Universidade Estadual de Londrina
Departamento de Computação

Programa de Mestrado em Computação

Módulo 3 - Processamento Digital de Imagens

Autor: Prof. Dr. Alan Salvany Felinto
email: alan@uel.br

(2017)

Sumário

- Transformações Rígidas: Translação, rotação e escala;
- Deformação;
- Filtros espaciais: Média, Laplaciano, Mediana, Sobel;
- Bibliografia.

Preenchimento de Regiões

Pixel 4-conectado

	1	
4	**p**	2
	3	

Pixel 8-conectado

1	2	3
8	**p**	4
7	6	5

Algoritmo recursivo para o preenchimento de regiões definido pela cor de fundo dos objetos contidos na imagem

Dados de entrada:

(x,y) - Coordenadas do ponto no interior da região

valor_velho - Valor antigo associado a cor do pixel (x,y)

valor_novo - Valor que substituirá o valor antigo do pixel

Preenchimento de Regiões

```
Procedimento Flood_fill_4(x1,y1,valor_velho,valor_novo:inteiro);
início
   se PixeldaImagem[x1,y1]=valor_velho então
   início
      PixeldaImagem[x1,y1] := valor_novo;
      flood_fill_4(x1,y1-1,valor_velho,valor_novo);
      flood_fill_4(x1,y1+1,valor_velho,valor_novo);
      flood_fill_4(x1-1,y1,valor_velho,valor_novo);
      flood_fill_4(x1+1,y1,valor_velho,valor_novo);
   fim;
fim;
```

Preenchimento de Regiões

Preenchimento de regiões definidas pelas bordas

Dados de entrada:
(x, y) - Coordenadas do ponto no interior da região
valor_borda - cor da borda que delimita a região
valor_novo - cor de preenchimento da região

Procedimento boundary_fill_4(x1,y1,valor_borda,valor_novo:inteiro);
início
  se ((PixeldaImage[x1,y1]<>valor_borda) ***E*** PixeldaImage[x1,y1]<>valor_novo))
      então
  início
      PixeldaImage[x1,y1] = valor_novo;
      boundary_fill_4(x1,y1-1,valor_borda,valor_novo);
      boundary_fill_4(x1,y1+1,valor_borda,valor_novo);
      boundary_fill_4(x1-1,y1,valor_borda,valor_novo);
      boundary_fill_4(x1+1,y1,valor_borda,valor_novo);
  fim;
fim;

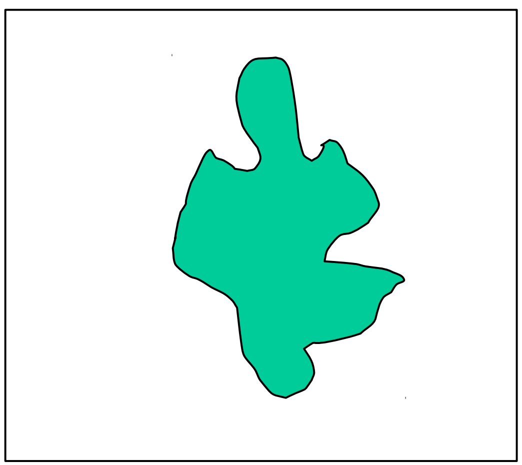
Transformadas Geométricas 2D (Translação)

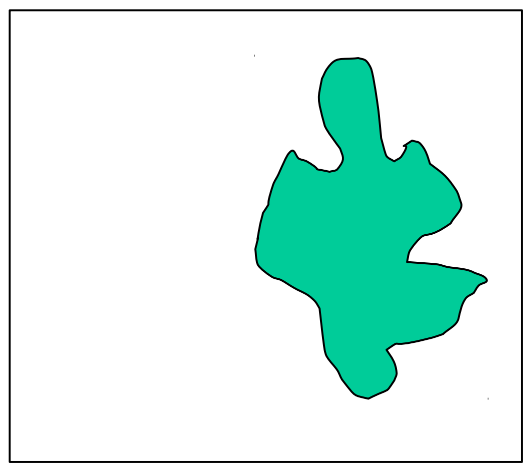
Translação:

*P´= T(△x, △y)*P*

Vetor de translação

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Leftrightarrow \begin{Bmatrix} x' = x + \Delta x \\ y' = y + \Delta y \end{Bmatrix}$$





$\triangle x = 30$

$\triangle y = 0$

Escala (2D)

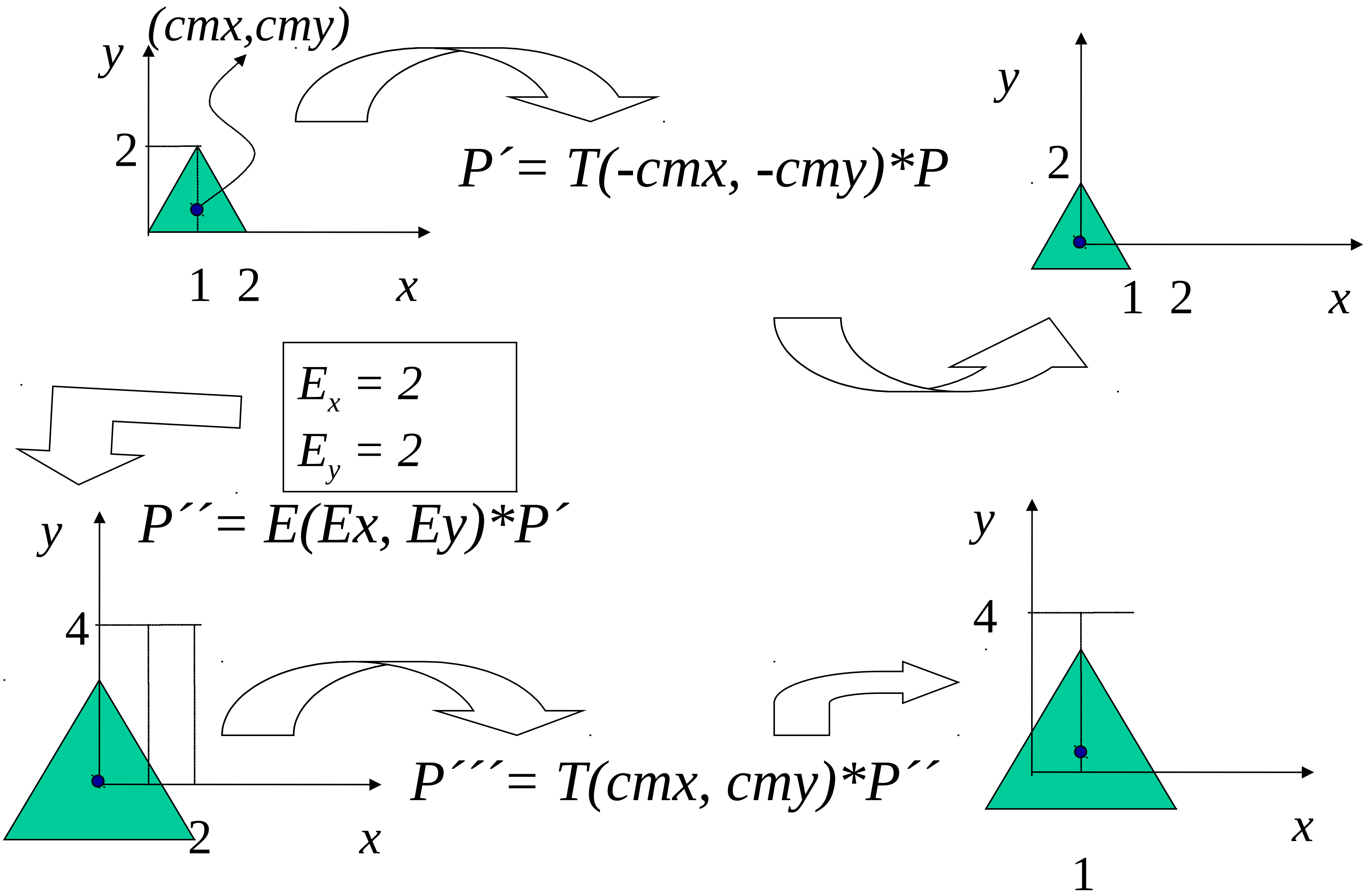
$$P' = E(Ex, Ey) * P \Rightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} E_x & 0 & 0 \\ 0 & E_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Leftrightarrow \begin{Bmatrix} x' = x * E_x \\ y' = y * E_y \end{Bmatrix}$$

Ao aumentarmos uma imagem deves-se interpolar os pontos inexistentes (Veja Zoom do módulo 8 parte 2)

Obs: A transformação em escala, além de alterar as dimensões do objeto, também produz o efeito da translação

cmx = centro de massa em x = média aritmética das coordenadas x
cmy = centro de massa em y = média aritmética das coordenadas y

