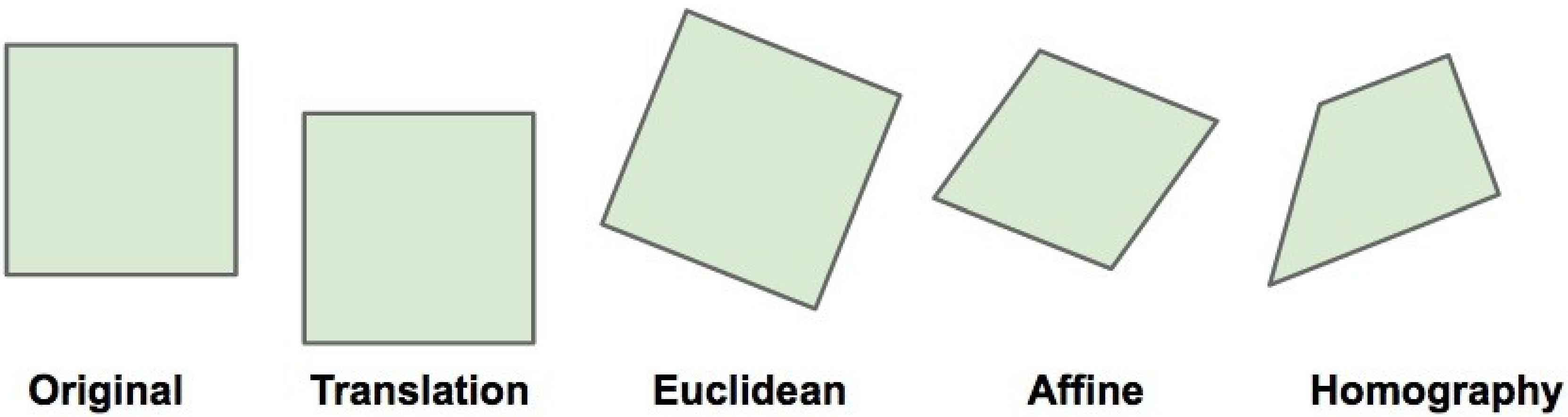
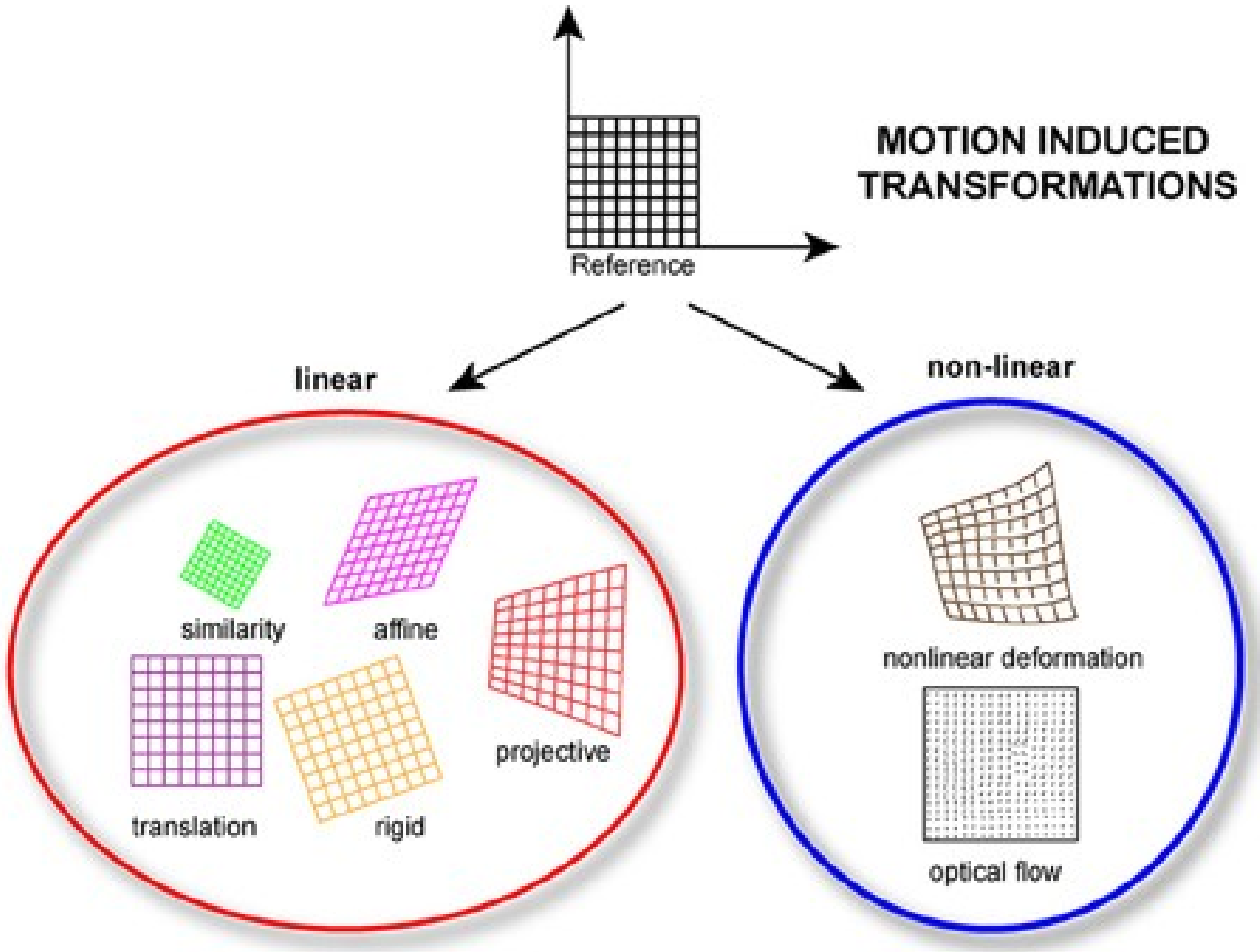
Transformações Geométricas

