Lecture 3 – Digital Image Processing

This lecture will cover:

– Digital Image

– Basic Image Operation

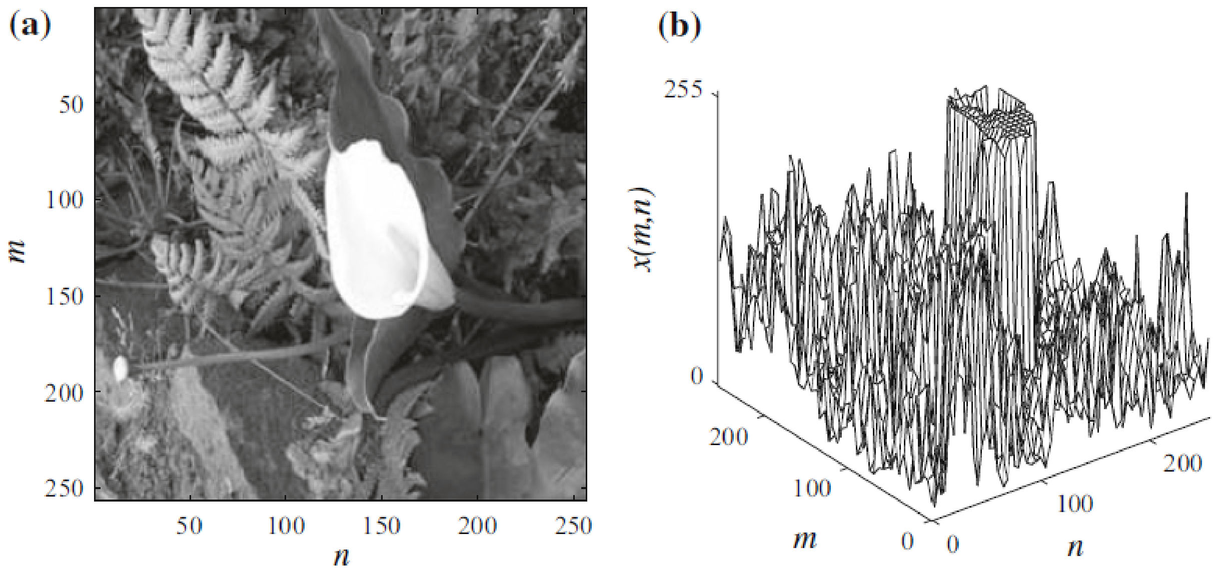
- Array and Matrix Operation
- Vector and Matrix Operation
- Linear and Nonlinear Operation
- Set and Logical Operation
- Arithmetic Operation

– Spatial Operation

– Image filtering

Digital image

- **A visual representation in form of a function *f*(*x*,*y*), where**
  - *f* is related to the intensity or brightness (color) at point
  - (*x*, *y*) are spatial coordinates
  - *x*, *y*, and the amplitude of *f* are finite and discrete quantities



(a) A 256X256 image with 256 gray levels; (b) its amplitude profile

Matrix Representation

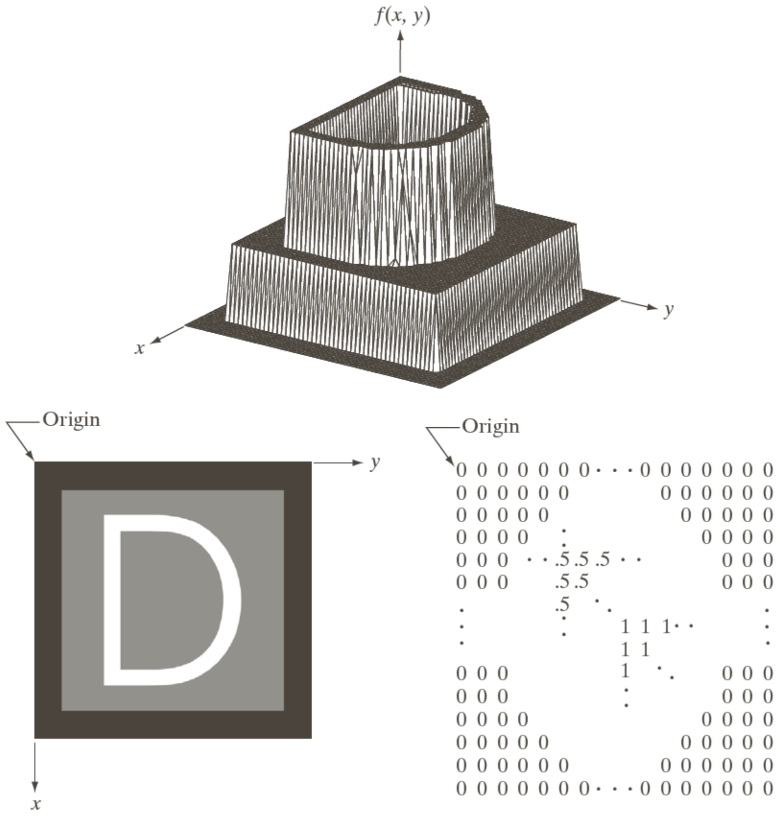
Three basic ways to represent *f*(*x*, *y*)

- Plot of function: *difficult to view and interpret*
- Visual intensity array: *for view*
- numerical array: *for processing and algorithm development*

$$[f(x,y)] = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \ddots & \cdots & \vdots \\ f(M-1,0) & f(M-1,1)\cdots & \cdots & f(M-1,N-1) \end{bmatrix}$$

