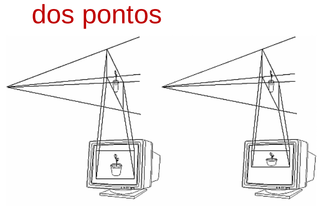
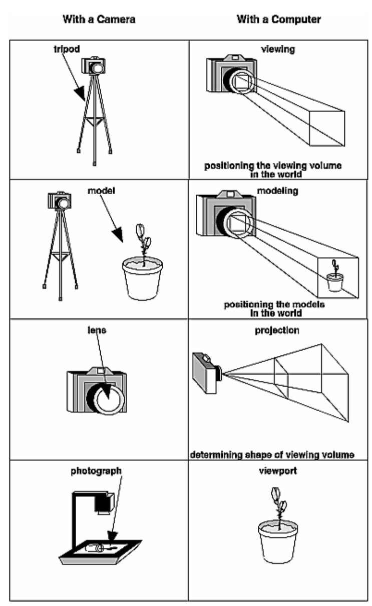
Transformações Geométricas

Técnicas para alterar as coordenadas geométricas dos pontos de um modelo geométrico

Transformações Geométricas

- Posicionar os blocos constituintes de uma cena
  - Alterar as coordenadas dos pontos
- Projetar a cena sobre o plano de imagem
  - Alterar as coordenadas dos pontos
- Enquadrar a cena na janela de exibição
  - Alterar as coordenadas dos pontos

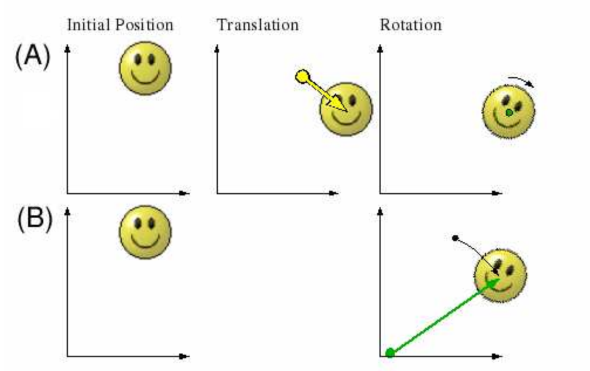




Motivação

Tarefa 1

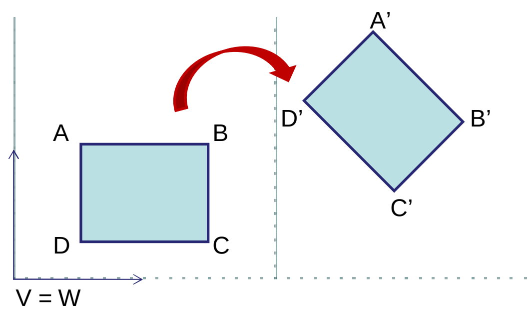
- Como podemos deslocar, rodar ou espelhar uma figura geométrica em um processador numérico, sem deformá-la?



Transformações Rígidas

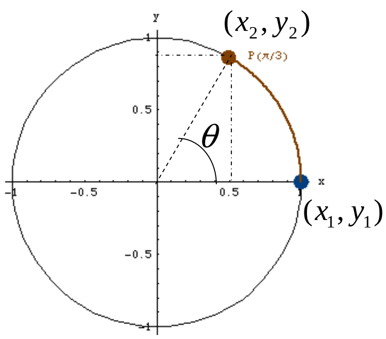
As formas das figuras não são alteradas:

- ➢Rotação
- ➢Reflexão
- ➢Deslocamento



f: P → P'

Rotação 2D



Ponto Inicial:

$$(x_1, y_1) = (R\cos\phi, Rsen\phi)$$

Ponto Final:

$$(x_2, y_2) = (R\cos(\phi+\theta), Rsen(\phi+\theta))$$

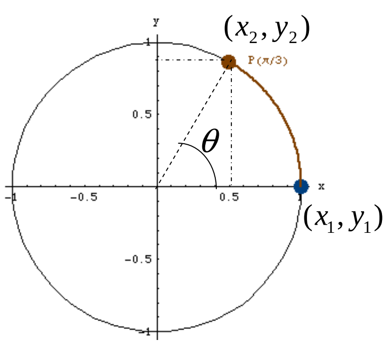
$$x_2 = R\cos(\phi+\theta) = R\cos\phi\cos\theta - Rsen\,\phi sen\,\theta = x_1\cos\theta - y_1 sen\,\theta$$

$$y_2 = Rsen\,(\phi+\theta) = Rsen\,\phi\cos\theta + R\cos\phi sen\,\theta = x_1 sen\,\theta + y_1\cos\theta$$

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -sen\,\theta \\ sen\,\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

Em linguagem matemática: Rotação pode ser traduzida em uma transformação linear!!!!

