# Ch.2 Basic Mathematical Tools in DIP

## Digital Image Processing

Instructor: Dr. Yu-Wei Wen

# Negative Image

- For each pixel,

$$Intensity_{\text{new}} = Intensity_{\text{max}} - Intensity_{\text{original}}$$

# Negative Image

- For each pixel,

$$Intensity_{\text{new}} = Intensity_{\text{max}} - Intensity_{\text{original}}$$

- Given a source image $f(x, y)$ of size $H \times W$, the negative image can be denoted

$$g(x, y) = 255 - f(x, y) \quad \forall \ 0 < x \leq H \text{ and } 0 < y \leq W$$

# Noise Reduction using Image Addition

- Noise of an image
  - Unwanted signal (electronic fluctuation)
  - Usually occurs in dark scenes
  - Sensor/circuit heat (illumination)

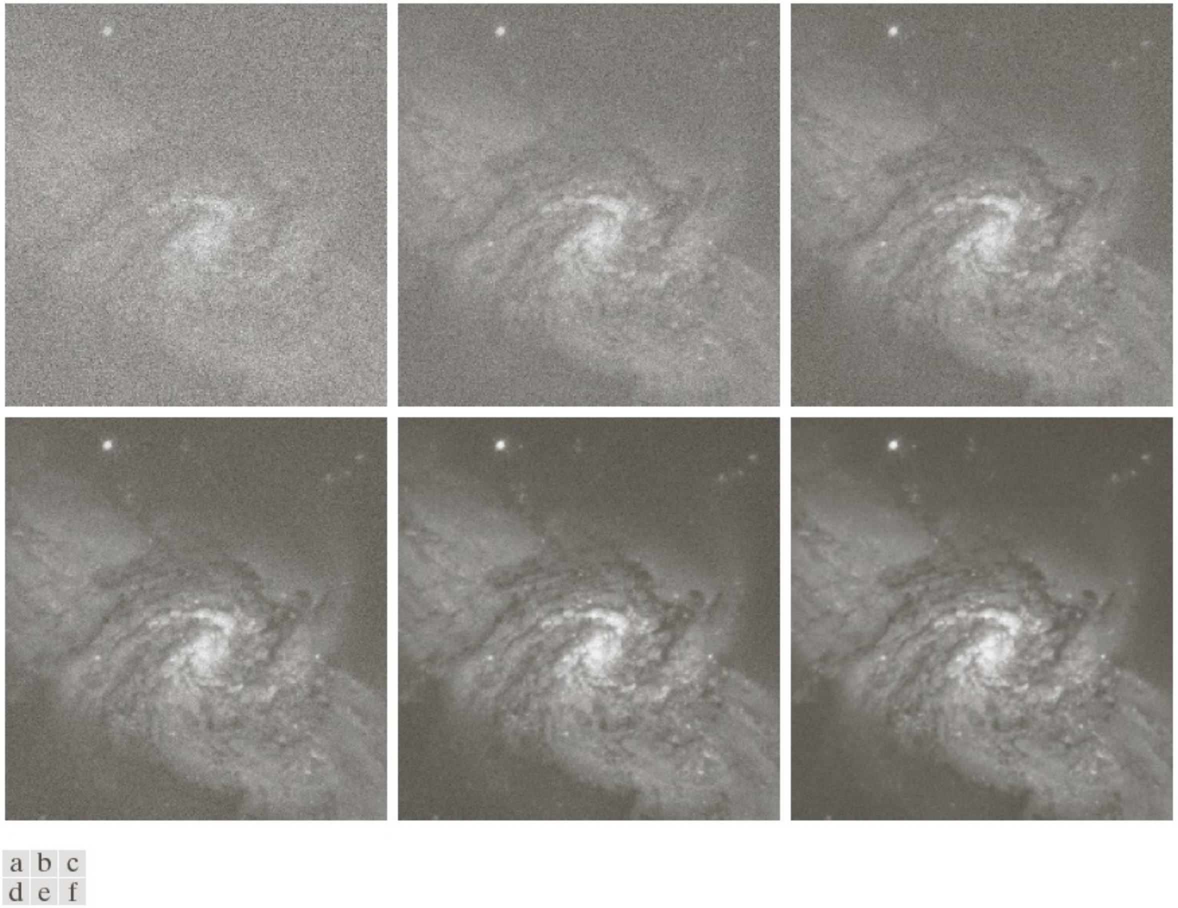



**FIGURE 2.26** (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

# Noise Reduction using Image Addition

- Given a *noiseless* image $f(x, y)$, a corrupted (noisy) image $g(x, y)$ can be formed by adding noise $\eta(x, y)$ to $f(x, y)$.
- Assumption
  - The noise is uncorrelated.
  - The noise and image values are uncorrelated.
