Digital Image Processing Operations

Overview

- Image operations.
- Point operations.
  - Arithmetic operations. (Add, Sub, Mul, Div).
  - Logical operations.(AND, OR, NOT).
- Neighborhood operations.
  - Averaging filter (mask).
  - Various Neighborhood operations.
- Geometric operations.
  - Translation.
  - Scaling.
  - Rotation.
  - Shearing.
  - Zooming.

Classification of Image Operations

- One way of classification is

Point → Those whose output value at a specific coordinate depends only on the input value.

Local → Those output value at a specific coordinate depends on the input value in the neighborhood of that pixel.

Global → Those output value at a specific coordinate depends on all the values in the input image

Image Vs Array Operations

Image operations are array operations. These operations are done on a pixel-by-pixel basis. Array operations are different from matrix operations. For example, consider two images

$$F_1 = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \text{and } F_2 = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

The multiplication of F_1 and F_2 is element-wise, as follows:

$$F_1 \times F_2 = \begin{pmatrix} AE & BF \\ CG & HD \end{pmatrix}$$

In addition, one can observe that $F_1 \times F_2 = F_2 \times F_1$, whereas matrix multiplication is clearly different, since in matrices, $A \times B \neq B \times A$. By default, image operations are array operations only.

Arithmetic operations - Addition

Two images can be added in a direct manner, as given by

$$g(x, y) = f_1(x, y) + f_2(x, y)$$

Table 3.1 Data type and allowed ranges

S. no.	Data type	Data range
1	Uint8	0–255
2	Uint16	0–65,535
3	Uint32	0–4,29,49,67,295
4	Uint64	0–1,84,46,74,40,73,70,95,51,615

Similarly, it is possible to add a constant value to a single image, as follows:

$$g(x, y) = f_1(x, y) + k$$

➢**To create double exposure / Superimposing an image on another image.**

➢**To increase the brightness of an image.**

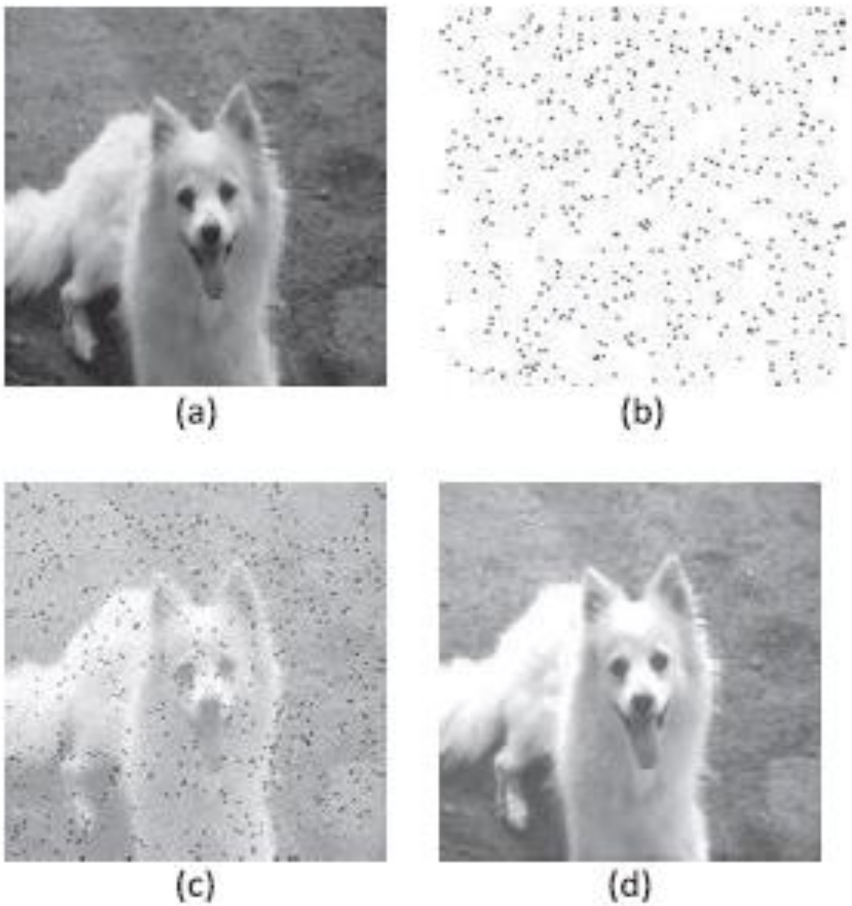


Fig. 3.14 Results of the image addition operation (a) Image 1 (b) Image 2 (c) Addition of images 1 and 2 (d) Addition of image 1 and constant 50