Rotação 2D



Ponto Inicial:

$$(x_1, y_1) = (R\cos\phi, Rsen\phi)$$

Quais são os pontos invariantes nesta formulação?

Ponto Final:

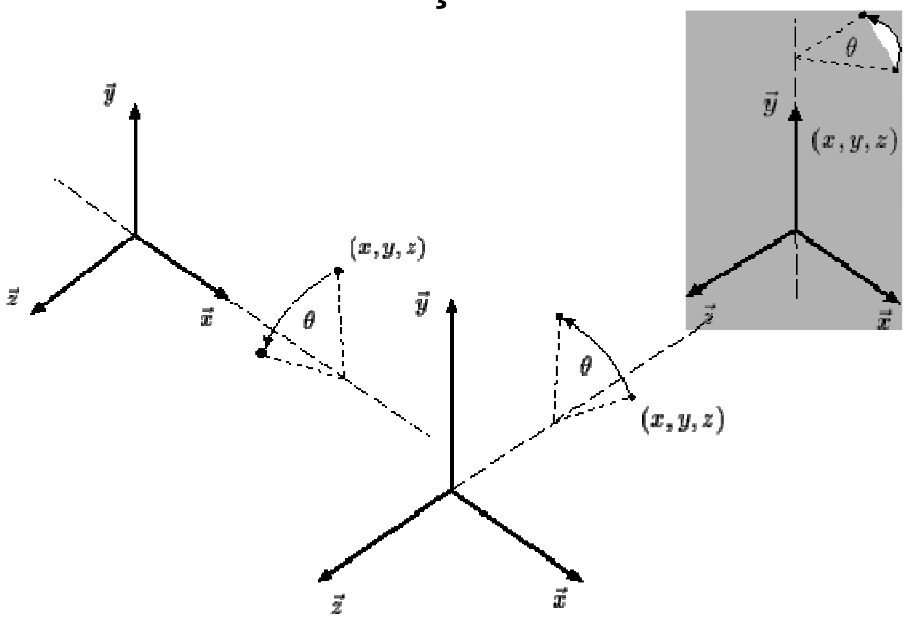
$$(x_2, y_2) = (R\cos(\phi+\theta), Rsen(\phi+\theta))$$

$$x_2 = R\cos(\phi+\theta) = R\cos\phi\cos\theta - Rsen\,\phi sen\,\theta = x_1\cos\theta - y_1 sen\,\theta$$

$$y_2 = Rsen\,(\phi+\theta) = Rsen\,\phi\cos\theta + R\cos\phi sen\,\theta = x_1 sen\,\theta + y_1\cos\theta$$

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -sen\,\theta \\ sen\,\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

Rotação 3D



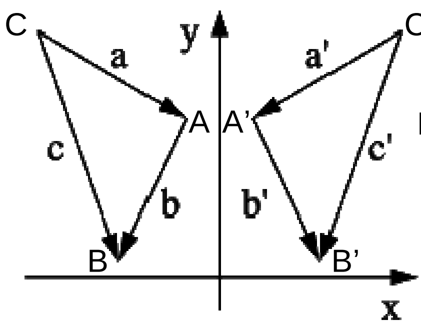
Notação Matricial

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\text{sen}\,\theta & 0 \\ \text{sen}\,\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\text{sen}\,\theta \\ 0 & \text{sen}\,\theta & \cos\theta \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \text{sen}\,\theta \\ 0 & 1 & 0 \\ -\text{sen}\,\theta & 0 & \cos\theta \end{pmatrix}$$

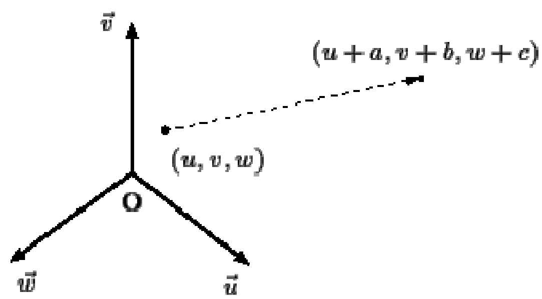
Reflexão 2D



$$\text{Reflexao}_y = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Nesta formulação, quais são os pontos invariantes?