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \ddots & \cdots & \vdots \\ a_{M-1,0} & a_{M-1,1}\cdots & \cdots & a_{M-1,N-1} \end{bmatrix}$$

$$\text{Intensity level} \quad L = 2^k, \ \text{then } b = M \times N \times k$$



Discrete Fourier Transform （离散傅里叶变换）

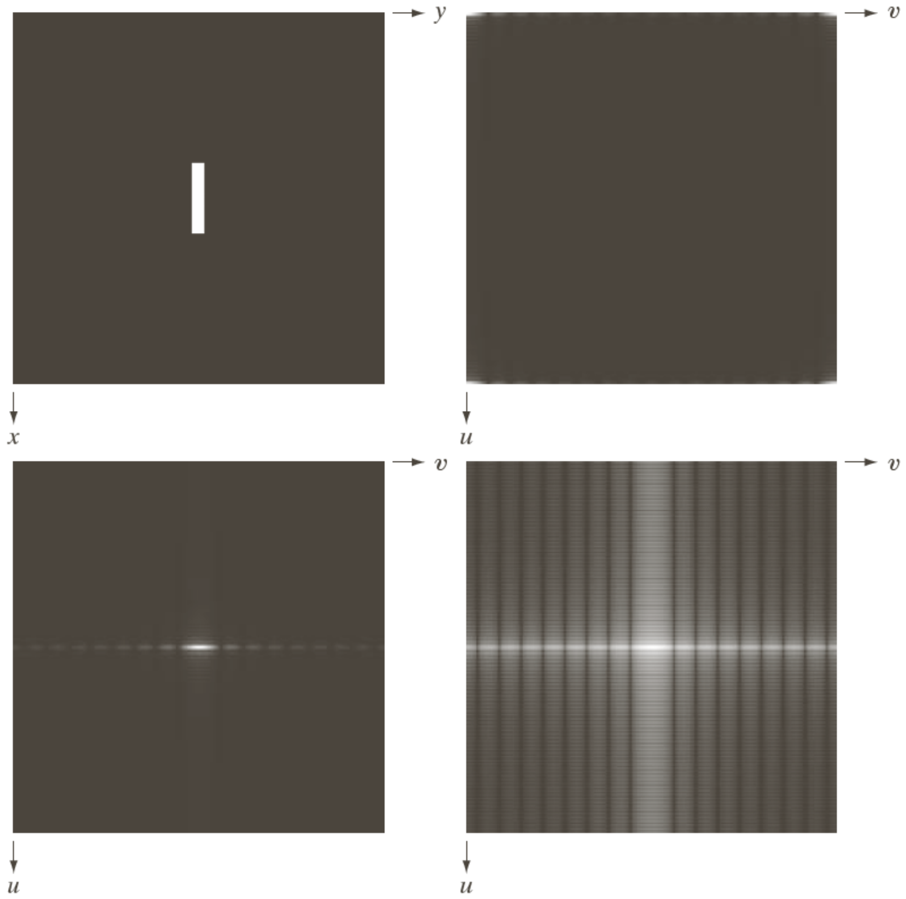
2D Discrete Fourier Transform (DFT)

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

2D Inverse Discrete Fourier Transform (IDFT)

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

- $f(x,y)$: M∗N input image
- (x,y): spatial variables
- (μ,ν): frequency variables, defines the continuous frequency domain



a b
c d

FIGURE 4.24
(a) Image. (b) Spectrum showing bright spots in the four corners. (c) Centered spectrum. (d) Result showing increased detail after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The coordinate convention used throughout the book places the origin of the spatial and frequency domains at the top left.

Spectrum（频谱）

- **2D DFT in polar form:** $F(u,v) = |F(u,v)|e^{-j\Phi(u,v)}$ **, then**
  - Fourier spectrum（频谱）: $|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{\frac{1}{2}}$
  - Phase angle（相角）： $\Phi(u,v) = \arctan\frac{I(u,v)}{R(u,v)}$
  - Power spectrum(功率谱): $P(u,v) = |F(u,v)|^2$
  - DC component(直流分量): $F(0,0) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) = MN\overline{f(x,y)}$
- **Convolution theorem (卷积定理)**

$$f(x,y) \star h(x,y) \Leftrightarrow F(u,v)H(u,v) \quad \text{or} \quad f(x,y)\,h(x,y) \Leftrightarrow F(u,v) \star H(u,v)$$

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**

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

- **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**

$$g = Hf + n$$

Linear and Nonlinear Operation

An operator

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

is linear if

$$\begin{aligned} H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] &= a_i H[f_i(x,y)] + a_j H\left[f_j(x,y)\right] \\ &= a_i g_i(x,y) + a_j g_j(x,y) \end{aligned}$$

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

Set Operation (Coordinates)

