Aula 9

Professores:

Anselmo Montenegro
Esteban Clua

Conteúdo:

- Transformações geométricas no espaço

ATENÇÃO: O professor menciona aula 8, mas na verdade o número correto é 9.

Fundação CECIERJ
Consórcio cederj

Transformações geométricas no espaço

Introdução

- As transformações geométricas são operações fundamentais para a modelagem, visualização e interação com objetos gráficos 3D.
- Descreveremos os seguintes tópicos:
  - Escalas, rotações e translações no espaço.
  - Esquemas para ***representação de orientações***.
  - ***Composição de transformações***:
    - Instanciação de objetos.
    - Hierarquia.

Transformações geométricas no espaço

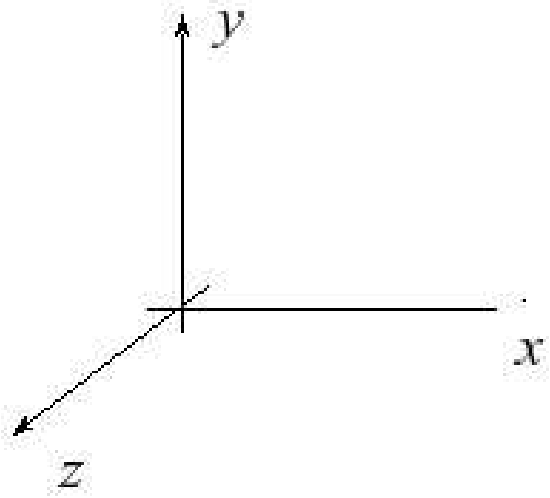
Introdução

- Transformações de escala, rotação e translação são fundamentais para a ***criação de cenas compostas por diversos objetos***.

- As matrizes de translação e escala são de fato uma simples extensão das matrizes de transformação definidas no plano.

Transformações geométricas no espaço

Translações e escalas



Translação

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

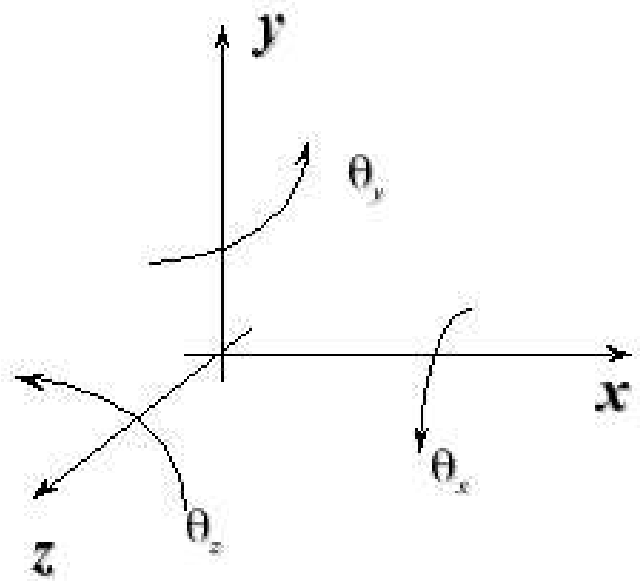
Escala

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformações geométricas no espaço

Rotações

- As operações de rotação no espaço são mais complexas do que no plano.
- Uma extensão natural é definirmos a rotação de um objeto a partir da ***rotação em torno dos eixos cartesianos***.



Transformações geométricas no espaço

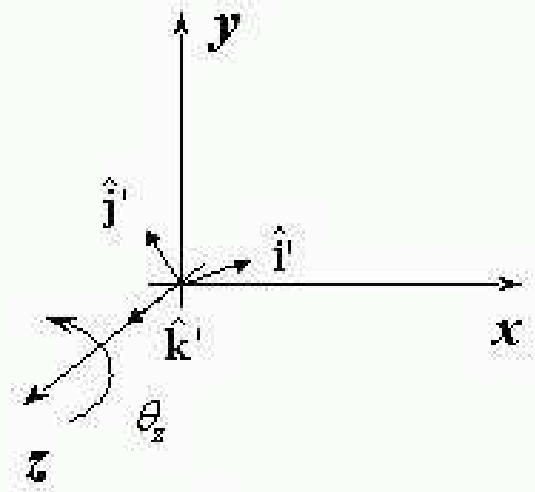
Rotações

- Podemos facilmente definir as matrizes de rotação em cada eixo.

- As ***colunas*** de uma matriz de rotação em torno de um certo eixo cartesiano são dados pela ***transformação dos vetores da base canônica***.

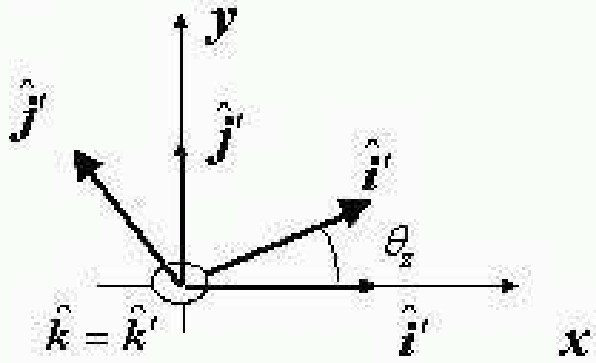
Transformações geométricas no espaço

Rotações no eixo z



$$\hat{i}' = \begin{bmatrix} \cos\theta_z \\ \sin\theta_z \\ 0 \end{bmatrix} \quad \hat{j}' = \begin{bmatrix} -\sin\theta_z \\ \cos\theta_z \\ 0 \end{bmatrix} \quad \hat{k}' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

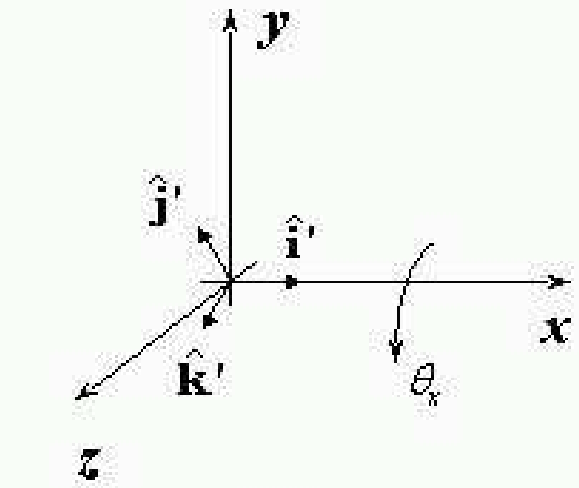
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

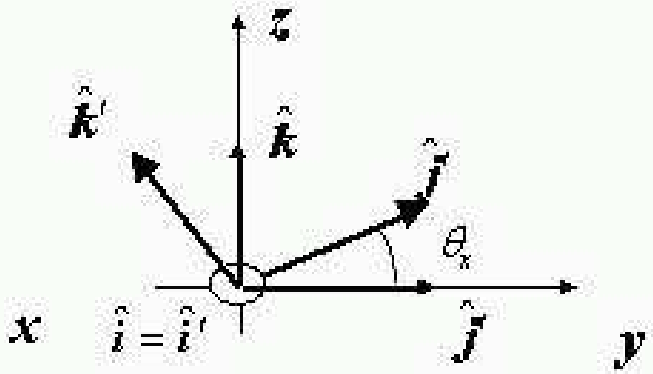


$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformações geométricas no espaço

Rotações no eixo x





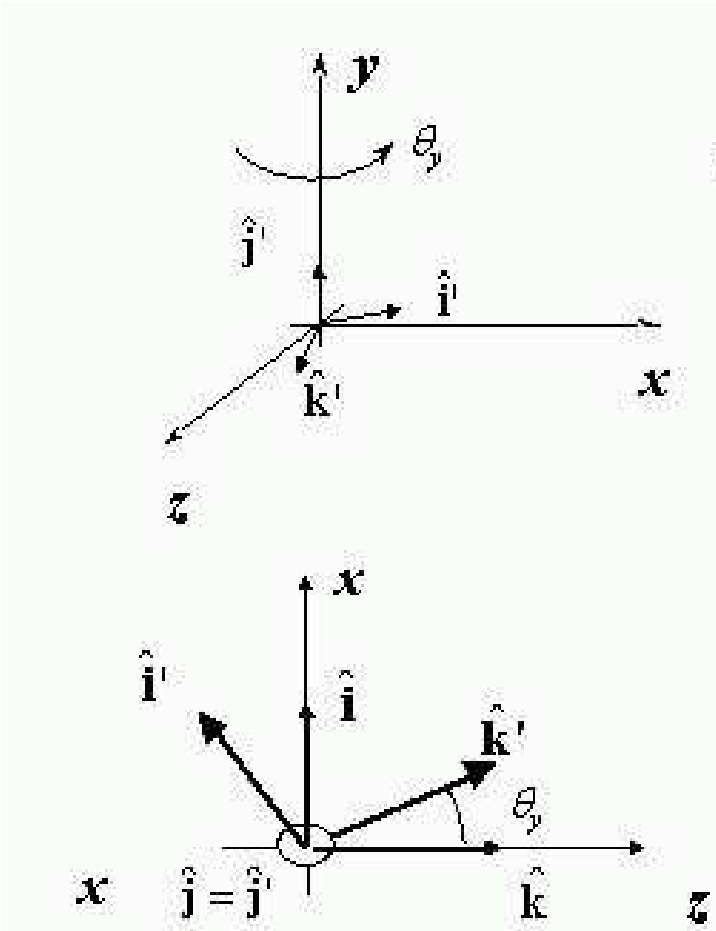
$$\hat{i}' = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \hat{j}' = \begin{bmatrix} 0 \\ \cos\theta_x \\ \sin\theta_x \end{bmatrix} \quad \hat{k}' = \begin{bmatrix} 0 \\ -\sin\theta_x \\ \cos\theta_x \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformações geométricas no espaço

Rotações no eixo y



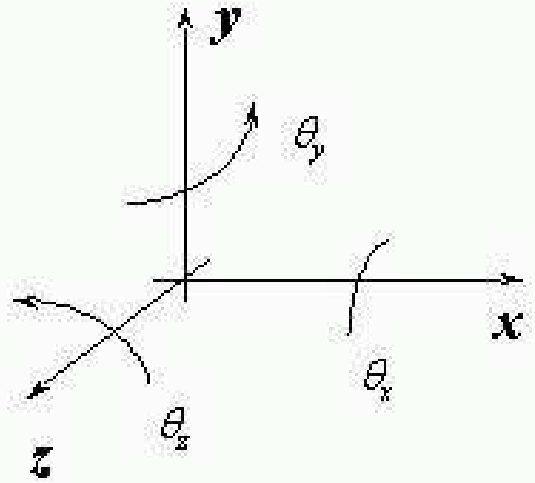
$$\hat{i}' = \begin{bmatrix} \cos\theta_y \\ 0 \\ -\sin\theta_y \end{bmatrix} \quad \hat{j}' = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \hat{k}' = \begin{bmatrix} \sin\theta_y \\ 1 \\ \cos\theta_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformações geométricas no espaço

Rotações nos eixos cartesianos



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformações geométricas no espaço

Matrizes de transformação

- Em geral, uma ***matriz de transformação em coordenadas homogêneas*** tem a seguinte estrutura:

$$\begin{bmatrix} M & T \\ s & 1 \end{bmatrix}$$

- Se a matriz *M* e o vetor *s* tem dimensões 2x2 e 1x2, respectivamente, então a transformação ocorre no ***plano homogêneo***.
- Se as dimensões forem 3x3 e 3x1, respectivamente, então a transformação ocorre no ***espaço homogêneo***.

