# Lecture 3 - Image Fundamentals

This lecture will cover:

- **Image operation (图像运算操作)**
- Image interpolation (图像插值)
- Image registration (图像配准)
- Image reconstruction (图像重建)

# Image Operations

- Array and Matrix Operation
- Vector and Matrix Operation
- Linear and Nonlinear Operation
- Set and Logical Operation
- Arithmetic Operation
- Spatial Operation
- Image Transformation
- Probabilistic Methods

# Array and Matrix Operation

**Consider two 2 x 2 image**

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

- **Array product**

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{11} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

- **Matrix product**

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

# Vector and Matrix Operation

- **Multispectral image processing**

A pixel in a n-dimensional space can be expressed as a column vector $Z = [z_1, z_2 \ldots z_n]^T$, then a vector norm between two pixels $Z$ and $A$

$$\|Z - A\| = [(Z - A)^T (Z - A)]^{\frac{1}{2}}$$
$$= [(z_1 - a_1)^2 + (z_2 - a_2)^2 + \cdots + (zn - an)^2]^{\frac{1}{2}}$$

- **Linear transformations**

$$g = Hf + n$$



上海科技大学 ShanghaiTech University

# Linear and Nonlinear Operation

An operator

$$H[f(x,y)] = g(x,y)$$

is linear if

$$H[a_i f_i(x,y) + ajfj(x,y)] = a_i H[f_i(x,y)] + ajH[f_j(x,y)]$$
$$= a_i g_i(x,y) + a_j g_j(x,y)$$

- Additivity (相加性)
- Homogeneity (同质性)

# Set Operation

**TABLE 2.1**
Some important set operations and relationships.

| Description | Expressions |
| --- | --- |
| Operations between the sample space and null sets | $\Omega^c = \varnothing;\ \varnothing^c = \Omega;\ \Omega \cup \varnothing = \Omega;\ \Omega \cap \varnothing = \varnothing$ |
| Union and intersection with the null and sample space sets | $A \cup \varnothing = A;\ A \cap \varnothing = \varnothing;\ A \cup \Omega = \Omega;\ A \cap \Omega = A$ |
| Union and intersection of a set with itself | $A \cup A = A;\ A \cap A = A$ |
| Union and intersection of a set with its complement | $A \cup A^c = \Omega;\ A \cap A^c = \varnothing$ |
| Commutative laws | $A \cup B = B \cup A$<br>$A \cap B = B \cap A$ |
| Associative laws | $(A \cup B) \cup C = A \cup (B \cup C)$<br>$(A \cap B) \cap C = A \cap (B \cap C)$ |
| Distributive laws | $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$<br>$(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$ |
| DeMorgan's laws | $(A \cup B)^c = A^c \cap B^c$<br>$(A \cap B)^c = A^c \cup B^c$ |

