Lecture 5-1 Geometry Operations and interpolation

Yuyao Zhang, Xiran Cai PhD

zhangyy8@shanghaitech.edu.cn caixr@shanghaitech.edu.cn

SIST Building 2 302-F/302-C

Course piazza link： piazza.com/shanghaitech.edu.cn/spring2021/cs270spring2021

Outline

- Spatial Operations
  - Affine transform （仿射变换）
  - Projective transform
- Image interpolation
  - Nearest-neighbor interpolation
  - Linear & bi-linear interpolation

Geometric operations

- Geometric

$$\boldsymbol{J(x,y) = I\left(T(x,y)\right);}$$

- Point operation

$$\boldsymbol{J(x,y) = T\left(I(x,y)\right);}$$

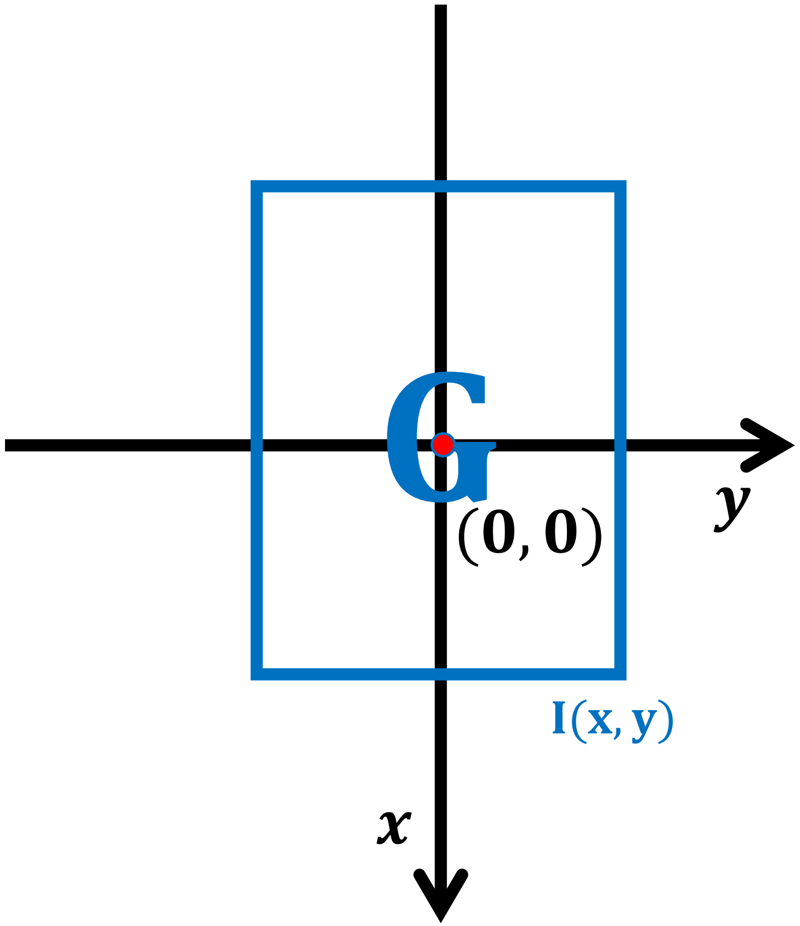
Shift translation (Affine)

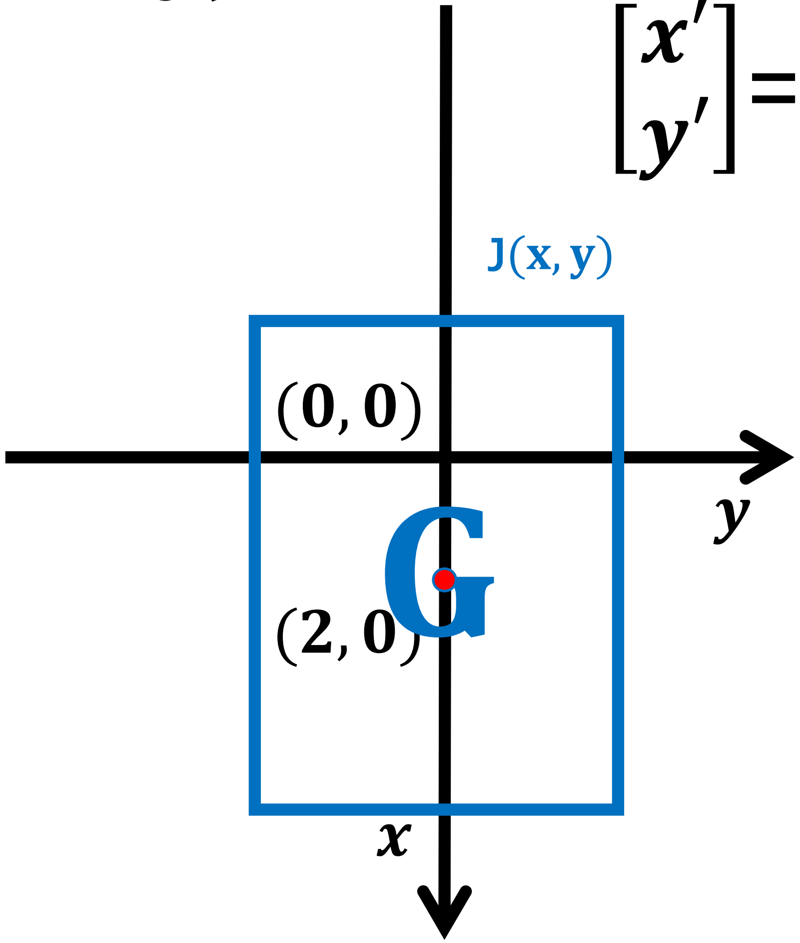
- Translate downwards by 2 pixels.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x-2 \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$\mathrm{J}(\mathbf{x},\mathbf{y}) = \mathbf{I}(\mathbf{x}-\mathbf{2},\mathbf{y})$$





