

Notação e Transformações Geométricas 2D

Representação matemática de pontos

A cada ponto $P(x,y)$ no plano far-se-á corresponder um vetor

$$P = \begin{bmatrix} x \\ y \end{bmatrix}$$

Assim, uma transformação linear, dada por uma matriz M , poderá transformar P em P' escrevendo-se:

$$P' = M.P$$

Por conseguinte, o operador M deverá ser uma matriz 2x2.

Mas, em alternativa, os pontos também se poderão especificar em Coordenadas Homogéneas^{*}.

M.Próspero

Pontos em Coordenadas Homogéneas

2D

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

$W \neq 0$

$$\begin{bmatrix} x.W \\ y.W \\ W \end{bmatrix}$$

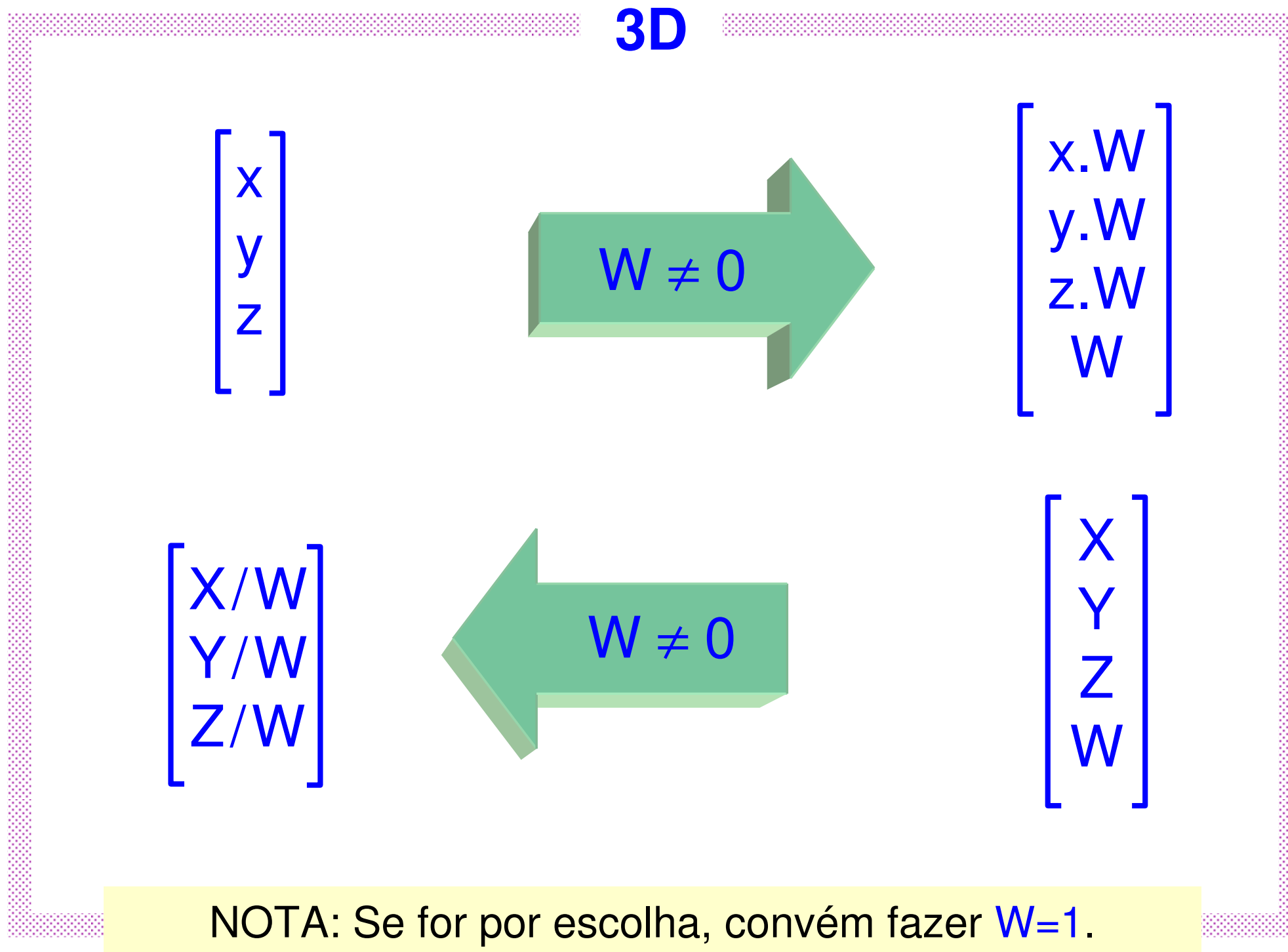
$$\begin{bmatrix} X/W \\ Y/W \end{bmatrix}$$

$W \neq 0$

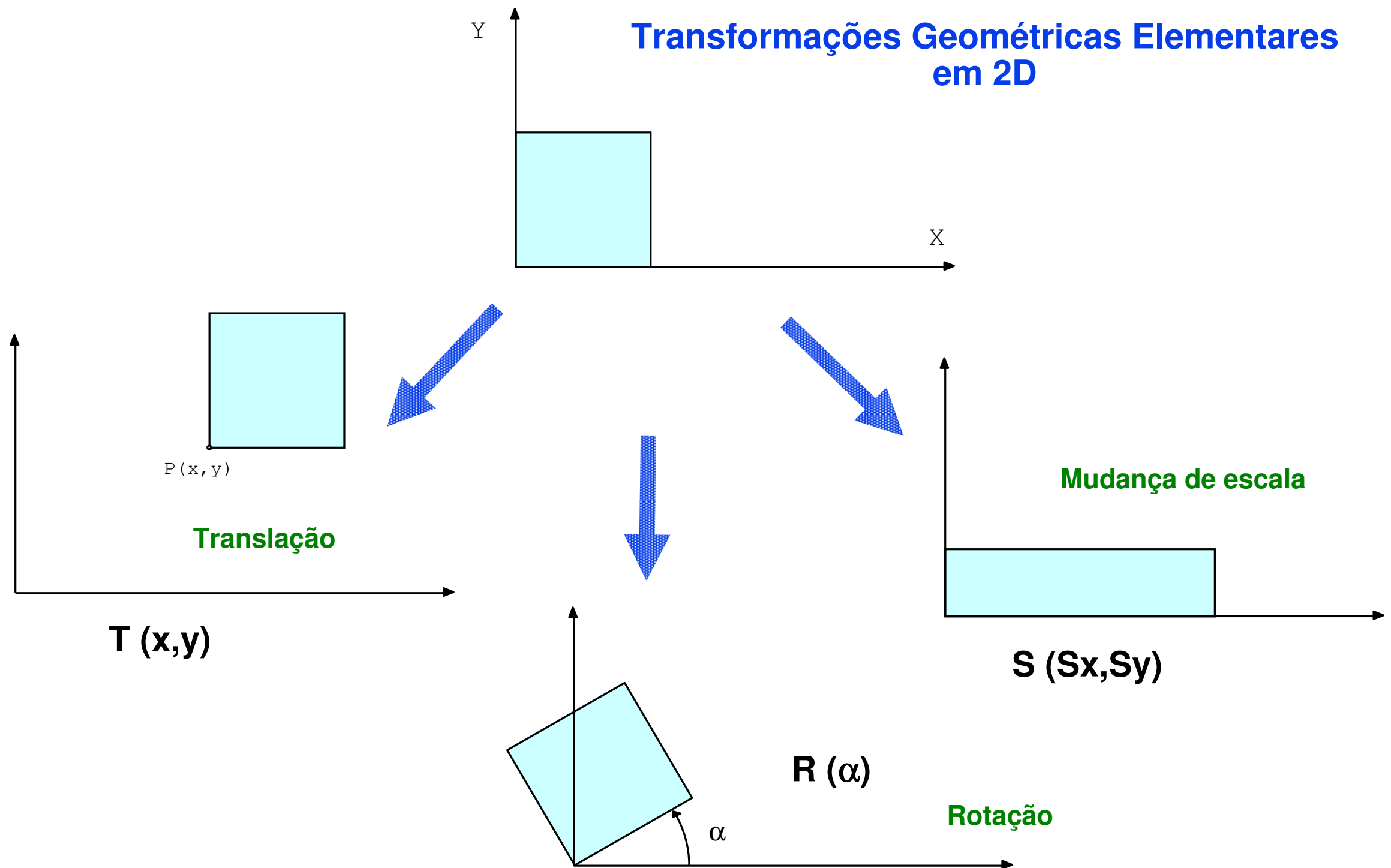
$$\begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

NOTA: Se for por escolha, convém fazer $W=1$.