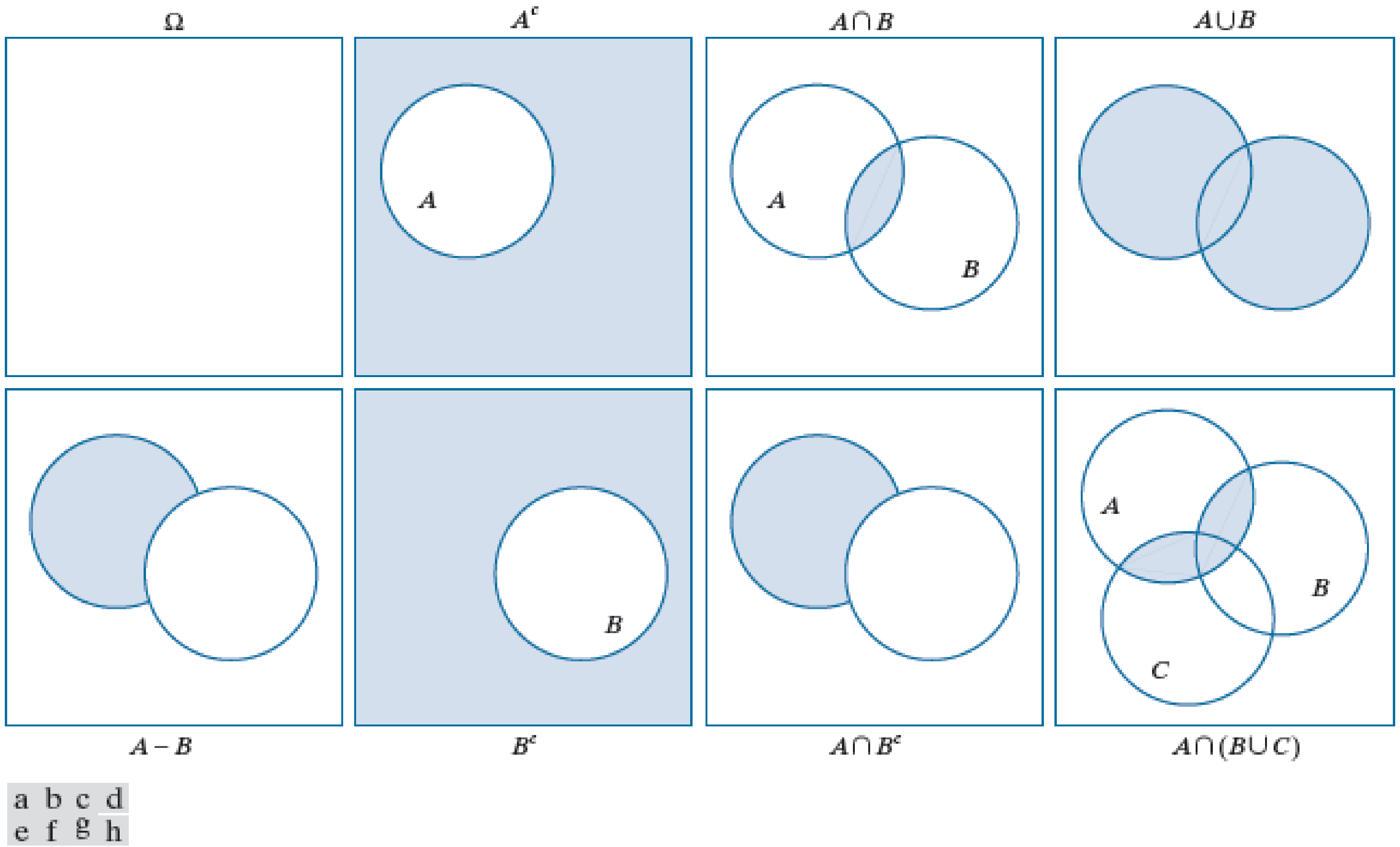


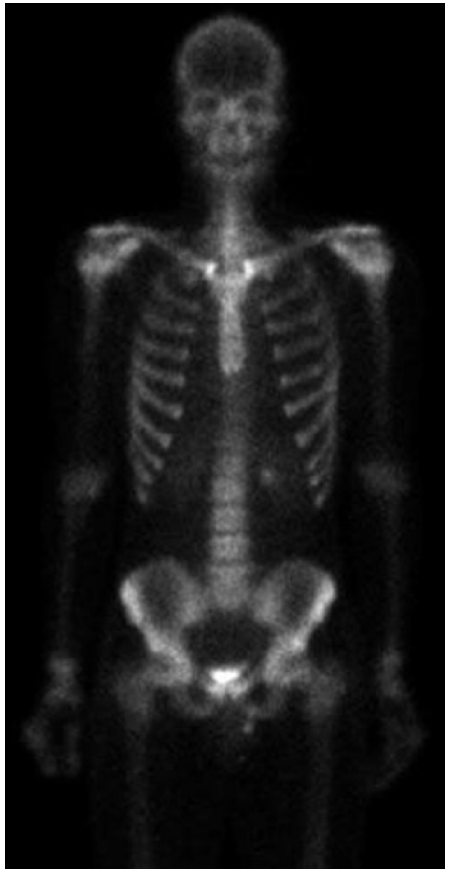
FIGURE 2.35 Venn diagrams corresponding to some of the set operations in Table 2.1. The results of the operations, such as A^c, are shown shaded. Figures (e) and (g) are the same, proving via Venn diagrams that $A - B = A \cap B^c$ [see Eq. (2-40)].

Set Operation (Intensity)

a b c

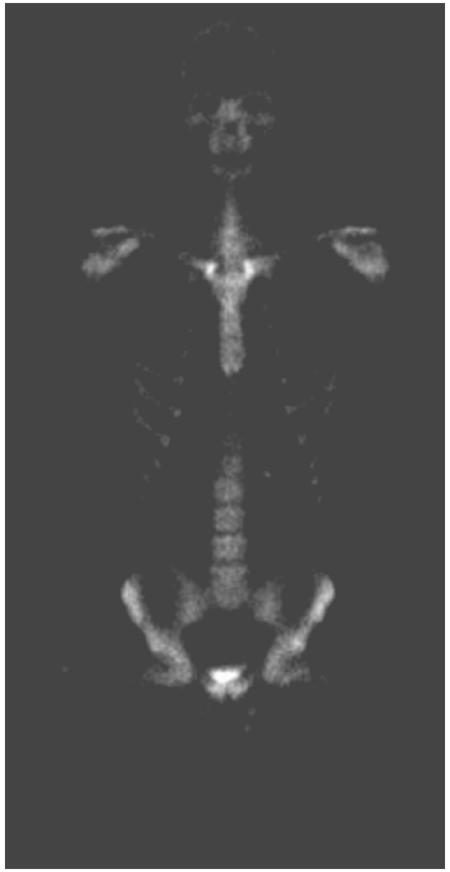
FIGURE 2.36 Set operations involving grayscale images. (a) Original image. (b) Image negative obtained using grayscale set complementation. (c) The union of image (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