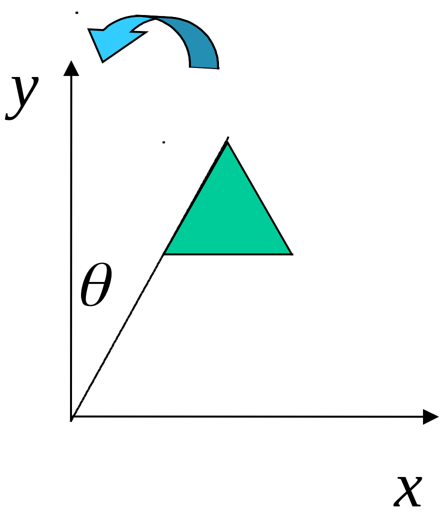
Escala ao longo do Centro de massa

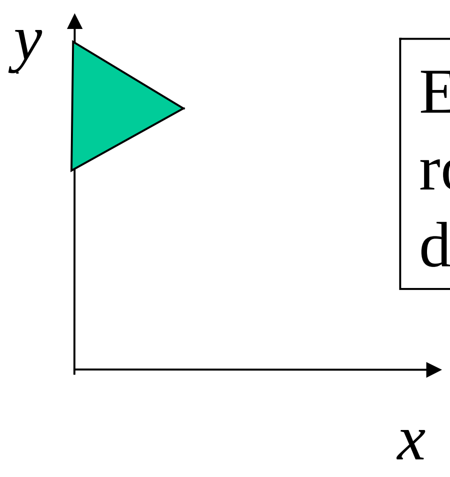


Rotação (2D)

$$P' = R(\theta) * P \Rightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Leftrightarrow$$

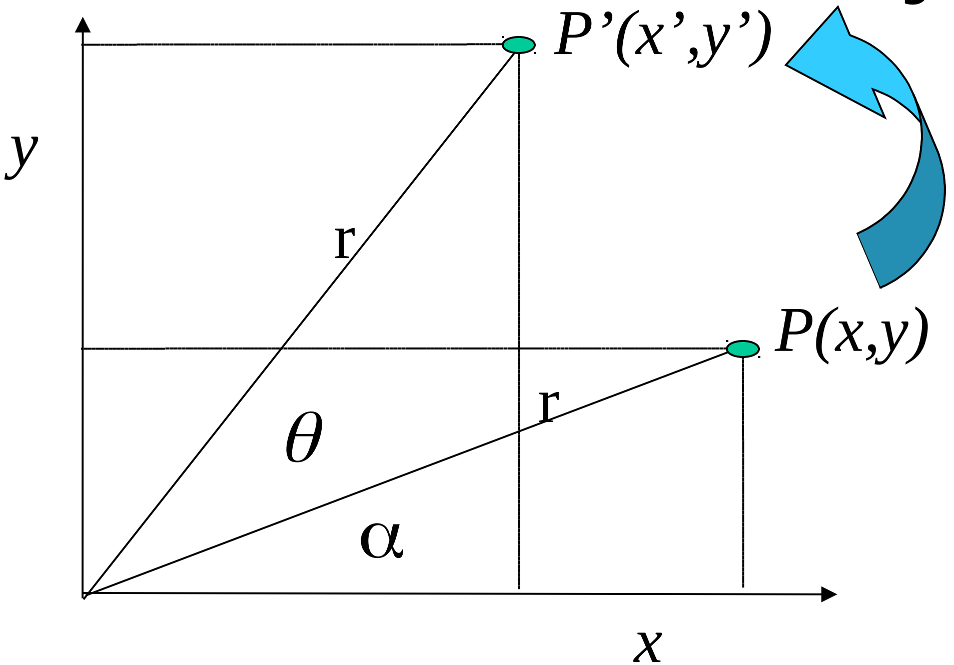
$$\Leftrightarrow \begin{cases} x' = x * \cos(\theta) - y * \sin(\theta) \\ y' = x * \sin(\theta) + y * \cos(\theta) \end{cases}$$





Existem problemas de rotação por ser um espaço discreto.

Rotação (2D)

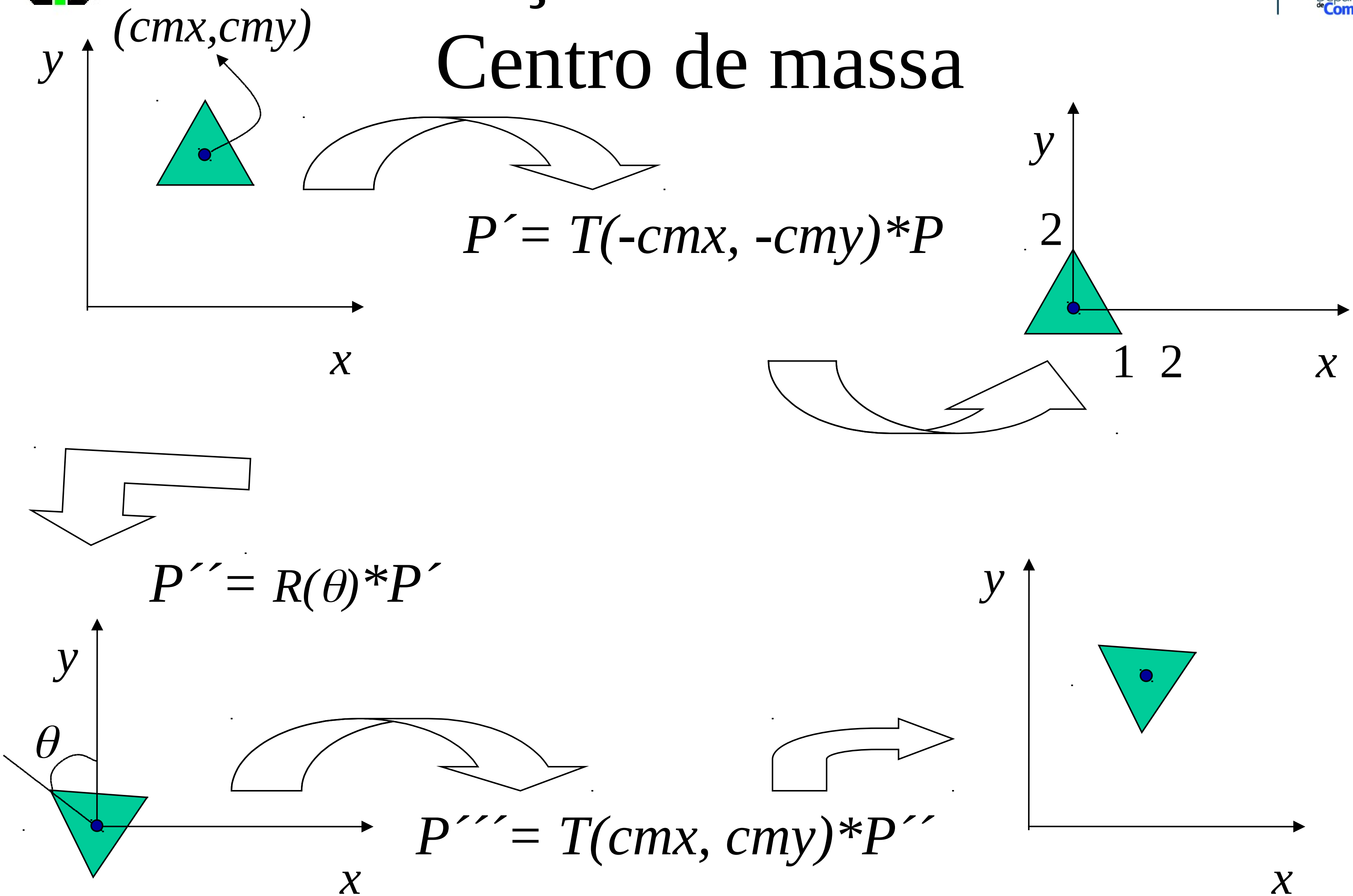


$$x = r.\cos(\alpha)$$
$$y = r.\sin(\alpha)$$

$$x' = r.\cos(\alpha+\theta) = r.\cos(\alpha).\cos(\theta) - r.\sin(\alpha) .\sin(\theta)$$
$$y' = r.\sin(\alpha+\theta) = r.\cos(\alpha).\sin(\theta) + r.\sin(\alpha) .\cos(\theta)$$

$$\boxed{\begin{aligned} x' &= x.\cos(\theta) - y .\sin(\theta) \\ y' &= x.\sin(\theta) + y .\cos(\theta) \end{aligned}}$$

Rotação ao redor do Centro de massa



Composições de Transformações Rígidas 2D

Dois exemplos:

Escala ao redor do centro de massa do objeto

$$P' = \underbrace{T(cmx, cmy) * E(Ex, Ey) * T(-cmx, -cmy)}_{\text{Uma única matriz 3x3 que resulta em duas translações e uma escala}} * P$$

Rotação ao redor do centro de massa do objeto

$$P' = T(cmx, cmy) * R(\theta) * T(-cmx, -cmy) * P$$

** Qual é a matriz para se produzir espelhamento (Flip) horizontal ou vertical?

** Quais são os problemas e/ou restrições que devemos nos preocupar ao aplicarmos transformações geométricas em uma imagem ?