Pontos em Coordenadas Homogéneas



Transformações Geométricas Elementares em 2D



M. Próspero

Tratamento matemático (1)

$$P(x, y) = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \leftarrow \text{em Coordenadas Homogéneas 2D}$$

$$P'(x', y') = M \cdot P(x, y)$$

Translação

$$T(T_x, T_y) = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x + T_x$$

$$y' = y + T_y$$

Mudança de escala

$$S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fatores de escala

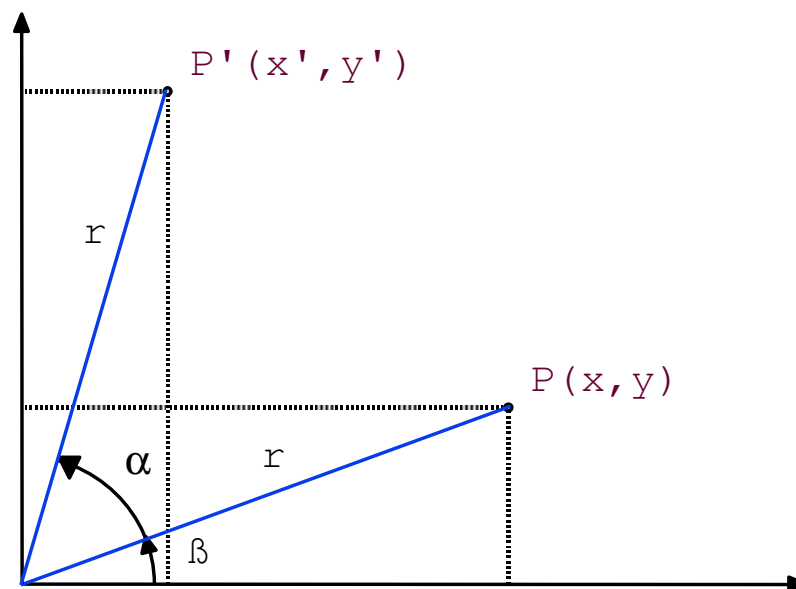
$$x' = S_x * x$$

$$y' = S_y * y$$

Tratamento matemático (2)

Rotação

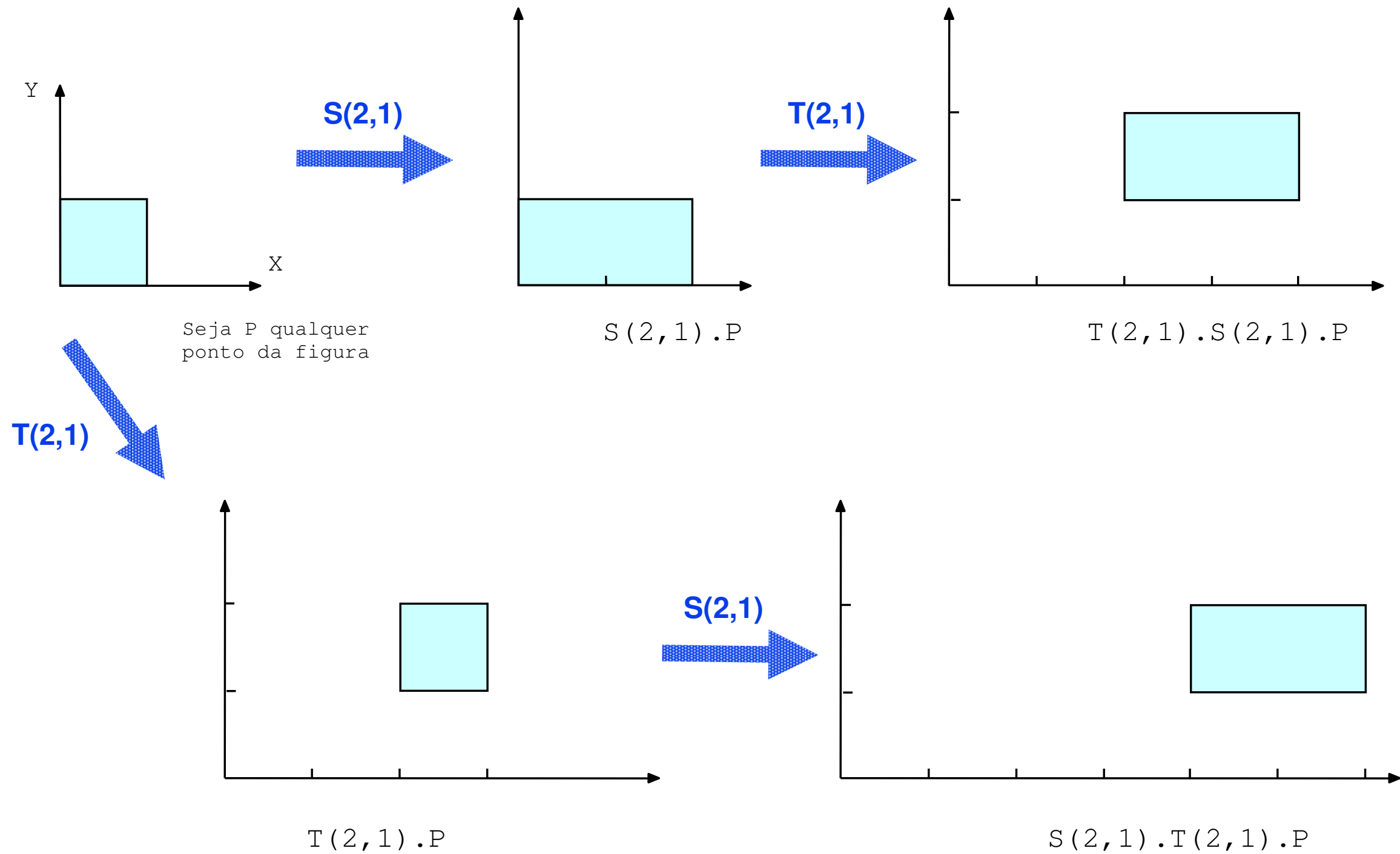
$$R(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



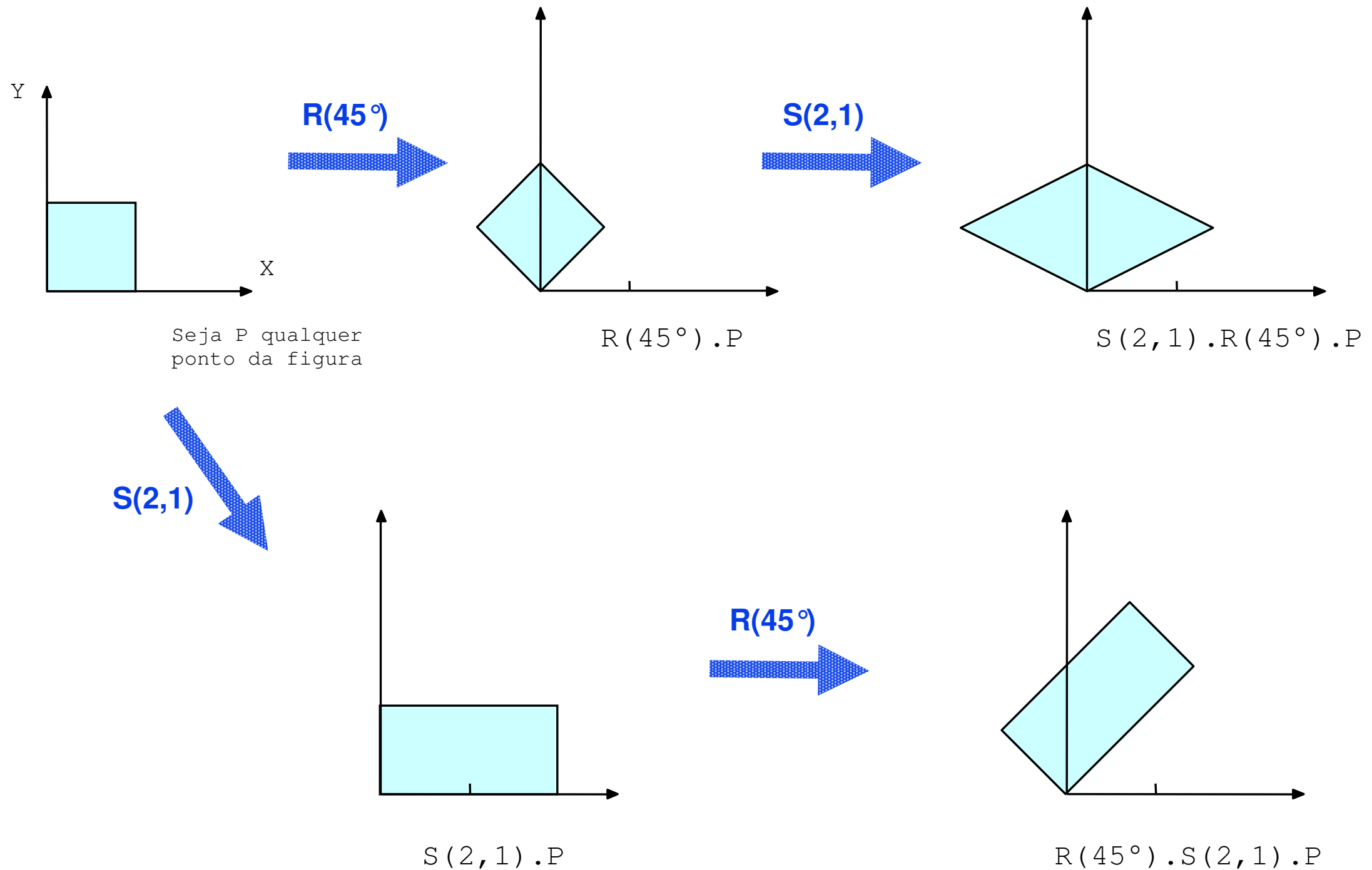
$$\begin{aligned} x' &= r \cdot \cos(\beta + \alpha) \\ &= r \cdot \cos \beta \cdot \cos \alpha - r \cdot \sin \beta \cdot \sin \alpha \\ &= x \cdot \cos \alpha - y \cdot \sin \alpha \end{aligned}$$

$$\begin{aligned} y' &= r \cdot \sin(\beta + \alpha) \\ &= r \cdot \cos \beta \cdot \sin \alpha + r \cdot \sin \beta \cdot \cos \alpha \\ &= x \cdot \sin \alpha + y \cdot \cos \alpha \end{aligned}$$

