Chapter: 3 (Image Enhancement & Processing)

DIP by Gonzalez 3rd.

Dept. of Computer Science & Engineering
International Islamic University Chittagong

Md. Khaliluzzaman

Image Enhancement (Spatial)

Image enhancement:

1. Improving the interpretability or perception of information in images for human viewers
2. Providing `better' input for other automated image processing techniques

Spatial domain methods:

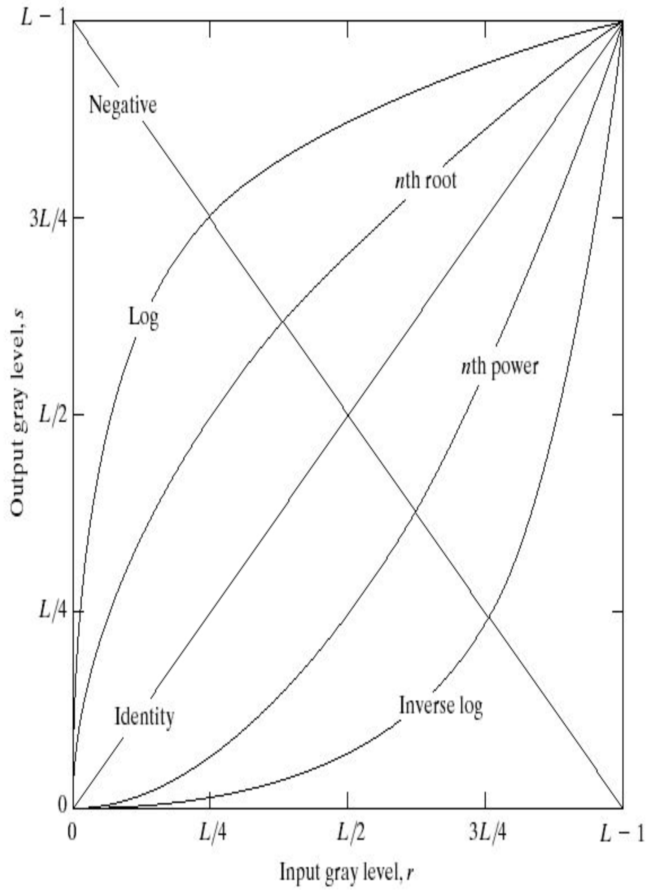
operate directly on pixels

Frequency domain methods:

operate on the Fourier transform of an image

Image Enhancement in the Spatial Domain

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



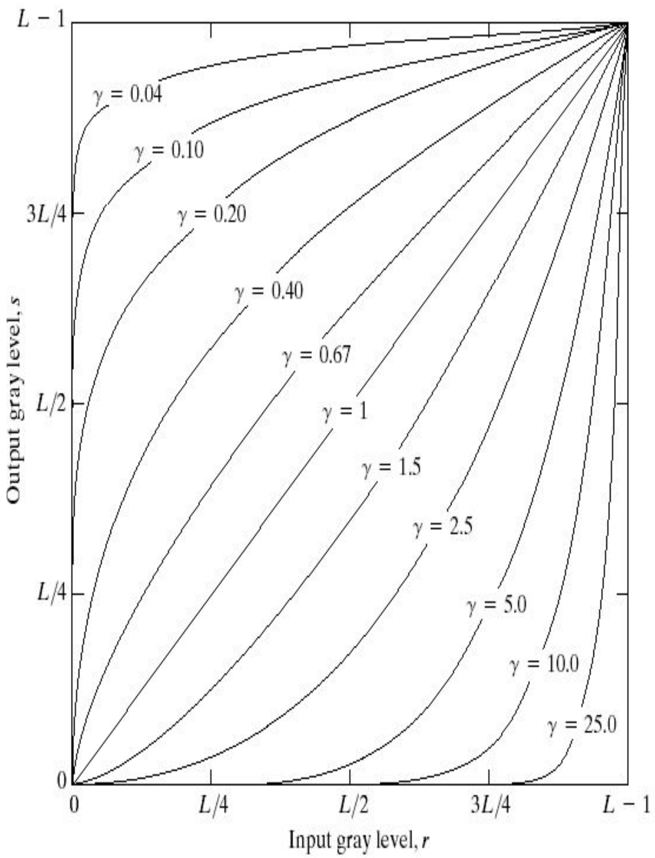


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Image Enhancement (Spatial)

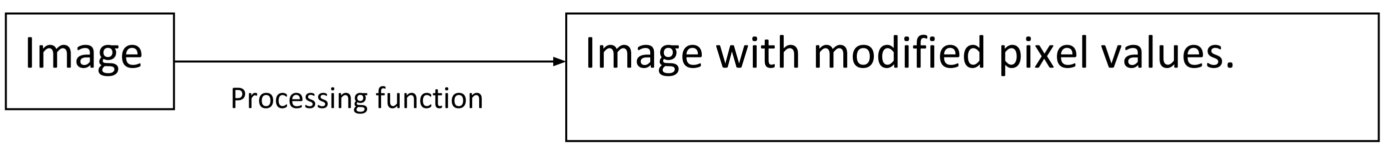
For processing two methods is used.

Spatial domain processing.

Frequency domain processing.

Spatial domain processing:

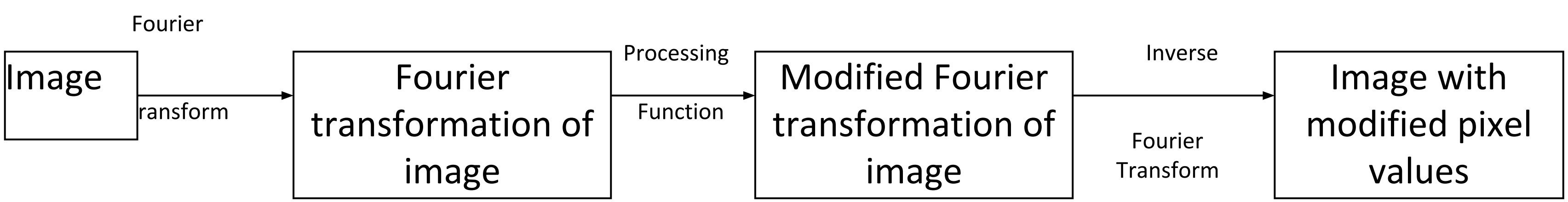
- Processing image directly is called spatial domain processing.



Frequency domain processing

Frequency domain processing:

- Don’t change it directly but transform.

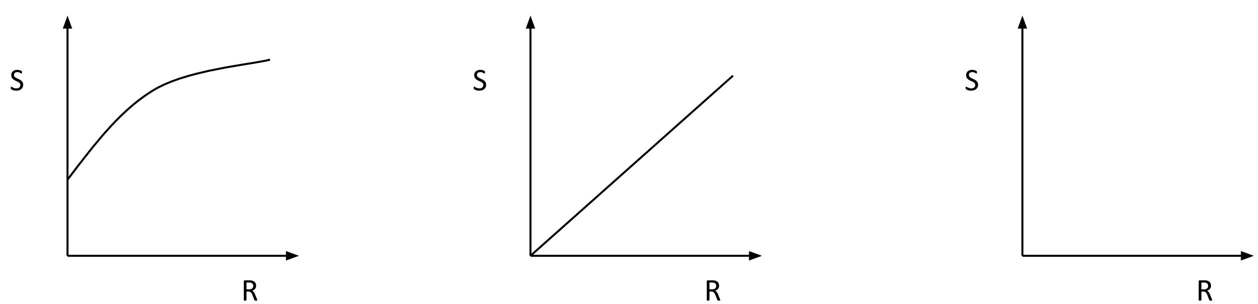


Spatial domain methods:

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

where, $g(x, y)$= Processed image (output image)
$f(x, y)$= Original image (input image)
T= Operation on f.

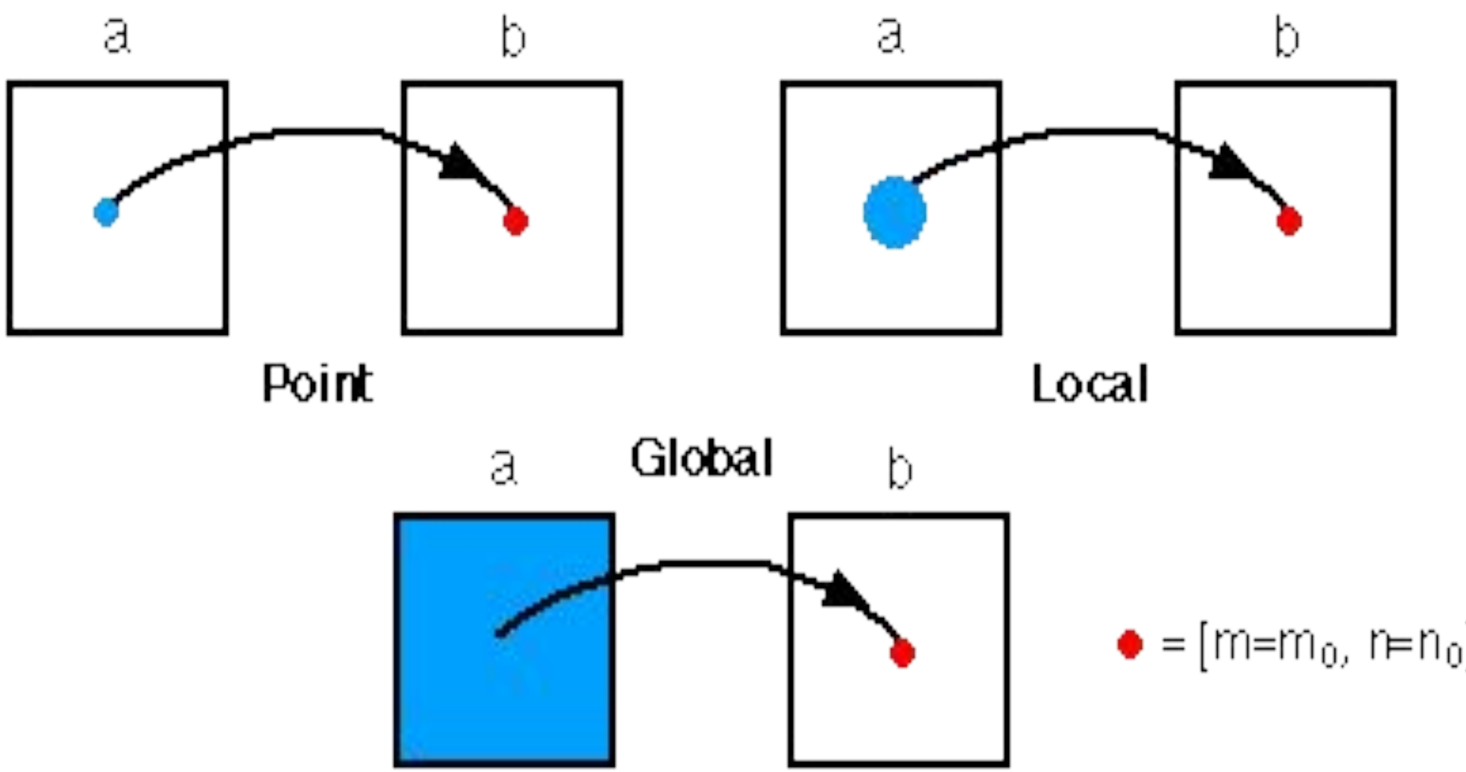
Examples:



Types of operations

Operation	**Characterization**
* *Point*	- the output value at a specific coordinate is dependent only on the input value at that same coordinate.
* *Local*	- the output value at a specific coordinate is dependent on the input values in the *neighborhood* of that same coordinate.
* *Global*	- the output value at a specific coordinate is dependent on all the values in the input image.

shown in graphically:



Types of processing

Another 5 types of processing:

- Point Processing.
- Spatial Filtering.
- Temporal Processing.
- Geometric Processing.
- Morphological Processing.

Some Basic Intensity Transformation Functions

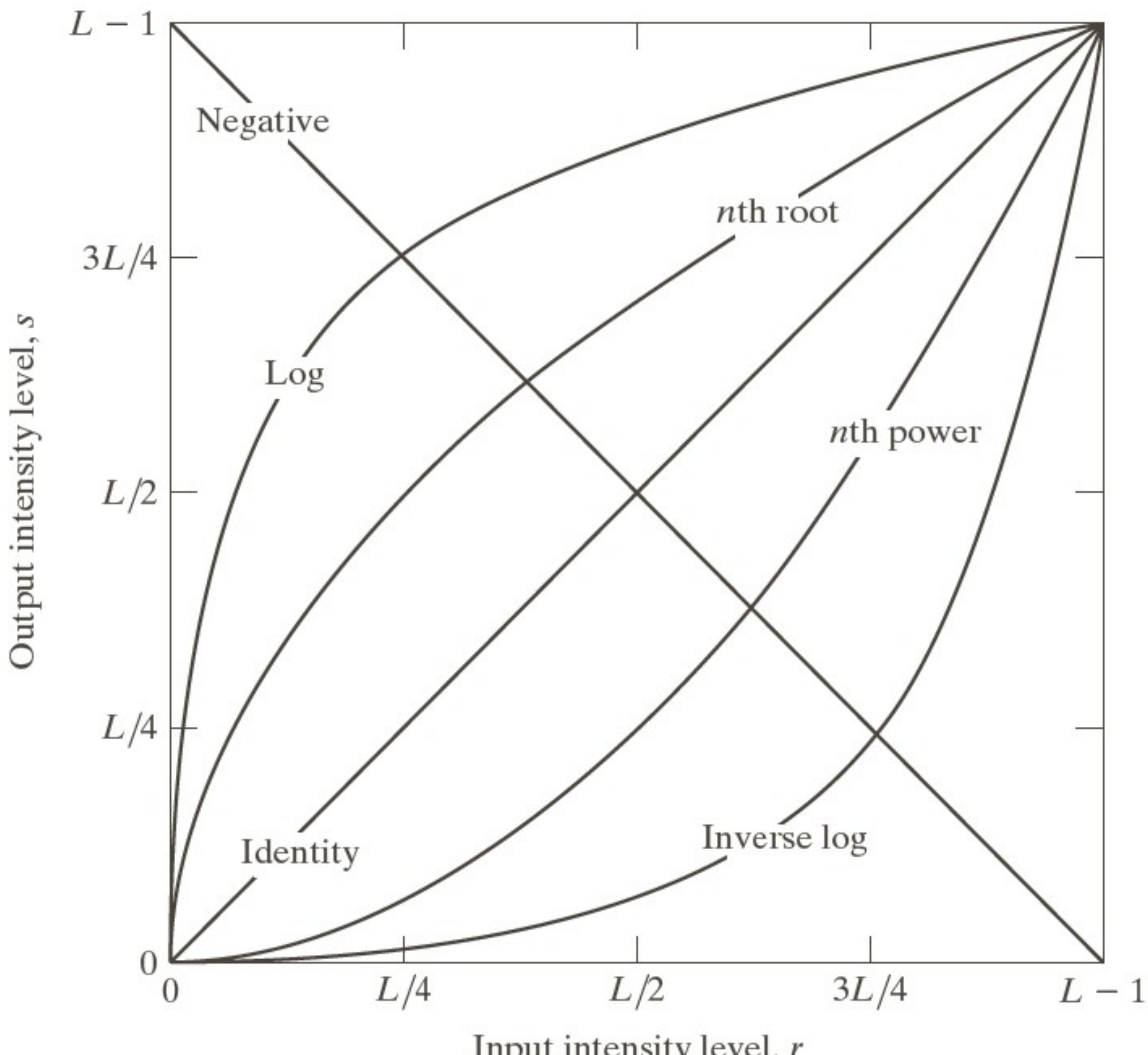


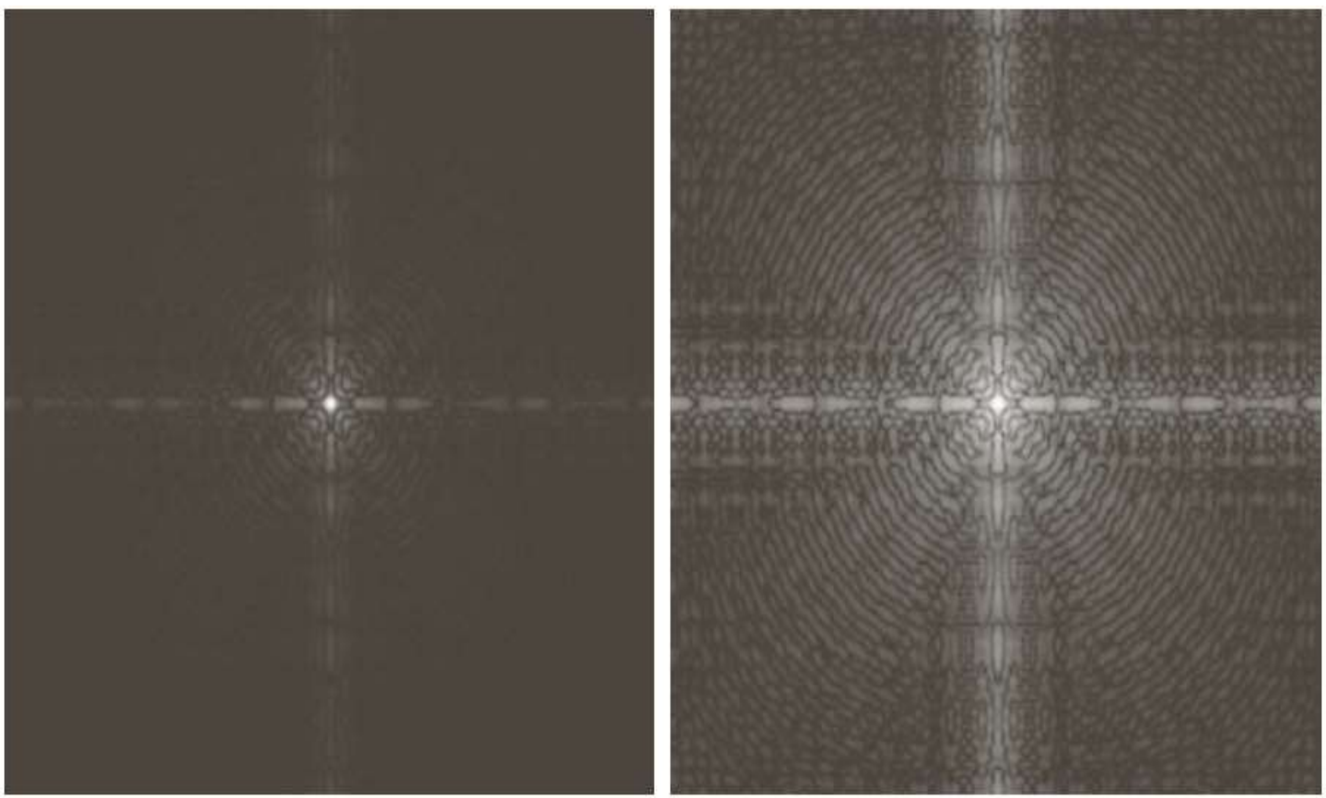
FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Some Basic Intensity Transformation Functions

- Log Transformations
  - $s = c \log(1 + r)$
    - c is constant
  - It maps a narrow range of low intensity values in the input into a wide range of output levels
  - The opposite is true of higher values of input levels
  - It expands the values of dark pixels in an image while compressing the higher level values
  - It compresses the dynamic range of images with large variations in pixel values

Some Basic Intensity Transformation Functions

- Log Transformations



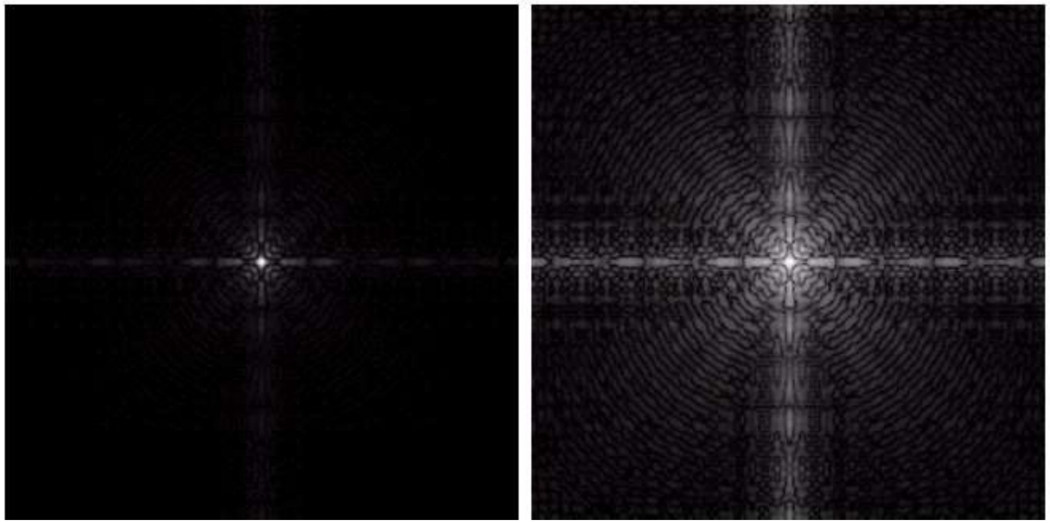
a b

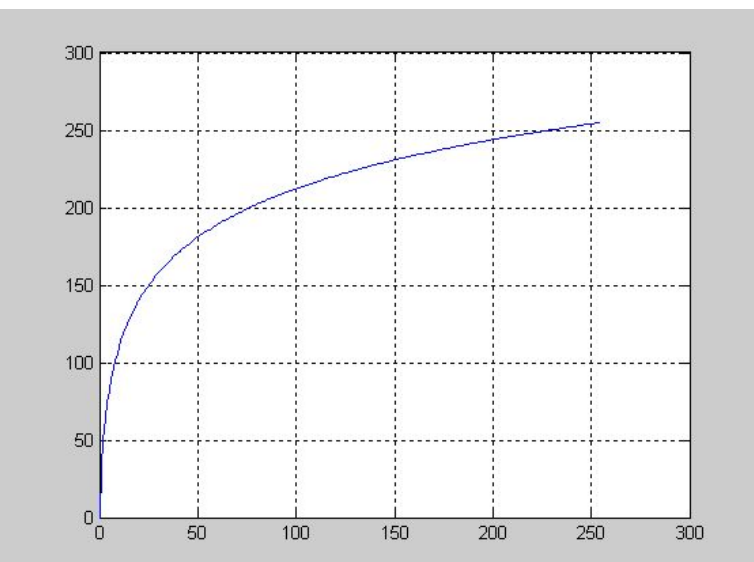
FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation in Eq. (3.2-2) with $c = 1$.