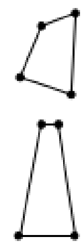
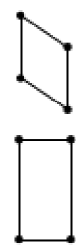
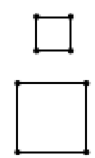
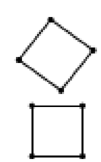
Pablo G. Cavalcanti

Motion Models



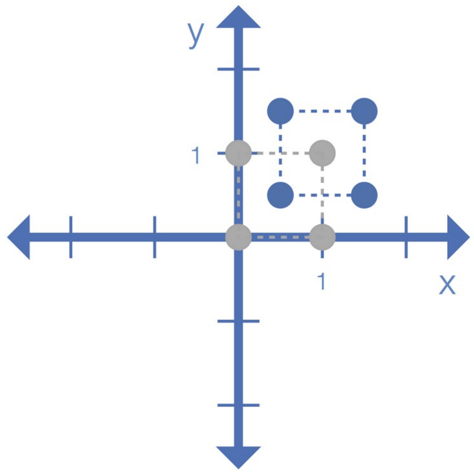
LearnOpenCV.com


MOTION INDUCED
TRANSFORMATIONS
Reference
linear
non-linear
similarity
affine
projective
translation
rigid
nonlinear deformation
optical flow

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, **I**, **J** (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

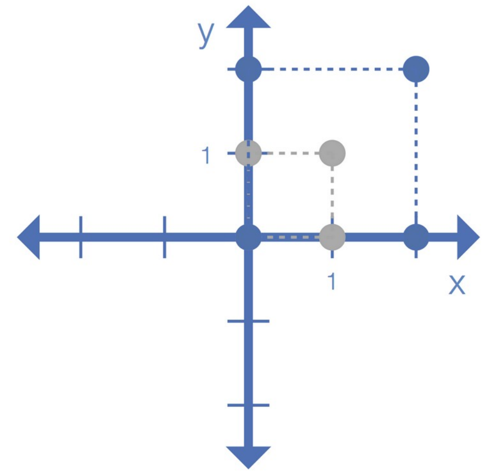
Translate

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$



Scale

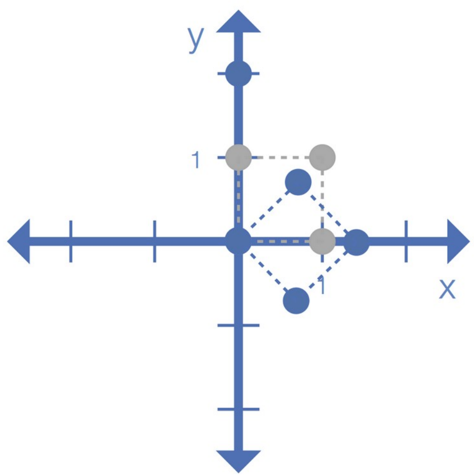
$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Rotate