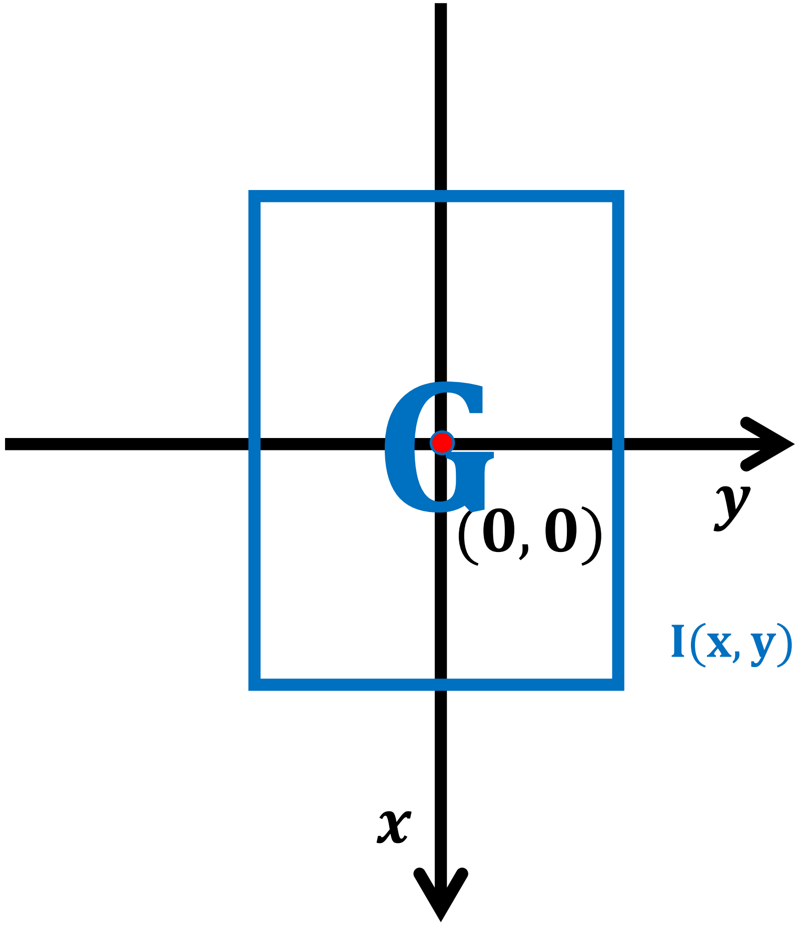
Scaling and translation

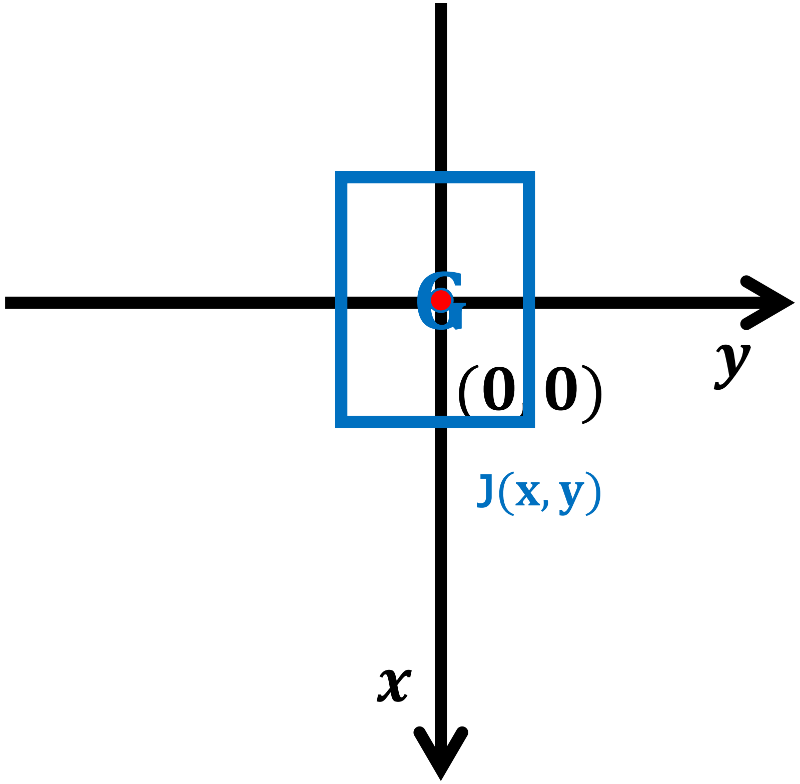
- Scale x and y by a factor of 2.

$$\mathrm{J}(\mathbf{x},\mathbf{y}) = \mathbf{I}(2*\mathbf{x}, 2*\mathbf{y})$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2*x \\ 2*y \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

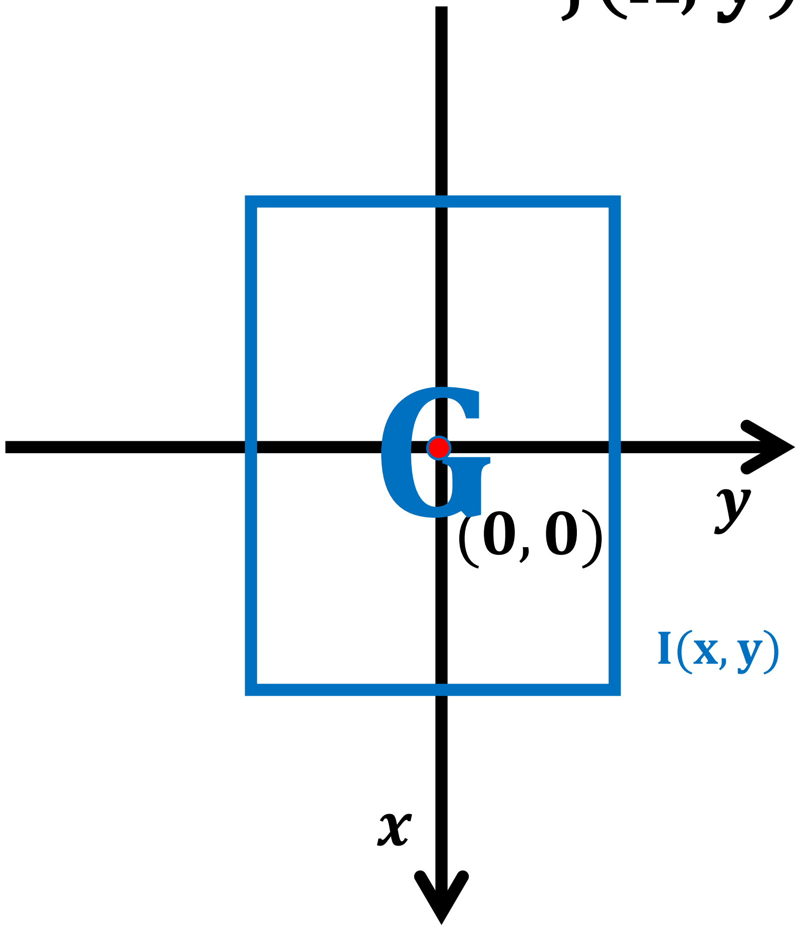


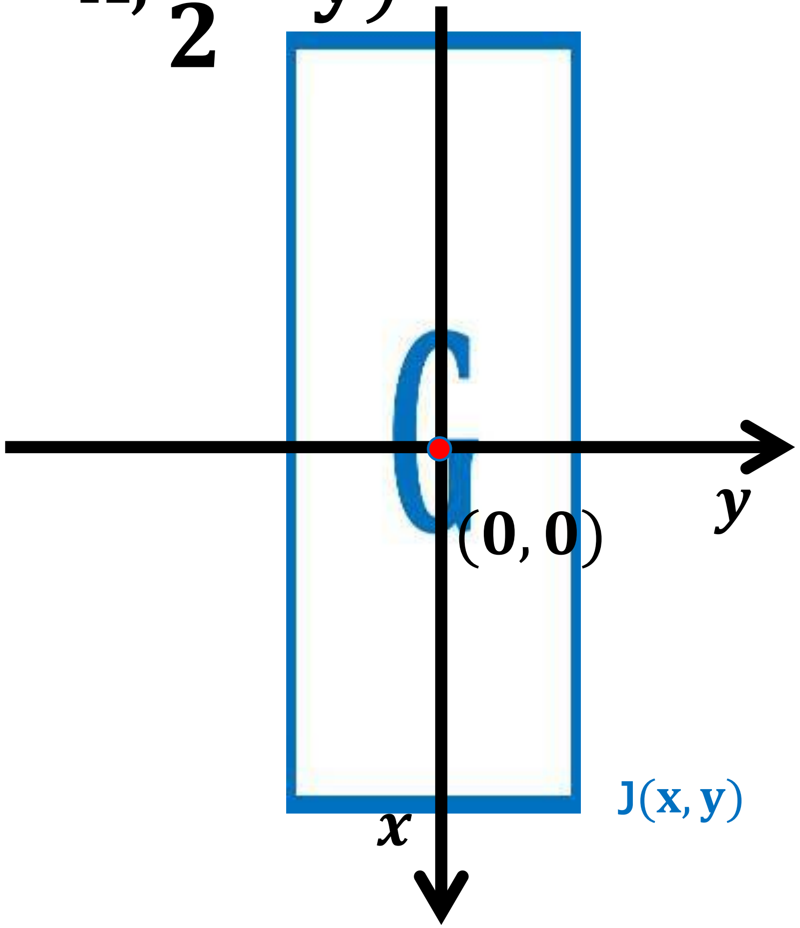


Scaling and translation

- Scale x and y by a factor of 2 and 1/2, respectively.