Log Transform

a b

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





$$T(r) = c \log(1 + r)$$

Some Basic Intensity Transformation Functions

- Power Law (Gamma) Transformations
  - $s = c\, r^{\gamma}$
    - c and γ are both positive constants
  - With fractional values($0<\gamma<1$) of gamma map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values ($\gamma > 1$)of input levels.
  - C=gamma=1 means it is an identity transformations.
  - Variety of devices used for image capture , printing, and display respond according to a power law.

Some Basic Intensity Transformation

- Power Law (Gamma) T

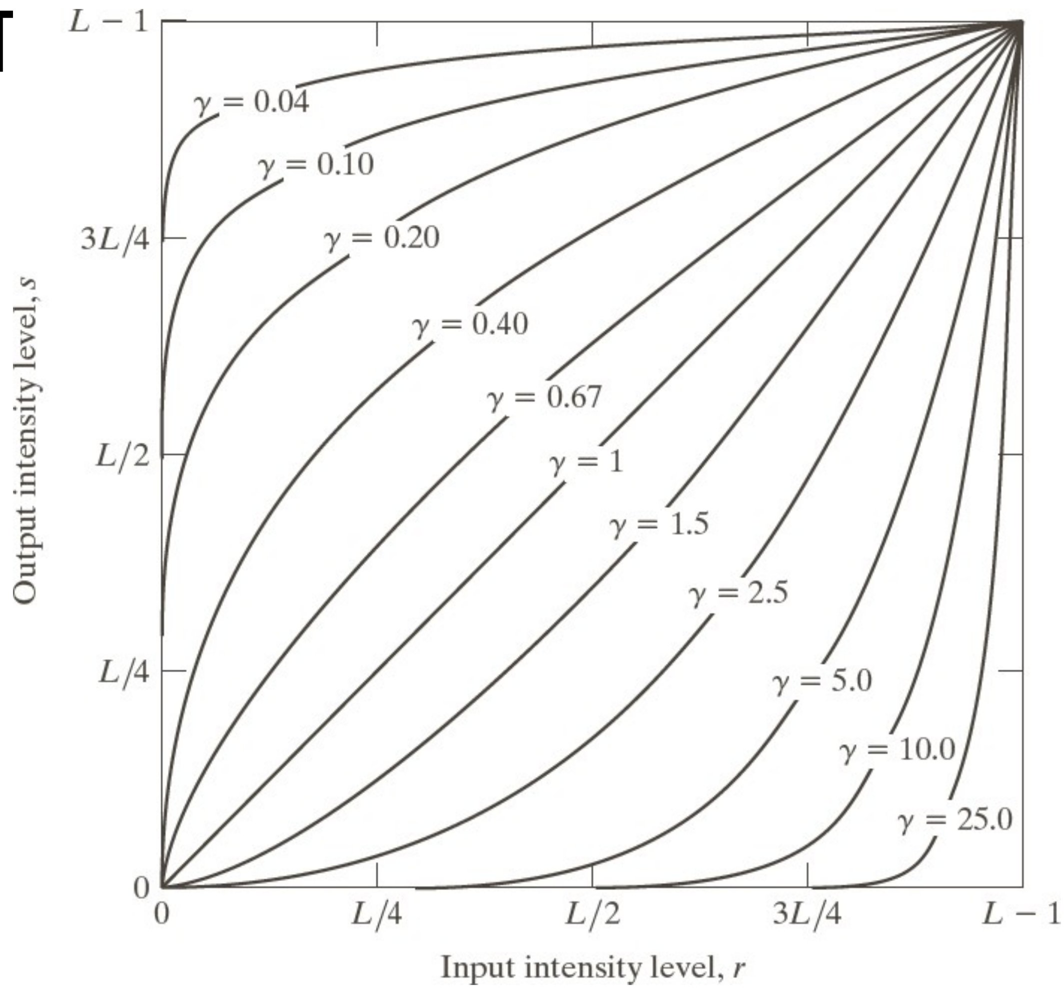
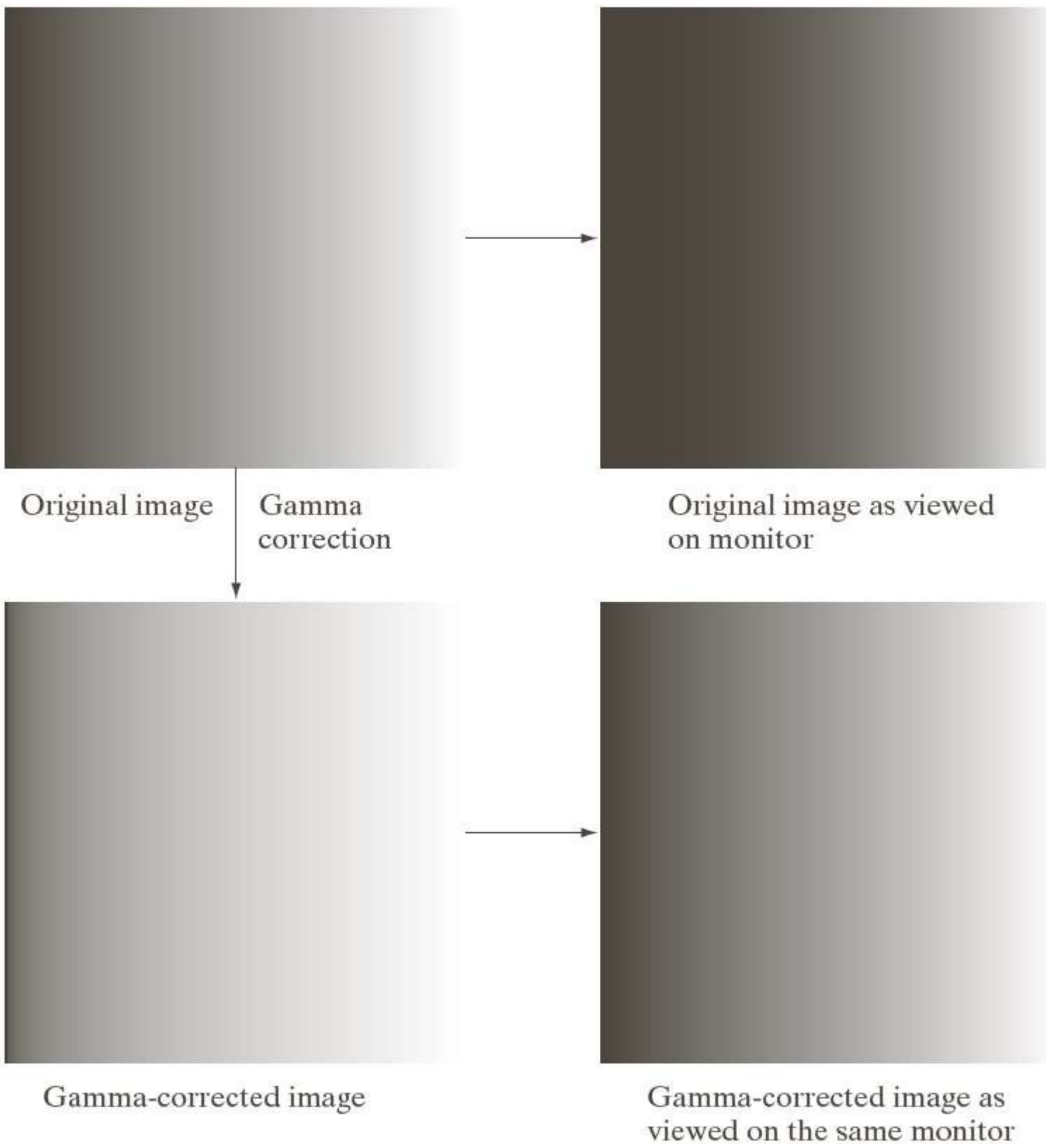


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Some Basic Intensity



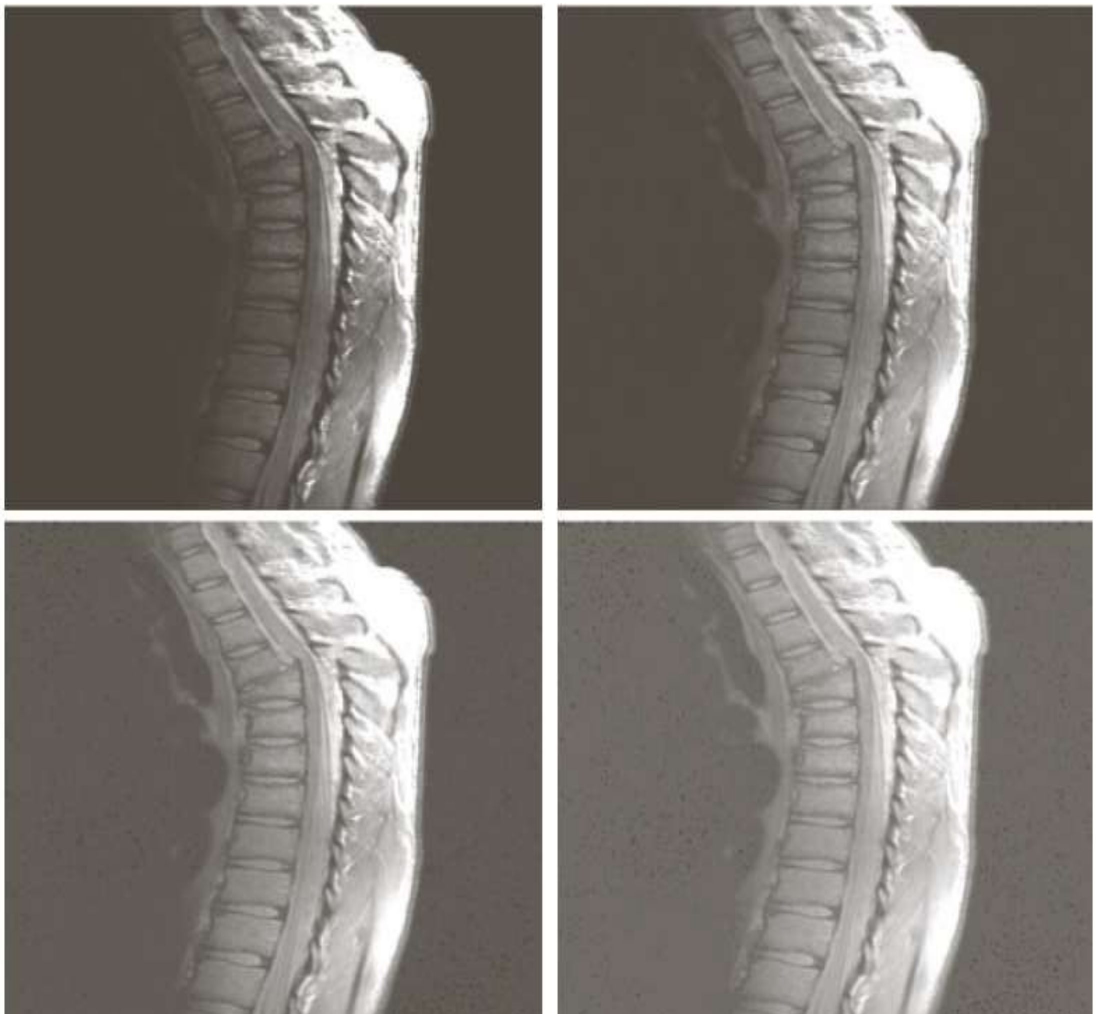
a b
c d

FIGURE 3.7
(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

Some Basic Intensity Transformation Functions

- Power Law (Gamma) Transformations
  - Images that are not corrected properly look either bleached out or too dark.
  - Varying gamma changes not only intensity, but also the ratio of red to green to blue in a color images.
  - Gamma correction has become increasingly important as the use of digital images over the internet.
  - Useful for general purpose contrast manipulation.
  - Apply gamma correction on CRT (Television, monitor), printers, scanners etc.
  - Gamma value depends on device.

Some Basic Intensity Transformation Functions



a b
c d

FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Some Basic Intensity Transformation Functions

a b
c d

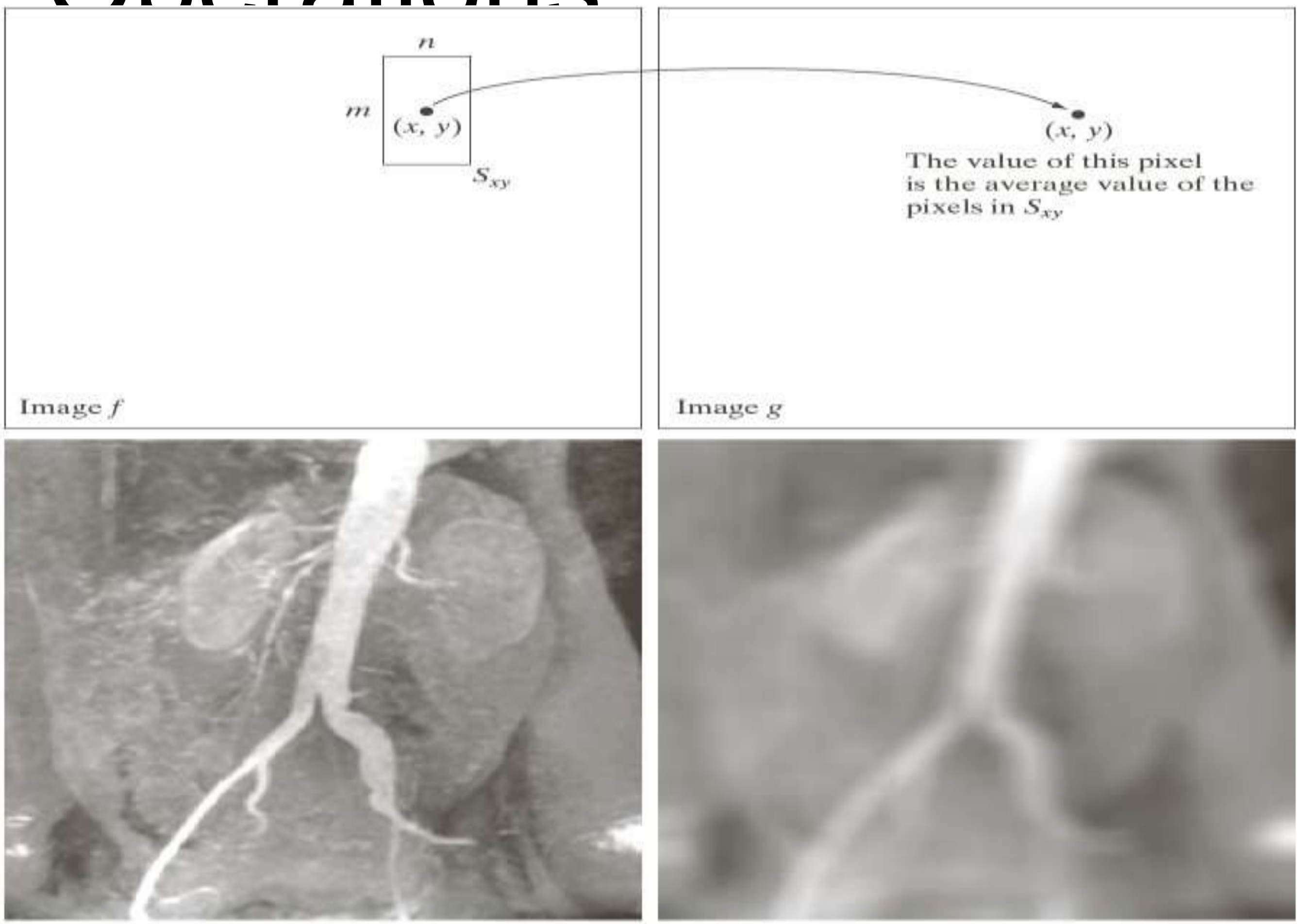
FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0$, 4.0, and 5.0, respectively. (Original image for this example courtesy of NASA.)

Spatial Operations

- Single-pixel operation (Intensity Transformation)
  - Negative Image, contrast stretching etc.
- Neighborhood operations
  - Averaging filter, median filtering etc.
- Geometric spatial transformations
  - Scaling, Rotation, Translations etc

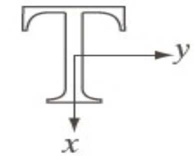
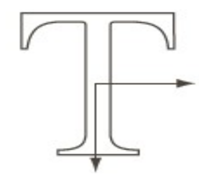
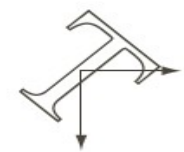
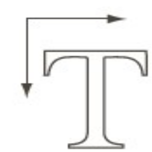
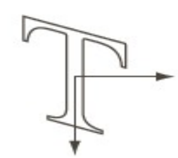
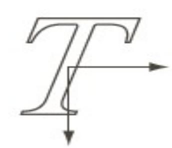
Single Pixel Operations

Neighborhood Operations

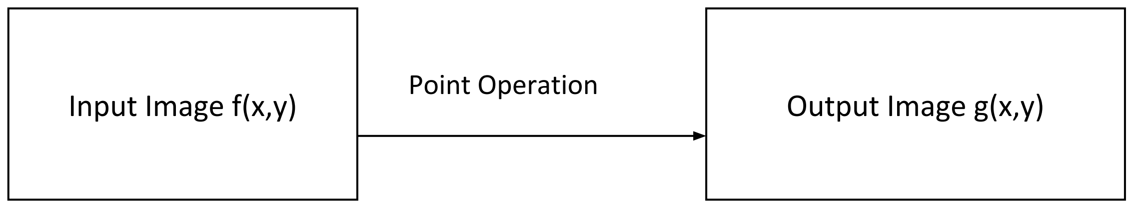


Geometric Spatial

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)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Point Processing



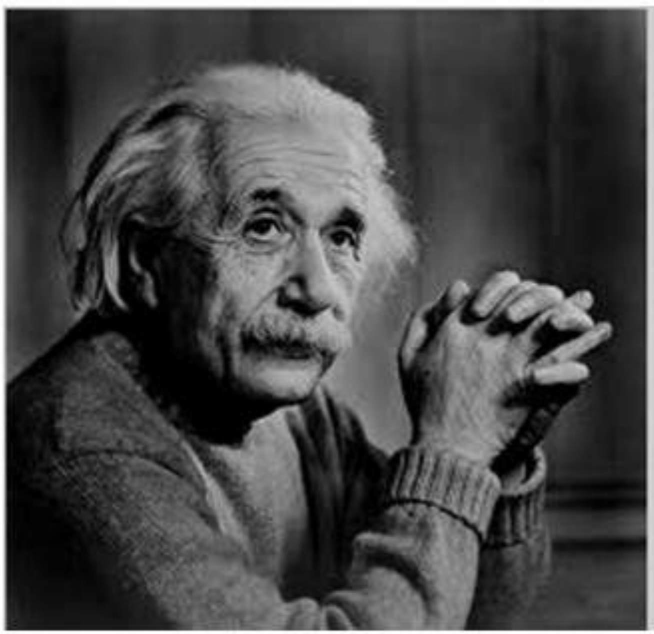
Monadic operation
Threshold
Contrast stretching
Gray level slicing
Image histogram
Histogram equalization
Other spatial functions

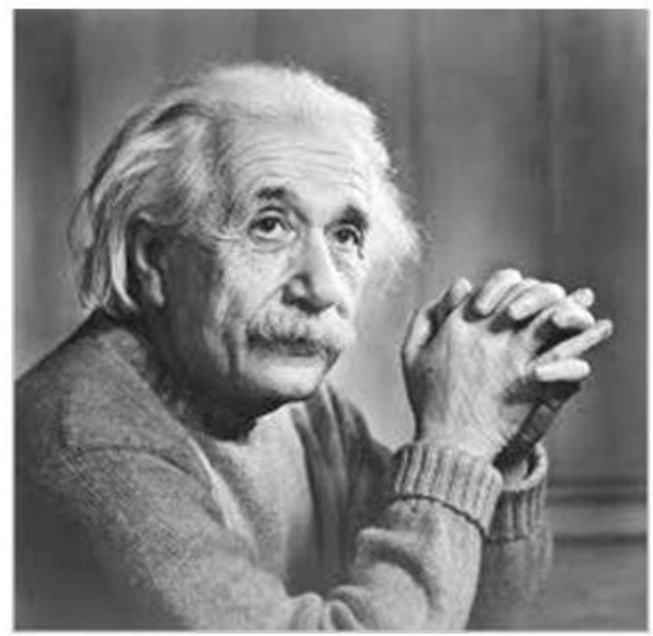
Monadic operation

Output image is obtained by applying arithmetic operation on input image.