Translação



Há pontos invariantes neste tipo de ação?

Notação Matricial

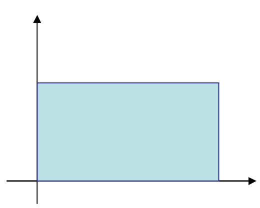
$$Tr = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

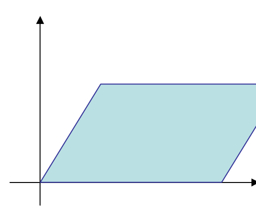
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Motivação

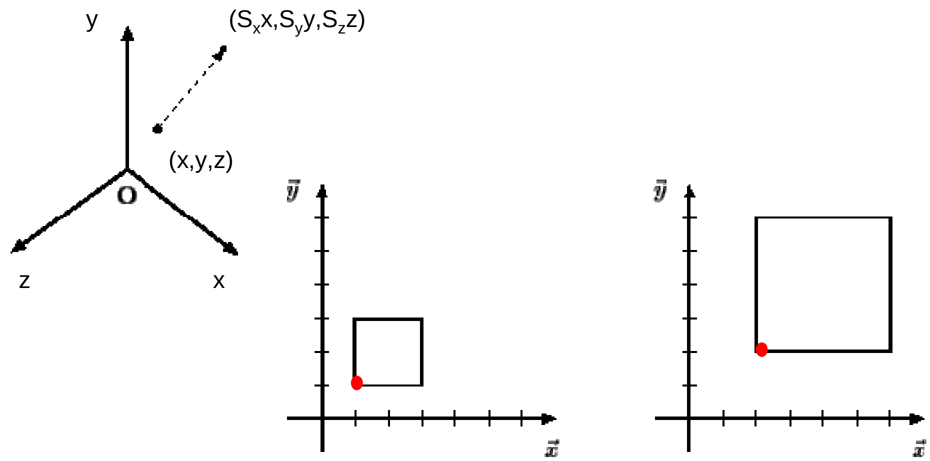
Tarefa 2

- E se quisermos redimensionar ou cisalhar uma figura geométrica em um processador numérico?





Mudança de Escala



Notação Matricial

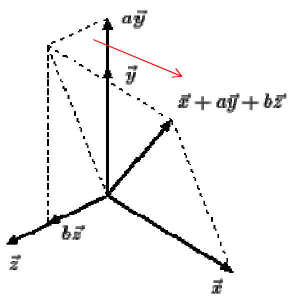
$$S = \begin{pmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & S \end{pmatrix} \qquad S = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{pmatrix}$$

Uniforme

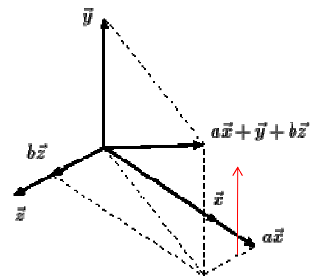
Não-Uniforme

Nesta formulação, quais são os pontos invariantes?

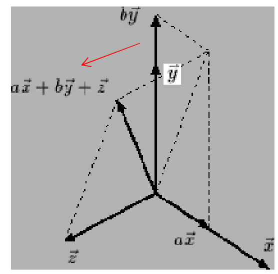
Cisalhamento (*Shearing*)



x em relação a y e z



y em relação a x e z



z em relação a x e y

Notação Matricial

$$Sh_z = \begin{pmatrix} 1 & 0 & \Delta x/\Delta z \\ 0 & 1 & \Delta y/\Delta z \\ 0 & 0 & 1 \end{pmatrix}$$

$$Sh_x = \begin{pmatrix} 1 & 0 & 0 \\ \Delta y/\Delta x & 1 & 0 \\ \Delta z/\Delta x & 0 & 1 \end{pmatrix}$$

$$Sh_y = \begin{pmatrix} 1 & \Delta x/\Delta y & 0 \\ 0 & 1 & 0 \\ 0 & \Delta z/\Delta y & 1 \end{pmatrix}$$

Nesta formulação, quais são os pontos invariantes?

