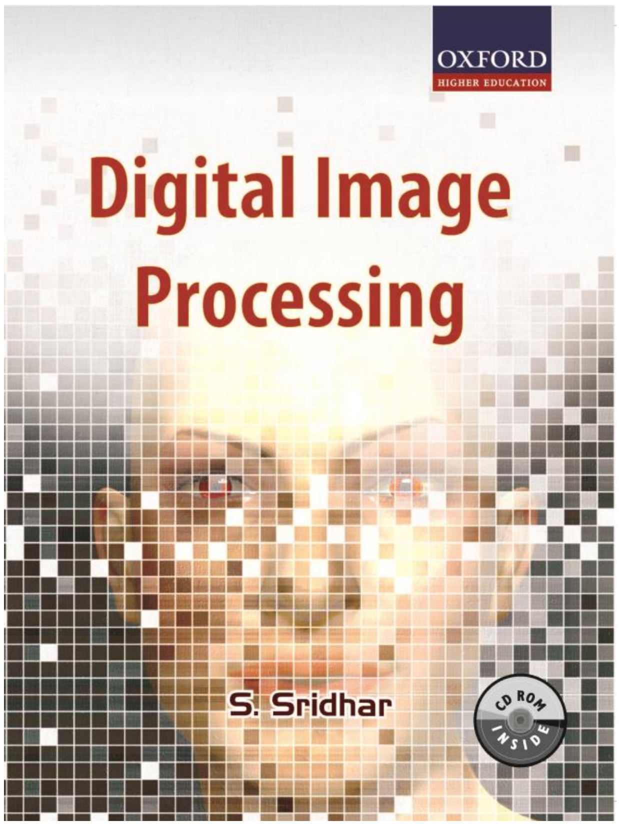


Digital Image Processing

S. Sridhar, Associate Professor,
Department of Information Science and Technology, College of Engineering Guindy Campus, Anna University, Chennai

Chapter – 3
Digital Image Processing Operations

BASIC RELATIONSHIPS AND DISTANCE METRICS

Image Coordinate System

$$
\begin{array}{r|lll}
2 & f(0,2) & f(1,2) & f(2,2) \\
1 & f(0,1) & f(1,1) & f(2,1) \\
y=0 & f(0,0) & f(1,0) & f(2,0) \\
\hline
 & x=0 & 1 & 2
\end{array}
$$

Fig. 3.1 Analog image *f(x, y)* in the first quadrant of Cartesian coordinate system

Image Coordinate system

$$
\begin{array}{c|ccc}
 & n=0 & n=1 & n=2 \\
\hline
m=0 & f(0,0) & f(0,1) & f(0,2) \\
m=1 & f(1,0) & f(1,1) & f(1,2) \\
m=2 & f(2,0) & f(2,1) & f(2,2)
\end{array}
$$

(a)

$$
\begin{array}{c|ccc}
 & n=1 & n=2 & n=3 \\
\hline
m=1 & f(1,1) & f(1,2) & f(1,3) \\
m=2 & f(2,1) & f(2,2) & f(2,3) \\
m=3 & f(3,1) & f(3,2) & f(3,3)
\end{array}
$$

(b)

Fig. 3.2 Discrete image (a) Image in the fourth quadrant of Cartesian coordinate system (b) Image coordinates as handled by software environments such as MATLAB

Image Topology

In $N_4(p)$, the reference pixel $p(x, y)$ at the coordinate position (x, y) has two horizontal and two vertical pixels as neighbours. This is shown graphically in Fig. 3.3.

$$\begin{pmatrix} 0 & X & 0 \\ X & p(x, y) & X \\ 0 & X & 0 \end{pmatrix}$$

Fig. 3.3 4-Neighbourhood $N_4(p)$

Diagonal Elements

$$\begin{pmatrix} X & 0 & X \\ 0 & p(x, y) & 0 \\ X & 0 & X \end{pmatrix}$$

Fig. 3.4 Diagonal elements $N_D(p)$

8-Neighbourhood

$$\begin{pmatrix} X & X & X \\ X & p(x, y) & X \\ X & X & X \end{pmatrix}$$

Fig. 3.5 8-Neighbourhood $N_8(p)$

Connectivity

4-Connectivity The pixels p and q are said to be in 4-connectivity when both have the same values as specified by the set V and if q is said to be in the set $N_4(p)$. This implies any path from p to q on which every other pixel is 4-connected to the next pixel.

8-Connectivity It is assumed that the pixels p and q share a common grey scale value. The pixels p and q are said to be in 8-connectivity if q is in the set $N_8(p)$.

Mixed connectivity Mixed connectivity is also known as m-connectivity. Two pixels p and q are said to be in m-connectivity when

1. q is in $N_4(p)$ or
2. q is in $N_D(p)$ and the intersection of $N_4(p)$ and $N_4(q)$ is empty.

8-connectivity Vs m-connectivity

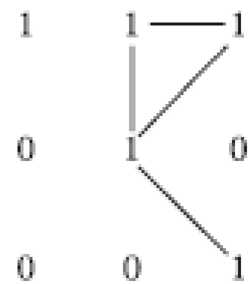


Fig. 3.6 8-Connectivity represented as lines

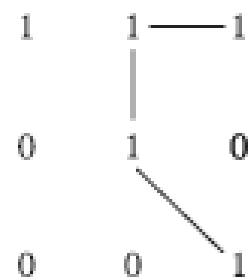


Fig. 3.7 *m*-Connectivity

Relations

Reflexive For any element a in the set A, if the relation aRa holds, this is known as a reflexive relation.

Symmetric If aRb implies that bRa also exists, this is known as a symmetric relation.

Transitive If the relations aRb and bRc exist, it implies that the relationship aRc also exists. This is called the transitivity property.

Distance Measures

0	1	1	1(*z*)
1	0	0	1
1	1	1	1(*q*)
1	1	1	1
(*p*)			

Fig. 3.9 Sample image

The distance function can be called metric if the following properties are satisfied:

1. $D(p, q)$ is well-defined and finite for all p and q.
2. $D(p, q) \geq 0$ if $p = q$, then $D(p, q) = 0$.
3. The distance $D(p, q) = D(q, p)$.
4. $D(p, q) + D(q, z) \geq D(p, z)$. This is called the property of triangular inequality.

Distance Measures

The Euclidean distance between the pixels p and q, with coordinates (x, y) and (s, t), respectively, can be defined as

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

The advantage of the Euclidean distance is its simplicity. However, since its calculation involves a square root operation, it is computationally very costly.