(A) Original image



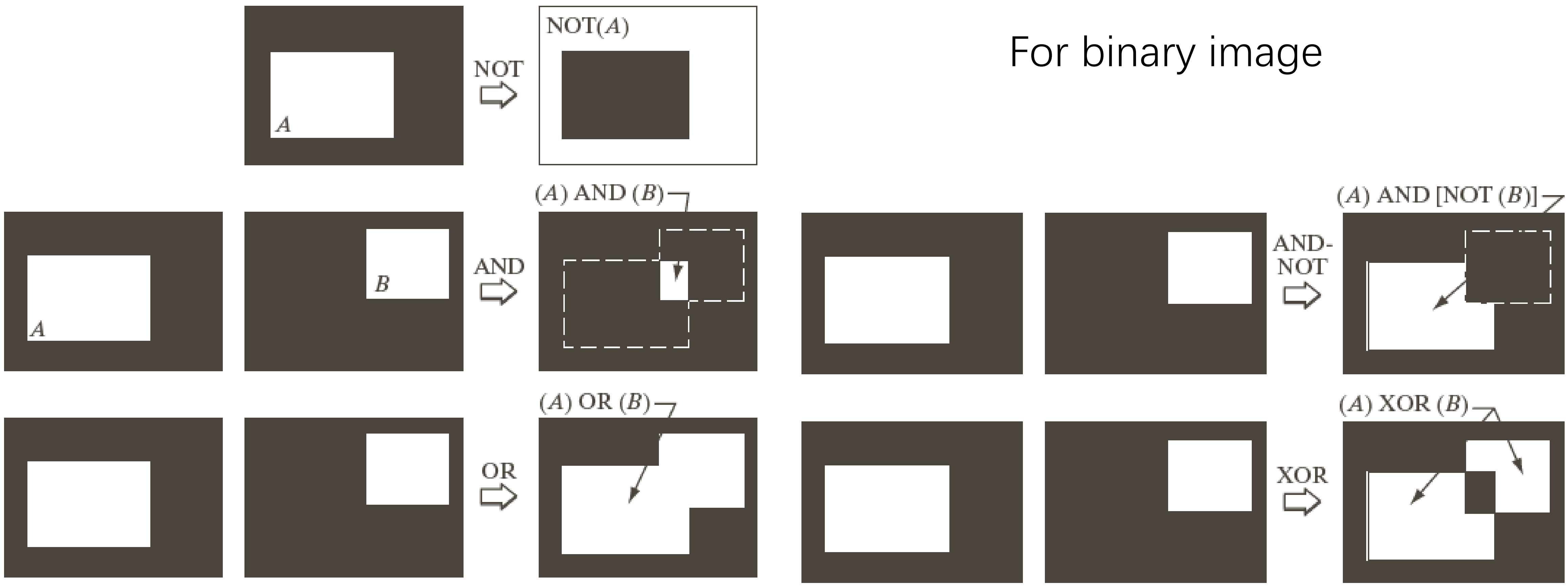
(B) Complement image
$Z = 255 - A$

(C) Union: $A \cup 3\bar{Z} = \{\max(a, 3\bar{Z})\,|a \in A\}$



Logical Operation

For binary image



Arithmetic Operation

- **Addition**

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

- **Subtraction**

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

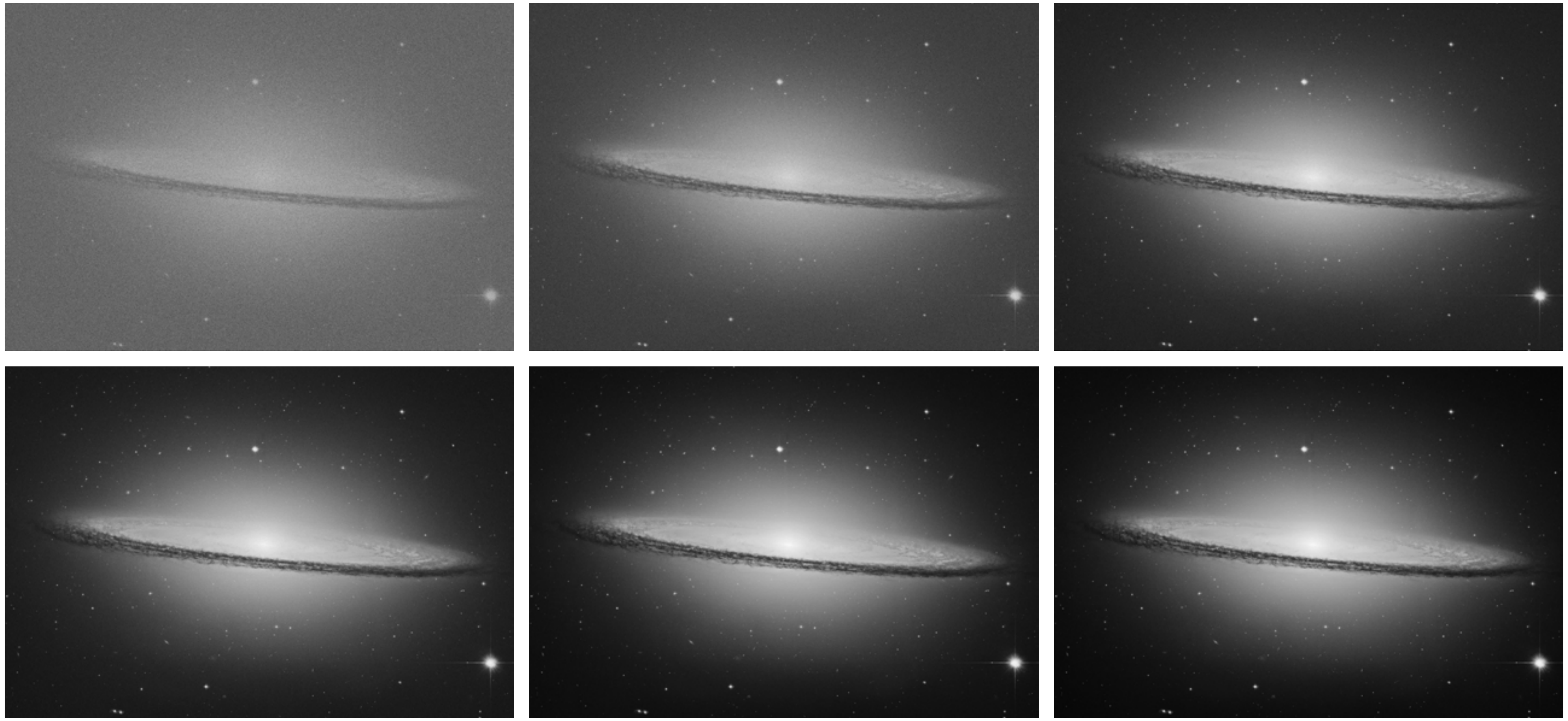
- **Multiplication**

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

- **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 1548×2238 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**