Transformações geométricas no espaço

Matrizes de transformação

- A forma matricial homogênea pode representar

  a) Transformações lineares.

  b) Translações

  c) Transformações afins.

$$\begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} \quad (a)$$

$$\begin{bmatrix} I & T \\ 0 & 1 \end{bmatrix} \quad (b)$$

$$\begin{bmatrix} M & T \\ 0 & 1 \end{bmatrix} \quad (c)$$

Transformações geométricas no espaço

Matrizes de transformação

- A matriz que representa uma ***transformação afim*** representa uma transformação linear seguida de uma translação:

$$\begin{bmatrix} I & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} M & T \\ 0 & 1 \end{bmatrix}$$

- Caso invertêssemos a ordem teríamos

$$\begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} M & MT \\ 0 & 1 \end{bmatrix}$$

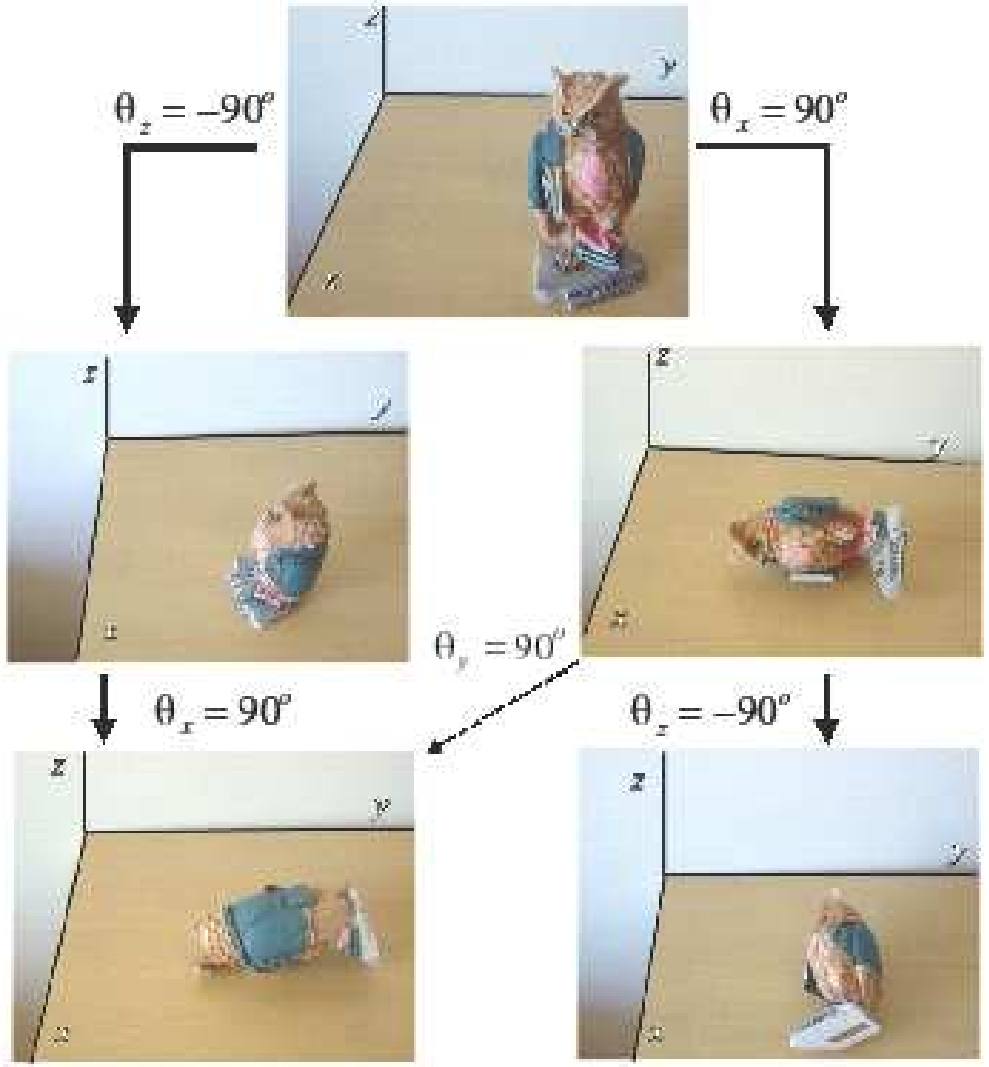
Transformações geométricas no espaço

Matrizes de transformação

- Como podemos ver, a ordem influi no resultado.
- Na segunda ordem, a translação não pode ser lida diretamente da última coluna última matriz.
- Como a última linha é o vetor [0..0 1] então ***estas matrizes mantêm os pontos no plano*** $w = 1$.
- Nas transformações projetivas, que serão vistas mais tarde, a última linha assume outros valores e os pontos podem ser deslocados do plano $w=1$.

Transformações geométricas no espaço

Rotações e orientações



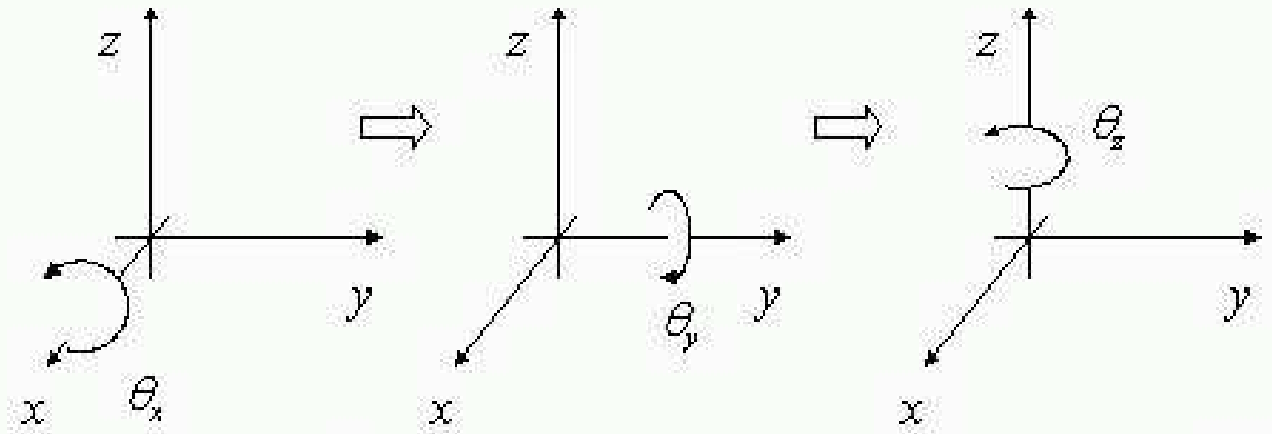
Transformações geométricas no espaço

Ângulos de Euler

- Já que as ***rotações não comutam devemos adotar uma ordem específica***.
- Esta forma de representar orientações é denominada ***Ângulos de Euler.***
- Na literatura de aeronáutica estas rotações são chamadas de
  - Roll - giro em torno do eixo longitudinal
  - Pitch - ângulo de ataque.
  - Yaw - giro em torno do eixo vertical.

Transformações geométricas no espaço

Ângulos de Euler



$$\mathbf{R}(\theta_x,\theta_y,\theta_z)=\begin{bmatrix} c_y c_z & c_y s_z & -s_y & 0 \\ s_x s_y c_z - c_x s_z & s_x s_y s_z + c_x c_z & s_x c_y & 0 \\ c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & c_x c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$s_x = \sin(\Theta_x),\ s_y = \sin(\Theta_y),\ s_z = \sin(\Theta_z)$$

$$c_x = \cos(\Theta_x),\ c_y = \cos(\Theta_y),\ c_z = \cos(\Theta_z)$$

Transformações geométricas no espaço

Ângulos de Euler

- Problemas:

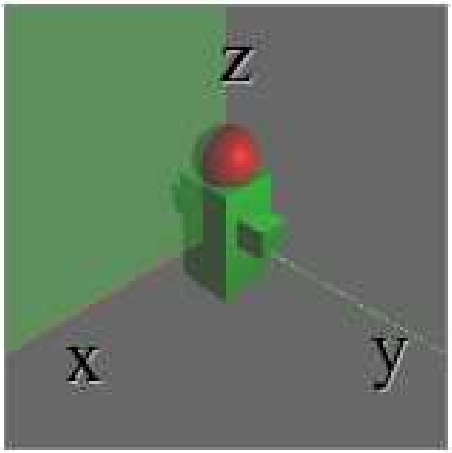
  - ***Gimbal lock***: perda de graus de liberdade em certas configurações.

  - Não são parâmetros adequados para interpolações.

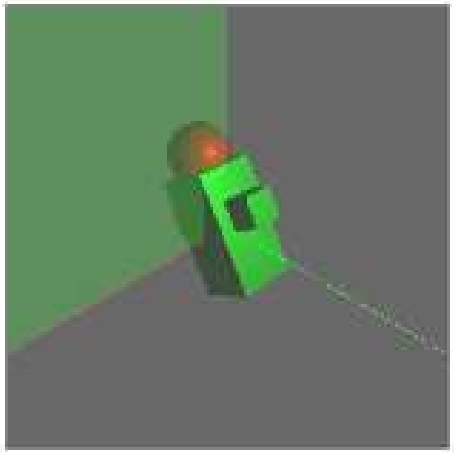
Transformações geométricas no espaço

Ângulos de Euler - Gimbal Lock

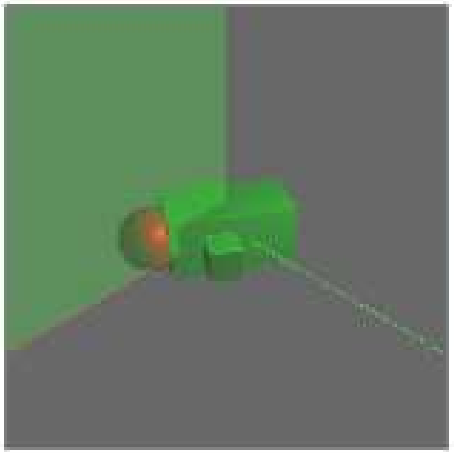
- Animador deseja rodar o boneco de lado ($\Theta_x = 30^o$ graus, incliná-lo para frente ($\Theta_y = 90^o$) e levantar seu braço esquerdo.
- A última operação não será possível.