The D_4 distance or city block distance can be simply calculated as

$$D_4(p,q) = |x-s| + |y-t|$$

The D_8 distance or chessboard distance can be calculated as

$$D_8(p,q) = \max\left(|x-s|, |y-t|\right)$$

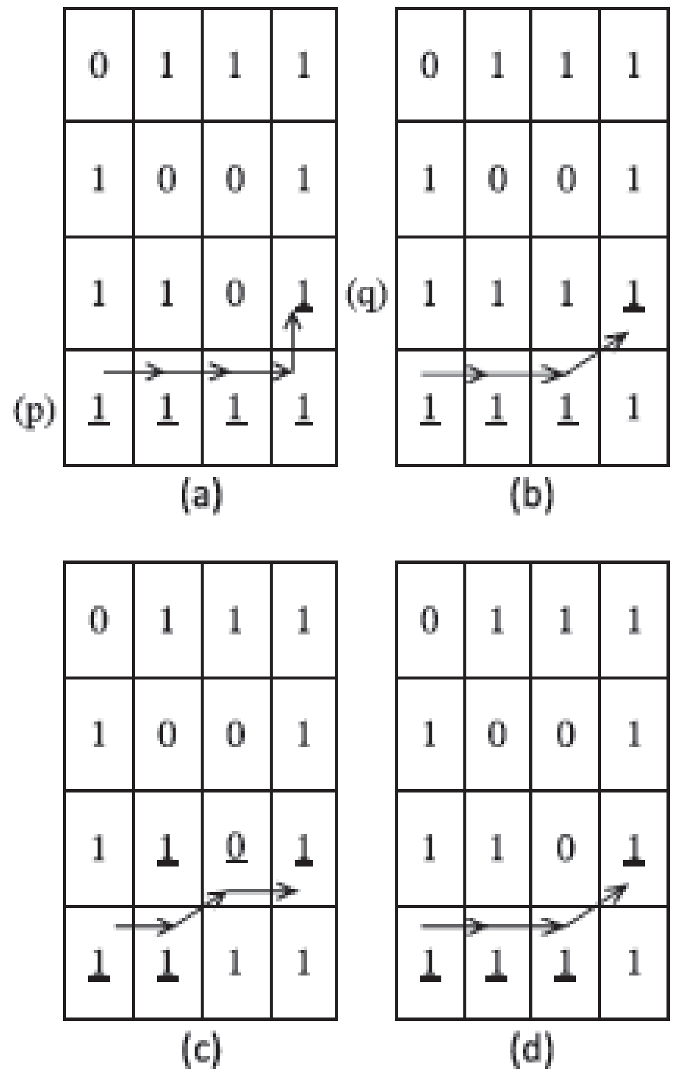


Fig. 3.11 Distance measures (a) Distance D_1 (b) Distance D_0 when $V = \{0, 1\}$ (c) Distance D_m when $V = \{0, 1\}$ (d) Distance D_m when $V = \{1\}$

Classification of Image Operations

- One way of classification is
  Point
  Local and
  Global

Classification

1. Linear operations 2. Non-linear operations

An operator is called a linear operator if it obeys the following rules of additivity and homogeneity. A non-linear operator, as the name suggests, does not follow these rules.

1. Property of additivity

$$
\begin{aligned}
H(a_1 f_1(x, y) + a_2 f_2(x, y)) &= H(a_1 f_1(x, y)) + H(a_2 f_2(x, y)) \\
&= a_1 H(f_1(x, y)) + a_2 H(f_2(x, y)) \\
&= a_1 \times g_1(x, y) + a_2 \times g_2(x, y)
\end{aligned}
$$

2. Property of homogeneity

$$H(k f_1(x, y)) = kH(f_1(x, y)) = kg_1(x, y)$$

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	Uint 8	0–255
2	Uint 16	0–65,535
3	Uint 32	0–4,29,49,67,295
4	Uint 64	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:

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

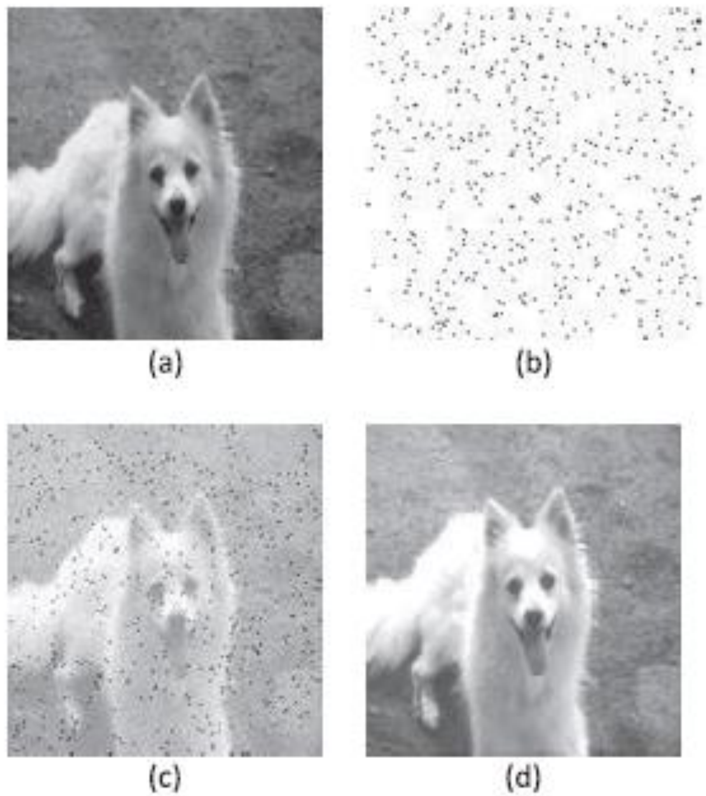


Fig. 3.14 Results of the image addition operation (a) Image 1 (b) Image 2 (c) Addition of images 1 and 2 (d) Addition of image 1 and constant 50

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

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

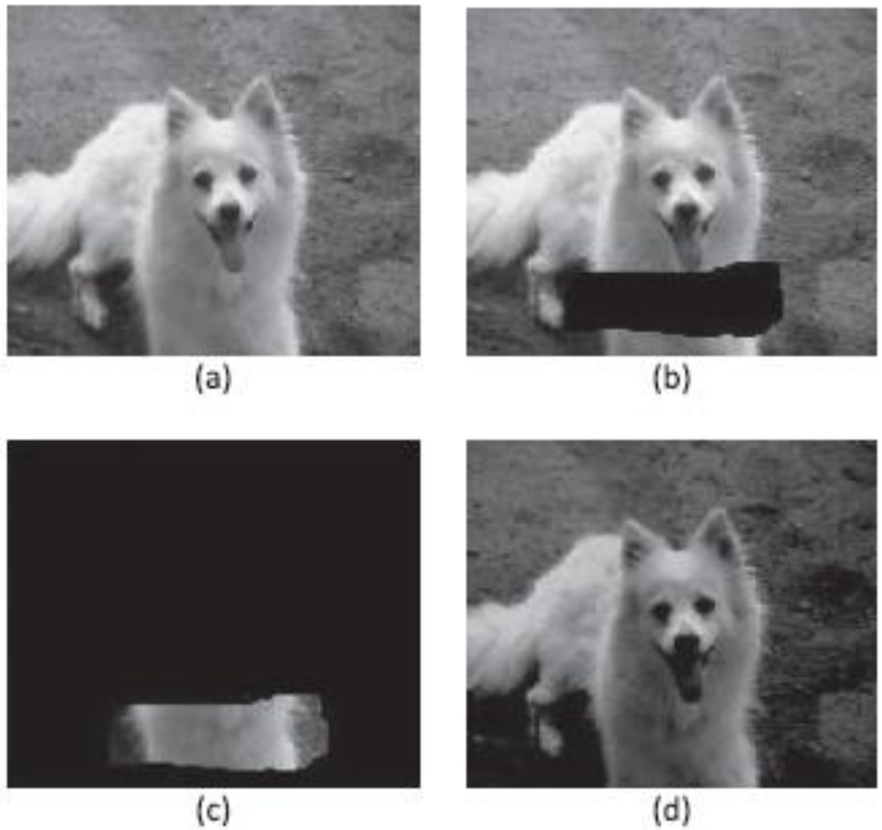


Fig. 3.15 Results of the image subtraction operation (a) Image 1 (b) Image 2 (c) Subtraction of images 1 and 2 (d) Subtraction of constant 50 from image 1

Image Multiplication

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

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

Fig. 3.16 Result of multiplication operation (image × 1.25) resulting in good 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.

