Digital Image Fundamentals II

1. Image modeling and representations
2. Pixels and Pixel relations
3. Arithmetic operations of images
4. Image geometry operation
5. Image processing with Matlab
   - Image Processing Toolbox (IPT)

1. Image modeling and representation

- A 2D image is a function of any quantity over a finite spatial extent:

$$f(x, y)$$

$$x_1 \le x \le x_2$$

$$y_1 \le y \le y_2$$

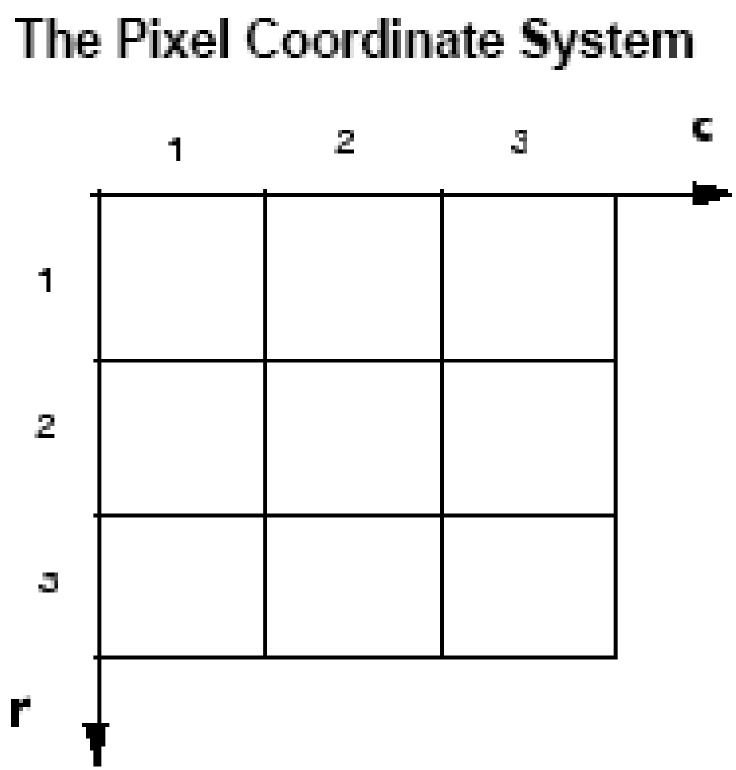
- *The value of f(x, y,) can be real number, integer, or 0,1*
  - *Continuous space image: x and y are chosen as real numbers*
  - *Discrete space: x and y are chosen as integers*

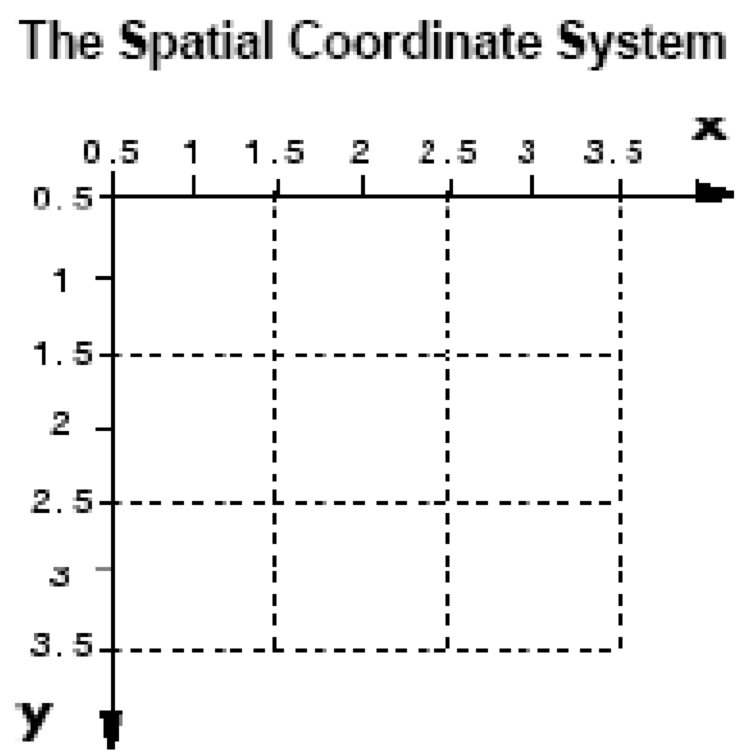
Digital images in discrete space

- *2D* digital image
  – x = 0, …, M-1, y = 0, …, N-1, and f(x, y) are integers in binary format of 8 bits, 16 bits (or d bits).
  – (i, j) is pixel, f(x, y) is its pixel or intensity
  – Resolution: M × N, i.e. number of pixels
  – Size: M × N × d
    i.e., the total number of bits to represent the image
  – Represented as a matrix

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Pixel coordinate and spatial coordinate