Transformadas Geométricas 3D (Translação)

Translação:

*P´= T(Δx, Δy, Δy)*P*

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Leftrightarrow \begin{cases} x' = x + \Delta x \\ y' = y + \Delta y \\ z' = z + \Delta z \end{cases}$$

Escala 3D

Escala:

$$P' = E(Ex, Ey, Ez) \; * \; P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Ex & 0 & 0 & 0 \\ 0 & Ey & 0 & 0 \\ 0 & 0 & Ez & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Leftrightarrow \begin{cases} x' = x * Ex \\ y' = y * Ey \\ z' = z * Ez \end{cases}$$

Escala ao redor do centro de massa do objeto

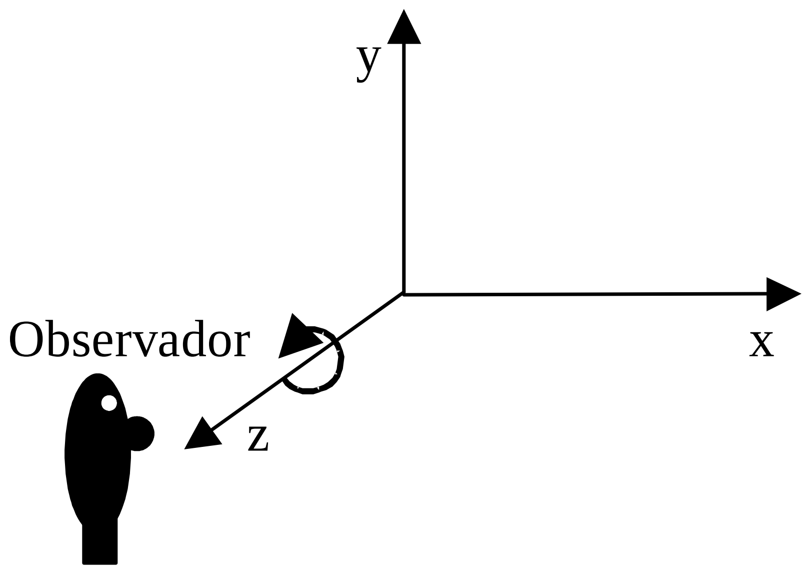
$$P' = T(cmx, cmy, cmz) * E(Ex, Ey, Ez) * T(-cmx, -cmy, -cmz) * P$$

Rotação (eixo z fixo)

$P' = Rz(\alpha) * P$ (sentido de x para y)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \begin{cases} x' = x\cos(\alpha) - y\sin(\alpha) \\ y' = x\sin(\alpha) + y\cos(\alpha) \\ z' = z \end{cases}$$



Rotação ao redor do centro de massa do objeto

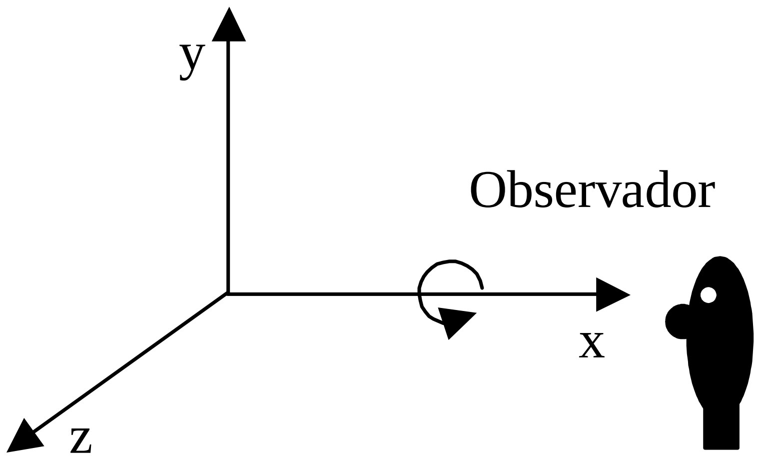
$P' = T(cmx, cmy, cmz) * Rz(\alpha) * T(-cmx, -cmy, -cmz) * P$

Rotação (eixo x fixo)

$P' = Rx(\alpha) * P$ (sentido de y para z)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \begin{cases} x' = x \\ y' = y\cos(\alpha) - z\sin(\alpha) \\ z' = y\sin(\alpha) + z\cos(\alpha) \end{cases}$$



Rotação ao redor do centro de massa do objeto

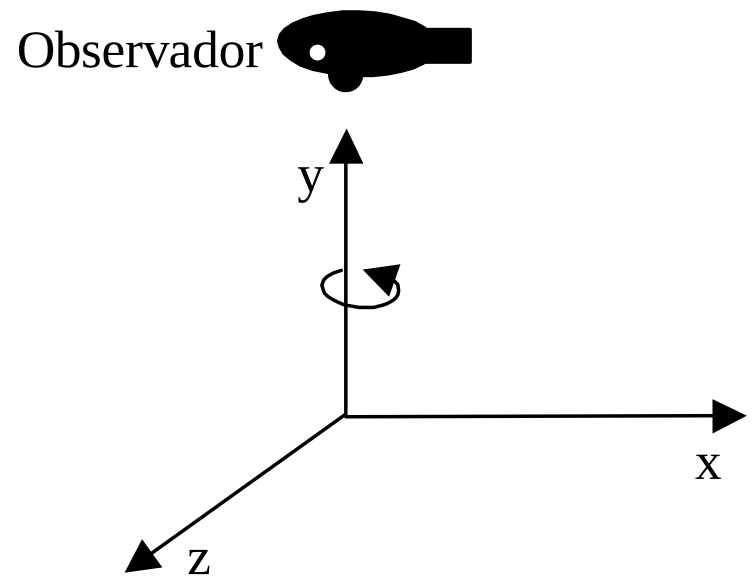
$P' = T(cmx, cmy, cmz) * Rx(\alpha) * T(-cmx, -cmy, -cmz) * P$

Rotação (eixo y fixo)

$P' = Ry(\alpha)*P$ (sentido de z para x)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \begin{cases} x' = z\sin(\alpha) + x\cos(\alpha) \\ y' = y \\ z' = z\cos(\alpha) - x\sin(\alpha) \end{cases}$$



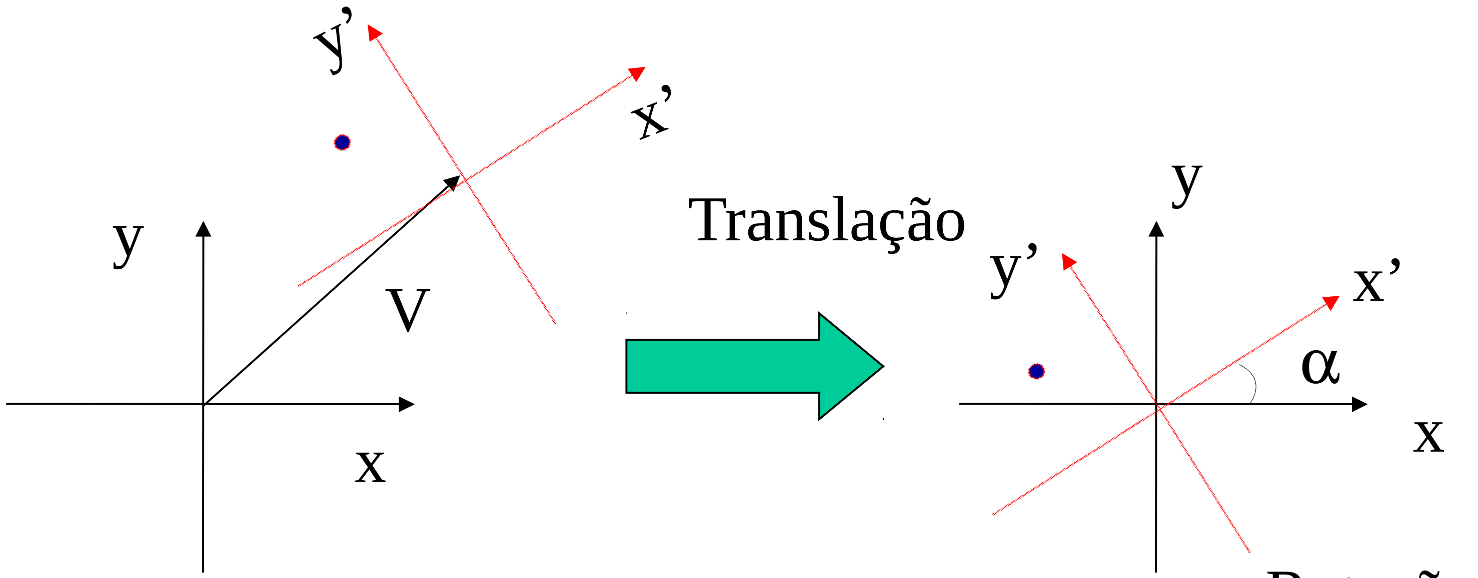
Rotação ao redor do centro de massa do objeto

$P' = T(cmx, cmy, cmz)* Ry(\alpha)*T(-cmx, -cmy, -cmz)*P$

Transformação genérica ao redor do centro de massa:

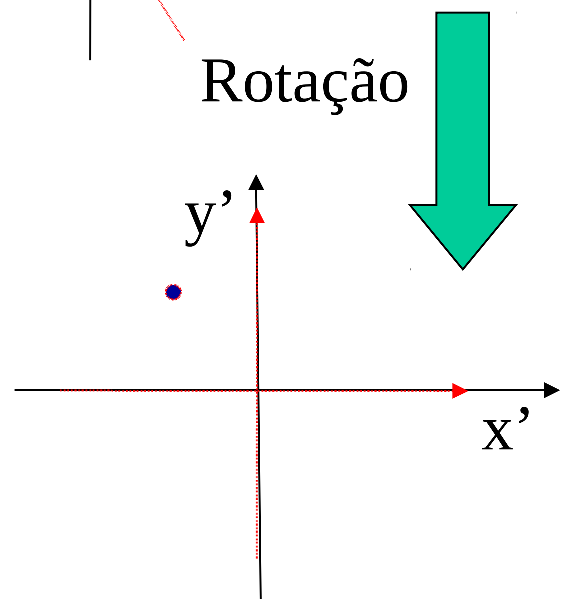
$P' = T(cmx, cmy, cmz)* Rz(\alpha)* Ry(\beta)* Rx(\theta)* T(-cmx, -cmy, -cmz)*P$

Transformações de coordenadas (x,y) para (x’,y’)



Matematicamente:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) & 0 \\ \sin(-\alpha) & \cos(-\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -Vx \\ 0 & 1 & -Vy \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Projeções no plano da Imagem

<u>Restrições:</u>

Considere a projeção em uma superfície plana ao invés de uma superfície curva e utilizando raios projetores lineares e não curvos.