Image Subtraction

The subtraction of two images can be done as follows. Consider

$$g(x, y) = f_1(x, y) - f_2(x, y)$$

where $f_1(x, y)$ and $f_2(x, y)$ are two input images and $g(x, y)$ is the output image. To avoid negative values, it is desirable to find the modulus of the difference as

$$g(x, y) = \left| f_1(x, y) - f_2(x, y) \right|$$

- **Background elimination.**

- **Brightness reduction**

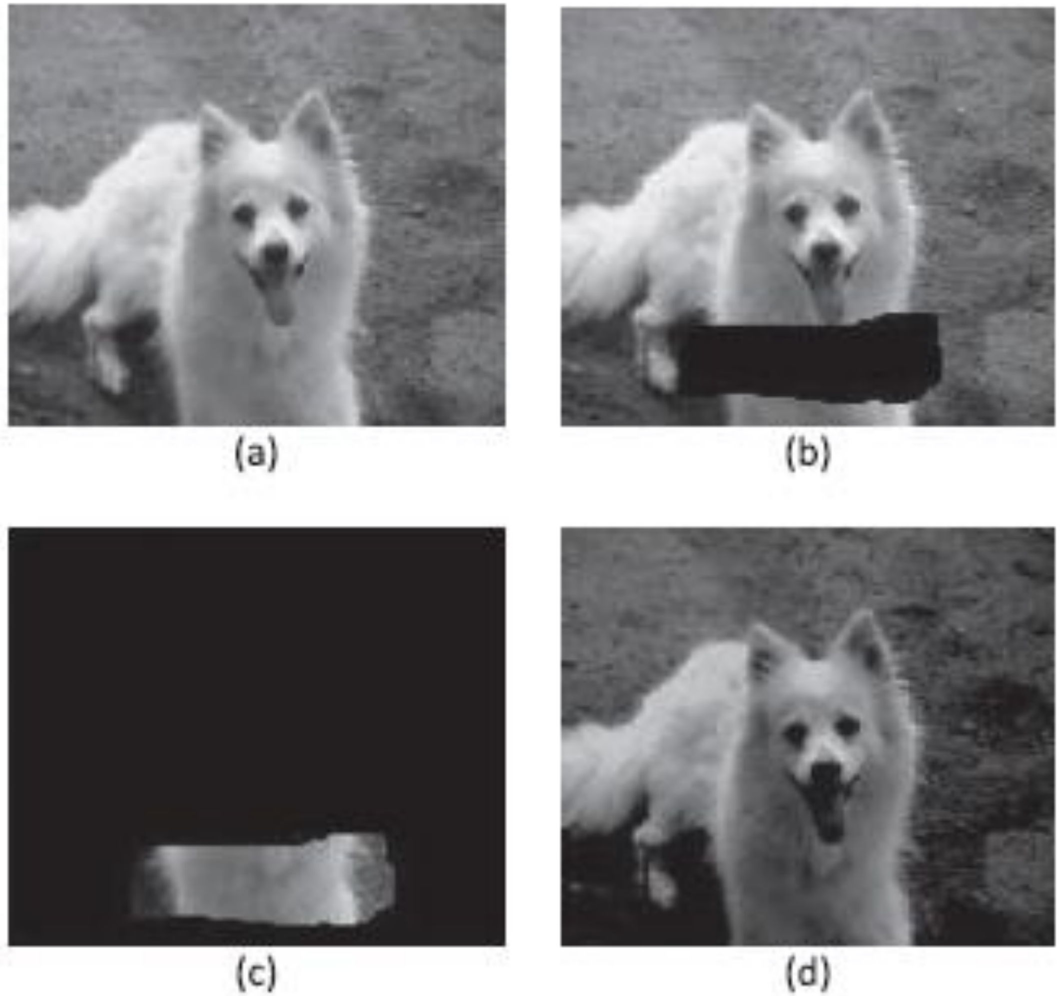


Fig. 3.15 Results of the image subtraction operation (a) Image 1 (b) Image 2 (c) Subtraction of images 1 and 2 (d) Subtraction of constant 50 from image 1