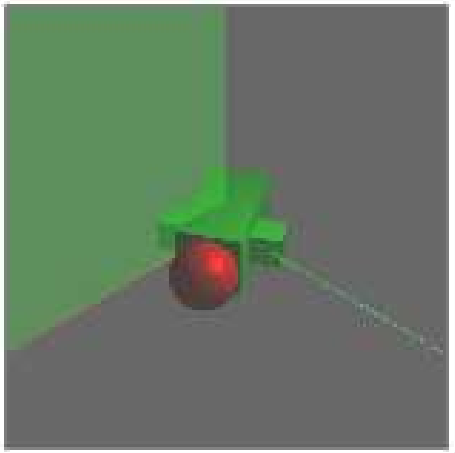
Original



$\Theta_x = 30^o$



$\Theta_x = 30^o$
$\Theta_y = 90^o$

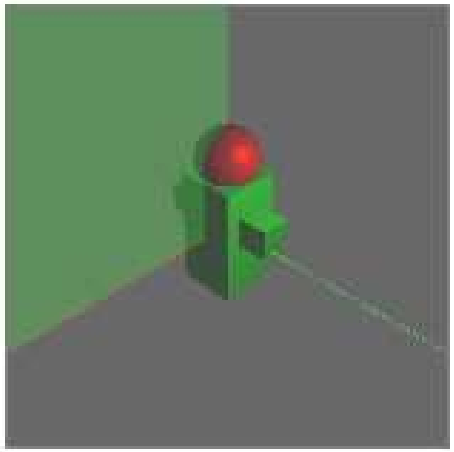


$\Theta_x = 30^o$
$\Theta_y = 90^o$
$\Theta_z = 60^o$

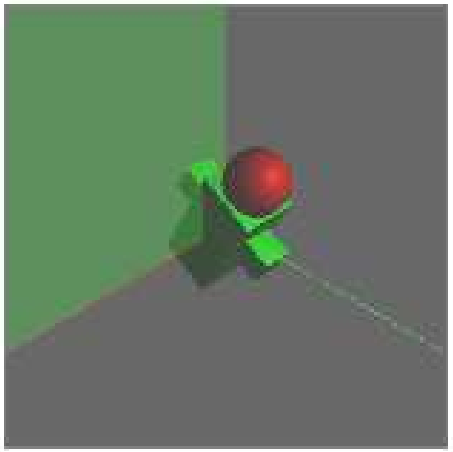
Transformações geométricas no espaço

Ângulos de Euler - Gimbal Lock

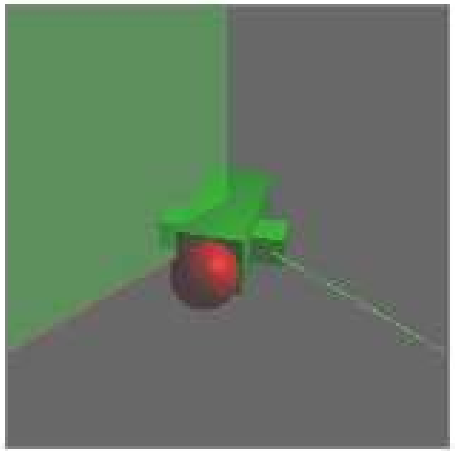
- Mesmo resultado obtido apenas com rotações no eixo x e y.



Original



$\Theta_x = -60^o$



$\Theta_x = -60^o$

$\Theta_y = 90^o$

Transformações geométricas no espaço

Ângulos de Euler

$$\mathbf{R}(\theta_x,\theta_y,\theta_z)=\begin{bmatrix} c_y c_z & c_y s_z & -s_y & 0 \\ s_x s_y c_z - c_x s_z & s_x s_y s_z + c_x c_z & s_x c_y & 0 \\ c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & c_x c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

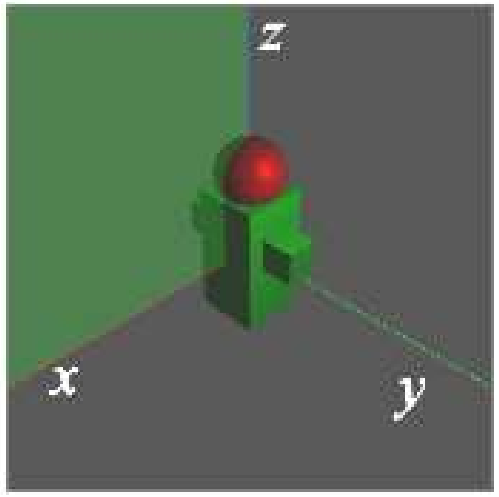
$$\mathbf{R}(\theta_x,90^\circ,\theta_z)=\begin{bmatrix} 0 & 0 & -1 & 0 \\ s_x c_z - c_x s_z & s_x s_z + c_x c_z & 0 & 0 \\ c_x c_z + s_x s_z & c_x s_z - s_x c_z & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ \sin(\theta_x-\theta_z) & \cos(\theta_x-\theta_z) & 0 & 0 \\ \cos(\theta_x-\theta_z) & \sin(\theta_x-\theta_z) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

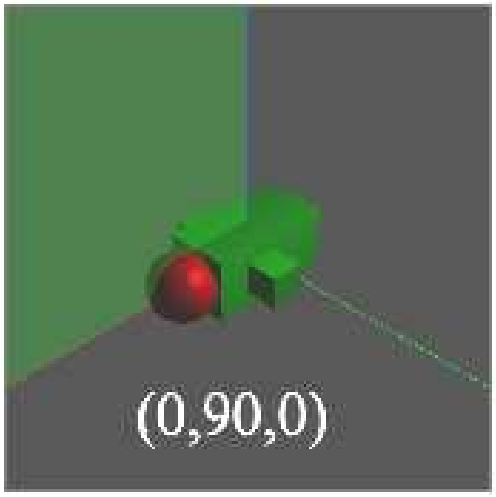
Apesar de especificarmos 2 parâmetros só temos 1 grau de liberdade

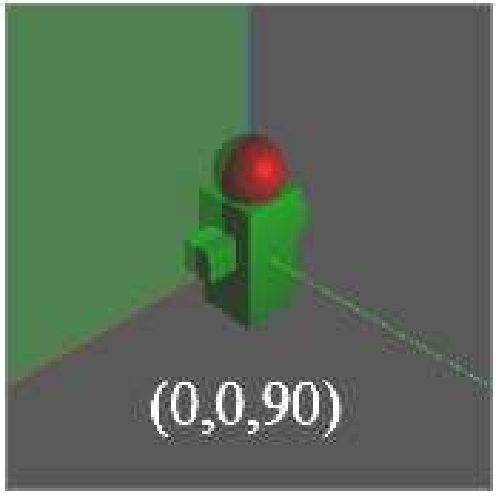
Transformações geométricas no espaço

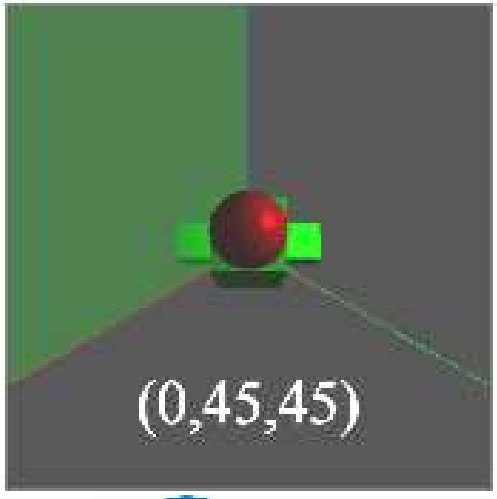
Ângulos de Euler - interpolação

- Interpolação entre as orientações (0,90,0) e (0,90,0).
- Note que em uma interpolação mais natural a cabeça não sairia tanto do plano *xz*.









Transformações geométricas no espaço

Outras formas de se expecificar orientações

- Rotações em torno de um eixo:

  - Euler provou em 1775 que dadas duas posições rotacionadas de um objeto, é sempre possível levar uma posição a outra através de uma ***rotação de um ângulo em torno de um eixo***.

  - Esta rotação tem a mesmo comportamento que a interpolação de duas posições através de um segmento de reta que os une.

  - Sai da primeira posição indo para a segunda sem oscilações.

Transformações geométricas no espaço

Outras formas de se expecificar orientações

- Quaternions

  - A especificação de orientações através de rotações em torno de um eixo não fornece uma álgebra simples para as diversas operações necessárias para animação.

  - Para isso existe uma estrutura matemática mais adequada denominada ***Quaternions***.

  - Um boa introdução aos *Quaternions* pode ser encontrada em: *http://www.gamasutra.com/features/19980703/quaternions_01.htm*

Transformações geométricas no espaço

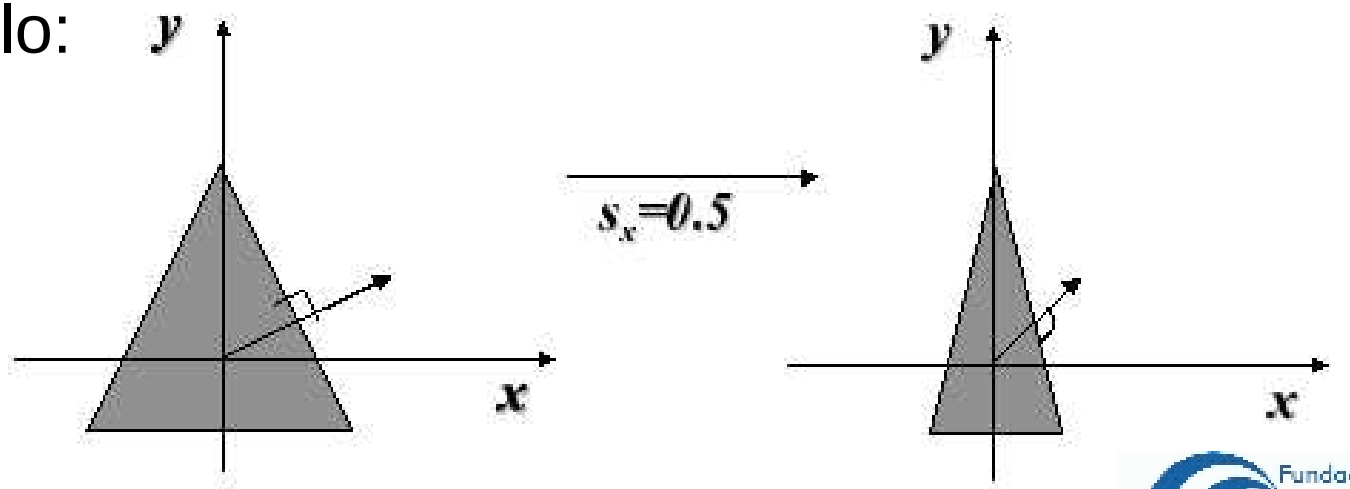
Transformações em OpenGL

- Translação
  - `glTranslate{fd}(TYPE x, TYPE y, TYPE z);`
- Rotação de angle graus em torno de um eixo *(x,y,z)*.
  - `glRotate{fd}(TYPE angle,TYPE x, TYPE y, TYPE z);`
- Escala
  - `glScale{fd}(TYPE sx, TYPE sy, TYPE sz);`