-- Adding a constant: $g(x,y)= f(x,y) + k$ K=constant
-- Subtracting a constant: $g(x,y)= f(x,y) - k$ K=constant
-- Negative: $g(x,y) = k- f(x,y)$ [usually k= 2l-1]
-- Multiply by constant: $g(x,y) = kf(x,y)$
-- Divide by constant: $g(x,y) = f(x,y)/k$
-- Divide into constant: $g(x,y) = k/f(x,y)$
-- OR constant: g(x,y) = k or f(x,y)
-- AND constant: g(x,y) = k and f(x,y)
-- XOR constant: g(x,y) = k xor f(x,y)

Adding a constant (Increase Brightness)

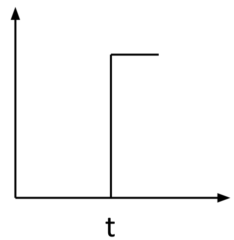




Negative

Threshold

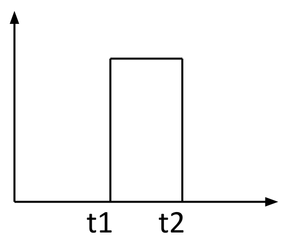
Operation to convert an image into two levels.t, t= threshold value
Single:



t= threshold value

g(x,y) = $\begin{cases} 0 \\ M \end{cases}$ [if f(x,y) <t and if f(x,y) >=t and usually M= 2l-1]

Dual Threshold: t2t1t1, t2 = threshold value

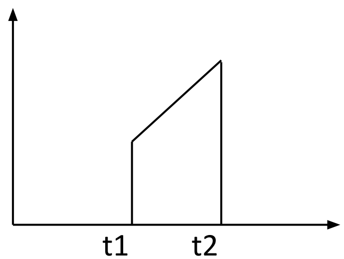


t1, t2 = threshold value

g(x,y) = $\begin{cases} 0 \\ M \\ 0 \end{cases}$ [if f(x,y) <t1 and if t1<=f(x,y) <=t2 and if f(x,y) > t2 usually M= 2l-1

Threshold (Cont.)

Gray scale threshold:t2t1t1, t2 = threshold value



t1, t2 = threshold value

g(x,y) = $\begin{cases} 0 \\ f(x,y) \\ 0 \end{cases}$ [if f(x,y) <t1 and if t1<=f(x,y) <=t2 and if f(x,y) > t2 usually M= 2l-1

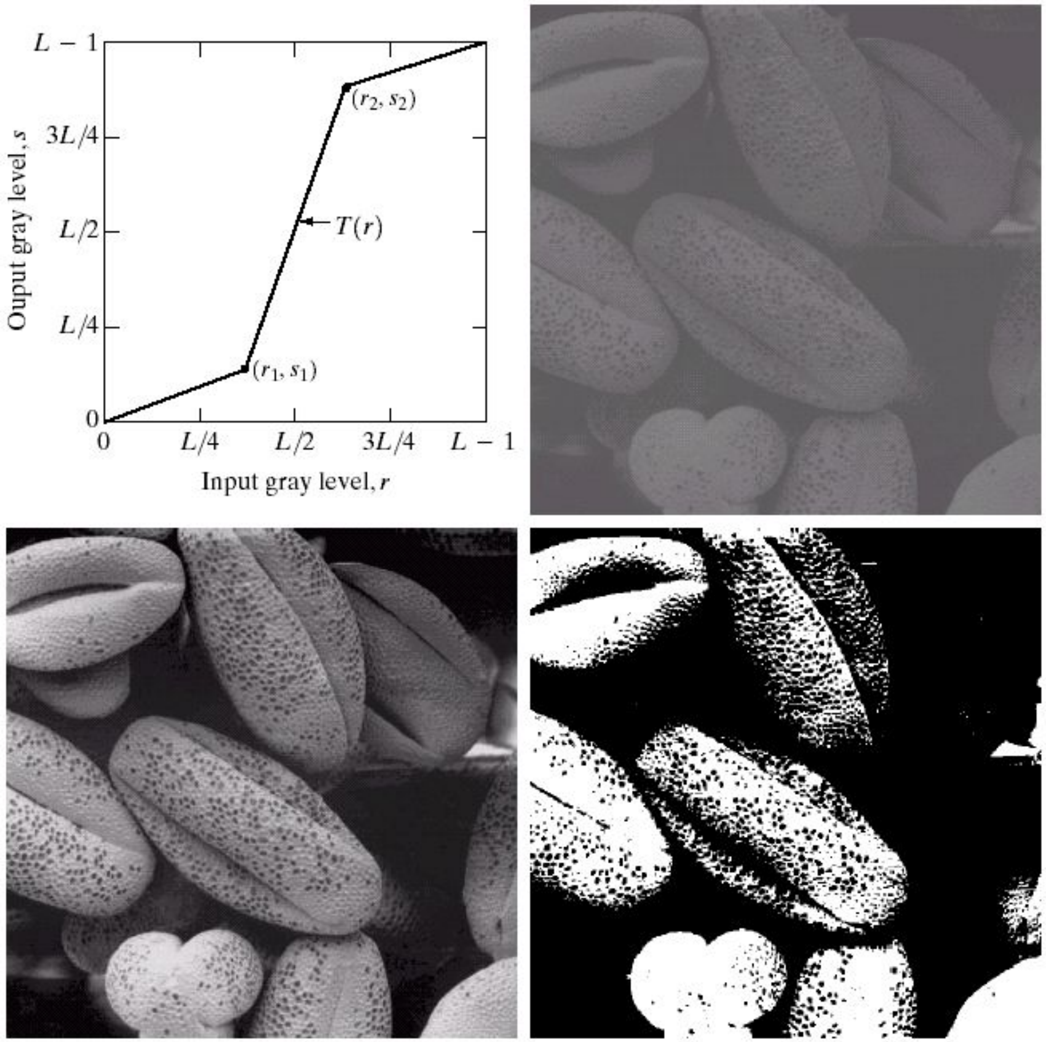
Contrast stretching

- Aim to increase the dynamic range of an image. It transforms the gray levels in the range {0,1,.....,L-1} by a piecewise linear function.
- The figure shows a typical example of contrast stretching.
- The locations of points (r1,s1) and (r2,s2) control the shape of the function.

For example the following piecewise linear function

$$s = \begin{cases} (227 * r - 5040)/47 & \text{if } 28 \le r \le 75 \\ r & \text{otherwise} \end{cases}$$

Contrast stretching



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.)

Contrast stretching (cont.)

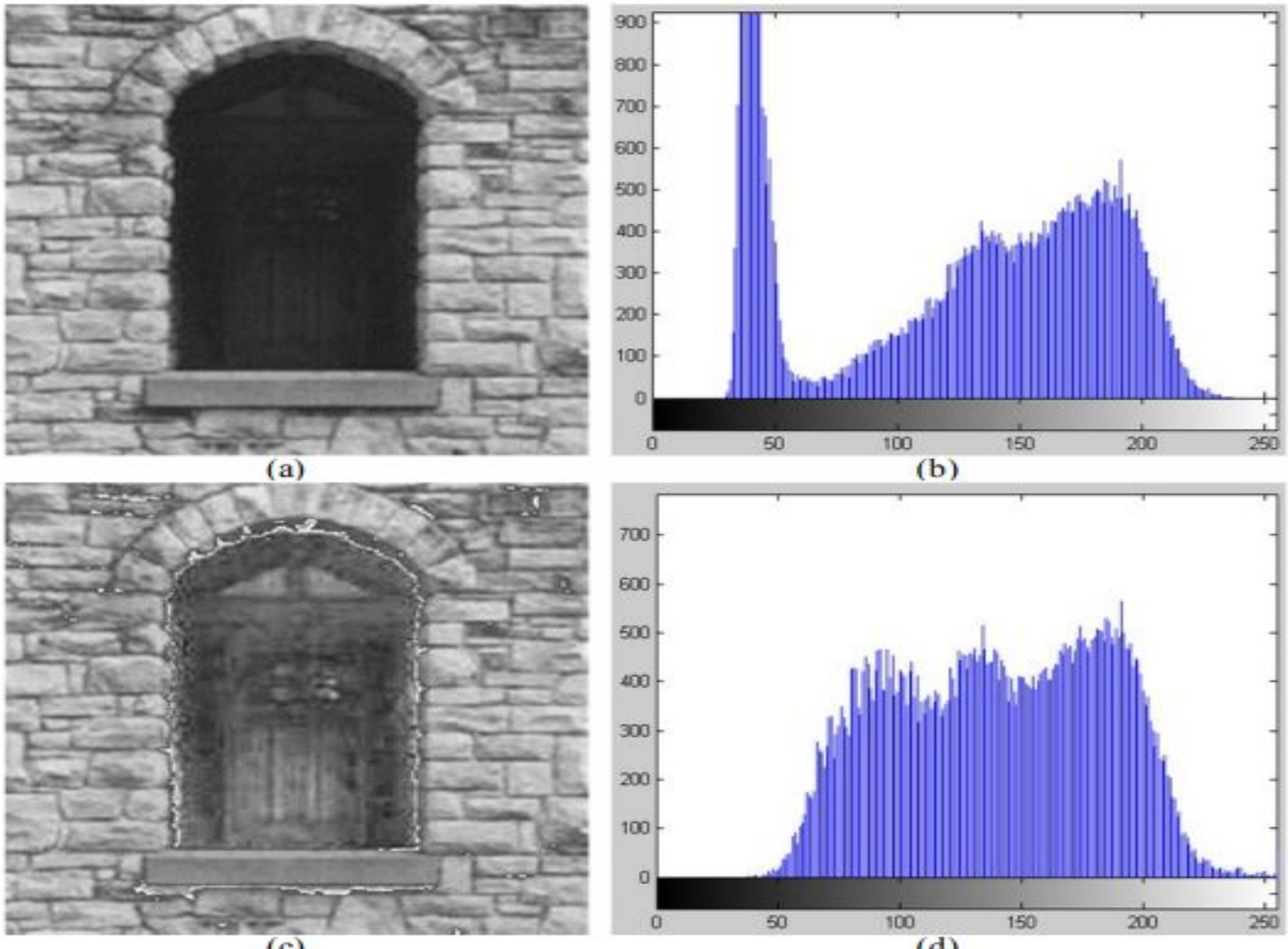


Figure Contrast stretching. (a) Original image. (b) Histogram of (a). (c) Result of contrast stretching. (d) Histogram of (c).

Contrast stretching (cont.)

Another form of contrast stretching is called automatic (full) contrast stretching as shown in the example below:

$$s = \begin{cases} 0 & if\ r < 90 \\ (255 * r - 22950)/48 & if\ 90 \le r \le 138 \\ 255 & if\ r > 138 \end{cases}$$

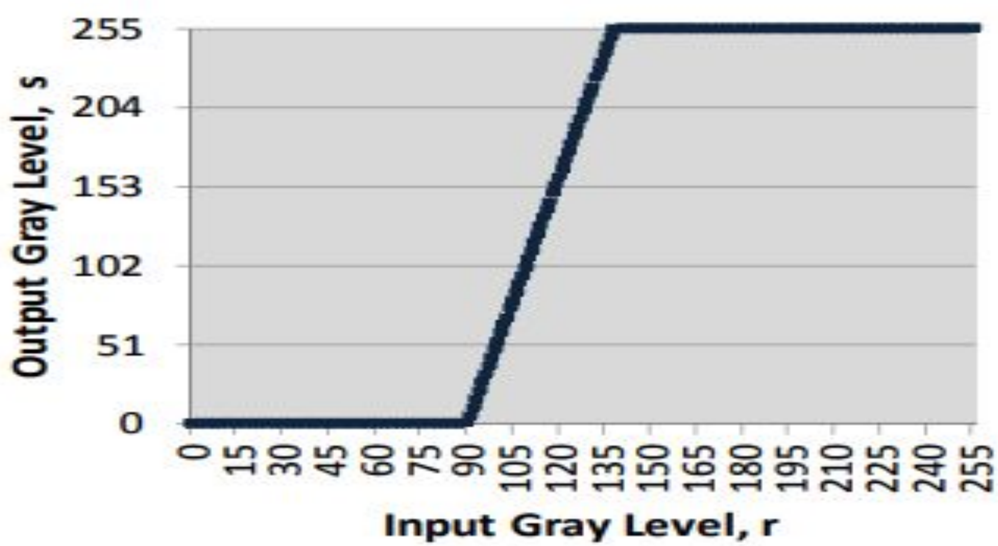


Figure 5.4 Full contrast-stretching

Contrast stretching (cont.)

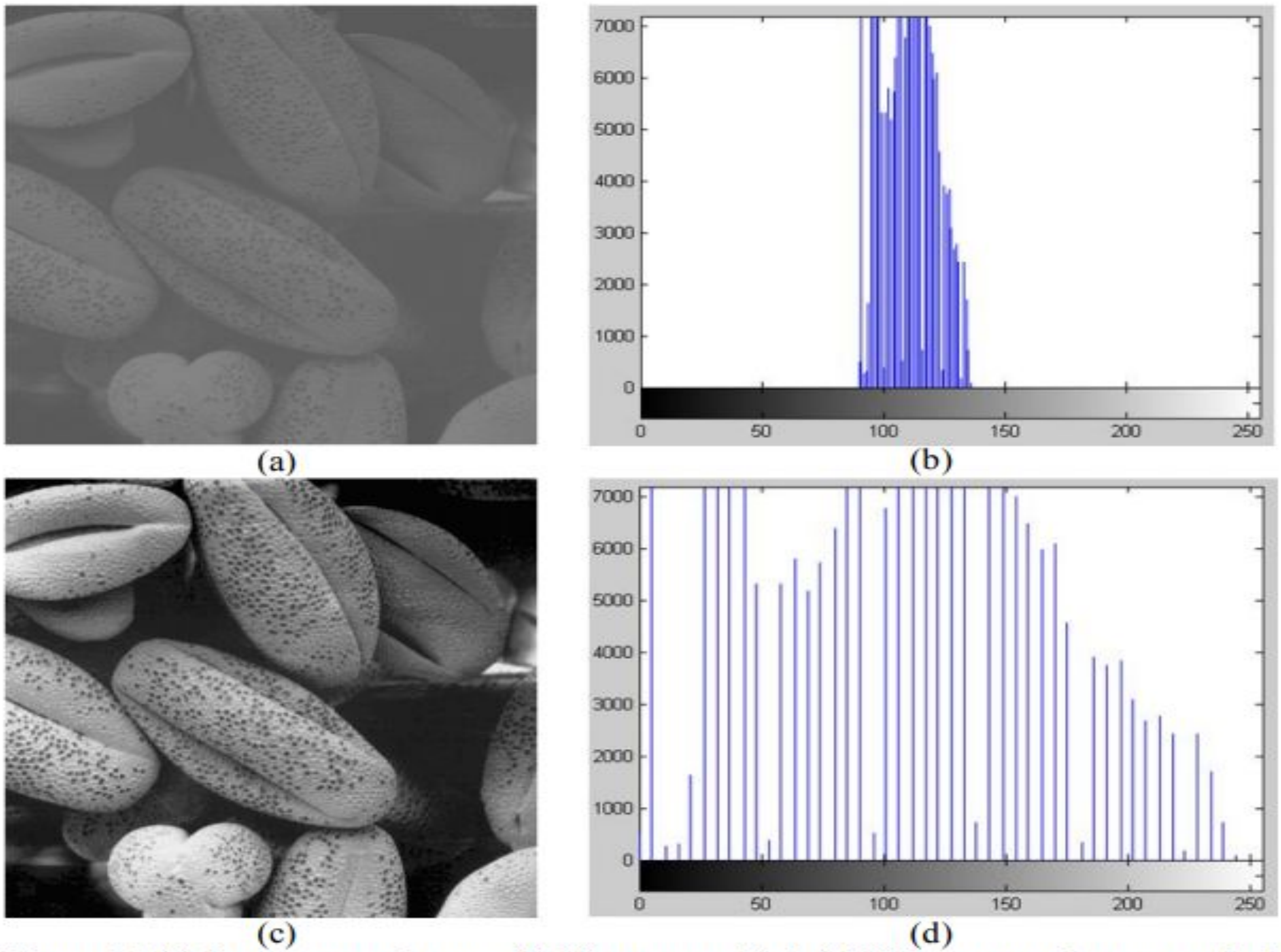


Figure 5.5 (a) Low-contrast image. (b) Histogram of (a). (c) High-contrast image resulted from applying full contrast-stretching in Figure 5.4 on (a). (d) Histogram of (c)

Piecewise-Linear Transformation Functions

- Intensity Level Slicing
  - Highlighting specific range of intensities in an image.
  - Enhances features such as masses of water in satellite imagery and enhancing flaws in X-ray images.
  - It can be Implemented two ways:
  - 1) To display only one value (say, white) in the range of interest and rests are black which produces binary image.
  - 2) brightens (or darkens) the desired range of intensities but leaves all other intensity levels in

Gray level slicing

Highlighting a specific range of gray levels in an image.
Two approaches:t1, t2 = threshold value AB

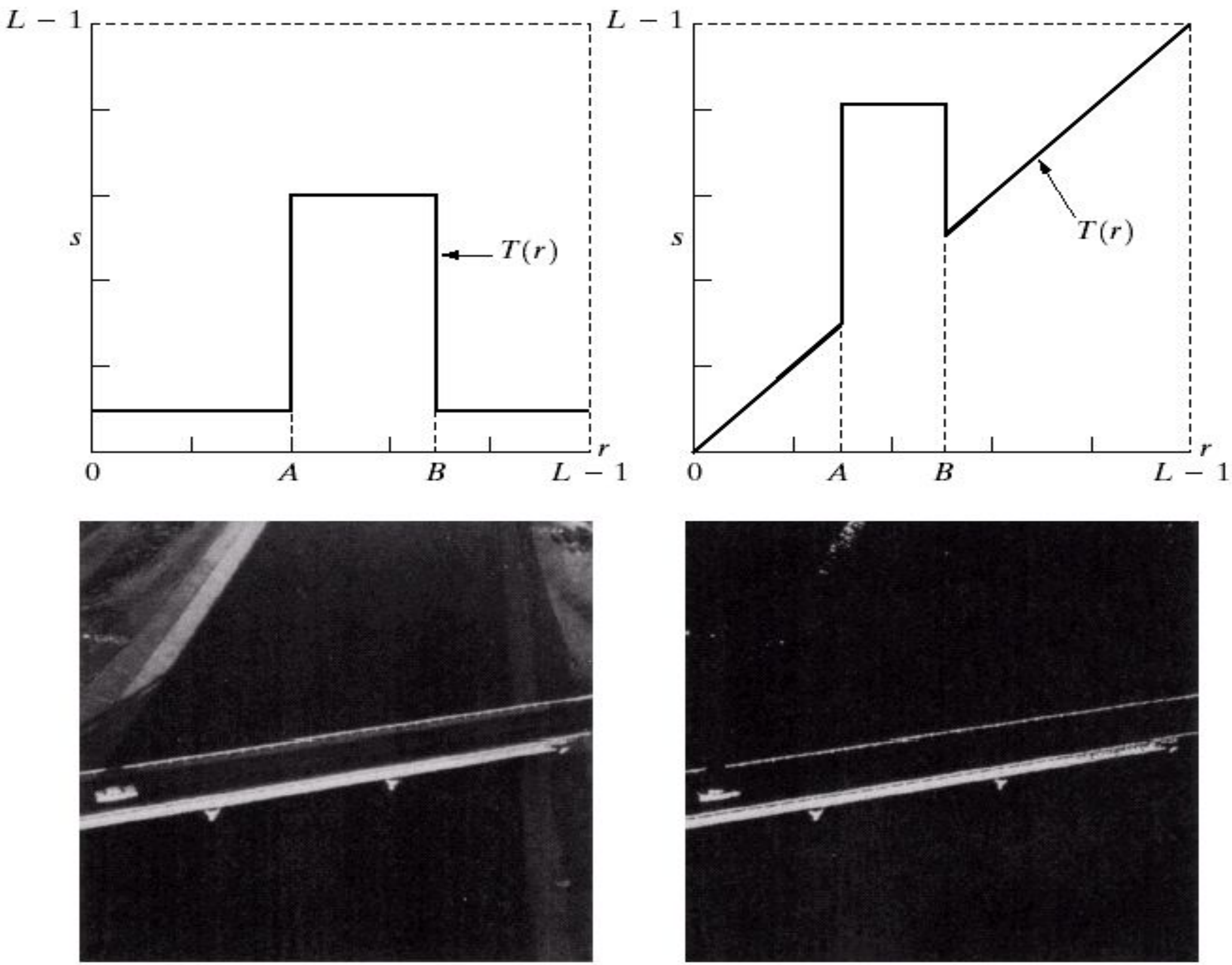
g(x,y) = $\begin{cases} 0 \\ p \\ 0 \end{cases}$

[if f(x,y) <A and if A<=f(x,y) <=B and if f(x,y) > B]
A, B = threshold value

g(x,y) = $\begin{cases} f(x,y) \\ p \\ f(x,y) \end{cases}$

[if f(x,y) <A and if A<=f(x,y) <=B and if f(x,y) > B]

Gray level slicing



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. (c) An image. (d) Result of using the transformation in (a).