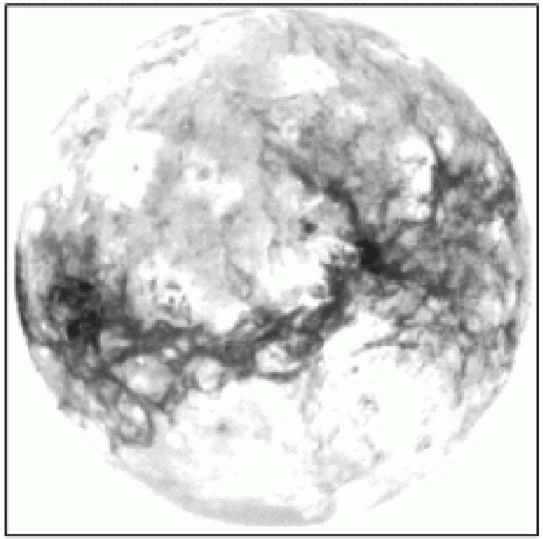
Image Multiplication

$$g(x, y) = f_1(x, y) \times f_2(x, y)$$

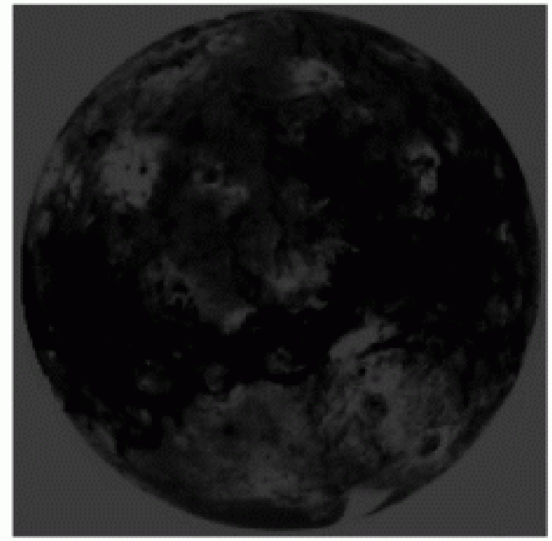
$$g(x, y) = f(x, y) \times k$$

Result of multiplication operation (image × 1.25) resulting in good contrast

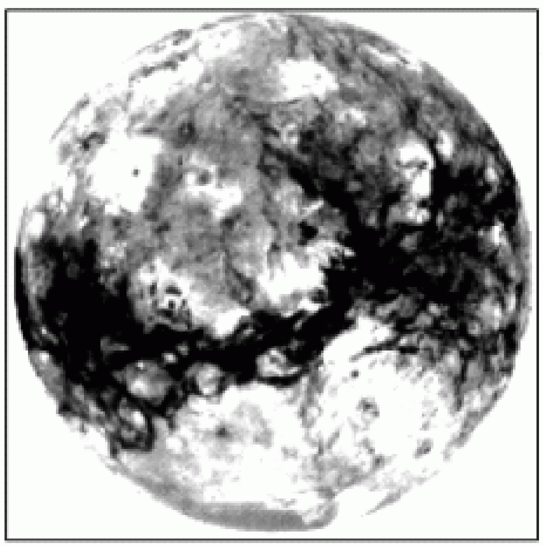
- Increase contrast.
- Designing filter masks.



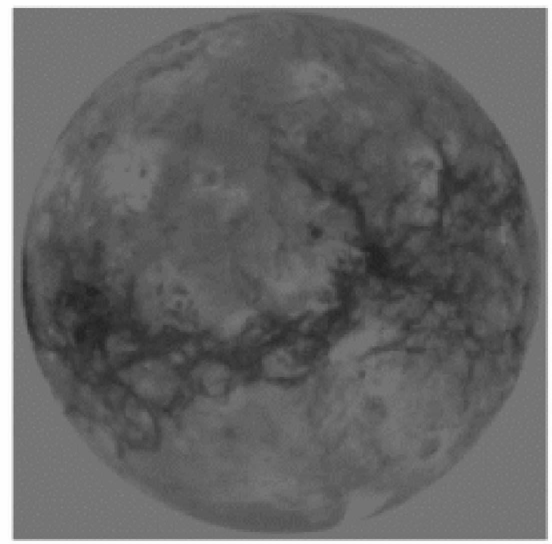
a. Brightness too high



b. Brightness too low



c. Contrast too high



d. Contrast too low

FIGURE 23-10
Brightness and contrast adjustments. Increasing the *brightness* makes every pixel in the image becomes lighter. In comparison, increasing the *contrast* makes the light areas become lighter, and the dark areas become darker. These images show the effect of misadjusting the brightness and contrast.

