INTENSITY TRANSFORMATION & SPATIAL FILTERING

SESSIONAL I

Text Books

- R.C.Gonzalez and R.E.Woods, "Digital Image Processing", Addison-Wesley Longman, Inc, 1999 .
- A.K.Jain, "Digital Image Processing", PHL
- M.Sonka, V.Hlavac, and R.Boyle – Image processing, Analysis and Machine vision, Thomson Asia pvt. Ltd, 1999

Example Bartlane transmitted image

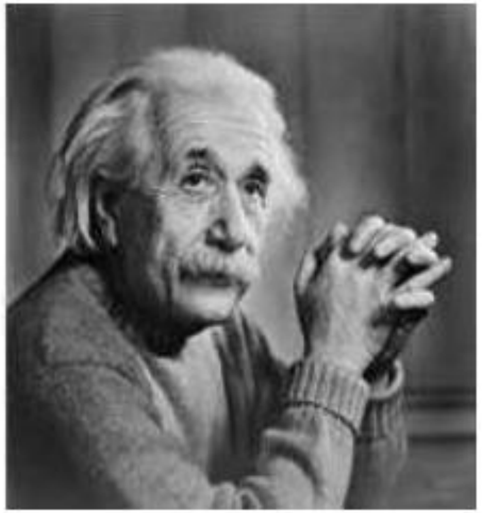
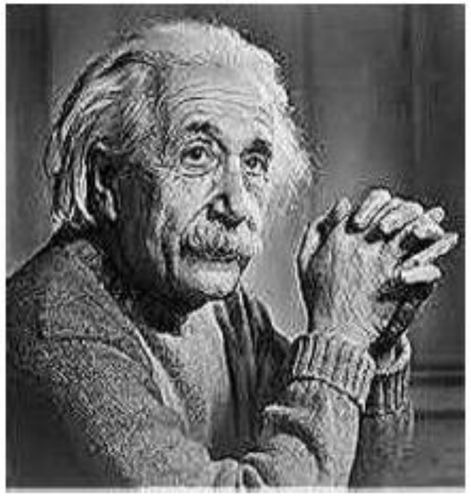
- Specialized printing equipment coded pictures for transmission
- Received and printed on a telegraph printer fitted with type faces to simulate a halftone pattern
- Initial problems
  - Poor visual quality related to printing process and the distribution of brightness levels
- Image produced in 1921 from a coded tape by a telegraph printer with special type faces

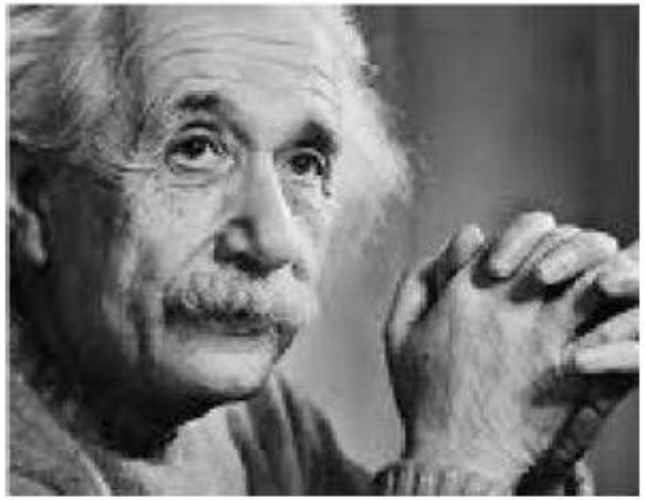
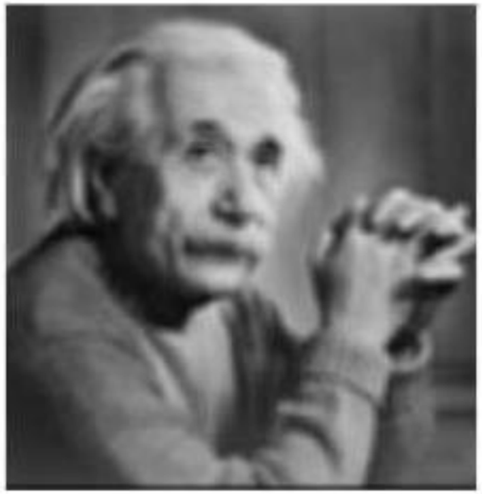
15 level Bartlane image

- Early images were transmitted using 5 distinct brightness levels
- The process was improved in 1929 to 15 levels
- A system for developing a film plate (as opposed to printing) from the coded picture tape improved the reproduction process considerably
- Cable picture of Generals Pershing and Foch, transmitted in 1929 by 15-tone equipment from London to New York

What is DIP?