Transformações geométricas no espaço

Transformações em OpenGL

- Para transformarmos um certo objeto poligonal basta aplicar a matriz de transformação em cada um dos seus vértices.
- Por outro lado, no caso geral, ***as normais destes objetos não seguem a mesma transformação***.
- Exemplo:



Transformações geométricas no espaço

Transformações em OpenGL

- Considere o plano com equação

$$n^T p = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Se incluirmos a matriz identidade $I = M^{-1}M$ não alteramos a equação abaixo:

$$n^T p = \begin{bmatrix} a & b & c & d \end{bmatrix} M^{-1} M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformações geométricas no espaço

Transformações em OpenGL

- A equação do plano transformado *n'.p'=0* é

$$\begin{bmatrix} a & b & c & d \end{bmatrix} M^{-1} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = 0$$

- Logo, temos que a normal transformada é

$$n' = \begin{bmatrix} a' \\ b' \\ c' \\ d' \end{bmatrix} = M^{-T} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = M^{-T} n$$

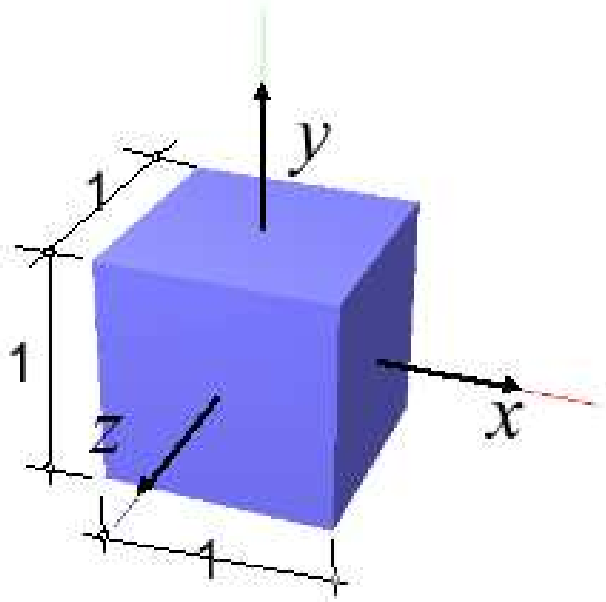
Transformações geométricas no espaço

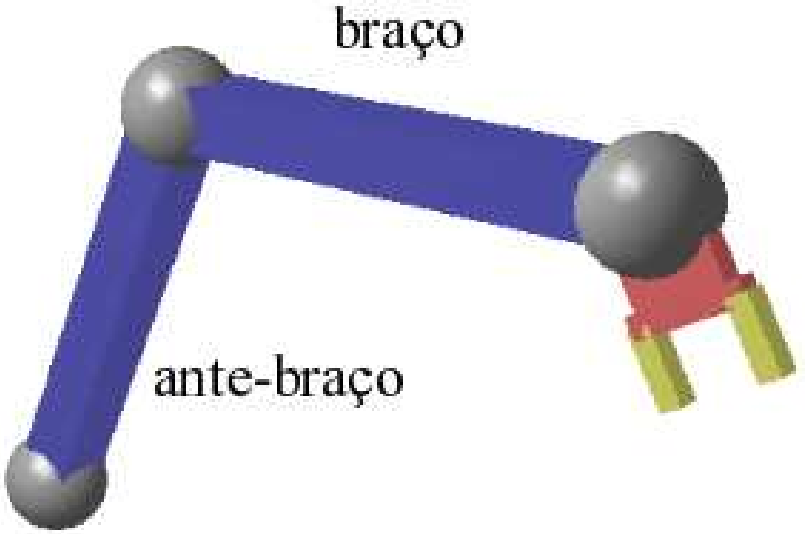
Transformações em OpenGL

- O processo de instanciação de objeto permite a ***especificação de modelos complexos através de modelos padrão simples e transformações geométricas.***
- É fundamental para a descrição de ***objetos compostos de várias partes***, principalmente quando há ***vínculo*** entre as mesmas.
- Devemos primeiramente esclarecer a questão de ordem e interpretação das transformações geométricas compostas.

Transformações geométricas no espaço

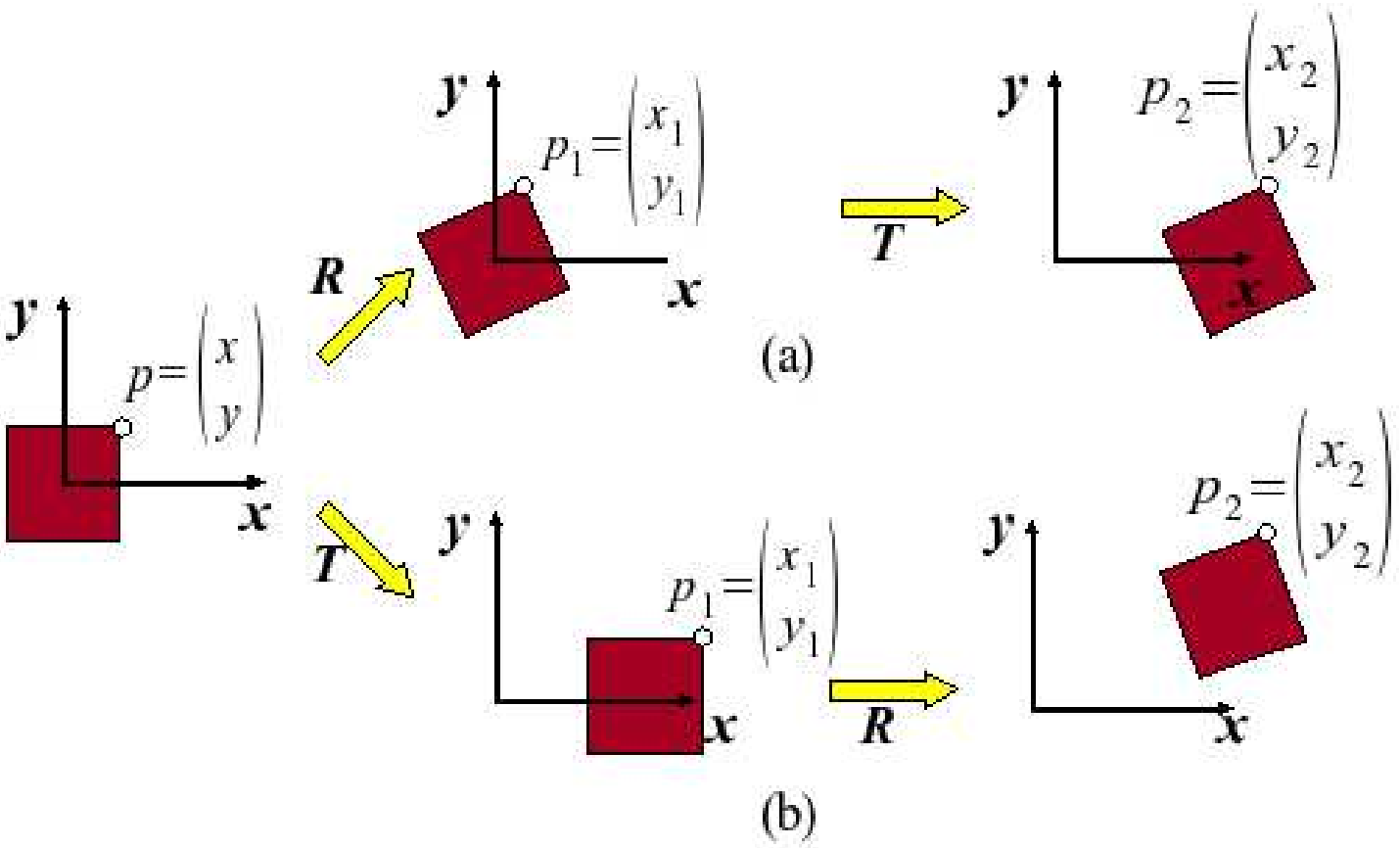
Transformações em OpenGL





Transformações geométricas no espaço

Transformações em OpenGL



Transformações geométricas no espaço

Transformações em OpenGL

- Às vezes é difícil especificar transformações geométricas nos casos em que a existe ***dependência*** entre a posição das partes de um objeto composto.
- Nestas situações é conveniente adotar uma outra interpretação geométrica para as transformações compostas.

Transformações geométricas no espaço

Transformações em OpenGL

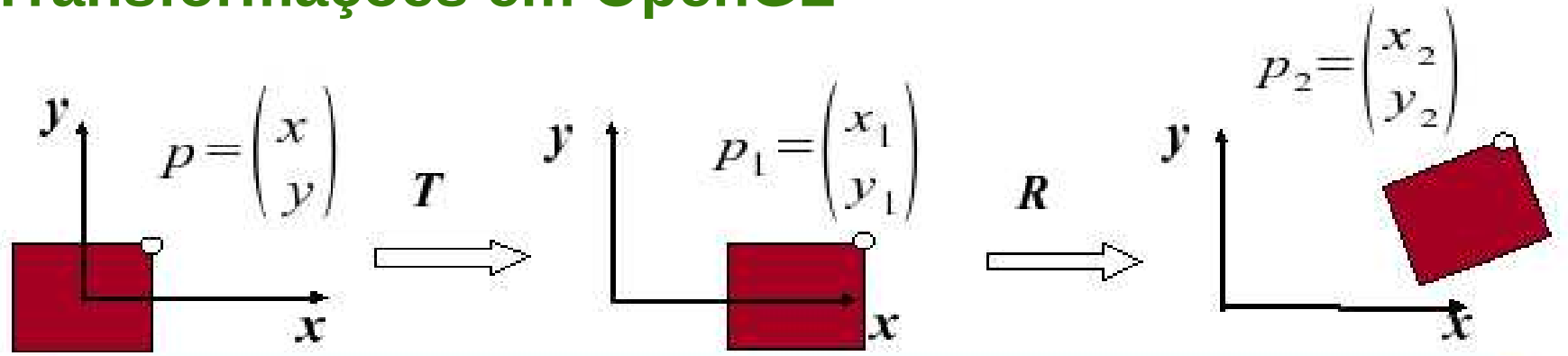
- Ao invés de considerarmos que as transformações ocorrem nos objetos, ***consideramos que elas ocorrem em um sistema de eixos locais*** que rodam e transladam.