$$\mathrm{J}(\mathbf{x},\mathbf{y}) = \mathbf{I}(2 * \mathbf{x}, \frac{1}{2} * \mathbf{y})$$

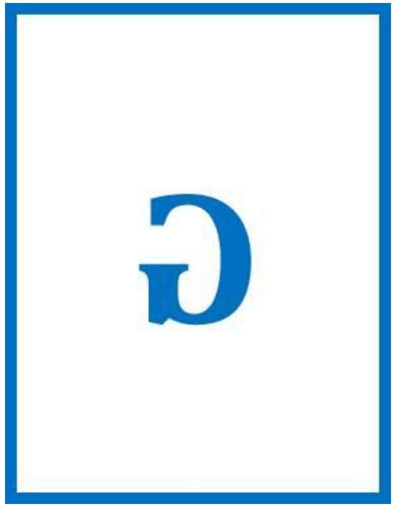




Flip translation

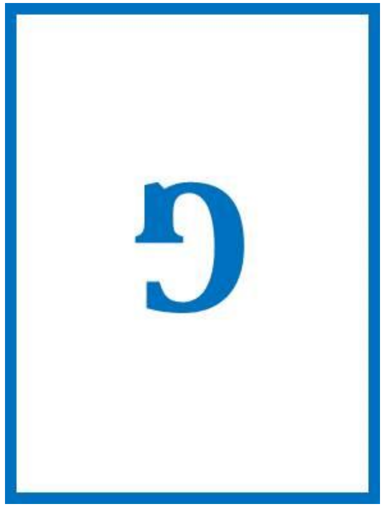
- $J(\mathbf{x}, \mathbf{y}) = \mathbf{I}(\mathbf{x}, -\mathbf{y})$
- $J(\mathbf{x}, \mathbf{y}) = \mathbf{I}(-\mathbf{x}, -\mathbf{y})$