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

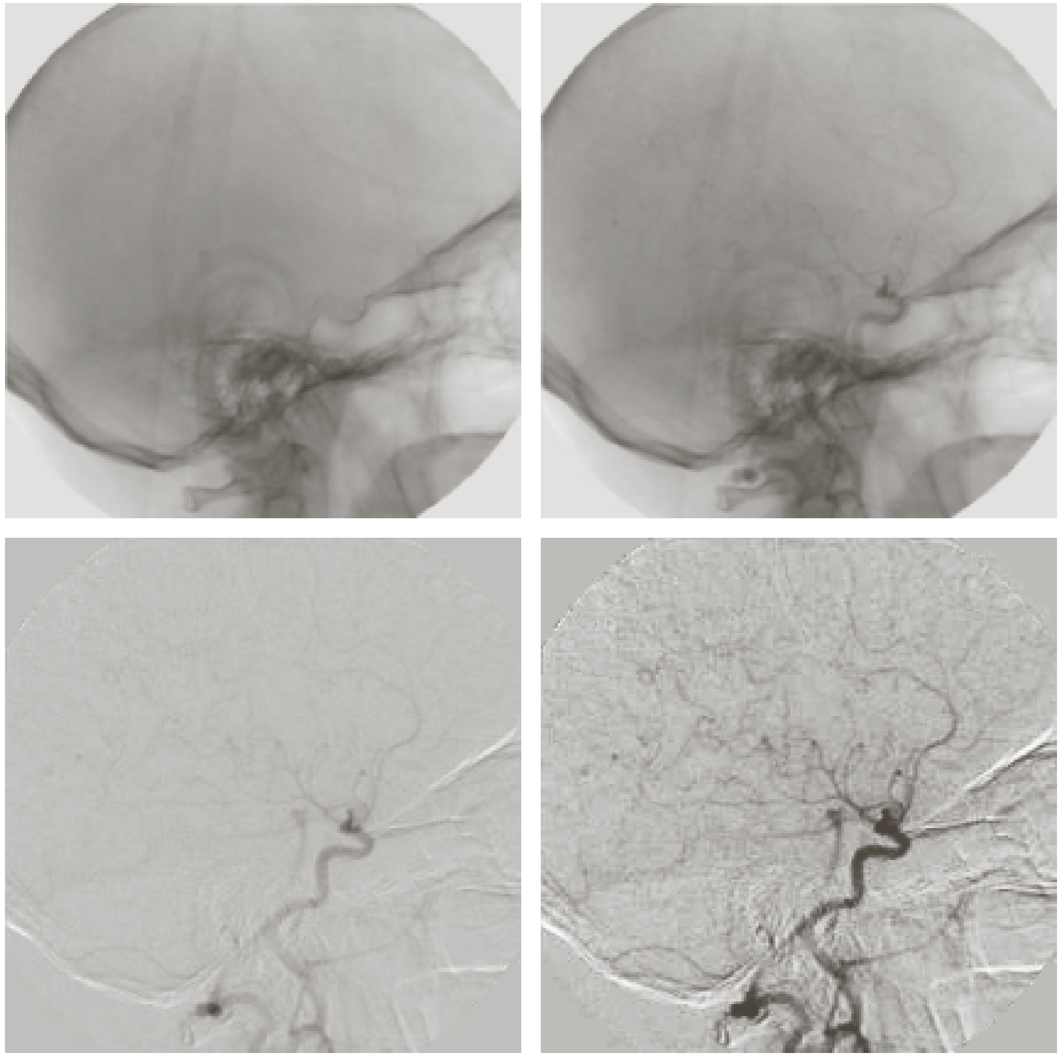
- **Scale**

$$s(x,y) = f(x,y) + g(x,y): \quad \{\min[s(x,y)], \max[s(x,y)]\} = \{0, L_{\max}\}$$