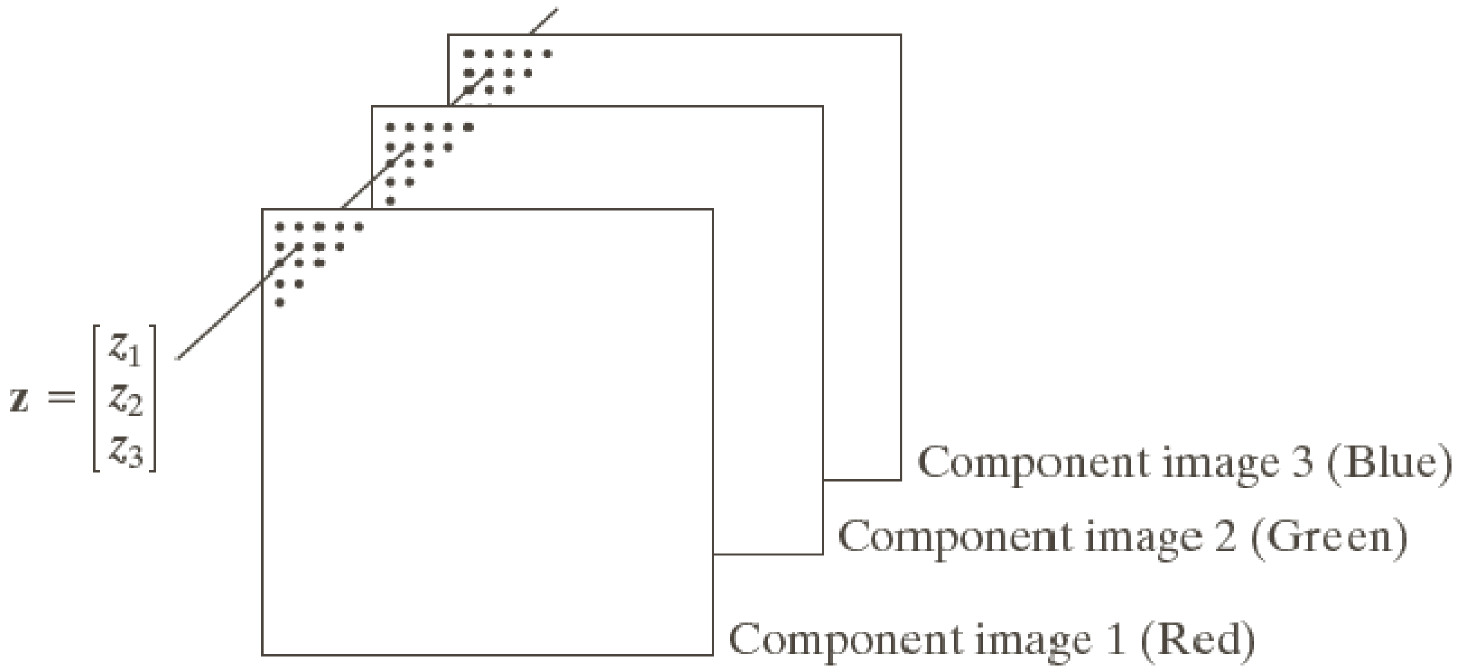


- Discrete domain

Continuous demain

For color image, f(x, y) is triple of integers:

$$f(x, y) = (z_1(x, y), z_2(x, y), z_3(x, y))$$



One-Dimensional Image

- 1D image is a function

$$f(x), x_1 \le x \le x_2$$

- One-dimensional (1D) image can be represented as a sequence of numbers:

$$x(n), n = 0, 1, \ldots, M - 1$$

- Graphical representation

Image representation

- A 1D image of one point

$$\delta(x) = \begin{cases} 1, x = 0 \\ 0, x \neq 0 \end{cases}$$

- Represent image of discrete domain by continuous image function.

$$x(n), n = 0, 1, \ldots, N-1$$

- Representation in continuous domain

$$X(x) = \sum_{n=0}^{N-1} x(n)\delta(x-n), -\infty < x < \infty$$

Image representation

- 2D image of one point

$$\delta(x, y) = \begin{cases} 1, (x, y) = (0, 0) \\ 0, (x, y) \neq (0, 0) \end{cases}$$

- Represent image of discrete domain by continuous image function.

$$f(x, y), x = 0, 1, \ldots, M-1, y = 0, 1, \ldots, N-1$$

$$F(x, y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(x, y)\delta(x - i, y - j),$$

$$-\infty < x < \infty, -\infty < y < \infty$$

2. Relations of pixels

- Neighbors of a pixel p at (x, y)

4-neighbors of p:

$N_4(p)$ ={ (x+1, y), (x-1,y),(x,y+1), (x,y-1)}

4-diagonal neighbors of p:

$N_D(p)$ = { (x+1, y+1), (x+1,y-1),(x-1,y+1), (x-1,y-1)}

8-neighbors of p :

$N_8(p) = N_4(p) \cup N_D(p)$

Connectivity

Let V be the set of gray-level values used to defined connectivity

- 4-connectivity : 2 pixels p and q with values from V are 4-connected if q is in the set $N_4(p)$
- 8-connectivity : 2 pixels p and q with values from V are 8-connected if q is in the set $N_8(p)$
- m-connectivity (mixed connectivity): 2 pixels p and q with values from V are m-connected if q is in the set $N_4(p)$, or q is in the set $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ is empty. (the set of pixels that are 4-neighbors of both p and q whose values are from V)

