Digital Image Processing 18MIT31C

UNIT-II: Image Enhancement (Spatial domain): - Introduction-Basic Gray Level Transformations-Histogram Processing-Arithmetic/Logic Operations-Basics of spatial filtering-Smoothing-Sharpening. Image restoration (Spatial domain): Model of the Image degradation/Restoration Process-Noise Models-Noise reduction filters-Mean Filters-Order statistics filters-Adaptive filters.

Textbook :

Gonzalez R C., and Woods R.E., “Digital Image Processing”, Prentice Hall, Third Edition.

Prepared By : Mrs. G. Shashikala, Assistant Professor, Department of Information Technology

- Spatial domain refers to the image plane itself, and image processing methods in this category are based on direct manipulation of pixels in an image.
- Two principal categories of spatial processing are intensity transformations and spatial filtering.
- Intensity transformations operate on single pixels of an image for the purpose of contrast manipulation and image thresholding.
- Spatial filtering deals with performing operations, such as image sharpening, by working in a neighborhood of every pixel in an image.

The Basics of Intensity Transformations and Spatial Filtering

- Generally, spatial domain techniques are more efficient computationally and require less processing resources to implement.
- The spatial domain processes can be denoted by the expression $g(x, y) = T[f(x, y)]$ where $f(x, y)$ is the input image, $g(x, y)$ is the output image, and T is an operator on f defined over a neighbourhood of point (x, y).
- The operator can apply to a single image or to a set of images.

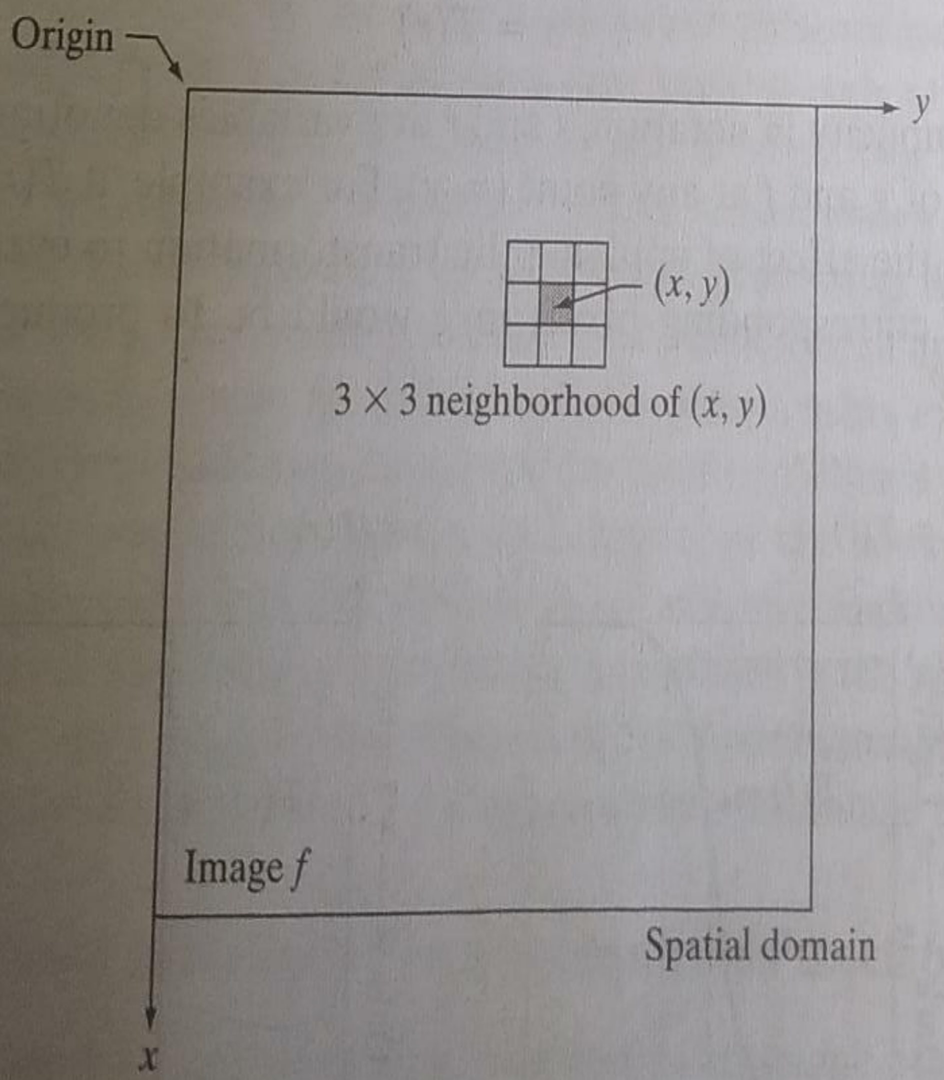
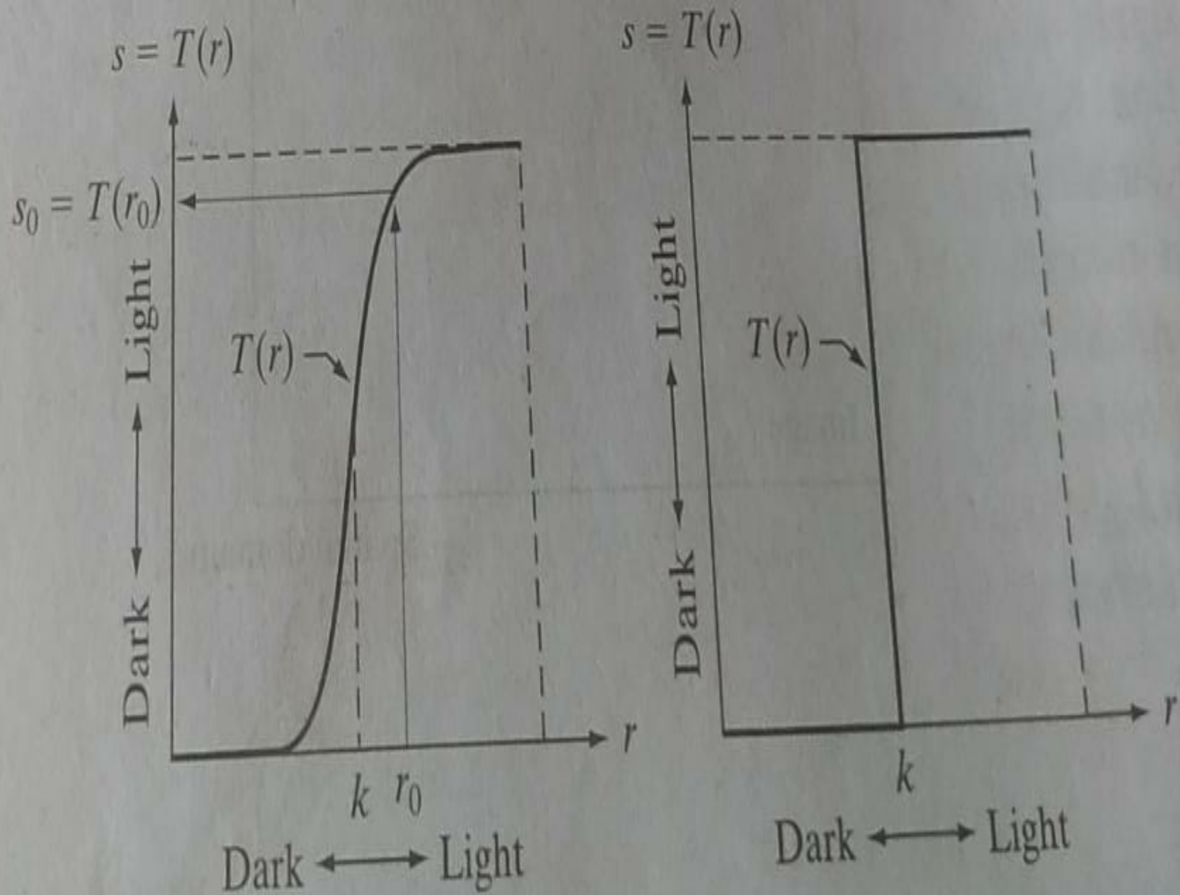


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

- 1 x 1 is the smallest possible neighborhood.
- In this case g depends only on value of f at a single point (x,y)
  and we call T an ***intensity (gray-level mapping) transformation*** and write

$$s = T(r)$$

where r and s denotes respectively the intensity of g and f at any point (x, y).