# Set Operation (Coordinates)

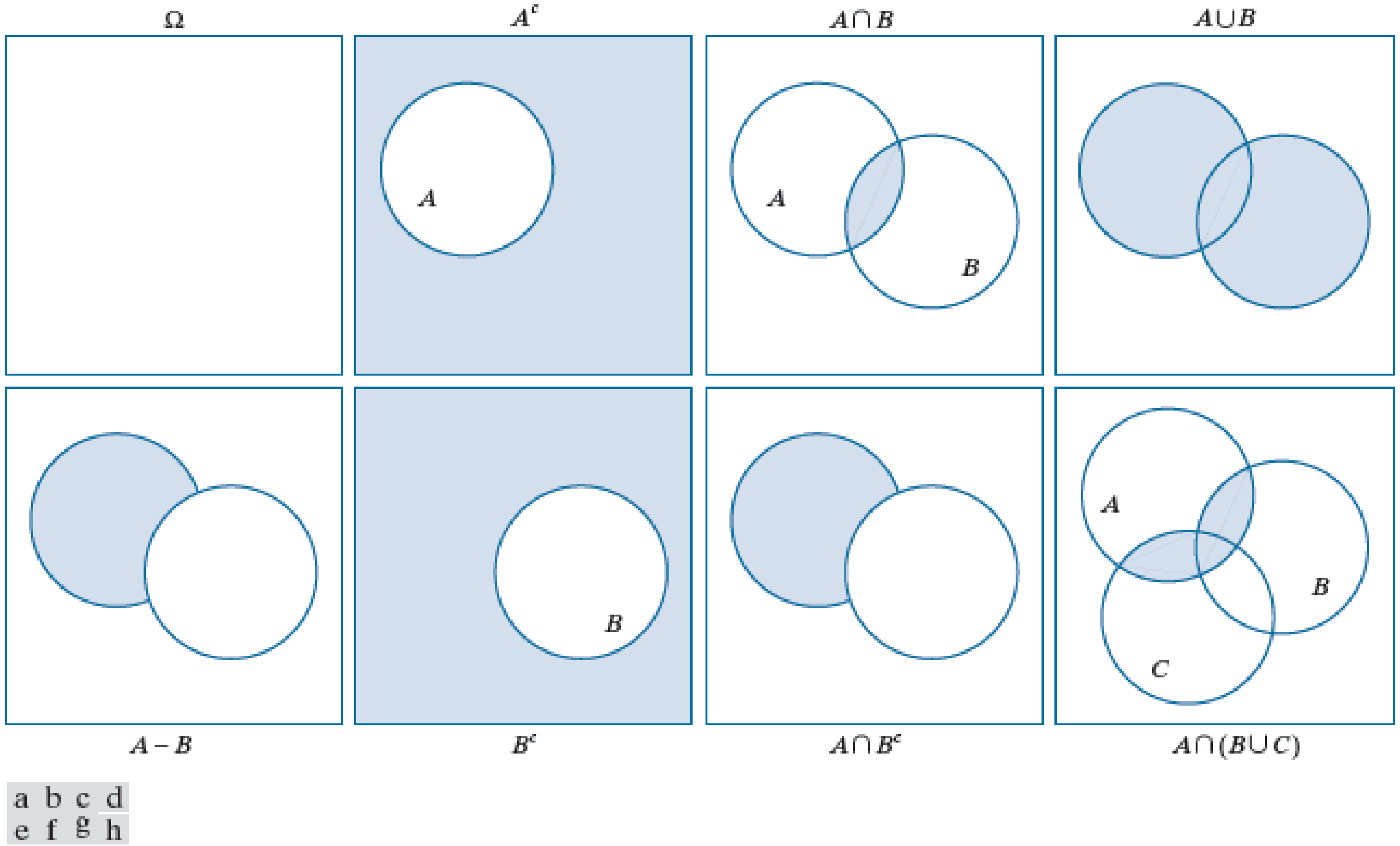
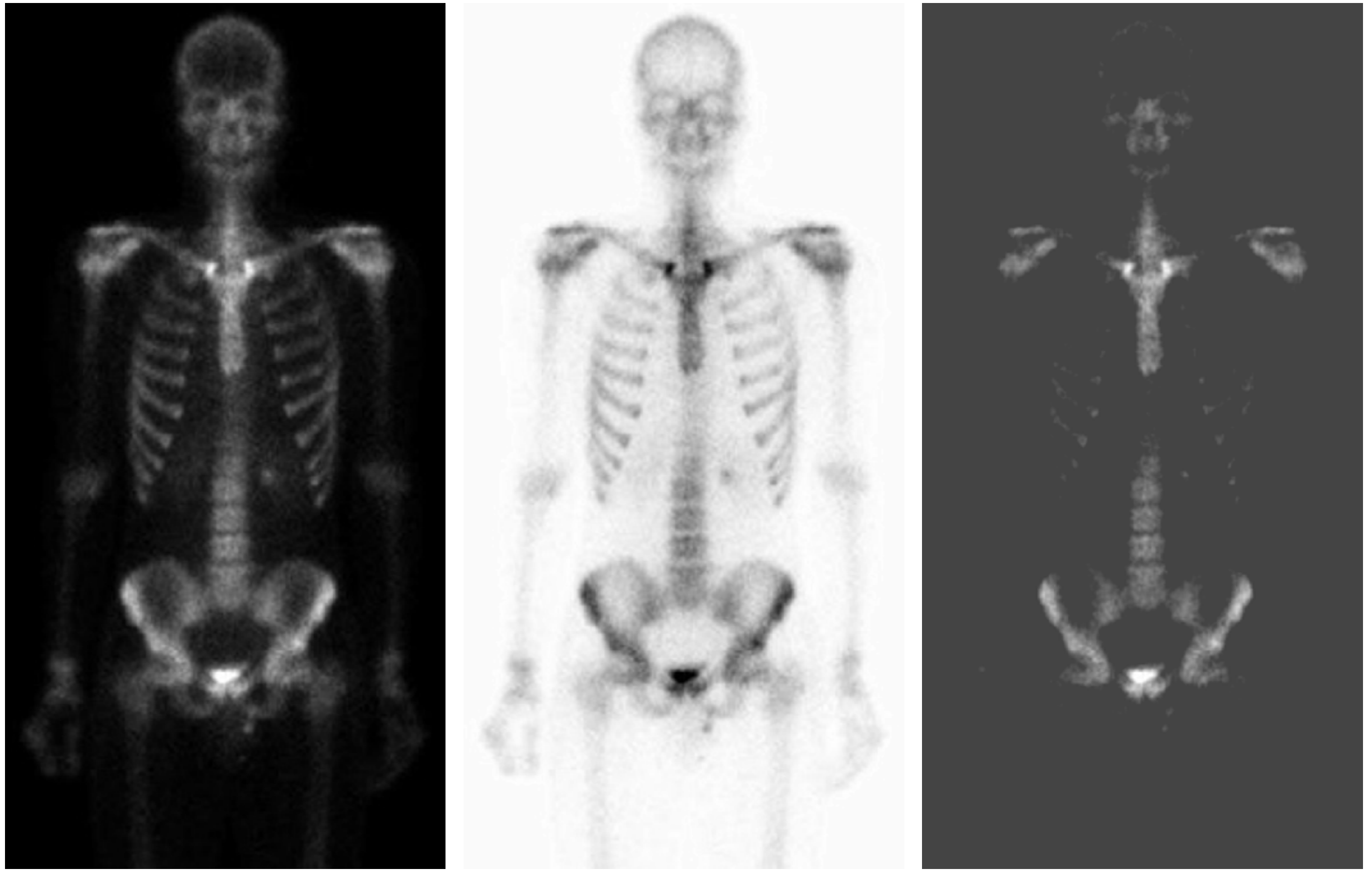



**FIGURE 2.35** Venn diagrams corresponding to some of the set operations in Table 2.1. The results of the operations, such as $A^c$, are shown shaded. Figures (e) and (g) are the same, proving via Venn diagrams that $A - B = A \cap B^c$ [see Eq. (2-40)].