Adjacent, path and connected component

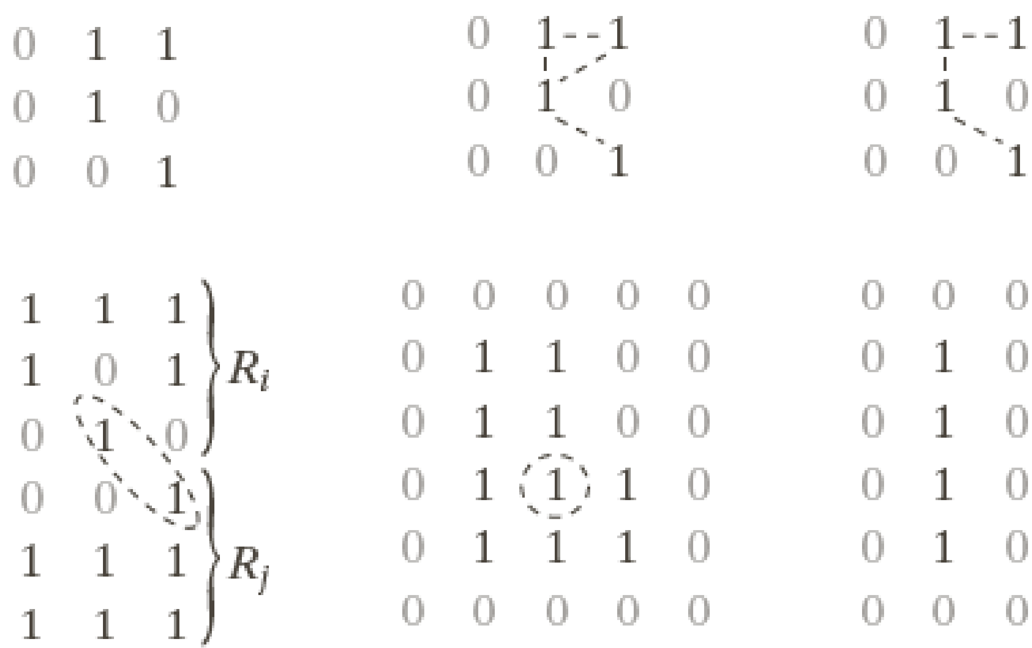
- Adjacent: a pixel p is adjacent to a pixel q if they are connected

- Two image area subsets S1 and S2 are adjacent if some pixel in S1 is adjacent to some pixel S2

- Path
  - a path from pixel p(x,y) to pixel q(s,t) is a sequence of distinct pixels with coordinates
    $(x,y)=(x_0,y_0), (x_1,y_1),\ldots, (x_n, y_n)$,
    where $(x_0, y_0) = (x, y)$, $(x_n, y_n) = (s, t)$ and (x_i, y_i) is adjacent to (x_{i-1},y_{i-1}), n is the length of the path

- Connected component: a group of pixels such that it contains a path connecting every pair of its pixels.

Example



a b c
d e f

FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) m-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

Example

Consider the two image subsets S1 and S2 :

For *V={1}, determine whether S1 and S2 are*

4-connected, 8-connected, m-connected

0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1	1	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	1	1	1

Labeling of Connected Components

scan the image from left to right

Let p denote the pixel at any step in the scanning process.

Let t denote the upper neighbor of p.

Let l denote the left-hand neighbors of p, respectively. when we get to p, points t and l have already been encountered and labeled if they were 1's.

if the value of p = 0, move on.

if the value of p = 1, examine t and l.

if they are both 0, assign a new label to p.

if only one of them is 1, assign its label to p.

if they are both 1

if they have the same label, assign that label to p.

if not, assign one of the labels to p and make a note that the two labels are equivalent. (t and l are connected through p).

at the end of the scan, all points with value 1 have been labeled.

do a second scan, assign a new label for each equivalent labels

Example

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & a \\ 0 & 0 & b & a \\ c & 0 & 0 & 0 \\ c & c & 0 & 0 \end{bmatrix}, (a \Leftrightarrow b) \rightarrow \begin{bmatrix} 0 & 0 & 0 & d \\ 0 & 0 & d & d \\ c & 0 & 0 & 0 \\ c & c & 0 & 0 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 0 & 0 & 0 & d \\ 0 & 0 & d & d \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ c & 0 & 0 & 0 \\ c & c & 0 & 0 \end{bmatrix}$$

- How to find the bounding box of a connected component?

Distance Measures

Given pixels p(x,y), q(s,t) and z(u,v). D is a distance function or metric if

(a) $D(p,q) \geq 0$; $D(p,q) = 0$ iff $p=q$

(b) $D(p,q) = D(q,p)$

(c) $D(p,z) \leq D(p,q) + D(q,z)$

Euclidean Distance

Euclidean distance between p and q

$$p(x, y), q(s, t),$$

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

Manhattan Distance

- Manhattan distance

$$p(x, y), q(s, t),$$

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

e.g. the diamond centered at (x, y)

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

Chessboard distance

- D_8 distance

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

Other distances

- D_4 distance
  – defined by the length of the shortest D_4 path
- D_8 distance
  – defined by the length of the shortest D_8 path
- m-connectivity's distance
  – defined by the length of the shortest m-connectivity path

Arithmetic Operators of Images

- Arithmetic operations of two images are carried out by the arithmetic operations of the corresponding pixels of the two images.
- Addition : p+q used in image average to reduce noise.
- Subtraction : p-q basic tool in medical imaging.
- Multiplication : pxq to correct gray-level shading result from non-uniformities in illumination or in the sensor used to acquire the image.
- Division : p/q