a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

Some basic intensity transformation functions

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

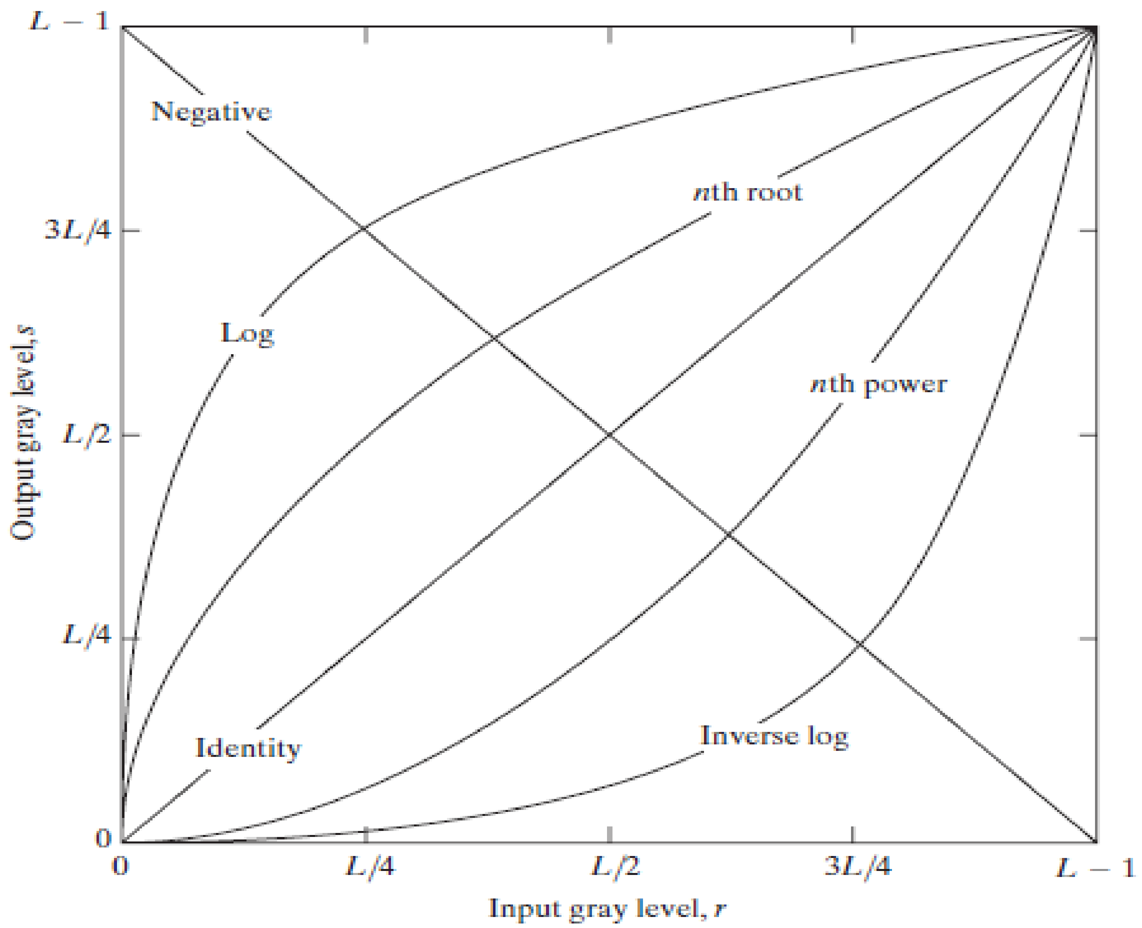


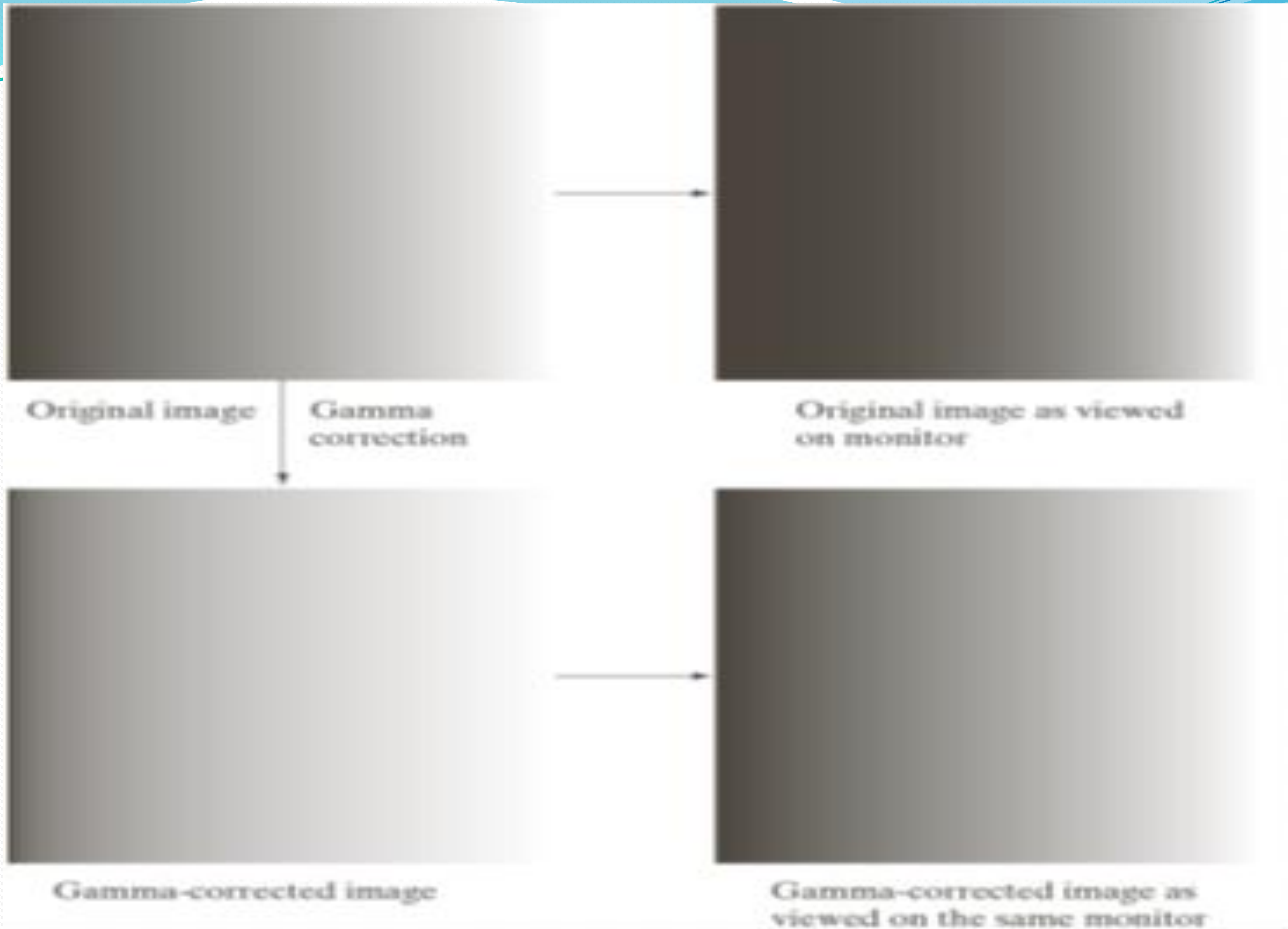
Image Negatives

- With intensity levels of the image in the range [0, L-1]
- Image negative is obtained by $s = L-1-r$
- Log Transformations

The general form of log transformations is

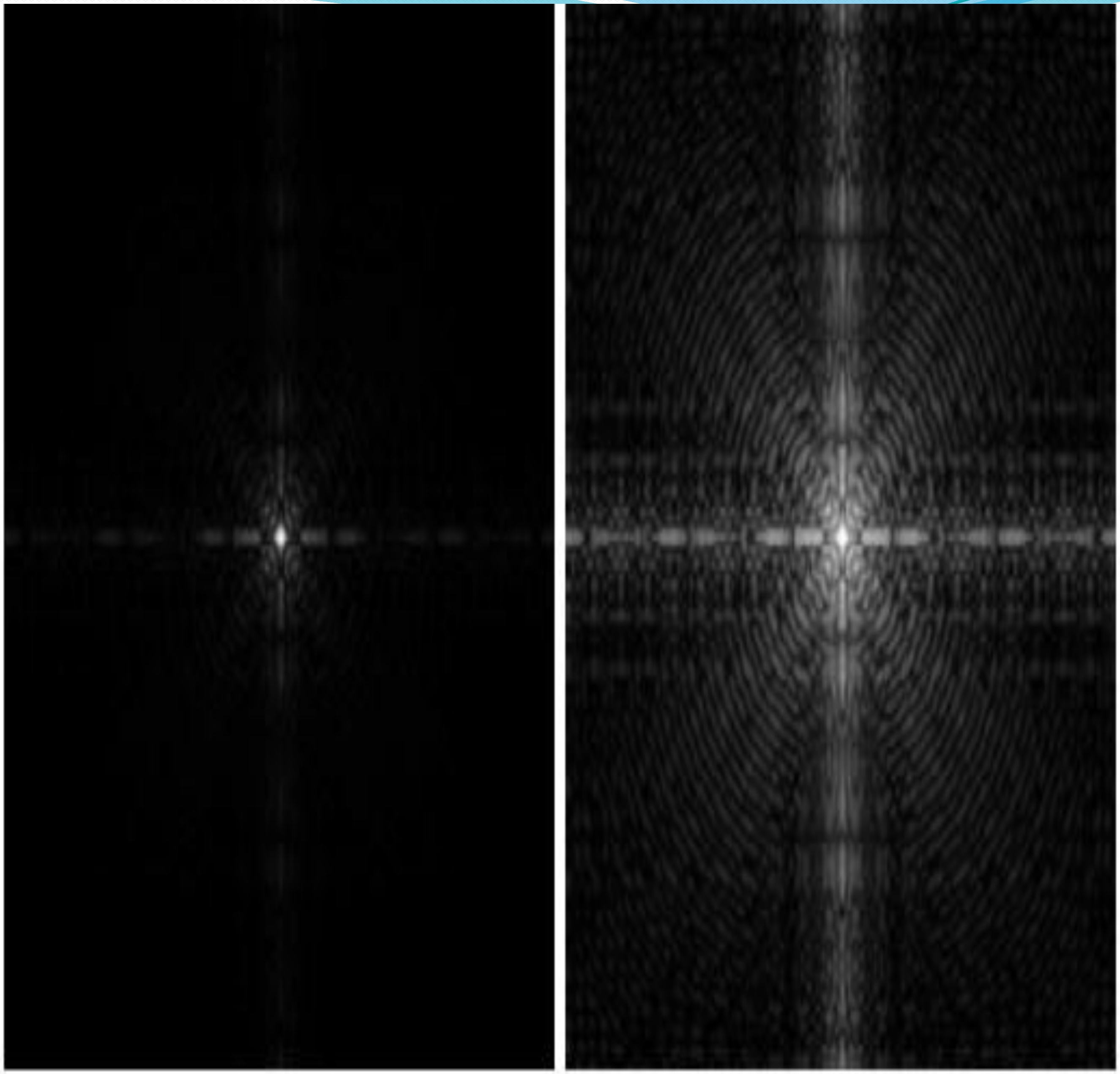
$$s = c \log(1+r),\ c - const,\ r \geq 0$$