  - The noise has a zero average value $\bar{\eta}(x, y) = 0$

# Noise Reduction using Image Addition

- Given a *noiseless* image $f(x, y)$, a corrupted (noisy) image $g(x, y)$ can be formed by adding noise $\eta(x, y)$ to $f(x, y)$.
- Assumption
  - The noise is uncorrelated.
  - The noise and image values are uncorrelated.
  - The noise has a zero average value $\bar{\eta}(x, y) = 0$
- Noise can be reduced by adding (averaging) a set of noisy images

$$\bar{g}(x, y) = \frac{1}{n} \sum_{i=0}^{n} g_i(x, y)$$

# Noise Reduction using Image Addition

- Noise can be reduced by adding (averaging) a set of noisy images

$$\bar{g}(x,y) = \frac{1}{n}\sum_{i=0}^{n} g_i(x,y)$$

- Since $g(x,y) = f(x,y) + \eta(x,y)$ and $\bar{\eta}(x,y) = 0$
- It follows that

$$E\{\bar{g}(x,y)\} = E\left\{\frac{1}{n}\sum_{i=0}^{n} g_i(x,y)\right\} = E\left\{\frac{1}{n}\sum_{i=0}^{n} f(x,y) + \eta_i(x,y)\right\}$$

$$= f(x,y)$$

# Noise Reduction using Image Addition

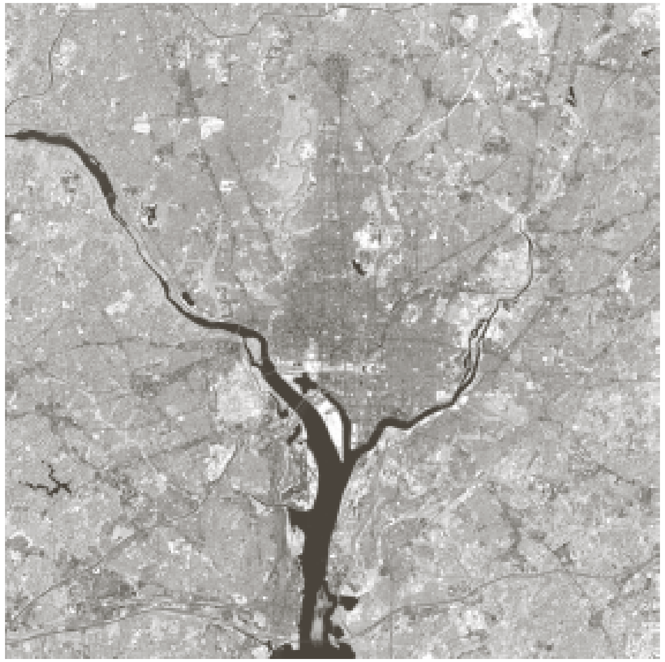
- Noise can be reduced by adding (averaging) a set of noisy images