# Set Operation (Intensity)

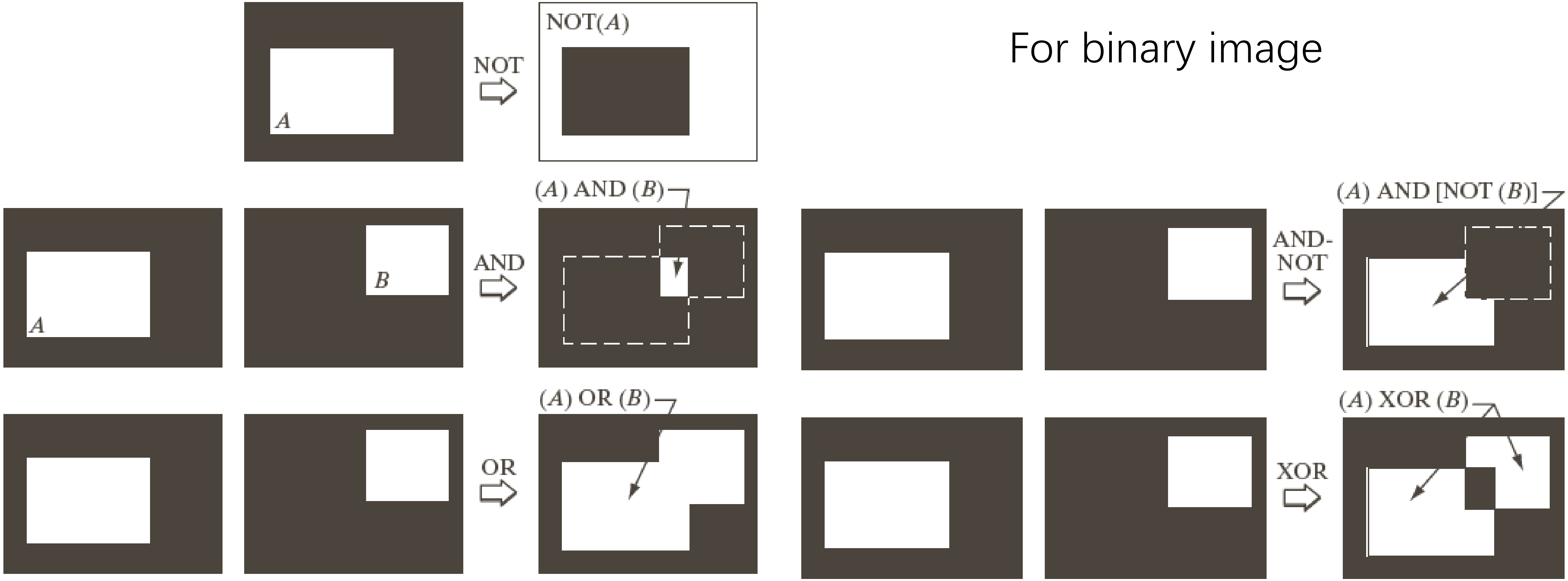


a b c

**FIGURE 2.36** Set operations involving grayscale images. (a) Original image. (b) Image negative obtained using grayscale set complementation. (c) The union of image (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

# Logical Operation

For binary image



# Arithmetic Operation

- **Addition**

$$s(x,y) = f(x,y) + g(x,y)$$

- **Subtraction**

$$d(x,y) = f(x,y) - g(x,y)$$

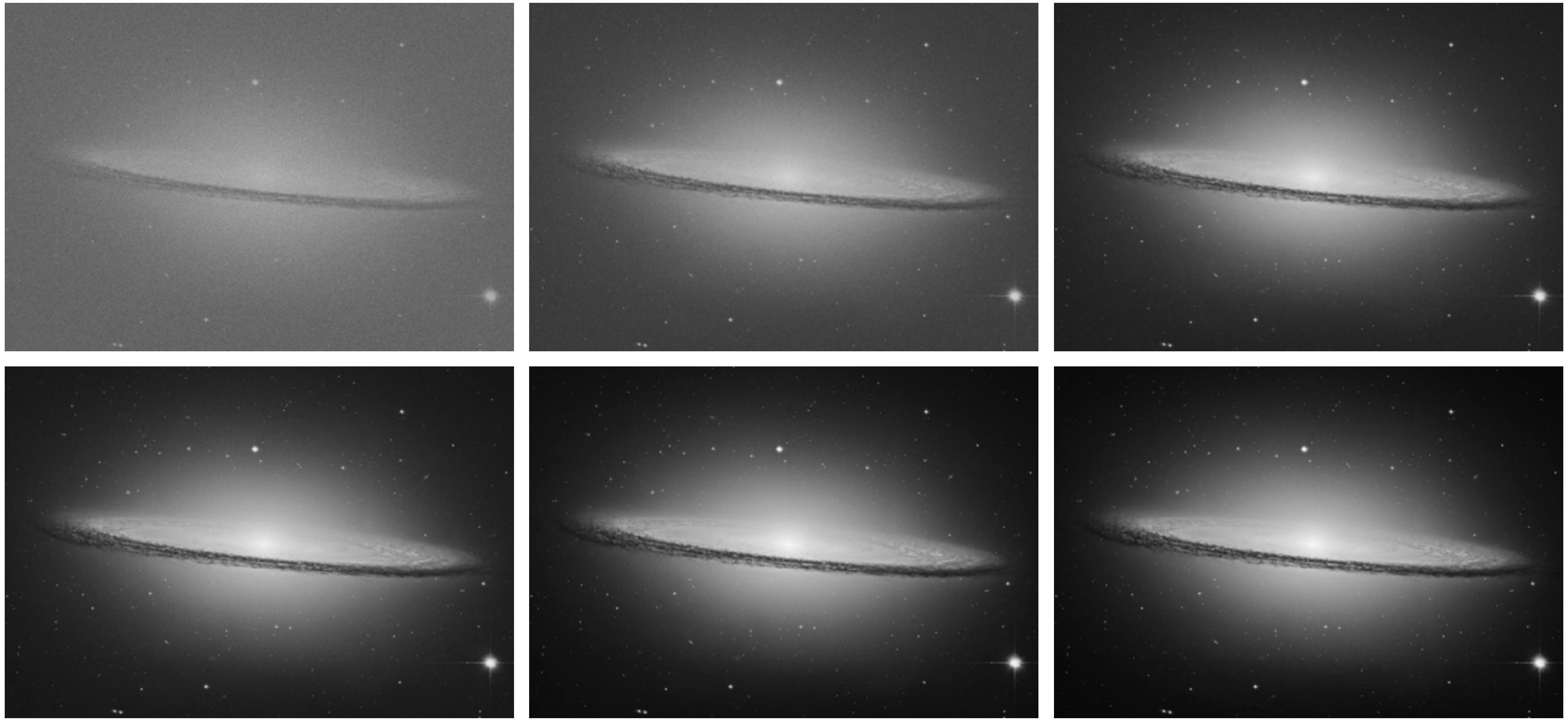
- **Multiplication**

$$p(x,y) = f(x,y) \times g(x,y)$$

- **Division**

$$v(x,y) = f(x,y) \div g(x,y)$$

# Image Addition



a b c
d e f

**FIGURE 2.29** (a) Sample noisy image of the Sombrero Galaxy. (b)-(f) Result of averaging 10, 50, 100, 500, and 1,000 noisy images, respectively. All images are of size $1548 \times 2238$ pixels, and all were scaled so that their intensities would span the full [0, 255] intensity scale. (Discovered in 1767, the Sombrero Galaxy is 28 light years from Earth. Original image courtesy of NASA.)