- **Max intensity value**

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

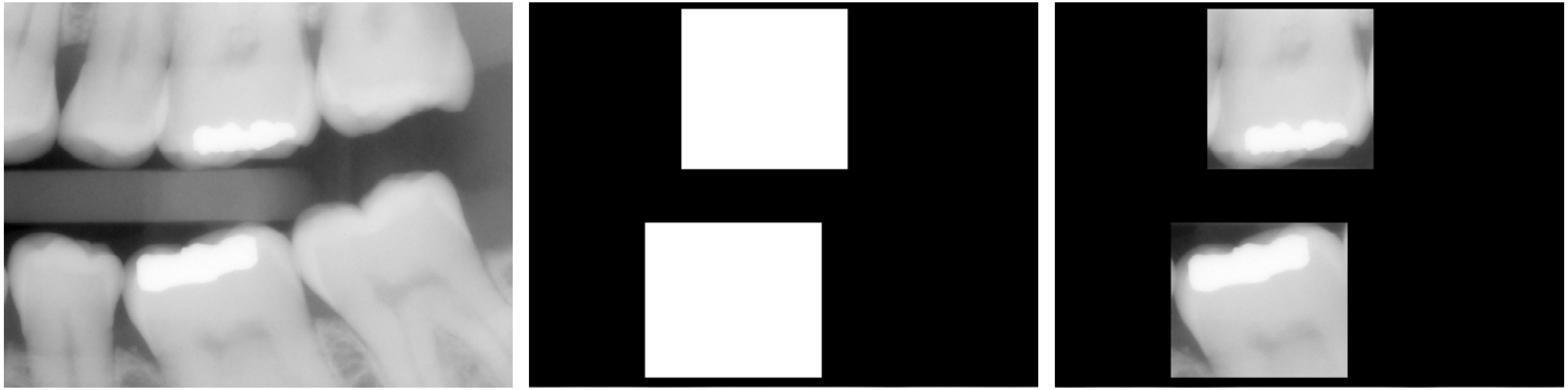
Image Subtraction



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

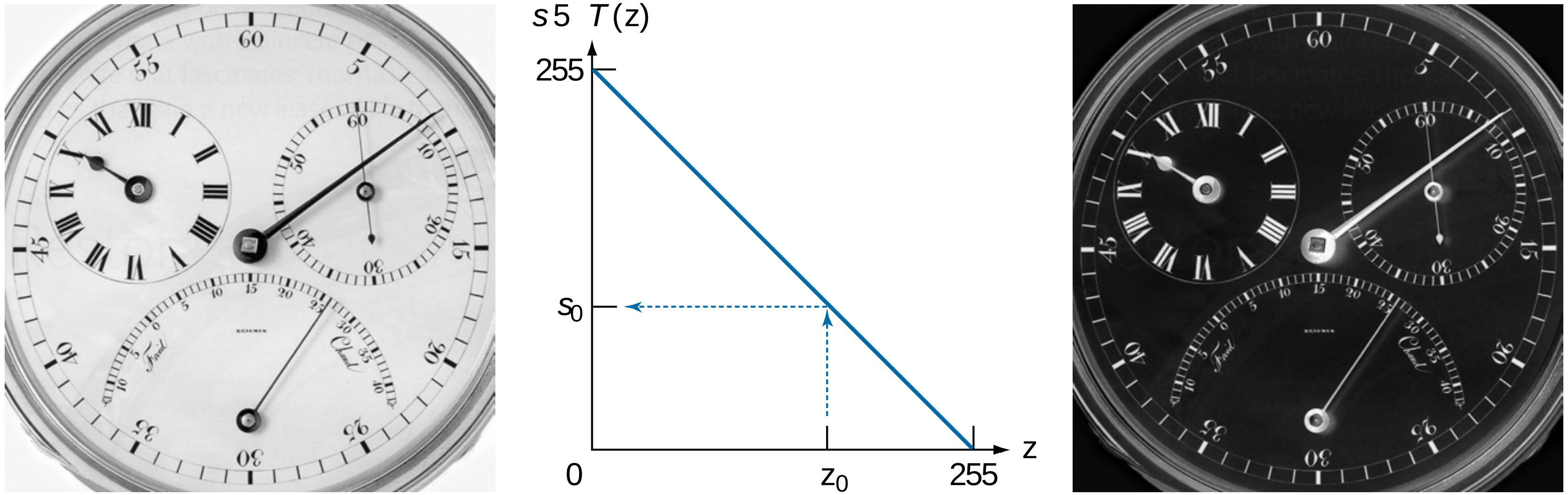
Performed directly on the pixels of the image

- Single-pixel operations
- Neighborhood operations
- Image geometry

  Scale, Rotate, Translate, Mirror, Transpose, Shear, etc.
- Interpolation

Single-pixel Operation

$$S = T(z)$$

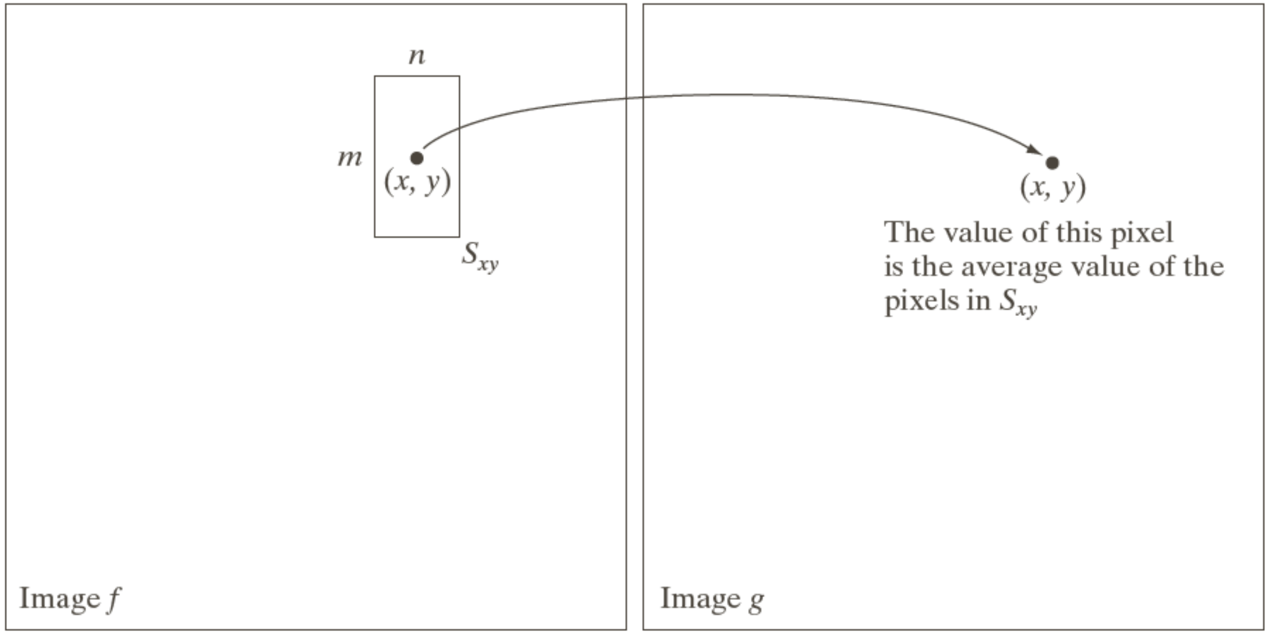


a b c

Figure 2.38 (a) An 8-bit image. (b) Intensity transformation function used to obtain the digital equivalent of a "photographic" negative of an 8-bit image. The arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0. (c) Negative of (a), obtained using the transformation function in (b).