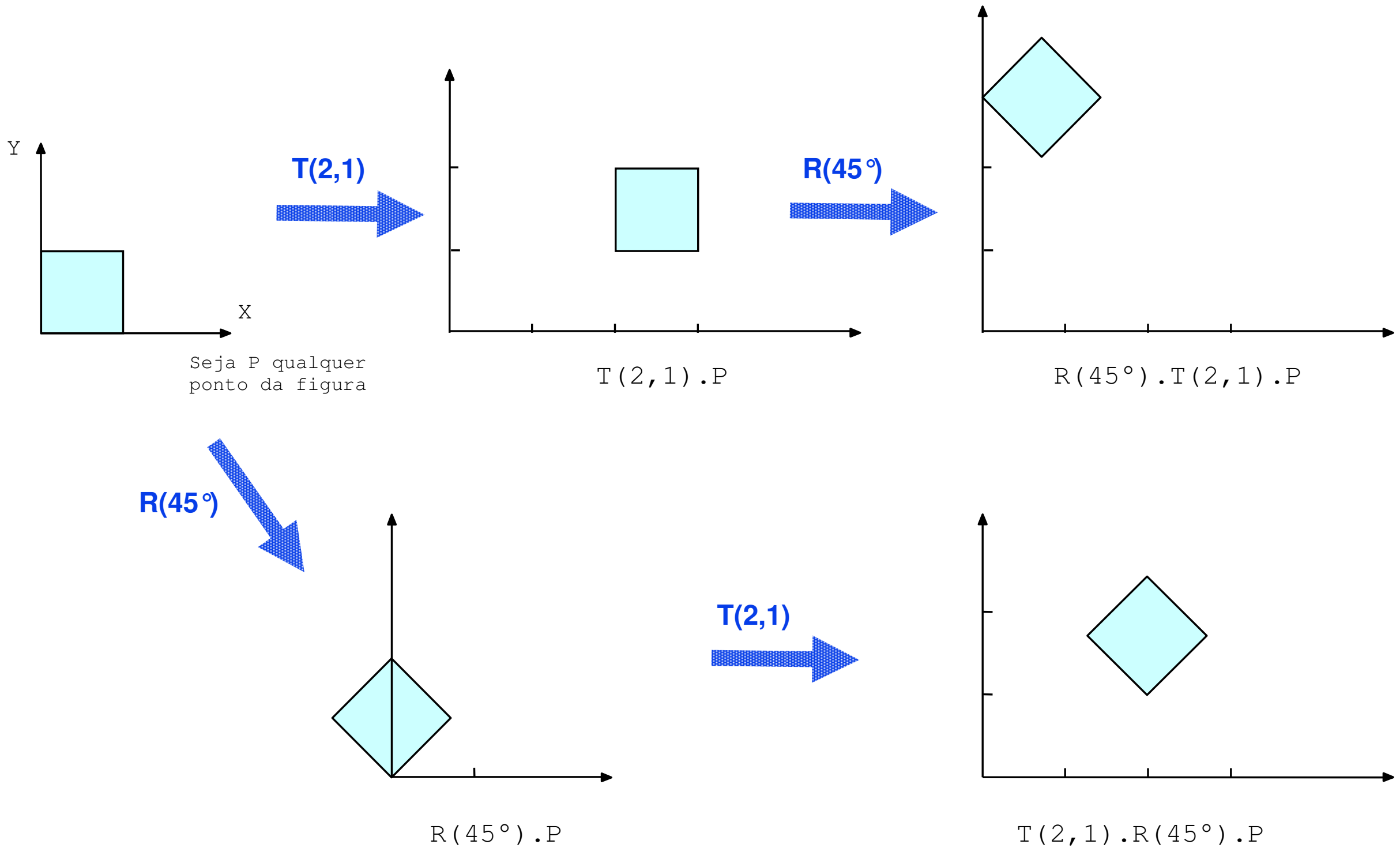
Composição de Transformações Geométricas: S e T



Composição de Transformações Geométricas: R e S



Composição de Transformações Geométricas: T e R



2D

Quando é que se pode garantir a comutatividade?

$R(\infty).R(\beta)$

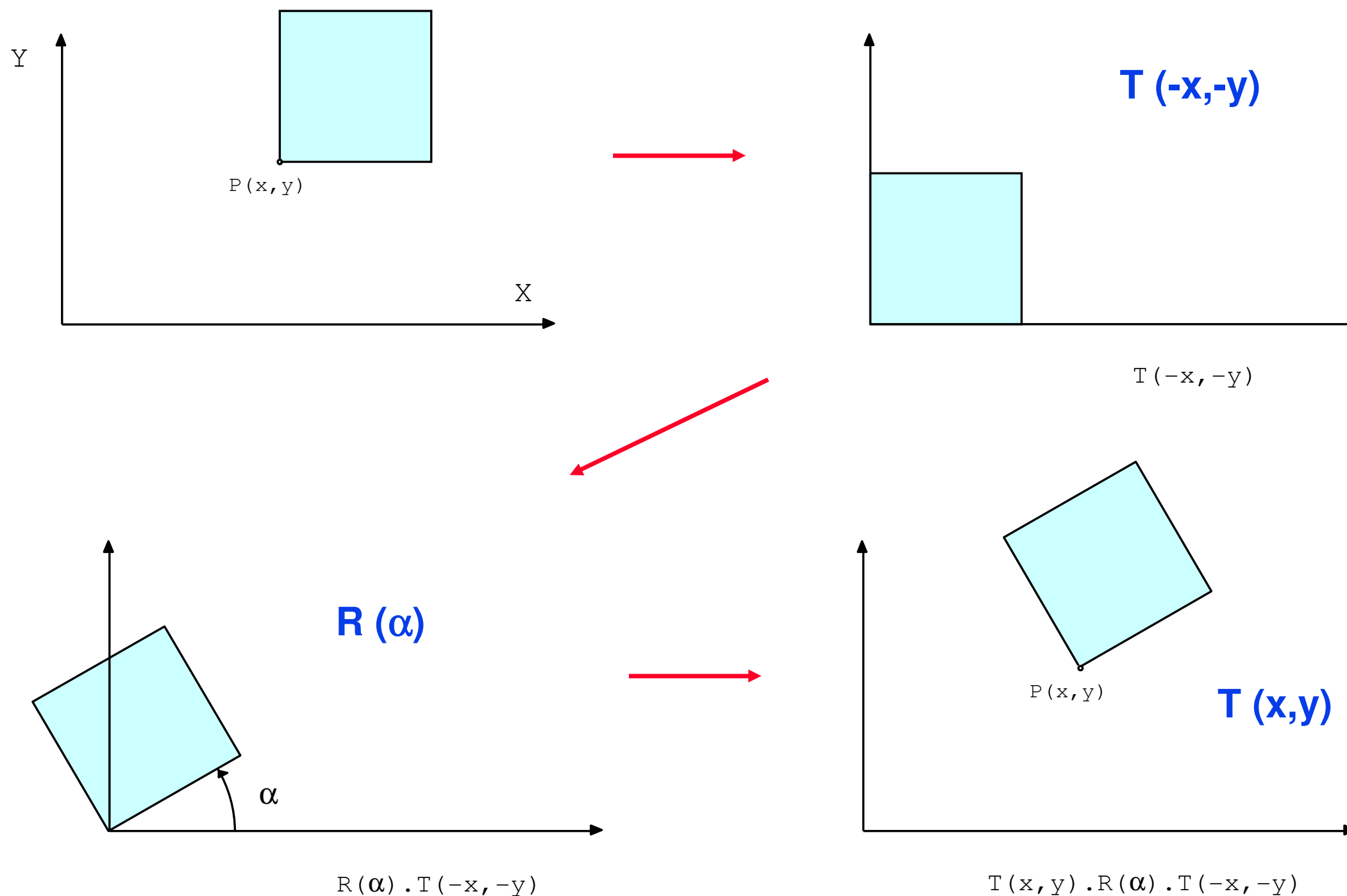
$S(K1,K2).S(K3,K4)$

$T(D1,D2).T(D3,D4)$

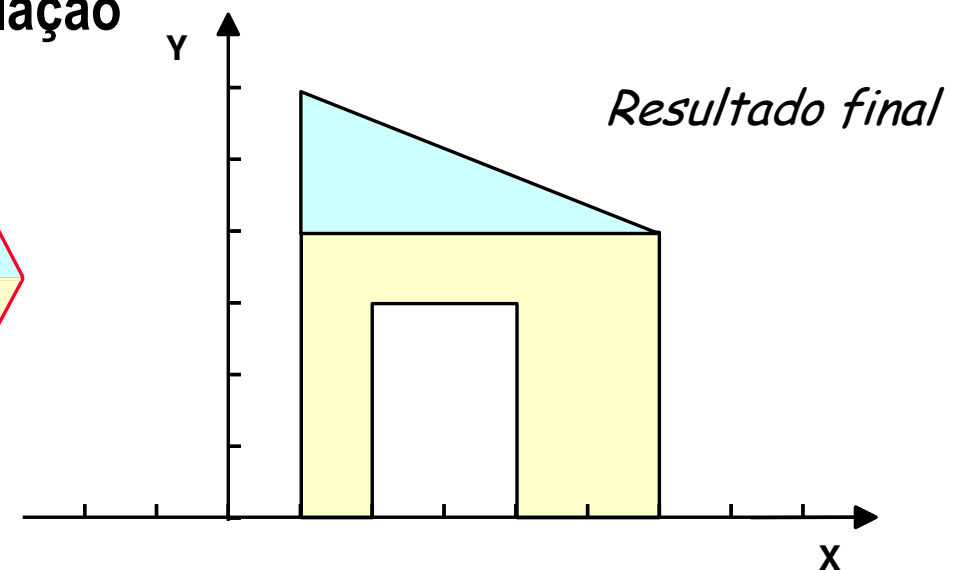
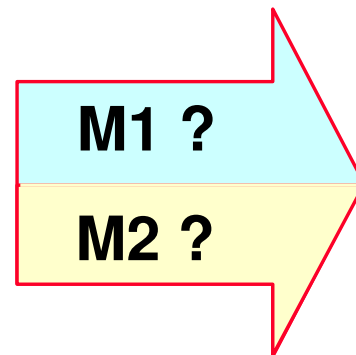
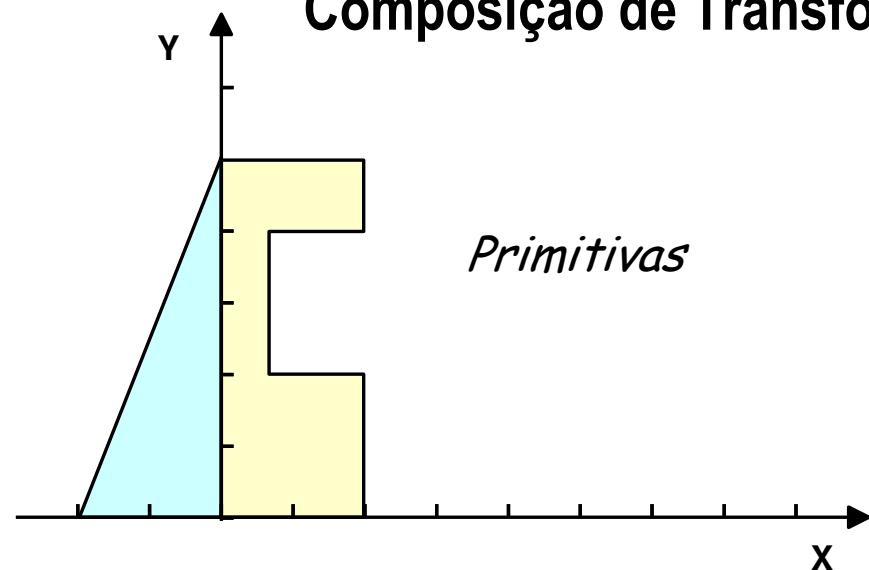
$S(K,K).R(\infty)$