Projeções geométricas planares paralelas ortográficas:

$x' = x$, $y' = y$ e $z' = 0$,

desconsidera-se o eixo z.

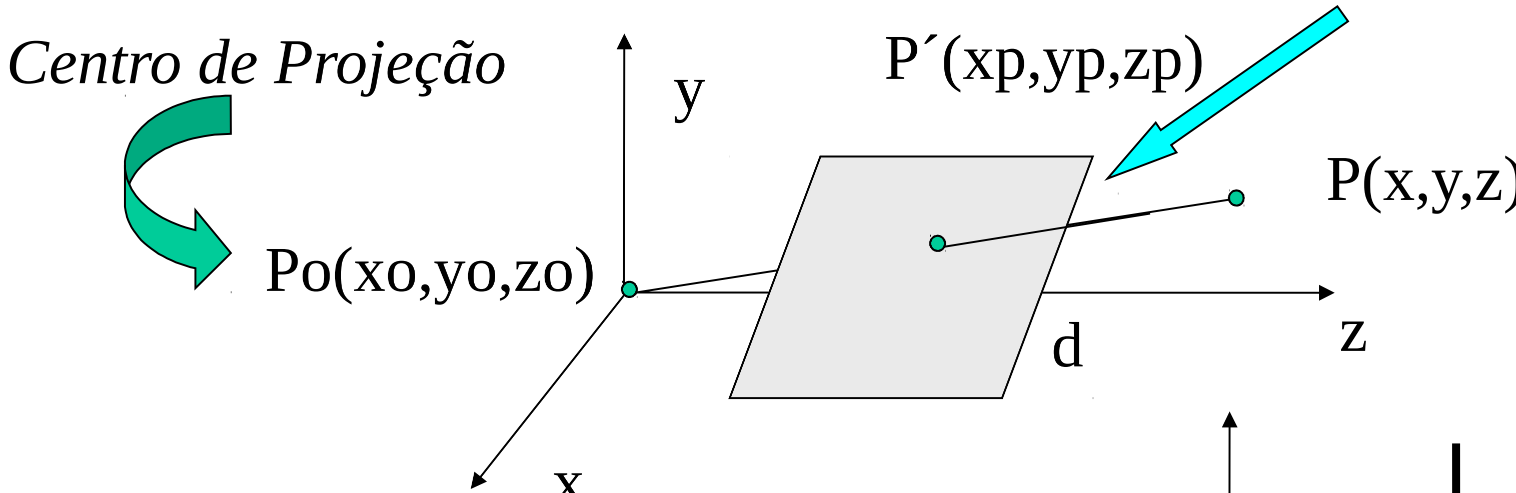
Projeções no plano da Imagem

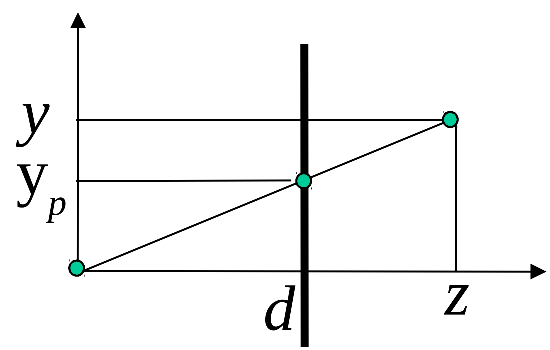
Projeções geométricas planares em perspectiva:

- Plano de projeção
- Centro de projeção (CP)

Obs: Se o centro de projeção estiver infinitamente afastado do plano de projeção então tem-se a projeção paralela, se o CP estiver a uma distância finita do plano de projeção então tem-se a projeção em perspectiva.

Projeções geométricas planares em perspectiva





Por semelhança de triângulos temos:

$$\frac{x_p}{z_p} = \frac{x}{z}, \qquad \frac{y_p}{z_p} = \frac{y}{z}$$

Como Z_p = d tem-se:

$$x_p = \frac{d.x}{z} = \frac{x}{z/d}, \qquad y_p = \frac{d.y}{z} = \frac{y}{z/d}$$

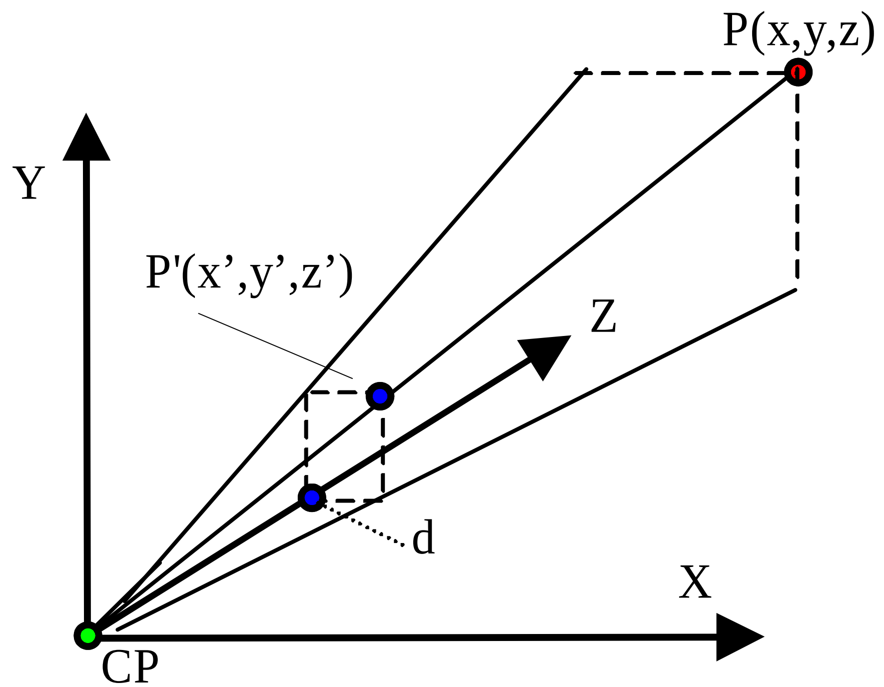
Projeções geométricas planares em perspectiva

Transformando em coordenadas homogêneas:

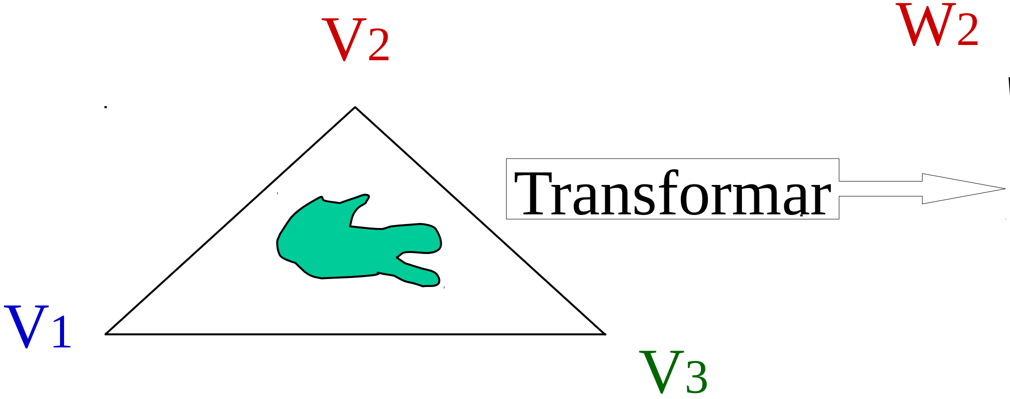
$$x_p = \frac{x}{z/d}, \qquad y_p = = \frac{y}{z/d}$$