- Digital image processing deals with manipulation of digital images through a digital computer.

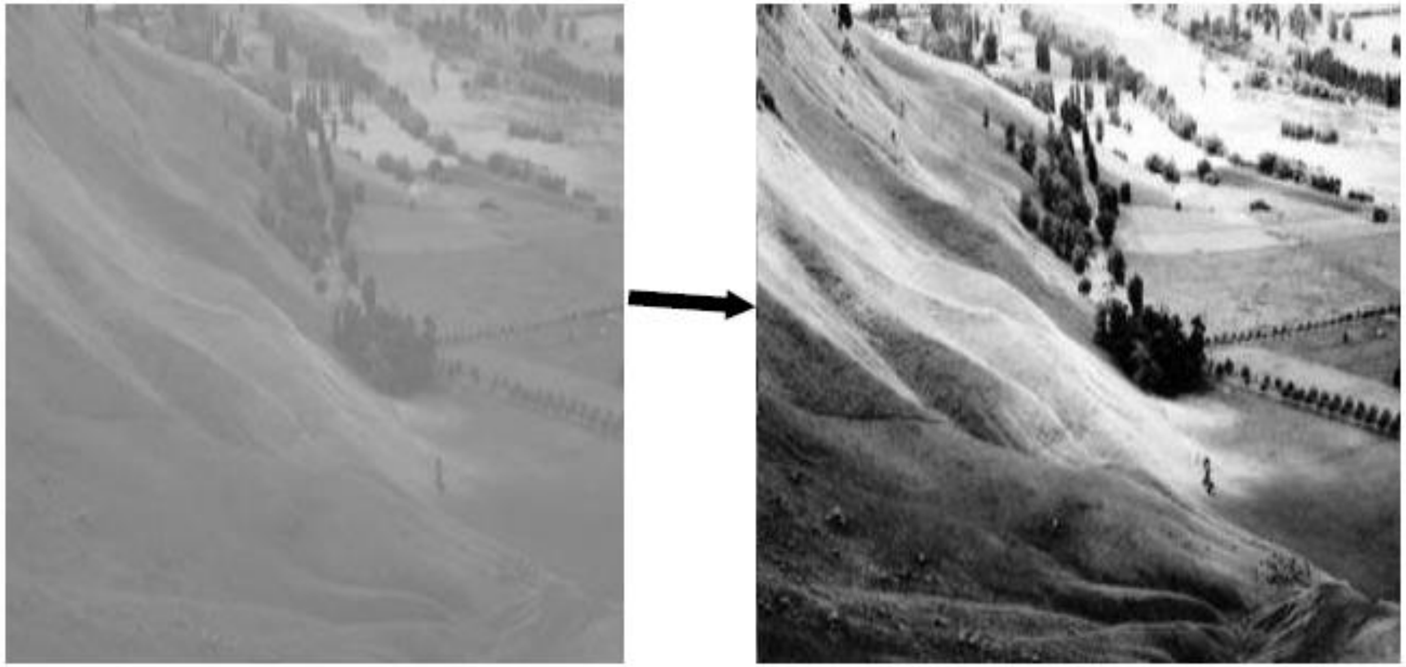




After
Before

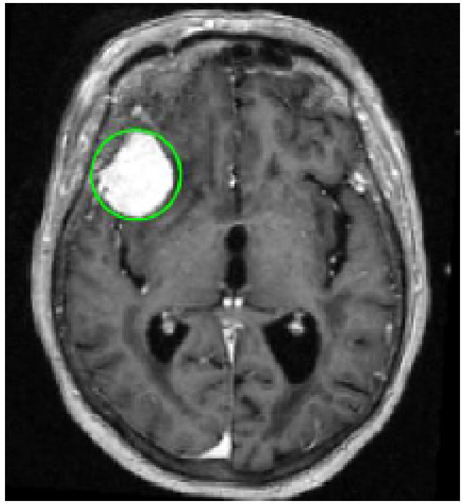
After
Before


Age:27 (Male)
Age:27 (Female)