Transformações Lineares

Funções *f* entre dois espaços vetoriais V e W que preservam as operações de adição vetorial e multiplicação por escalar do conjunto K

f: V $\rightarrow$ W

$$x_2 = a_{00}\, x_1 + a_{01}\, y_1 + a_{02}\, z_1 + a_{03}$$
$$y_2 = a_{10}\, x_1 + a_{11}\, y_1 + a_{12}\, z_1 + a_{13}$$
$$z_2 = a_{20}\, x_1 + a_{21}\, y_1 + a_{22}\, z_1 + a_{23}$$

Preserva paralelismo!

Motivação

Tarefa 3

- Há alguma ferramenta que consegue descrever todas as transformações apresentadas anteriormente, deslocamento, rotação, reflexão, mudança de escala e cisalhamento, de uma forma algébrica única?

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

reflexão

$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

re-dimensionamento

$$\begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

deslocamento

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -sen\theta \\ 0 & sen\theta & \cos\theta \end{pmatrix}$$

rotação

$$\begin{pmatrix} 1 & 0 & sh_{xz} \\ 0 & 1 & sh_{yz} \\ 0 & 0 & 1 \end{pmatrix}$$

cisalhamento

Transformações Afins

Afins = lineares + deslocamento

$\Updownarrow$

Aditividade: f(x+y) = f(x) + f(y)
Homogeneidade: f(ax) = af(x)

$\downarrow$

Reflexões
Rotação
Mudança de escala
Cisalhamento

Transformações rígidas

Concatenação Matricial

Transformação Linear

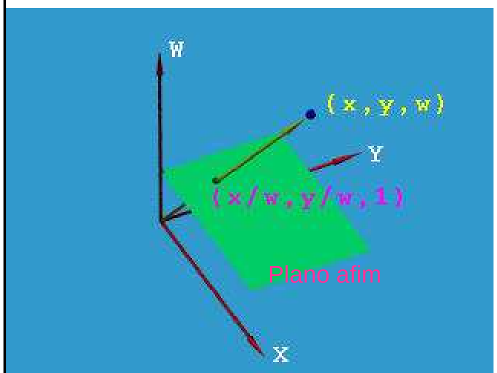
Translação

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Coordenadas Homogêneas = pontos R^n representados por (n+1) escalares

Coordenadas Homogêneas

(x,y,z,w)



Ponto

(x/w,y/w,z/w,1), ou seja, projeção de (x,y,z,w), sobre o plano w=1 com centro de projeção na origem.

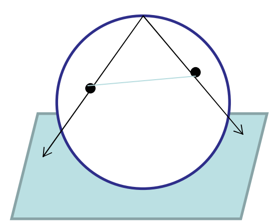
Vetor

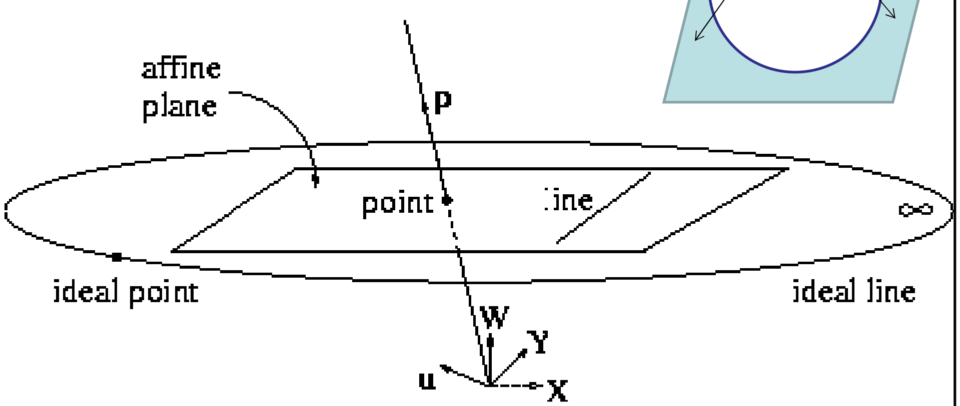
"diferença" de 2 pontos homogeneizados → Quarta coordenada(w) é nula