$\mathbf{I}(\mathbf{x}, \mathbf{y})$



$\mathbf{J}(\mathbf{x}, \mathbf{y})$

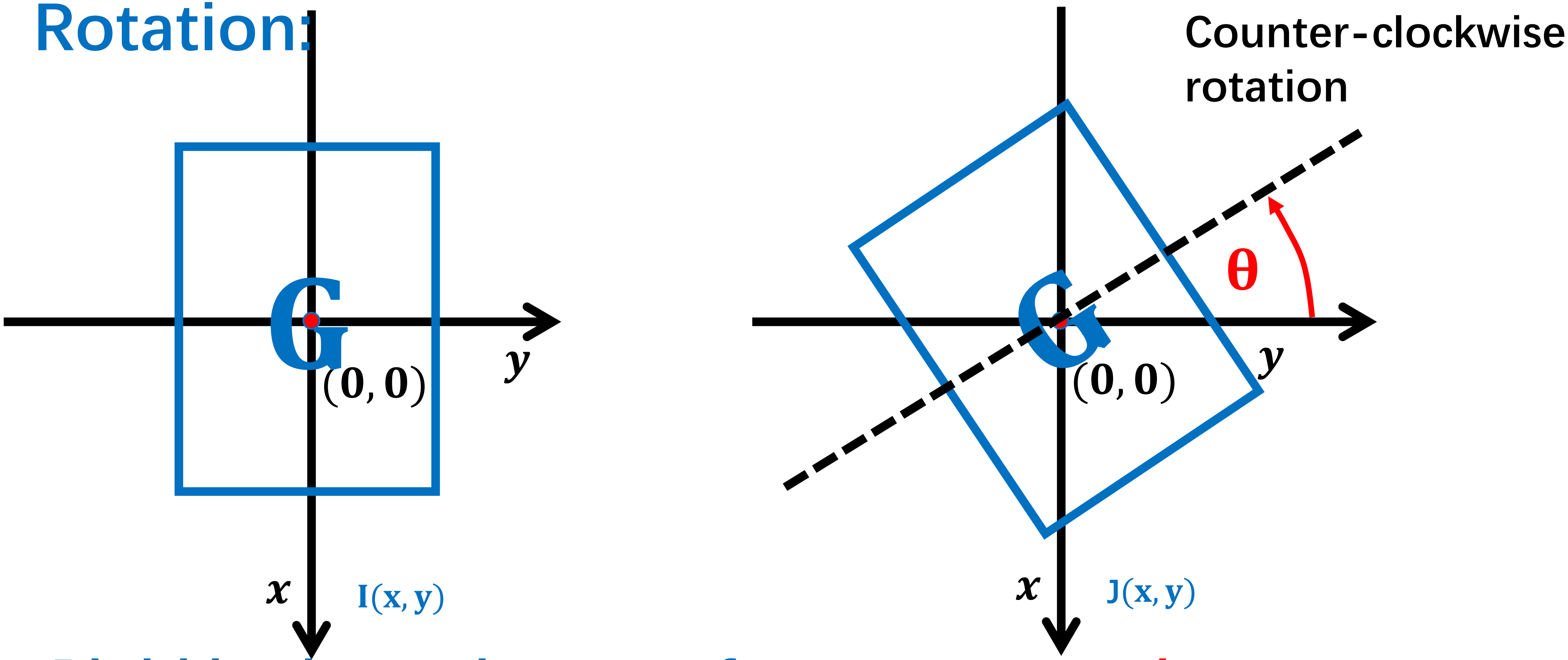
$\mathbf{I}(\mathbf{x}, \mathbf{y})$



$\mathbf{J}(\mathbf{x}, \mathbf{y})$

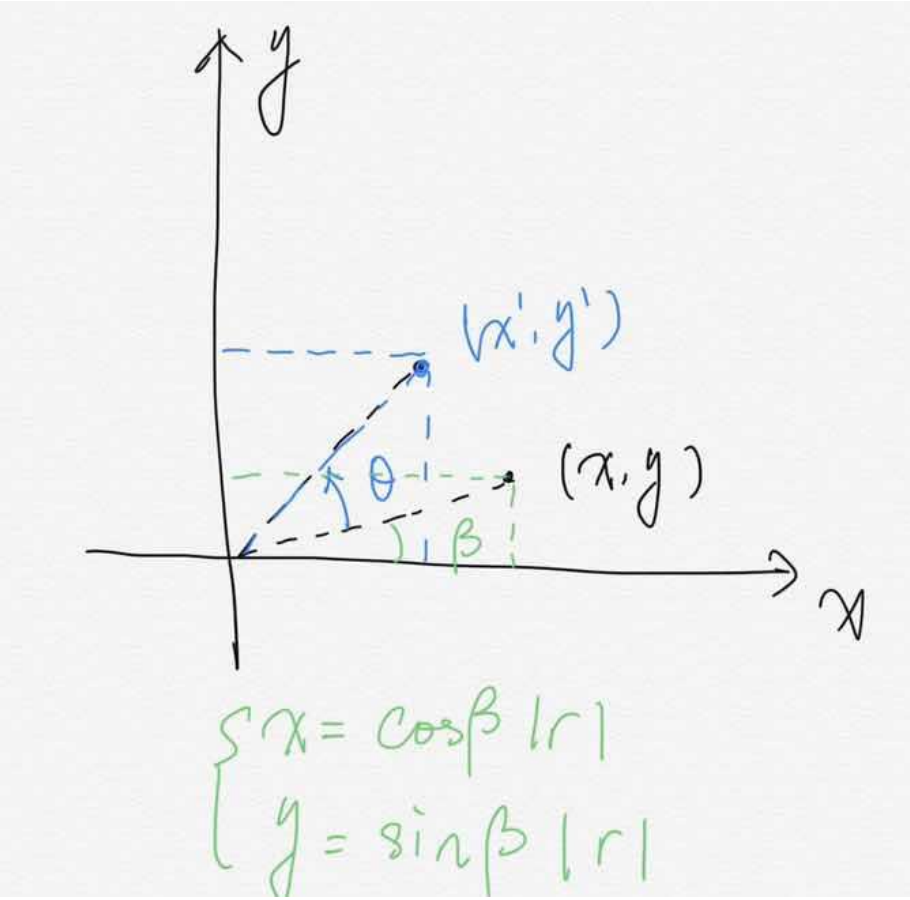
Rotation Transform

- Rotation:



Rigid body motion transform: preserve shapes and angles.

Rotation Transform



$$x' = \cos(\theta+\beta)\,|r| = (\cos\theta\cos\beta - \sin\theta\sin\beta)\ |r|$$

$$y' = \sin(\theta+\beta)\,|r| = (\sin\theta\cos\beta + \cos\theta\sin\beta)\ |r|$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\text{Rotation matrix}} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Linear Transform

- It is common for scale + rotate + shift to be considered into a 2D linear transformation.

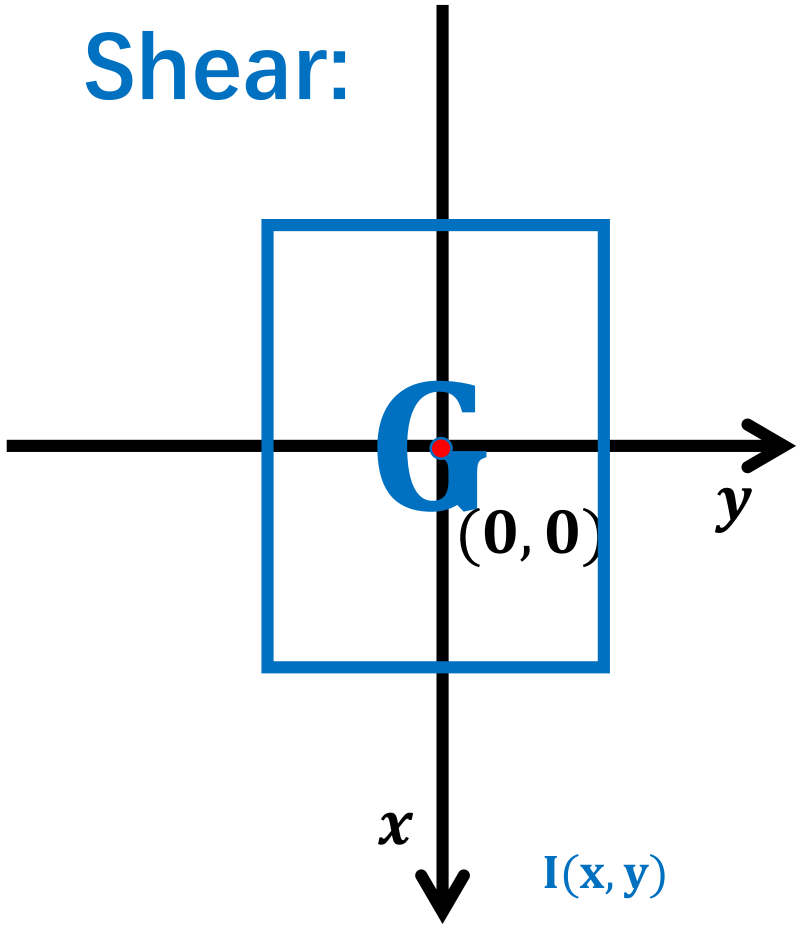
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} shift_x \\ shift_y \end{bmatrix}$$