Other applications:

- Security: biometrics



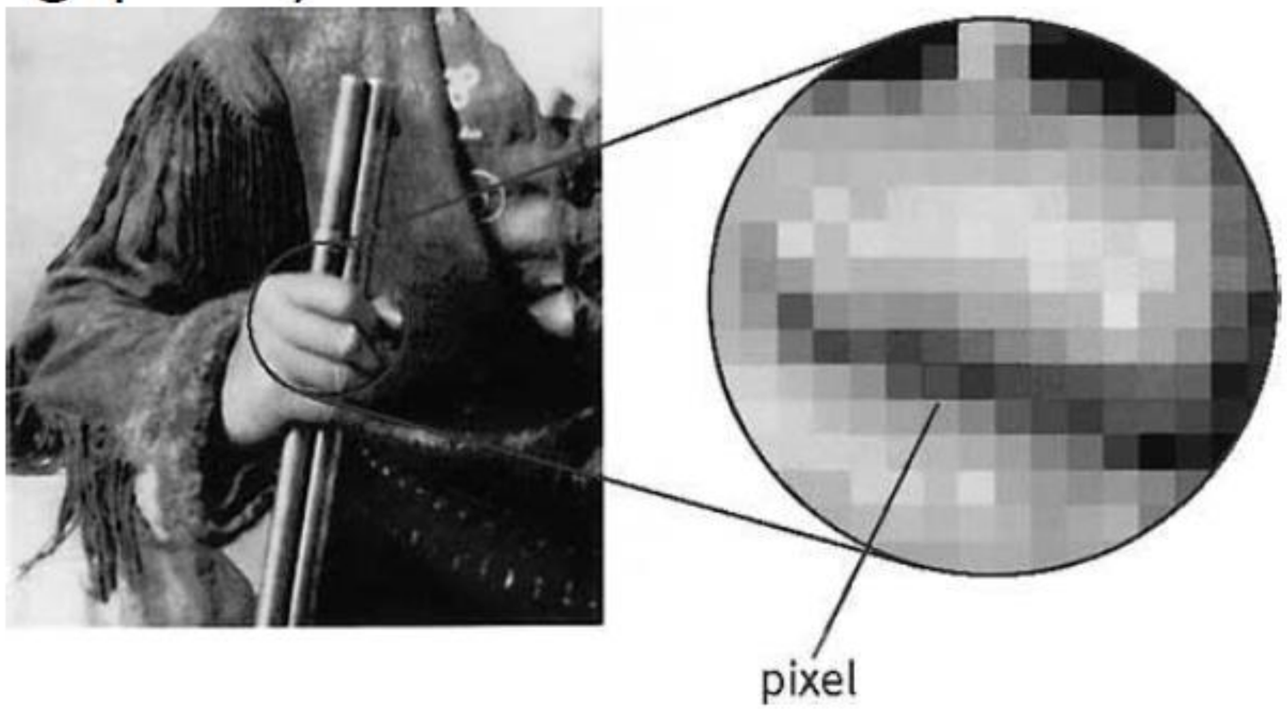
- Medical Imaging

And many more!!