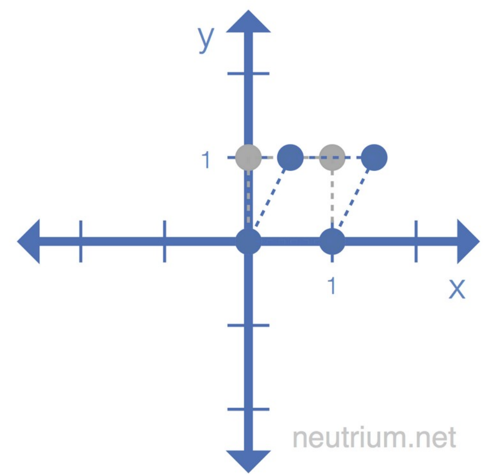
$$\begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$c = s = \sin(45^{\circ})$



Shear

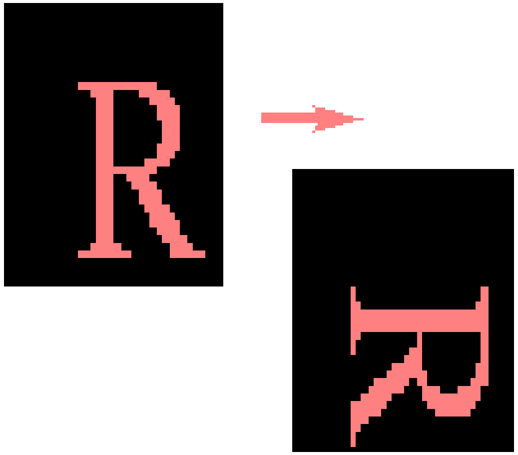
$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$



neutrium.net

Rotação

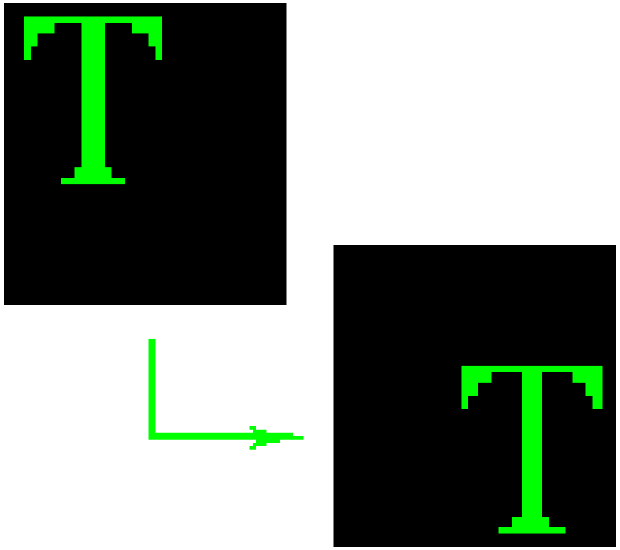
$x_2 = x_1 \cos(\alpha) + y_1 \sin(\alpha)$
$y_2 = -x_1 \sin(\alpha) + y_1 \cos(\alpha)$



Translação

$x_2 = x_1 + \Delta x$
$y_2 = y_1 + \Delta y$

Podendo $\Delta x == \Delta y$ **ou** $\Delta x != \Delta y$



Implementação

Olhar documentação e testar funções:

- resize
- rescale
- rotate

https://scikit-image.org/docs/dev/api/skimage.transform.html

https://en.wikipedia.org/wiki/Affine_transformation

Transformada Afim

Uma transformada afim se dá na forma:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Ou:

$$\mathbf{q} = \mathbf{T}\mathbf{p}$$

Onde **q** e **p** são vetores com as coordenadas dos pixels e **T** define a transformada.