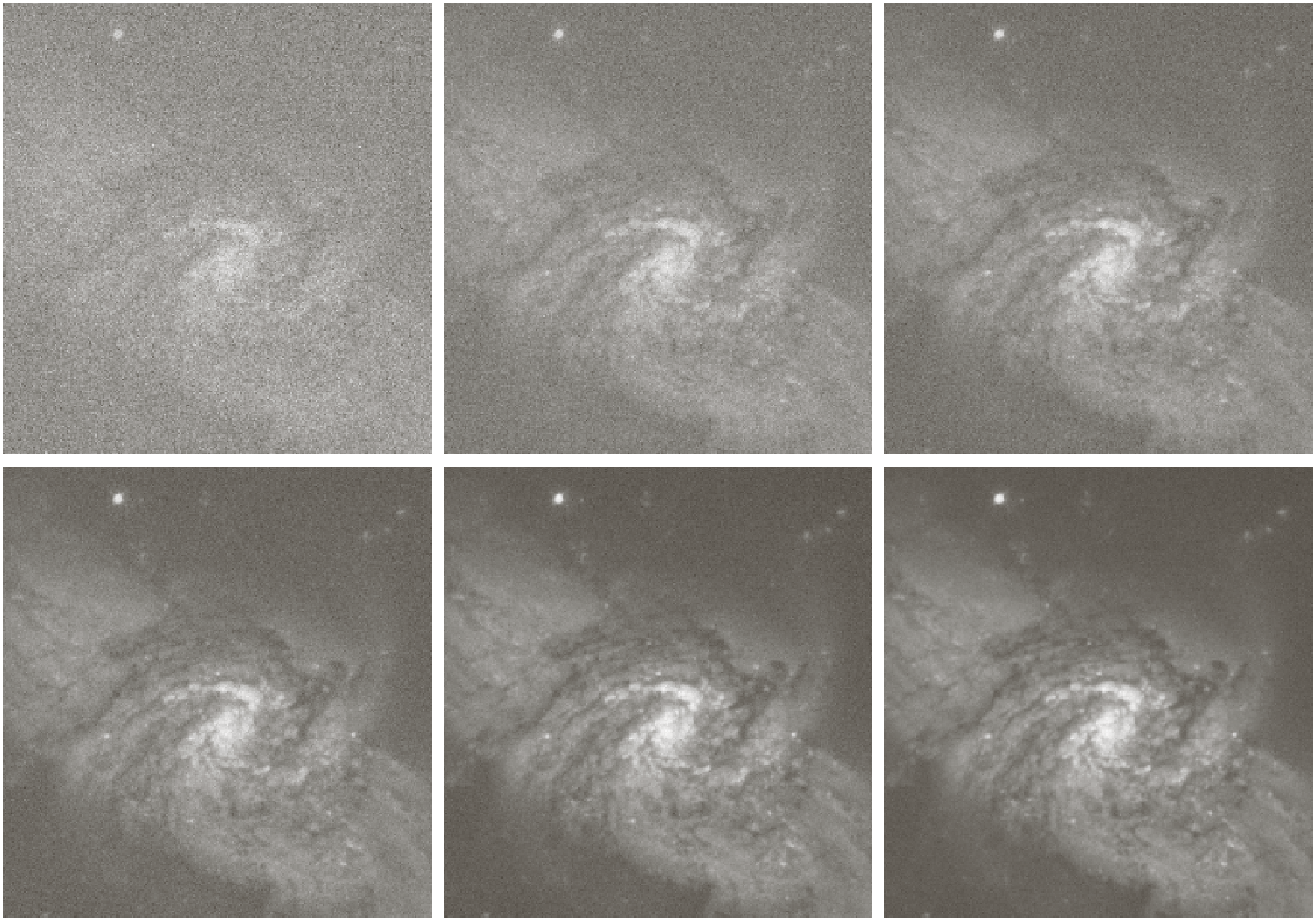
Logic operations

- AND : p AND q
- OR :   p OR q
- COMPLEMENT : NOT q ( )

- logic operations apply only to binary images

- Arithmetic operations apply to multi-valued pixels

- logic operations used for tasks such as masking, feature detection, and shape analysis.



a b c
d e f

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.)