Piecewise-Linear Transformation Functions

- Intensity Level

a b c

FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

Histogram Processing

- Histogram of a digital image with intensity levels in the range [0,L-1] is a discrete function $h(r_k) = n_k$, where r_k is the kth intensity value and n_k is the number of pixels in the image with intensity r_k
- Normalized histogram $p(r_k)=n_k/MN$, for $k = 0,1,2.......$ L-1.
- Histogram manipulation can be used for image enhancement.
- Information inherent in histogram also is quite useful in other image processing applications, such as image compression and segmentation.

Histogram processing

$$p(r_k) = \frac{n_k}{n}$$

Where, r_k =kth gray level.
n_k= number of gray level r_k.
n= total number of pixels in the image.

Example:

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

$r_k = 0$
$n_k = 1$
$n = 16$
$p(r_0)=1/16$

$r_k = 1$
$n_k = 4$
$n = 16$
$p(r_1)=4/16$

$r_k = 2$
$n_k = 3$
$n = 16$
$p(r_2)=3/16$

$r_k = 3$
$n_k = 2$
$n = 16$
$p(r_3)=2/16$

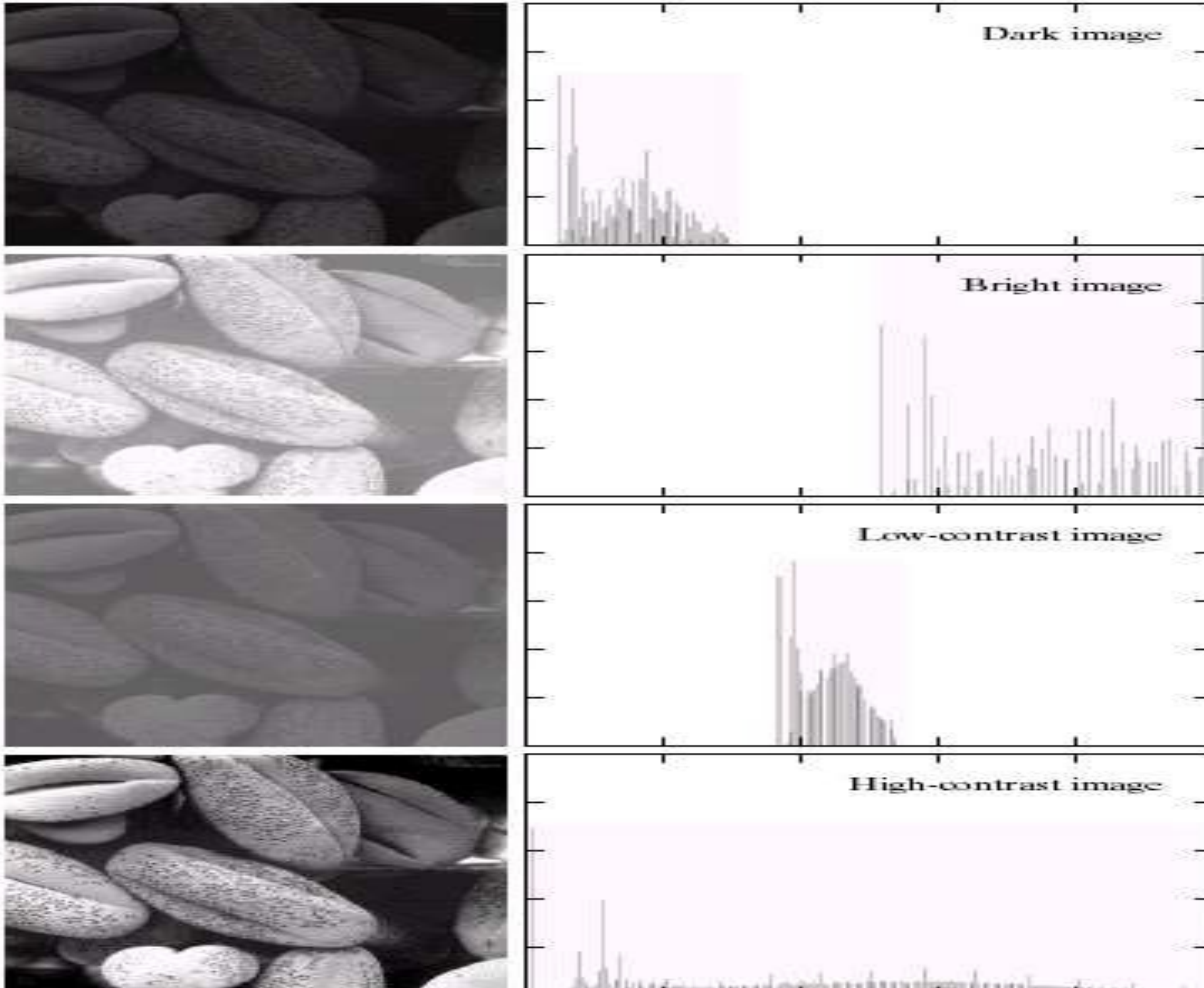
$r_k = 4$
$n_k = 2$
$n = 16$
$p(r_4)=2/16$

$r_k = 5$
$n_k = 1$
$n = 16$
$p(r_5)=1/16$

$r_k = 6$
$n_k = 1$
$n = 16$
$p(r_6)=1/16$

$r_k = 7$
$n_k = 2$
$n = 16$
$p(r_7)=2/16$

Histogram



Dark

Light

Low
Contrast

Hight
Contrast

Histogram Equalization

- Intensity mapping form

$$s = T(r),\ 0 \le r \le L-1$$

Conditions:

a) T(r) is a monotonically increasing function in the interval [0, L-1] and

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

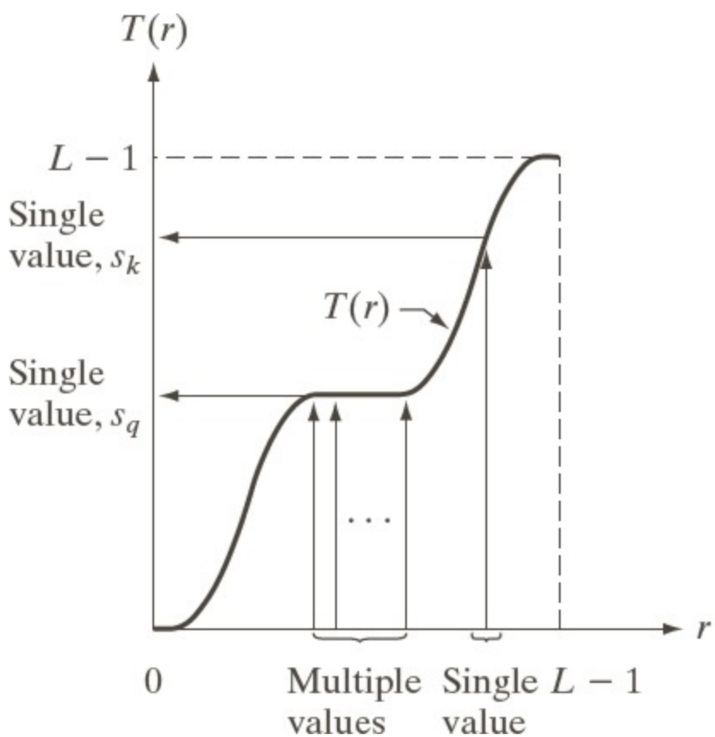
In some formulations, we use the inverse in which case

$$r = T^{-1}(s), \quad 0 \le s \le 1$$

(a) change to

a') T(r) is a strictly monotonically increasing function in the interval [0, L-1]

Histogram Processing



a b

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.

Histogram equalization/ linearization

Convert image from uneven histogram to the even histogram is called histogram equalization.

$$N(g) = \max\left\{0, round\left(\frac{2^l.c(g)}{n}\right) - 1\right\}$$

Where, N(g) =new gray level of the pixel.

C(g) = cumulative pixel count upon old gray level g.

Round = rounding operation to nearest integer value.

2^l = number of gray levels.

n= total number of pixels.

Histogram equalization/ linearization

$$N(g) = \max\left\{0, round\left(\frac{2^l.c(g)}{n}\right) - 1\right\}$$

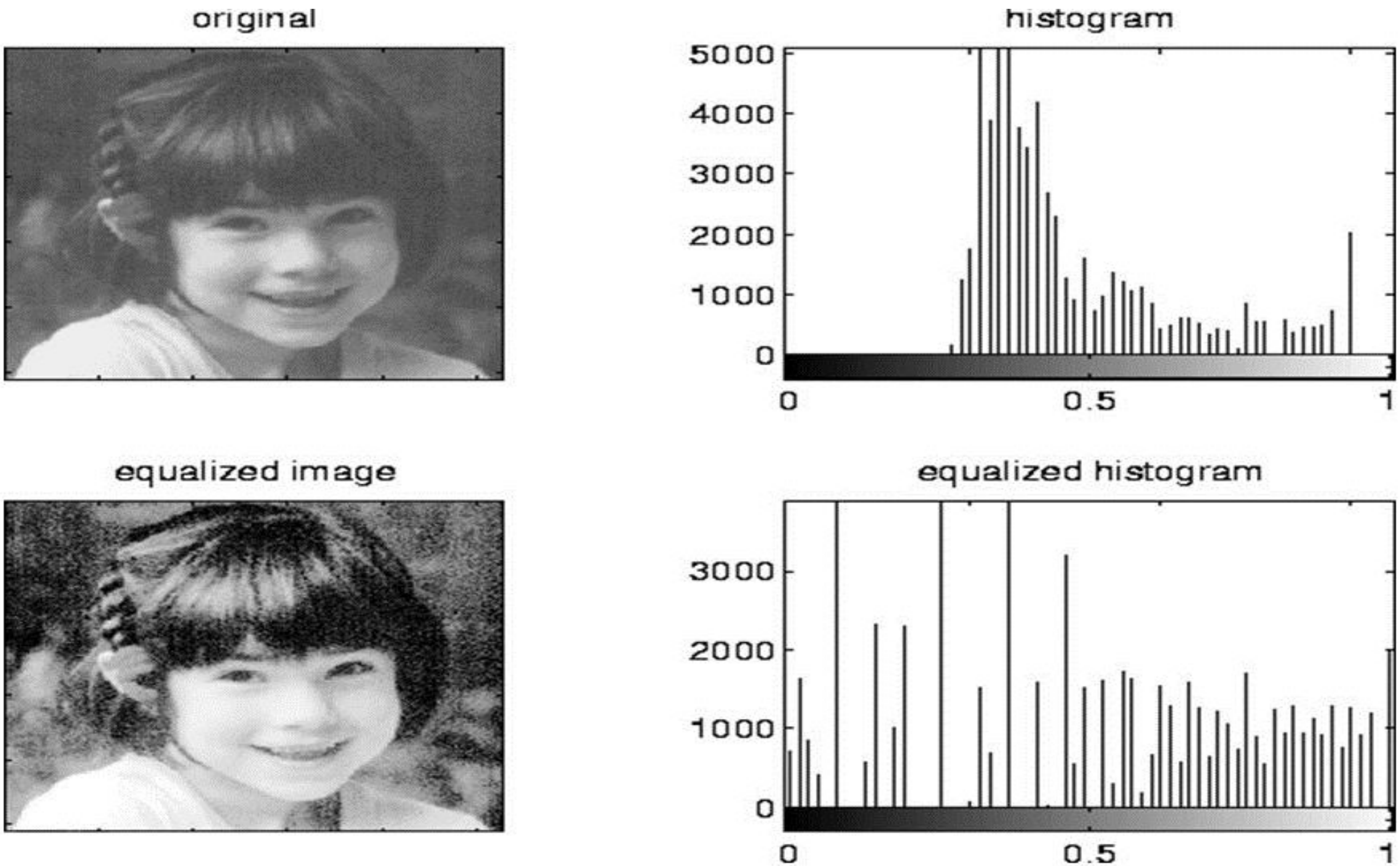
Example:

n= ∑f=64

$2^l = 7$

g	f	C(g)	N(g)
0	8	8	0
1	22	30	2
2	20	50	5
3	2	52	5
4	2	54	5
5	8	62	6
6	2	64	7
7	0	64	7

Histogram equalization/linearization



The original image and its histogram, and the equalized versions. Both images are quantized to 64 grey levels.

Histogram Equalization

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64 × 64 digital image.

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k}\frac{n_j}{MN}, \quad k = 0,1,2,\dots,L-1$$

$S_0 = T(r_0) = 7 * 790/4096 = 1.33 \to 1$

$S_1 = T(r_1) = 7 * (790+1023)/4096 = 3.08 \to 3$

$S_3 = 4.55 \to 4$

$S = 6.23 \to 6$

$S_5 = 7.00 \to 7$

$S_6 = 7.00 \to 7$

$S_7 = 7.00 \to 7$

Histogram Equalization

$S_0 = T(r_0) = 7 * 790/4096 = 1.33 \rightarrow 1$

$S_1 = T(r_1) = 7 * (790+1023)/4096 = 3.08 \rightarrow 3$

$S_2 = 4.55 \rightarrow 5$

$S_3 = 5.67 \rightarrow 6$

$S_4 = 6.23 \rightarrow 6$

$S_5 = 6.65 \rightarrow 7$

$S_6 = 6.86 \rightarrow 7$

$S_7 = 7.00 \rightarrow 7$

For Hist. Eq. in Fig. 3.19(c)

r_0 mapped to S_0=1 with value= 790/4096=0.19

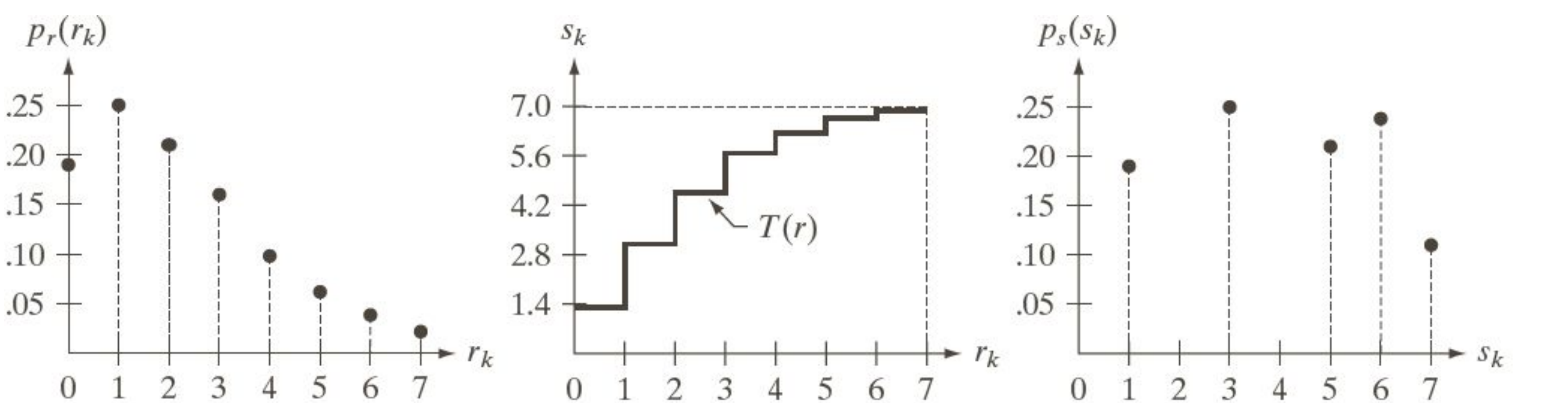
r_1 mapped to S_1=3 with value = 1023/4096=0.24