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

Image Division

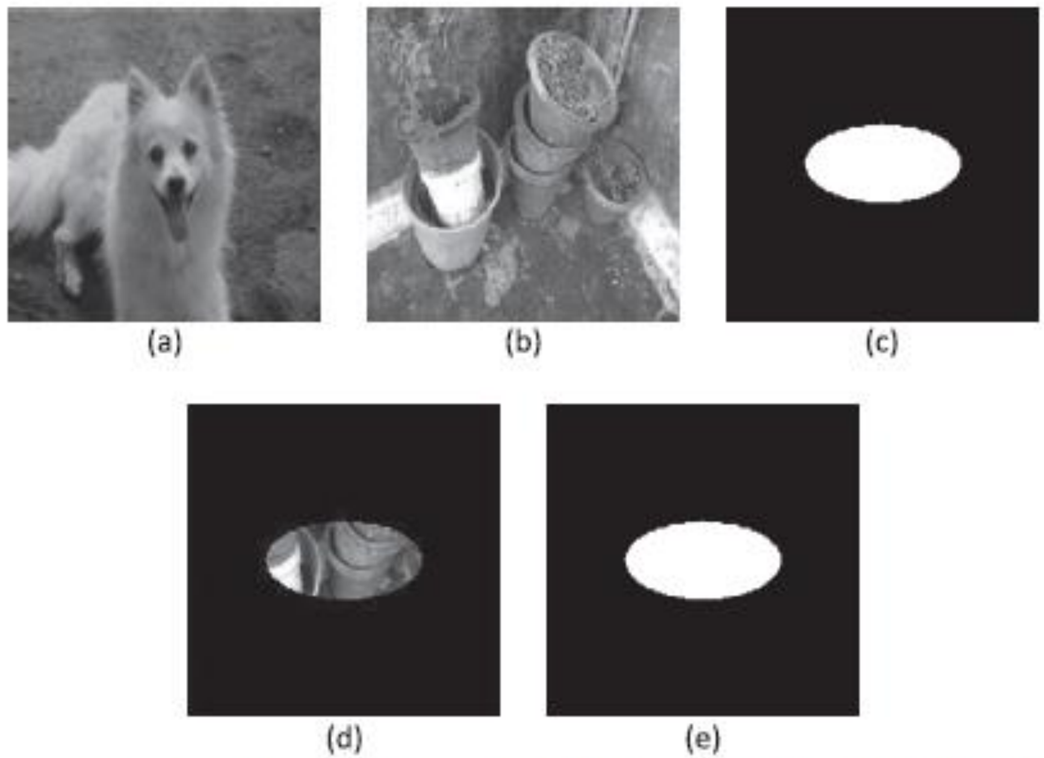


Fig. 3.17 Image division operation (a) Result of the image division operation (image/1.25) (b) Image 1 (c) Image 2 used as a mask (d) Image 3 = image 1 × image 2 (e) Image 4 = image 3/image 1

Logical Operations

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

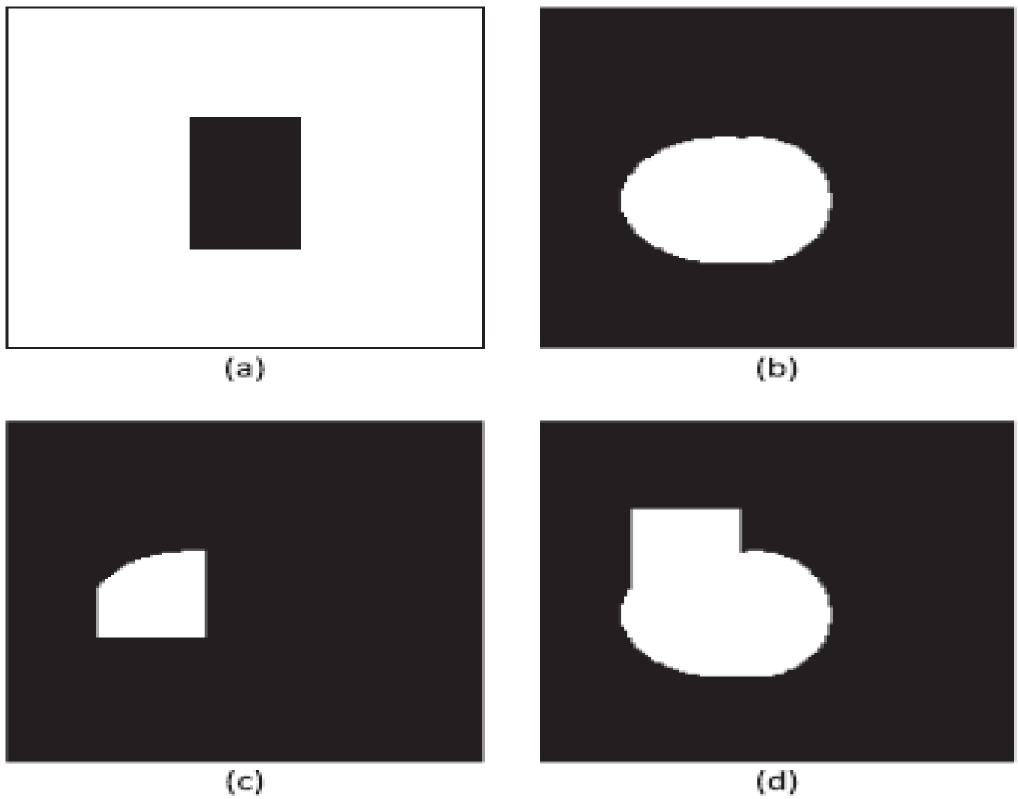


Fig. 3.18 Results of the AND and OR logical operators (a) Image 1 (b) Image 2 (c) Result of image 1 AND image 2 (d) Result of image 1 OR image 2