Image Division

Similar to the other operations, division can be performed as

$$g(x, y) = \frac{f_1(x, y)}{f_2(x, y)}$$

where $f_1(x, y)$ and $f_2(x, y)$ are two input images and $g(x, y)$ is the output image.

$$g(x, y) = \frac{f(x, y)}{k}, \text{ where } k \text{ is a constant.}$$

- **Decrease in contrast.**

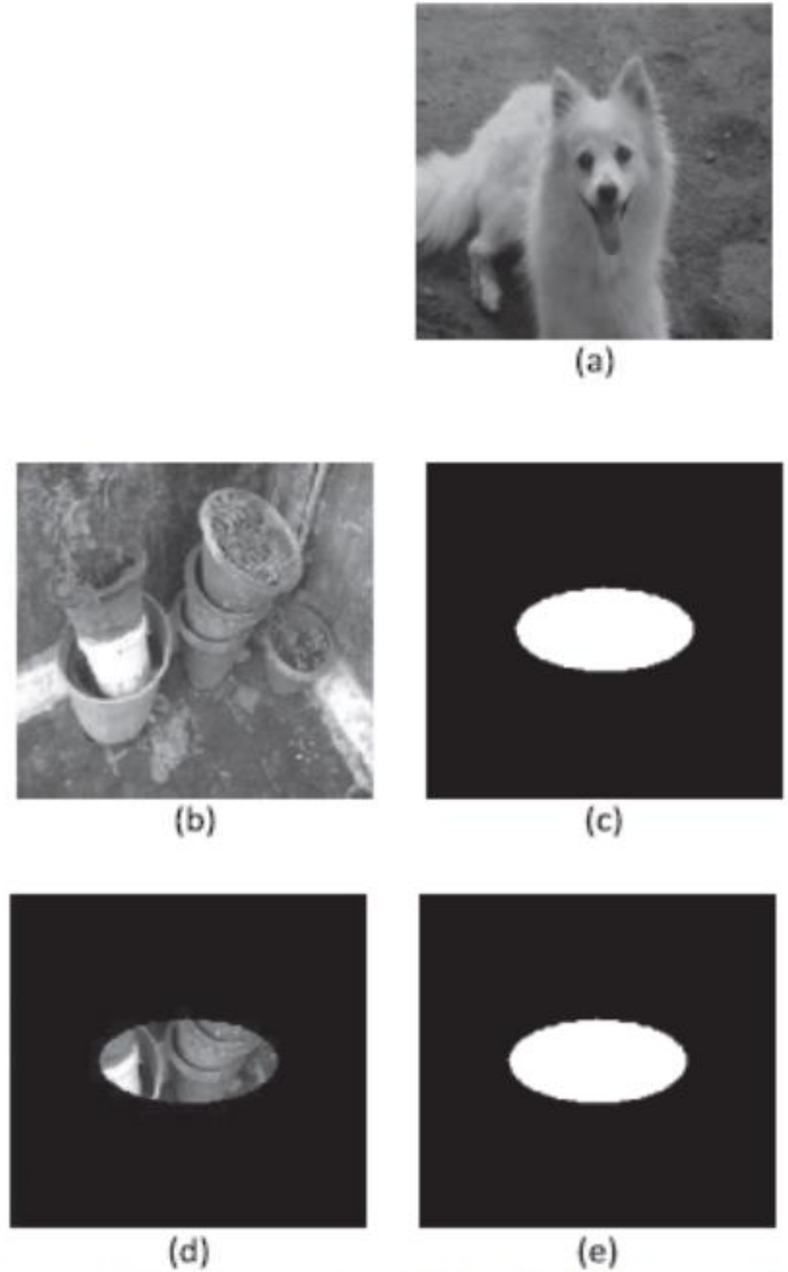
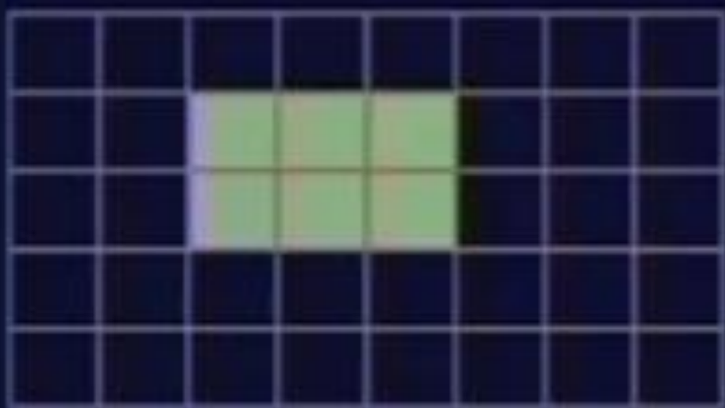