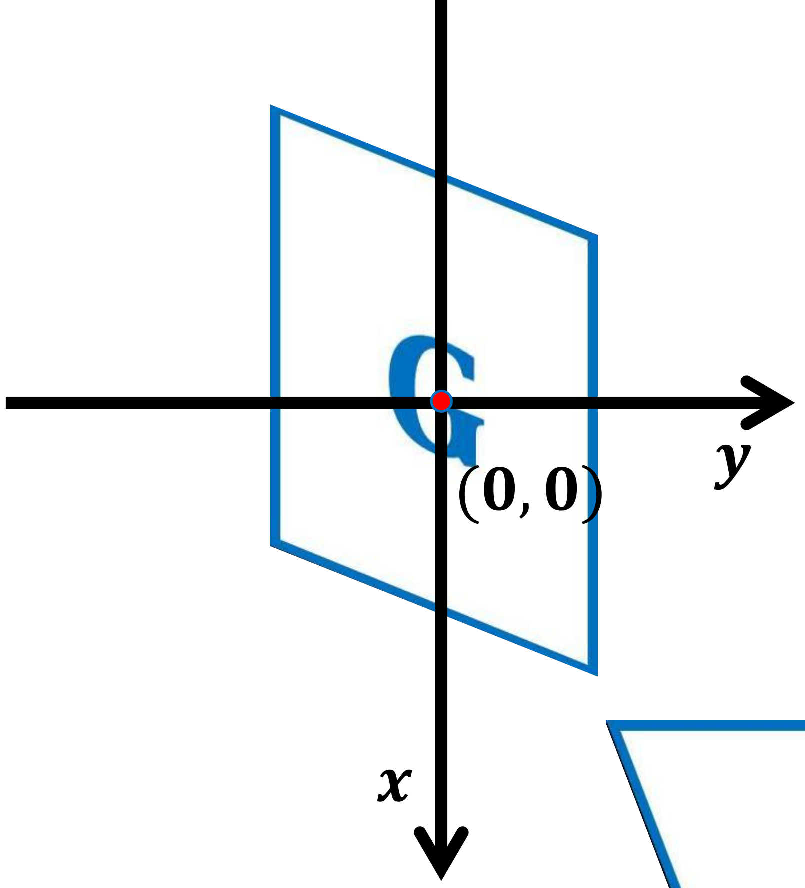
$\mathbf{J(x, y)}$ $\mathbf{I(x, y)}$

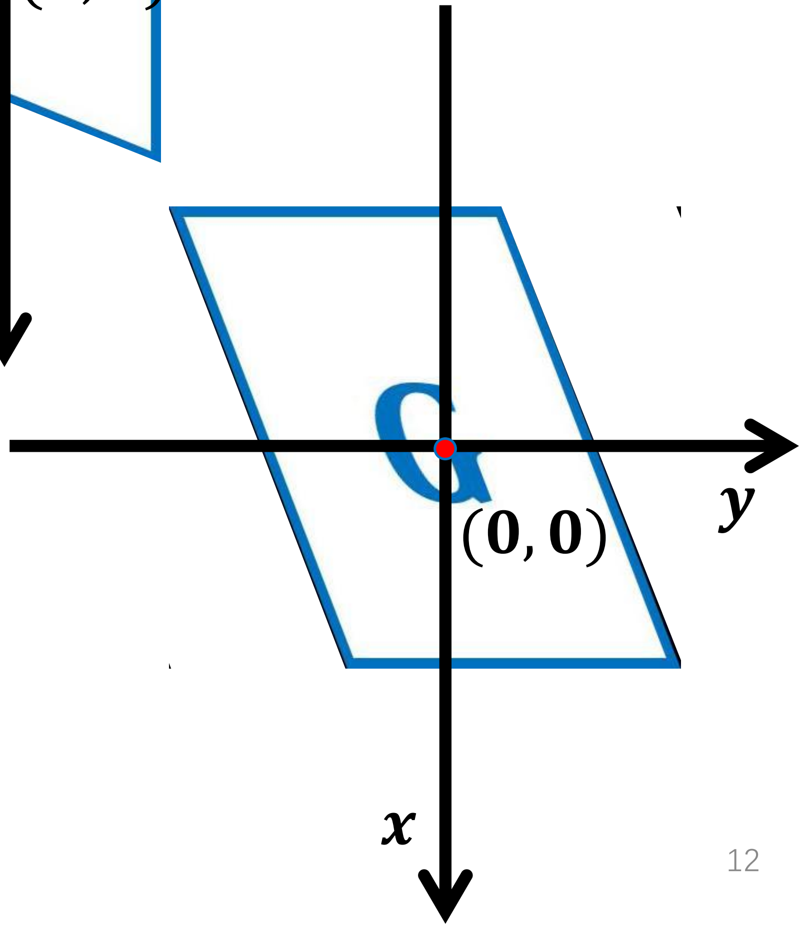
2D Linear Transform

- Scale: $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
- Rotate: $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
- Shift: $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} shift_x \\ shift_y \end{bmatrix}$$

- **Shear:**

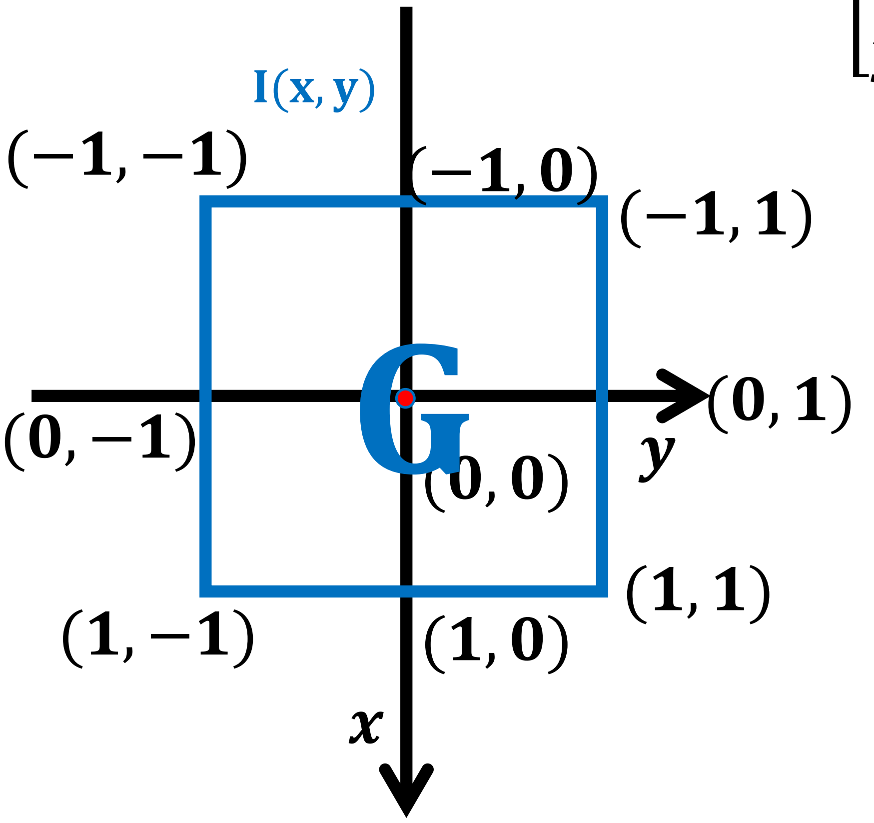






Shear in x-axis

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



$(1, 0) \longrightarrow (1, 0)$

$(0, 0) \longrightarrow (0, 0)$

$(-1, 0) \longrightarrow (-1, 0)$

$(1, -1) \longrightarrow (1 - \mathrm{b}, -1)$

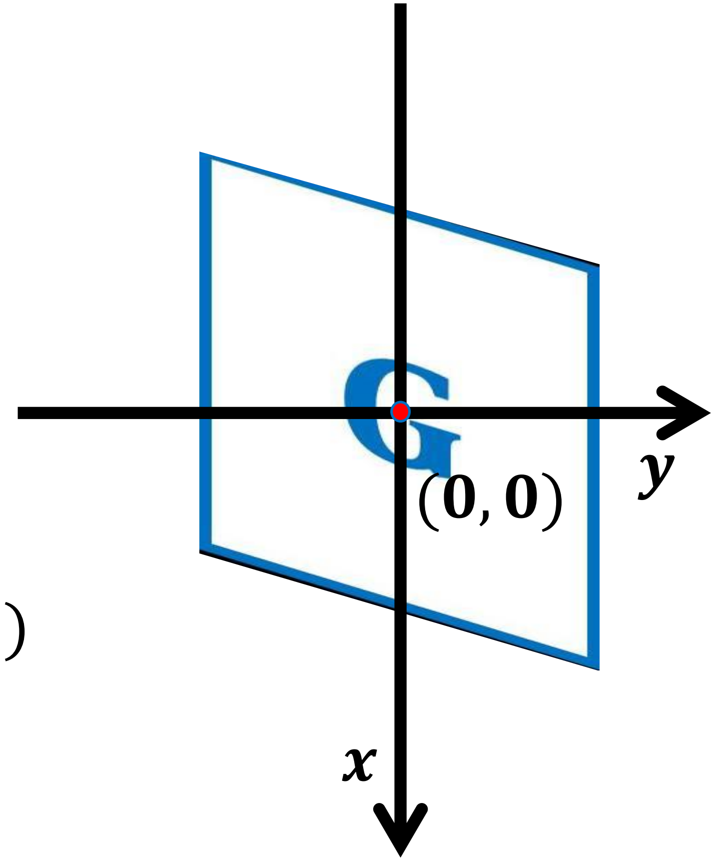
$(-1, -1) \longrightarrow (-1 - \mathrm{b}, -1)$