$\lim_{w \to 0} (x/w, y/w, z/w, w/w) = (\infty,\infty,\infty,1)$

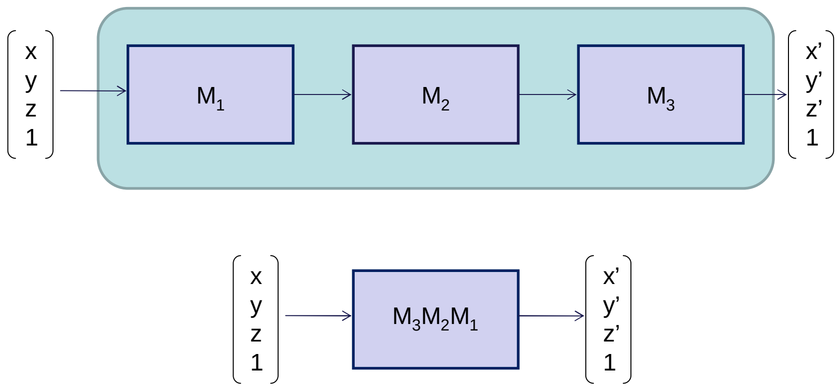
na direção (x,y,z)

Plano Afim





Transformações Afins
Notação Matricial

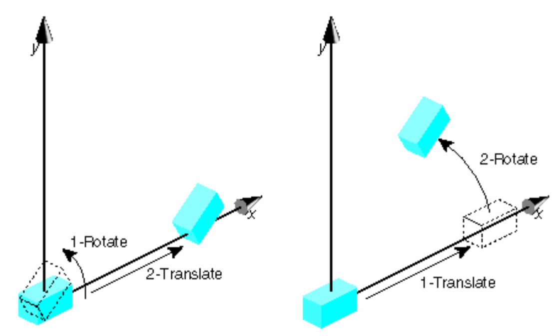


Composição de matrizes

Propriedades de Matrizes

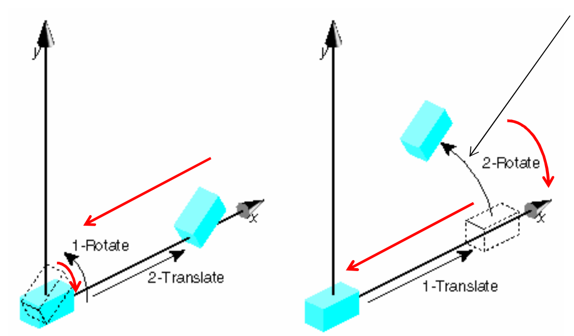
- Multiplicação:
  - Nem sempre vale a comutatividade
  - Associatividade ((AB)C = A(BC))
  - Nem sempre vale o cancelamento (AC=BC não implica em A=B)
- Transposta: as linhas se transformam em colunas
  - $(A^t)^t = A$
  - $(A+B)^t = A^t + B^t$
  - $(AB)^t = B^tA^t$
- Matrizes com propriedades especiais
  - Matriz identidade I: IA= A
  - Matriz inversa: $AA^{-1} = A^{-1}A = I$
  - Matriz ortonormal: $A^{-1} = A^t$

Dois Exemplos



$P' = T.(R.P) = (T.R).P$ $P' = R.(T.P) = (R.T).P$

Transformações Inversas



$P' = T.(R.P) = (T.R).P$
$(T.R)^{-1}\, P' = P$
$R^{-1}\, T^{-1}\, P' = P$

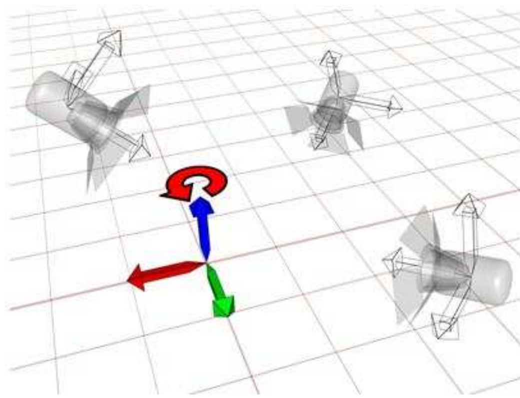
$P' = R.(T.P) = (R.T).P$
$(R.T)^{-1}\, P' = P$
$T^{-1}\, R^{-1}\, P' = P$