Composição de Transformações: Rotação em torno de um ponto arbitrário $P(x,y)$

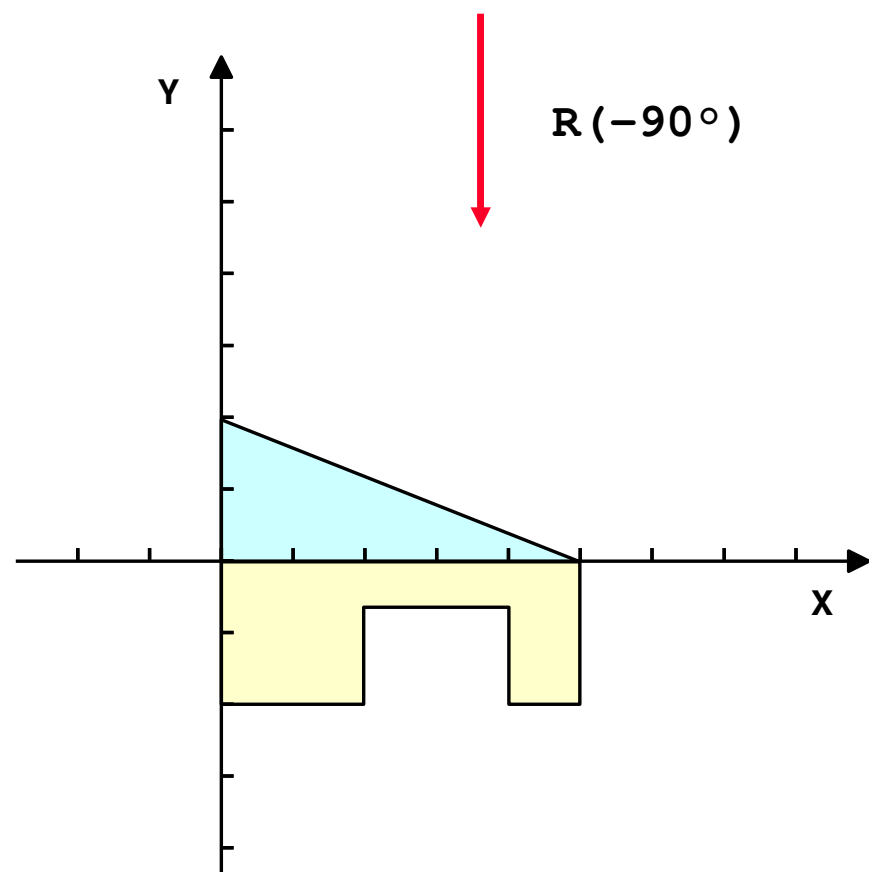


Composição de Transformações em Modelação

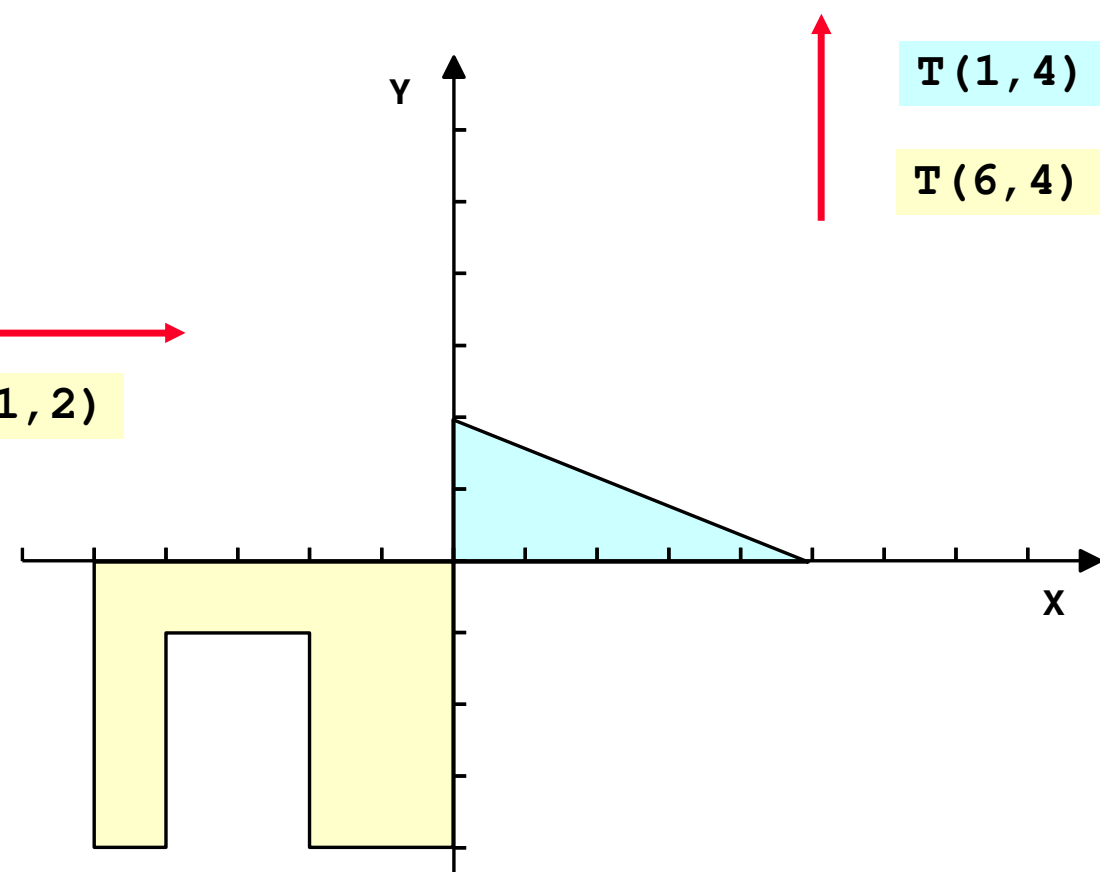


$$M1 = T(1,4).R(-90^\circ)$$

$$M2 = T(6,4).S(-1,2).R(-90^\circ)$$



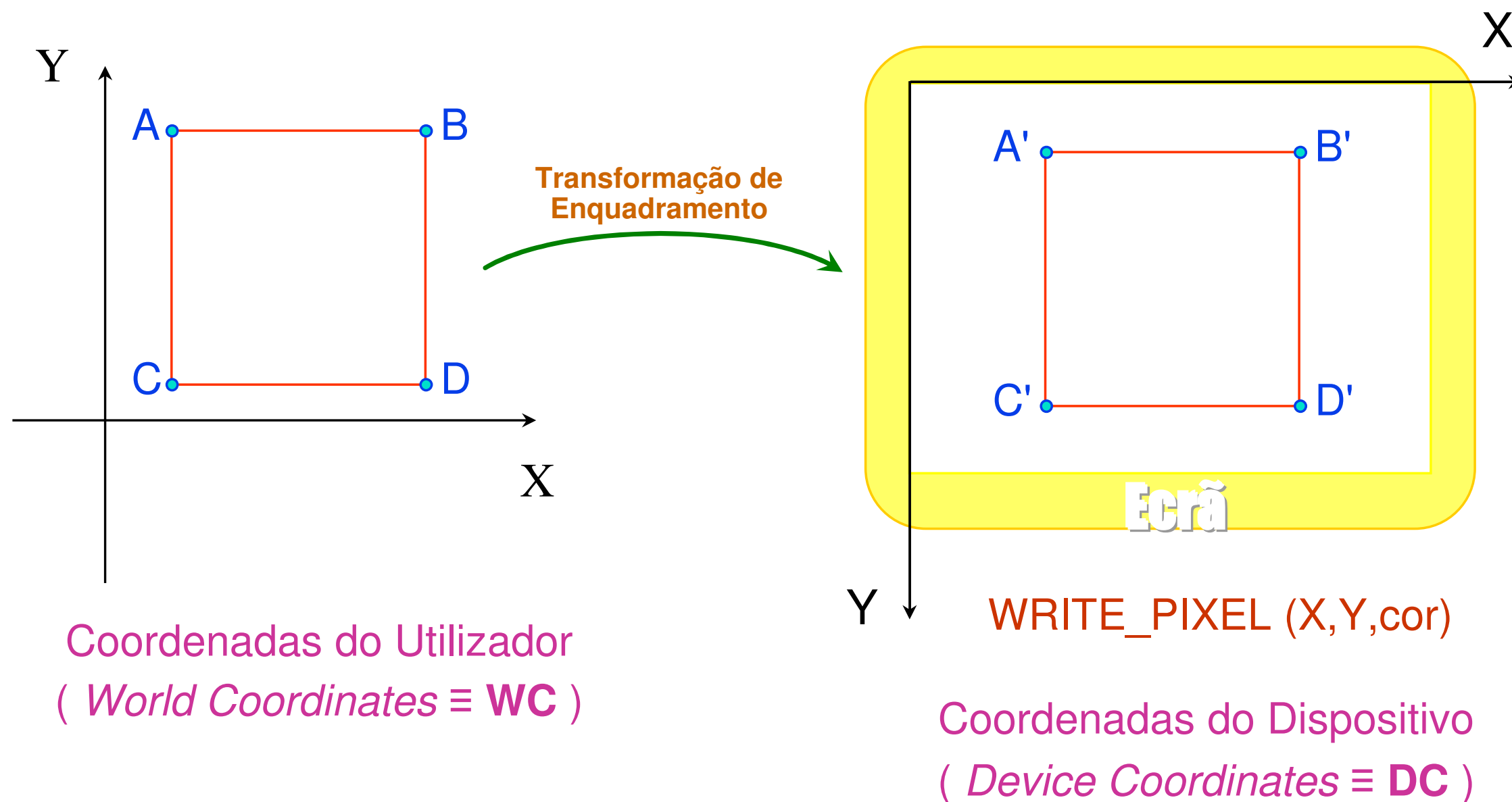
$S(-1, 2)$



Enquadramento

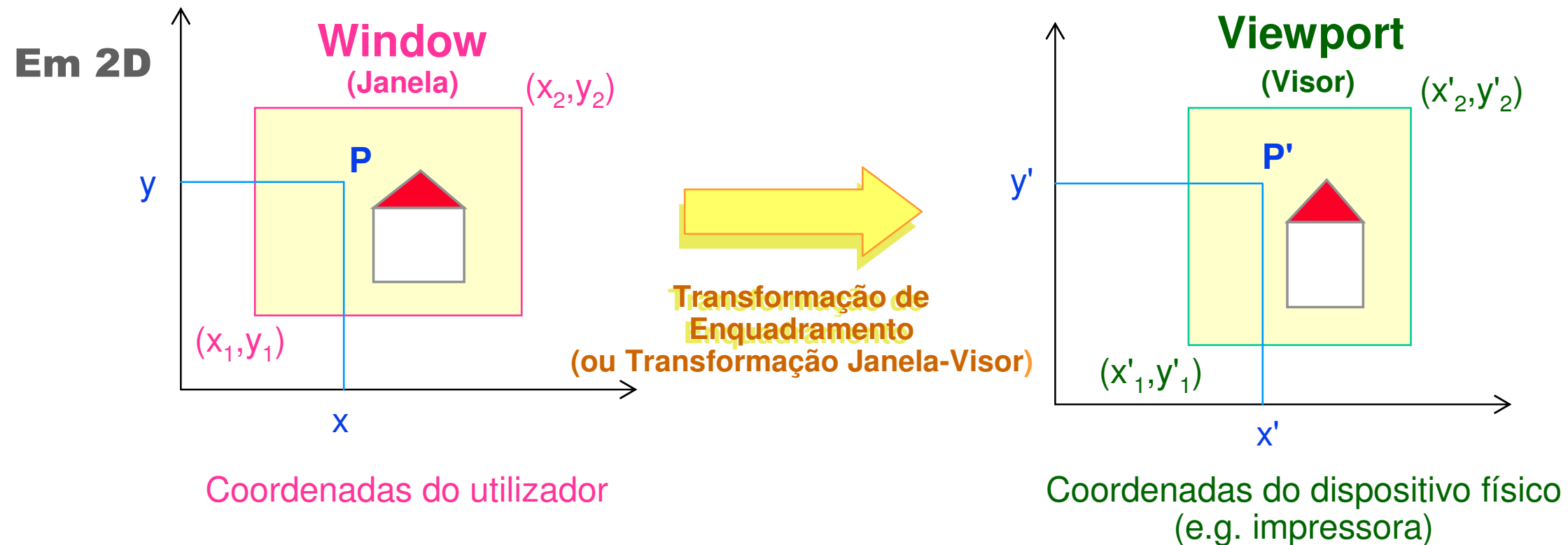
Janela/Visor

Enquadramento

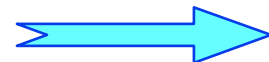


M.Próspero

Enquadramento



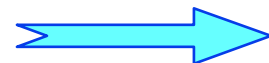
$$\frac{x_2 - x_1}{x - x_1} = \frac{x'_2 - x'_1}{x' - x'_1}$$



$$x' = (x - x_1) (x'_2 - x'_1) / (x_2 - x_1) + x'_1$$

$$x' = (x - A) \cdot B + C$$

$$\frac{y_2 - y_1}{y - y_1} = \frac{y'_2 - y'_1}{y' - y'_1}$$



$$y' = (y - y_1) (y'_2 - y'_1) / (y_2 - y_1) + y'_1$$

$$y' = (y - D) \cdot E + F$$

Pseudo-código da especificação: `janela(x1, x2, y1, y2)` `visor(x1', x2', y1', y2')`

M.Próspero