Transformada Afim

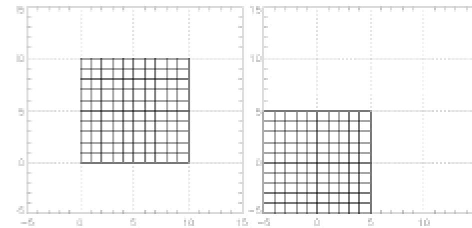
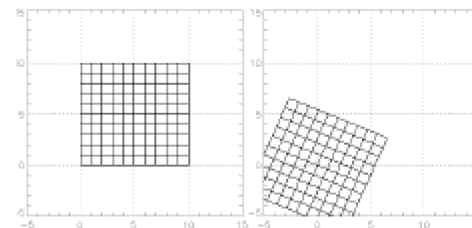
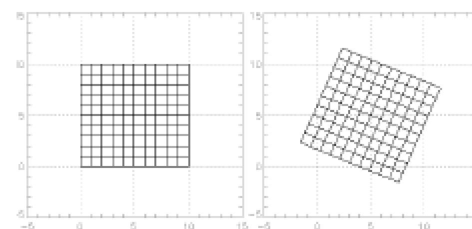
$$\mathbf{T} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$ Translation by (x_0, y_0)

$$\mathbf{T} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ Scale by s_1 and s_2

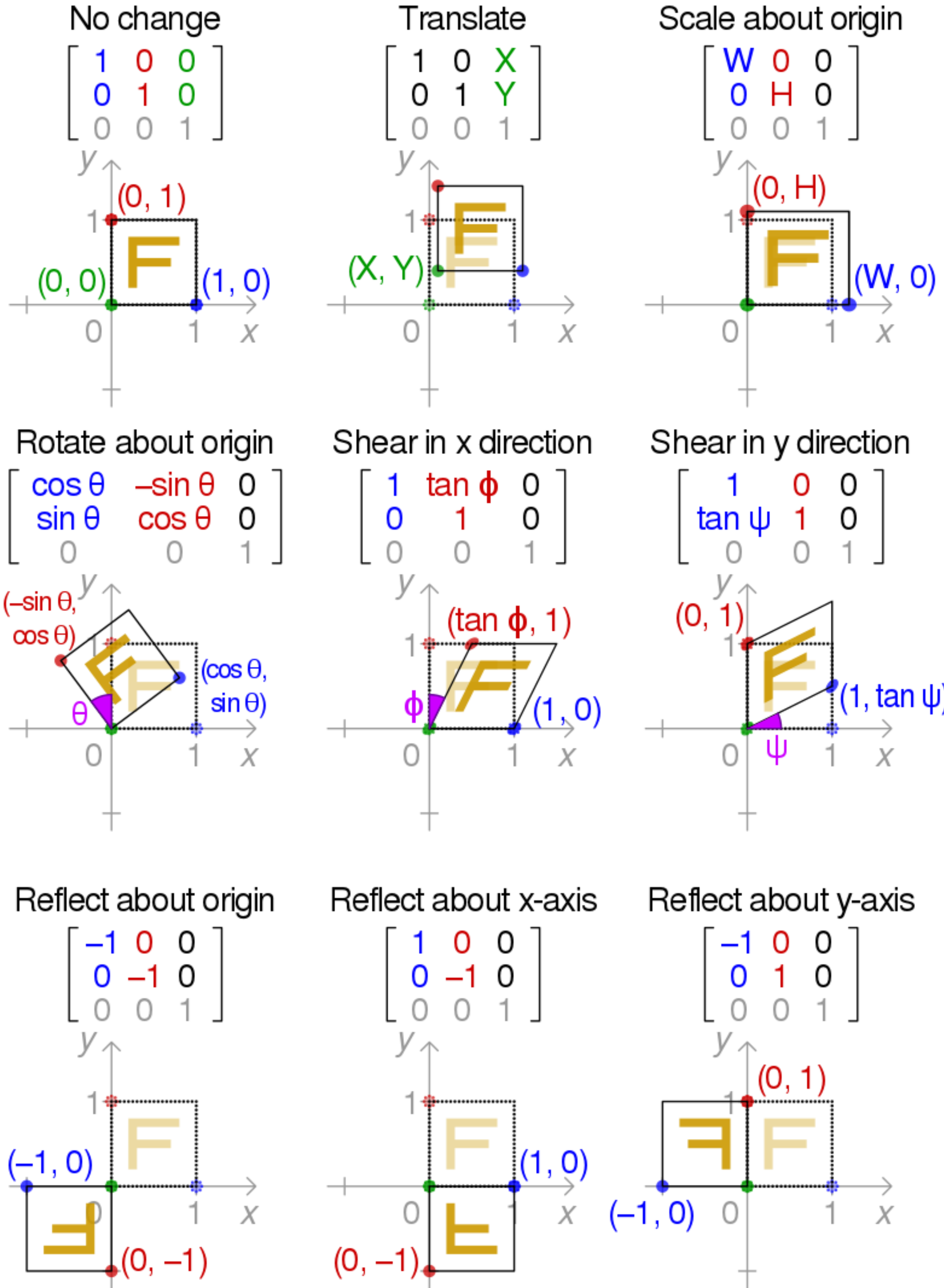
$$\mathbf{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ Rotate by θ