- A idéia é que os eixos locais inicialmente coincidem com o sistema de referência global.

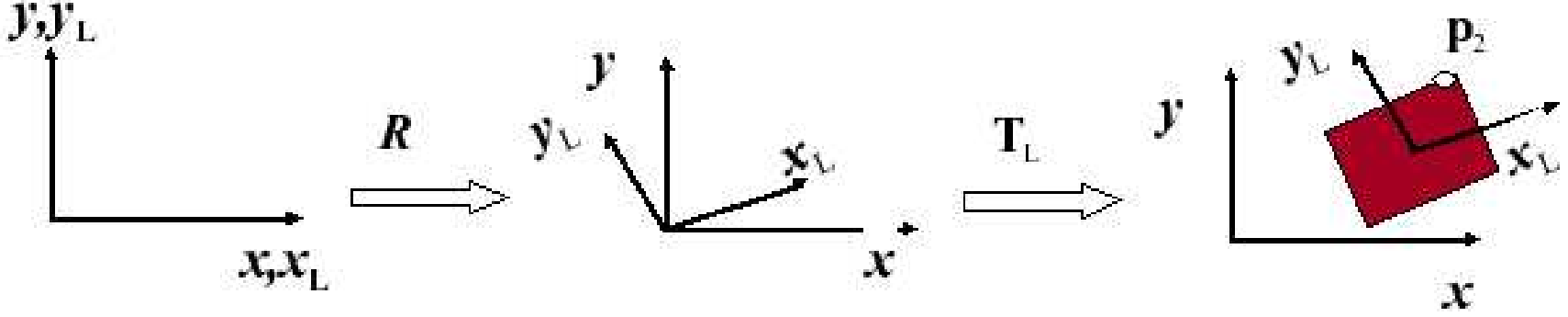
- A cada rotação e translação, ***um dado eixo local muda de posição e/ou orientação***.

Transformações geométricas no espaço

Transformações em OpenGL



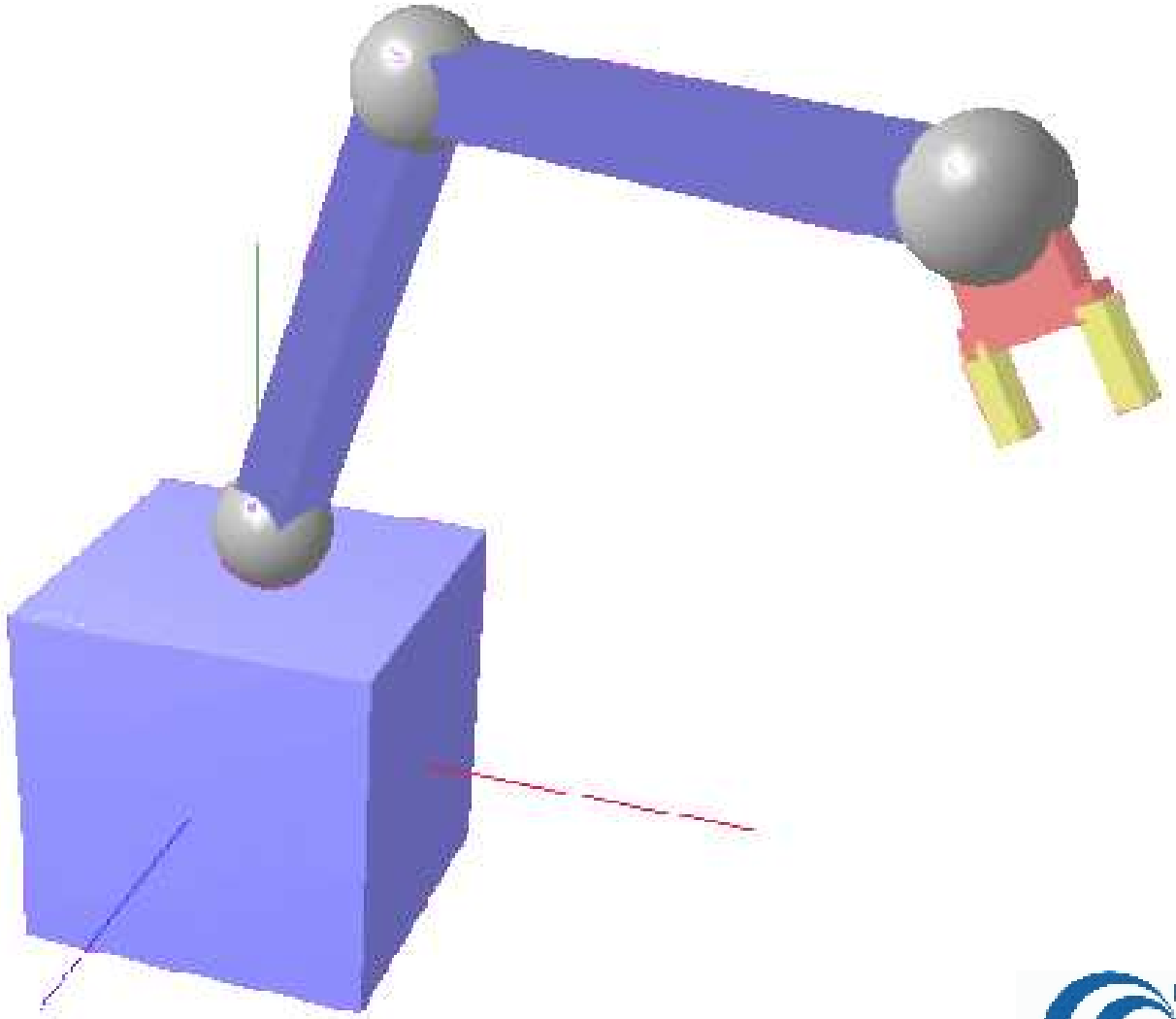
$$p_1 = T\,p \text{ e } p_2 = R\,p_1 \Rightarrow p_2 = R\,T\,p$$



$$p_2 = T_L R\,p,\ T_L = RTR^{-1} \Rightarrow p_2 = R\,T\,R^{-1}R\,p \quad \text{ou} \quad p_2 = R\,T\,p$$