$(0, -1) \longrightarrow (-\mathrm{b}, -1)$

$(1, 1) \longrightarrow (\mathrm{b} + 1, 1)$

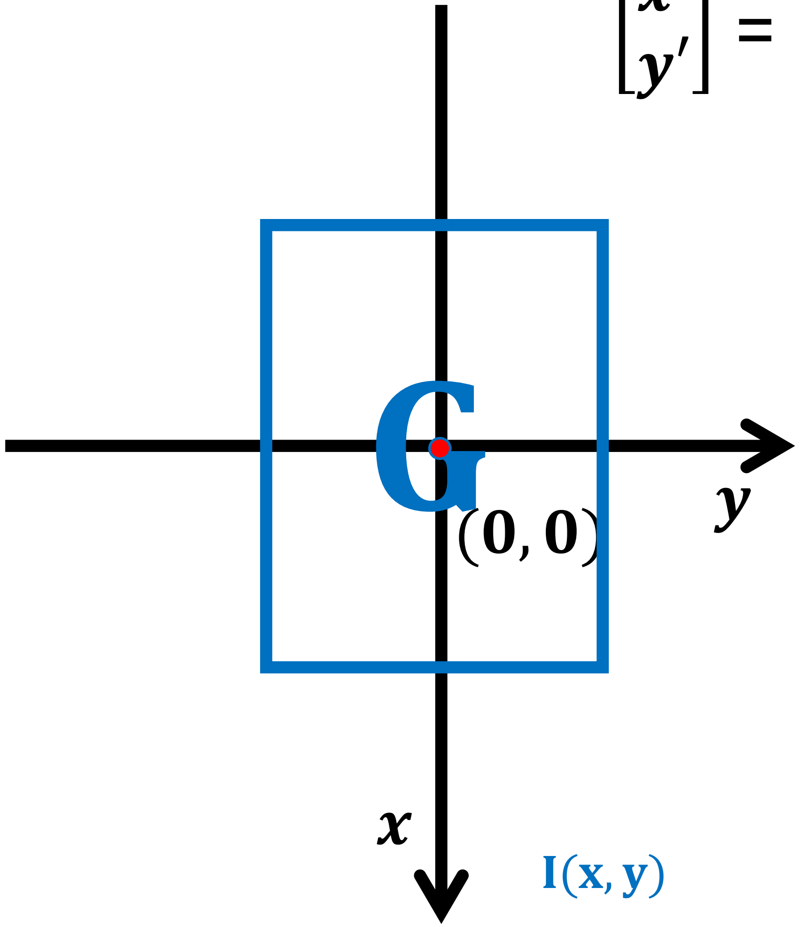
$(0, 1) \longrightarrow (\mathrm{b}, 1)$

$(-1, 1) \longrightarrow (b - 1, 1)$

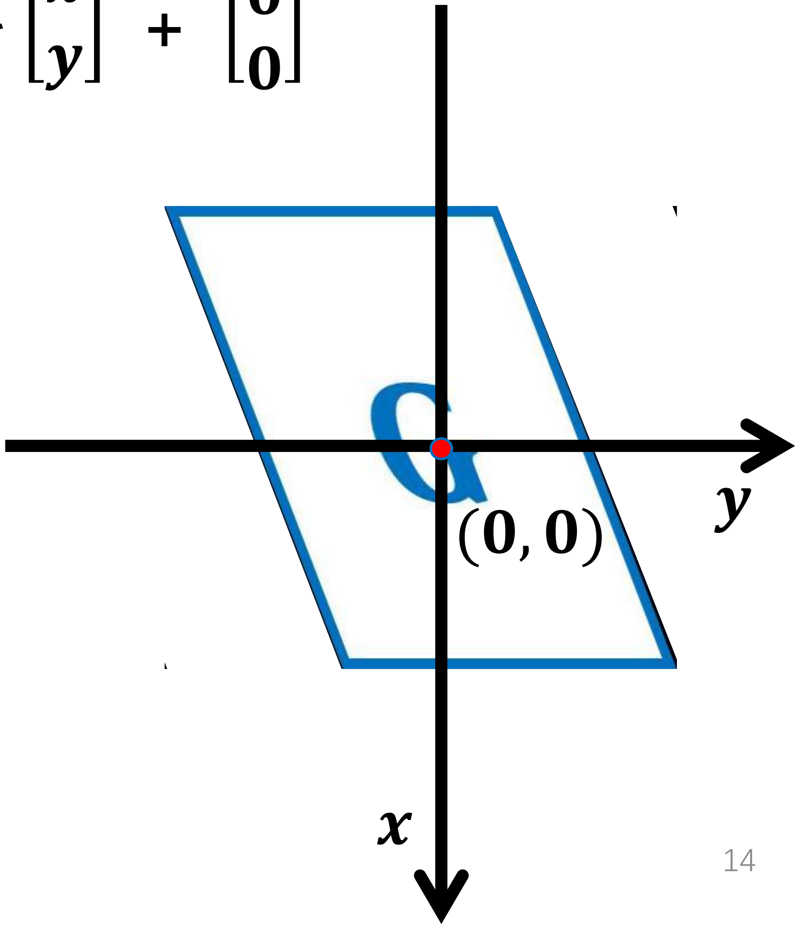


Shear in y-axis

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



$\mathbf{I(x, y)}$



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} shift_x \\ shift_y \end{bmatrix}$$

2D Linear Transform

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} shift_x \\ shift_y \end{bmatrix}$$

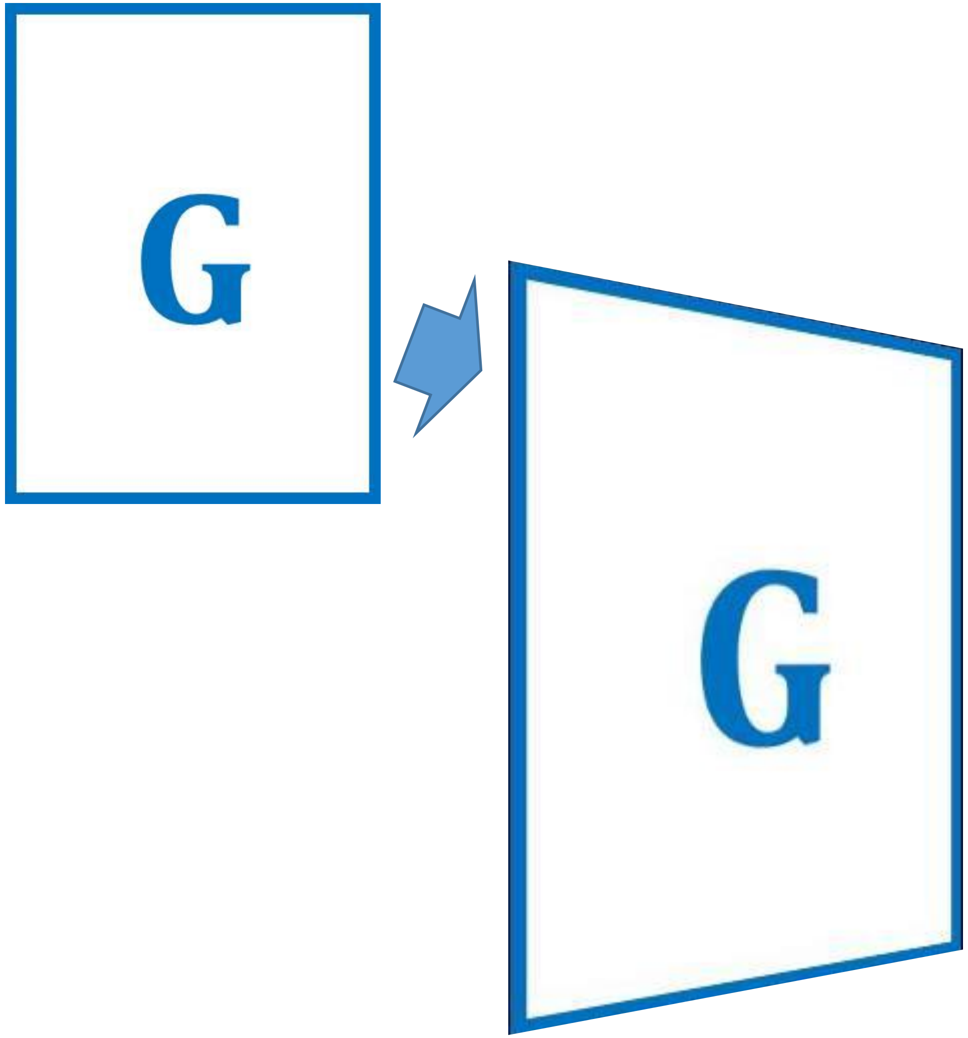
$\mathbf{J(x, y)}$ $\mathbf{I(x, y)}$

- Rotation, Scaling, Shifting are all restricted 2D linear transform and are combined as rigid body transform.
- Shear transform preserves parallel lines from the original image.
- Shear + Rotation+ Scaling+ Shifting combined all the 2D linear/affine transform. (DOF = 6)