Transformada Afim

Exemplos:

Operation	Expression	Result
Translate to Origin	$\mathbf{T}_1 = \begin{bmatrix} 1.00 & 0.00 & -5.00 \\ 0.00 & 1.00 & -5.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	
Rotate by 23 degrees	$\mathbf{T}_2 = \begin{bmatrix} 0.92 & 0.39 & 0.00 \\ -0.39 & 0.92 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	
Translate to original location	$\mathbf{T}_3 = \begin{bmatrix} 1.00 & 0.00 & 5.00 \\ 0.00 & 1.00 & 5.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	

Se combinarmos $T = T_1 T_2 T_3$, T é capaz de aplicar as 3 operações juntas, assim como T^{-1} é a transformada inversa.


No change
1 0 0
0 1 0
0 0 1
(0, 1)
(0, 0)
(1, 0)
Translate
1 0 X
0 1 Y
0 0 1
(X, Y)
Scale about origin
W 0 0
0 H 0
0 0 1
(0, H)
(W, 0)
Rotate about origin
cos θ –sin θ 0
sin θ cos θ 0
0 0 1
(–sin θ, cos θ)
(cos θ, sin θ)
θ
Shear in x direction
1 tan ϕ 0
0 1 0
0 0 1
(tan ϕ, 1)
(1, 0)
ϕ
Shear in y direction
1 0 0
tan ψ 1 0
0 0 1
(0, 1)
(1, tan ψ)
ψ
Reflect about origin
–1 0 0
0 –1 0
0 0 1
(–1, 0)
(0, –1)
Reflect about x-axis
1 0 0
0 –1 0
0 0 1
(1, 0)
(0, –1)
Reflect about y-axis
–1 0 0
0 1 0
0 0 1
(0, 1)
(–1, 0)

Aplicação: Registro de Imagens

Image A

Image B

Aplicação: Registro de Imagens

$$\mathbf{P} = \begin{bmatrix} x_0 & x_1 & \dots & x_{n-1} \\ y_0 & y_1 & \dots & y_{n-1} \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \dots & \mathbf{p}_{n-1} \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} u_0 & u_1 & \dots & u_{n-1} \\ v_0 & v_1 & \dots & v_{n-1} \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \dots & \mathbf{q}_{n-1} \end{bmatrix}$$

Sendo P as coordenadas da imagem A e Q da imagem B, podemos:

$$\mathrm{Q} = \mathrm{HP}$$

Onde H é a transformada afim que mapeia todas coordenadas de P em Q.