r_2 mapped to S_2=5 with value = 850/4096=0.21

r_3 ,r_4 mapped to 6 with value = (656+329)/4096

r_5 ,r_6, r_7 mapped to 7 with value

= (245+122+81)/4096 = 0.10



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Histogram

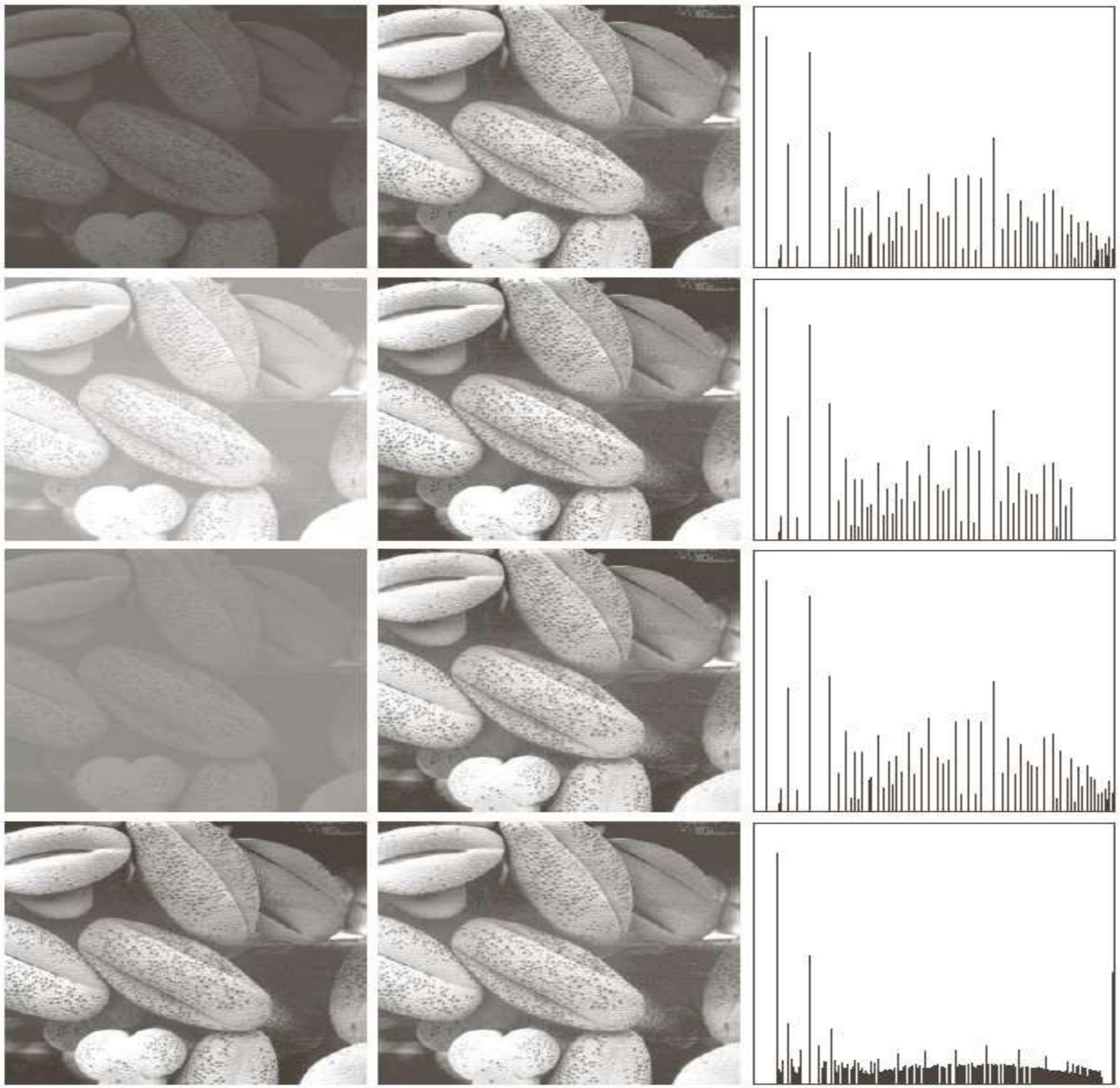


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Histogram Equalization

- Transformation functions

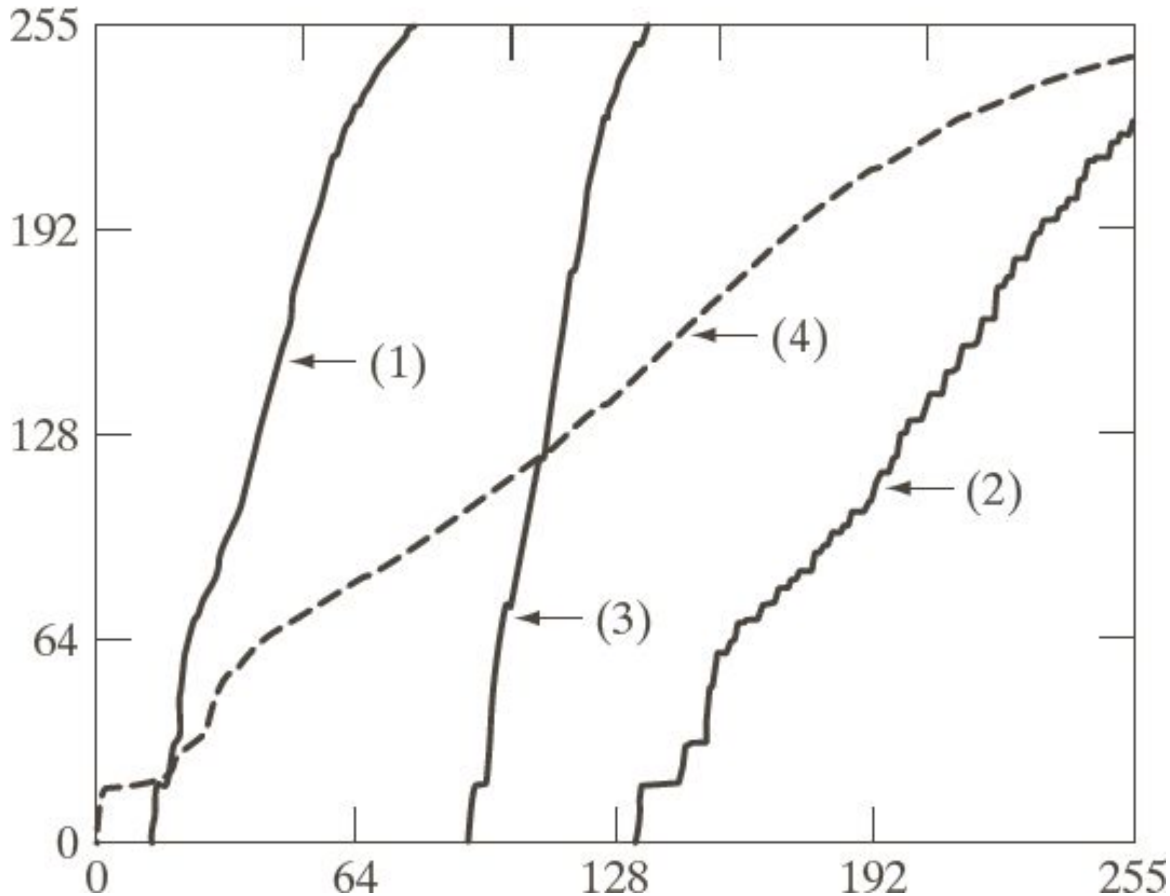


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

MathLab information

HISTOGRAM EQUALIZATION

```
>> f=imread(fig3.15(a).jpg');          %load in figure 3.15(a)
>> imshow(f)                           % show figure in a new window
>> figure, imhist(f)                   % show histogram in a new window
                                       % histogram is not normalized.
>> ylim('auto')                        % set histogram tick marks and axis limits automatically
>> g=histeq(f,256);                    % histogram equalize this figure
                                       % you can also do this with the cumsum function
>> hnorm=imhist(f)./numel(f);          % computes normalized histogram
>> figure, imshow(g)                   % show this figure in a new window
>> figure, imhist(g)                   % generate a histogram of the equalized image
>> ylim('auto')                        % set limits again
```

SEE GWE, Section 3.3.3 for a discussion of histogram specification using MATLAB

Histogram Specification

Here we use a specified histogram to process the image.

The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

Histogram Matching (Specification)

- Histogram equalization automatically determines a transformation function produce uniform histogram
- When automatic enhancement is desired, equalization is a good approach
- There are some applications in which attempting to base enhancement on a uniform histogram is not the best approach
- In particular, it is useful sometimes to be able to specify the shape of the histogram that we wish the processed image to have.
- The method used to generate a processed image that has a specified histogram is called *histogram matching* or *specification*

Histogram Matching (Specification)

- Histogram Specification Procedure:

1) Compute the histogram $p_r(r)$ of the given image, and use it to find the histogram equalization transformation in equation

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} \frac{n_j}{MN}, \quad k = 0,1,2,\ldots,L-1$$

and round the resulting values to the integer range [0, L-1]

2) Compute all values of the transformation function G using the equation

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(r_i), \quad q = 0,1,2,\ldots,L-1$$

and round values of G

3) For every value of s_k, k = 0,1,…,L-1, use the stored values of G to find the corresponding value of z_q so that $G(z_q)$ is closet to s_k and store these mappings from s to z.

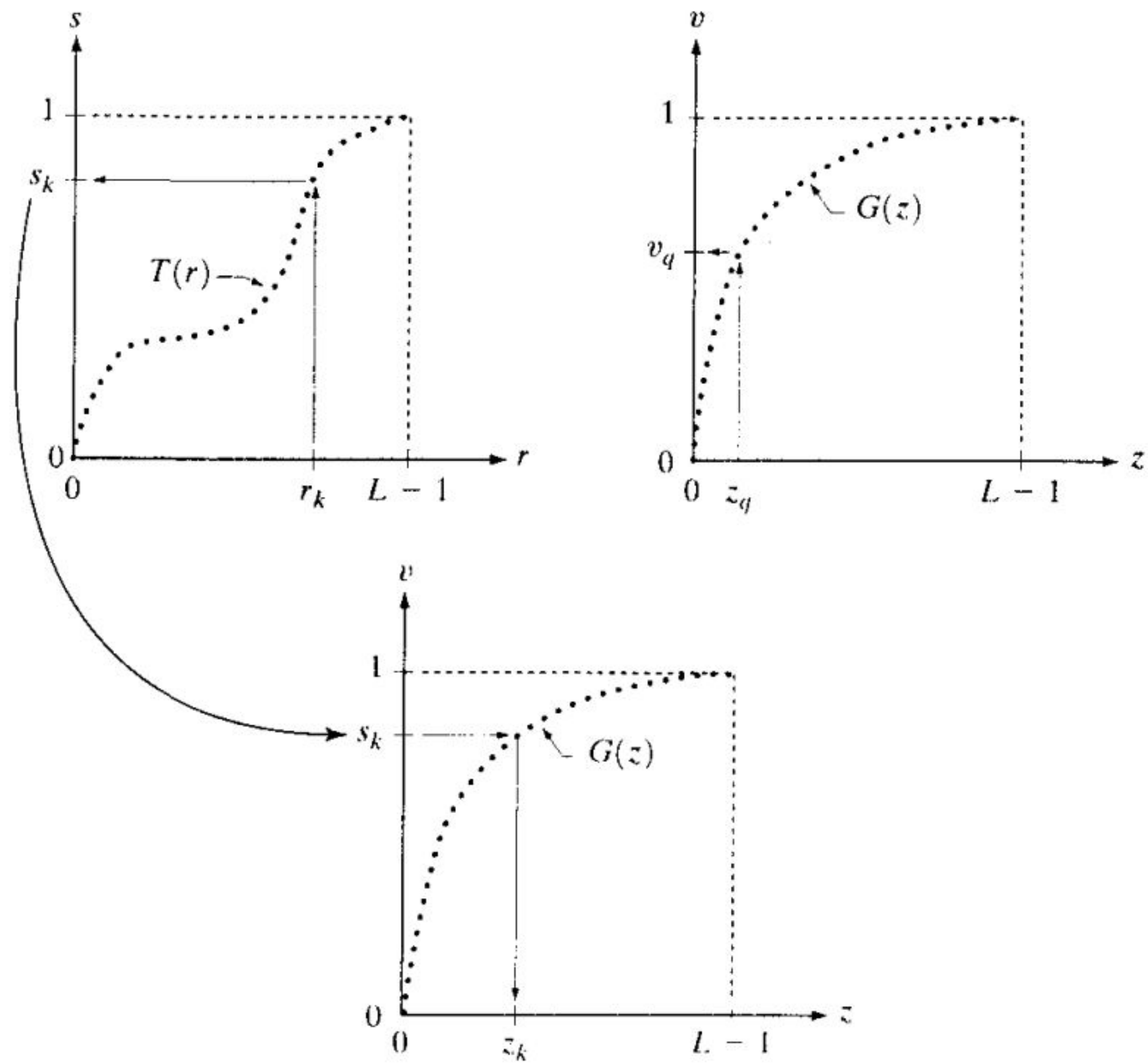
Histogram Matching (Specification)

- Histogram Specification Procedure:

4) Form the histogram-specified image by first histogram- equalizing the input image and then mapping every equalized pixel value, s_k , of this image to the corresponding value z_q in the histogram-specified image using the mappings found in step 3.

a b
c

FIGURE 3.19
(a) Graphical interpretation of mapping from r_k to s_k via $T(r)$.
(b) Mapping of z_q to its corresponding value v_q via $G(z)$.
(c) Inverse mapping from s_k to its corresponding value of z_k.



Histogram Matching

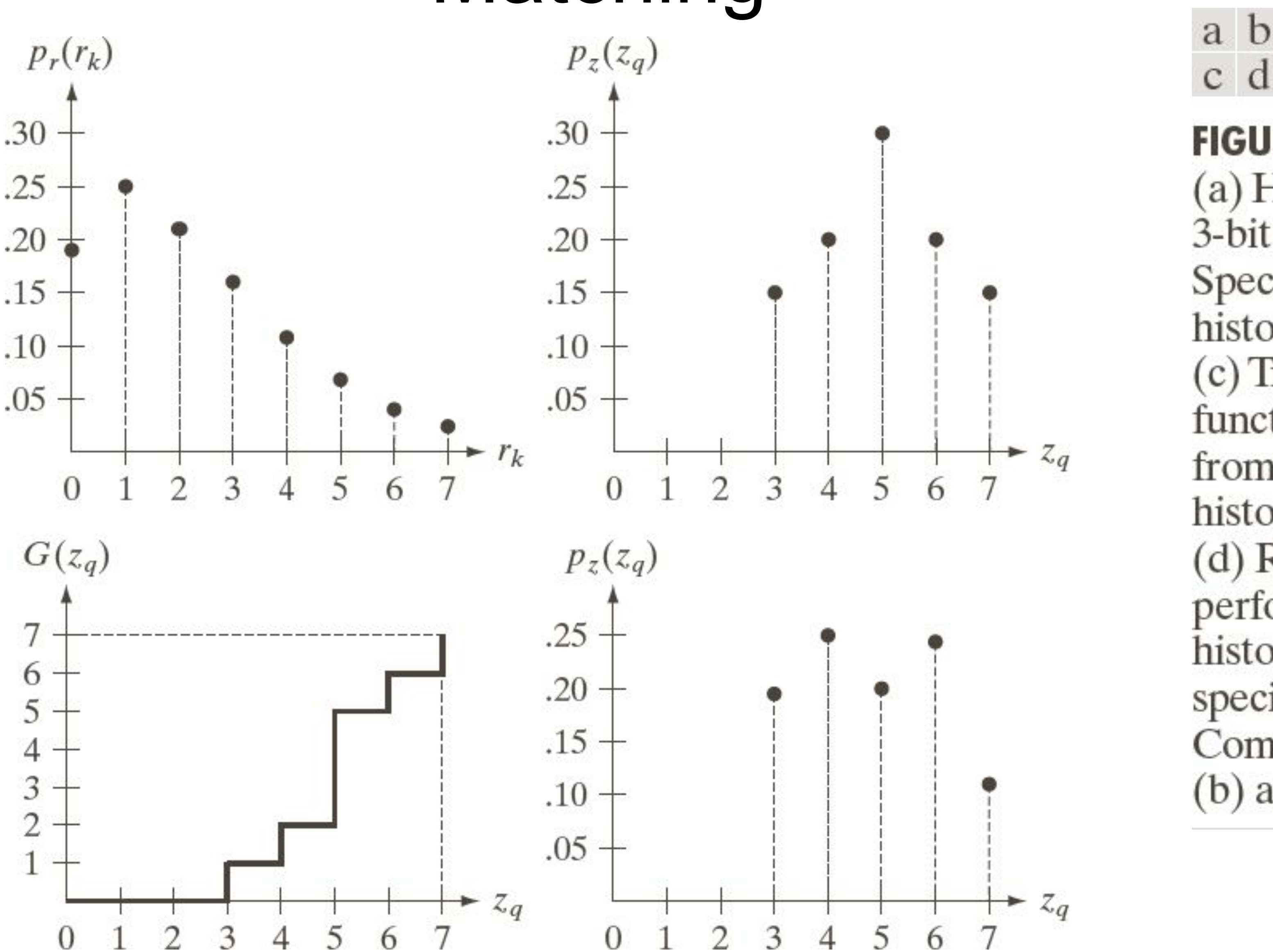


FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

Histogram Matching

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).

Histogram Matching

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

TABLE 3.3 All possible values of the transformation function G scaled, rounded, and ordered with respect to z.

r_k	s_k	$G(Z_q)$
0	1	0
1	3	0
2	5	0
3	6	1
4	6	2
5	7	5
6	7	6
7	7	7

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

TABLE 3.4 Mappings of all the values of s_k into corresponding values of z_q.

Histogram Matching

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

For Hist. Eq. in Fig. 3.22(d)

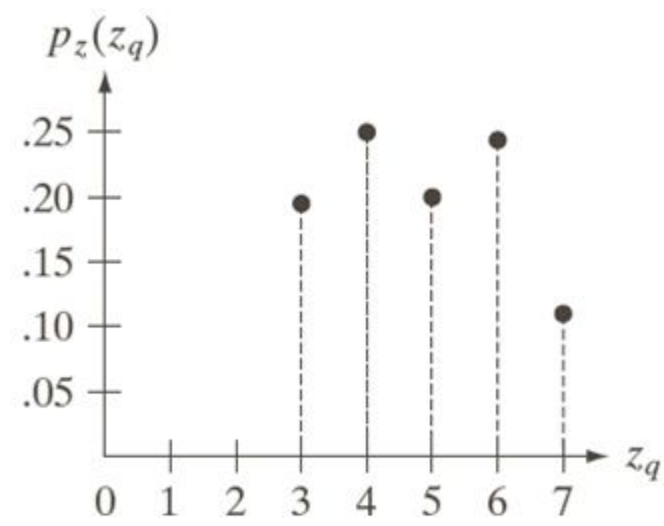
z_3 mapped to S_0=1 with value= 790/4096=0.19

z_4 mapped to S_1=3 with value = 1023/4096=0.24

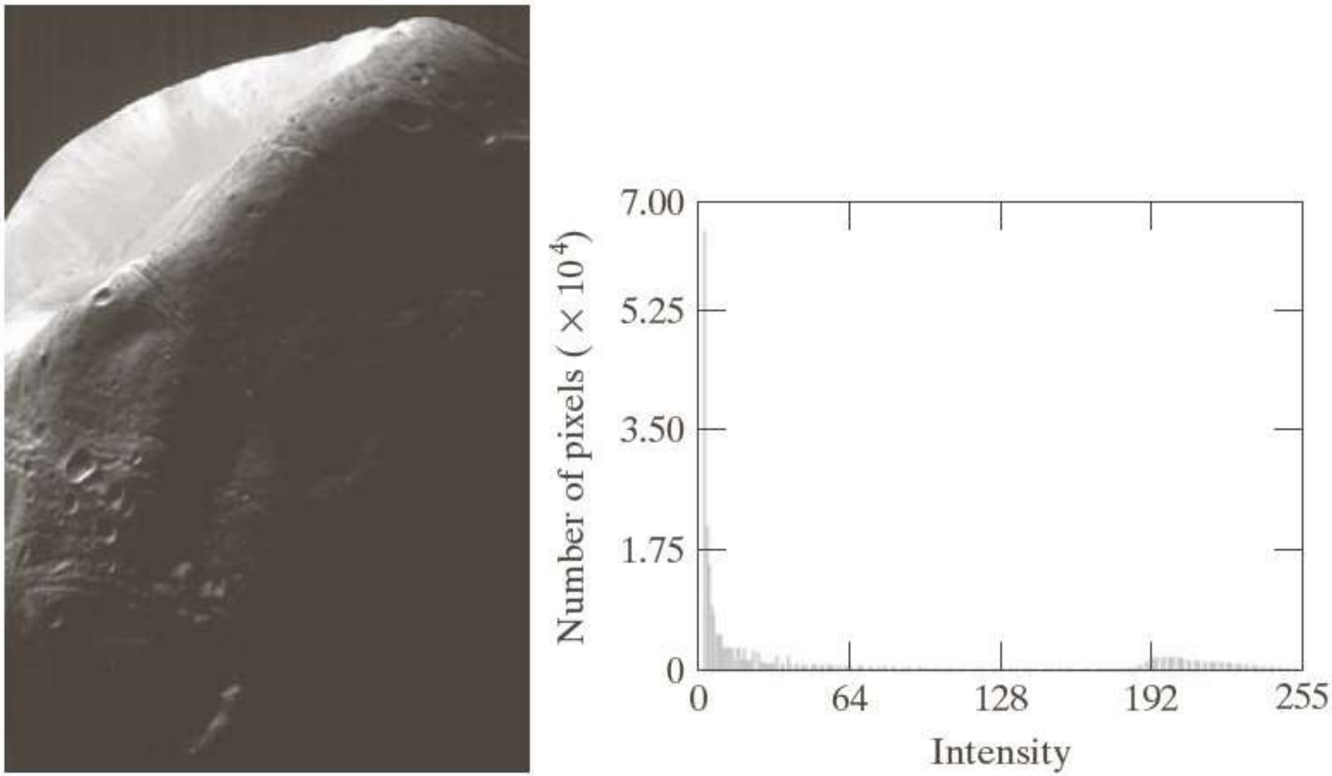
z_5 mapped to S_2=5 with value = 850/4096=0.21

z_3 ,z_4 mapped to 6 with value = (656+329)/4096=0.24

z_5 ,z_6 ,z_7 mapped to 7 with value = (245+122+81)/4096 = 0.10



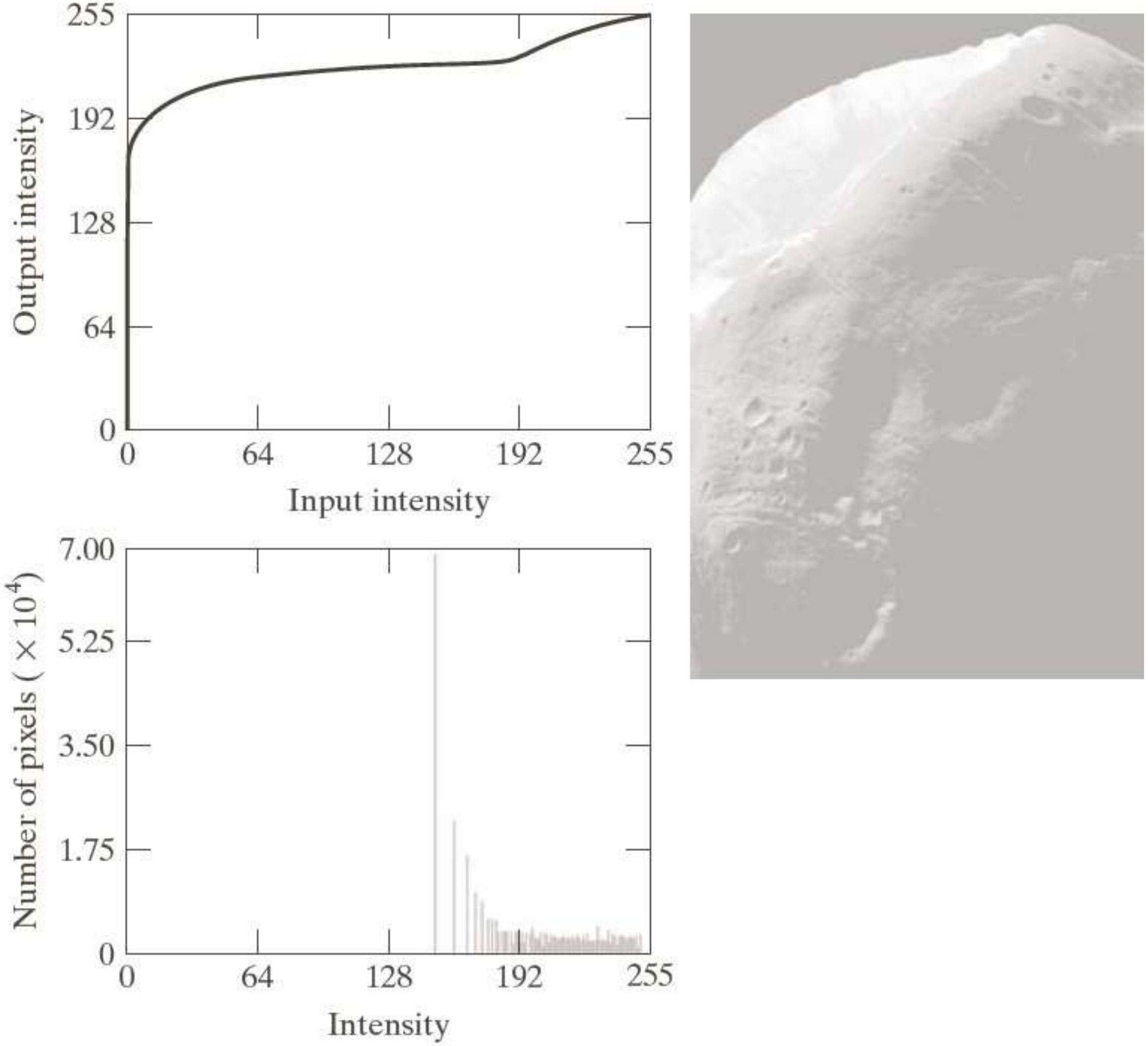
Histogram Matching



a b

FIGURE 3.23
(a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)