This maps a narrow range of low intensity values in the input into a wider range of output levels. The opposite is true for higher values of input levels.


Original image
Gamma correction
Original image as viewed on monitor
Gamma-corrected image
Gamma-corrected image as viewed on the same monitor

a b

FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Power–Law (Gamma) transformation

$s = cr^{\gamma}$, c, γ *–positive constants*

---Used to either to brighten the intensity (when $\gamma < 1$) or darken the intensity (when $\gamma > 1$).

The above equation can also be written as

$$s = c.(r+\epsilon)^{\gamma}$$

To account for an offset (i.e., a measurable o/p when the i/p is zero

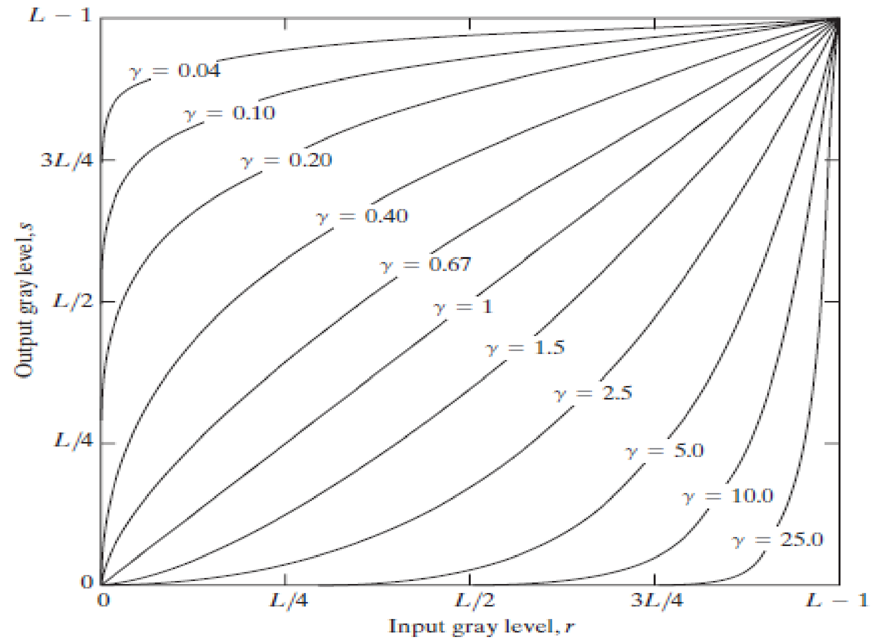


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Piecewise –Linear Transformation Functions

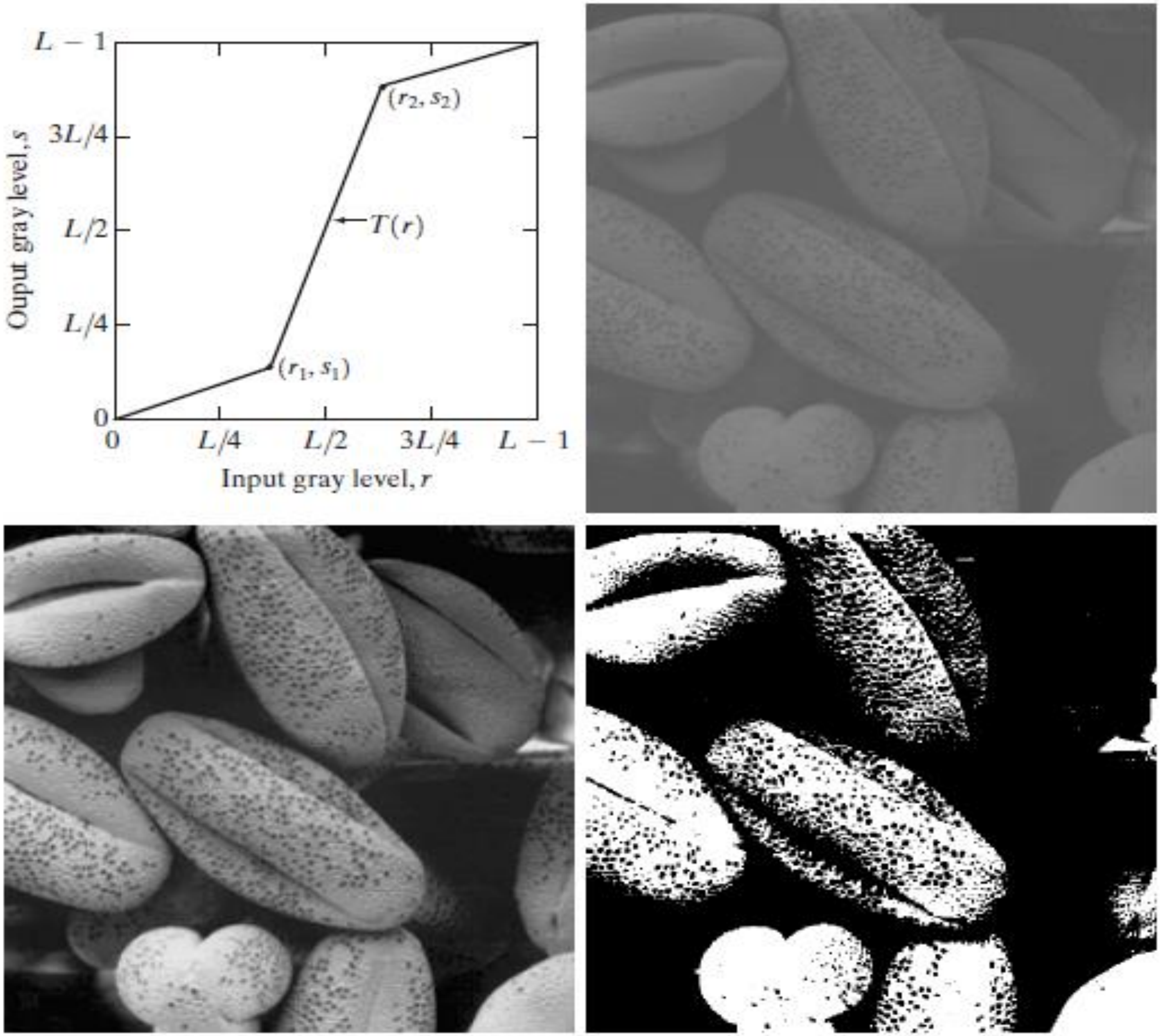
- Contrast Stretching

Contrast stretching is a process that expands the range of intensity levels in a image so that it spans the full intensity range of the recording medium or display device.

Contrast-stretching transformations increase the contrast between the dark and the light values