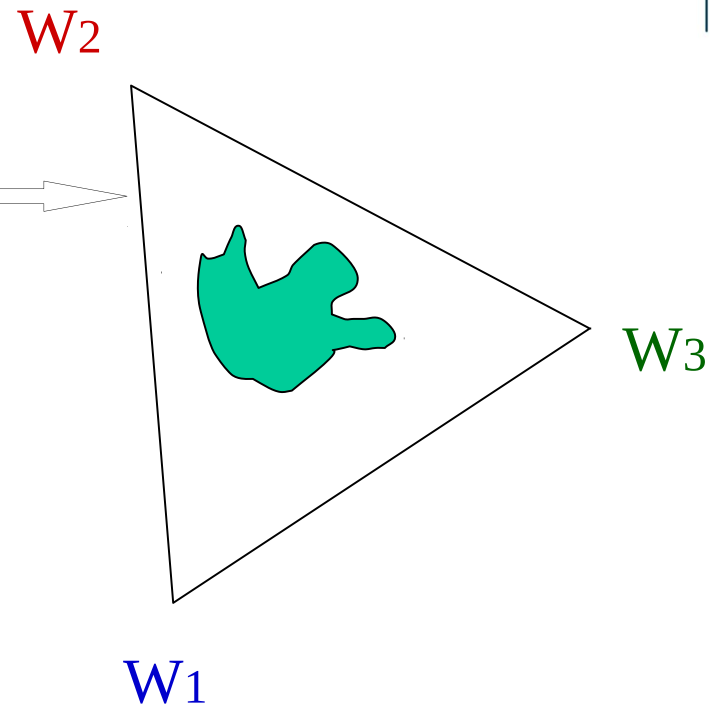
$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \frac{z}{d} \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{X}{W} \\ \frac{Y}{W} \\ \frac{Z}{W} \\ 1 \end{bmatrix} = \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{bmatrix}$$



Deformação





$$V = c_1 V_1 + c_2 V_2 + c_3 V_3$$

$$W = c_1 W_1 + c_2 W_2 + c_3 W_3$$

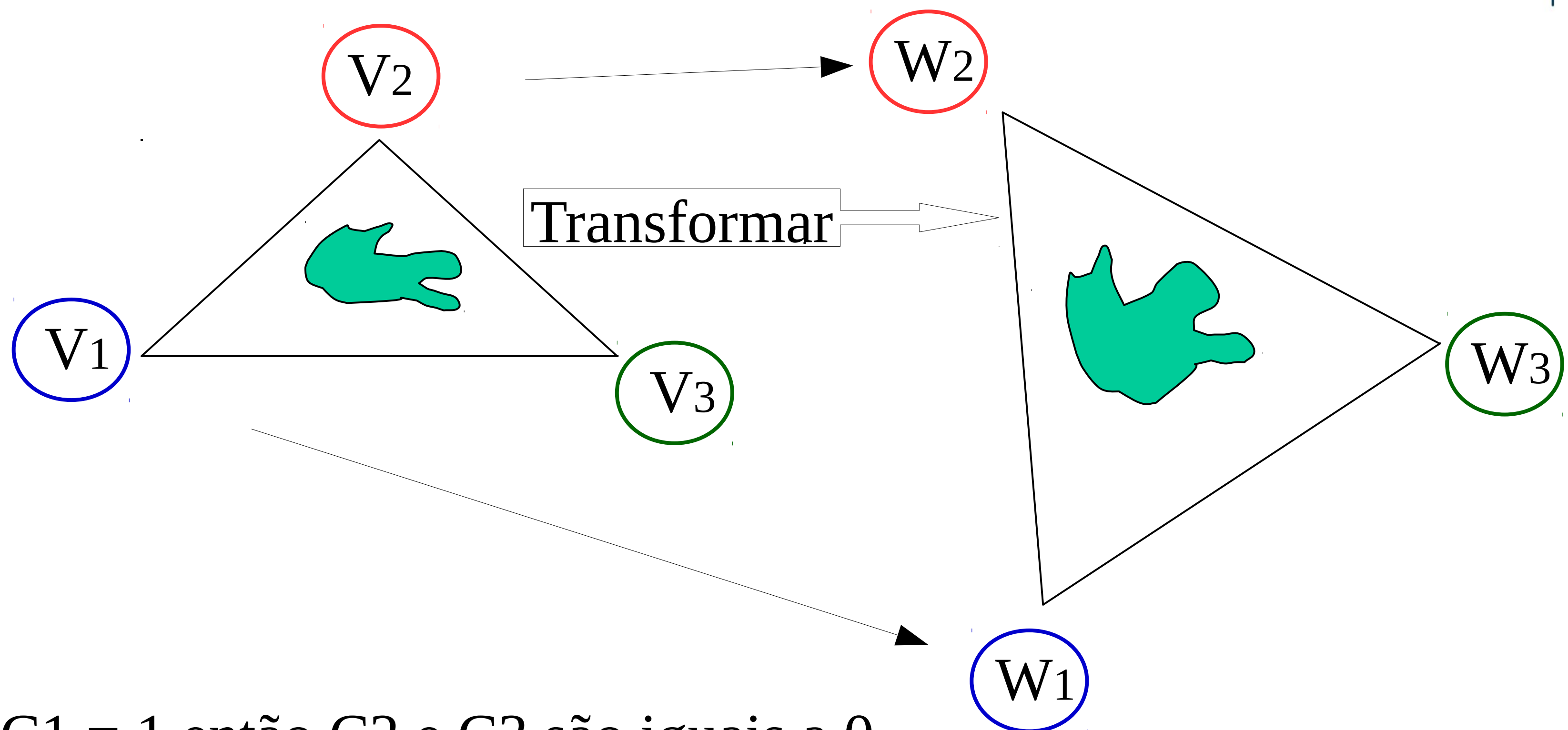
$$Onde: c_1 + c_2 + c_3 = 1$$

$$e \text{ são valores não negativos.}$$

V e W são um pontos no espaço 2d (x,y).

Para cada ponto de V(x,y) existe associado uma cor
Obs: Para implementar o algoritmo escolha o passo de variação de C_1, C_2 e C_3

Deformação



Se C1 = 1 então C2 e C3 são iguais a 0
Neste caso V1 está associado a W1 e por tanto a cor do pixel em V1 será atribuída a posição W1.

Se C1 = ½ e C2 = ½ então C3 = 0. Então a cor que está na metade entre V1 e V2 será atribuída as coordenadas localizada na metade entre W1 e W2

Imageamento Estéreo

Projeção do Espaço 3D para o espaço 2D

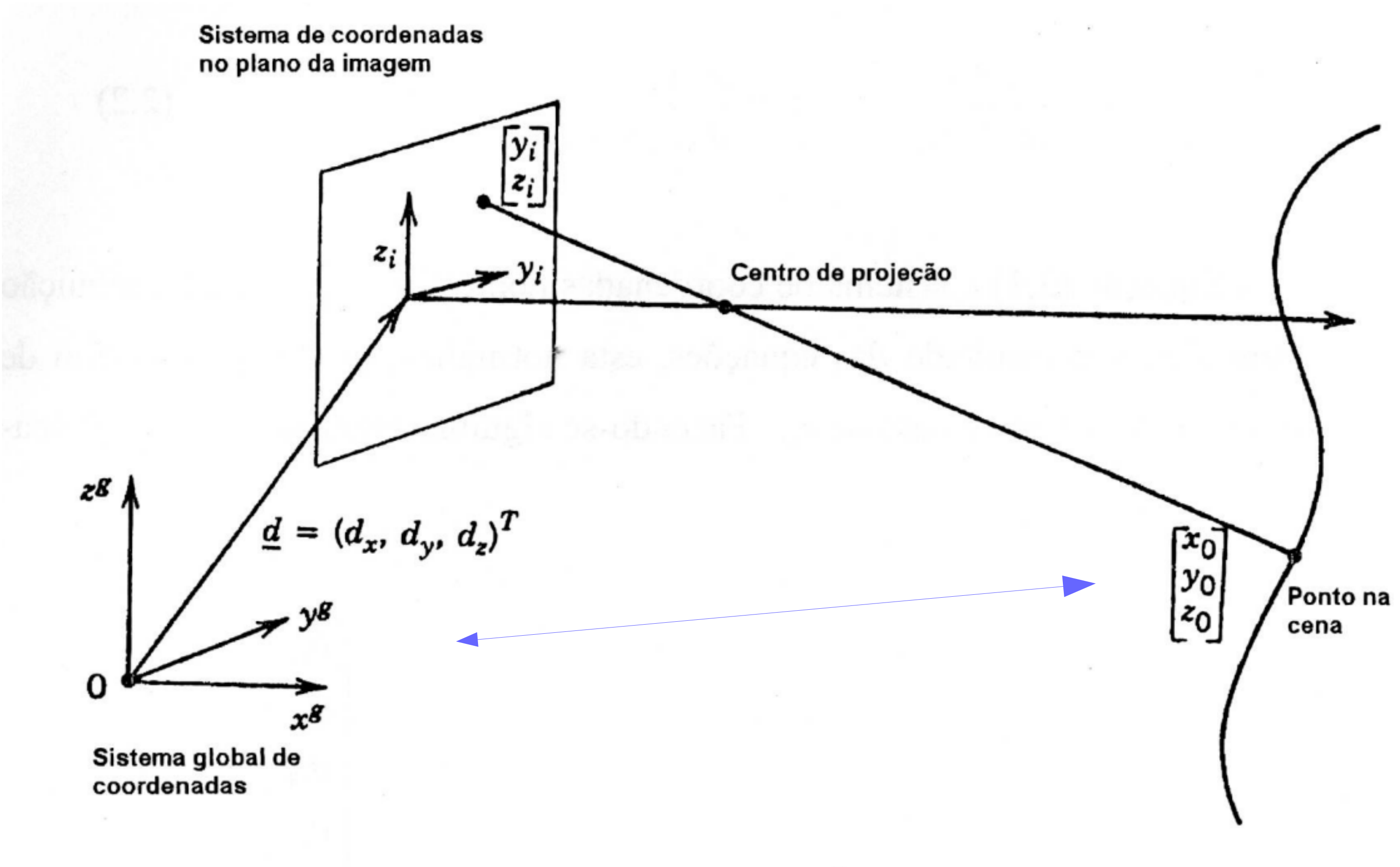


Figura - Sistema Global de Coordenadas (Schalkoff (1989))

Projeção do Espaço 3D para o espaço 2D

$$\begin{bmatrix} w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}$$

Note que $x_i = 0$ que é o plano da imagem, portanto não há necessidade de ser calculado

Ao isolarmos y_i, e z_i no sistema de Equações , temos:

$$y_i = \frac{a_{11}x_0 + a_{12}y_0 + a_{13}z_0 + a_{14}}{a_{31}x_0 + a_{32}y_0 + a_{33}z_0 + a_{34}} \quad \text{e} \quad z_i = \frac{a_{21}x_0 + a_{22}y_0 + a_{23}z_0 + a_{24}}{a_{31}x_0 + a_{32}y_0 + a_{33}z_0 + a_{34}}$$

Na Equação anterior o sistema de coordenadas homogêneas permite a atribuição a_{34} =1 sem alterar o resultado das equações, esta normalização é imposta a fim de garantir uma **solução única para os $a_{i,j}$**.