Fig. 3.17 Image division operation (a) Result of the image division operation (image/1.25) (b) Image 1 (c) Image 2 used as a mask (d) Image 3 = image 1 × image 2 (e) Image 4 = image 3/image 1

Logical Operations

1. AND/NAND
2. OR/NOR
3. EXOR/EXNOR
4. Invert/Logical NOT

Arithmetic / Logical Operation



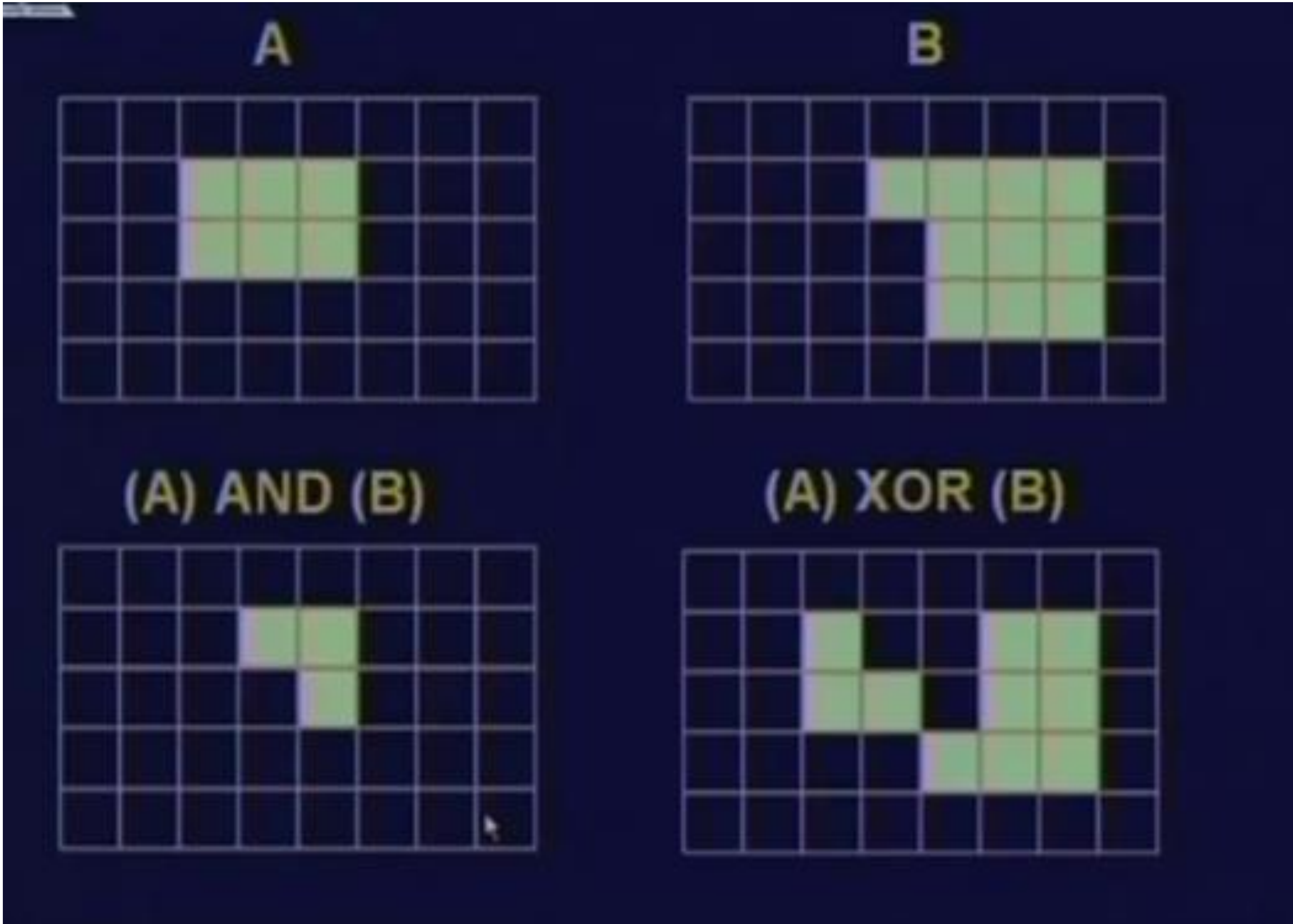
A

NOT (A)