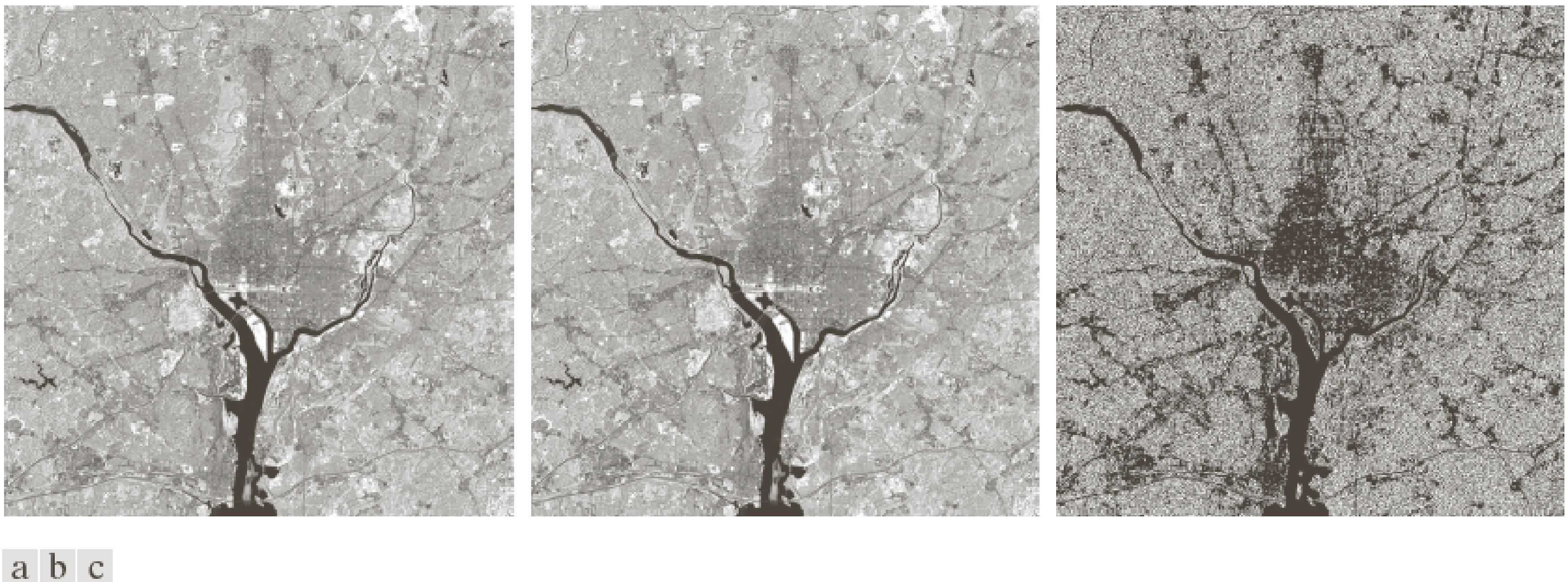


FIGURE 2.27 (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range [0, 255] for clarity.

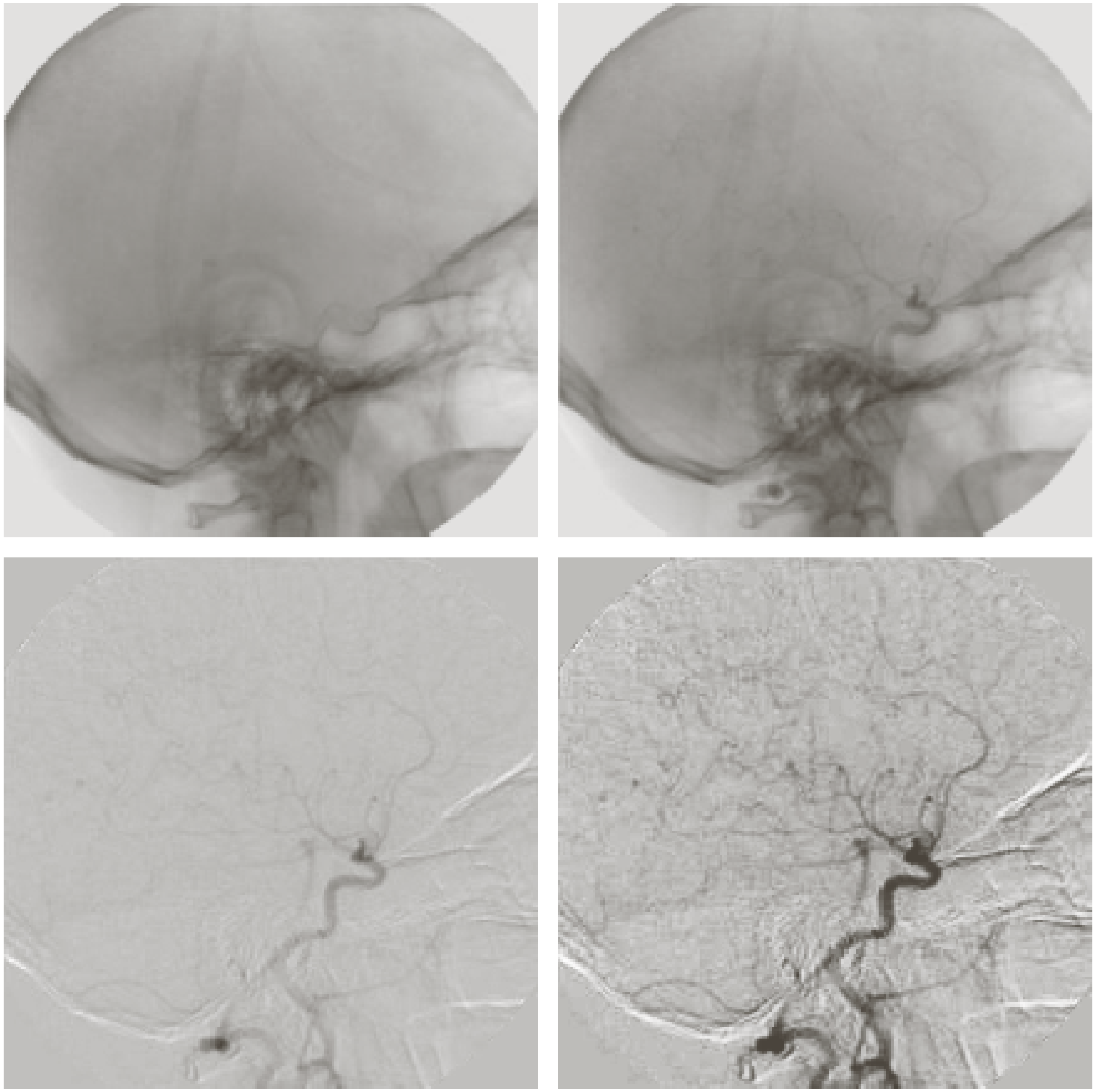
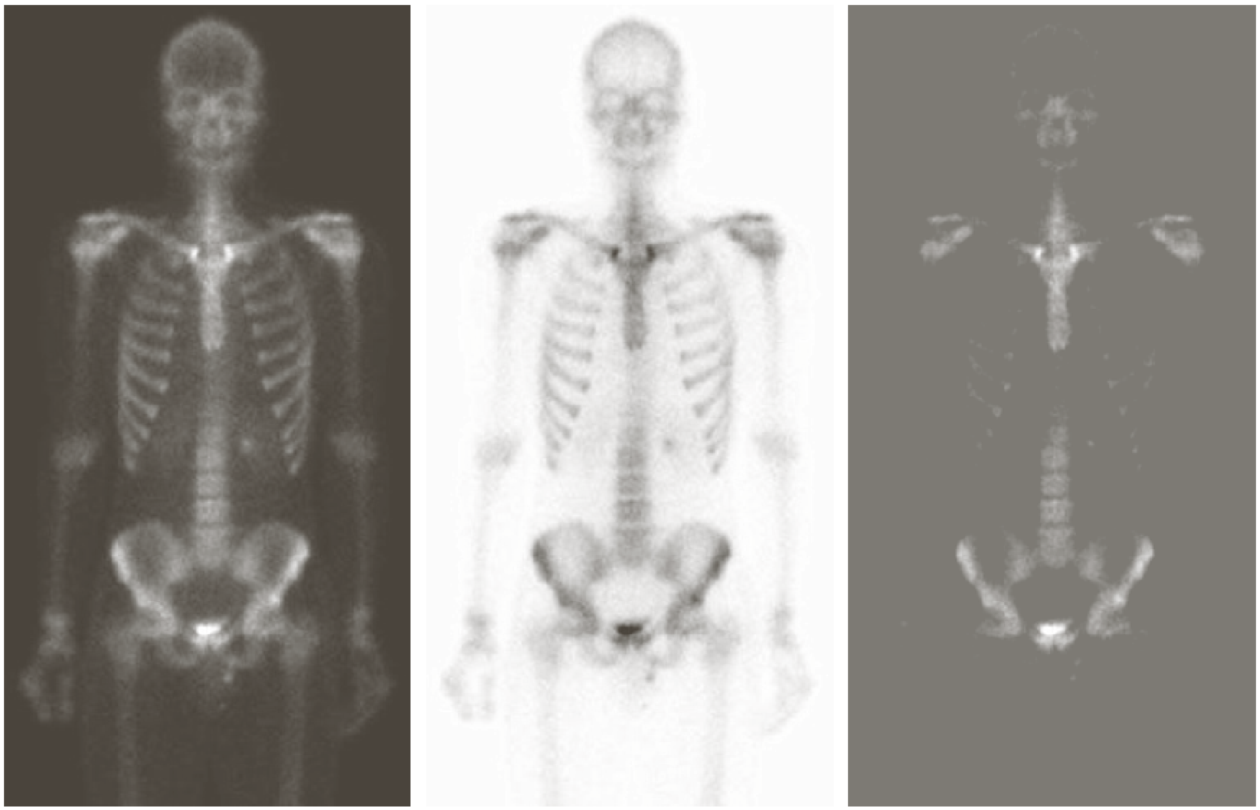


FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

a b c

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)



a b c

FIGURE 2.32 Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

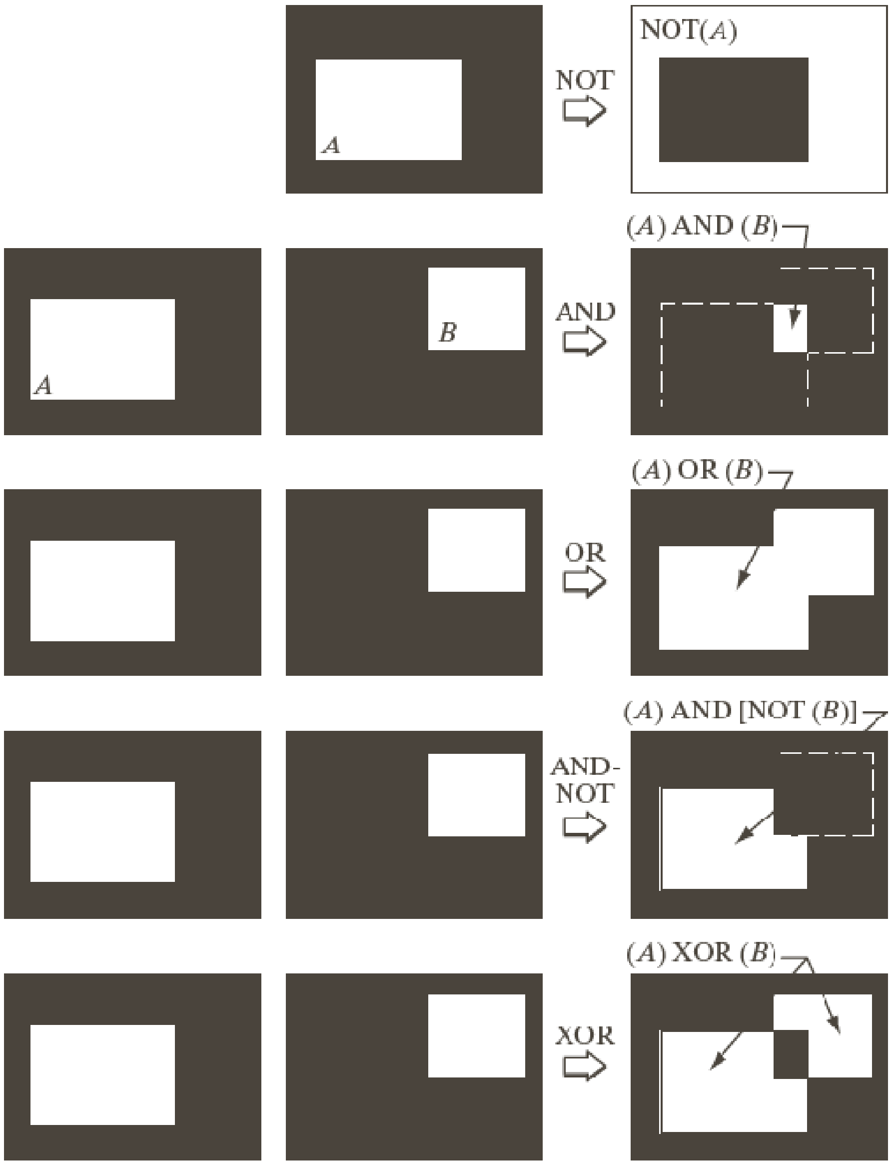


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Mask Operation

- Besides pixel-by-pixel processing on entire images, arithmetic and Logical operations are used in neighborhood oriented operations.
- The value of a pixel is arithmetic result of its neighbors.