Projeção do Espaço 3D para o espaço 2D

Fazendo-se algumas manipulações algébricas tem-se:

$$\begin{bmatrix} x_0 & y_0 & z_0 & 1 & 0 & 0 & 0 & 0 & -(y_i x_0) & -(y_i y_0) & -(y_i z_0) \\ 0 & 0 & 0 & 0 & x_0 & y_0 & z_0 & 1 & -(z_i x_0) & -(z_i y_0) & -(z_i z_0) \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \begin{bmatrix} y_i \\ z_i \end{bmatrix}$$

- A cada ponto $[y_i, z_i]^T$ na imagem, que corresponde a um ponto $[k_o, y_o, z_o]^T$ da cena, forma-se um sistema de 2 equações e 11 incógnitas.
- Há a necessidade de no mínimo 6 pontos para solucionar esse sistema de equação linear.

Utilizando Câmeras em Estéreo para Obter a Informação da Profundidade

No caso da utilização de duas câmeras em estéreo, tem-se dois sistemas usando o mesmo sistema global de coordenadas $[x_o, y_o, z_o]^T$.

Considere $A_k = [a_{kij}]$, onde k = 1,2, e k representa as câmeras.

Manipulando-se as equações obtém-se: $P\underline{x_0} = \underline{F}$ Onde:

$$P = \begin{bmatrix} a_{111} - a_{131}y_{i1} & a_{112} - a_{132}y_{i1} & a_{113} - a_{133}y_{i1} \\ a_{121} - a_{131}z_{i1} & a_{122} - a_{132}z_{i1} & a_{123} - a_{133}z_{i1} \\ a_{211} - a_{231}y_{i2} & a_{212} - a_{232}x_{i2} & a_{213} - a_{233}y_{i2} \\ a_{221} - a_{231}z_{i2} & a_{222} - a_{232}z_{i2} & a_{223} - a_{233}z_{i2} \end{bmatrix}, \quad \underline{x_0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \underline{F} = \begin{bmatrix} a_{134}y_{i1} - a_{114} \\ a_{134}z_{i1} - a_{124} \\ a_{234}y_{i2} - a_{214} \\ a_{234}z_{i2} - a_{224} \end{bmatrix},$$

$[y_{i1}, z_{i1}]^T$ =coordenadas em pixeis da imagem obtida a partir da câmera 1.

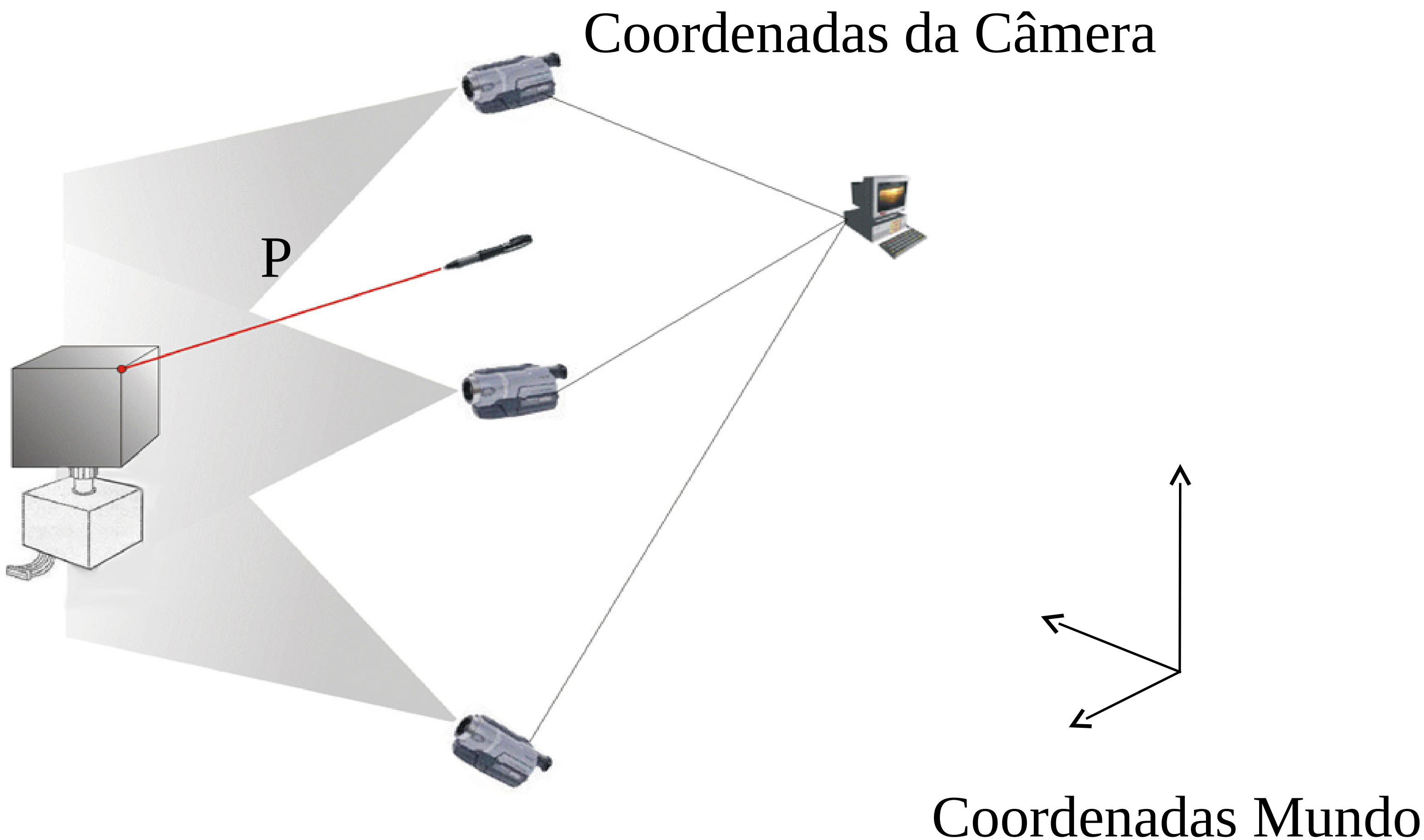
$[y_{i2}, z_{i2}]^T$ =coordenadas em pixeis da imagem obtida a partir da câmera 2.