Image Negative

Original Image

Image negative


A
B
(A) AND (B)
(A) XOR (B)

Neighborhood Operations

The value assigned to a pixel is a function of its gray label and the gray labels of its neighbors.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

$Z = 1/9\ (Z_1 + Z_2 + Z_3 + \ldots\ldots + Z_9) = \text{Average}$

More general form

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

$$Z = W_1Z_1 + W_2Z_2 + \ldots\ldots\ldots + W_9Z_9$$

$$= \sum_{i=1}^{9} W_iZ_i$$

Same as averaging if $W_i = 1/9$

Neighborhood Operations

Various important operations can be Implemented by proper selection of Coefficients W_i

----- Noise filtering

----- Thinning

----- Edge detection

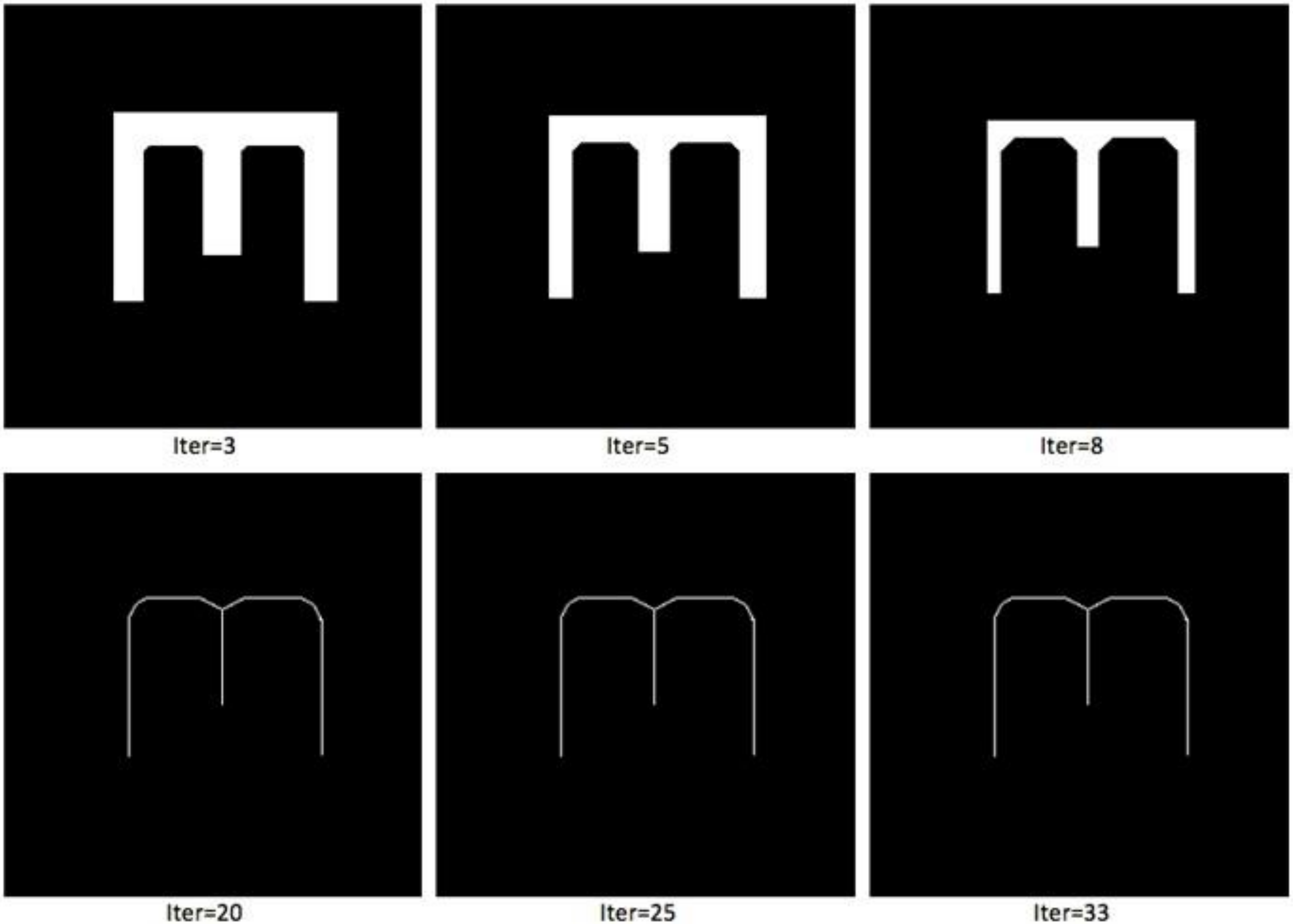
etc…

Noise Filtering

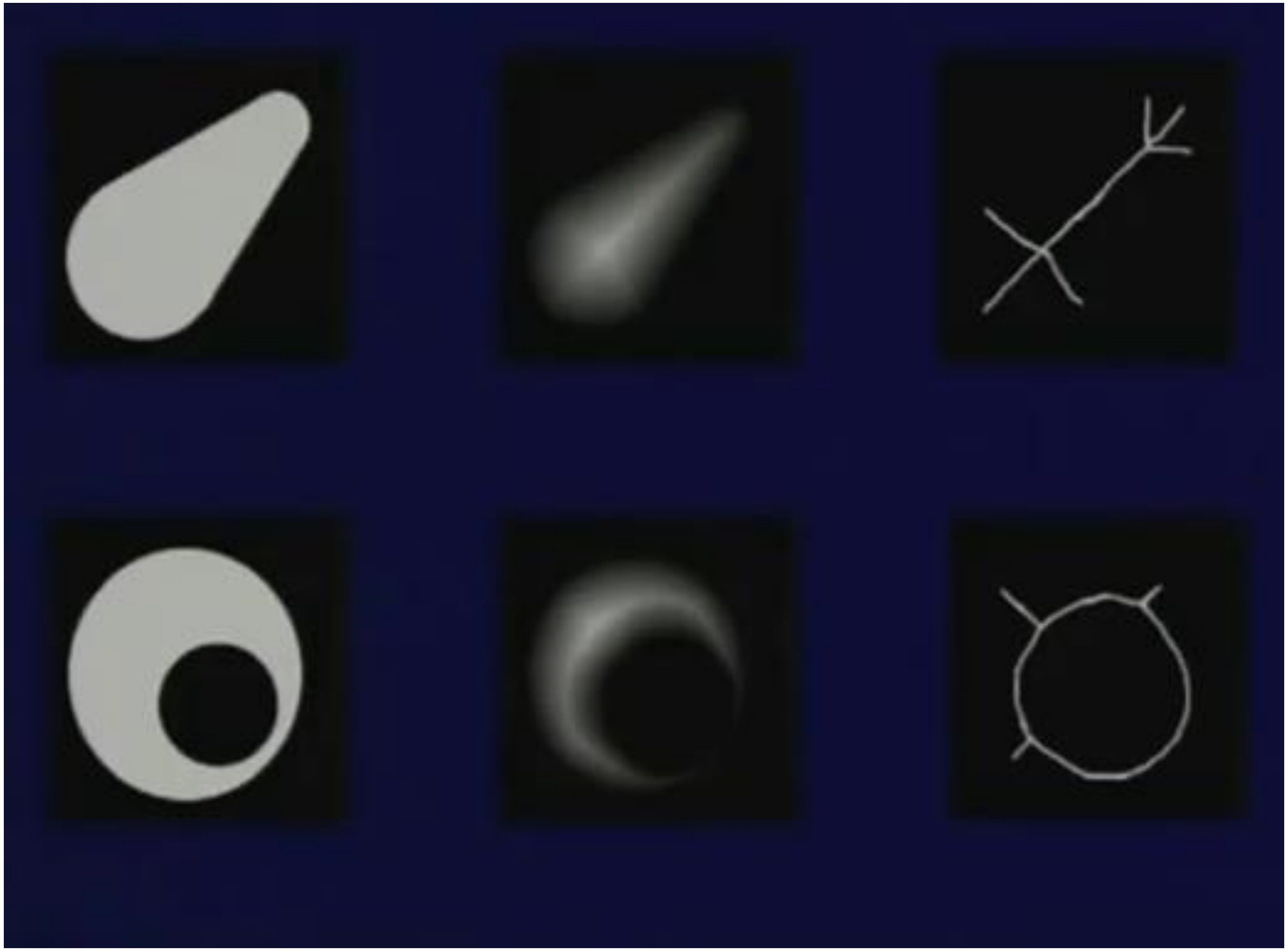
Original

Filtered