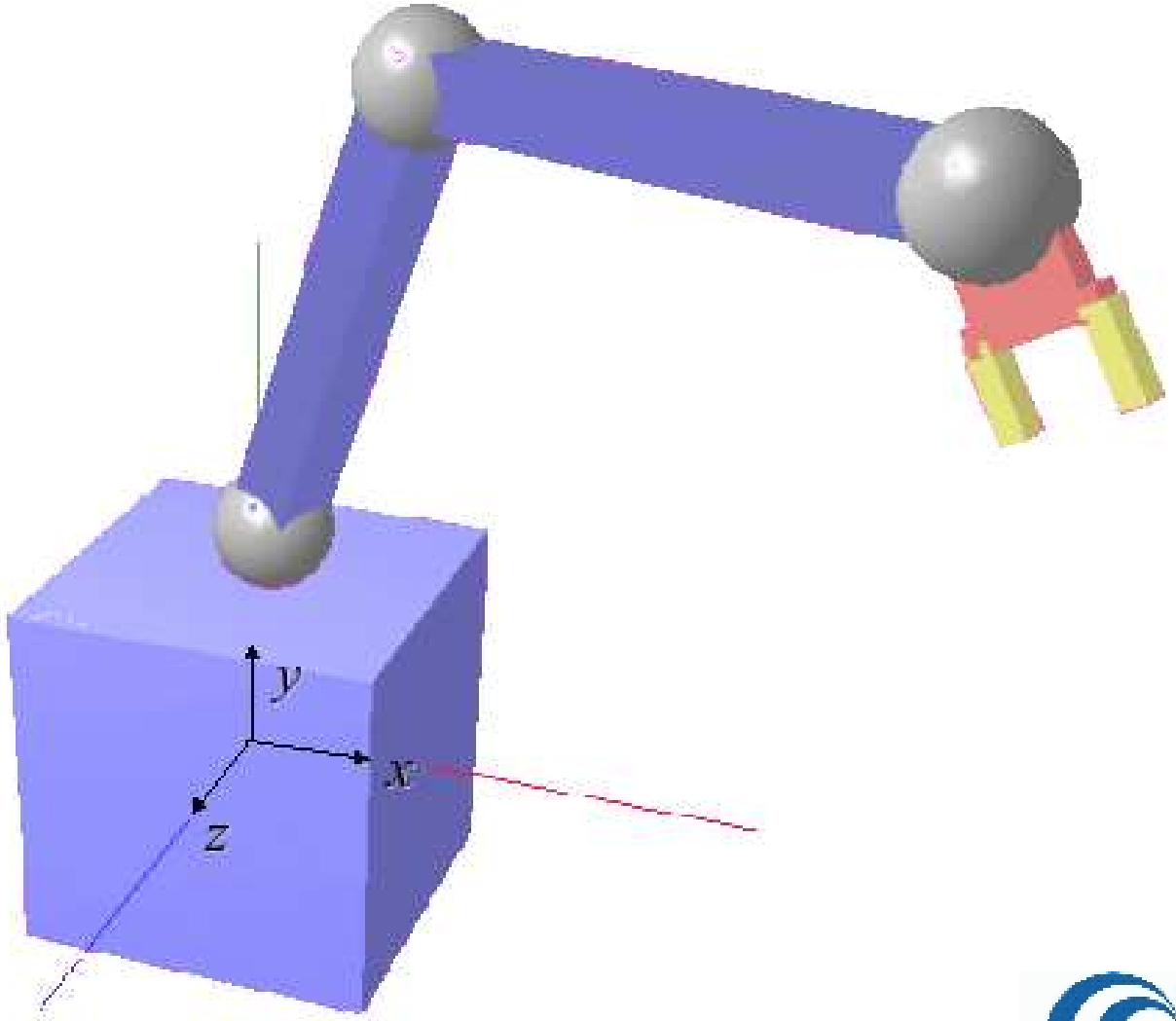
Transformações geométricas no espaço

Transformações em OpenGL



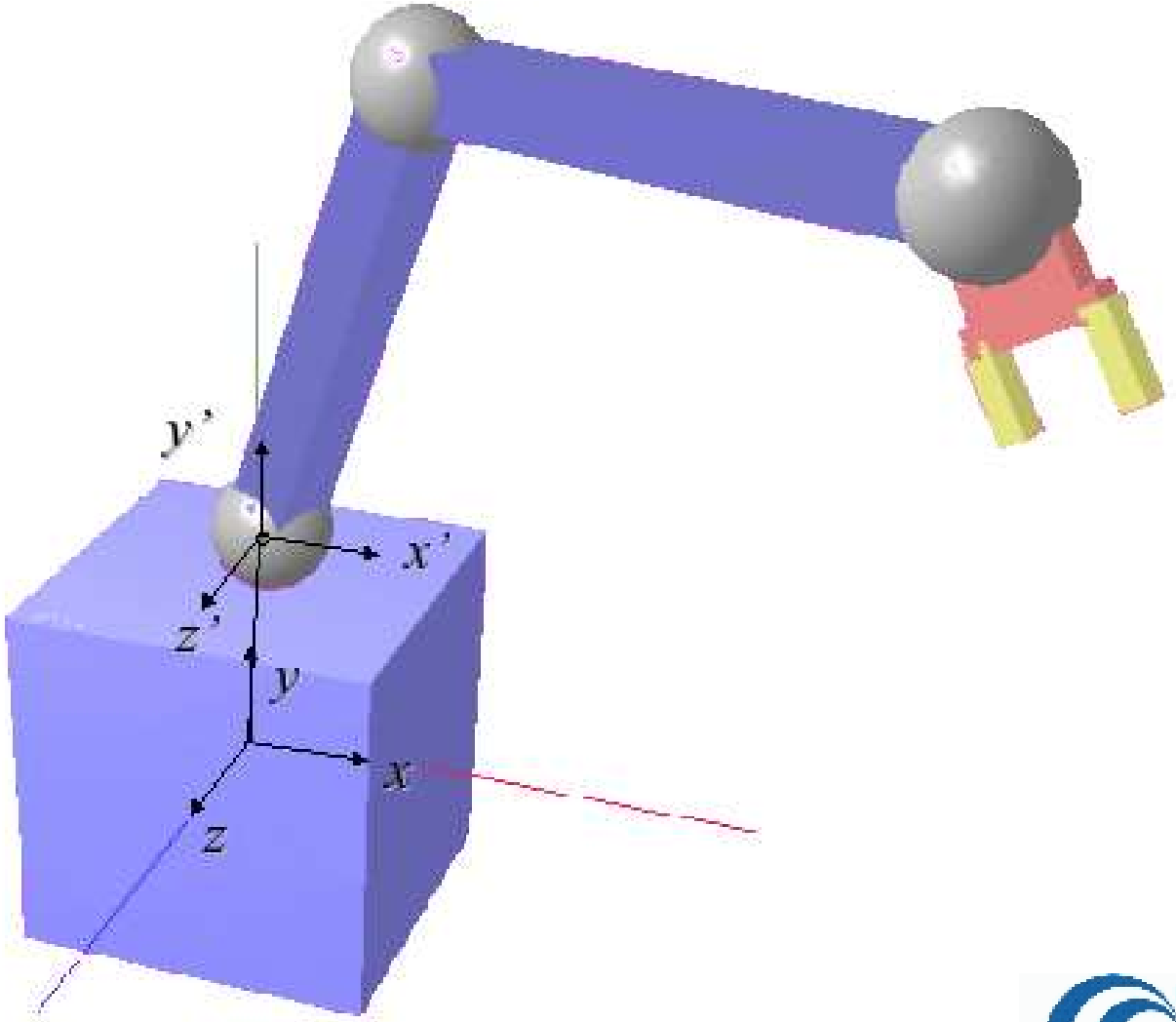
Transformações geométricas no espaço

Transformações em OpenGL



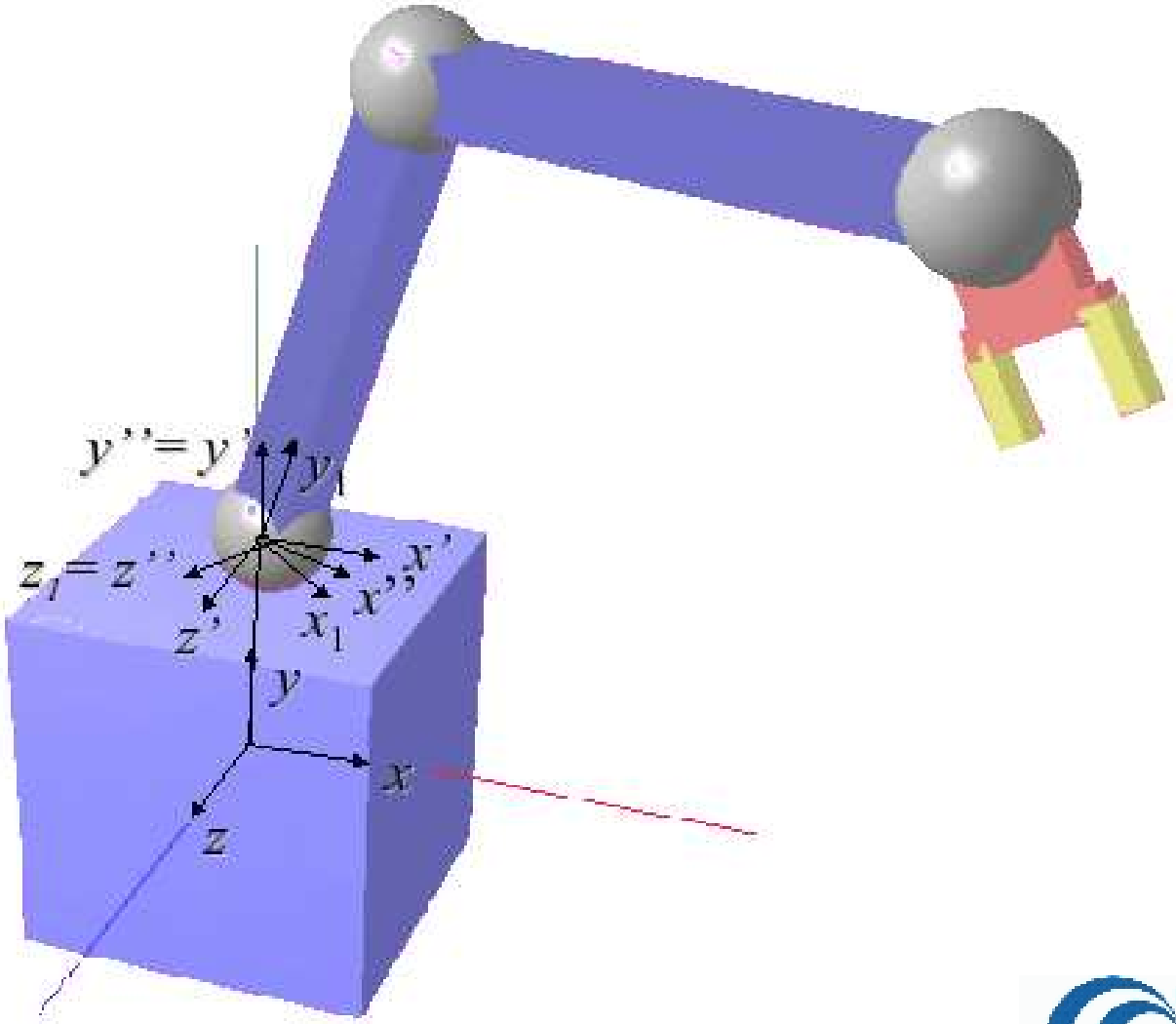
Transformações geométricas no espaço

Transformações em OpenGL



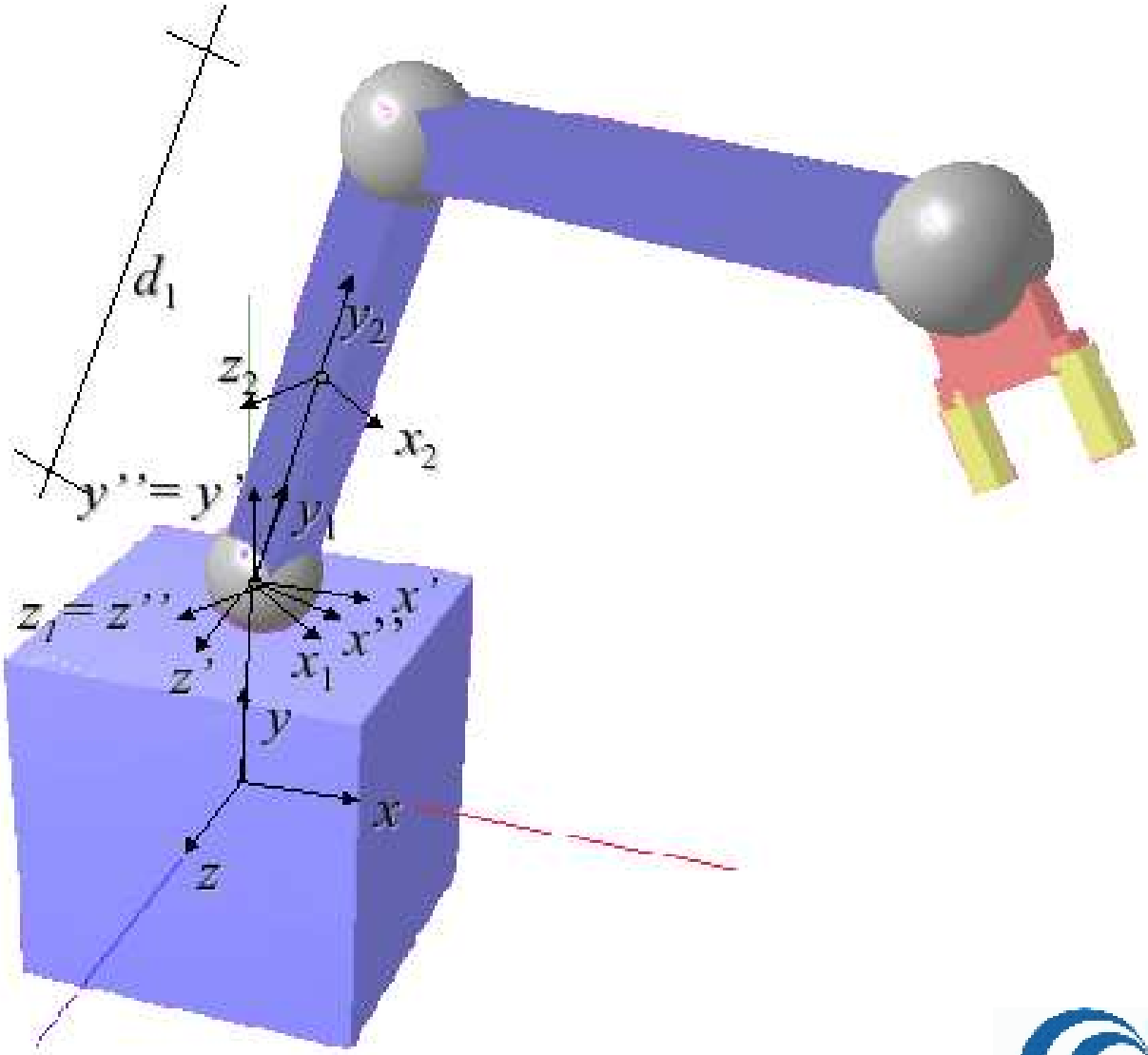
Transformações geométricas no espaço

Transformações em OpenGL



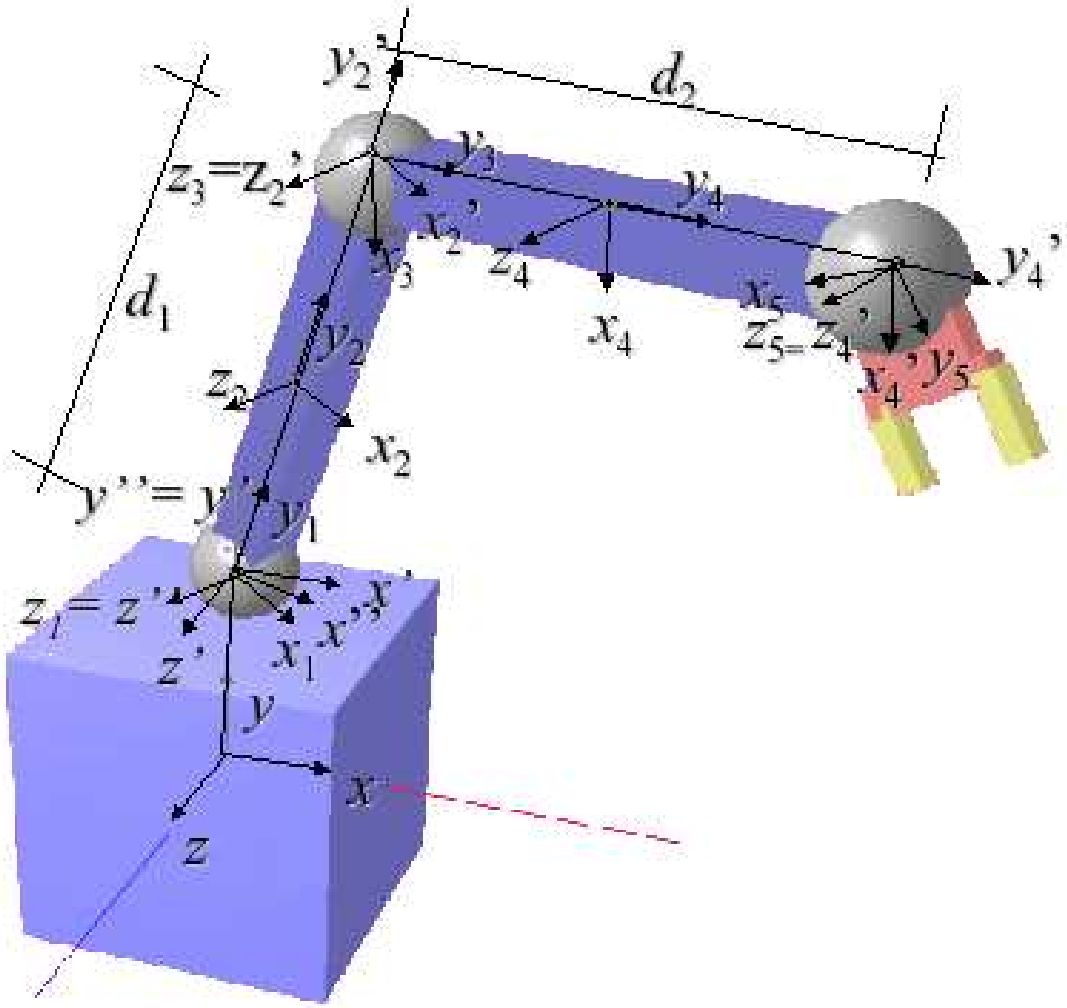
Transformações geométricas no espaço

Transformações em OpenGL



Transformações geométricas no espaço

Transformações em OpenGL




```
Desenha a base (em xyz);
Translada em y;
Roda em y';
Roda em z'';
Desenha o ombro (em x1y1z1)
Translada em y1 de d1/2;
Desenha o ante-braço (em x2y2z2);
Translada em y2 de d1/2;
Roda em z2'
Desenha o cotovelo (em x3y3z3);
Translada em y3 de d2/2;
Desenha o braço (em x4y4z4);
Translada em y4 de d2/2;
Roda em z4';
Desenha o pulso (em x5y5z5);
Translada em y5
Desenha a mão (em x6y6z6);
```

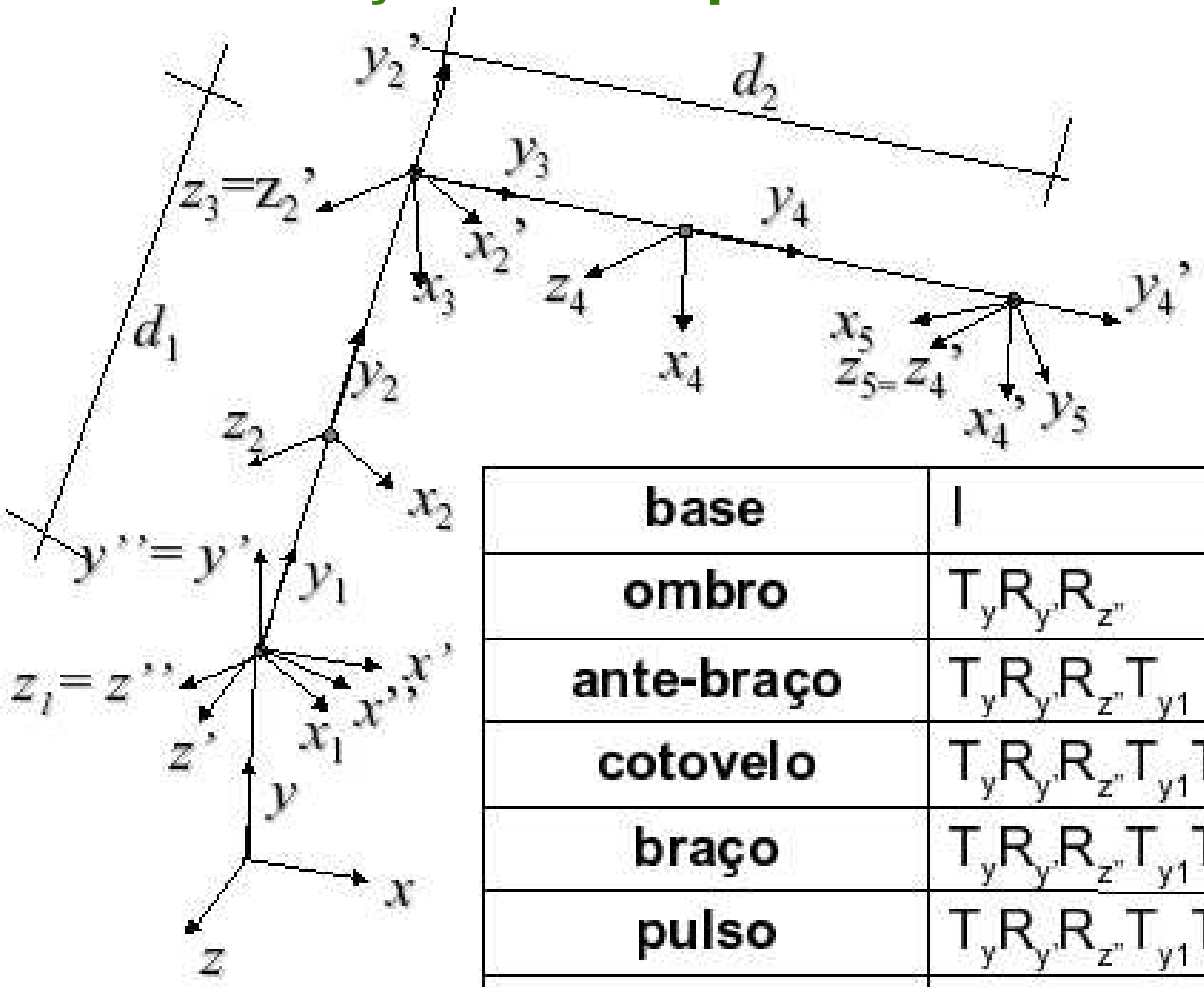
Transformações geométricas no espaço

Transformações em OpenGL

```
glMatrixMode(GL_MODELVIEW);   /* transformações do modelo          */
glLoadIdentity( );           /* carrega a identidade como corrente */
desenhaBase( );               /* em xyz                            */
glTranslatef(0,d0,0.);        /* translada em y                    */
glRotatef(angy0, 0.,1.,0.);   /* roda em y'                        */
glRotatef(angz0, 0.,0.,1.);   /* roda em z''                       */
desenhaOmbro( );              /* em x1y1z1                          */
glTranslatef(0.,d1/2,0.);     /* translada em y1                   */