# Image Addition

If $f(x, y)+g(x, y) > L_{max}$ , $s(x, y)$ can be calculated as

- **Average**

$$s(x,y) = \frac{f(x,y) + g(x,y)}{2}$$

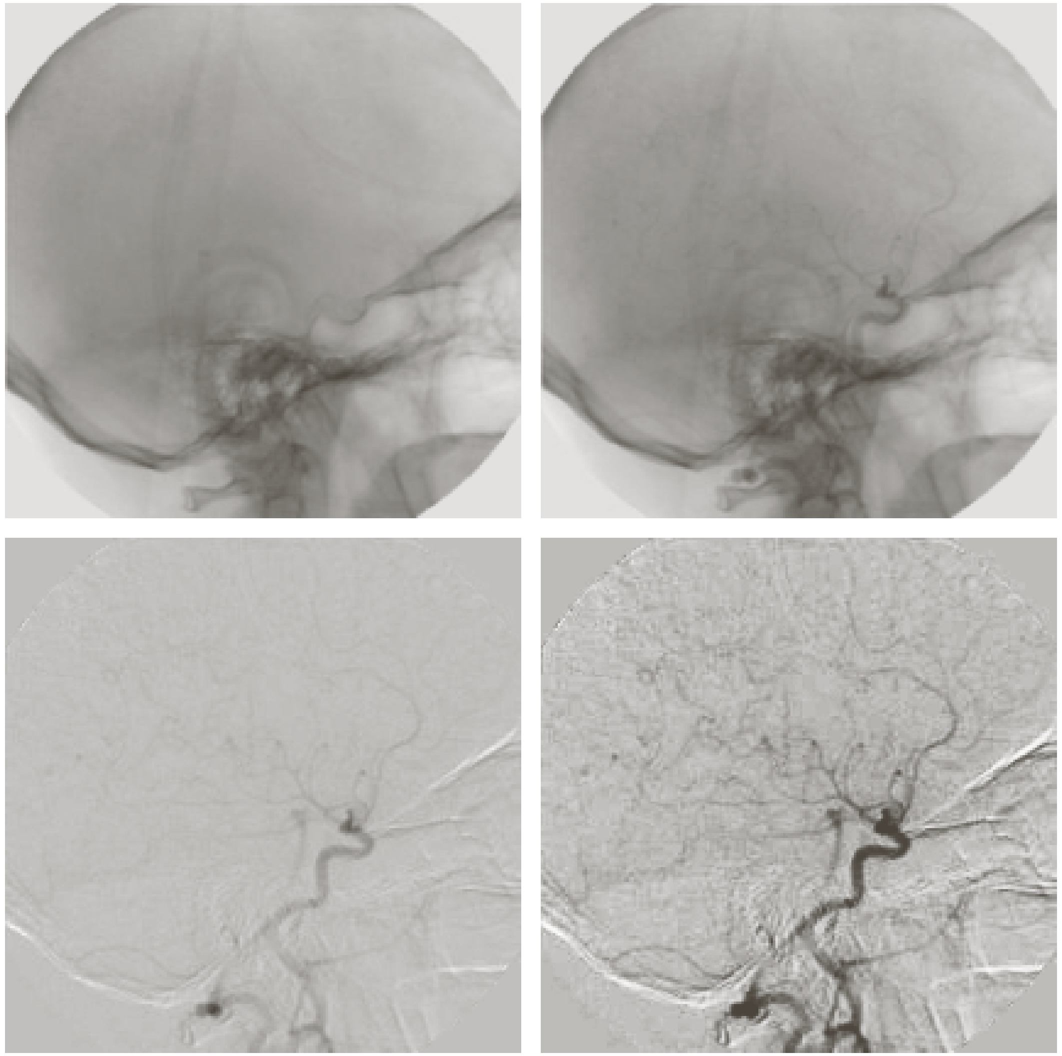
- **Scale**

$$\{\min[s(x,y)], \max[s(x,y)]\} = \{0, L_{\max}\}$$

- **Max intensity value**

$$\text{If } s(x,y) > L_{\max}, \qquad s(x,y) = L_{\max}$$

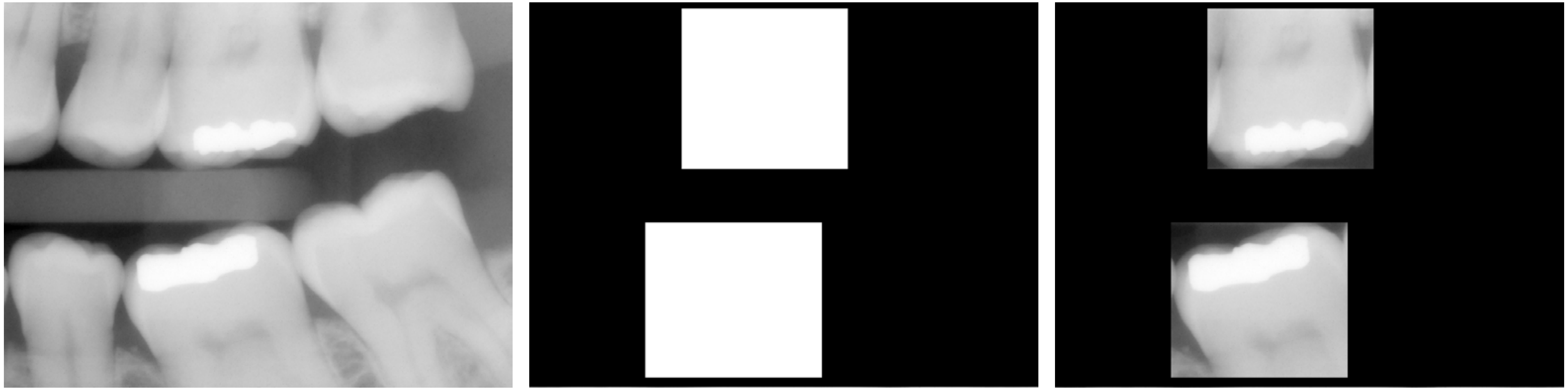
# Image Subtraction



a b
c d

**FIGURE 2.32** Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of the Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