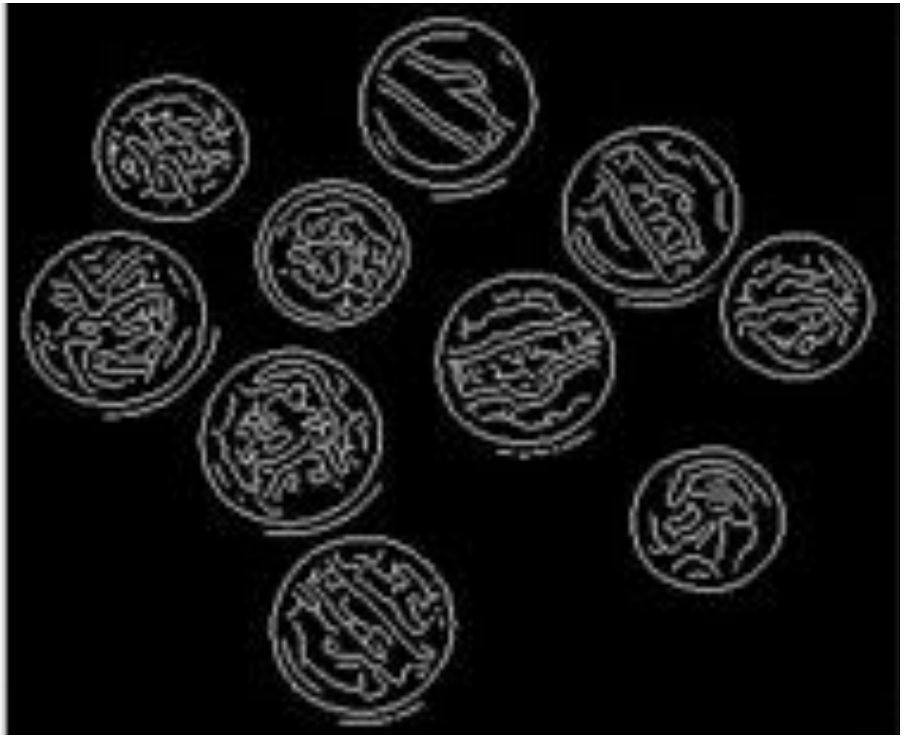
Thinning of Images



Thinning of Images



Edge Detection



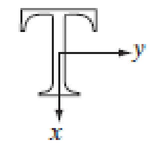
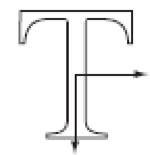
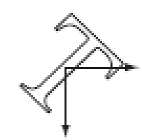
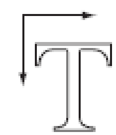
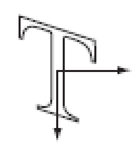
Geometric Transformation

$$(x, y) = T\{(v, w)\}$$

$$[x \;\; y \;\; 1] = [v \;\; w \;\; 1]\,\mathbf{T} = [v \;\; w \;\; 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

Geometric Transformation

TABLE 2.2
Affine transformations based on Eq. (2.6-23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Geometric Operations

Input image

Translated image with
Tx=15, Ty=30
using zero pad

Geometric Operations

Input image

Translated image with
$Tx=15$, $Ty=30$
using wrap around

Geometric Operations

Input image

Scaled image with Sx=1.1, Sy=0.9 using zero pad

Geometric Operations

- Rotation:
  - $Mx(x,y) = x*\cos(A) - y*\sin(A)$
  - $My(x,y) = x*\sin(A) + y*\cos(A)$

- Rotates image clockwise by A degrees
- Used to correct for camera tilt and/or orient object of interest in the image

Geometric Operations

Input image

Rotated image with
Angle=10 using zero pad

Geometric Operations

Input image

Rotated image with
Angle=10 using wrap around

Zooming

For example, the image F is replicated as follows:

$$\begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} = \begin{array}{|c|c|c|c|} \hline 2 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 3 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$


Enhanced Digital Zoom

Data Structures

1. Matrix
2. Chain code
3. Graphs
4. Relational databases
5. Hierarchical data structures