Neighborhood operation

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



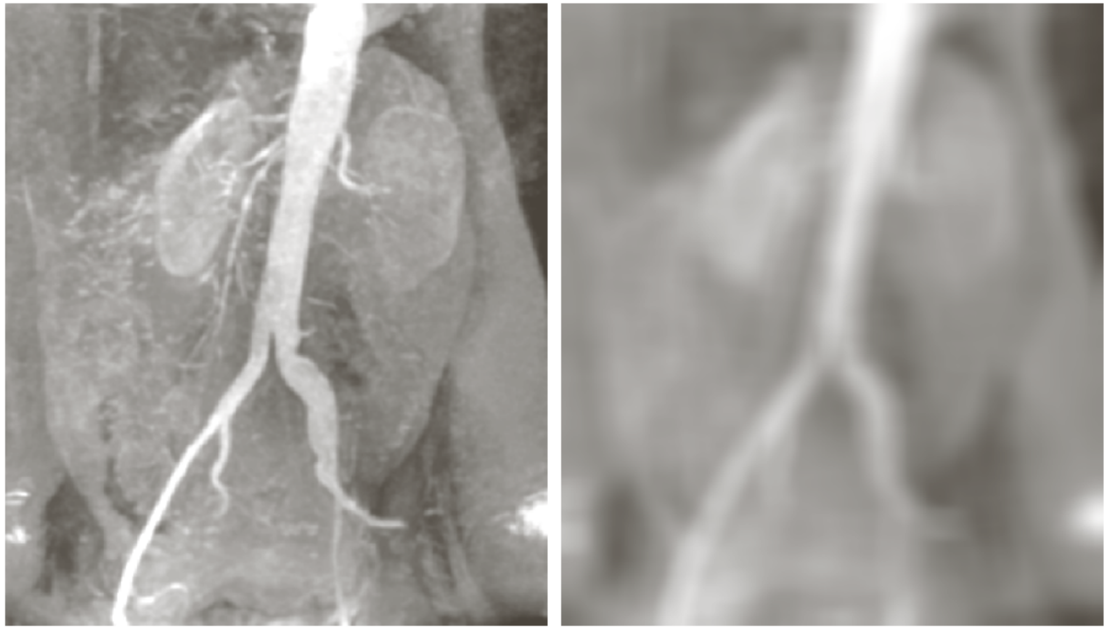


Image geometry

- **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)**

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

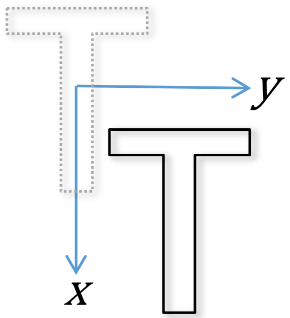
or

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

Affine Transform

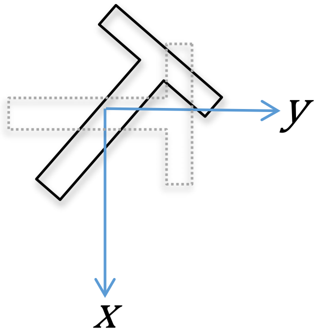
- **Translation**

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



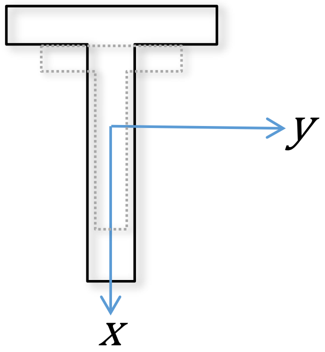
- **Rotation**

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



- **Scaling**

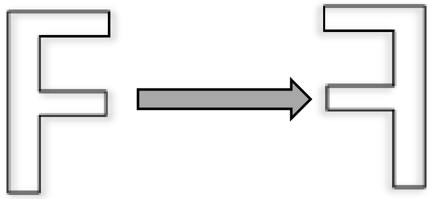
$$\begin{cases} x = c_x v \\ y = c_y w \end{cases} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$$



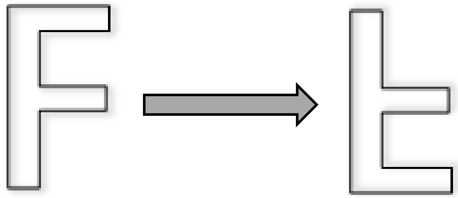
Affine Transform

- **Mirror**

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

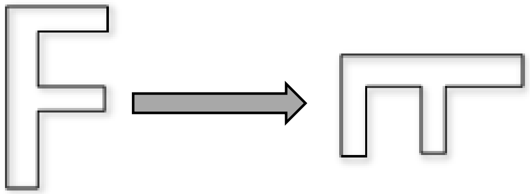


Vertical: $\begin{cases} x = v \\ y = H - w \end{cases} \Longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & H \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}$



- **Transpose**

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



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

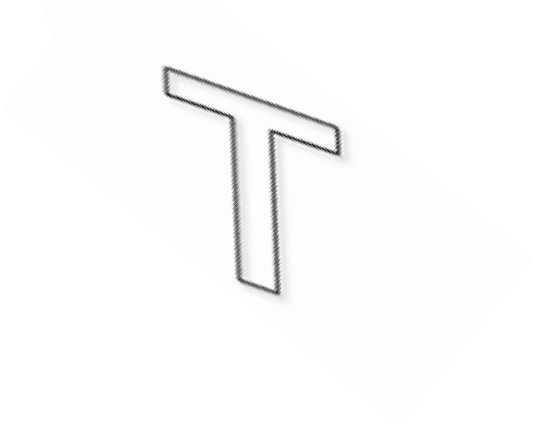
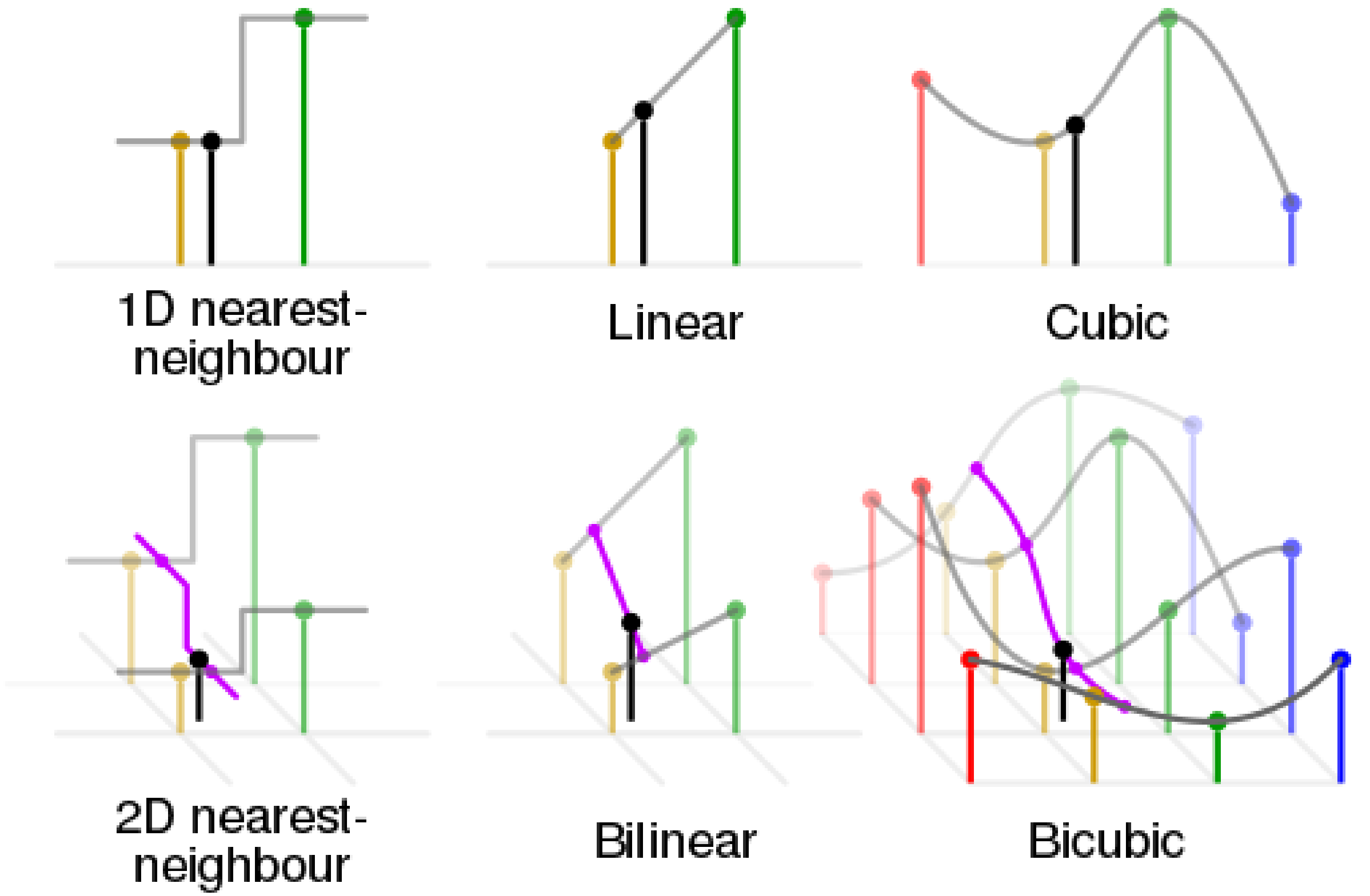