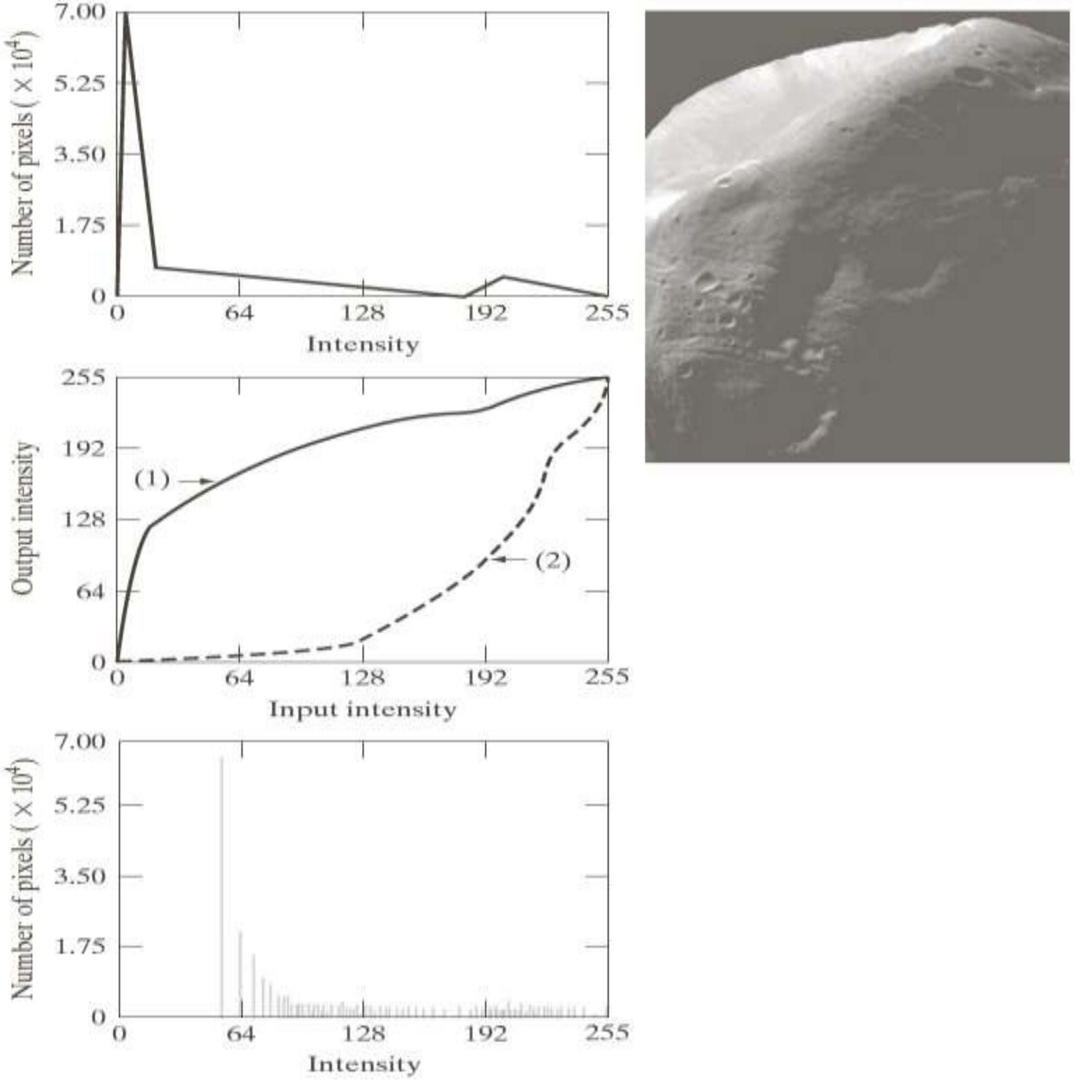
Histogram Matching



a b
c

FIGURE 3.24
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).

Histogram Matching



a	c
b	
d	

FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Adaptive histogram

Adaptive histogram equalization (AHE) is a technique used to improve **contrast** in images.
It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image.
It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.

Local Histogram

In this process a square or rectangular neighborhood area is chosen. Then histogram equalization of specification is used to process the center of the area. And then work like filtering.

Local Histogram Processing

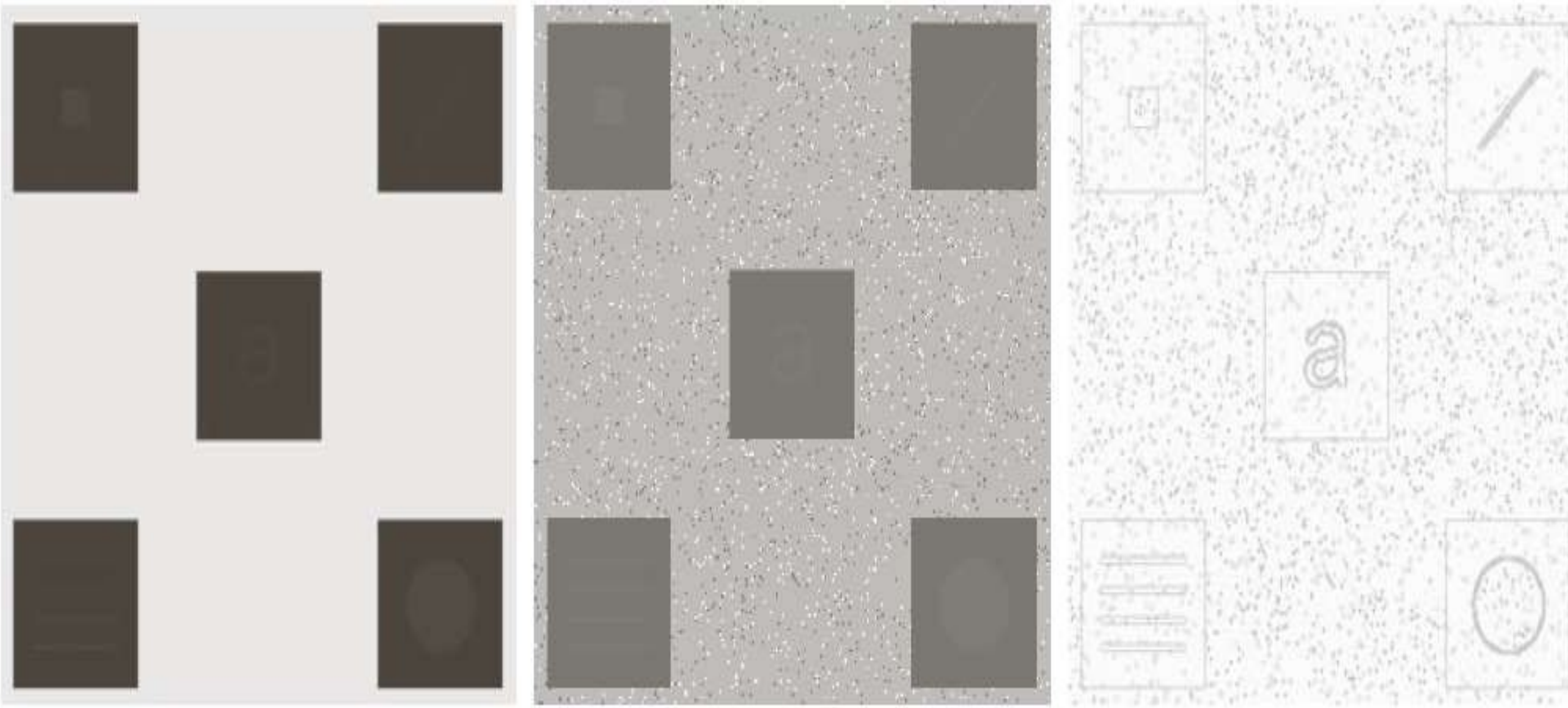
- Histogram Processing methods discussed in the previous two sections are *Global*, in the sense that pixels are modified by a transformation function based on the intensity distribution of an entire image.
- There are some cases in which it is necessary to enhance detail over small areas in an image.
- This procedure is to define a neighborhood and move its center pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed and either a histogram equalization or histogram specification transformation function is obtained.
- Map the intensity of the pixel centered in the neighborhood.
the neighborhood region is then moved to an adjacent pixel location and the procedure is

Local Histogram Processing

- This approach has obvious advantages over repeatedly computing the histogram of all pixels in the neighborhood region each time the region is moved one pixel location.

- Another approach used sometimes to reduce computation is to utilize non overlapping regions, but this method usually produces an undesirable "*blocky*" effect.

Local Histogram Processing



a b c

FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3×3.

Temporal processing

Processing involving more than one image (frame).
If 2 frame---- dyadic processing.
Addition: $g(x,y)= (f1(x,y) + f2(x,y)).k$
Subtraction : $g(x,y)= f1(x,y) - f2(x,y)$
Signal Averaging: $g(x,y)= 1/n(f1(x,y) + f2(x,y) + \ldots\ldots.+ fn(x,y))$
Multiplication: $g(x,y)= f1(x,y) * f2(x,y)$
OR : $g(x,y)= f1(x,y)$ OR $f2(x,y)$
AND : $g(x,y)= f1(x,y)$ AND $f2(x,y)$
XOR : $g(x,y)= f1(x,y)$ XOR $f2(x,y)$

Geometric Processing

Transpose/Rotation:

```
for i = 1 : 512
    for j = 1 : 512
        B(j; i) = A(i; j); OR >> B = A0;
    end
end
```

Flip: The vertical flipped image B (N x M) of A (N x M) can be obtained as B(i,M+1-j) = A(i,j) (i = 1….N and j = 1….M).

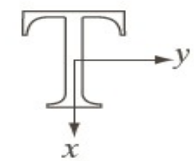
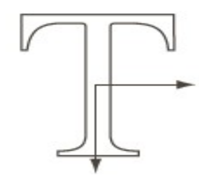
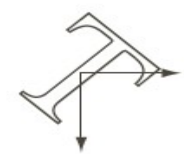
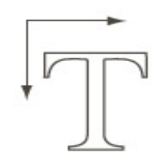
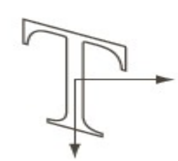
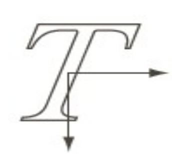
```
for i = 1 : 512
    for j = 1 : 512
        B(i, 512 + 1 - j) = A(i, j);
    end
end
```

Crop:

The cropped image B (N1xN2) of A(NxM), starting from (n1; n2), can be obtained as B(k; l) = A(n1+k; n2+l) (k = 0;…..N1-1; l = 0……..N2-1).

Geometric Spatial

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)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Spatial Filtering

- Also called spatial masks, kernels, templates, and windows.
- It consists of (1) a neighborhood (typically a small window), and (2) a predefined operation that is performed on the image pixels encompassed by the neighborhood.
- Filtering creates a new pixel with coordinates equal to the center of the neighborhood.
- If operation is linear, then filter is called a *linear spatial filter* otherwise *nonlinear*.

Mechanics of Spatial Filteri

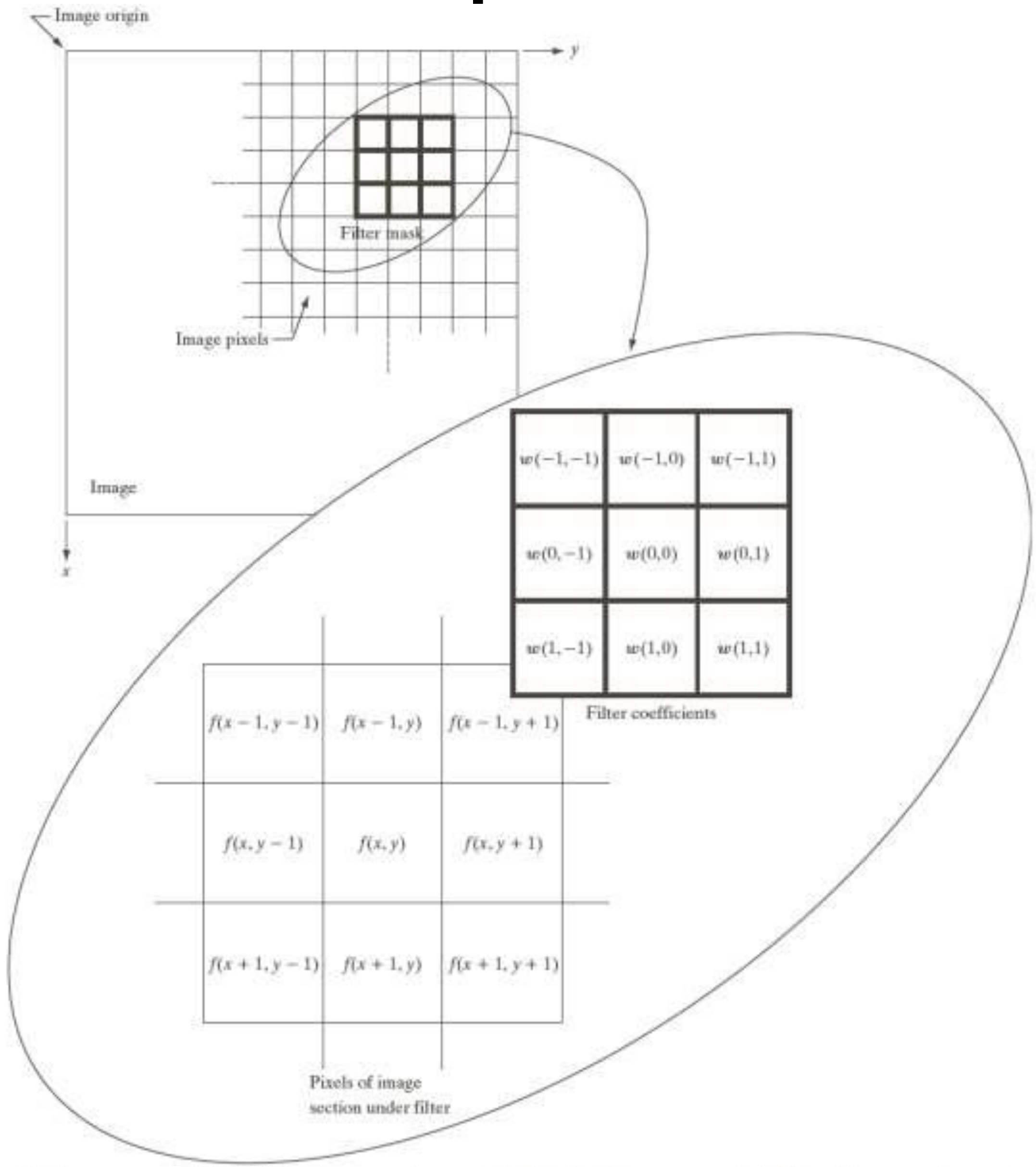


FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Spatial Correlation & Convolution

- Correlation is the process of moving a filter mask over the image and computing the sum of the products at each location.
- Convolution process is same except that the filter is first rotated by 180 degree.

Correlation

Origin f w

(a) 0 0 0 1 0 0 0 0 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 w rotated 180°

0 0 0 1 0 0 0 0 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)

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

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.

Mask operation

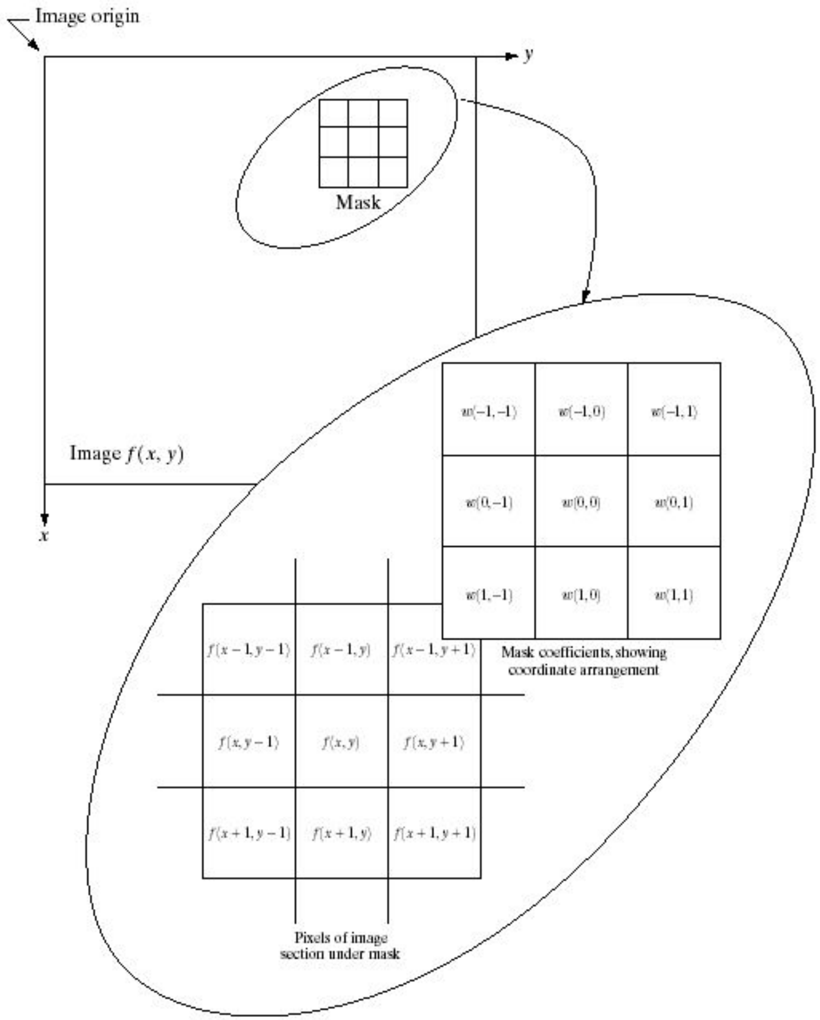


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Mask operation

FIGURE 3.33 Another representation of a general 3 × 3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Spatial filtering

There are two types of spatial filtering:

1. Smoothing filtering:

Smoothing filtering are used for

-- Blurring image

-- Noise reduction

2. Sharpening filtering

Smoothing Spatial Linear Filters

- Also called *averaging filters* or *Lowpass filter*.
- By replacing the value of every pixel in an image by the average of the intensity levels in the neighborhood defined by the filter mask.
- Reduced “sharp” transition in intensities.
- Random noise typically consists of sharp transition.
- Edges also characterized by sharp intensity transitions, so averaging filters have the undesirable side effect that they blur edges.
- If all coefficients are equal in filter than it is also

Smoothing Spatial Linear Filters

- The mask is called *weighted average* terminology used to indicate that pixels are multiplied by different coefficient.
- Center point is more weighted than any other points.
- Strategy behind weighing the center point the highest and then reducing value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process.
- Intensity of smaller object blends with background

Smoothing Linear/ lowpass Filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

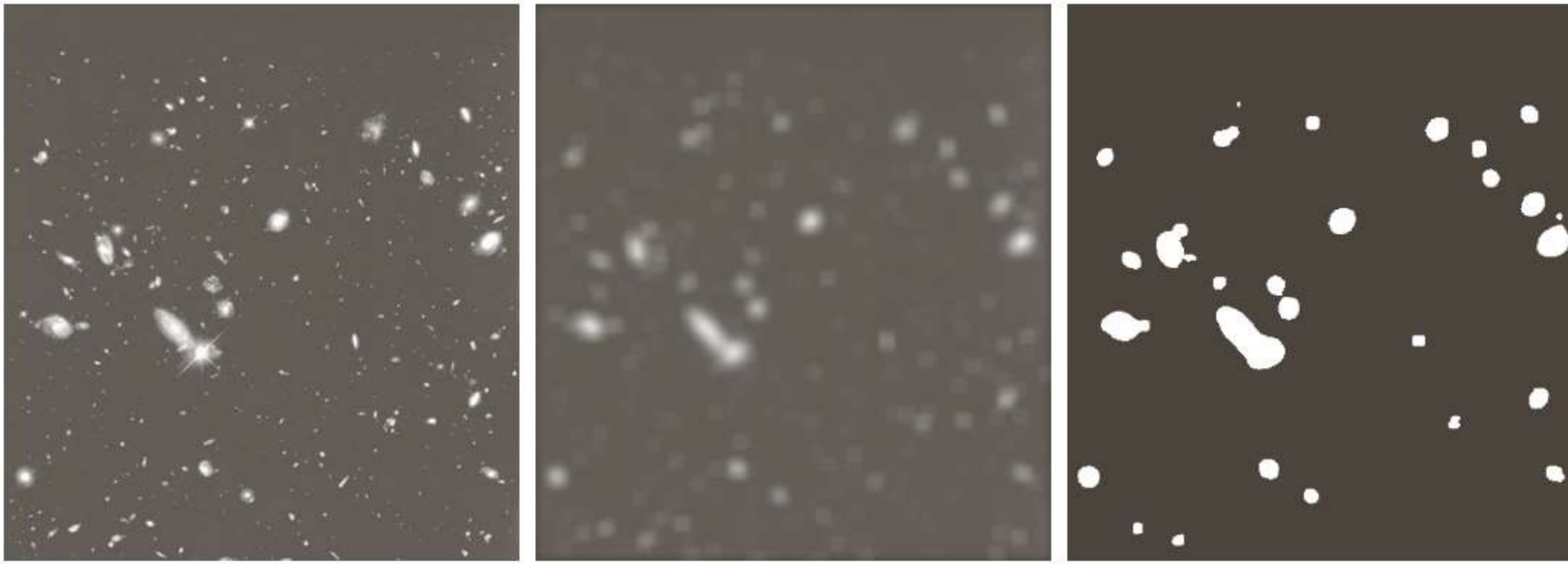
a b

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.