XOR

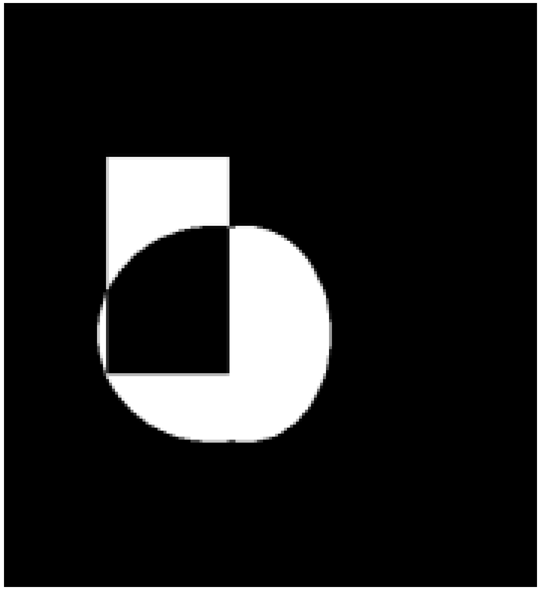


Fig. 3.19 Result of the XOR operation

NOT Operation

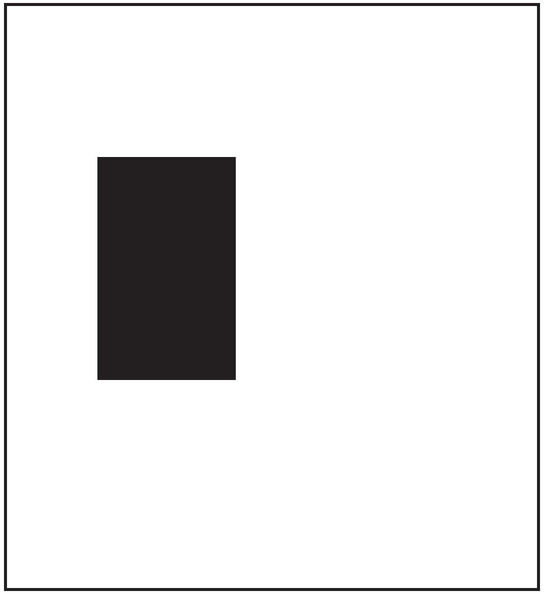


Fig. 3.20 Result of the NOT operation

Geometrical Operation

$$x' = x + \delta x$$

$$y' = y + \delta y$$

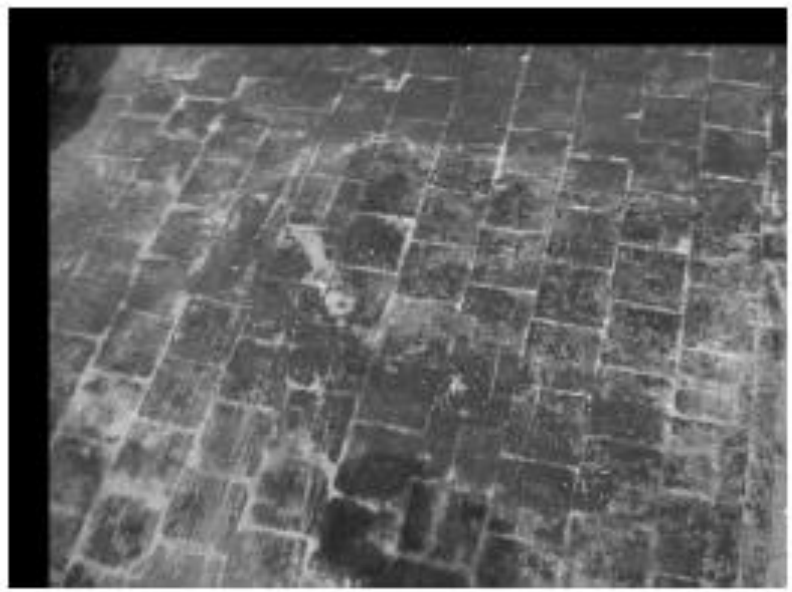


Fig. 3.21 Result of translation by 50 units

Scaling Operations

$$x' = x \times Sx$$
$$y' = y \times Sy$$

$$[x', y'] = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} [x, y]$$

$$[x', y', 1] = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} [x, y, 1]^T$$

The matrix $S = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is called scaling matrix.

Zooming

For example, the image F is replicated as follows:

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

Linear Interpolation

Consider the image

$$H = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$$

Linear interpolation is equivalent to fitting a straight line by taking the average along the rows and the columns. The process is described as follows:

1. For example, the matrix H can be zero-interlaced as

$H =$

2	0	1	0
0	0	0	0
1	0	3	0
0	0	0	0

2. Interpolate the rows. This is achieved by taking the average of the columns. This yields

2	1.5	1	0.5
0	0	0	0
1	2	3	1.5
0	0	0	0

3. Interpolate the columns. This is achieved by taking the average of the rows. This yields

2	1.5	1	0.5
1.5	1.75	2	1
1	2	3	1.5
0.5	1	1.5	0.75

Reflection

Reflection along X

$$F' = [-x, y] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \times [x, y]^{T}$$

Similarly, the reflection along the y-axis is given by

$$F' = [x, -y] = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \times [x, y]^{T}$$

Similarly, the reflection about the line $y = x$ is given as

$$F' = [x, -y] = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \times [x, y]^{T}$$

The reflection about $y = -x$ is given as

$$F' = [x, -y] = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \times [x, y]^{T}$$

Shearing

Shearing can be done using the following calculation and can be represented in the matrix form as

$$x' = sh_x \times y$$
$$y' = y$$
$$X_{\mathbf{shear}} = \begin{pmatrix} 1 & 0 & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Similarly, $Y_{\mathbf{shear}}$ can be given as

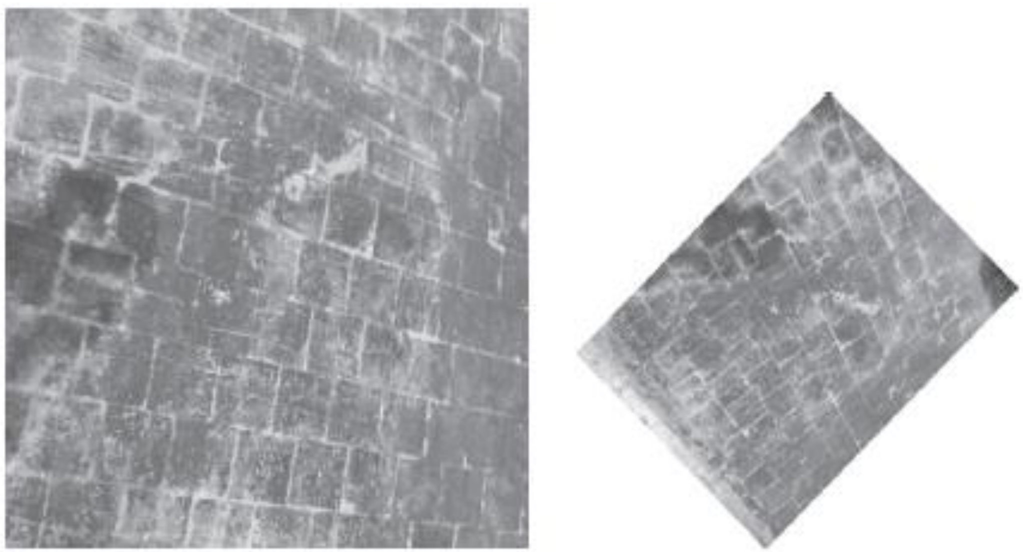
$$x' = x$$
$$y' = y \times sh_y$$
$$Y_{\mathbf{shear}} = \begin{pmatrix} 1 & sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where sh_x and sh_y are shear factors in the x and y directions, respectively.

Rotation

$$[x', y', 1] = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} [x, y, 1]^T$$

If θ is substituted with $-\theta$, this matrix rotates the image in the clockwise direction.



(a) (b)

Fig. 3.23 Rotation (a) Original image (b) Result of rotation by 45°

Affine Transform

$$x' = T_x(x, y)$$

$$y' = T_y(x, y)$$

T_x and T_y are expressed as polynomials. The linear equation gives an affine transform.

$$x' = a_0 x + a_1 y + a_2$$

$$y' = b_0 x + b_1 y + b_2$$

This is expressed in matrix form as

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Inverse Transform

$$\text{Inverse transform for translation} = \begin{pmatrix} 1 & 0 & -\delta x \\ 0 & 1 & -\delta y \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Inverse transform for scaling} = \begin{pmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Inverse transform for rotation can be obtained by changing the sign of the transform term. For example, the following matrix performs inverse transform.

$$\begin{pmatrix} \cos\theta & +\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Translation} = \begin{pmatrix} 0 & 0 & 0 & \delta x \\ 0 & 1 & 0 & \delta y \\ 0 & 0 & 1 & \delta z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Scaling} = \begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation along z-axis

$$\boldsymbol{R}_{z,\theta} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation along x-axis

$$\boldsymbol{R}_{x,\theta} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation along y-axis

$$\boldsymbol{R}_{y,\theta} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Compactly, these can be represented as

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = C \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Similarly, the reflection transform matrices can be given as

$$\text{Reflection}_{z\text{-axis}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Reflection}_{y\text{-axis}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Reflection}_{x\text{-axis}} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The shear matrices for the shear quantities a and b can be given as

$$\text{Shear}_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Shear}_x = \begin{pmatrix} 1 & a & b & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Shear}_y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ a & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Image Interpolation

- Downsampling

Example 3.5 Perform subsampling on the following image.

$$F = \begin{array}{|c|c|c|c|} \hline 3 & 3 & 3 & 3 \\ \hline 9 & 9 & 9 & 9 \\ \hline 3 & 3 & 3 & 3 \\ \hline 9 & 9 & 9 & 9 \\ \hline \end{array}$$

Solution

Subsampling can be done by choosing an upper-left pixel and replacing the neighbourhood with a chosen pixel value, i.e.,

$$F = \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$$

This method is called single pixel selection.