# Image Multiplication



a b c

FIGURE 2.34 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

# Image Division

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

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

# Spatial Operation

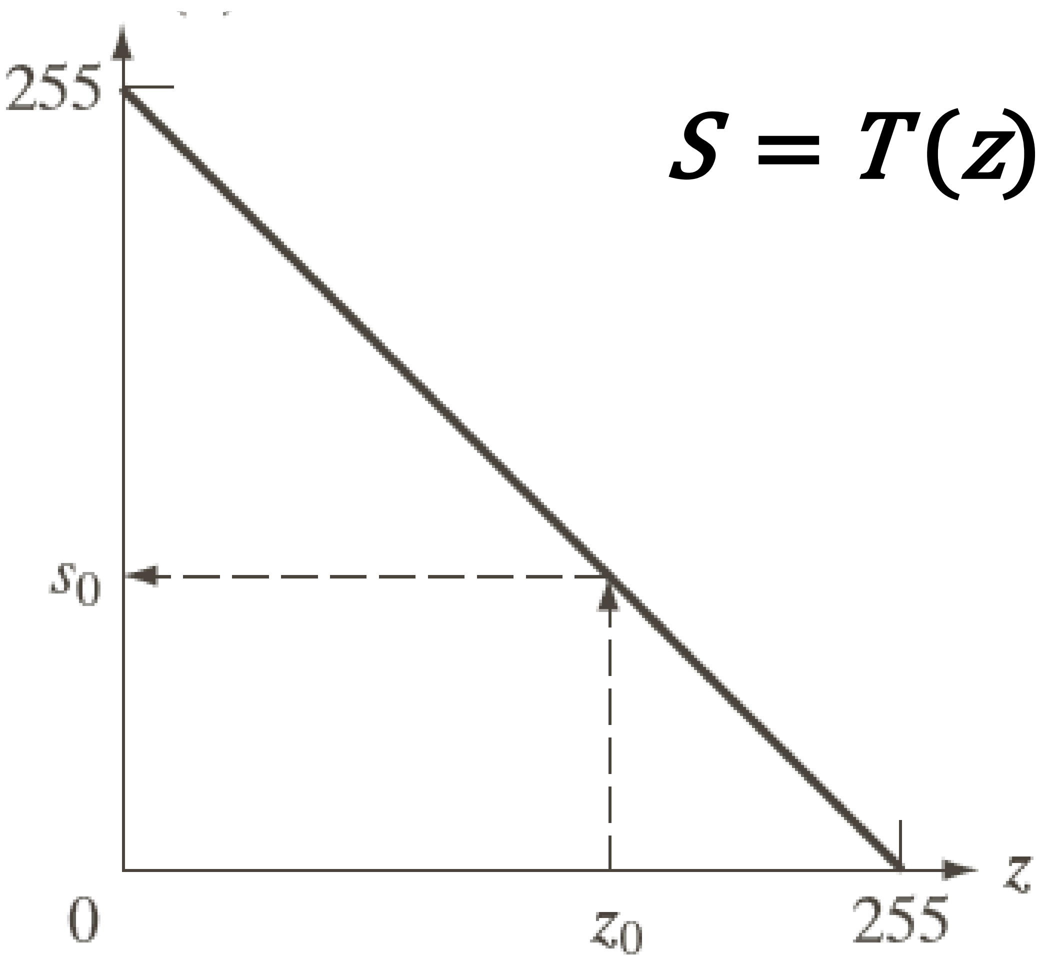
**Performed directly on the pixels of the image**