Visão Estério

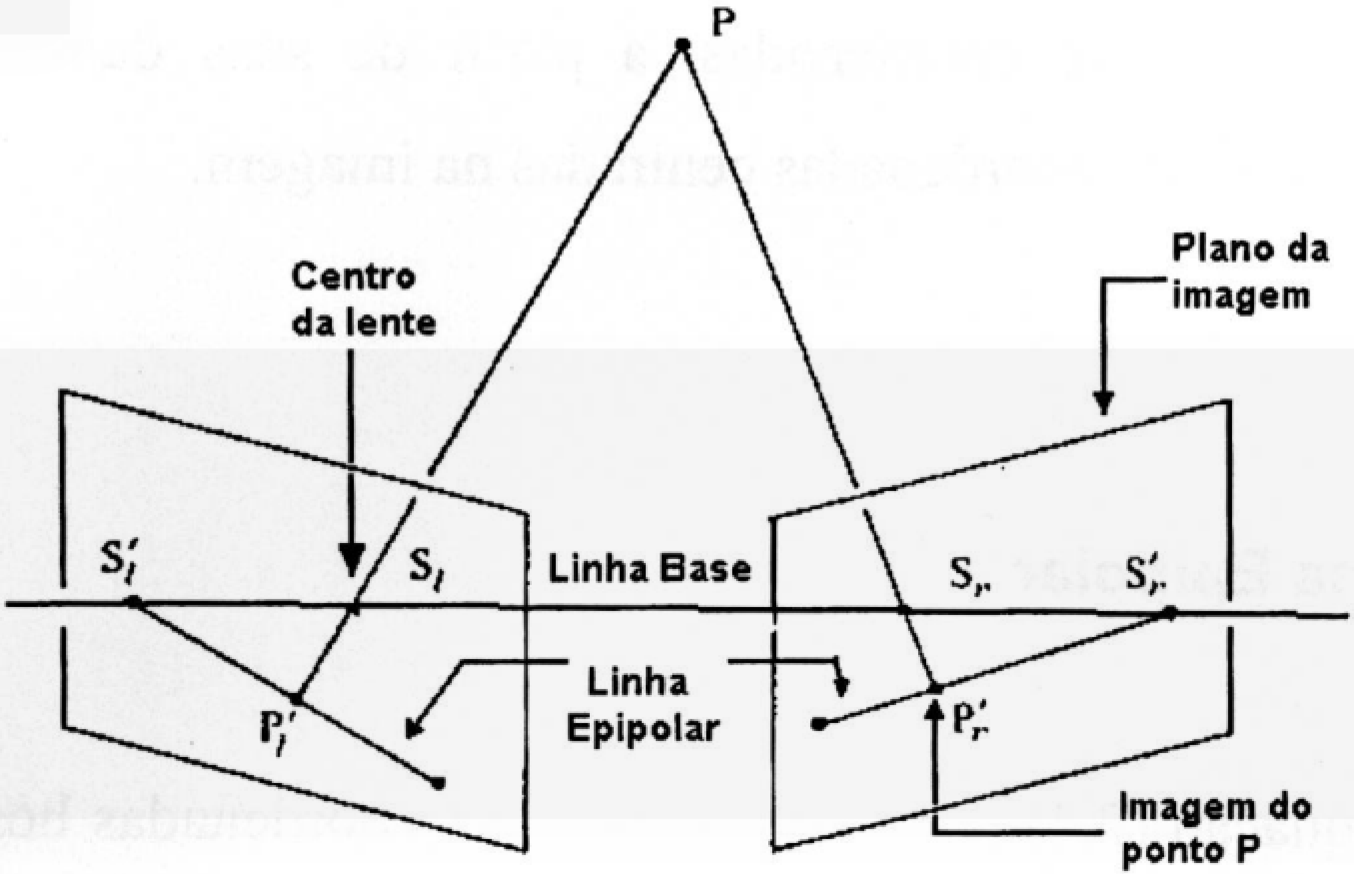
$$P = \begin{bmatrix} a_{111} - a_{131}y_{i1} & a_{112} - a_{132}y_{i1} & a_{113} - a_{133}y_{i1} \\ a_{121} - a_{131}z_{i1} & a_{122} - a_{132}z_{i1} & a_{123} - a_{133}z_{i1} \\ a_{211} - a_{231}y_{i2} & a_{212} - a_{232}x_{i2} & a_{213} - a_{233}y_{i2} \\ a_{221} - a_{231}z_{i2} & a_{222} - a_{232}z_{i2} & a_{223} - a_{233}z_{i2} \end{bmatrix}, \quad \underline{x_0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \underline{F} = \begin{bmatrix} a_{134}y_{i1} - a_{114} \\ a_{134}z_{i1} - a_{124} \\ a_{234}y_{i2} - a_{214} \\ a_{234}z_{i2} - a_{224} \end{bmatrix},$$

- 4 equações e 3 incógnitas ($[k_o, y_o, z_o]^T$).
- Dados os pontos em estéreo ($[y_{i1}, z_{i1}]^T$ e $[y_{i2}, z_{i2}]^T$) calcula-se o ponto correspondente no espaço 3D.
- E como faríamos se usássemos mais de duas câmeras ?
- Qual é a vantagem de se utilizar mais de 2 câmeras ?

Modelo Genérico de Visão Estéreo

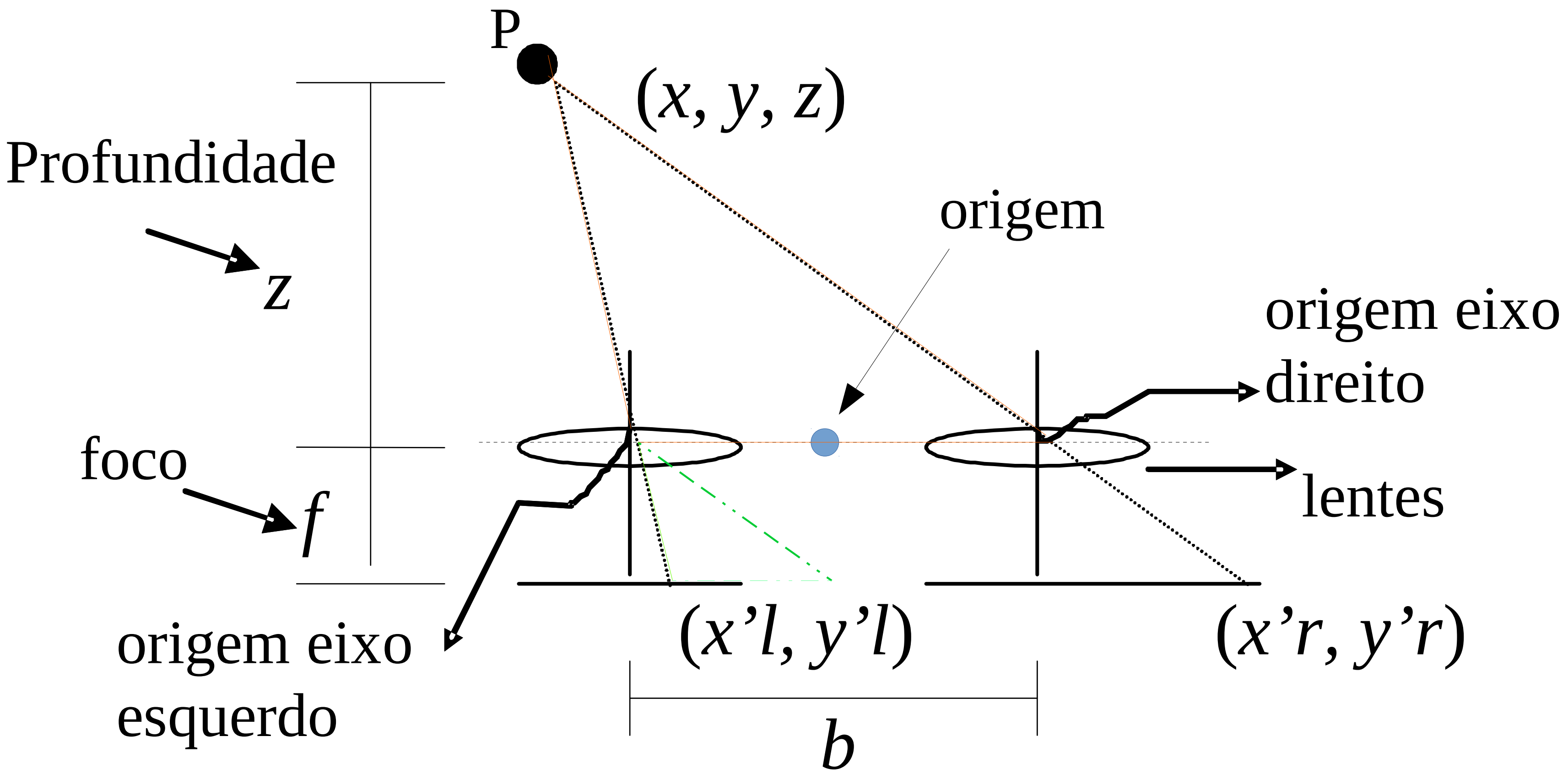


Linha Epipolar



A linha epipolar é formada pela intersecção do plano da imagem com o plano formado pelos pontos S_i, S_r (centro da lente esquerda e direita respectivamente) e P (ponto de interesse na cena).

Visão Estéreo Simplificada



Semelhança de triângulos: Z/b = f/(x'r - x'l)

Portanto, $Z = f\,b/(x'r - x'l)$

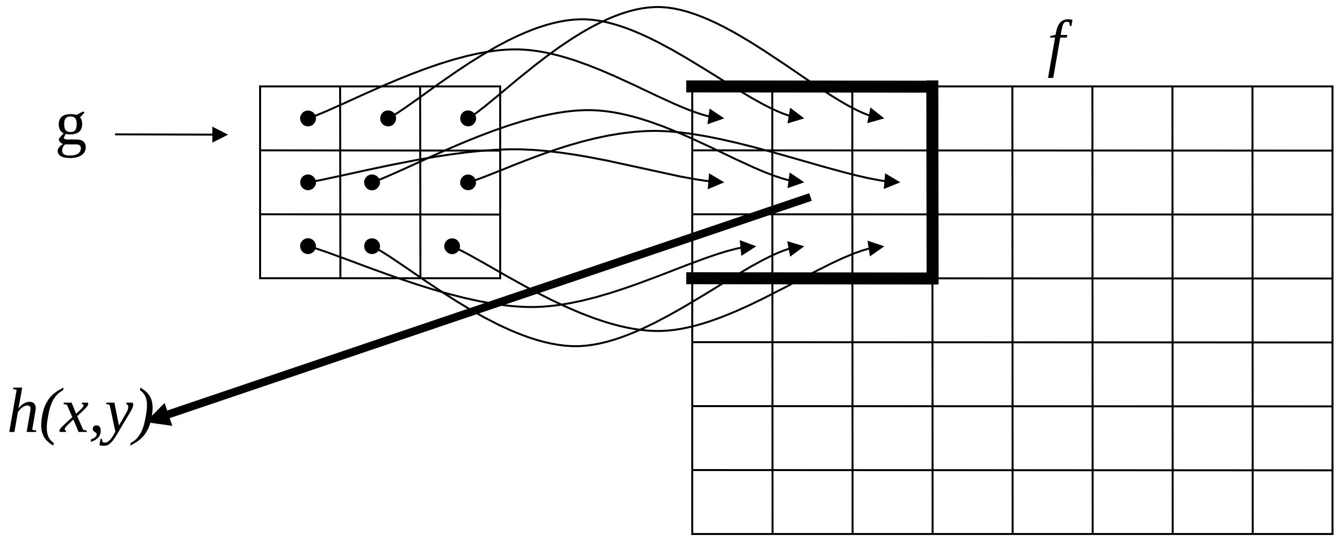
Filtragem espacial:

Convolução:

Núcleo de convolução

$$h(x,y) = f(x,y) * g(x,y) = \int_{-\infty}^{\infty}\int f(\alpha,\beta)\,.\,g(x-\alpha, y-\beta)\,d\alpha\, d\beta$$

Exemplo na forma discreta com um núcleo de convolução 3x3:



Filtragem Espacial e o Filtro da Média

(x-1,y-1)	*(x,y-1)*	*(x+1,y-1)*
(x-1,y)	*(x,y)*	*(x+1,y)*
(x-1,y+1)	*(x,y+1)*	*(x+1,y+1)*

```
for ( x =1;  x < m - 2; ++x )
for ( y =1;  y < n - 2; ++y )
   h[x][y] =  f[x-1][y-1]*g[0][0] + f[x][y-1] *g[0][1] + f[x+1][y-1]*g[0][2]
           +  f[x-1][y]   *g[1][0] + f[x][y]   *g[1][1]+ f[x+1][y]   *g[1][2]
           +  f[x-1][y+1]*g[2][0] + f[x][y+1]*g[2][1] + f[x+1][y+1]*g[2][2]
```

Filtro da média

1/9	*1/9*	*1/9*
1/9	*1/9*	*1/9*
1/9	*1/9*	*1/9*

- Média dos pixeis vizinhos.
- Retira ruído porém borra a imagem.
- Quanto maior o tamanho da janela melhor é o filtro do ruído.

Laplaciano

Extração de Bordas utilizando gradiente

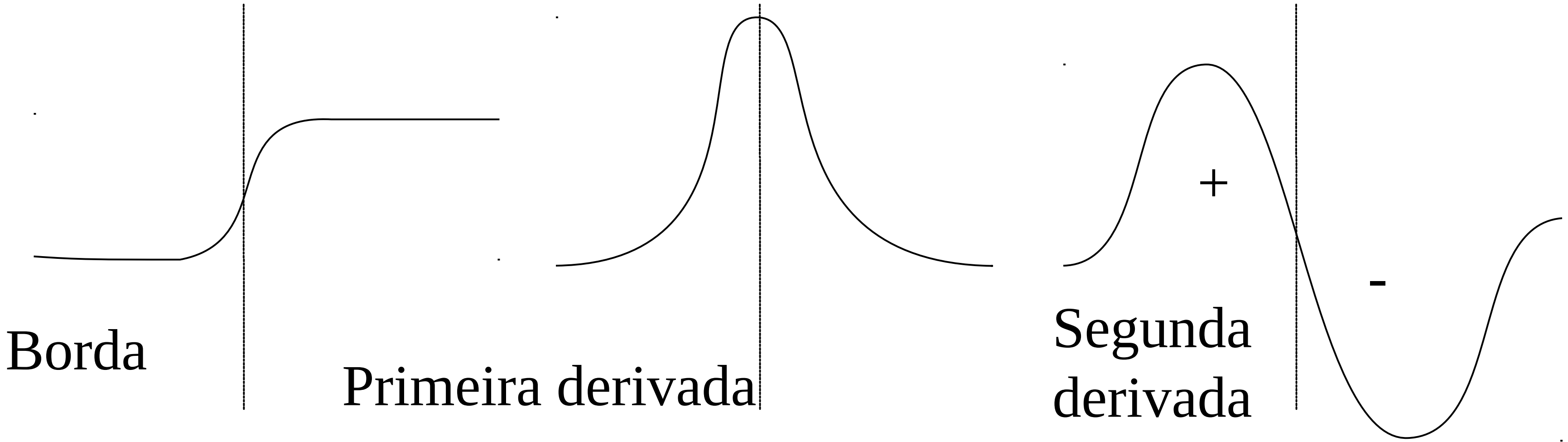
O operador laplaciano é uma derivada segunda de uma imagem

0	*-1*	*0*
-1	*4*	*-1*
0	*-1*	*0*

ou

-1	*-1*	*-1*
-1	*8*	*-1*
-1	*-1*	*-1*

Para detectar bordas procure os cruzamentos por zeros



Roberts, Sobel, Prewitt

Roberts:

1	0
0	-1

R_1

0	1
-1	0

R_2

Utilize um limiar no *resultado*

$$resultado = \sqrt{R_1^2 + R_2^2}$$

Sobel

-1	-2	-1
0	0	0
1	2	1

R_1

-1	0	1
-2	0	2
-1	0	1

R_2

Prewitt

-1	-1	-1
0	0	0
1	1	1

R_1

-1	0	1
-1	0	1
-1	0	1

R_2

Sobel

- Leitura de uma imagem
- Para os pixeis da imagem faça:
  - Para cada pixel central calcula-se R1 e R2 utilizando as matrizes do Sobel.
  - R = raiz_quadrada (R^2_1 +R^2_2)
  - Se (R > limiar) então o pixel é pintado de preto
  - Senão o pixel é pintado de branco.
- Mostre a imagem

Kirsch

Kirsch

$$R_1 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad R_2 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad R_3 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad R_4 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \quad R_5 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$$

$$R_6 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad R_7 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad R_8 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

Obs:

-Utilize um limiar no resultado.

-A direção da borda é dada pela máscara de maior resultado

$$resultado = \sqrt{R_1^2 + R_2^2 + R_3^2 + R_4^2 + R_5^2 + R_6^2 + R_7^2 + R_8^2}$$

Mediana

Filtro da Mediana: ordene uma região da imagem e utilize o pixel mediano como resultado do processamento.

Exemplo:

10	*20*	*100*
200	*5*	*15*
30	*18*	*40*

5 - 10 - 15 - 18 - 20 - 30 - 40 - 100 - 200

Resultado do processamento

O filtro da mediana retira ruído sem borrar a imagem, é um filtro não linear

Exercícios

1) Implemente os filtros no domínio do espaço: Média, Sobel, Mediana.

2) Faça um programa que: Mostre a quantidade de manchas na imagem; pinte de azul as manchas sem buracos; e pinte de verde as manchas com buracos. Pinte de vermelho a maior mancha, Pinte de amarelo a menor mancha,

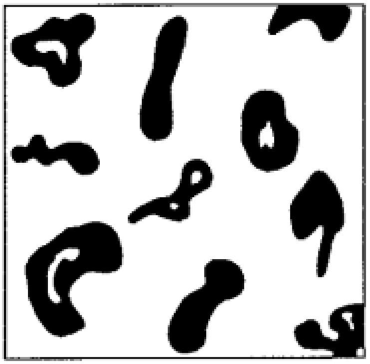
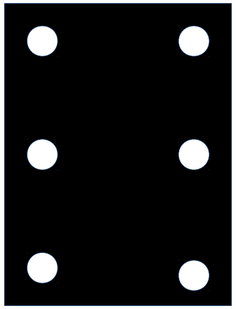
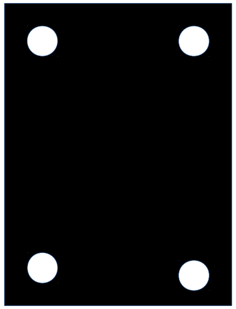


Imagem exemplo de entrada.

3) Determine se a peça possui 6 ou 4 buracos. Se a peça contiver 6 buracos classifique-a como tipo **A**, Se a peça contiver 4 buracos classifique-a como tipo **B**, caso contrário rejeite a peça classificando a como tipo **R.**





OBS:
- Pode utilizar qualquer rotina do OpenCv nos exercícios.
- Para preenchimento de regiões utilize o FloodFill do opencv.

Bibliografias

- [Castleman (1996)] Castleman, K. R. Digital Image Processing. Prentice Hall pp-667. 1996.
- [Conci 2008] Conci, A; Azevedo, E; Leta, F. R. Computação Gráfica, volume2. Elsevier. p. 407. 2008.
- [Gonzalez (1993)] Gonzalez, R. F.; Woods, R. E. Digital Image Processing. Addison-Wesley, p 716. 1993.
- [Gonzalez (2009)] Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing. PHI Learning Pvt. Ltd-new Delhi; 3ª edição (2009).
- [Hearn (1997)] Hearn, D; Baker, M. P. Computer Graphics, C Version. Prentice Hall, 2ª edição, p. 650. 1997.
- [FOLEY_90] Foley, James D. et al : Computer Graphics - Principles and Practice, Addison-Wesley Publishing Company, 1990.
- [PERSIANO_89] Persiano, R.C.M.; Oliveira, A.A.F. :Introdução à Computação Gráfica, Livros Técnicos e Científicos Editora Ltda., 1989.
- [Pratt (1991)] Pratt, Willian K. Digital Image Processing. A Wiley-Interscience Publication, 2a edição. 698 p. 1991.