Formato ou **Relação de Aspecto** (*Aspect Ratio*)
de um retângulo:

$$a = \frac{x_{\max} - x_{\min}}{y_{\max} - y_{\min}}$$

Exemplos bem conhecidos

4:3
16:9

Condição para manter as proporções da imagem no enquadramento:

$$a_{\text{janela}} = a_{\text{visor}}$$



$$\frac{x_2 - x_1}{y_2 - y_1} = \frac{x'_2 - x'_1}{y'_2 - y'_1}$$



$$E = B$$

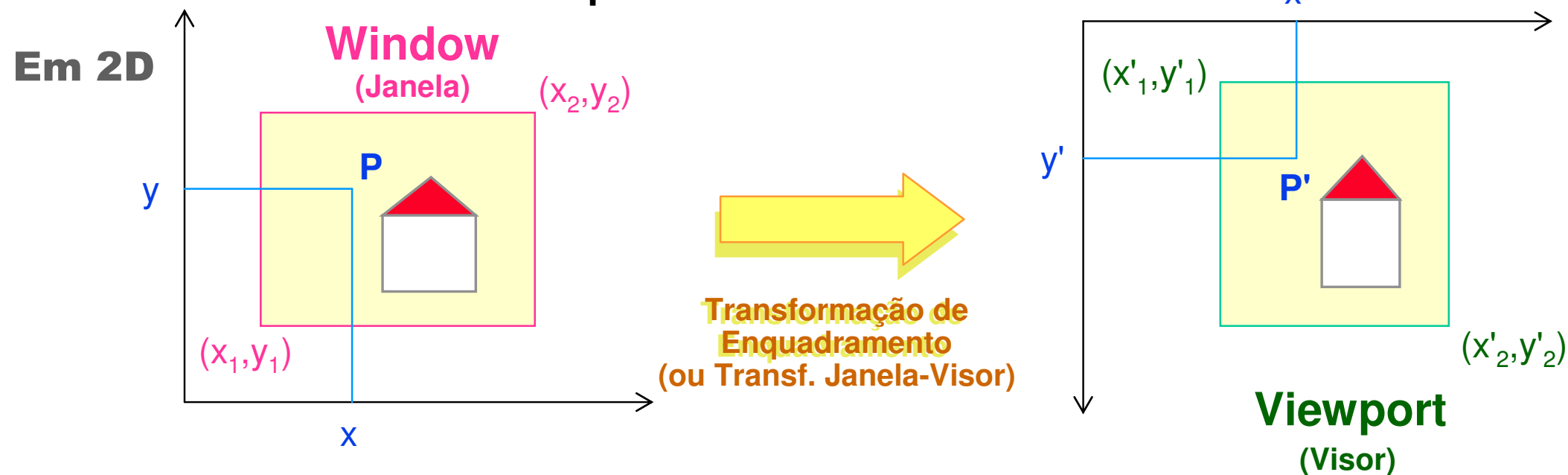
Caso contrário



distorção da imagem

Enquadramento

para coordenadas do Ecrã



$$\frac{x_2 - x_1}{x - x_1} = \frac{x'_2 - x'_1}{x' - x'_1}$$



$$x' = (x - x_1) (x'_2 - x'_1) / (x_2 - x_1) + x'_1$$

$$x' = (x - A) \cdot B + C$$

$$\frac{y_2 - y_1}{y - y_1} = \frac{y'_2 - y'_1}{y'_2 - y'}$$



$$y' = - (y - y_1) (y'_2 - y'_1) / (y_2 - y_1) + y'_2$$

$$y' = (y - D) \cdot G + H$$

$$G = -E$$

Pseudo-código da especificação: **janela**(x_1, x_2, y_1, y_2) **visor**(x'_1, x'_2, y'_1, y'_2)

M.Próspero

Especificação OpenGL de uma JANELA:

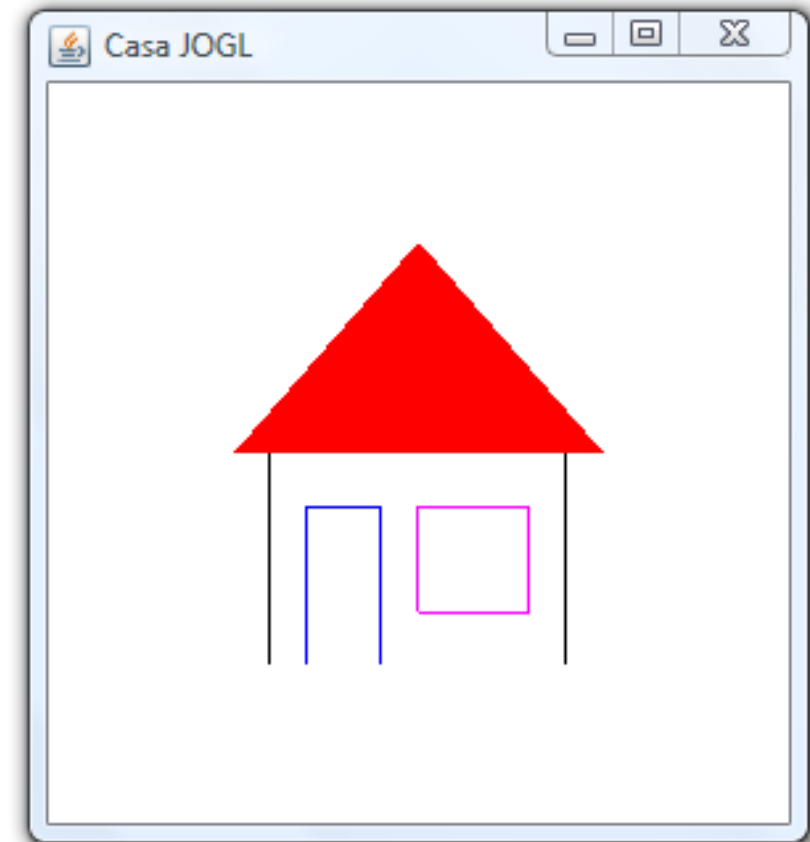
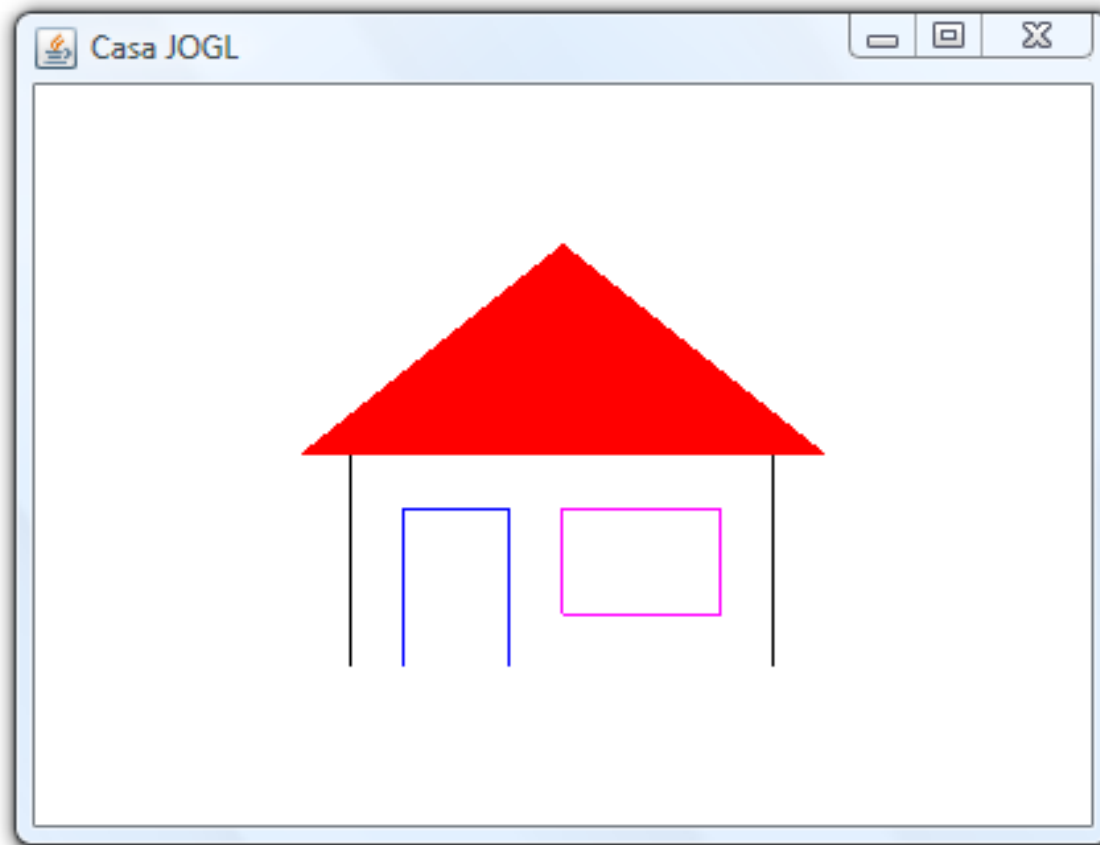
```
gluOrtho2D (0.0, 400.0, -280.0, 0.0);
```

Em WC

Especificação OpenGL de um VISOR:

```
glViewport (0, 0, 400, 280);
```

Em pixels, mas
não exatamente
em DC



janela(0, 400, -280, 0)

visor(0, 280, 0, 280) $\rightarrow a_v = 1:1$

Pseudo-código,
em WC

janela(0, 400, -280, 0)

Pseudo-código,
em DC

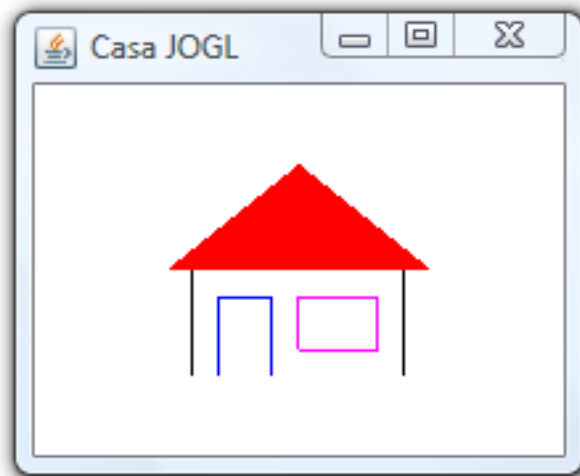
visor(0, 400, 0, 280)

$a = 10:7$

$a_j \neq a_v$

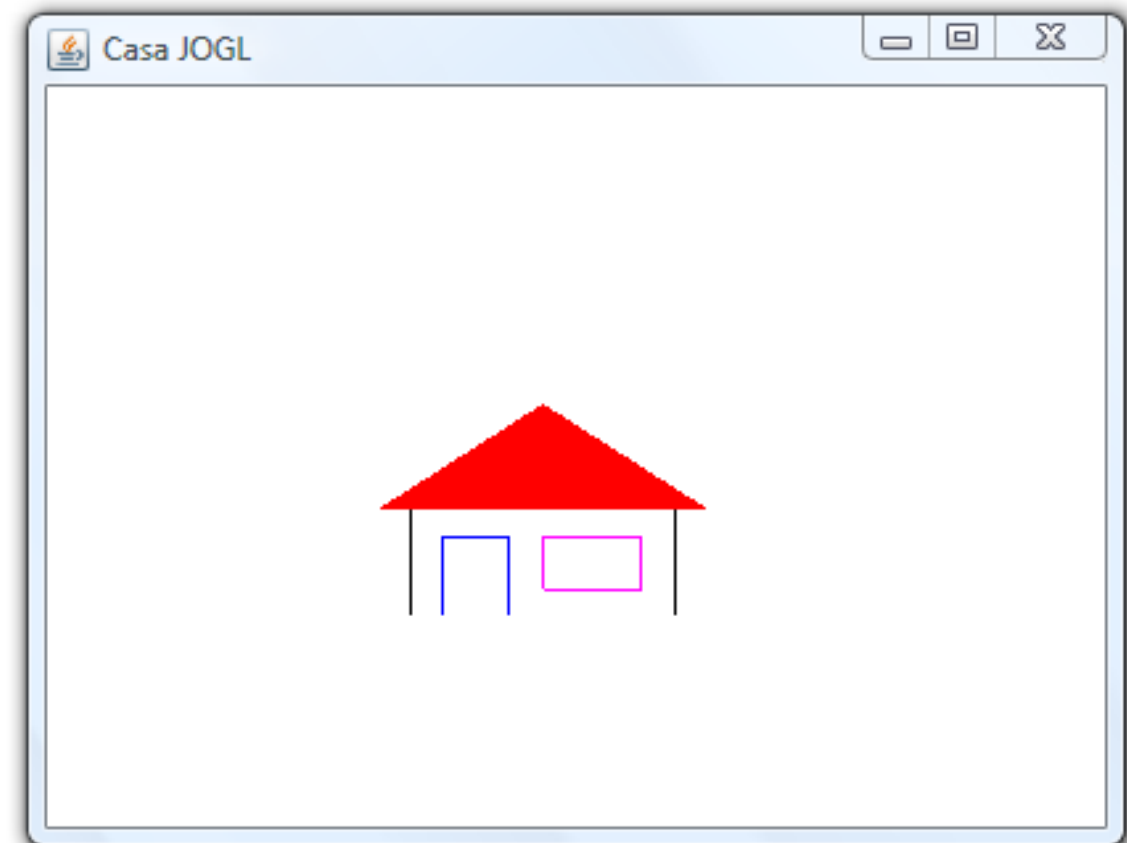
distorção da imagem

M.Próspero



`janela(0, 400, -280, 0)`
`visor(0, 200, 0, 140)`

} $a = 10:7$



`janela(-100, 540, -380, 180)` $\rightarrow a_j = 8:7$
`visor(0, 400, 0, 280)`



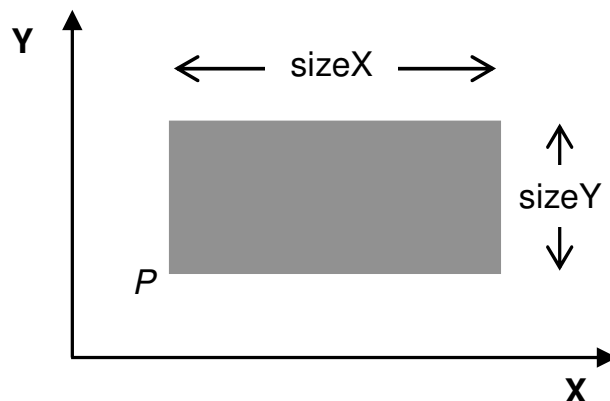
distorção da imagem

ESPECIFICAÇÃO DO VISOR NA ÁREA DE DESENHO (1)

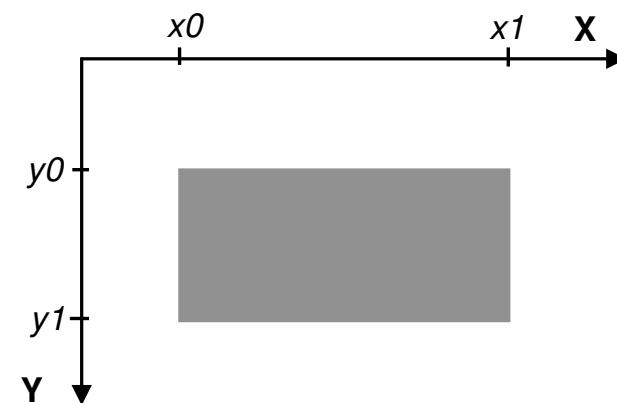
```
void glViewport(GLint x, GLint y, GLsizei width, GLsizei height);
```

Defines a pixel rectangle in the window into which the final image is mapped. The (x, y) parameter specifies the lower-left corner of the viewport, and *width* and *height* are the size of the viewport rectangle. By default, the initial viewport values are $(0, 0, \text{winWidth}, \text{winHeight})$, where *winWidth* and *winHeight* are the size of the window.