$$\bar{g}(x, y) = \frac{1}{n}\sum_{i=0}^{n} g_i(x, y)$$

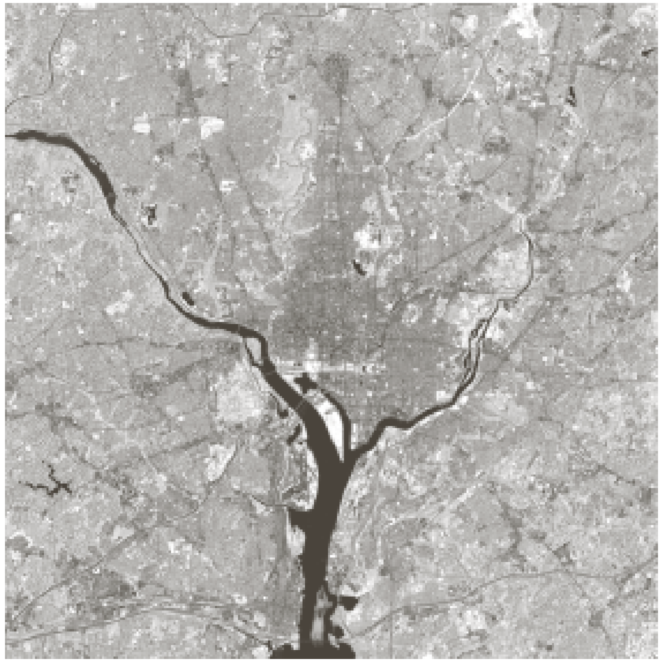
- Other applications
  - Moving object removal
  - Light trace photo (could use average or max)

# Comparing Images using Subtraction

- In some cases, the difference between images is so minor that the images are **indistinguishable to human eyes**.
- Example:



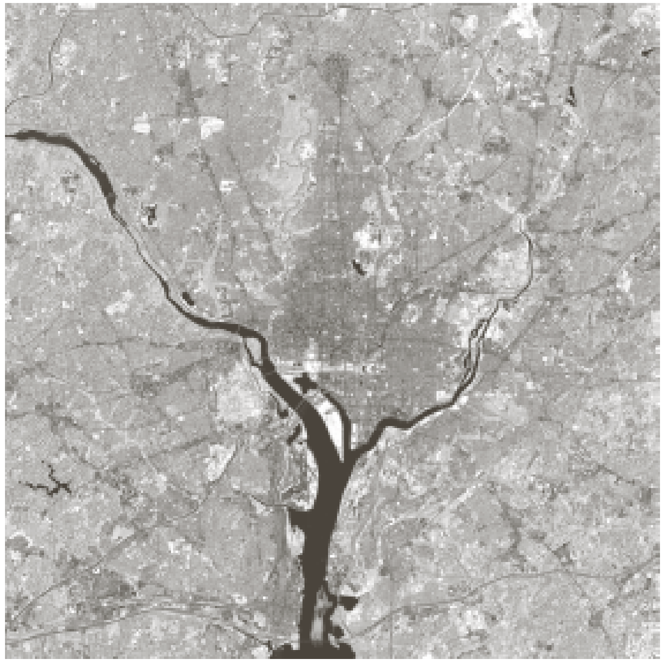
$A_1$



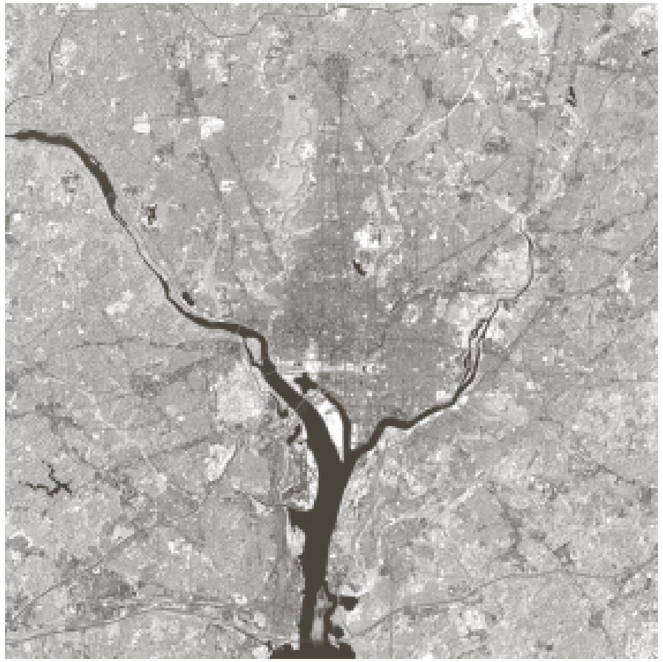
$A_2$

# Comparing Images using Subtraction

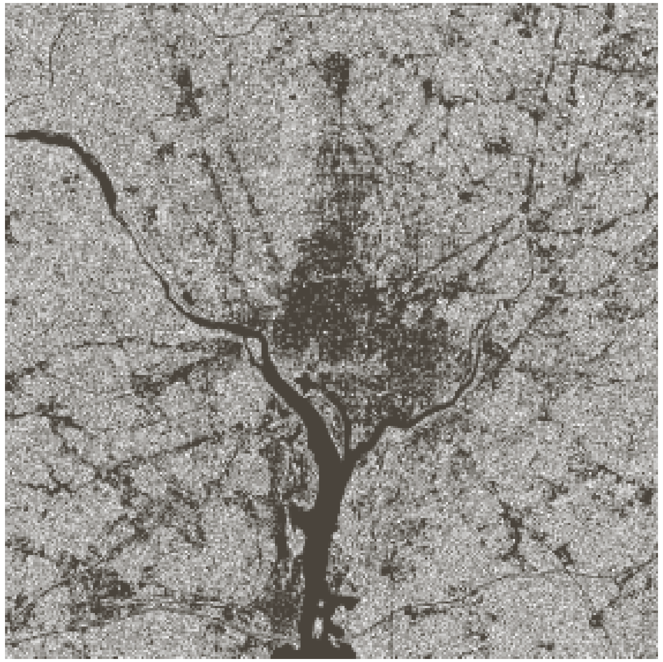
- In some cases, the difference between images is so minor that the images are **indistinguishable to human eyes**.
- Example:



$A_1$



$A_2$



$A_1 - A_2$

# Shading Correction using Image Division

- Suppose an imaging sensor produces **shaded images** due to lens/sensor fatigue. The shaded image is denoted

$$g(x, y) = f(x, y)h(x, y)$$

where $f(x, y)$ is the perfect image and $h(x, y)$ is the shading function.

# Shading Correction using Image Division

- Suppose an imaging sensor produces **shaded images** due to lens/sensor fatigue. The shaded image is denoted
$$g(x, y) = f(x, y)h(x, y)$$
where $f(x, y)$ is the perfect image and $h(x, y)$ is the shading function.

- The perfect image can be obtained by image division
$$g(x, y)/h'(x, y)$$
where $h'(x, y)$ is the estimated shading function.

# Image Masking using Multiplication

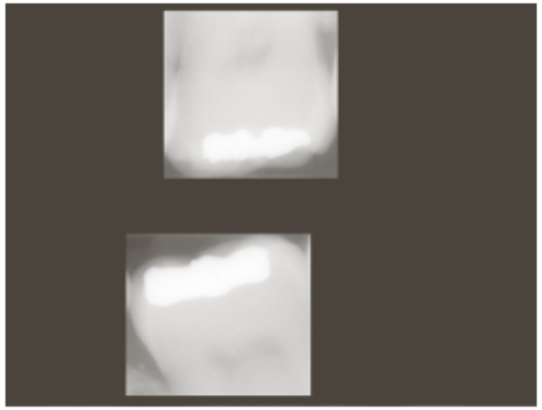
- Given an image $f(x, y)$ and a **mask** $h(x, y)$, the masked image can be denoted

$$g(x, y) = f(x, y)h(x, y)$$

- Note that the multiplication here refers pixel-wise product.

$f(x, y)$

$h(x, y)$



$f(x, y)h(x, y)$

# Spatial Operations

# Single-Pixel Operations

- The simplest operation to a digital image
- Altering the intensity of individual pixel using a ***transformation*** function $T: \mathbb{Z} \rightarrow \mathbb{Z}$ as

$$p' = T(p)$$

- For example, the following transformation yields negative image effect:

$$T(p) = 255 - p$$

# Single-Pixel Operations

- The simplest operation to a digital image
- Altering the intensity of individual pixel using a ***transformation*** function $T: \mathbb{Z} \rightarrow \mathbb{Z}$ as

$$p' = T(p)$$

- For example, the following transformation yields negative image effect:

$$T(p) = 255 - p$$

# Single-Pixel Operations

Questions—What image of these transformation function yields?

$$T_1(p) = \min(p + 100, 255)$$

$$T_2(p) = \max(p - 100, 0)$$

$$T_3(p) = \begin{cases} \max(p - 50, \quad 0), & \text{if } p < \bar{p} \\ \min(p + 50, \quad 255), & \text{if } p \geq \bar{p} \end{cases}$$

# Single-Pixel Operations

Questions—What image of these transformation function yields?