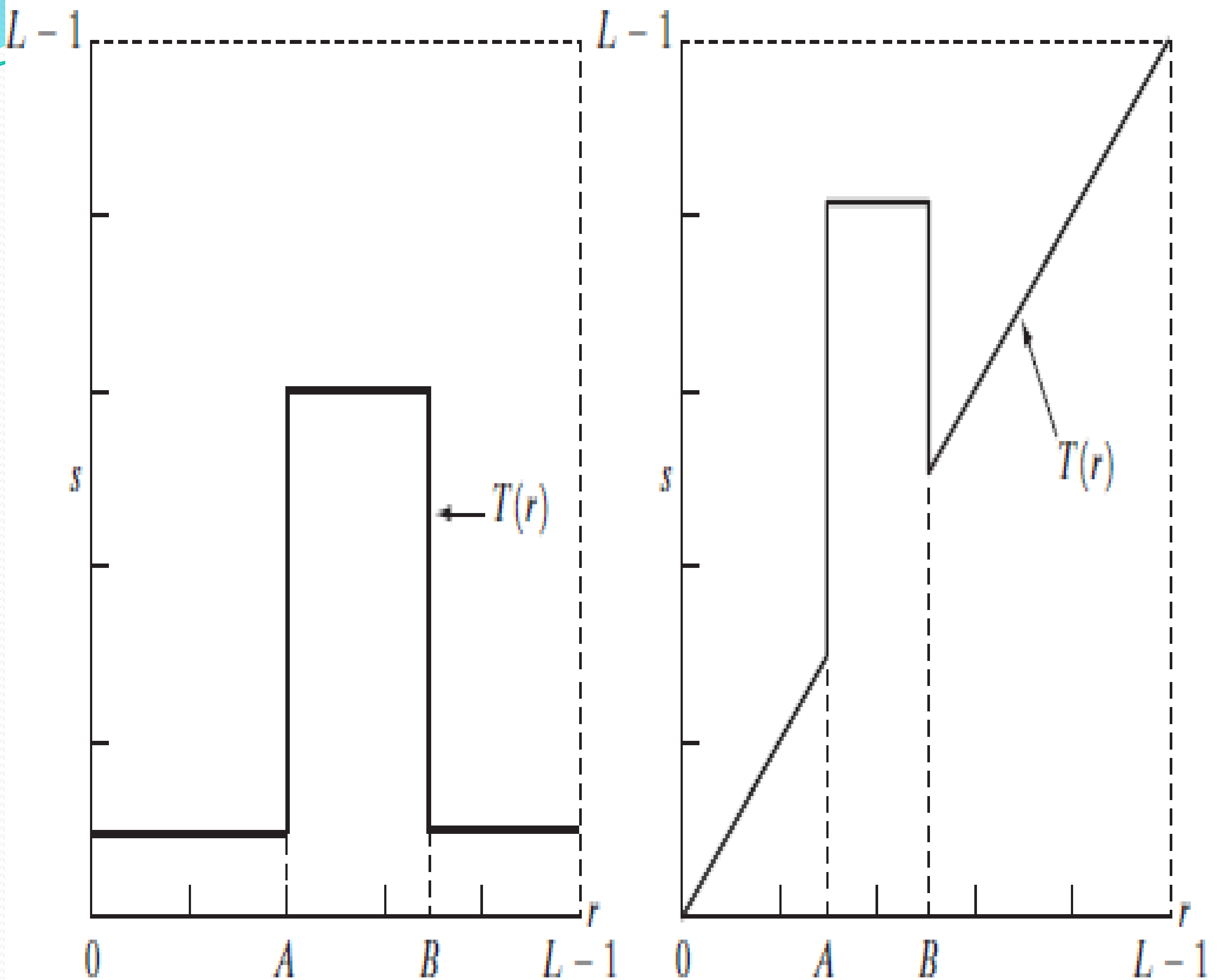
- Intensity – level slicing
- Highlighting a specific range of gray levels in an image
- Two approaches :
- To display in one value (white) all the values of interest and in another (black) , all other intensities
- Brightens or darkens the desired range of intensities and leaves other intensities unchanged.

- Bit-plane slicing
- Pixels are digital numbers composed of bits.
- The intensity of each pixel in a 256- level gray scale image is composed of 8-bits.
- An 8-bit plane may be considered as being composed of eight 1-bit planes, with plane 1 containing the lower order bits of all pixels in the image and plane 8 all the higher order bits



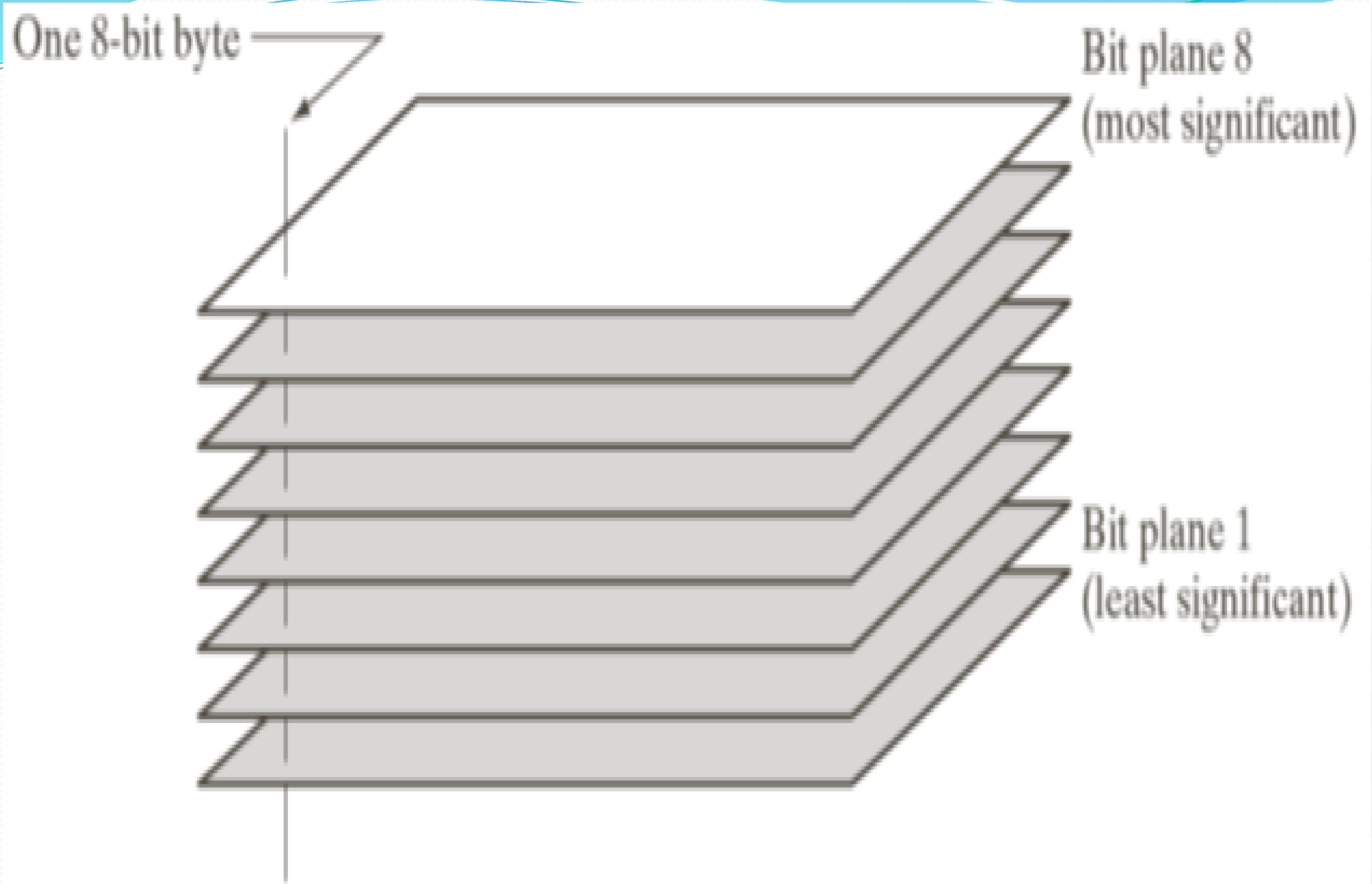
a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)



a b
c d

FIGURE 3.11 (a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels.


One 8-bit byte
Bit plane 8
(most significant)
Bit plane 1
(least significant)

a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

Histogram

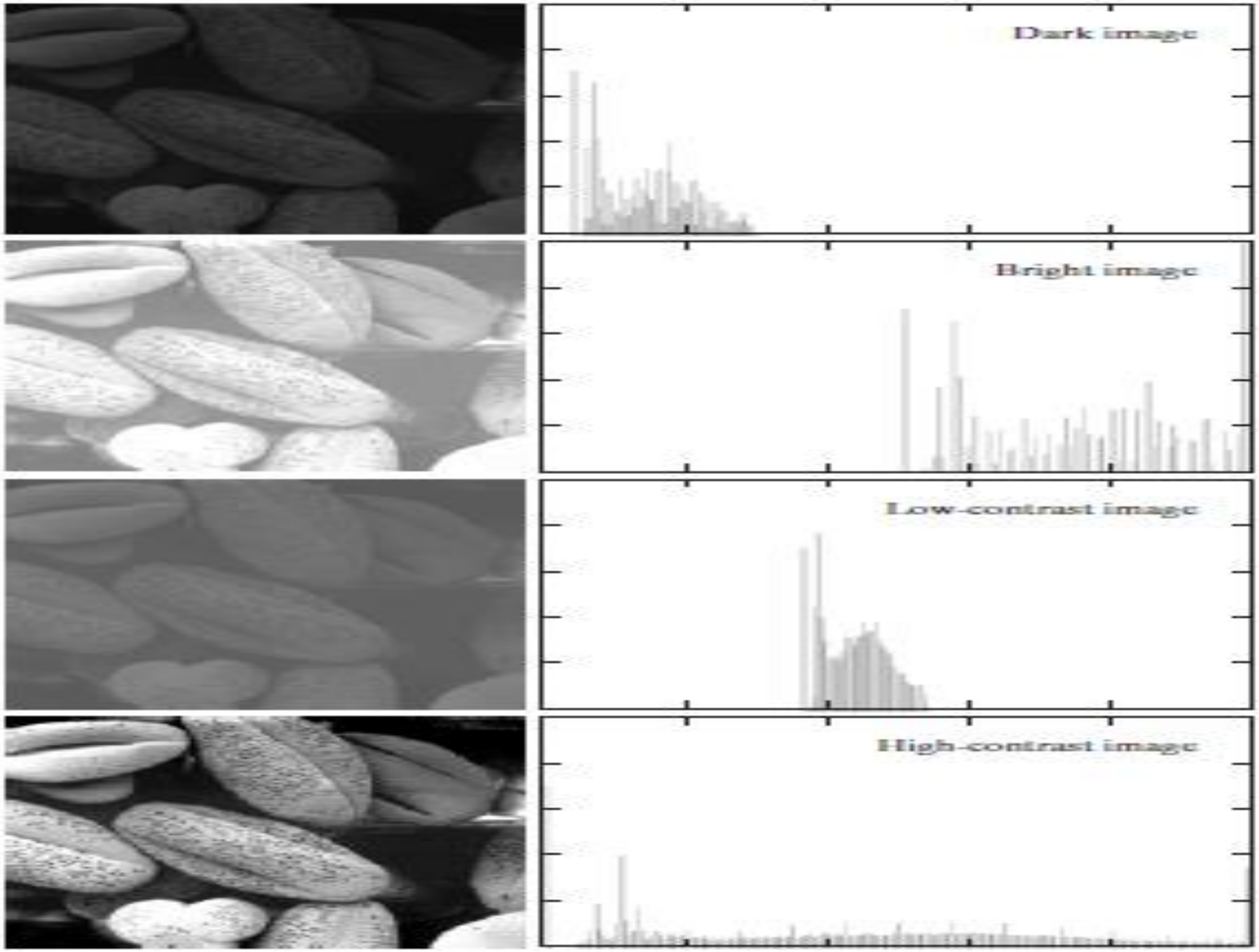
The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $h(r_k)=n_k$, where r_k is the *kth gray level and* n_k *is the number* of pixels in the