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$

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



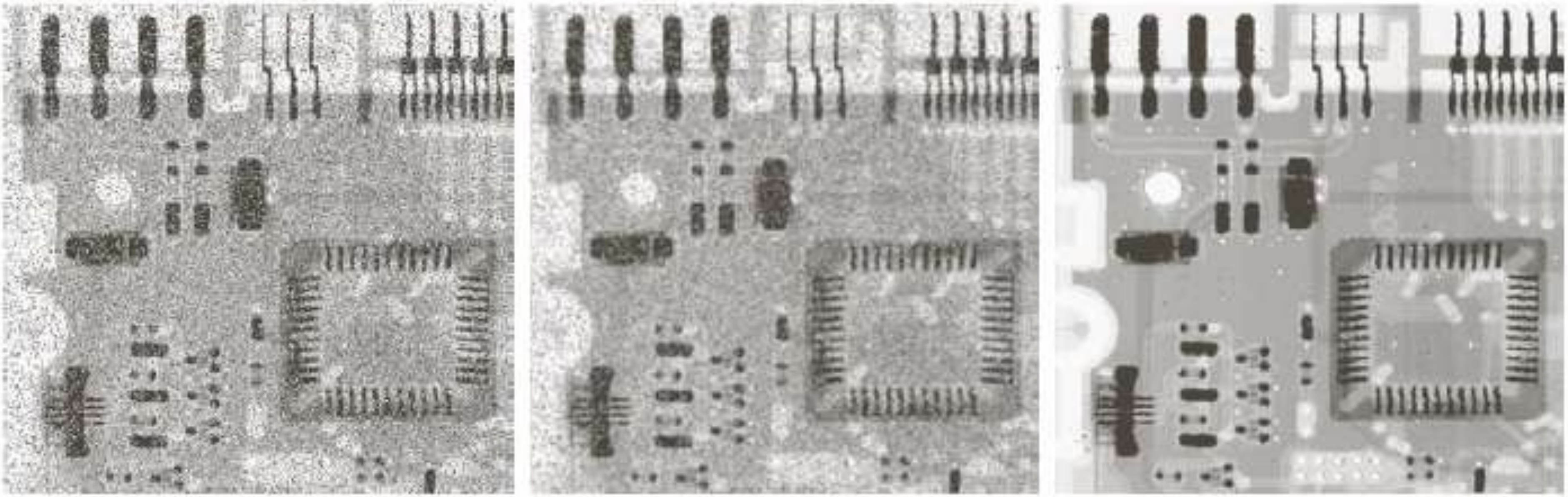
a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Order-Statistic (Nonlinear) Filters

- Response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.
- Best-known filter is *median filter*.
- Replaces the value of a center pixel by the median of the intensity values in the neighborhood of that pixel.
- Used to remove *impulse or salt-pepper noise*.
- Larger clusters are affected considerably less. ?
- Median represents the 50^{th} percentile of a ranked set of numbers while 100^{th} or 0^{th} percentile results in the so- called *max filter or min filter* respectively.

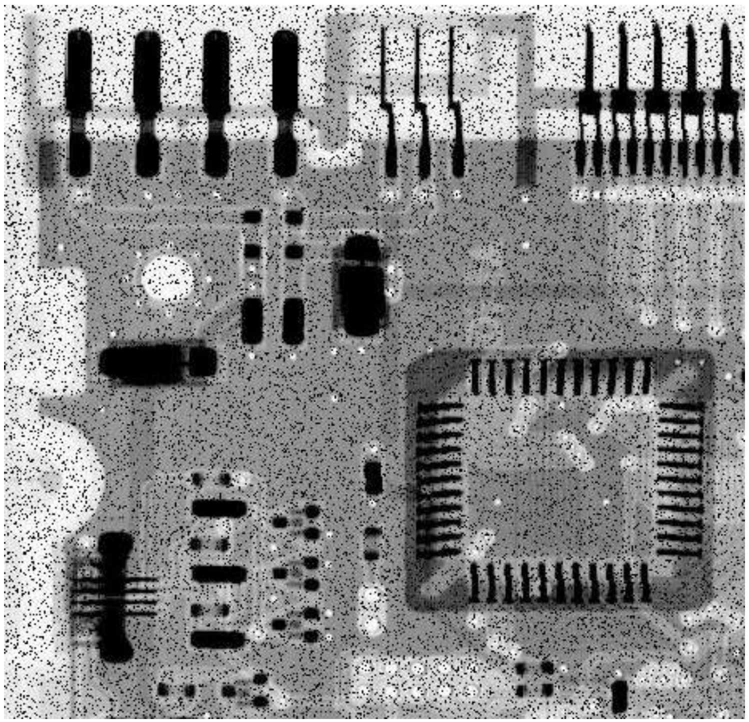
Median Filter (Nonlinear)

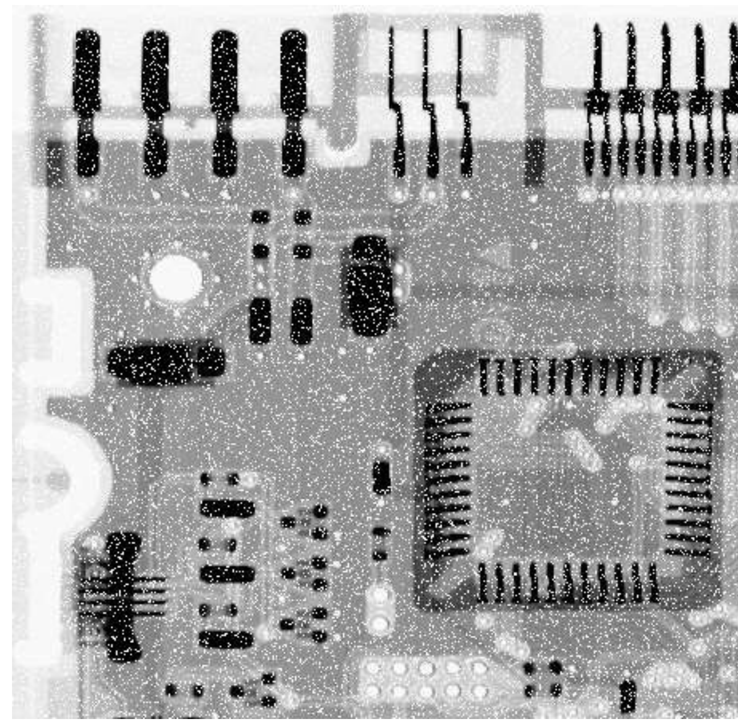


a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Median Filter (Nonlinear)



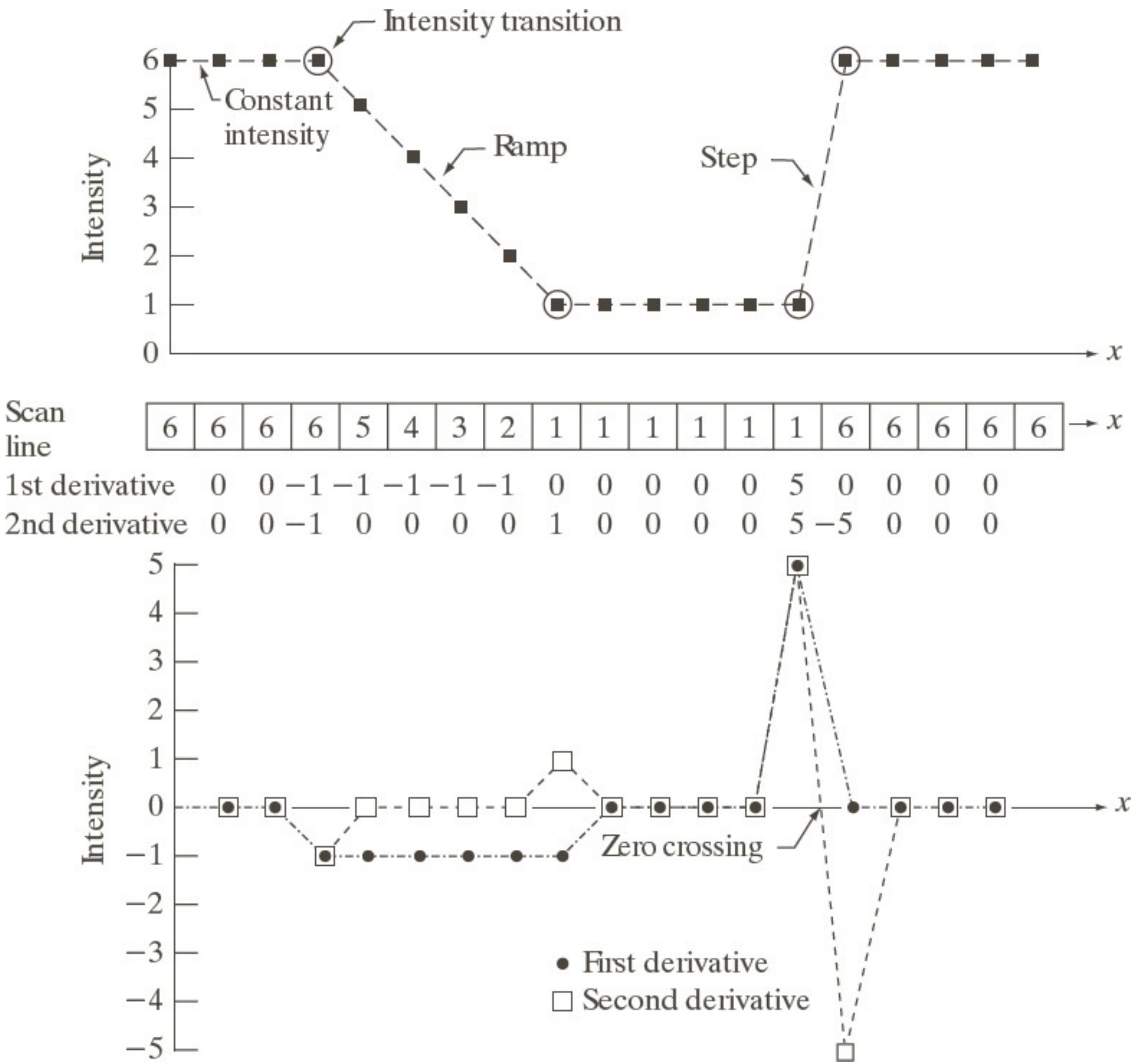


Sharpening Spatial Filters

- Objective of sharpening is to highlight transitions in intensity.
- Uses in printing and medical imaging to industrial inspection and autonomous guidance in military systems.
- Averaging is analogous to integration, so sharpening is analogous to spatial differentiation.
- Thus, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying intensities.

Foundation

- Definition for a first order derivative (1) must be zero in areas of constant intensity (2) must be nonzero at the onset of an intensity step or ramp and (3) must be nonzero along ramps.
- For a second order derivatives (1) must be zero in constant areas (2) must be nonzero at the onset and (3) must be zero along ramps of constant slope.
- First order derivative of a one dimensional function f(x) is the difference of f(x+1) – f(x).
- Second order = 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.

Smoothing filtering

b) Order Statistic Filtering:

- Max
- Min
- Median filtering
  - To remove impulse noise, also called salt and pepper noise.

Sharpening filtering

High pass filtering

-1	-1	-1
-1	8	-1
-1	-1	-1

Sharpening filtering

High boost filtering

Original image= high pass+ low pass filter
High boost = A. original –low pass
= (A-1) original + High pass.
[A=multiplication factor]

W=9A-1

-1	-1	-1
-1	W	-1
-1	-1	-1

Sharpening filtering

Derivative filtering
- First order:
  - Prewitt
  - Sobel
  - Roberts
- Second order:
  - Laplacian

Sharpening filtering

First order derivative

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

a	
b	c
d	e
f	g

FIGURE 10.14 A 3×3 region of an image (the z's are intensity values) and various masks used to compute the gradient at the point labeled z_5.

Sharpening filtering

First order derivative

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

a
b c
d e

FIGURE 3.41 A 3×3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

The Laplacian (2nd order derivative)

Shown by Rosenfeld and Kak[1982] that the simplest isotropic derivative operator is the Laplacian is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

The Laplacian (2^{nd} order derivative)

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

Implementation

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \textit{If the center coefficient is negative} \\ f(x,y)+\nabla^2 f(x,y) & \textit{If the center coefficient is positive} \end{cases}$$

Where f(x,y) is the original image

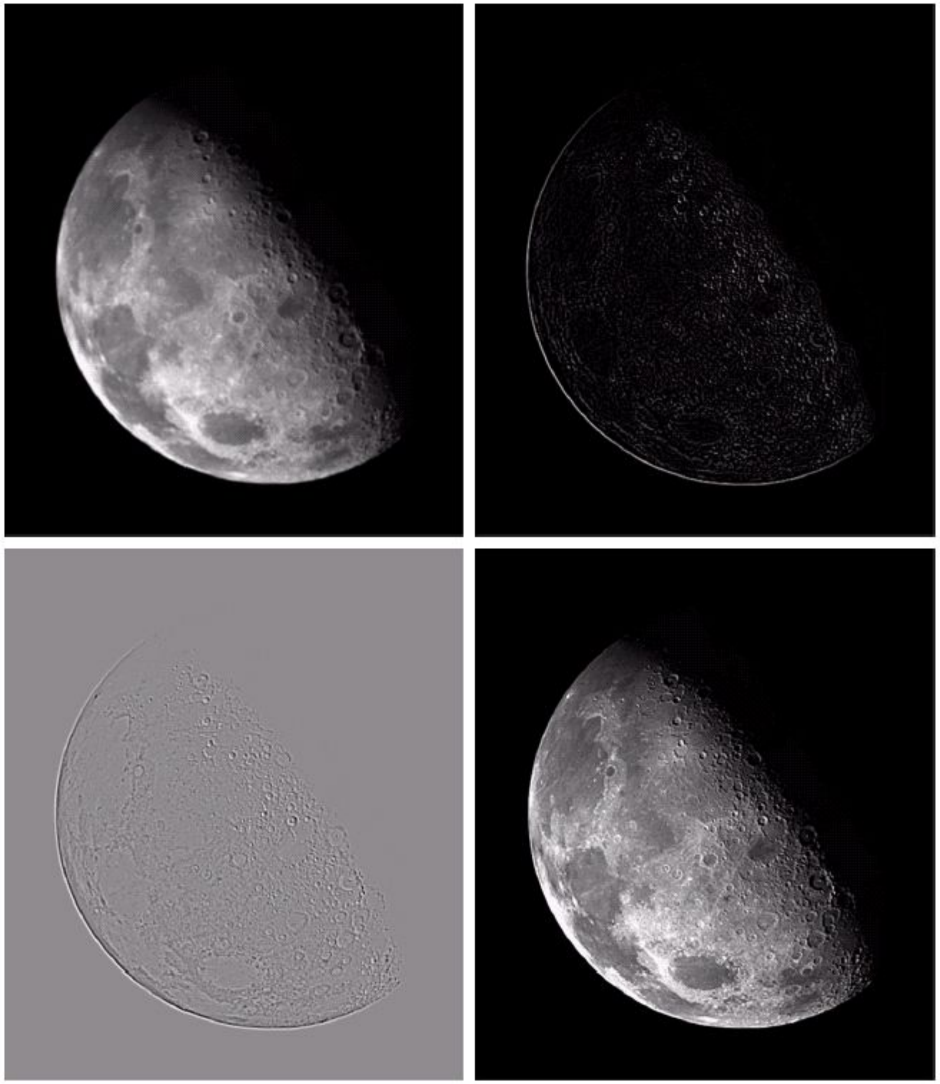
$\nabla^2 f(x,y)$ is Laplacian filtered image

g(x,y) is the sharpen image

The Laplacian (2^{nd} order derivative)

a	b
c	d

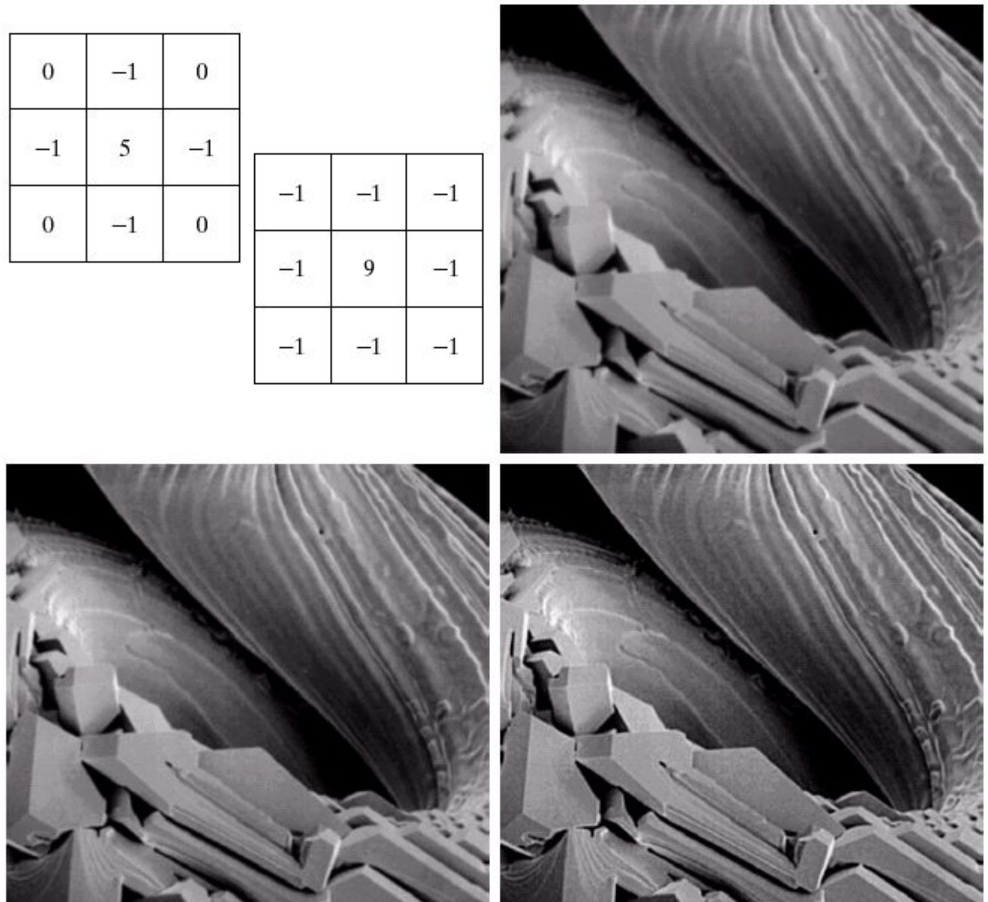
FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)