$T_1$

$T_2$

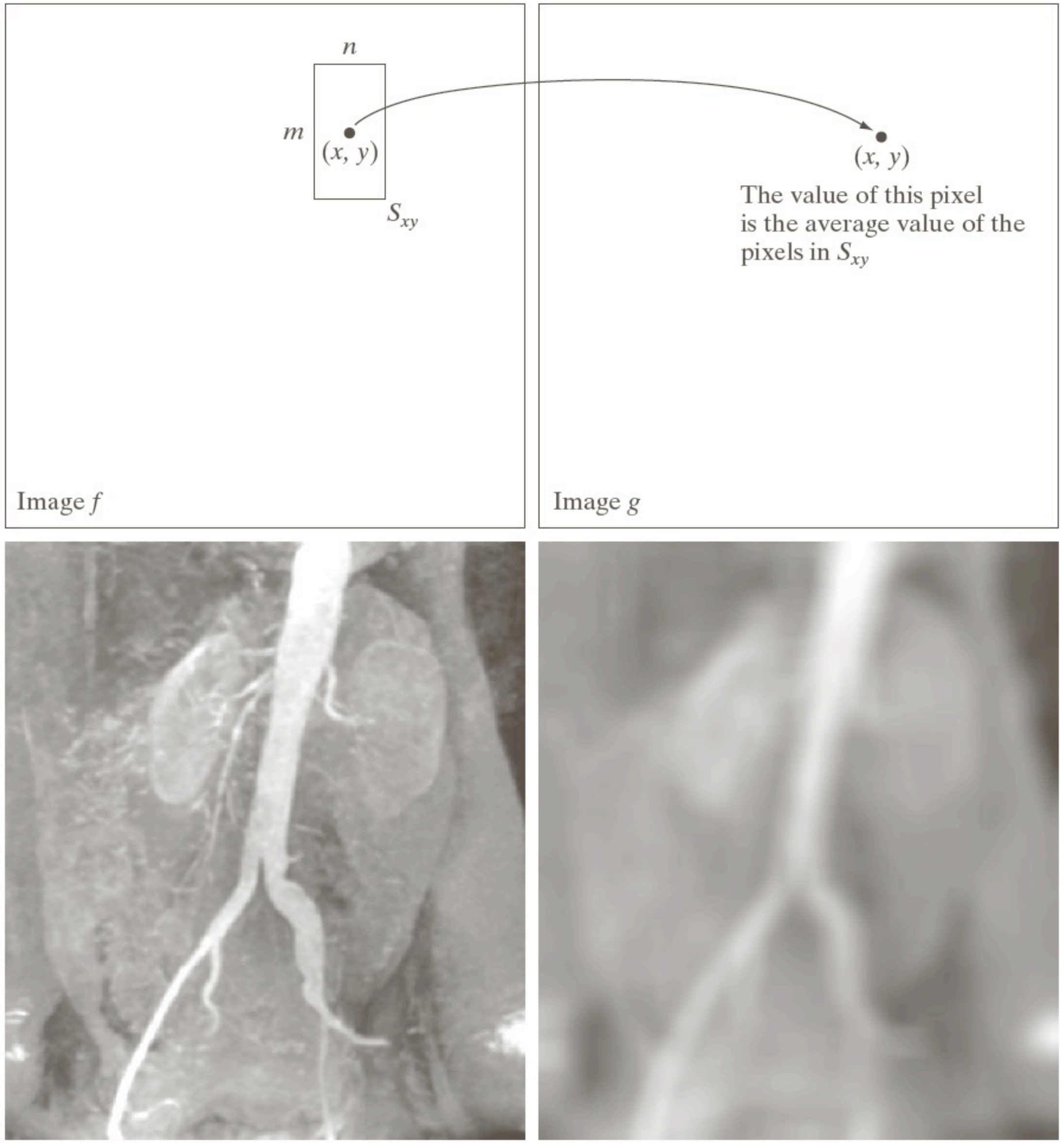
$T_3$

# Neighborhood Operations

- Let $S_{xy}$ denote the set of coordinates of a neighborhood centered on an arbitrary point $(x, y)$ in an image $f$.
- The following neighborhood operation yields a blurred image $g$.