Alternatively, a statistical sample can be chosen. This can be the mean of the pixels; it replaces the neighbourhood. This technique yields

$$F = \begin{pmatrix} \frac{3+3+9+9}{4} & \frac{3+3+9+9}{4} \\ \frac{3+3+9+9}{4} & \frac{3+3+9+9}{4} \end{pmatrix} = \begin{pmatrix} 6 & 6 \\ 6 & 6 \end{pmatrix}$$

Upsampling

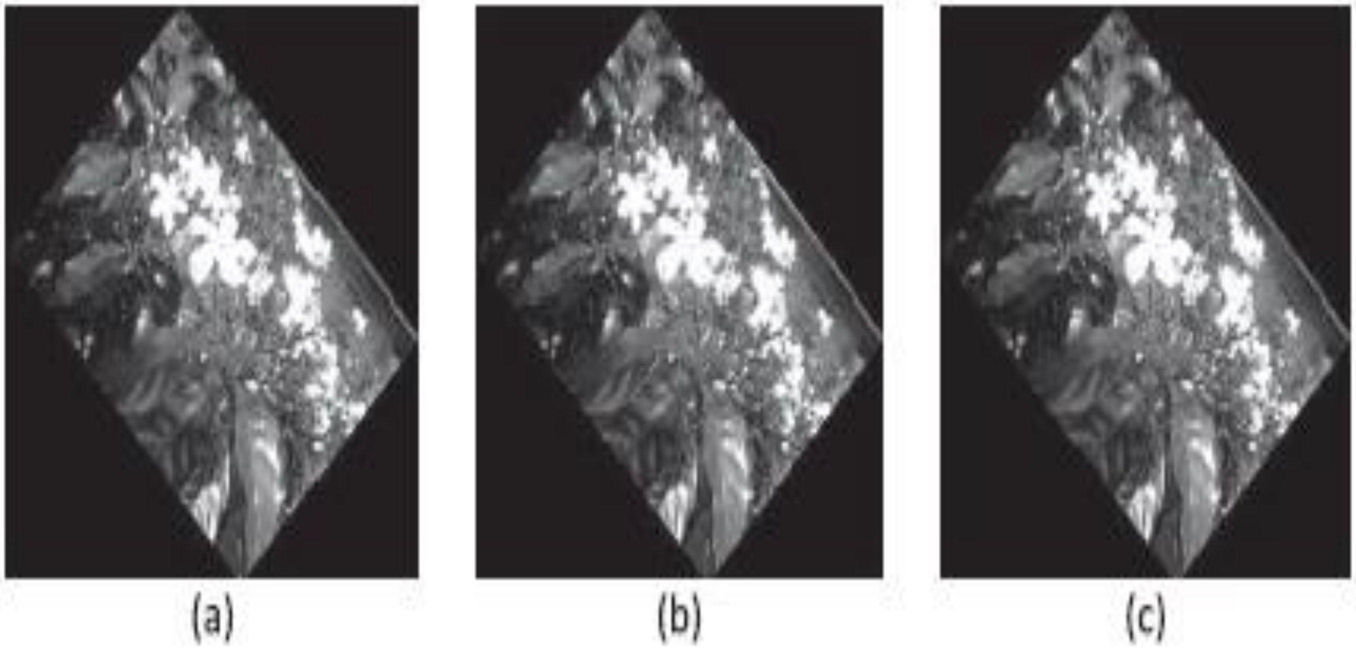


Fig. 3.24 Results of interpolation (a) Nearest neighbour (b) Bilinear (c) Bicubic

Set Operations

$$
\begin{array}{cc}
 & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\
\begin{array}{r} 0 \\ F = 1 \\ 2 \end{array} & \left|\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{array}\right|
\end{array}
$$

Fig. 3.25 Sample binary image

The complement of set A can be defined as the set of pixels that does not belong to the set A.

$$A^c = \{c/c \notin A\}$$

The reflection of the set is defined as

$$\hat{A} = \{c = -a, a \in A\}$$

The union of two sets, A and B, can be represented as

$$A \cup B = \{c/(c \in A) \vee (c \in B)\}$$

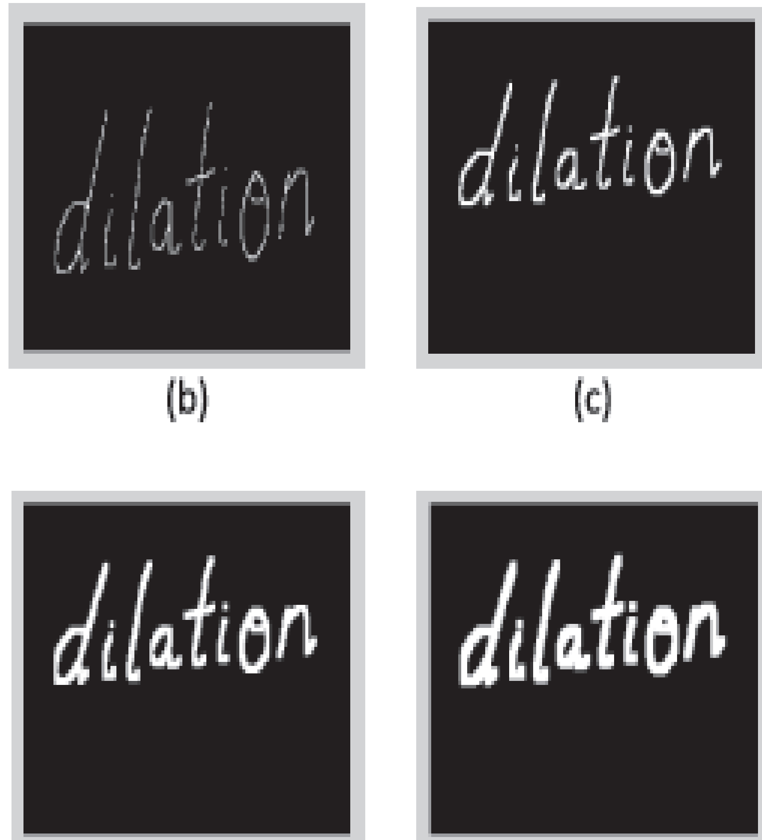
where the pixel c belongs to A, B, or both.

The intersection of two sets is given as $A \cap B = \{c/(c \in A) \wedge (c \in B)\}$. The pixel c belongs to A, B, or both.

The difference can be expressed as

$$A - B = \{c/(c \in A) \wedge (c \notin B)\}$$

which is equivalent to $A \cap B^c$.



(b) (c)

(d) (e)

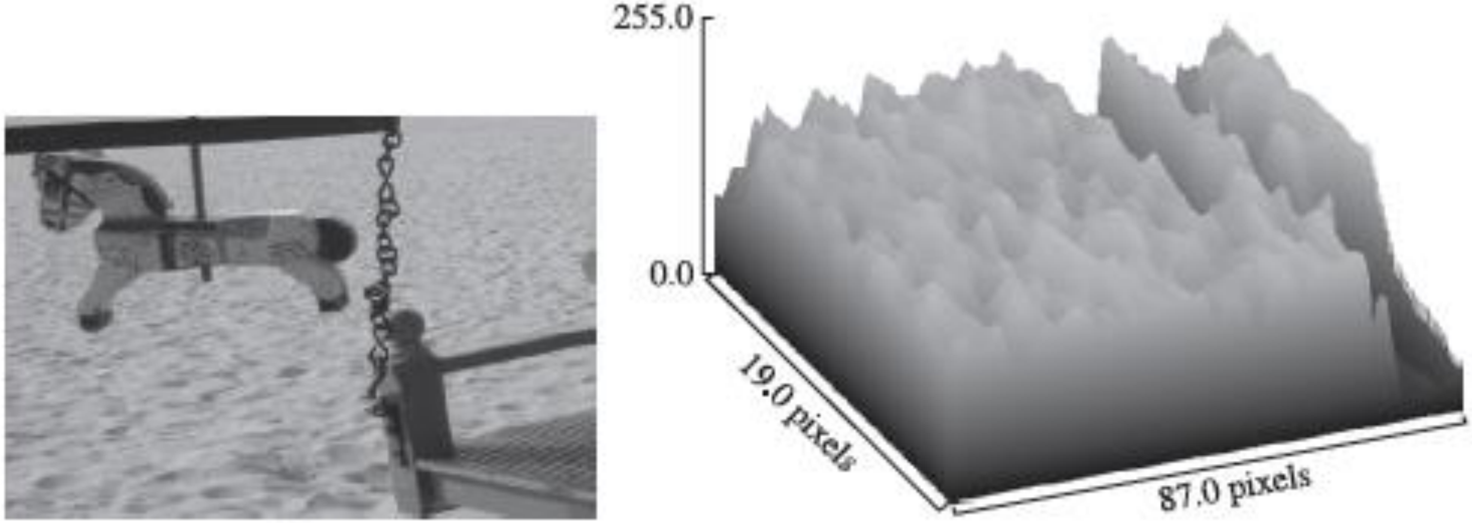
Fig. 3.26 Dilation operation (a) Effects of dilation and erosion for a numerical example (b) Original large image (c) Dilation operation with structural element (3, 3) (d) Dilation operation with structural element (9, 9) (e) Dilation operation with structural element (13, 13)