Podemos obter $H = QP^{-1}$. A inversa pode ser aproximada pela pseudo-inversa:

$$H = QP^{+} = QP^{T}(PP^{T})^{-1}$$

Aplicação: Registro de Imagens

Image A

Image B

Aplicação: Registro de Imagens

Table of Matching Points					
X_a	Y_a	X_b	Y_b	X'_a	Y'_a
30.5	325.3	125.8	322.5	126.0	322.8
86.8	271.3	199.3	295.3	198.7	294.9
330.3	534.0	320.0	632.0	320.5	632.2
62.0	110.3	238.0	137.0	238.4	136.8
342.0	115.0	494.0	250.0	493.9	250.4
412.0	437.0	434.3	574.8	433.3	574.7
584.5	384.8	611.8	594.0	612.2	593.8

$$\mathbf{H} = \mathbf{Q}\mathbf{P}^{\dagger} = \begin{bmatrix} 0.92 & -0.39 & 224.17 \\ 0.39 & 0.92 & 10.93 \\ 0.00 & -0.00 & 1.00 \end{bmatrix}$$

Aplicação: Registro de Imagens

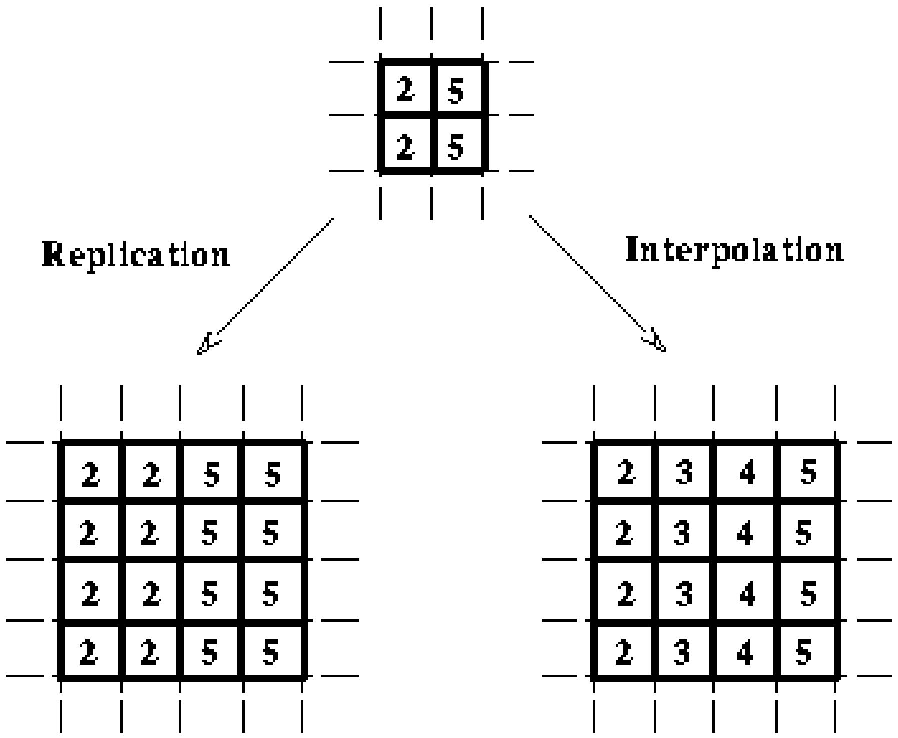
Mapped A

Original B

Escala

$x_2 = a\ x_1$
$y_2 = b\ y_1$

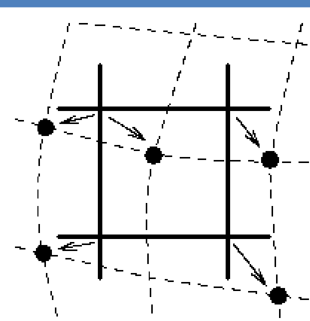
Podendo a==b ou a!=b

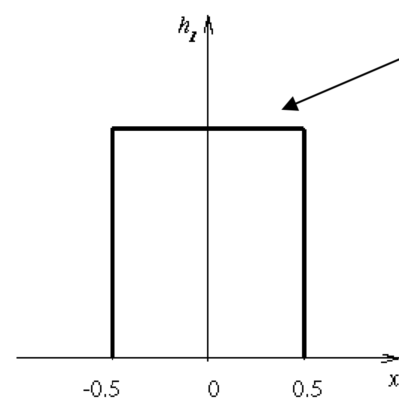


Interpolação nível de cinza

- Quando o resultado da transformada for um valor real, será preciso interpolar
- Método mais simples: *nearest neighbour*
  - atribui-se ao pixel o valor do pixel mais próximo na imagem transformada.
  - Apresenta erro de posição de, no máximo, metade do pixel que pode ser perceptível em objetos com bordas “retas”.