Matrizes inversas

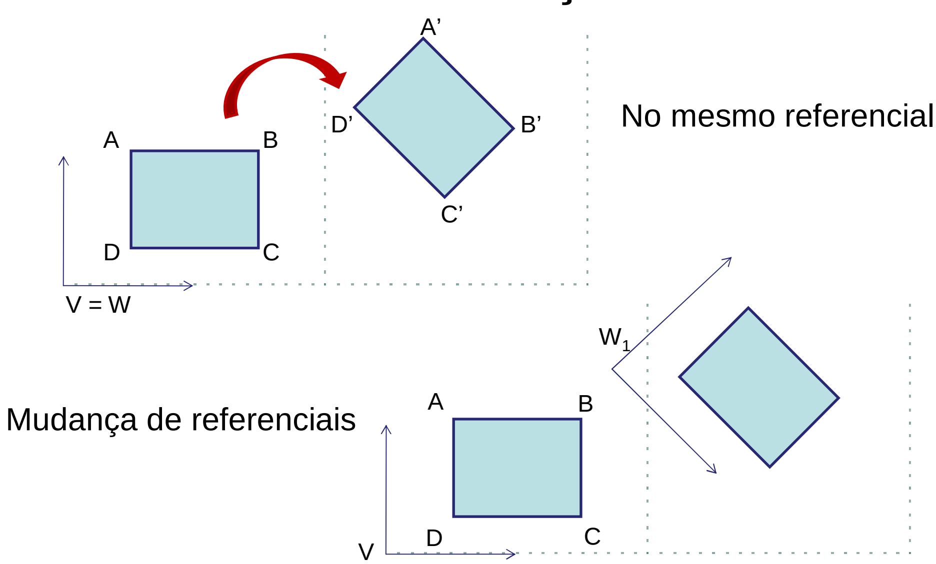
Motivação

Tarefa 4

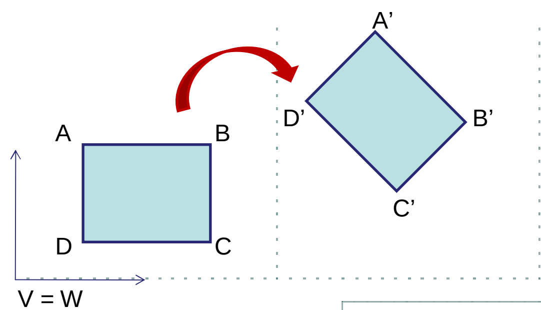
- Se tivermos objetos representados em distintos referenciais, como poderemos posicioná-los relativamente em um mesmo espaço?



Dois Paradigmas de Transformações

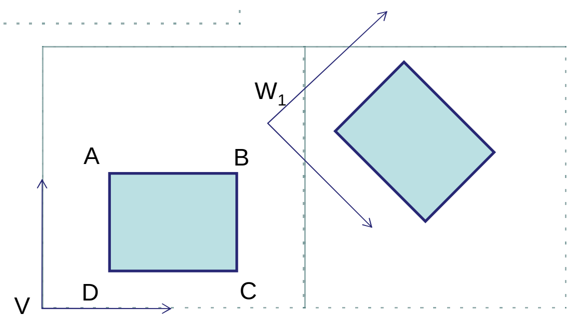


Notação Matricial



$$P' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} (TP)$$

$$P' = (T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}) P$$



Mudança de Referenciais

Reposicionamento de Objetos

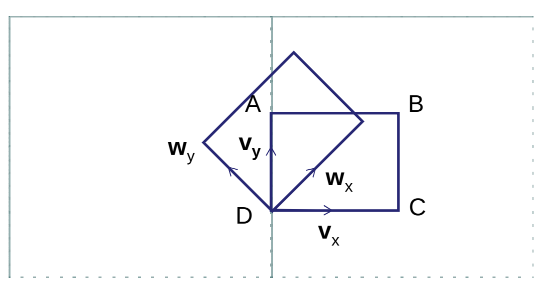
P' =TP

T = P'P^{-1},

se P for inversível

⇕

"Transformação de Base"