$$g(x, y) = \frac{1}{mn} \sum_{(r,c) \in S_{xy}} f(r, c)$$

# Neighborhood Operations




$$g(x,y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} f(r,c)$$

# Geometric Transformations

- Modify spatial arrangement of pixels
- *Rubber-sheet transformations*
  - Painting an image on a rubber sheet.
  - Stretching or shrinking the sheet.
- Consist of two basic operations
  - Spatial transformation of coordinates.
  - Intensity interpolation that assigns intensity values to the spatially transformed pixels.

# Geometric Transformations

- Affine Transformations
  - Straight **lines** and **parallelism** are preserved.
  - Scaling, rotation, shearing, translation, etc.
- Let $(x, y)$ be the pixel coordinates in the **original** image and $(x', y')$ be the pixel coordinates in the **transformed** image.
- For example, $(x', y') = (x/2, y/2)$ shrinks the original image to half of its size.

# Geometric Transformations

- For example, $(x', y') = (x/2, y/2)$ shrinks the original image to half of its size.
- Matrix form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- General form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

# Geometric Transformations

- To support translation, an additional dimension is required. The general form of affine transformation is rewritten as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

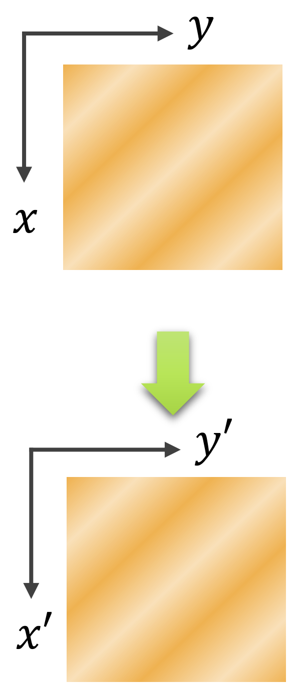
# Geometric Transformations

- Identity transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Identity

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



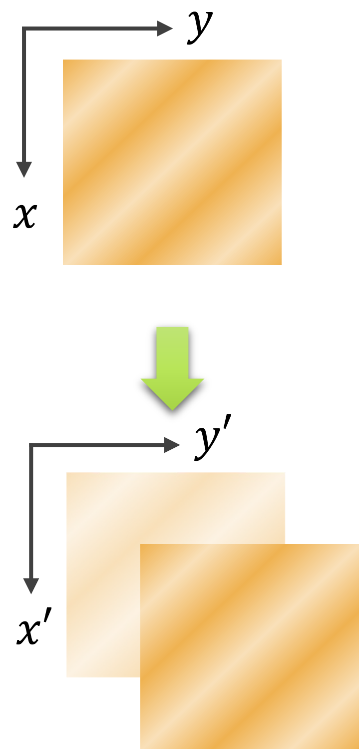
# Geometric Transformations

- Affine transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Translation

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



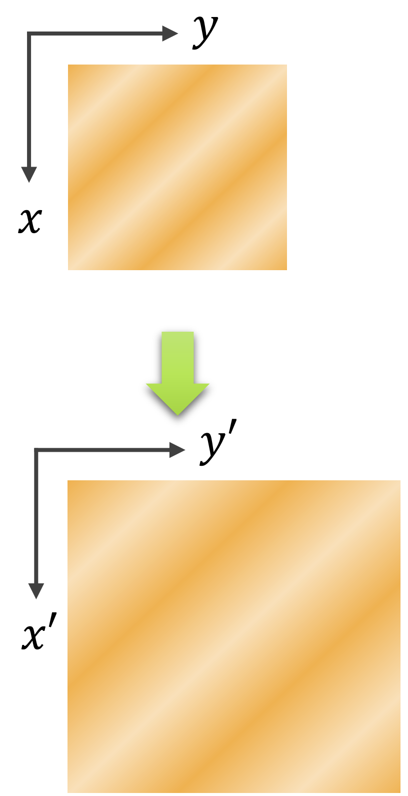
# Geometric Transformations

- Affine transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Scaling