draw area, not WC



`glViewport (Px, Py, sizeX, sizeY)`

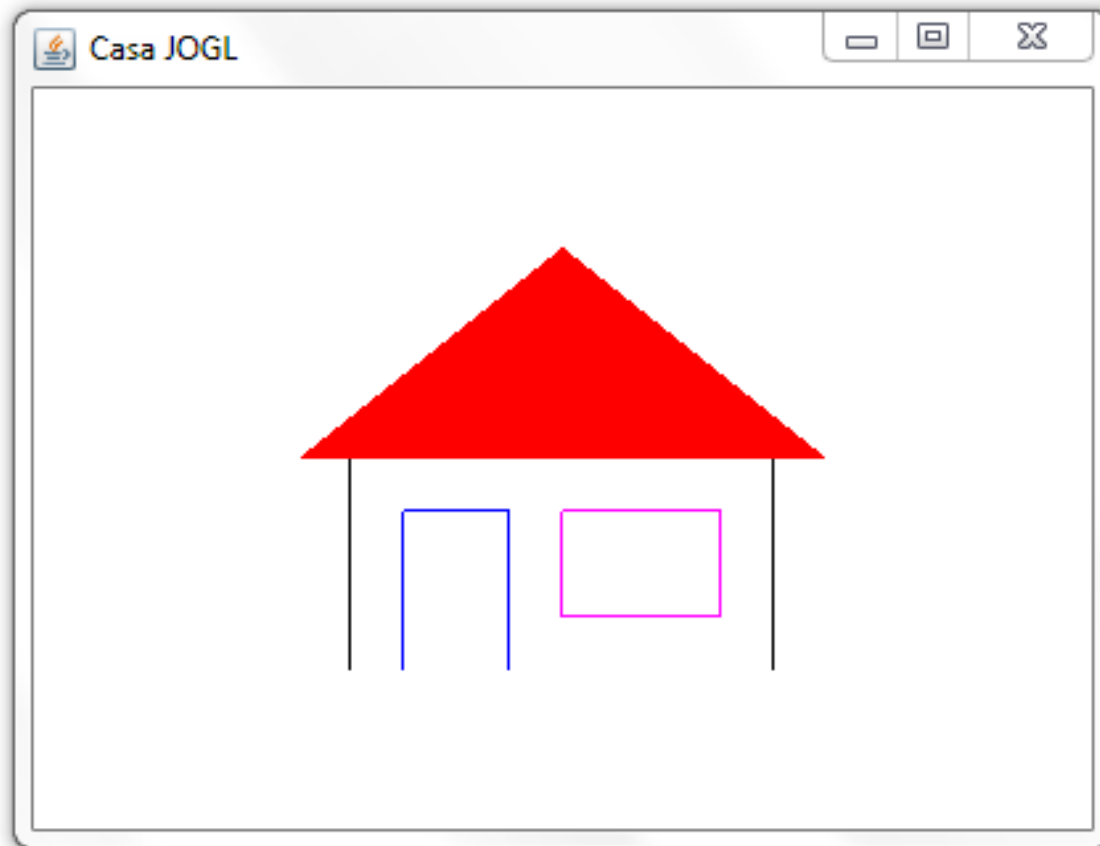


DC num ecrã

`visor (x0, x1, y0, y1)`

ESPECIFICAÇÃO DO VISOR NA ÁREA DE DESENHO (2)

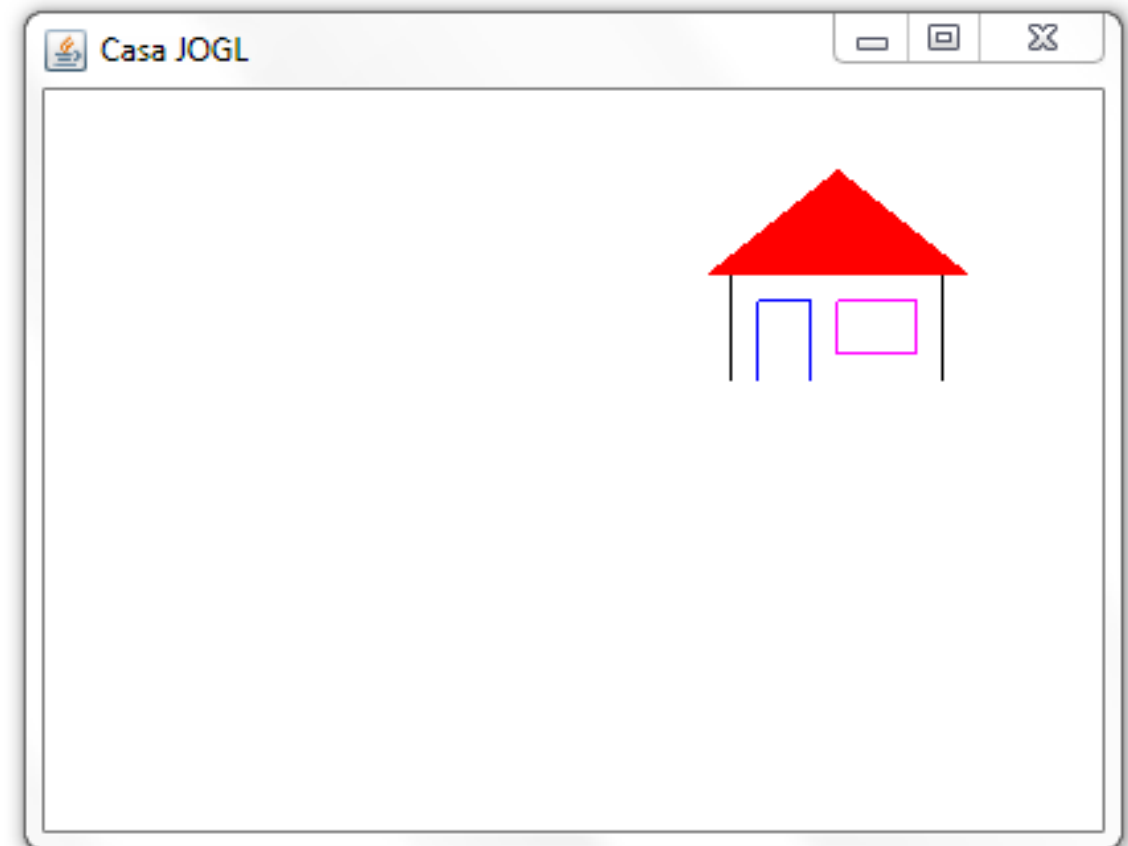
EXEMPLOS numa área de desenho (ou canvas) com 400x280 pixels



`glViewport (0, 0, 400, 280)`

OU

`visor(0, 400, 0, 280)`



`glViewport (200, 140, 200, 140)`

OU

`visor(200, 400, 0, 140)`

Enquadramento janela-visor como transformação linear

$$x' = (x - A) \cdot B + C = x \cdot B - A \cdot B + C$$

$$y' = (y - D) \cdot J + K = y \cdot J - D \cdot J + K$$

$$J = E \text{ ou } G$$

$$K = F \text{ ou } H$$

$$P' = M \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} B & 0 & C-A \cdot B \\ 0 & J & K-D \cdot J \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Pontos **P** e **P'** necessariamente em coordenadas homogêneas!

Obs.: Mais tarde se apresentará esta matriz como composição de transformações geométricas.

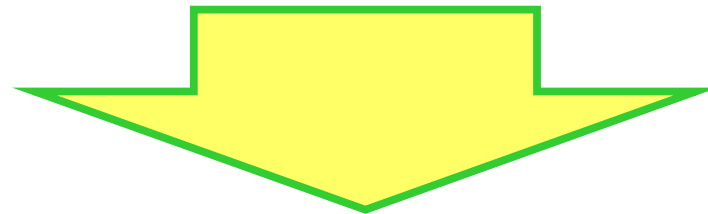
M.Próspero

Aplicação da composição de transformações geométricas no enquadramento janela-visor