image having gray level r_k. It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image,

denoted by *the product MN.*

Thus, a normalized histogram is given by $h(r_k)=n_k/MN$

The sum of all components of a normalized histogram is equal to 1.



a b

FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Histogram Equalization

- Histogram equalization can be used to improve the visual appearance of an image.
- ***Histogram equalization*** automatically determines a transformation function that produce and output image that has a near uniform histogram
- s = T®, 0<=r<=L-1

a. T(r) is a strictly monotonically increasing function in the interval $0 \le r \le L-1$;

b. $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.

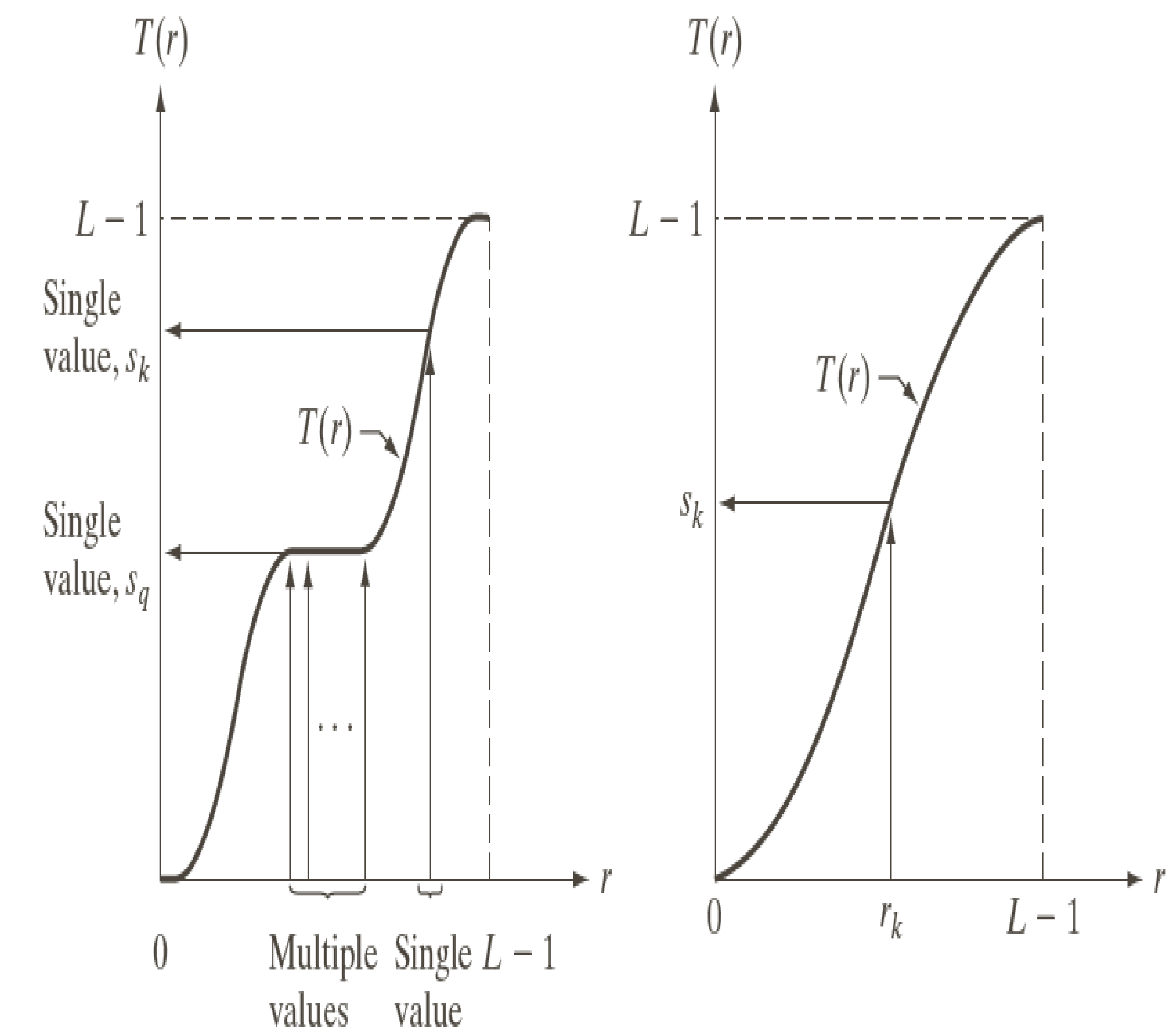
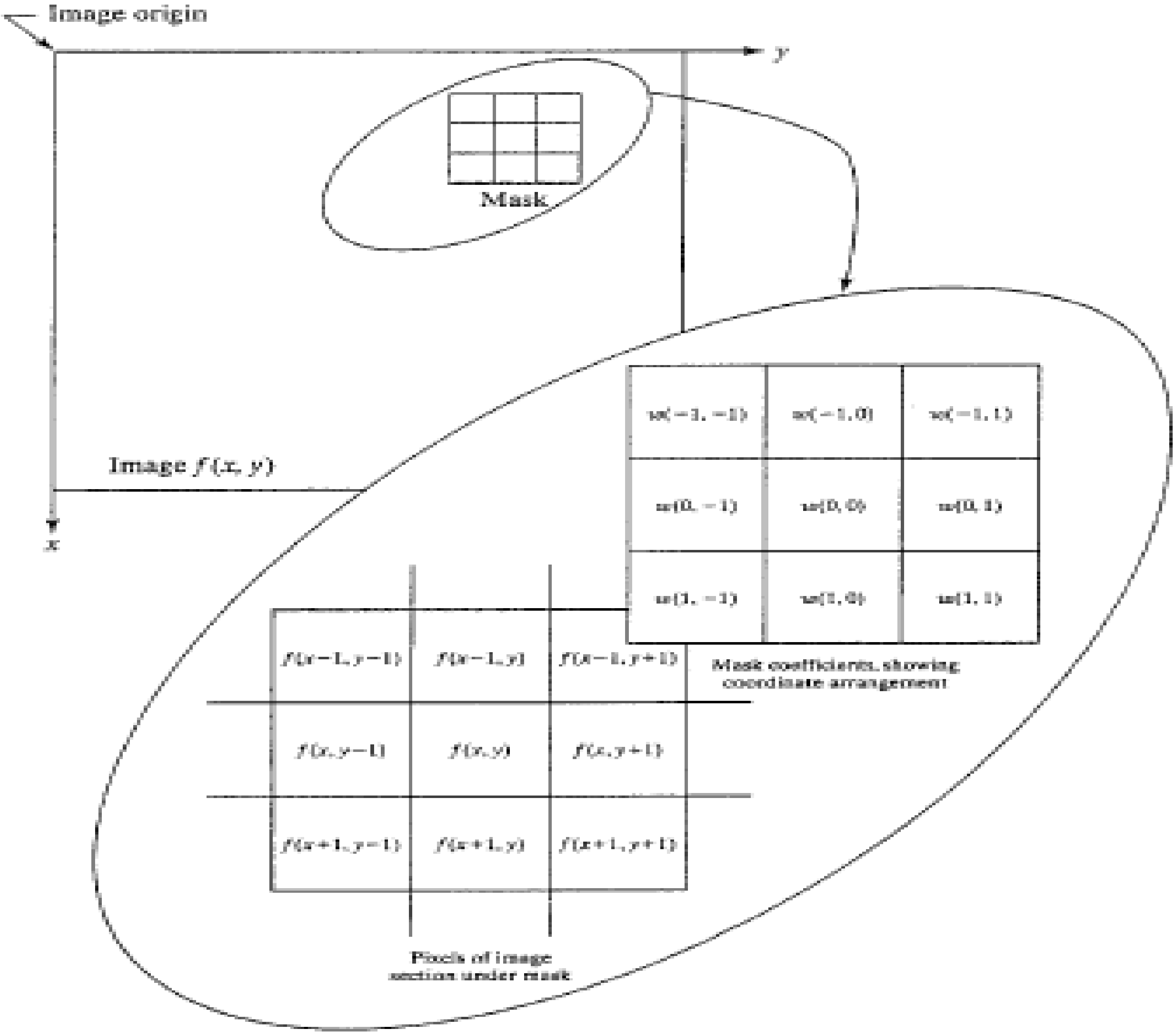