Statistical Operations

- Mean
- Mode
- Standard deviation
- Variance

Entropy

$$\text{Entropy } H = -\sum_{i=1}^{n} P_i \log_2 P_i$$



(a) (b)

Fig. 3.27 Surface plot (a) Original image (b) Surface plot

Image Convolution

The one-dimensional convolution formula is as follows:

$$g(x) = t * f(x)$$
$$= \sum_{i=-n}^{n} t(i) f(x - i)$$

1D-Convolution

Let $F = \{0, 0, 2, 0, 0\}$ and the kernel be $\{7\ 5\ 1\}$. As mentioned, the template has to be rotated by 180°. The rotated mask of this original mask [7 5 1] is a convolution template whose dimension is 1×3 with value $\{1, 5, 7\}$.

To carry out the convolution process, first, the process of zero padding should be carried out. Zero padding is process of creating more zeros and is done as shown in Table 3.7. Added zeros are underlined.

Table 3.7 Zero padding process for convolution

1	5	7				1	5	7
$\underline{0}$	$\underline{0}$	0	0	2	0	0	$\underline{0}$	$\underline{0}$

Table 3.8 Convolution process

(a) Initial position

Template

1	5	7						
0	0	0	0	2	0	0	0	0
	0							

Output is produced in the centre pixel.

(b) Position after one shift

Template is shifted by one bit.

	1	5	7					
0	0	0	0	2	0	0	0	0
	0	0						

Output produced is zero.

(c) Position after two shifts

Template is shifted again.

		1	5	7				
0	0	0	0	2	0	0	0	0
	0	0	14					

Output produced is 14.

(d) Position after three shifts

Template is shifted again.

			1	5	7			
0	0	0	0	2	0	0	0	0
	0	0	14	10				

Output produced is 10.

(e) Position after four shifts

Template is shifted again.

				1	5	7		
0	0	0	0	2	0	0	0	0
	0	0	14	10	2			

Output produced is 2.

(f) Position after five shifts

Template is shifted again.

					1	5	7	
0	0	0	0	2	0	0	0	0
	0	0	14	10	2	0		

Output produced is 0.

(g) Final position

Template is shifted again.

						1	5	7
0	0	0	0	2	0	0	0	0
	0	0	14	10	2	0	0	

Output produced is 0. Further shift crosses the range.
Hence the process is stopped.

So in the final position, the output produced is [0 0 14 10 2 0 0].

1D-Correlation

Table 3.9 Zero padding process for correlation

7	5	1				7	5	1
0	0	0	0	2	0	0	0	0

The padded zeroes are underlined. The correlation process is similar to the convolution process described. This process is shown in Table 3.10.

Table 3.10 Correlation process

(a) Initial position

Template

7	5	1						
0	0	0	0	2	0	0	0	0
	0							

Output produced is 0.

(b) Position after one shift

Template is shifted by one bit.

	7	5	1					
0	0	0	0	2	0	0	0	0
	0	0						

Output produced is 0.

(c) Position after two shifts

Template is shifted again.

		7	5	1				
0	0	0	0	2	0	0	0	0
	0	0	2					

Output produced is 2.

(d) Position after three shifts

Template is shifted again.

			7	5	1			
0	0	0	0	2	0	0	0	0
	0	0	2	10				

Output produced is 10.

(e) Position after four shifts

Template is shifted again.

				7	5	1		
0	0	0	0	2	0	0	0	0
	0	0	2	10	14			

Output produced is 14.

(f) Position after five shifts

Template is shifted again.

					7	5	1	
0	0	0	0	2	0	0	0	0
	0	0	2	10	14	0		

Output produced is 0.

(g) Final position

Template is shifted again.

						7	5	1
0	0	0	0	2	0	0	0	0
	0	0	2	10	14	0	0	

Output produced is 0.

So in the final position, the output produced is [0 0 2 10 14 0 0].

2D-Convolution

If $T(i, j)$ is a template of dimension $n \times m$ and the image $f(i, j)$ is the input image, the convolution of f with T is written as

$$F * T = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} t(i, j) \times f(x-i, y-j)$$

The convolution process on a 2D image can be shown as follows. Consider an image

$F = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$ and the template or mask is $t = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

$$\begin{array}{cccc} 0_1 & 0_1 & 0 & 0 \\ 0_1 & 2_1 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(a)

$$\begin{array}{cccc} 0 & 0_1 & 0_1 & 0 \\ 0 & 2_1 & 2_1 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(b)

$$\begin{array}{cccc} 0 & 0 & 0_1 & 0_1 \\ 0 & 2 & 2_1 & 0_1 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(c)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0_1 & 2_1 & 2 & 0 \\ 0_1 & 2_1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(d)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 2_1 & 2_1 & 0 \\ 0 & 2_1 & 2_1 & 0 \\ 0 & 0 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 0 & 0 \\ 4 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(e)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 2 & 2_1 & 0_1 \\ 0 & 2 & 2_1 & 0_1 \\ 0 & 0 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 0 & 0 \\ 4 & 8 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(f)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0_1 & 2_1 & 2 & 0 \\ 0_1 & 0_1 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 0 & 0 \\ 4 & 8 & 4 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(g)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2_1 & 2_1 & 0 \\ 0 & 0_1 & 0_1 & 0 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 0 & 0 \\ 4 & 8 & 4 & 0 \\ 2 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(h)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2_1 & 0_1 \\ 0 & 0 & 0_1 & 0_1 \end{array} \Rightarrow \begin{array}{cccc} 2 & 4 & 0 & 0 \\ 4 & 8 & 4 & 0 \\ 2 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(i)

Fig. 3.28 Convolution process (a) Initial position (b) Position after first shift (c) Position after second shift (d) Position after first vertical shift (e) Position after third shift (f) Position after fourth shift (g) Position after second vertical shift (h) Position after fifth shift (i) Position after sixth shift