$$x' = (x - A) \cdot B + C$$

$$y' = (y - D) \cdot J + K$$

$$J = E \text{ ou } G, K = F \text{ ou } H$$

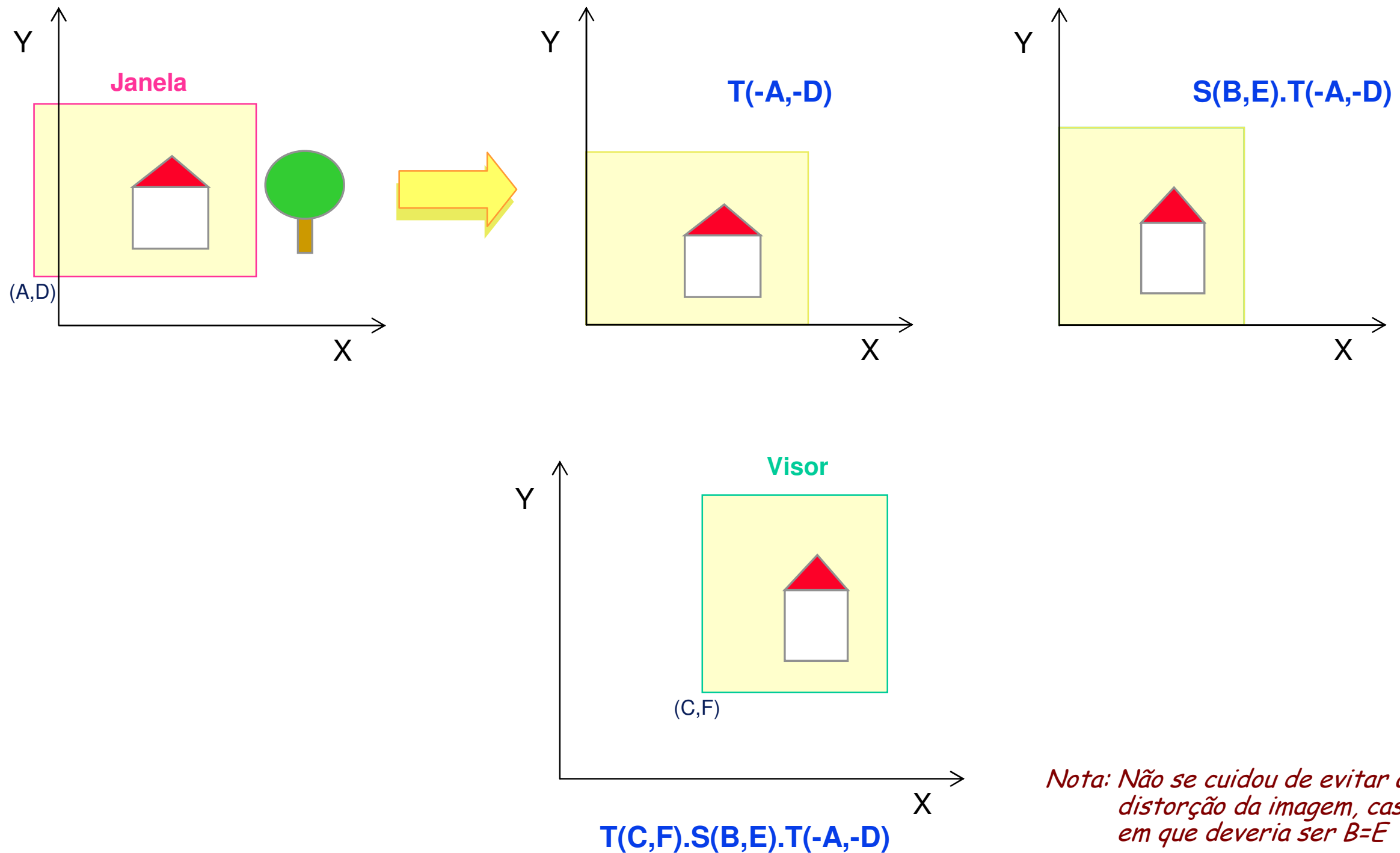


$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & C \\ 0 & 1 & K \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} B & 0 & 0 \\ 0 & J & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -A \\ 0 & 1 & -D \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

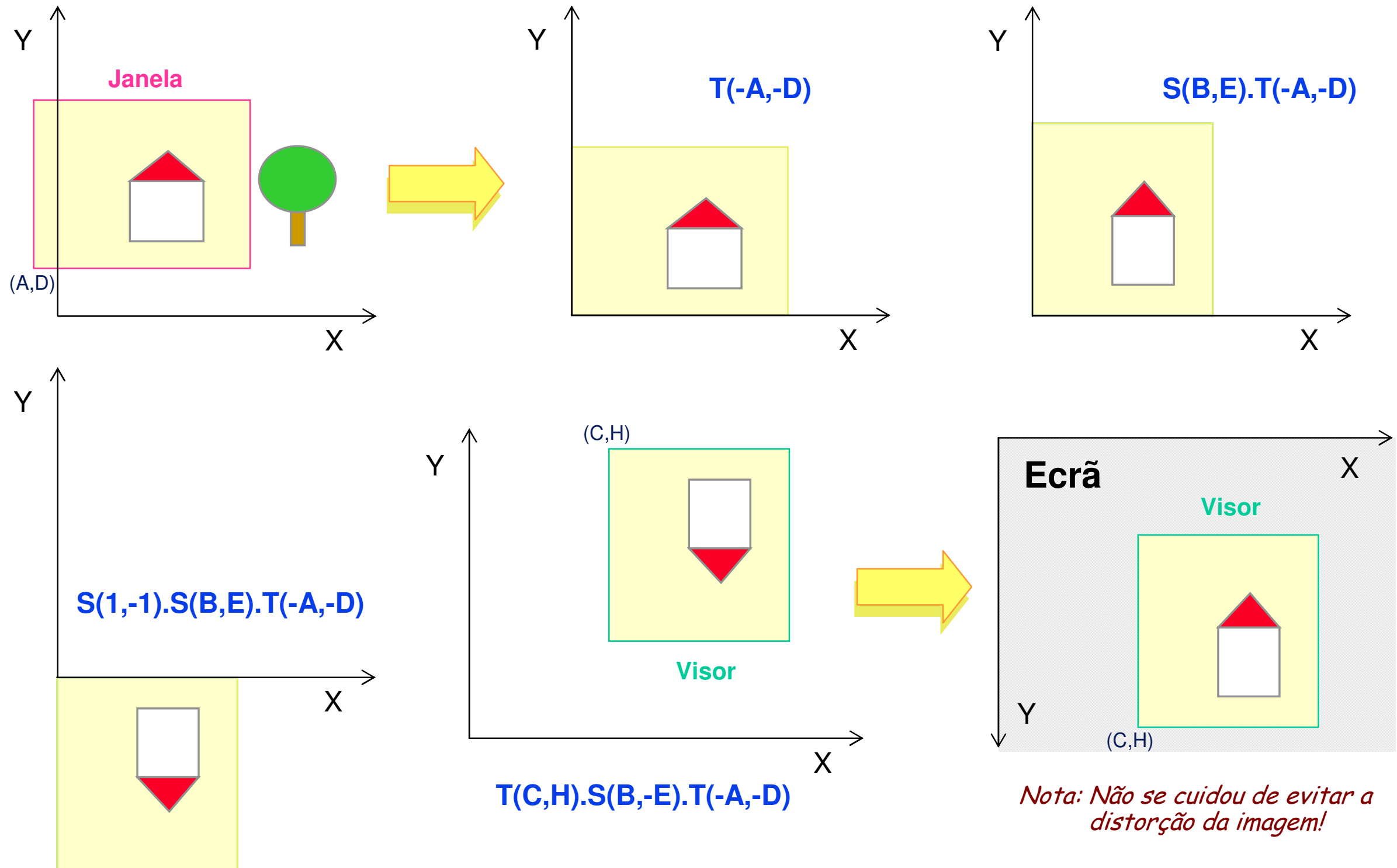
ou, numa forma mais compacta:

$$P' = T(C, K) \cdot S(B, J) \cdot T(-A, -D) \cdot P$$

Aplicação na dedução da transformação de enquadramento



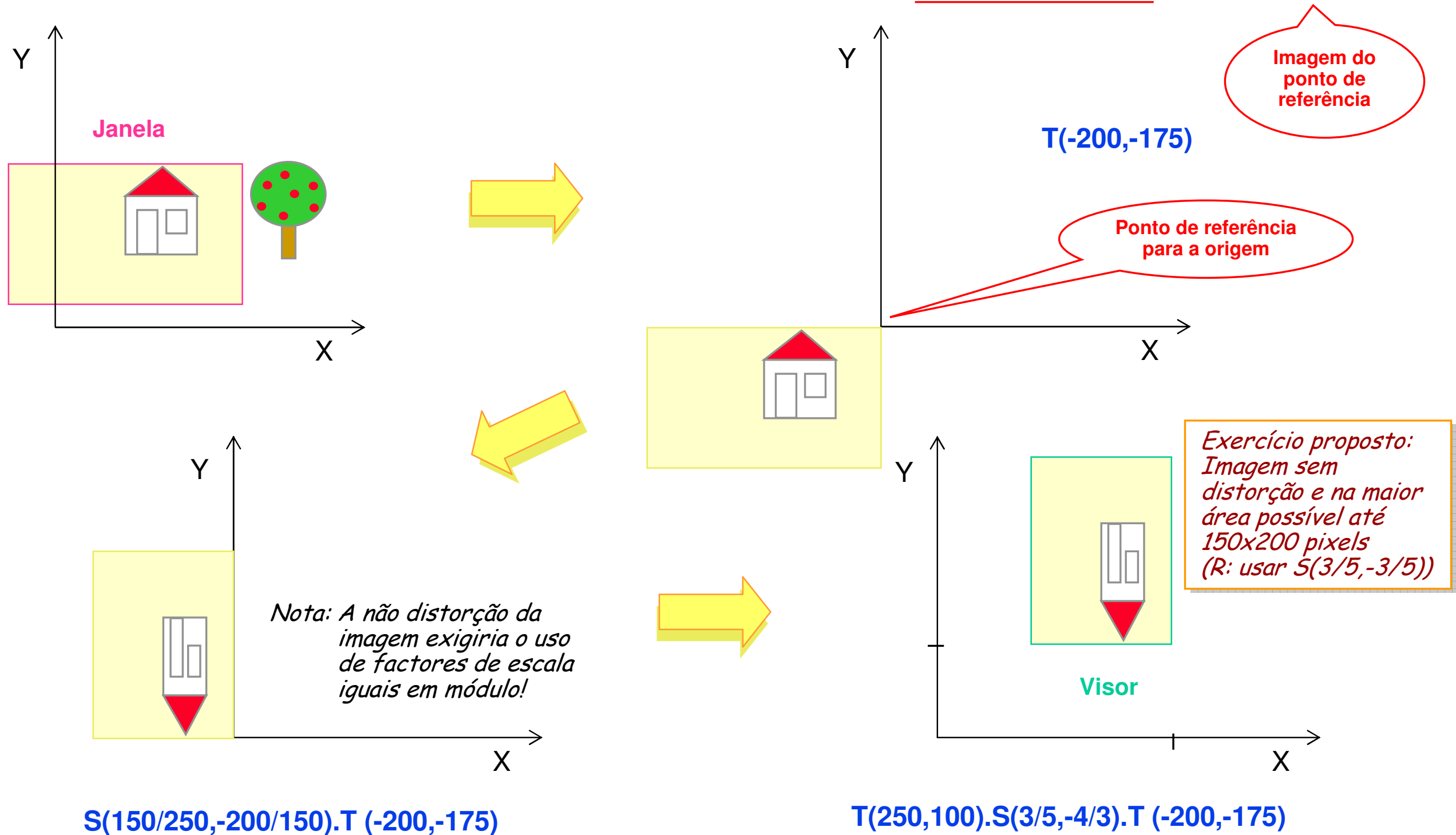
Aplicação na dedução da transformação de enquadramento (coordenadas do ecrã)



M. Próspero

Exemplo numérico (à escala) para ecrã:

Transformar a Janela $-50 \leq x \leq 200 \wedge 25 \leq y \leq 175$ num Visor de 150x200 pixels e canto superior direito no ponto (250,100)



M.Próspero