Image Interpolation (插值)

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



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 Filtering

- **High-pass**
  - Small objects and edges
  - Sharpening
  - Improving spatial resolution
  - More noise
  - SNR decrease
- **Low-pass**
  - Smoothing
  - Little effect on spatial resolution
  - Attenuating noise
  - SNR increase

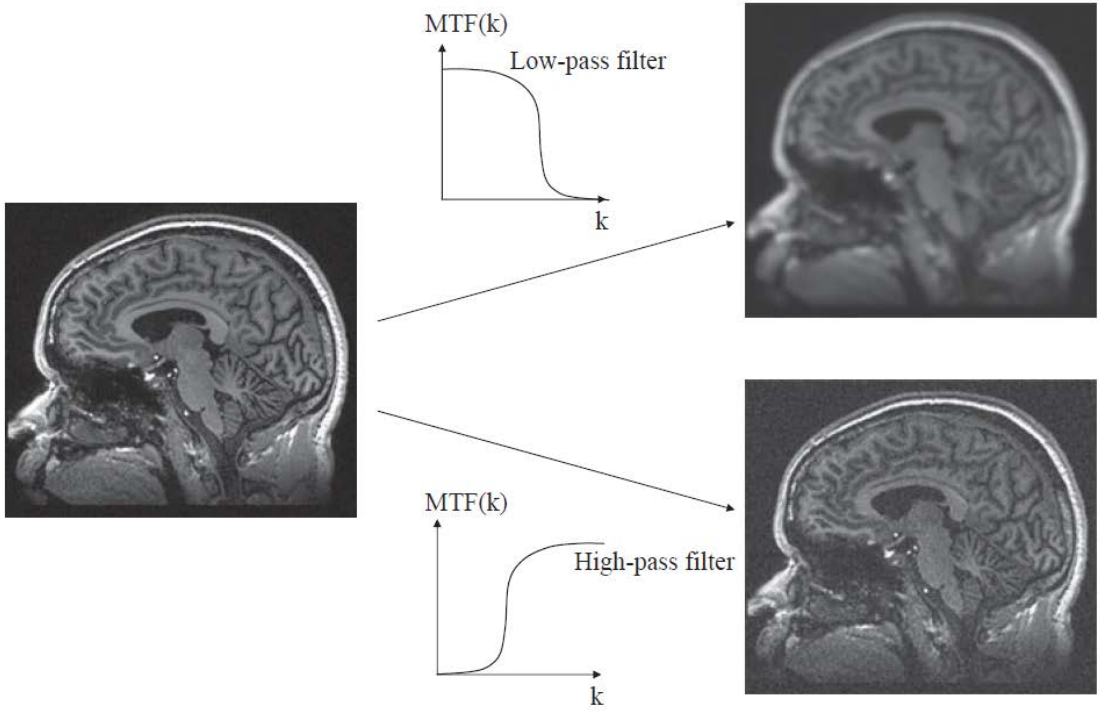


Image Filtering

original image

1	5	3	5	4	6
4	3	32	5	6	9
6	10	4	8	8	7

*

filter

1/12	1/12	1/12
1/12	4/12	1/12
1/12	1/12	1/12

=

filtered image

1	5	3	5	4	6
4	a	b	c	d	9
6	10	4	8	8	7

$$\left\{\begin{array}{l}
a=(1)(1/12)+(5)(1/12)+(3)(1/12)+(4)(1/12)+(3)(4/12)+(32)(1/12)+(6)(1/12)+(10)(1/12)+(4)(1/12)=6.4 \\
b=(5)(1/12)+(3)(1/12)+(5)(1/12)+(3)(1/12)+(32)(4/12)+(5)(1/12)+(10)(1/12)+(4)(1/12)+(8)(1/12)=14.3 \\
c=(3)(1/12)+(5)(1/12)+(4)(1/12)+(32)(1/12)+(5)(4/12)+(6)(1/12)+(4)(1/12)+(8)(1/12)+(8)(1/12)=7.5 \\
d=(5)(1/12)+(4)(1/12)+(6)(1/12)+(5)(1/12)+(6)(4/12)+(9)(1/12)+(8)(1/12)+(8)(1/12)+(7)(1/12)=6.3
\end{array}\right\}$$

1	5	3	5	4	6
4	6.4	14.3	7.5	6	9
6	10	4	8	8	7

filtered image