FIGURE 3.17
(a) Monotonically increasing function, showing how multiple values can map to a single value.
(b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

The mechanics of spatial filtering

- A spatial filter consists of (i) a neighborhood, (ii) a predefined operation that is performed on the image pixels encompassed by the neighborhood.
- Filtering creates a new pixel with co-ordinates equal to the co-ordinates of the center of the neighborhood, and whose value is the result of the filtering operation.
- A processed image (filtered) is generated as the center if the filter visits each pixel in the input image. If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter. otherrwise., the filter is non-linear
- $g(x, y) = w(-1, -1) f(x-1, y-1) + \cdots + w(1, 1) f(x+1, y+1)$

FIGURE 3.12 The mechanics of linear spatial filtering. The magnified drawing shows a 3 × 3 mask and the corresponding image neighborhood directly under it. The neighborhood is shown displaced out from under the mask for ease of readability.



Linear Spatial Filtering

- For linear spatial filtering, the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. For the 3 X 3 mask shown in the previous figure, the result (response), R, of linear filtering with the filter mask at a point (x,y) in the image is:

  $g(x,y) = w(-1,-1)\ f(x-1, y-1) + w(-1,0)\ f(x-1, y) + \dots + w(0,0)\ f(x,y) + \dots + w(1,0)\ f(x+1, y) + w(1,1)\ f(x+1, y+1)$

  The center coeff. of the filter , w(0,0) , aligns with the pixel at location (x,y).
- For a mask of size, mxn, we assume that m=2a+1 and n=2b+1, where a and b are positive integers.
- Linear spatial filtering of an image of size MxN with a filter of size mxn, is given by the expression :

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$\text{where } a = (m-1)/2, \quad b = (n-1)/2$$

$$x = 0, 1, 2, \ldots\ldots, M-1, \quad y = 0, 1, 2, \ldots\ldots, N-1,$$

Spatial correlation and convolution

- Correlation is the process of moving a filter mask over the image and computing the sum of products at each location
- In convolution, the filter is first rotated by 180 degrees , and the same process is done.

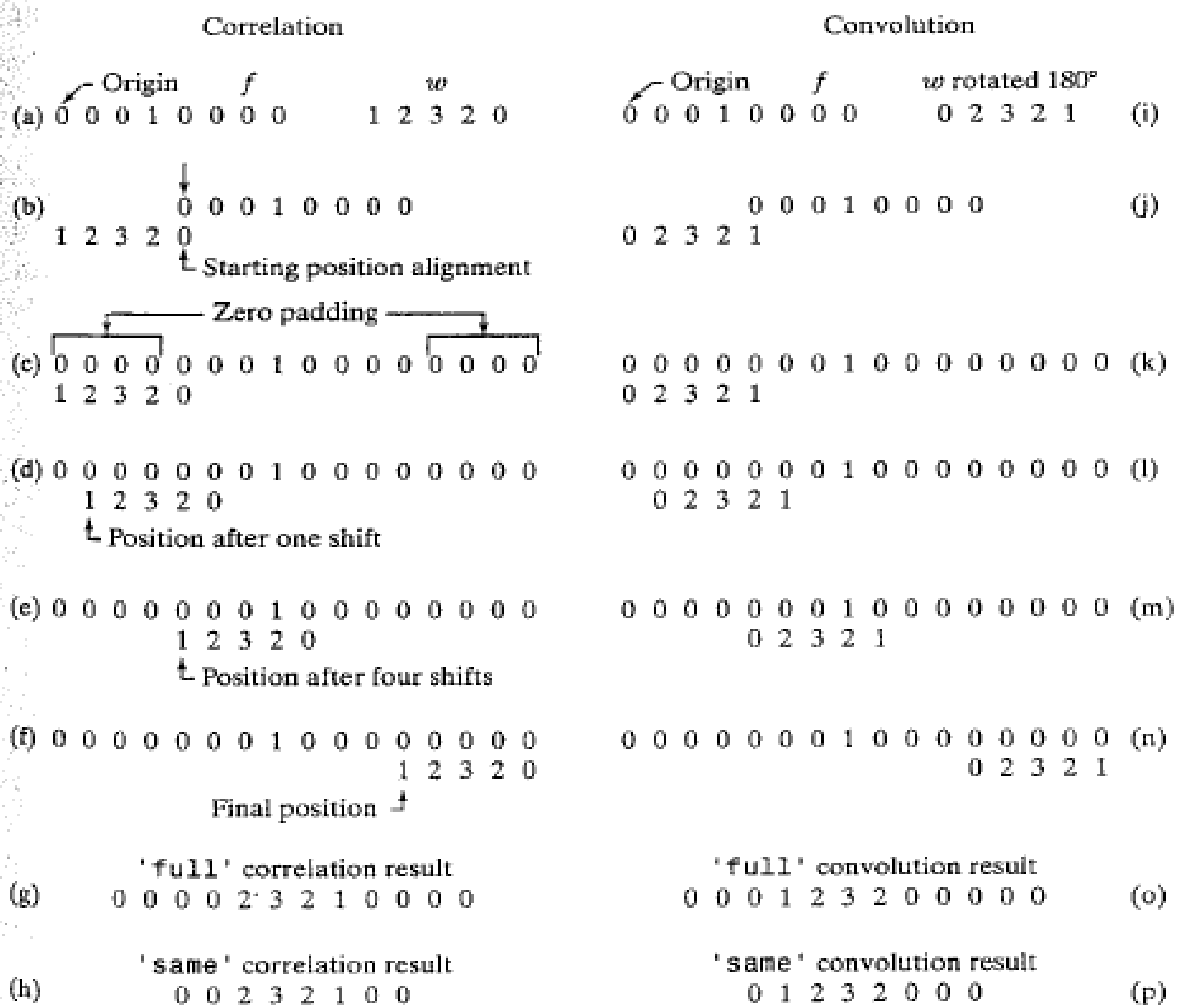


FIGURE 3.13 Illustration of one-dimensional correlation and convolution.

Correlation

(a) Origin f: 0 0 0 1 0 0 0 0 w: 1 2 3 2 8

(b) 0 0 0 1 0 0 0 0
1 2 3 2 8
Starting position alignment

Zero padding

(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8

(d) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Position after one shift

(e) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Position after four shifts

(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Final position

Full correlation result

(g) 0 0 0 8 2 3 2 1 0 0 0 0

Cropped correlation result

(h) 0 8 2 3 2 1 0 0

Convolution

Origin f: 0 0 0 1 0 0 0 0 w rotated 180°: 8 2 3 2 1 (i)

0 0 0 1 0 0 0 0 (j)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
8 2 3 2 1

Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0 (o)

Cropped convolution result

0 1 2 3 2 8 0 0 (p)

URE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note
relation and convolution are functions of *displacement*.

The solution to this problem is to

Origin $f(x, y)$

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

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

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

(b)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(f)

Full convolution result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(g)