Projective Transform

- $$\begin{bmatrix} \widetilde{x}' \\ \widetilde{y}' \\ \widetilde{z}' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \dfrac{a_{11}x+a_{12}y+b_1}{c_1x+c_2y+1} \\ \dfrac{a_{21}x+a_{22}y+b_2}{c_1x+c_2y+1} \end{bmatrix} = \begin{bmatrix} \dfrac{\widetilde{x}'}{\widetilde{z}'} \\ \dfrac{\widetilde{y}'}{\widetilde{z}'} \end{bmatrix}$$



Matlab commands

- A = [1 0 0; 0.4 1 0; 0 0 1]; % vertical shear
- tf = affine2d(A);
- im2 = imwarp(im,tf);
- figure;imshow(im2);

Practice

- **Using the zombie.jpg image, make a transform with $35°$ clock-wise rotation, 0.6 scaling and 50 shift on X-axial; 0.8 scaling and 15 shift on Y-axial.**

Outline

- Spatial Operations
  - Affine transform （仿射变换）
  - Projective transform
- Image interpolation
  - Nearest-neighbor interpolation
  - Linear & bi-linear interpolation

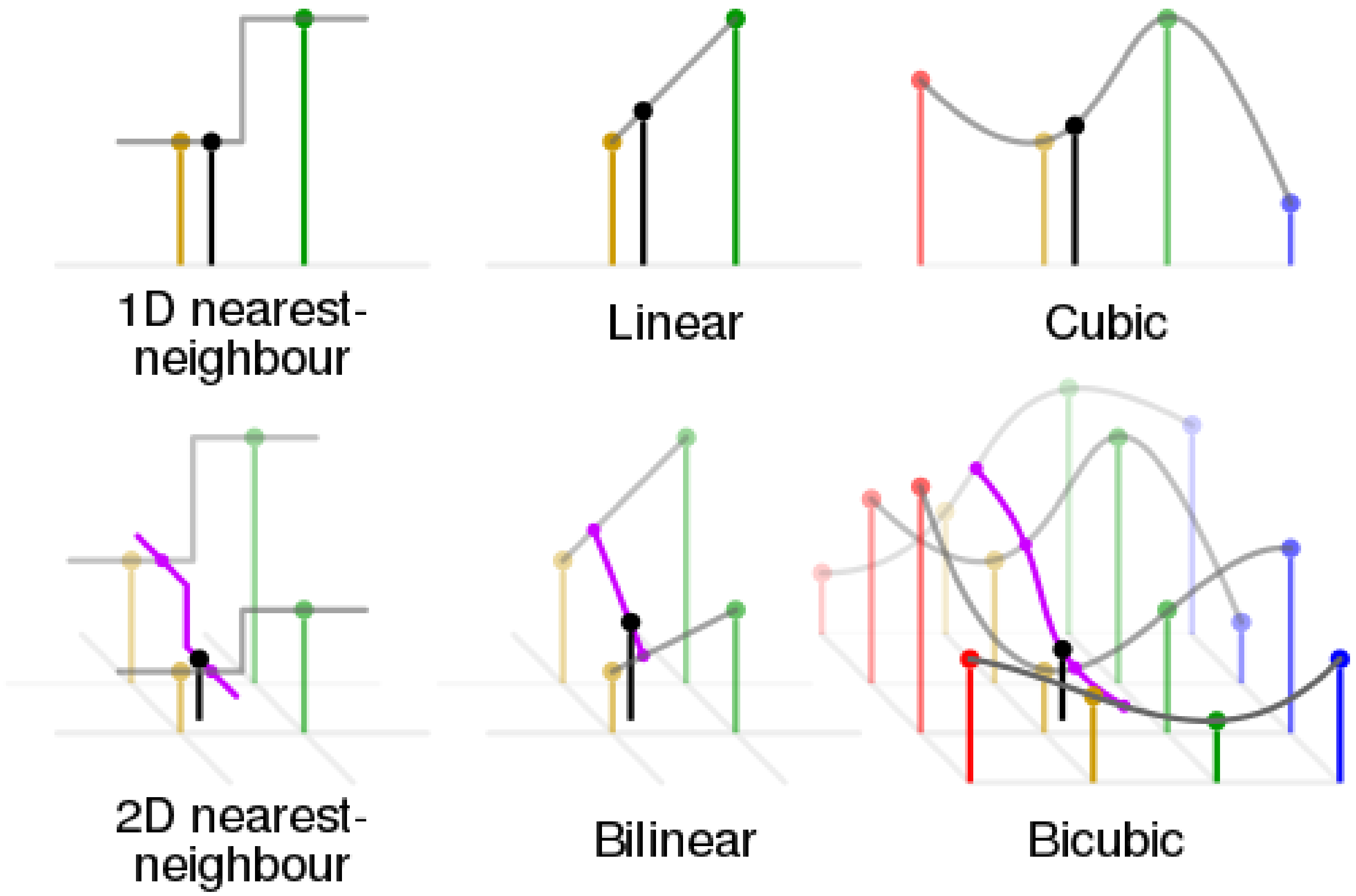
A real example

- $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} shift_x \\ shift_y \end{bmatrix}$

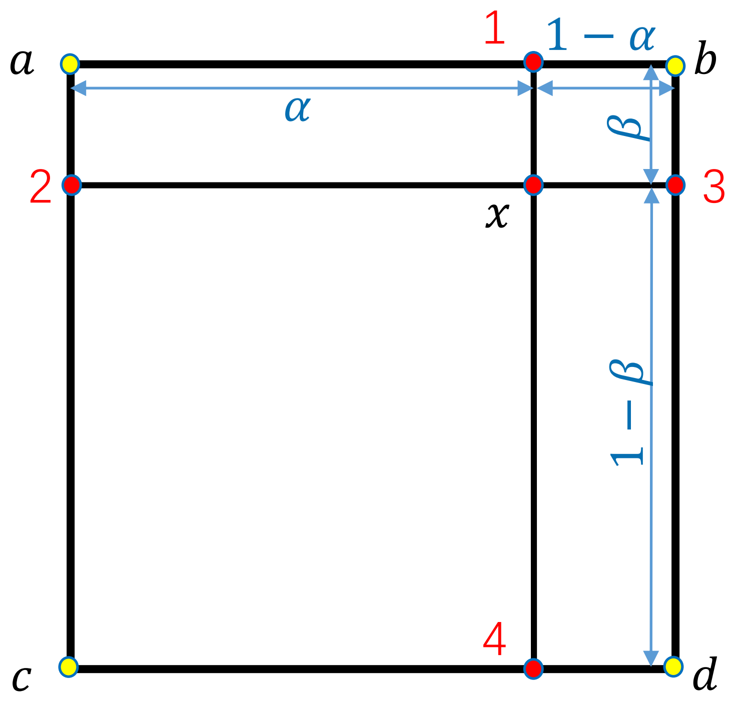
  a =1.2, $b = 1$, c =1, d =0.8, $shift_x = 3.2$, $shift_y = -1.6$

- $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 5.4 \\ 0.2 \end{bmatrix}$
- **Coordinate is not integer!! Then how to determinate intensity?**

