $-t_y$, $-t_z$, ref_x, ref_y, ref_z)