- Single-pixel operations
- Neighborhood operations
- Image geometry

    *Scale, Rotate, Translate, Mirror, Transpose, Shear, etc.*

- Interpolation

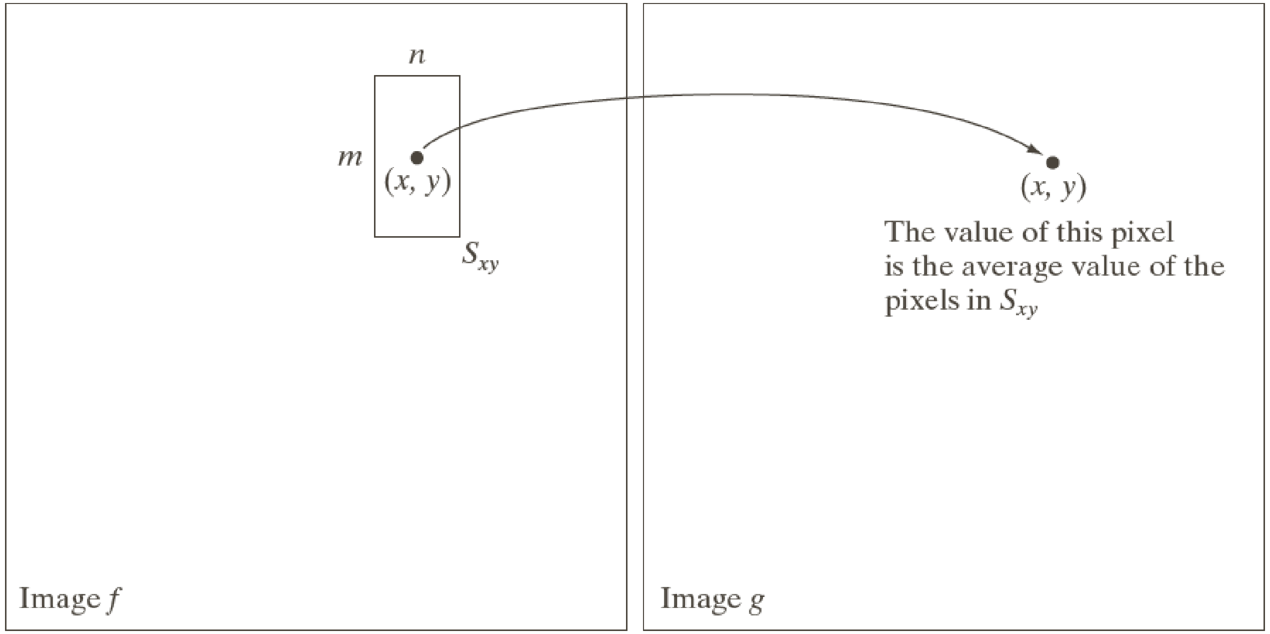
# Single-pixel Operation




$$S = T(z)$$

# Region operation

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




# Image geometry

- **Modify spatial relationship between pixels –** *rubber-sheet*
  - Forward mapping (前向映射): $(x\ y) = T(v\ w)$
  - Inverse mapping (反向映射): $(v\ w) = T^{-1}(x\ y)$
- **Affine transform (仿射变换)**

$$[x\ y\ 1] = [v\ w\ 1]T = [v\ w\ 1]\begin{bmatrix} t_1 & t_4 & 0 \\ t_2 & t_5 & 0 \\ t_3 & t_6 & 1 \end{bmatrix}$$

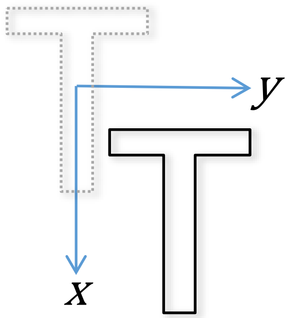
or

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = T\begin{bmatrix} v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$

# Affine Transform

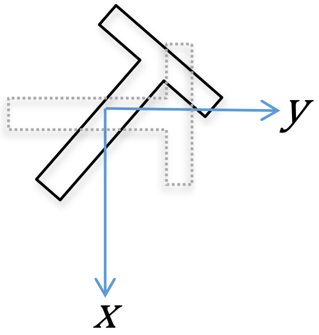
- **Translation**

$$\begin{cases} x = v + \Delta v \\ y = w + \Delta w \end{cases} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta v \\ 0 & 1 & \Delta w \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$



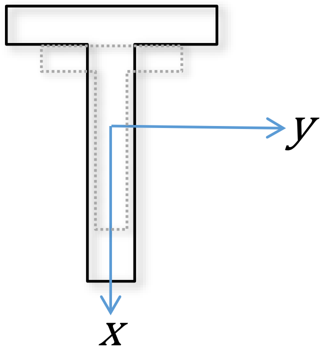

- **Rotation**

$$\begin{cases} x = v\cos\beta - w\sin\beta \\ y = v\sin\beta + w\cos\beta \end{cases} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$




- **Scaling**

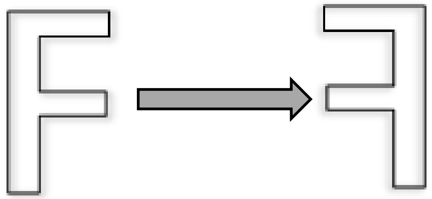
$$\begin{cases} x = c_x v \\ y = c_y w \end{cases} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$



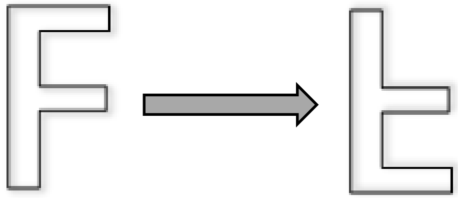
# Affine Transform

- **Mirror**

**Horizontal:** $\begin{cases} x = W - v \\ y = w \end{cases} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & W \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$

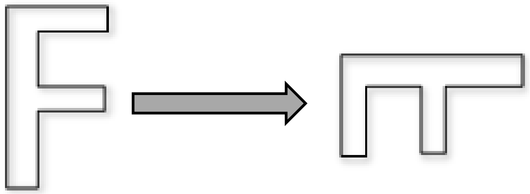


**Vertical:** $\begin{cases} x = v \\ y = H - w \end{cases} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$



- **Transpose**

$$\begin{cases} x = w \\ y = v \end{cases} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$