Properties of Convolution

1. Convolution is separable.
2. Convolution is commutative.

$$f * T = T * f$$

3. Convolution is associative.

$$(f * T) * h = f * (T * h)$$

4. Convolution follows the principle of superposition.

$$(f + g) * T = f * T + g * T$$

Data Structures

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

Chain Code

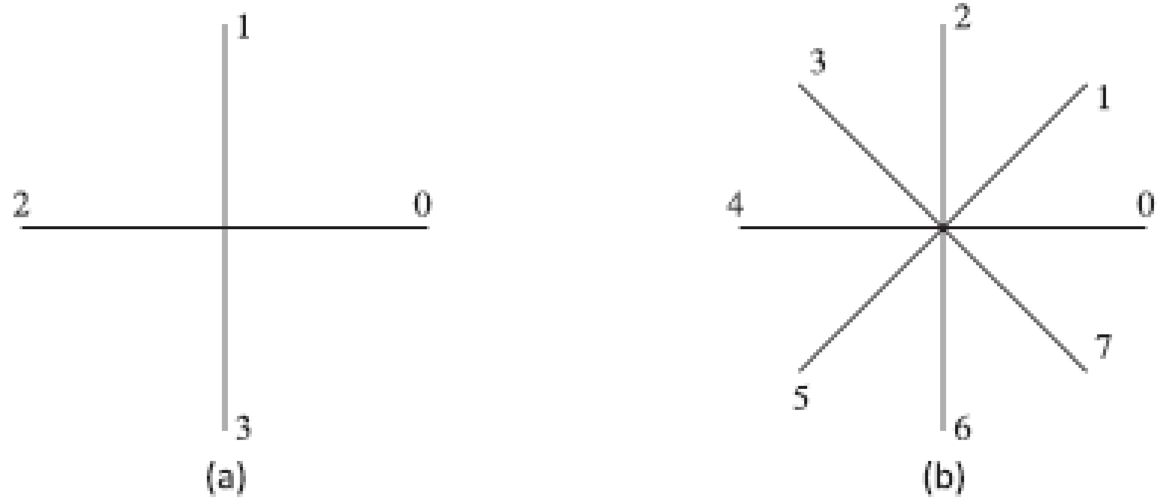


Fig. 3.29 Image chain codes (a) 4-Directional code (b) 8-Directional code

RAG

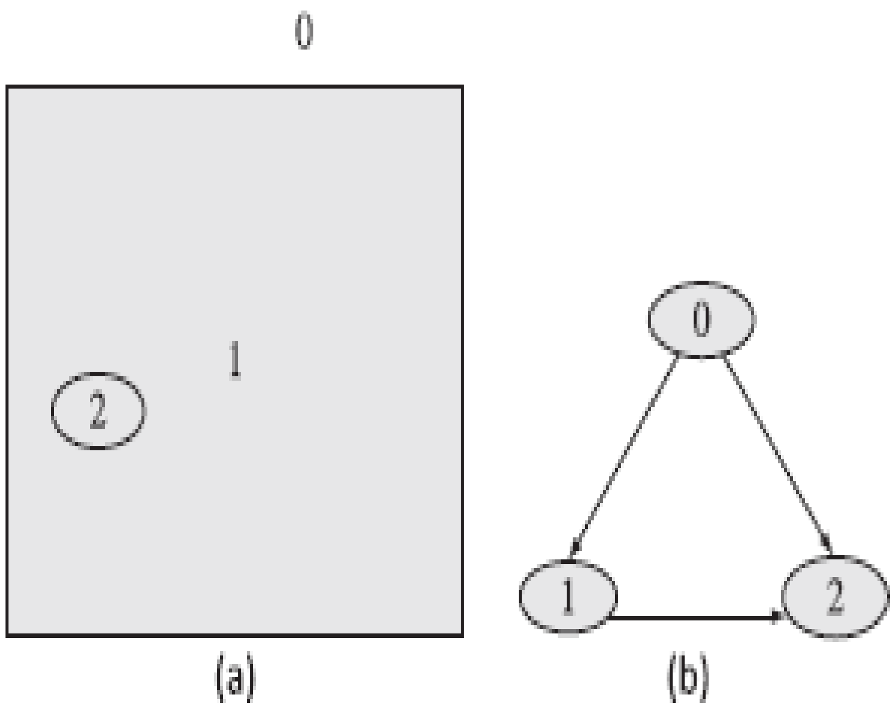


Fig. 3.30 Region adjacency graph (a) Sample image (b) RAG for given sample image

Relational Structures

Table 3.11 Relational structure

Object number	Object name	Object attributes such as row, column, and colour
1	1	....
2	2	....

Hierarchical Structures

Fig. 3.31 Structure of *M*-pyramid (a) Original image (b) Pyramid level 1 (c) Pyramid level 2