Cropped convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

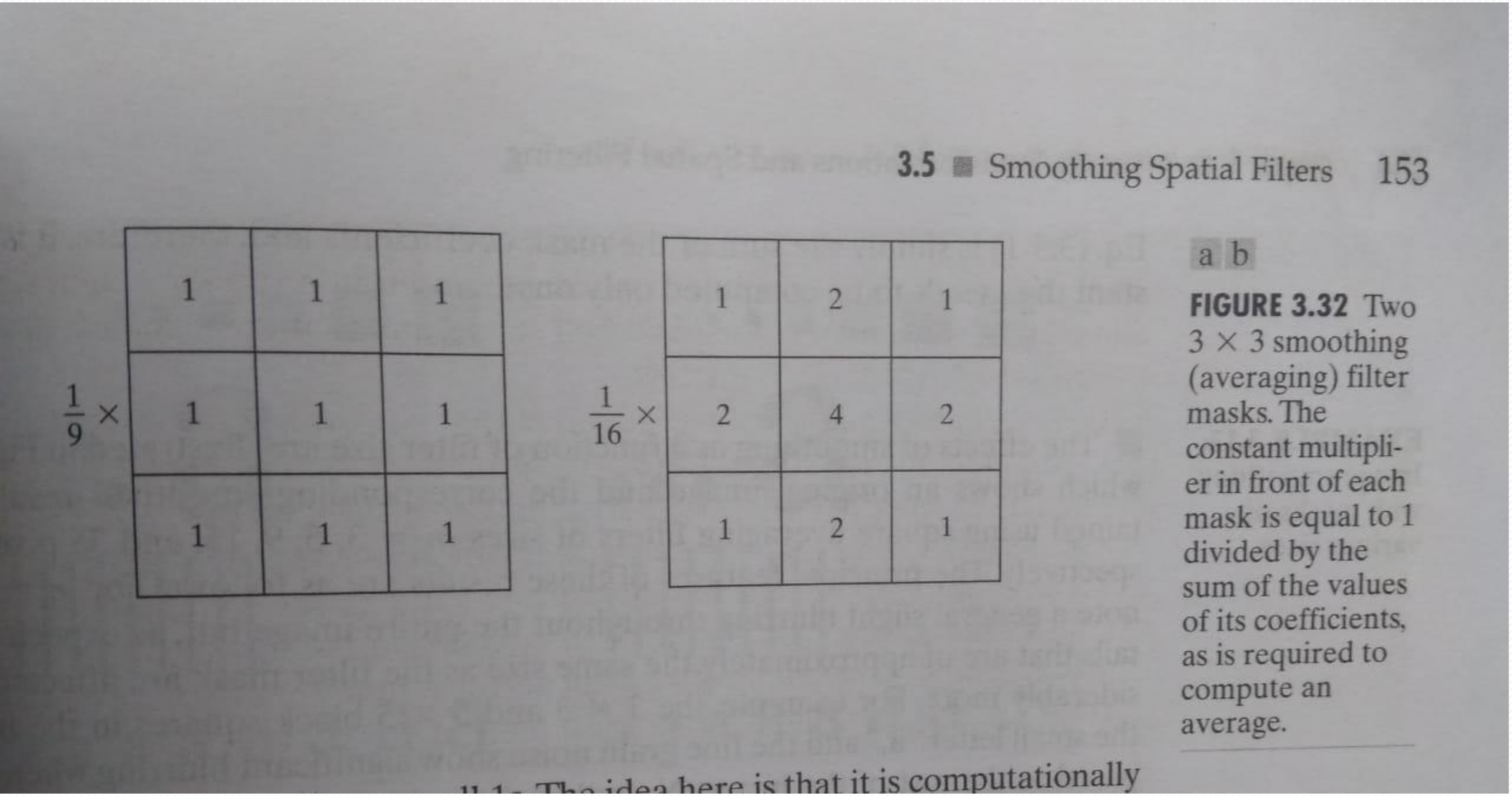
(h)

FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Smooting spatial filters

- Smoothing filters are used for blurring and for noise reduction
- Blurring is used for preprocessing tasks, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves.
- Noise reduction can be accomplished by blurring with linear filtering or non-linear filtering

Smoothing Linear Filters



$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

a b

FIGURE 3.32 Two 3 × 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

The idea here is that it is computationally spanned by these masks at any one location in an image is so small.

With reference to Eq. (3.4-1), the general implementation for filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ (m and n odd) is given by the expression

$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)} \qquad (3.5\text{-}1)$$

The parameters in this equation are as defined in Eq. (3.4-1). As before, it is un-

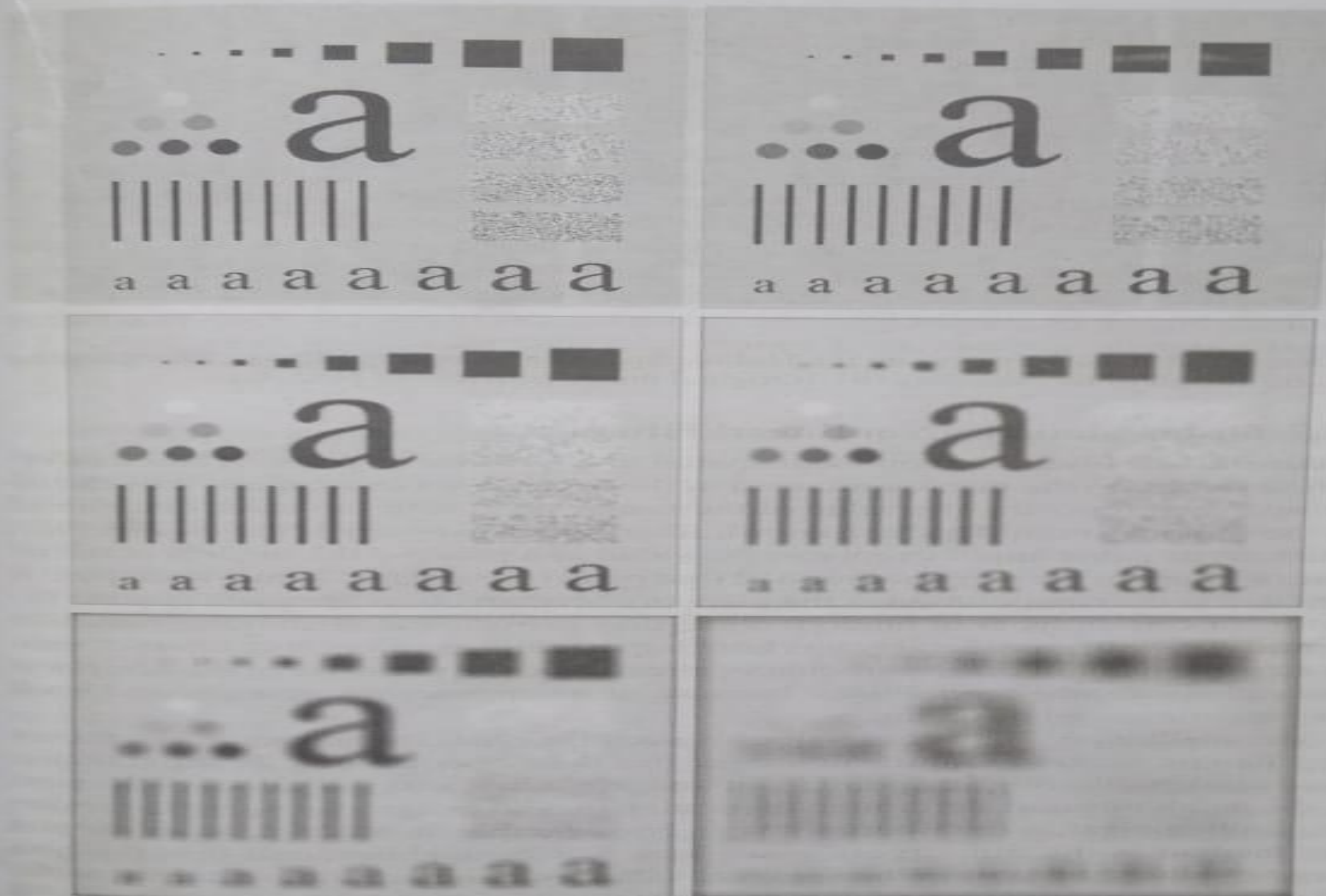


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes m = 3, 5, 9, 15, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f

Order-statistics(Non-linear)Filters

- Order statistic filters are non-linear spatial filters whose response is based on the ordering(ranking) of the pixels contained in the image area encompassed by the filter, and then replacing the value in the center pixel with the value determined by the ranking result.

- The different types of order statistics filters include Median Filtering, Max and Min filtering and Mid-point filtering.

- Median Filtering : Replaces the value of a pixel by the median of the pixel values in the neighborhood of that pixel

  Median filters are used to remove impulse noise also called salt and pepper noise which are white and black dots superimposed on an image

  First the values of the pixels in the neighborhood are sorted, the median is identified and that value is assigned to the corresponding pixel in the filtered image

- The median represents the 50^{th} percentile of a ranked set of numbers. The use of 100^{th} percentile results in the max filter given by R=max{Zk/k=1,2,…9}
- The zeroth percentile filter is the min filter
- Max & Min Filtering : The max filtering is achieved using the following equation
- $f(x,y) = \max g(s,t)$
- The min filtering is achieved using the following equation
- $f(x,y) = \min g(s,t)$
- Mid-point filtering : Replaces the value of a pixel by the midpoint between the maximum and minimum pixels in a neighborhood

Sharpening spatial filters

- The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as an natural effect of a particular method of image acquisition.
- Blurring vs Sharpening
  - Blurring/smooth is done in spatial domain by pixel averaging in a neighbors, it is a process o f integration
  - Sharpening is an inverse process, to find the difference by the neighborhood, done by spatial differentiation.