desenhaAnteBraco( );          /* em x2y2z2                                 */
glTranslatef(0.,d1/2,0.);     /* translada em y2                   */
glRotatef(angz1, 0.,0.,1.);   /* roda em z2´                        */
desenhaCotovelo( );           /* em x3y3z3                                 */
glTranslatef(0.,d2/2,0.);     /* translada em y3                   */
desenhaBraco( );              /* em x4y4z4                                 */
glTranslatef(0.,d2/2,0.);     /* translada em y4                   */
glRotatef(rz5, 0.,0.,1.);     /* roda em z4'                       */
desenhaPulso( );              /* em x5y5z5                                 */
glTranslatef(0.,d3,0.0);      /* translada em y5                   */
desenhaMao( );                /* em x6y6z6                          */
```

Transformações geométricas no espaço

Transformações em OpenGL



base	I
ombro	$T_y R_{y'} R_{z''}$
ante-braço	$T_y R_{y'} R_{z''} T_{y1}$
cotovelo	$T_y R_{y'} R_{z''} T_{y1} T_{y2} R_{z2'}$
braço	$T_y R_{y'} R_{z''} T_{y1} T_{y2} R_{z2'} T_{y3}$
pulso	$T_y R_{y'} R_{z''} T_{y1} T_{y2} R_{z2'} T_{y3} T_{y4} R_{z4'}$
mão	$T_y R_{y'} R_{z''} T_{y1} T_{y2} R_{z2'} T_{y3} T_{y4} R_{z4'} T_{y5}$

Transformações geométricas no espaço

Transformações em OpenGL

- Sistemas como o OpenGL utilizam o conceito de ***matriz corrente***.

- Desta forma, apenas uma matriz é responsável por realizar as transformações geométricas.

- Esta matriz é denominada ***matriz de modelagem e visualização (model view matrix).***

Transformações geométricas no espaço

Transformações em OpenGL

- Quando fornecemos uma nova matriz *M*, ela é multiplicada pela esquerda pela matriz corrente *C*. Isto é $C_{nova} = CM$.