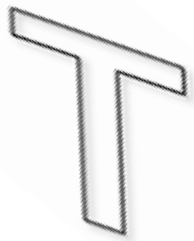


# Affine Transform

- **Shear**

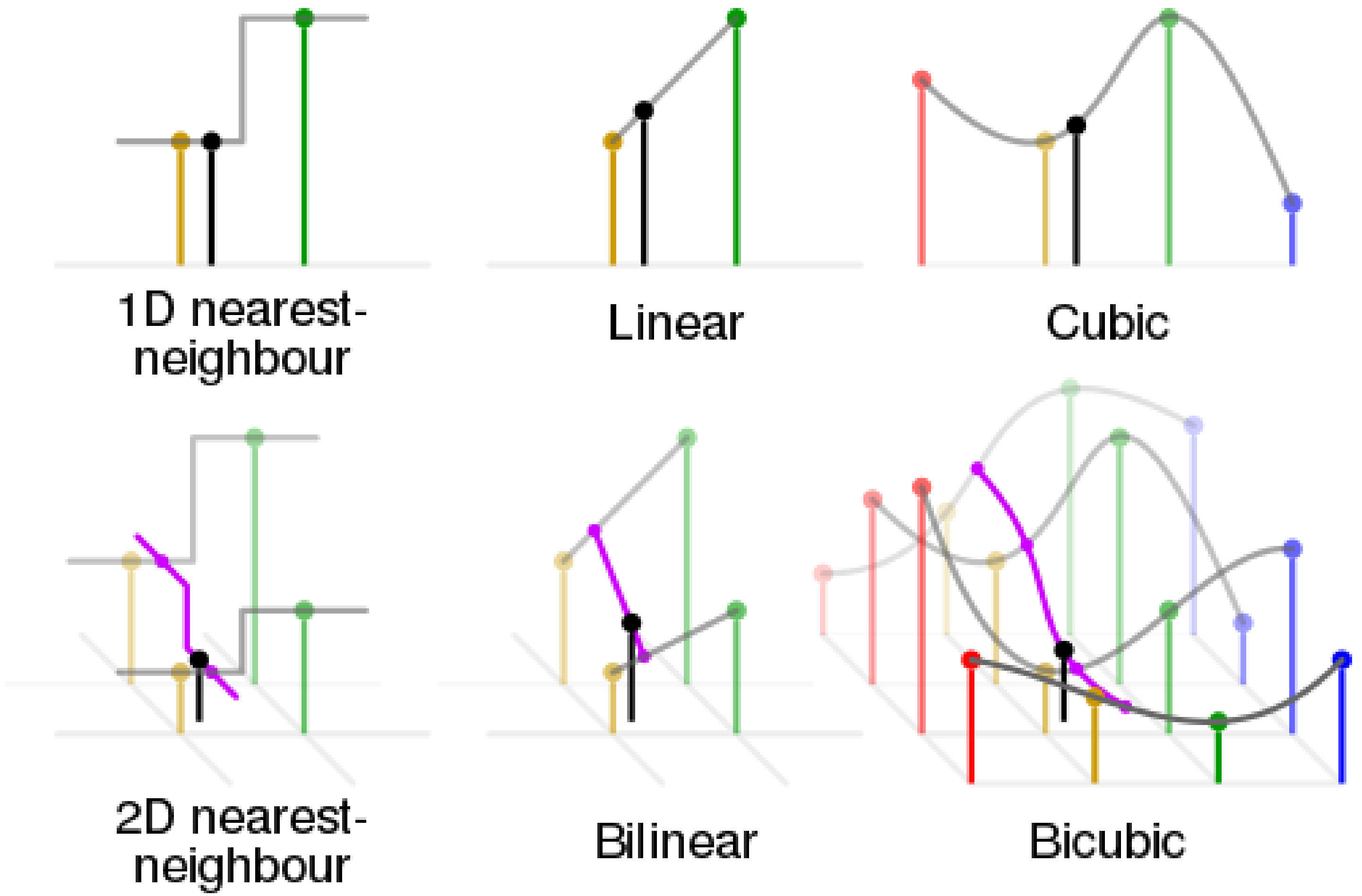
**Horizontal:** $\begin{cases} x = v + c_y w \\ y = w \end{cases} \Longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & c_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$

**Vertical:** $\begin{cases} x = v \\ y = c_x v + w \end{cases} \Longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ c_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$