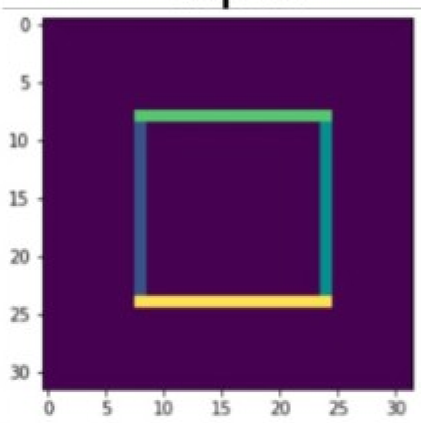
$$f_1(x,y) = g_s[\mathrm{round}(x), \mathrm{round}(y)]$$



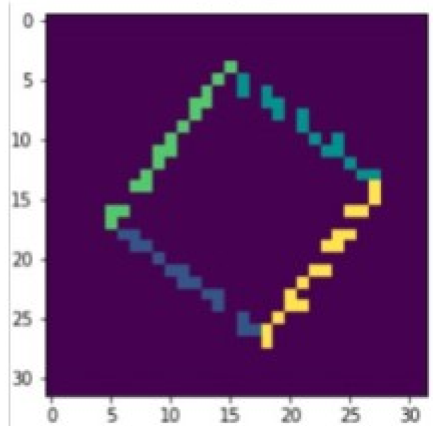


Kernel de interpolação (1D)

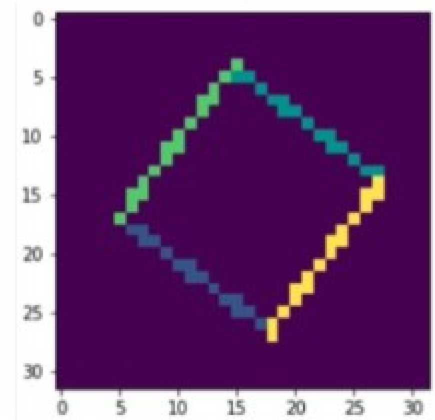
Input



AB

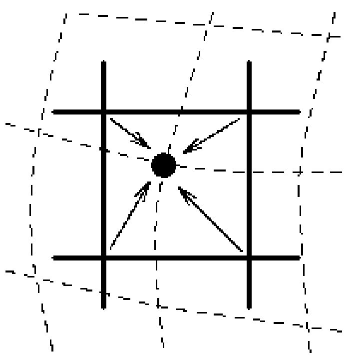


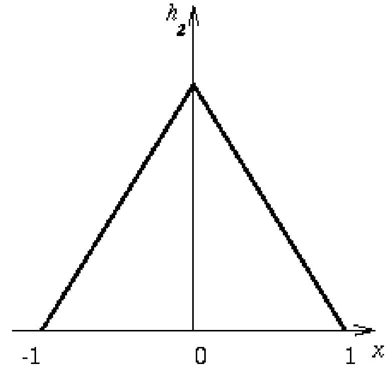
C



Interpolação linear

- Explora os 4 pontos da vizinhança de (x,y).
- A função de intensidade é linear nessa vizinhança.





$$
\begin{aligned}
f_2(x,y) = \quad & (1-a)(1-b)\, g_s(l,k) \\
& +a(1-b)\, g_s(l+1,k) \\
& +b(1-a)\, g_s(l,k+1) \\
& +ab\, g_s(l+1,k+1)
\end{aligned}
$$

$$
\begin{aligned}
l &= round\,(x), & a &= x - l \\
k &= round\,(y), & b &= y - k
\end{aligned}
$$

Fontes:

http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm

https://www.cis.rit.edu/class/simg782/lectures/lecture_02/lec782_05_02.pdf

Implementação

Exemplos de transformadas em:

https://scikit-image.org/docs/0.13.x/auto_examples/xx_applications/plot_geometric.html