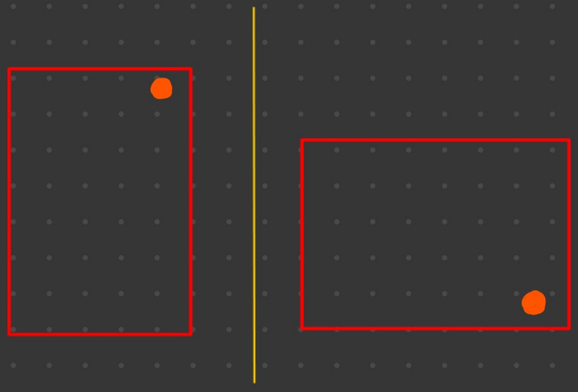
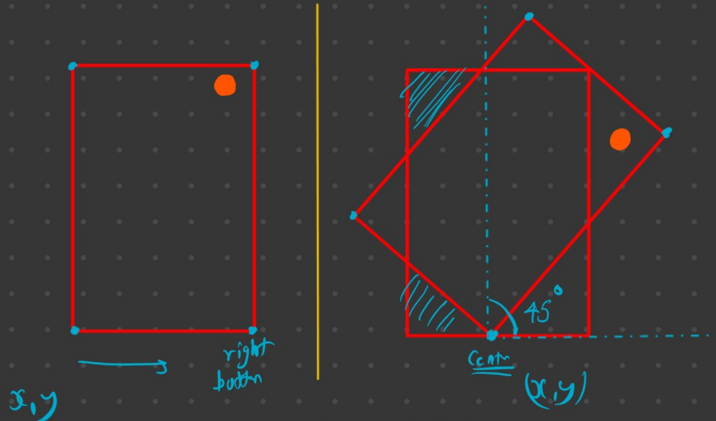
KRISHAI TECHNOLOGIES

Affine & Perspective Transformation

www.krishnaik.in



Rotate 90 degree clockwise



Rotate 45 degree clockwise

Affine Transformation

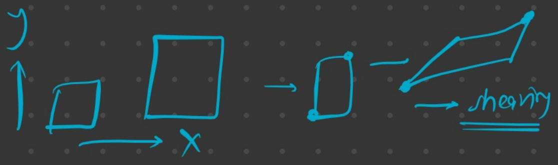
Perspective Transformation

$$(x, y) \longrightarrow$$

$$(x', y') \longrightarrow M \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2x3 matrix

$$M = \begin{bmatrix} a & b & tx \\ c & d & ty \end{bmatrix}$$



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & tx \\ c & d & ty \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a.x + b.y + tx \\ c.x + d.y + ty \end{bmatrix}$$

a,b → Controls scaling & shearing along x-axis.

c,d → Controls scaling & Shearing along Y-axis.

tx, ty → Controls translation components.

Scaling

Set a and d to be the scaling factors along the x & y axes. Others should be set to zero.

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

Rotation

$a = \cos(\theta)$ $b = (-\sin\theta)$

$c = \sin(\theta)$ $d = (\cos\theta)$

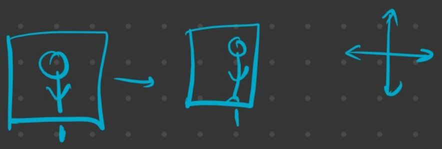
if $\theta = 90$, $a = 0$, $b = -1$, $c = 1$, $d = 0$, tx & $ty = 0$

Shearing

Shearing along x-axis (horizontal shear) → $\boxed{b \,!= 0}$

Shearing along y-axis (vertical shear) → $\boxed{c \,!= 0}$

Keep a, d same & set either x or y axis.