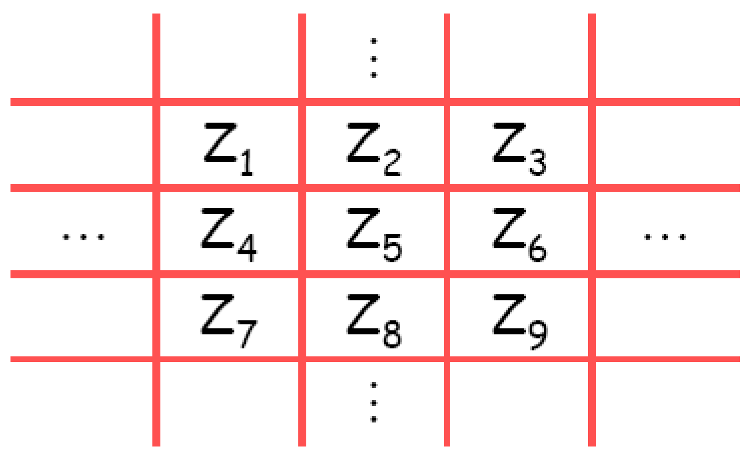
e.g

$$Z = \sum_{i}^{9} w_i Z_i$$

W_i are called mask coefficients.

When $w_i = 1/9, i = 1, \ldots, 9$

$$Z = \frac{1}{9} \sum_{i}^{9} Z_i$$



Mask coefficient

- Proper selection of the coefficients and application of the mask at each pixel position in an image makes possible a variety of useful image operations
  - noise reduction
  - region thinning
  - edge detection

- Applying a mask at each pixel location in an image is a computationally expensive task.

More mathematics will be involved.

Image Geometry Transformation

- Geometry transformation

  p(x, y) => p' (x', y')

- Basic transformations
  - Translation
  - Scaling
  - Rotation
- The composition of transformations

- Transformation matrix

$$p' = Tp$$

Translate

- Translate point P to point P' along a vector (x_0, y_0)
- Homogeneous coordinates

$$x' = x + x_0$$

$$y' = y + y_0$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, T(x_0, y_0) = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation

$$x' = x\cos\theta + y\sin\theta$$

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

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

$$T_x(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Rotate w.r.t x -axis

Scaling

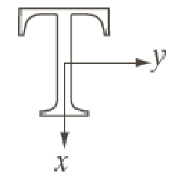
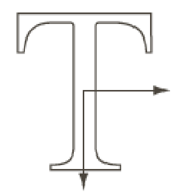
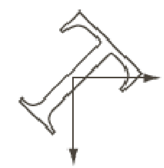
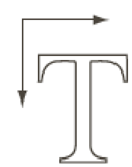
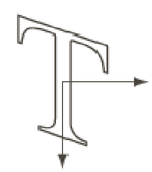
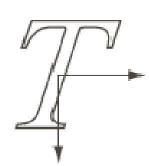
$$x' = c_x x$$

$$y' = c_y y$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$T_s(c_x, c_y) = \begin{pmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

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

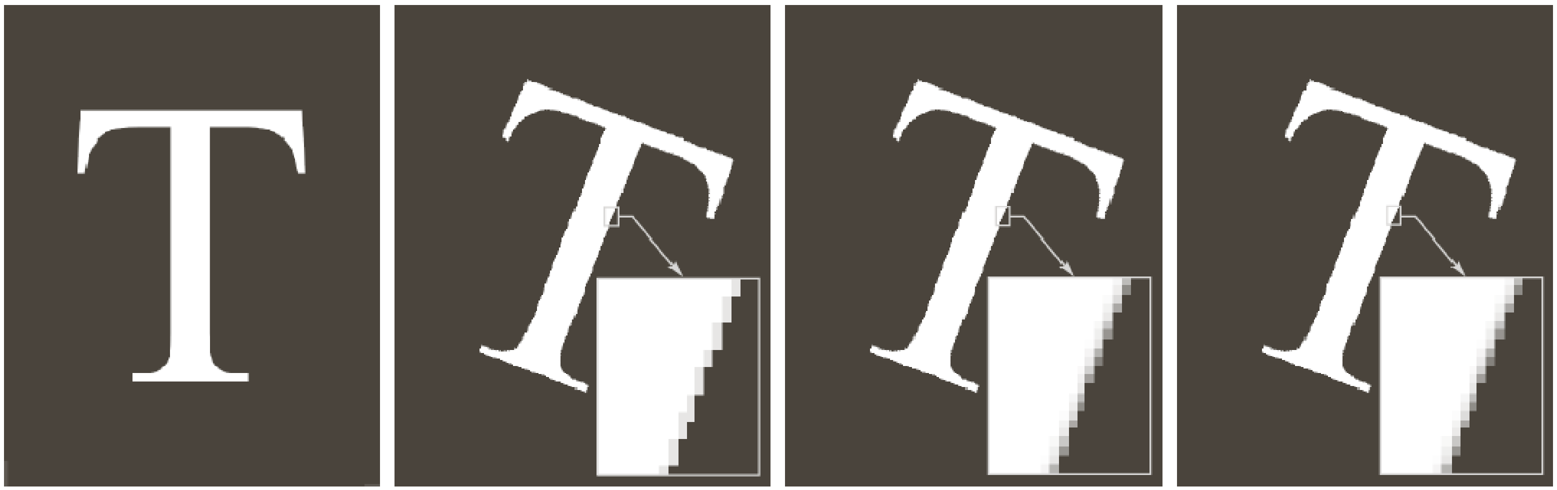
Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta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

Composition of Transformations

- The transformation matrix of a sequence of transformation is equal to the product of transformation matrices of each individual matrix.