$$\mathbf{A} = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



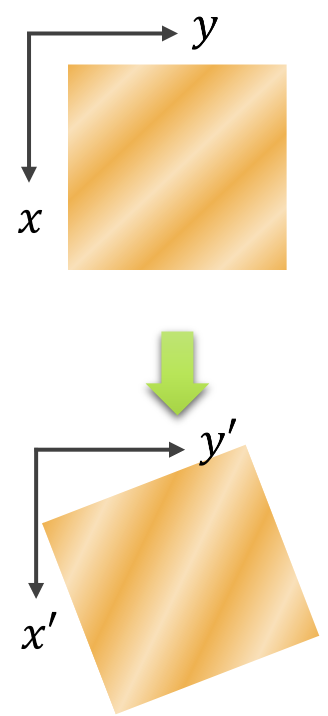
# Geometric Transformations

- Affine transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Rotation

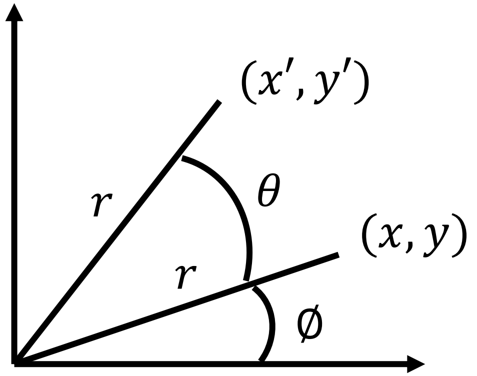
$$\mathbf{A} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



# Geometric Transformations

- Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$




$$x = r\cos\emptyset \qquad x' = r\cos(\emptyset + \theta)$$
$$y = r\sin\emptyset \qquad y' = r\sin(\emptyset + \theta)$$

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

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

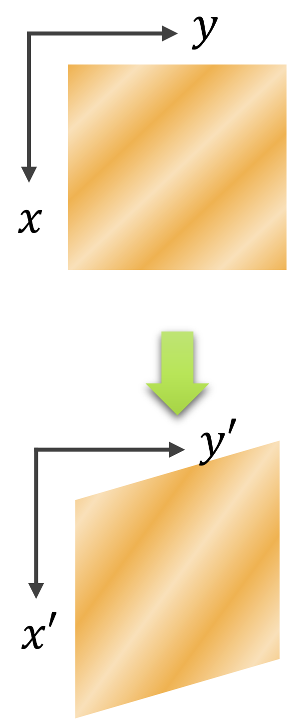
# Geometric Transformations

- Affine transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Shearing (vertical)

$$\mathbf{A} = \begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$




$s_v < 0$

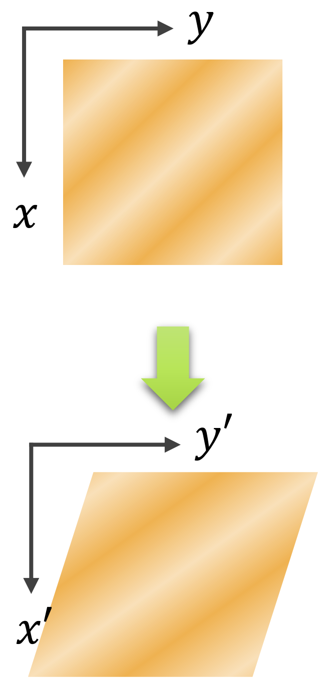
# Geometric Transformations

- Affine transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Shearing (horizontal)

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$s_h < 0$

# Geometric Transformations

- Forward mapping

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Problems of forward mapping
  - Many-to-one mapping
  - Unmapped pixels

# Geometric Transformations

- Forward mapping

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Backward mapping

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

# Geometric Transformations

- Backward mapping

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

- Two basic operations
  - Coordinates transformation (see above)
  - Intensity interpolation (previously discussed)

# Geometric Transformations

- Two basic operations
  - Coordinates transformation (see above)
  - Intensity interpolation (previously discussed)