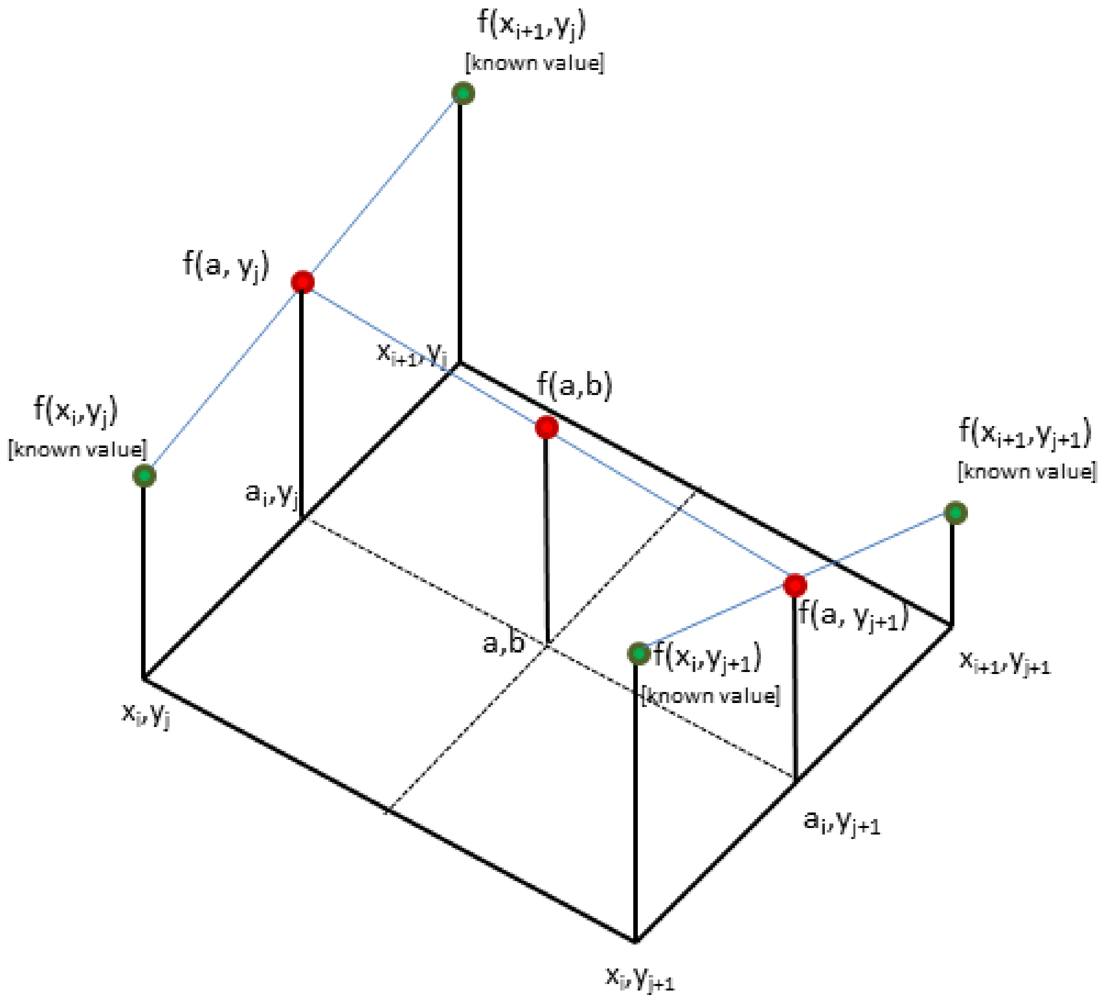


# Image Interpolation (插值)

- Use known data to estimate values at unknown locations
- A resampling method
- Intensity interpolation



# Bilinear interpolation



# Interpolation

a b c
d e f

Image interpolation:

interpolate the image from 72dpi to 1250 dpi using (a)nearest neighbor method, (b) Bilinear method, (c) Bicubic method;

interpolate the image from 150dpi to 1250 dpi using (a)nearest neighbor method, (b) Bilinear method, (c) Bicubic method;

# Image registration (图像配准)

- 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
- Tie point (约束点): $x = c_1 v + c_2 w + c_3 vw + c_4;\ y = c_5 v + c_6 w + c_7 vw + c_8$
- For large number of tie points
  - quadrilateral subimage formed by a group of 4 tie points
  - More complex model: polynomials fitted by least squares algorithms

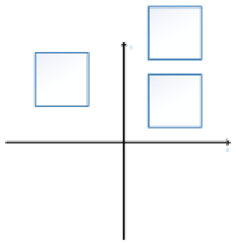


上海科技大学 ShanghaiTech University

# Image reconstruction (图像重建)

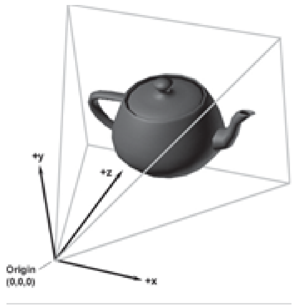
**Some useful coordinate spaces**

1. World Space（世界空间）



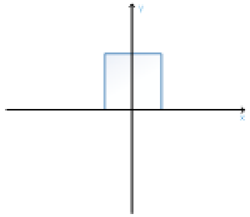
The *world coordinate system* is a special coordinate system that establishes the "global" reference frame for all other coordinate systems to be specified.

2. Camera/View Space（摄像机/观察空间）



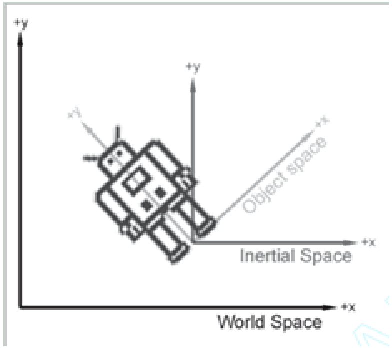

Camera space is the coordinate space associated with an observer.

3. Object Space（物体空间）



*Object space* is the coordinate space associated with a particular object. Every object has its own independent object space. For example, Fig.3 shows the image self-coordinate space

4. Inertial Space（惯性空间）

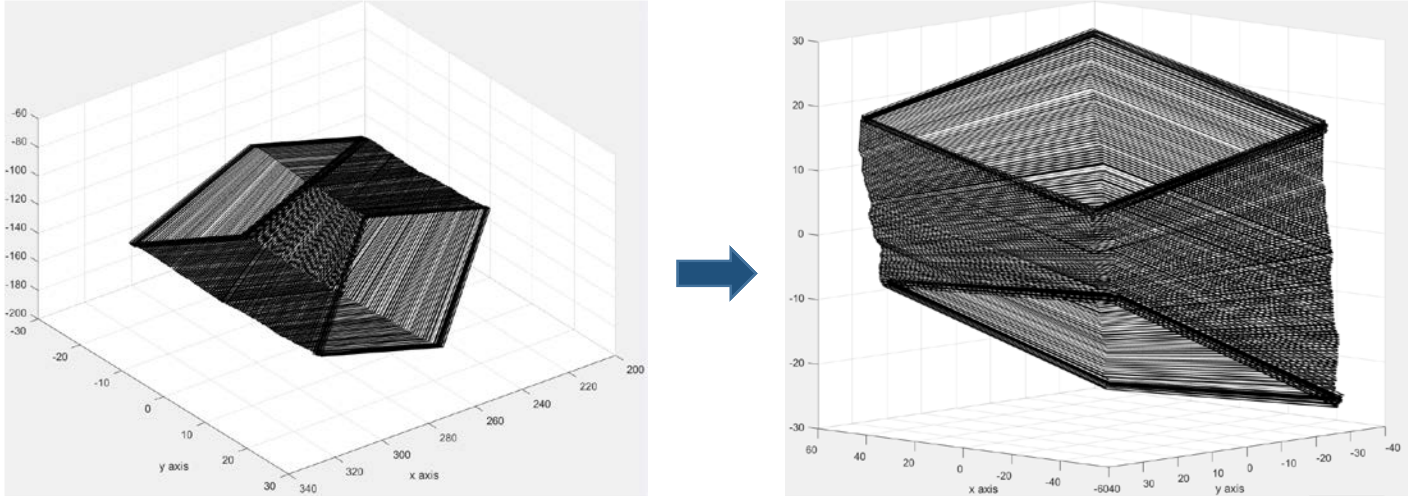



The origin of *inertial space* is the same as the origin of the *object space*, and the axes of *inertial space* are parallel with the axes of *world space*. Fig. 4 illustrates this principle in 2D.



上海科技大学 ShanghaiTech University

# Image reconstruction (图像重建)




$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$