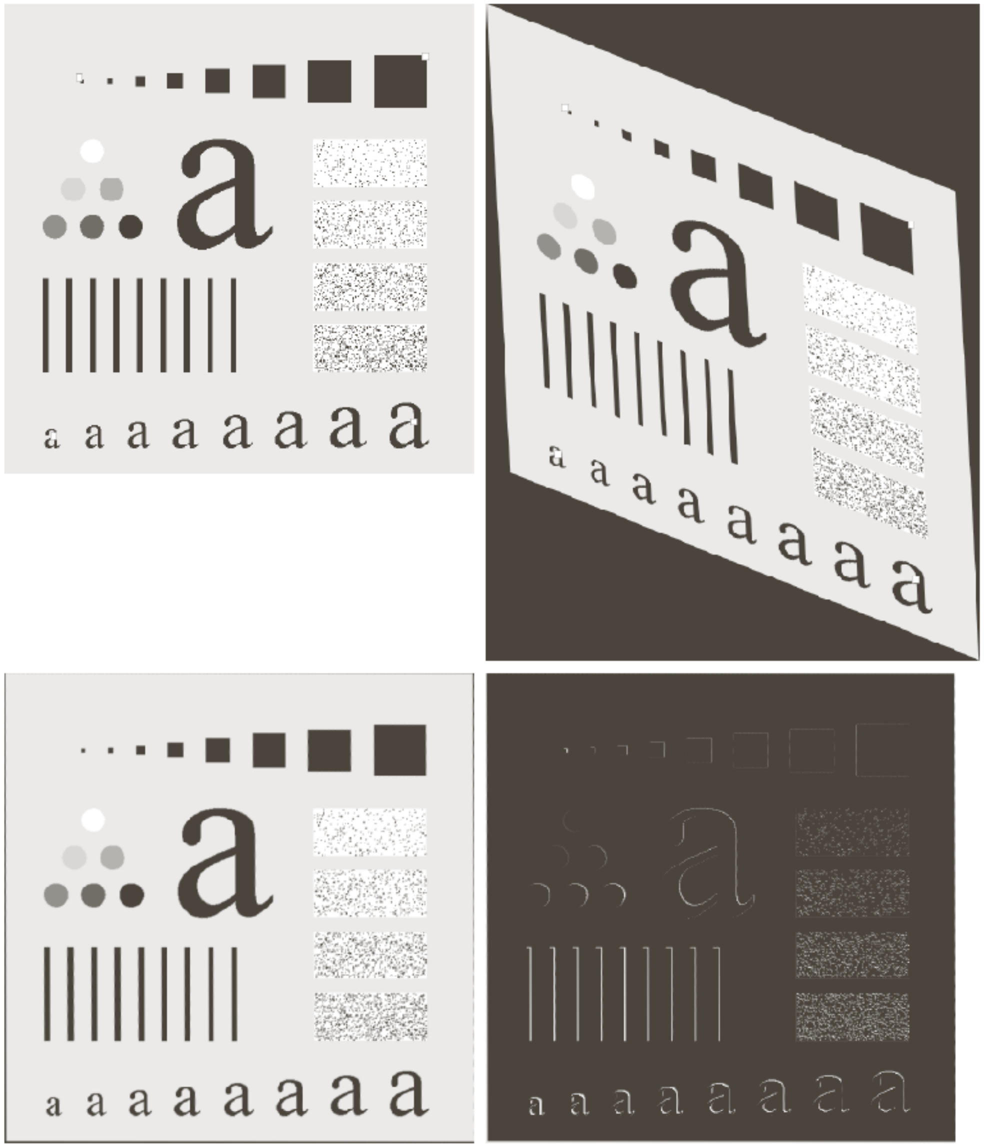
$$P_1 \xrightarrow{T_1} P_2 \xrightarrow{T_2} \ldots \xrightarrow{T_{k-1}} P_k$$

$$P_k = TP_1, T = ?$$



a b c d

FIGURE 2.36 (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.



a b
c d

FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.

Image Transformation

- Input an image output another image

$$f(x,y) \xrightarrow{T} g(x,y)$$
$$g = T(f)$$

- Geometry transformation

$$g(x,y) = f(M(x,y)')$$

- Intensity transformation

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

- These transformation works directly on pixels.
  They are spatial transformation, or in spatial domain.

Image Transformation

- The transformation can be complicate in transformation domain

$$f(x,y) \xrightarrow{T} g(x,y), g = T(f)$$

$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) r(x,y,u,v)$$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) s(x,y,u,v)$$

Image Transformation

- Example: 2-D Fourier transformation

$$r(x, y, u, v) = e^{-j2\pi(ux/M + vy/N)}$$

$$s(x, y, u, v) = \frac{1}{MN} e^{j2\pi(ux/M + vy/N)}$$

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

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