The Laplacian (2nd order derivative)

0	–1	0
–1	5	–1
0	–1	0

–1	–1	–1
–1	9	–1
–1	–1	–1



a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Second Derivatives-The Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

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

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

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y).$$

Second Derivatives - The Laplacian

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

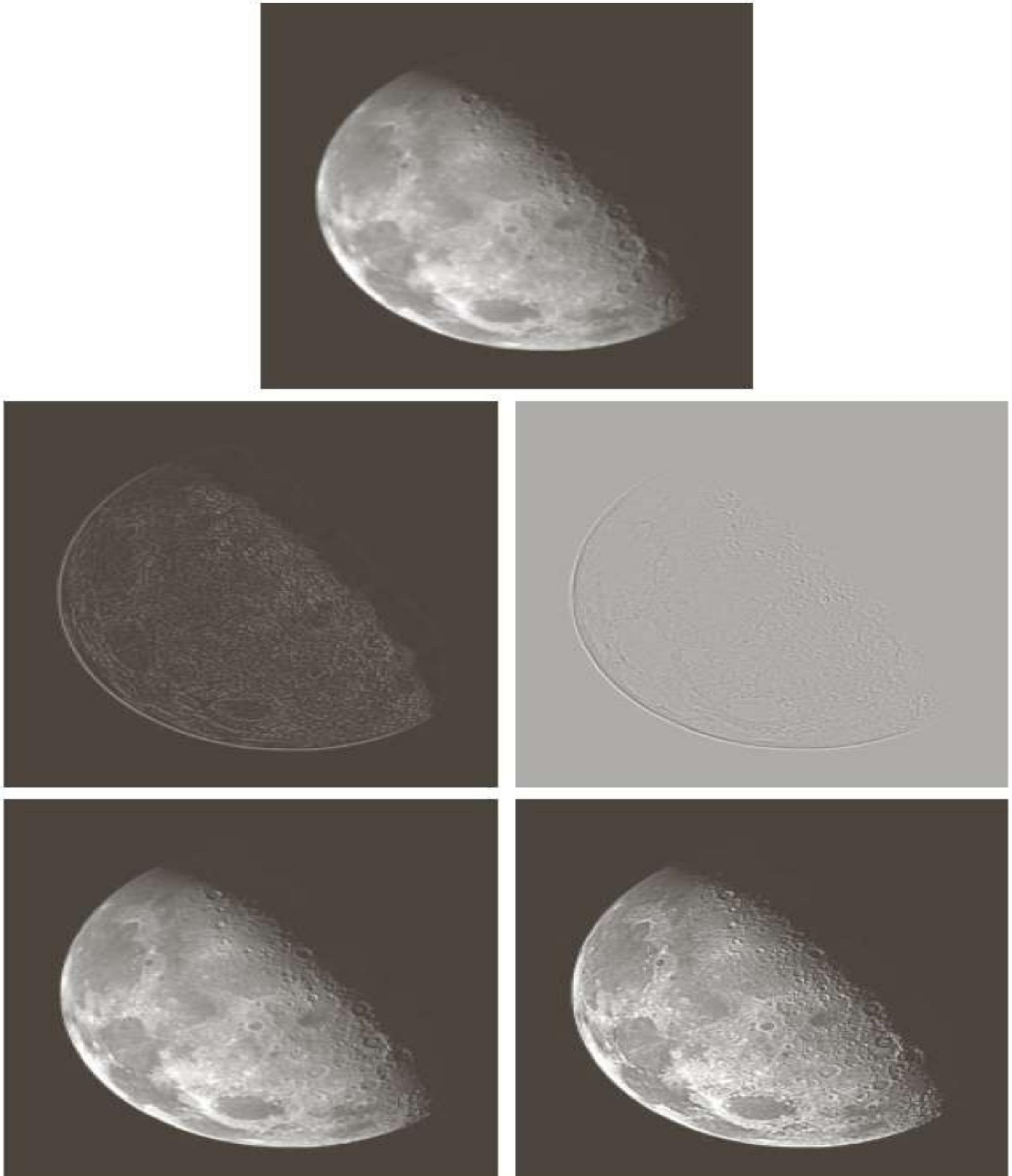
−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.37
(a) Filter mask used to implement Eq. (3.6-6).
(b) Mask used to implement an extension of this equation that includes the diagonal terms.
(c) and (d) Two other implementations of the Laplacian found frequently in practice.

Second Derivatives-The Laplacian

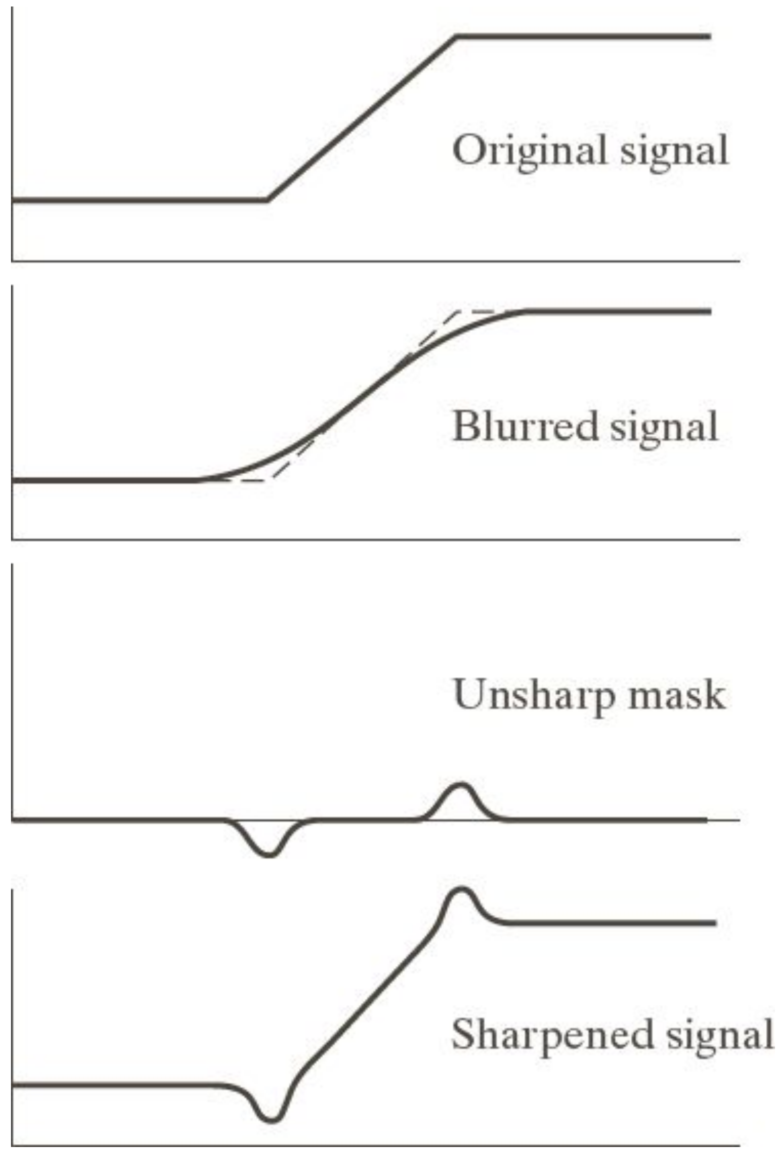
$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the} \\ & \text{Laplacian mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the} \\ & \text{Laplacian mask is positive.} \end{cases}$$



a
b c
d e

FIGURE 3.38
(a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

Unsharp Masking and High boost Filtering



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Unsharp Masking and High boost Filtering

- Unsharp Masking
  - Read Original Image $f(x,y)$
  - Blurred original image $f'(x,y)$
  - Mask = $f(x,y) - f'(x,y)$
  - $g(x,y) = f(x,y) + \text{Mask}$
- High Boost Filtering
  - Read Original Image $f(x,y)$
  - Blurred original image $f'(x,y)$
  - Mask = $f(x,y) - f'(x,y)$
  - $g(x,y) = f(x,y) + k*\text{Mask}$, where $k>1$

Unsharp Masking and High boost Filtering

a
b
c
d
e

FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

First Derivative – The Gradient

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$\begin{aligned} \nabla f &= \mathrm{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \end{aligned}$$

$$\nabla f \approx |G_x| + |G_y|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

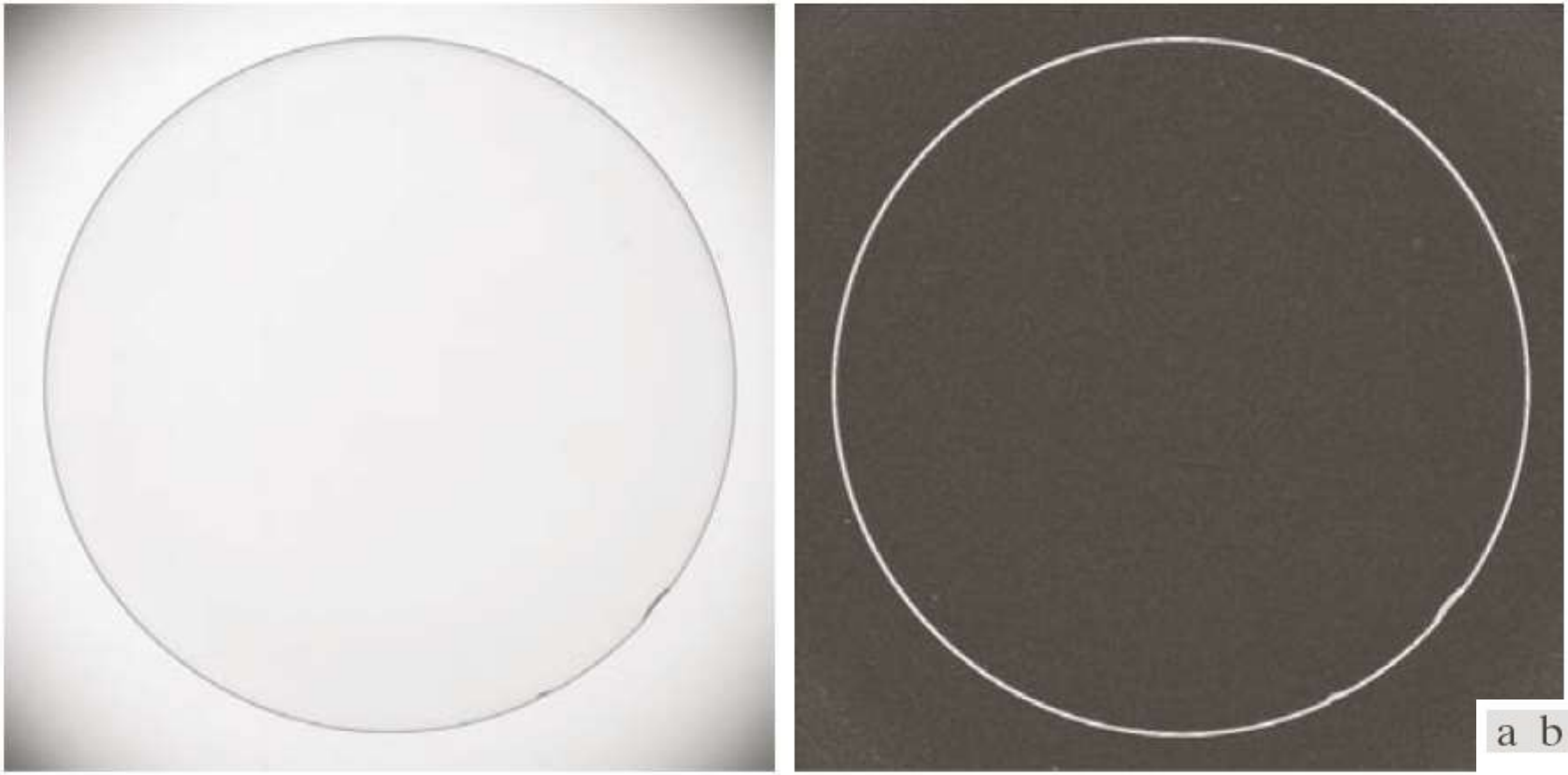
0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

a
b c
d e

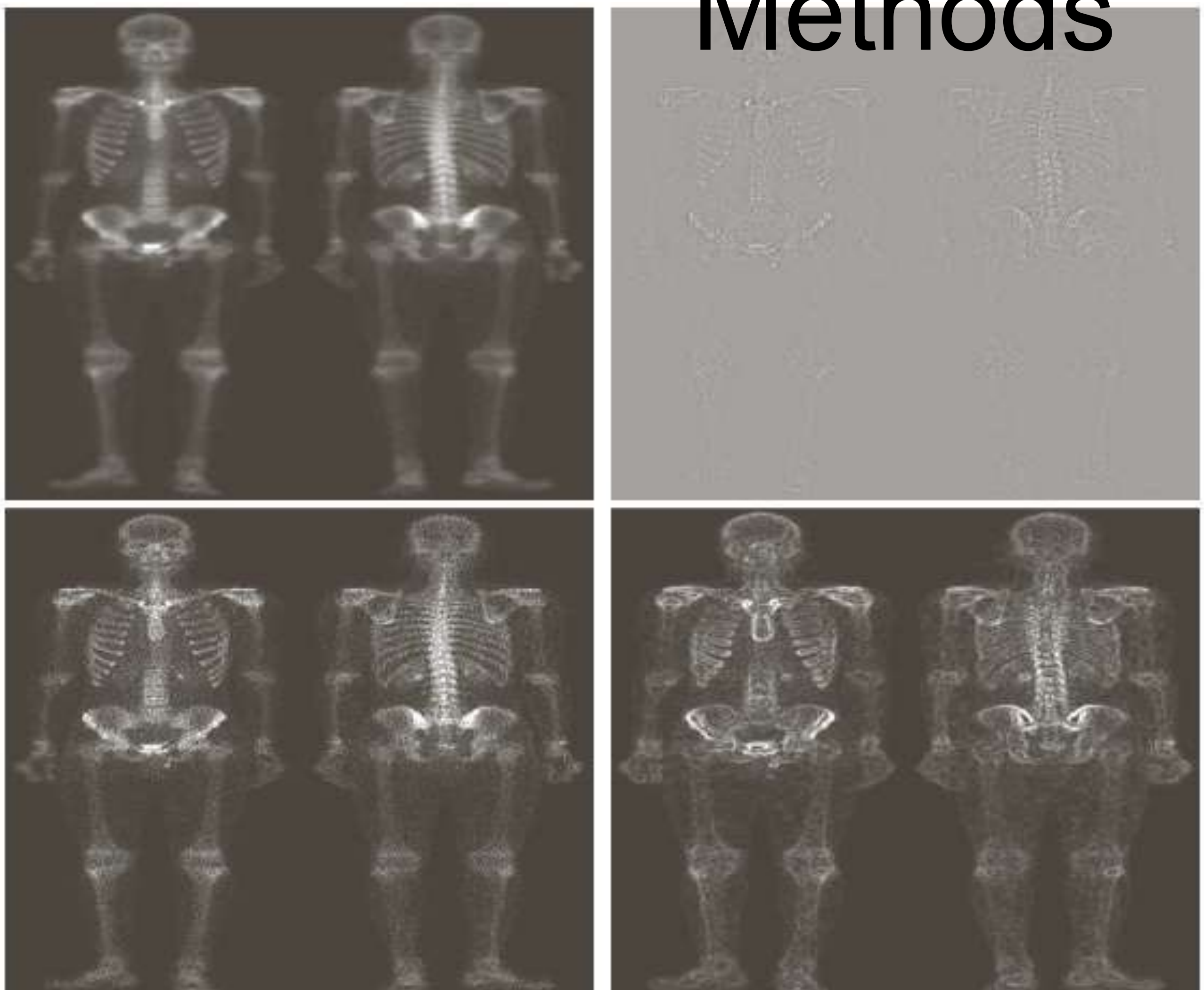
FIGURE 3.41
A 3×3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.



a b

FIGURE 3.42
(a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.
(Original image courtesy of Pete Sites, Perceptics Corporation.)

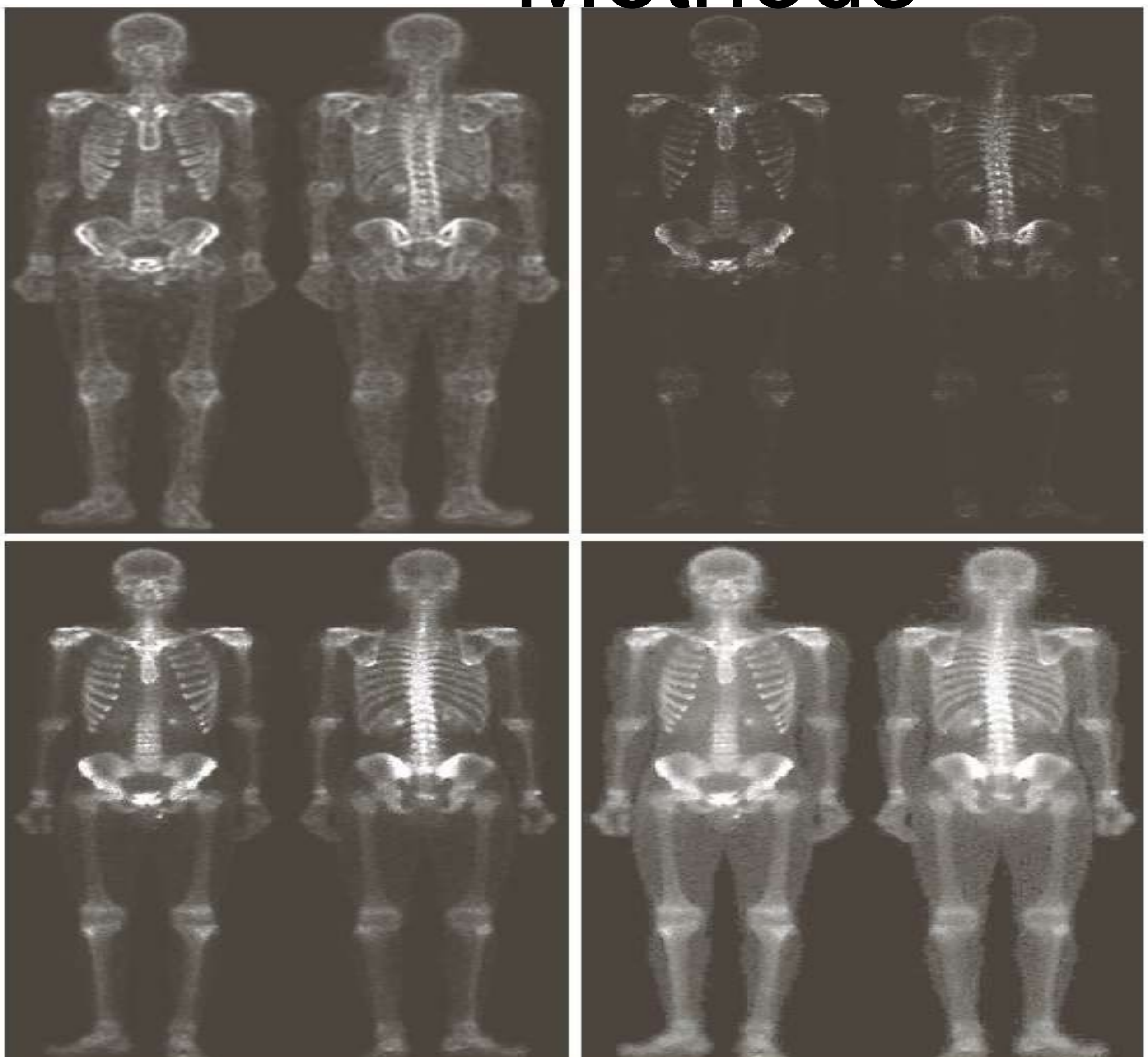
Combining Spatial Enhancement Methods



a b
c d

FIGURE 3.43
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b).
(d) Sobel gradient of (a).

Combining Spatial Enhancement Methods



e	f
g	h

FIGURE 3.43 *(Continued)* (e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)