Pyramid Structures

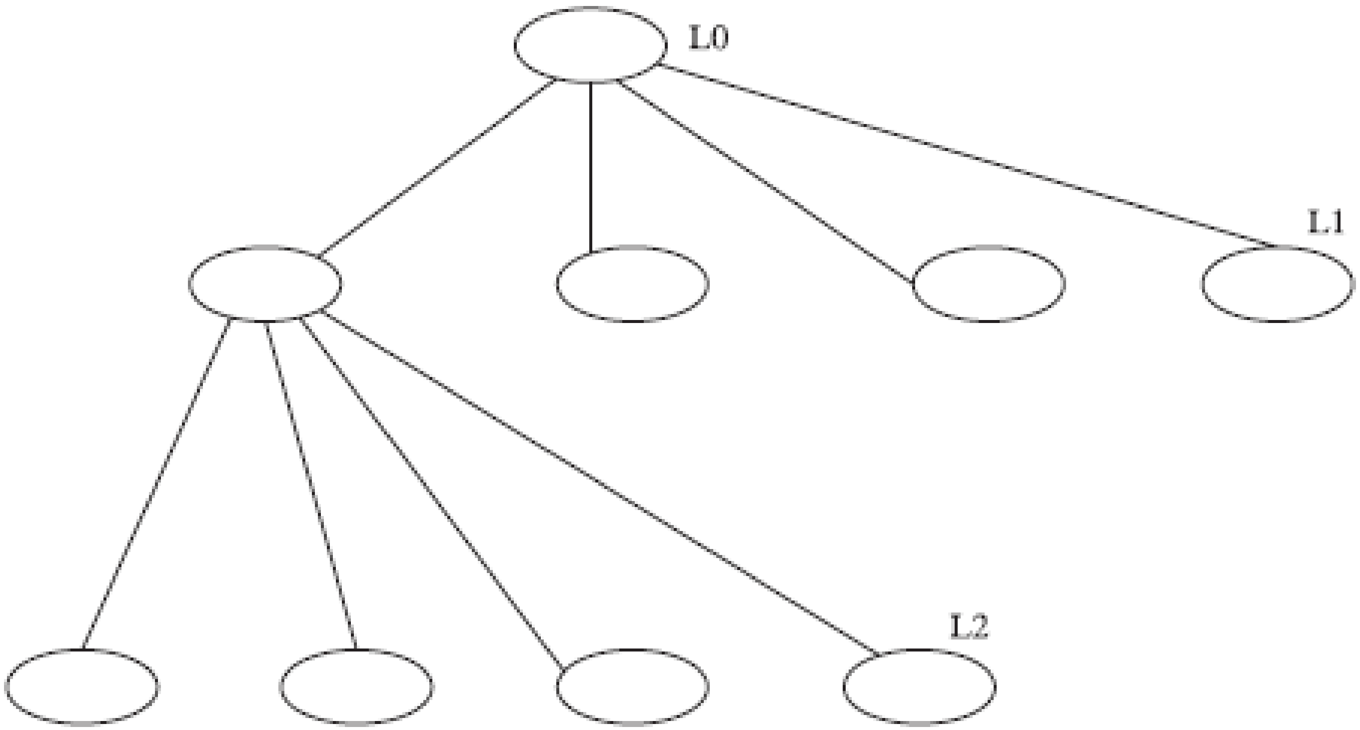


Fig. 3.32 *T*-pyramid data structure

Quadtree

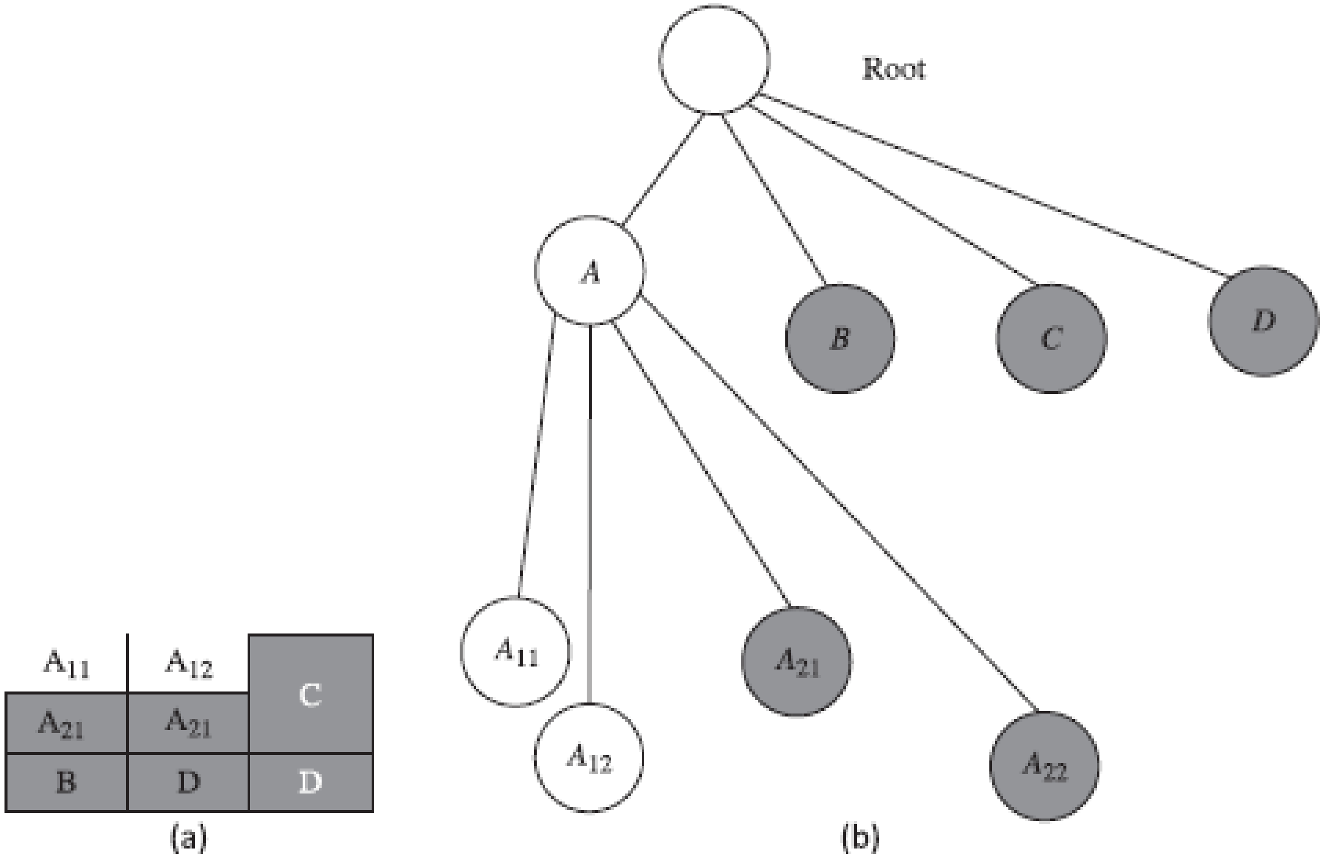


Fig. 3.33 Quadtrees (a) Sample image (b) Quadtree

Application Development

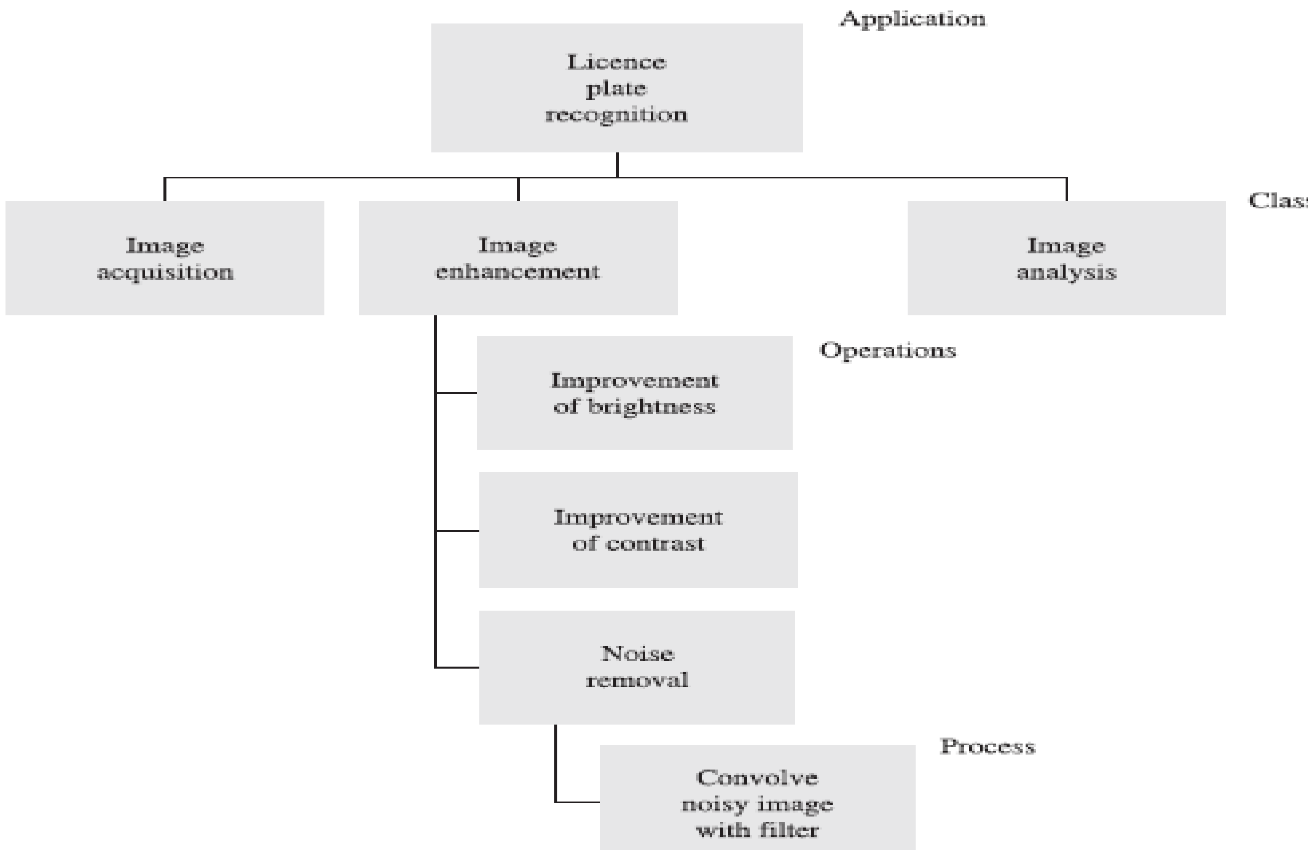


Fig. 3.34 Organization of an image processing application—number plate recognition system

SUMMARY

1. The digital image is characterized by the image coordinate system and the neighbourhood concept.
2. The distance between the pixels p and q in an image can be given by distance measures such as Euclidian distance, D_4 distance, and D_8 distance.
3. Point operations are operations whose output value at a specific coordinate is dependent only on the input value.
4. A local operation is an operation whose output value at a specific coordinate is dependent on the input values in the neighbourhood of that pixel.
5. Global operations are operations whose output value at a specific coordinate is dependent on all the values in the input image.
6. Some of the widely used image operations are arithmetic, logical, geometrical, statistical, and spatial operations.
7. Whenever a geometric transformation is performed, a resampling process should be carried out so that the desirable quality is maintained in the resultant image.
8. Upsampling is a process of increasing the spatial resolution of the image. Downsampling decreases the spatial resolution.
9. Image processing, image analysis, and computer vision require good form of data representation or organization. Some data structures that are traditionally used to manipulate digital images are matrices, chain codes, graphs, and relational databases.
10. Matrix is one of the most popular data structures for storing and manipulating the images in the initial level of pixels.
11. Chain codes or Freeman codes are used to represent the boundary of an image.
12. The region adjacency graph (RAG) is a data structure that is used to represent regions and their adjacency.
13. Relational databases can also be used for representation of information of an image. The table can record the different objects that are present in the image. The objects can then be searched using the keys.
14. Pyramids are among the simplest hierarchical data structures. Pyramids are helpful in working at the different resolutions of the image.
15. Image processing applications should be developed in a methodical, structured, and disciplined manner. The hierarchy of the image processing applications can be defined as applications $\rightarrow$ classes $\rightarrow$ operations $\rightarrow$ process.