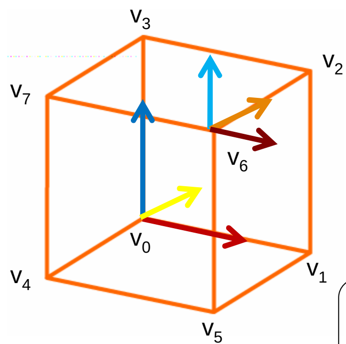


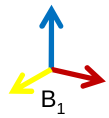
$$\begin{bmatrix} \mathbf{w}_x & \mathbf{w}_y \end{bmatrix} = T \begin{bmatrix} \mathbf{v}_x & \mathbf{v}_y \end{bmatrix}$$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{w}_{x,x} & \mathbf{w}_{y,x} \\ \mathbf{w}_{x,y} & \mathbf{w}_{y,y} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{x,x} & \mathbf{v}_{y,x} \\ \mathbf{v}_{x,y} & \mathbf{v}_{y,y} \end{pmatrix}^{-1}$$

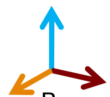
$$T = B_W B_V^{-1}$$

Exemplo





$$P = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

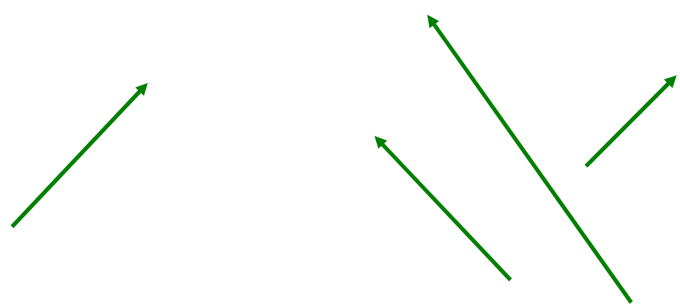
$$P' = \begin{pmatrix} -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$P' = B_1 B_2^{-1} P$$

Motivação

Tarefa 5

- Vimos como os pontos se transformam em um espaço ou entre os espaços.E os vetores seguem as mesmas regras de transformação?



Transformações de Vetores

Pontos

$$\begin{pmatrix} A_x & B_x & C_x \\ A_y & B_y & C_y \end{pmatrix} \xrightarrow{f} \begin{pmatrix} A'_x & B'_x & C'_x \\ A'_y & B'_y & C'_y \end{pmatrix}$$

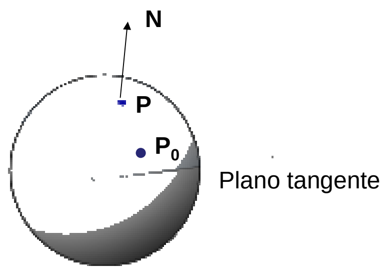
Vetores: diferença de 2 pontos

$$\begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \end{pmatrix} \xrightarrow{f} \begin{pmatrix} a'_x & b'_x & c'_x \\ a'_y & b'_y & c'_y \end{pmatrix}$$

$$MP_2 - MP_1 = M(P_2 - P_1) = M\overrightarrow{P_2P_1}$$

Vetores Normais

Em superfície suave, pode-se definir localmente um plano tangente a cada ponto. O vetor normal deste plano coincide com o vetor normal da superfície neste ponto **P**.



Antes da transformação T:

$N(P_0-P) = 0$

Após a transformação T:

$N'(TP_0-TP) = 0$

$N(P_0-P) = N'T(P_0-P)$

$N = N'T$

$NT^{-1} = N'$

Se for deslocamento d:

$N'(P_0+d - (P+d)) = N(P_0-P) = 0$

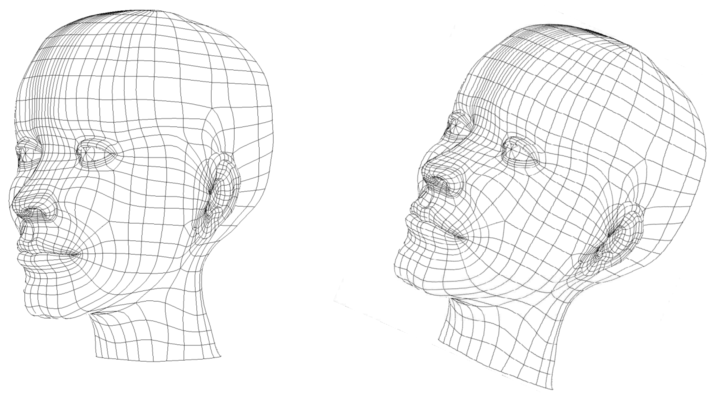
$N' = N$

N e N' em vetor-linha!