- Geometricamente isto significa que ***a transformação descrita por M ocorrerá primeiro que a transformação dada por C.***

- Isto é bastante conveniente para o esquema de interpretação do processo de instanciação baseado em ***eixos locais***.

Transformações geométricas no espaço

Transformações em OpenGL

- Existem situações em que o processo de instanciação não é descrito por uma seqüência de transformações com estrutura linear.

- É comum por exemplo, encontrarmos objetos que são descritos por ***seqüências de transformação estruturadas em forma de árvore***.

- Nestes casos é definida uma ***hierarquia*** sobre o conjunto de transformações e partes do objeto que especificam o modelo.

Transformações geométricas no espaço

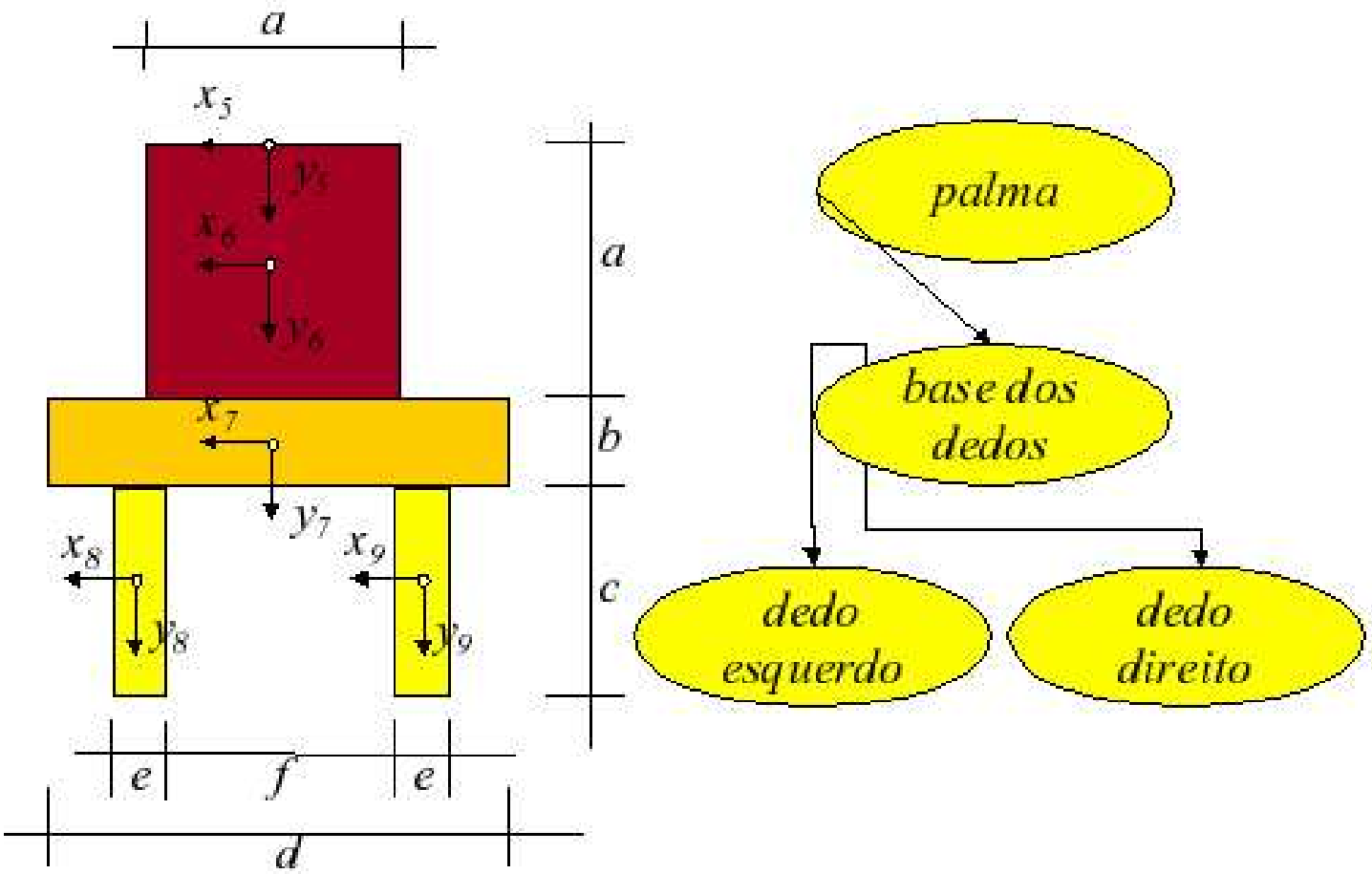
Transformações em OpenGL

- Nestes casos, ao término do percorrimento de um dos ramos, é necessário ***recuperar a matriz*** do nó quando primeiro chegamos a ele.

- Por exemplo, podemos definir os dedos direito e esquerdo da mão de um robô a partir da base da mão.

Transformações geométricas no espaço

Transformações em OpenGL



Transformações geométricas no espaço

Transformações em OpenGL

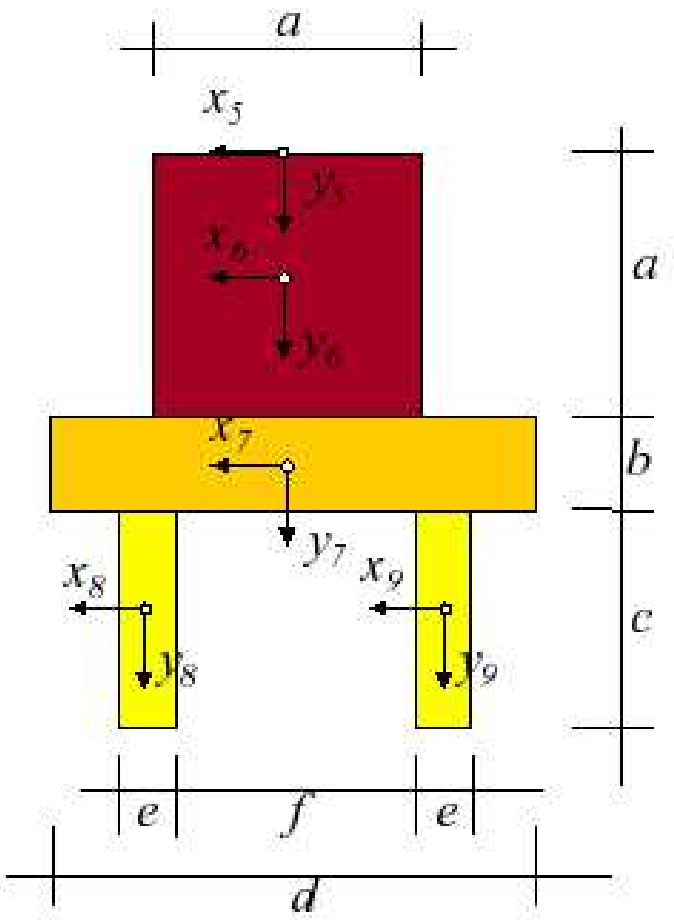
- Sistemas como o OpenGL implementam uma ***estrutura de pilha para matrizes de transformação.***

- Desta forma, é possível percorrer a árvore saltando e recuperando as matrizes dos nós pai através de instruções ***pop*** e ***push***.

- Com o mecanismo de pilha, podemos garantir que uma função retorna sem alterar o estado corrente das transformações.

Transformações geométricas no espaço

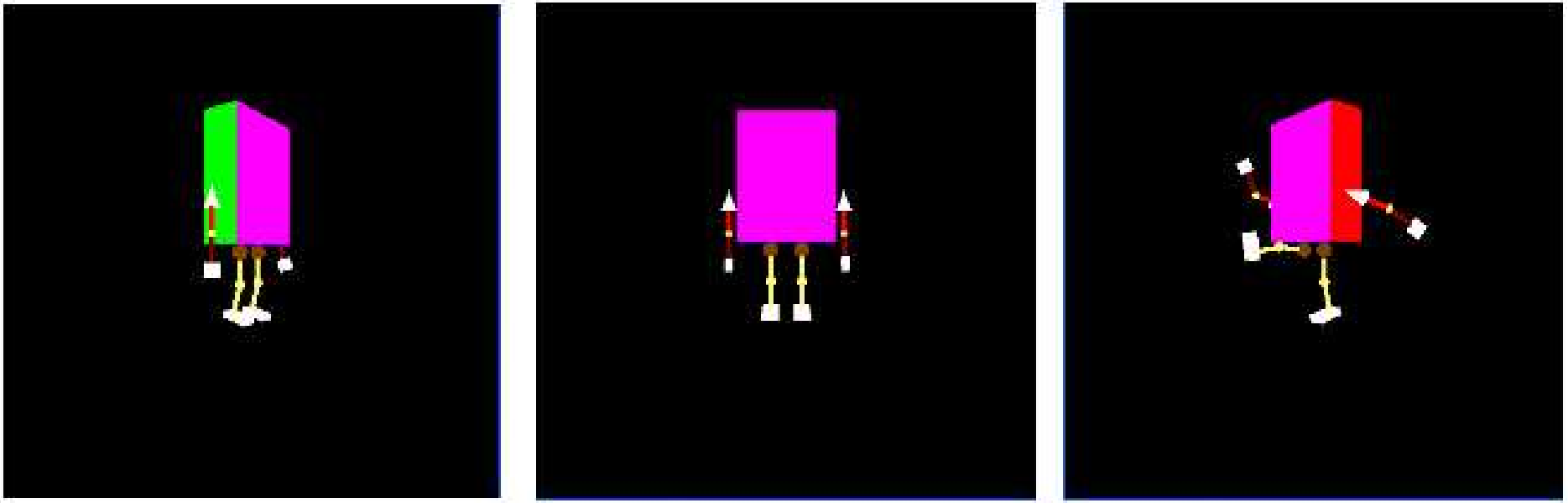
Transformações em OpenGL




```
void desenhaDedos(float b,float c, float f, float f )

  /* dedo esquerdo */

  glPushMatrix();     /* Salva matriz corrente C0 */
    glTranslatef((f+e)/2,(b+c)/2,0.);  /* C=CTesq  */
    glScalef(e,c,e);                   /* C=CS     */
    glutSolidCube(1.0);
  glPopMatrix();       /* Recupera da pilha C=C0  */

  /* dedo direito */

  glPushMatrix();     /* Salva matriz corrente C0 */
    glTranslatef((f+e)/2,(b+c)/2,0.);  /* C=CTdir  */
    glScalef(e,c,e);                   /* C=CS     */
    glutSolidCube(1.0);
  glPopMatrix();       /* Recupera da pilha C=C0  */

}
```

Transformações geométricas no espaço

Transformações em OpenGL - exemplo



Aula 9

Professores:

Anselmo Montenegro
Esteban Clua

Conteúdo:

- Transformações geométricas no espaço