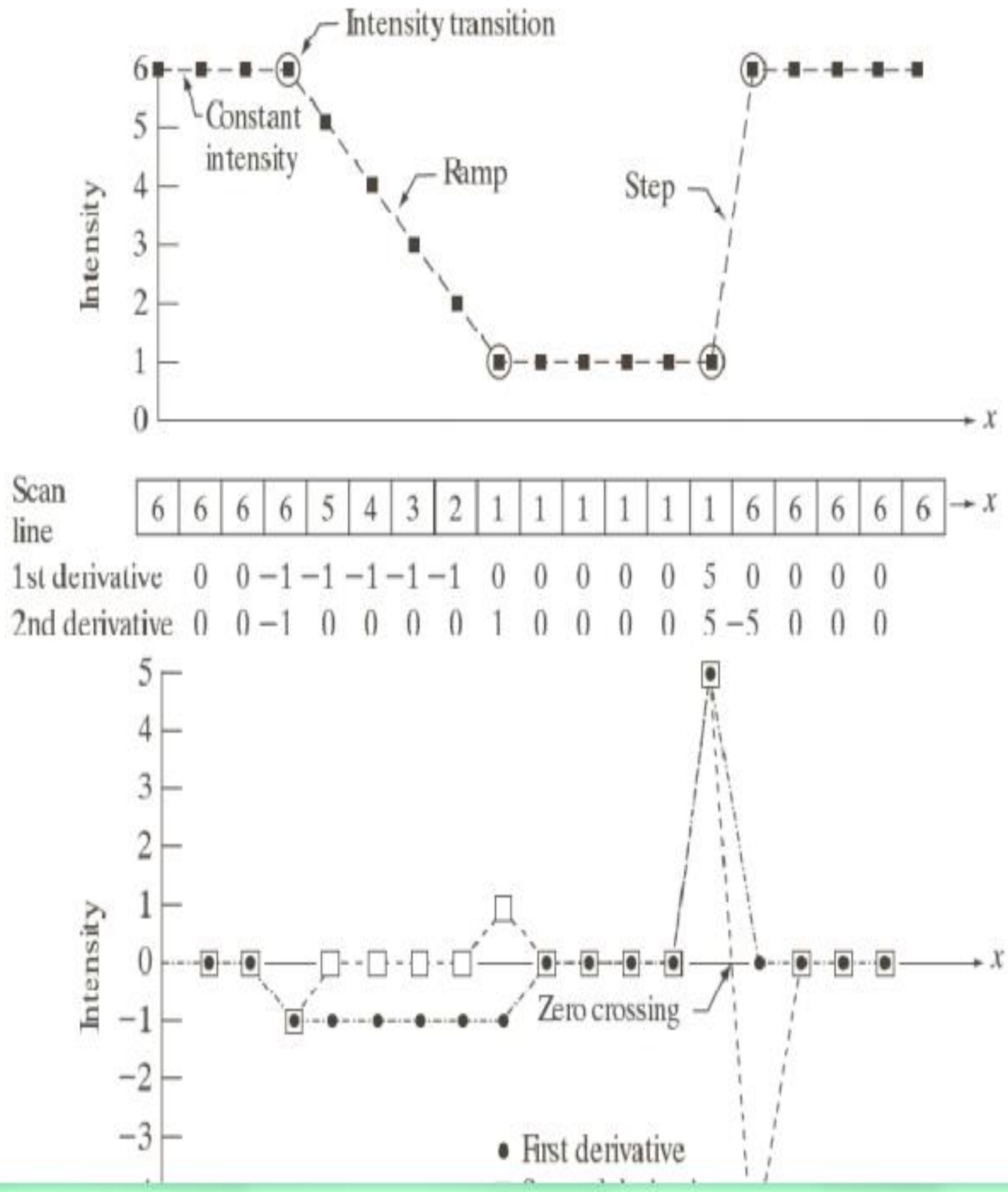
First and second order difference of 1D

- The basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- The second-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$



a
b
c

FIGURE 3.36
Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Sharpening spatial filters

- Sharpening is used to highlight transitions in intensity. It is accomplished by spatial differentiation. Image blurring in the spatial domain is done by pixel averaging in a neighborhood
- <u>Foundation</u>
  - The discontinuities in a signal are used to model noise points, lines and edges in an image

- The derivatives of a digital function are defines in terms of differences. It is required that the first derivative value
  (i) Must be zero in areas of constant intensity
  (ii) Must be non-zero at the onset of an intensity step or ramp, and
  (iii) Must be non-zero along ramps

- The zero-crossing property is useful for locating edges. Edges are ramp-like transitions in intensity , in which case, the first derivative would result in thick edges, because the derivative is non-zero along a ramp. The second derivative produce a double edge one pixel thick, separated by zeroes

Image restoration and reconstruction

The sources of noise in digital images arise during image acquisition (digitization) and transmission

- Imaging sensors can be affected by ambient conditions
- Interference can be added
  to an image during transmission

We can consider a noisy image to be modelled as follows:

$$g(x,y)=f(x,y)*h(x,y)+\eta(x,y)$$

where $f(x, y)$ is the original image pixel, $\eta(x, y)$ is the noise term and $g(x, y)$ is the resulting noisy pixel

If we can estimate the noise model we can figure out how to restore the image

A model of the image degradation/restoration process

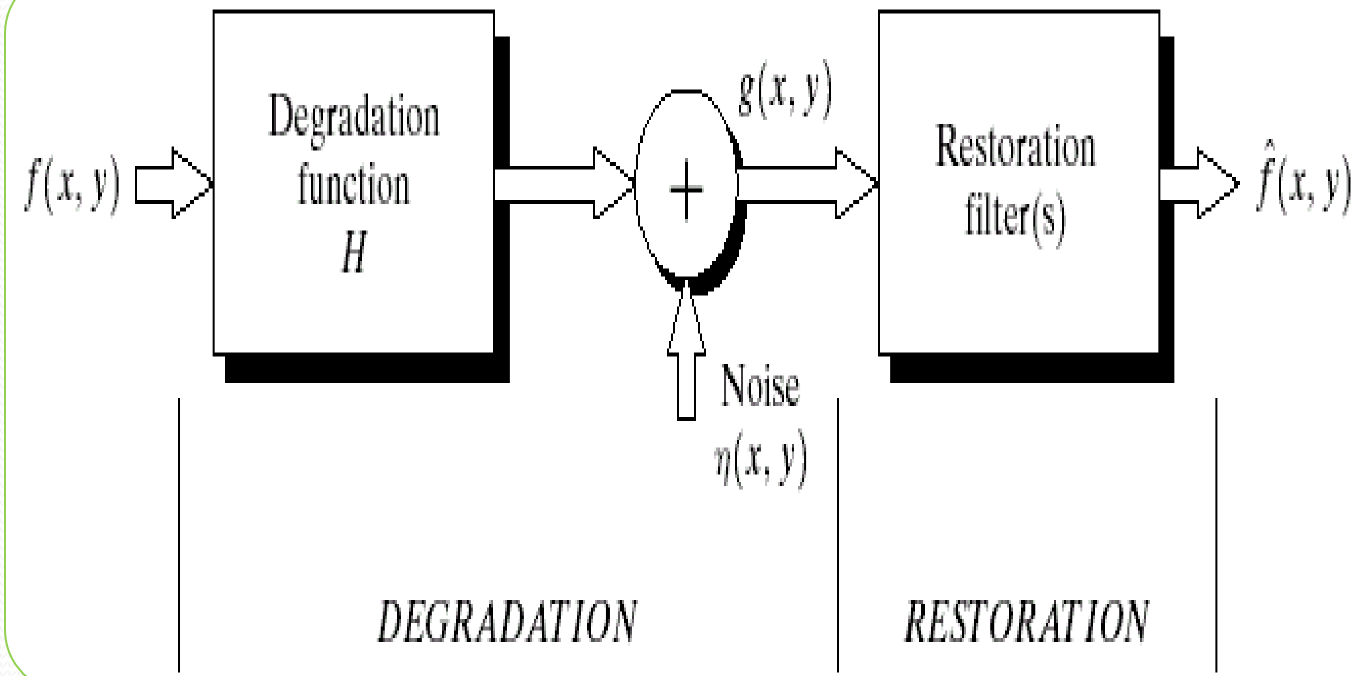


FIGURE 5.1 A model of the image degradation/ restoration process.

Mean Filters

- This is the simply methods to reduce noise in spatial domain.
  - Arithmetic mean filter
  - Geometric mean filter
  - Harmonic mean filter
  - Contraharmonic mean filter
- Let S_{xy} represent the set of coordinates in a rectangular subimage window of size mxn, centered at point (x,y).

Arithmetic mean filter

- Compute the average value of the corrupted image g(x,y) in the aread defined by $S_{x,y}$.
- The value of the restored image $\hat{f}$ at any point (x,y)

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{x,y}} g(s,t)$$

Note: Using a convolution mask in which all coefficients have value 1/mn. Noise is reduced as a result of blurring.

Geometric mean filter

- Using a geometric mean filter is given by the expression

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

Harmonic mean filter

- The harmonic mean filter operation is given by the expression

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

Contraharmonic mean filter

- The contraharmonic mean filter operation is given by the expression

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Where Q is called the order of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise.

Order-Statistics Filters

- Order-Statictics filters are spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter
  - Median filter
  - Max and Min filter
  - Midpoint filter
  - Alpha-trimmed mean filter

Median filter

- Process is replaces the value of a pixel by the median of the gray levels in region S_{xy} of that pixel:

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

Max and Min filter

- Using the 100 th percentile results in the so-called max filter, given by

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

This filter is useful for finding the brightest points in an image. Since pepper noise has very low values, it is reduced by this filter as a result of the max selection processing the subimage area S_{xy}.

- The 0^{th} percentile filter is min filter:

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

This filter is useful for finding the darkest points in an image. Also, it reduces salt noise as a result of the min operation.

Midpoint filter

- **The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter:**

$$f(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

Note: This filter works best for randomly distributed noise, like Gaussian or uniform noise.

Alpha-trimmed mean filter

- **Suppose that we delete the d/2 lowest and the d/2 highest gray-level values of g(s,t) in the area S_{xy}.**
- **Let $g_r(s,t)$ represent the remaining mn-d pixels. And averaging these remain pixels is denoted as:**

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

Where the value of d can range from 0 to mn-1. When d=0, It is arithmetic mean filter and d=(mn-1)/2 is a median filter. It is useful for the multiple types of noise such as the combination of salt-and-pepper and Gaussian noise.