Motivação

Tarefa 6

- E como podemos reposicionar or redimensionar figuras geométricas representadas por funções analíticas?



Funções Analíticas

$$\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Equação implícita de uma circunferência

$$x^2 + y^2 = R^2 \quad \Longrightarrow \quad (a_{00}x + a_{01}y)^2 + (a_{10}x + a_{11}y)^2 = R^2$$

Equação paramétrica de uma circunferência

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} R\cos\theta \\ Rsen\theta \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} a_{00}x + a_{01}y \\ a_{10}x + a_{11}y \end{pmatrix} = ???$$

Funções Analíticas

Equação de um segmento

$P(t) = (1\text{-}t)P_1 + t\,P_2$

Transformação linear:

$(1\text{-}t)\ T_L P_1 + t T_L P_2 = T_L[(1\text{-}t)P_1 + tP_2] = T_L P(t)$

Translação:

$(1\text{-}t)\ (P_1+d) + t(P_2+d) = (1\text{-}t)P_1 + tP_2 + (1+t\text{-}t)d = P(t)+d$

Equação de uma curva de Bézier

$P(t) = \Sigma\ B_{n,i}(t)\ P_i$

Transformação linear:

$\Sigma\ B_{n,i}(t)\ (T_L\ P_i) = T_L(\Sigma\ B_{n,i}(t)\ P_i) = T_L P(t)$

Translação:

$\Sigma\ B_{n,i}(t)\ (P_i + d) = \Sigma\ B_{n,i}(t)\ P_i + \Sigma\ B_{n,i}(t)\ d = P(t) + d$

OpenGL

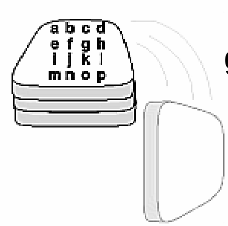
Pilha de Matrizes de Transformação

MODELVIEW
PROJECTION

$$\begin{bmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{bmatrix}$$

glTranslate
glRotate
glScale
glMultMatrix
glFrustum
glOrtho
glViewport

glPushMatrix



glPopMatrix

OpenGL

```
void init(void) {
    /* Enable a single OpenGL light. */
    glLightfv(GL_LIGHT0, GL_DIFFUSE,
    light_diffuse);
    glLightfv(GL_LIGHT0,
    GL_POSITION, light_position);
    glEnable(GL_LIGHT0);
    glEnable(GL_LIGHTING);
    /* Use depth buffering for hidden
    surface elimination. */
    glEnable(GL_DEPTH_TEST);
    /* Setup the view of the cube. */
    glMatrixMode(GL_PROJECTION);
    gluPerspective( /* field of view in
    degree */ 40.0, /* aspect ratio */ 1.0,
    /* Z near */ 1.0, /* Z far */ 10.0);
    glMatrixMode(GL_MODELVIEW);
    gluLookAt(0.0, 0.0, 5.0, /* eye is at
    (0,0,5) */ 0.0, 0.0, 0.0, /* center is at
    (0,0,0) */ 0.0, 1.0, 0.); /* up is in
    positive Y direction */
}
```

```
/* Adjust cube position to be
    asthetic angle. */

void display(void) {
    glClear(GL_COLOR_BUFFE
    R_BIT |
    GL_DEPTH_BUFFER_BIT);
      /* Adjust cube position to
    be asthetic angle. */
     glTranslatef(0.0, 0.0, -1.0);
     glRotatef(60, 1.0, 0.0, 0.0);
     glPushMatrix();
     glRotatef(-20, 0.0, 0.0, 1.0);
     drawBox();
     glPopMatrix();
     glTranslatef(1.0, -1.0, 0.5);
     drawBox();
     glutSwapBuffers();
}
```