Translation

→ 50 px Right
→ 100 px Down

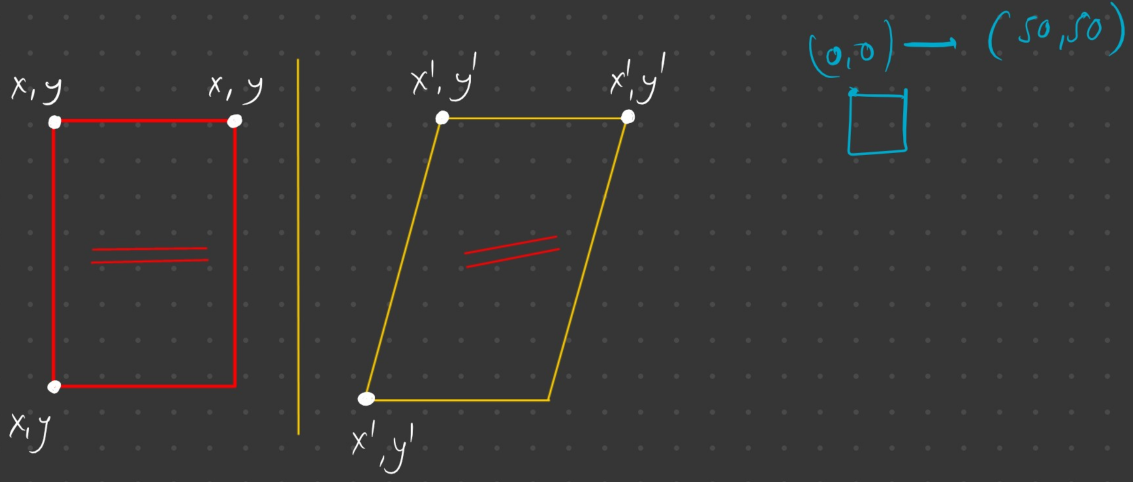
T_x, T_y move the image along x and y axes without changing shape.

$$\begin{bmatrix} 1 & 0 & 50 \\ 0 & 1 & 100 \end{bmatrix}$$

Affine Transformation

linear mapping method that preserves points, straight lines & planes.

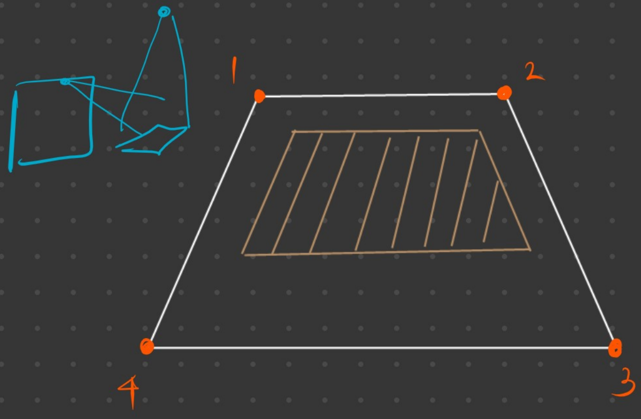
→ Straight lines remains straight
→ parallelism preserved.
→ It applies rotation, scaling, Translation & shearing.



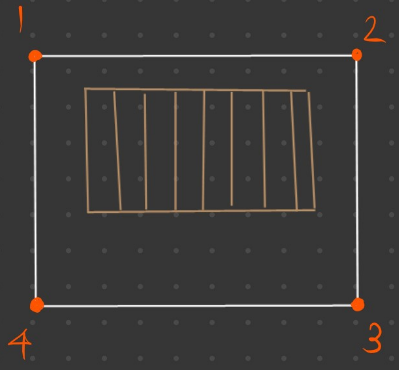
It uses 3x2 matrix (M) to calculate new pixel positions based on the linear relationship between three points in the input image and their corresponding points in the output image.

Perspective Transformation

Perspective transformation maps points from one quadrilateral in the input image to another quadrilateral in the output image.



Parking lot from top view (CCTV camera)



Perspective changed (Drone view)

→ Simulate the effect of looking at the image from a different angle.

→ Straight lines are preserved, but parallelism is not.