Interpolation



Bilinear interpolation



$$x_{①} = (1-\alpha)\,a + \alpha \cdot b \quad \text{or} \quad \alpha \cdot a + \cancel{(1-\alpha)b} \; ?$$

$$\text{if} \quad \alpha = 0.7$$

$$x_{②} = (1-\beta)\,a + \beta \cdot c$$

$$x_{③} = (1-\beta)\,b + \beta \cdot b$$

$$x_{④} = (1-\alpha)\,c + \alpha \cdot d$$

$$x = (1-\beta) \cdot x_{①} + \beta \cdot x_{④}$$

$$\text{or} = (1-\alpha)\,x_{②} + \alpha \cdot x_{③}$$

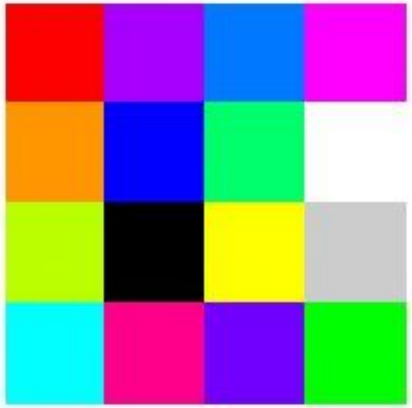
$$= (1-\beta)\left[(1-\alpha)a + \alpha b\right] + \beta\left[(1-\alpha)c + \alpha d\right]$$

$$= (1-\beta)(1-\alpha) \cdot a + \alpha(1-\beta)b + \beta(1-\alpha) \cdot c + \alpha\beta \cdot d$$

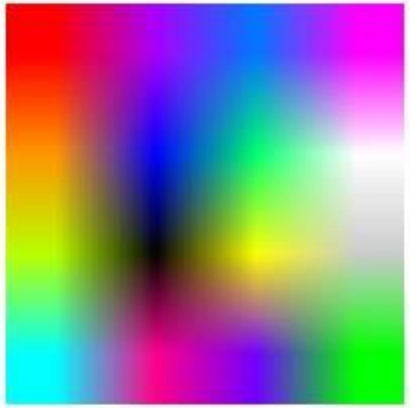
$$\alpha = 0.7, \quad \beta = 0.2$$

$$= 0.24\,a + 0.56\,b + 0.06\,c + 0.14\,d$$

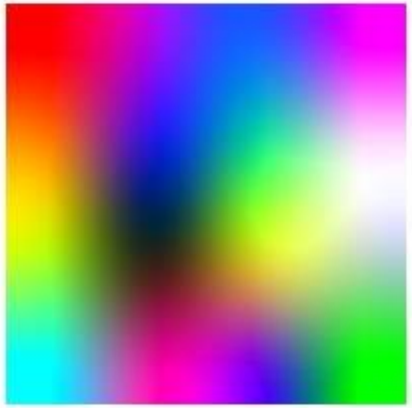
Nearest-neighbor vs Bilinear vs Bicubic interpolation



Nearest neighbor



Bilinear



Bicubic

Image Registration

- To align two or more images of the same scene
- Given input and output images, to estimate the transformation functions and then use it to register the two images

Image 1

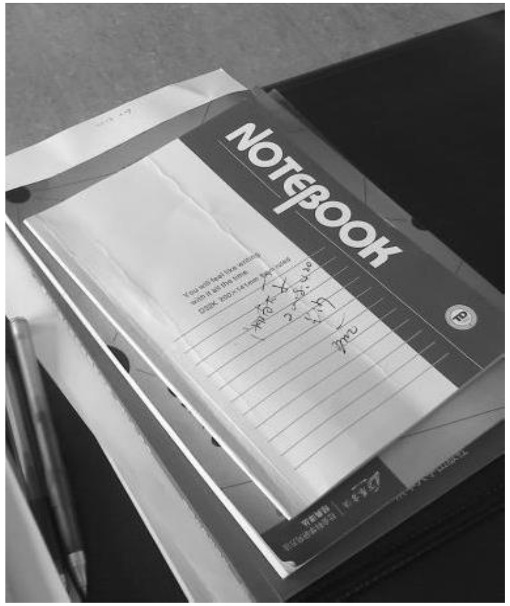
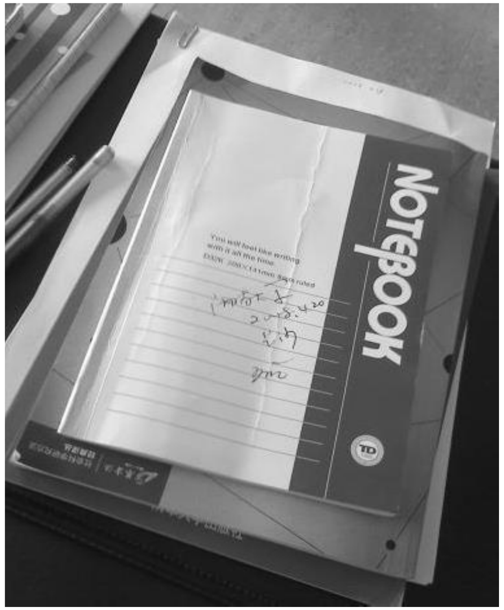
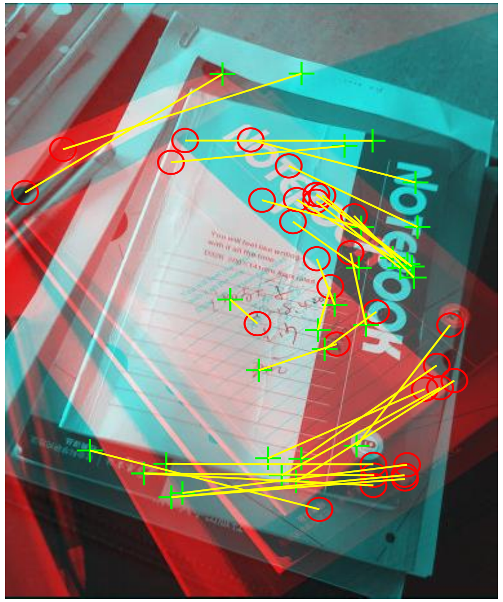


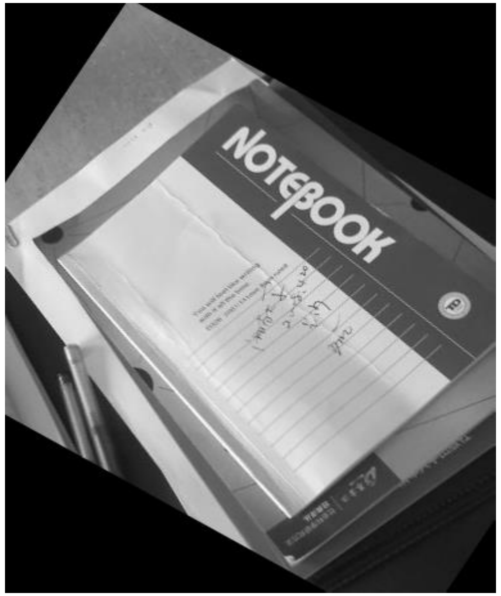
Image 2



Key points matching



Estimated rotated image



SIFT- Scale Invariant Feature Transform

Approach:

- Create a scale space of images
  - Construct a set of progressive Gaussian blurred images
  - Take the differences to get a difference of Gaussian pyramid (like a Laplacian)
- Find local extrema in this space. Choose the key points from extrema.
- For each key point, in a 16x16 window, find the histograms of gradient directions
- Create a feature vector out of these.

Take home message

- **Spatial transform： how does parameter effect on special transform. Rigid-body, shear, projective.**
- **Interpolation: To look like the nearest-neighbors and to involve more close-by neighbors.**