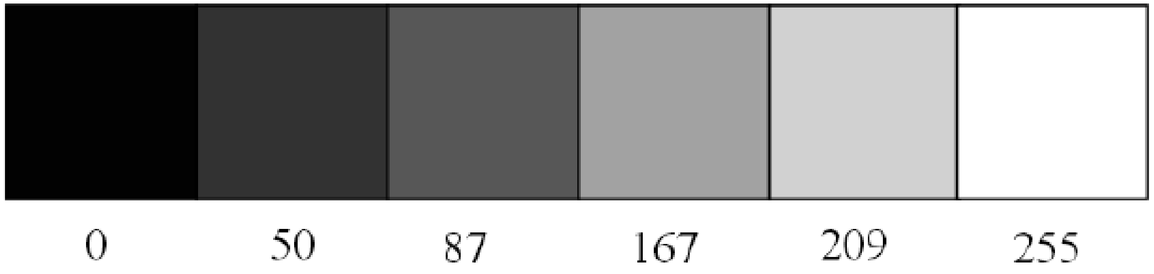
Some important Terms

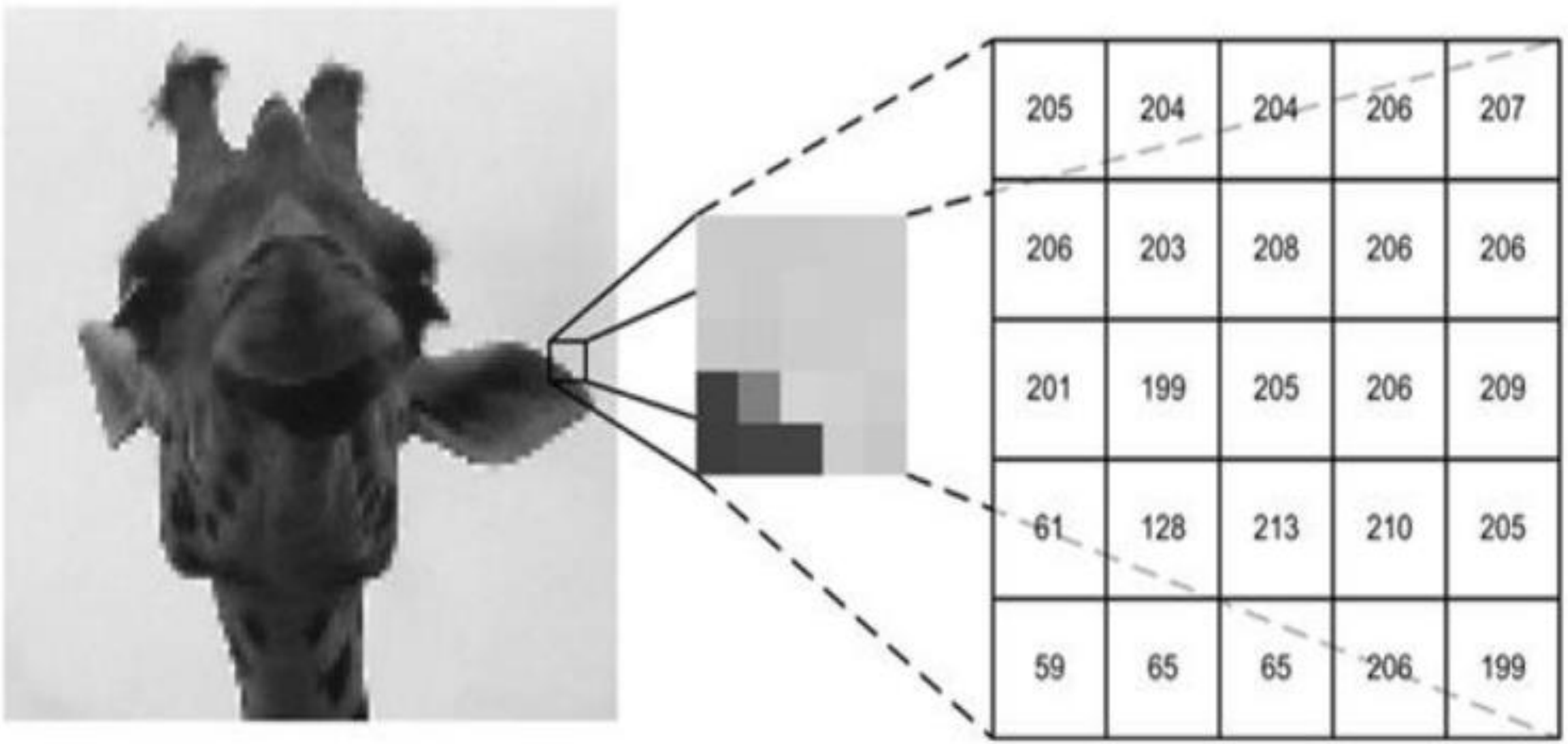
- Pixel
- Intensity
- Brightness
- Contrast
- Gray levels
- Resolution : 1024 x 768, 1280 x 1024, 1920 x 1080 ???

- Resolution :Number of pixels in an image.
- 2048 x 1536 : An image that is 2048 pixels wide and 1536 pixels high.
- It contains (multiply) 3,145,728 pixels (or 3.1 Megapixels).



Information on Pixels:





Manipulation of pixel value:

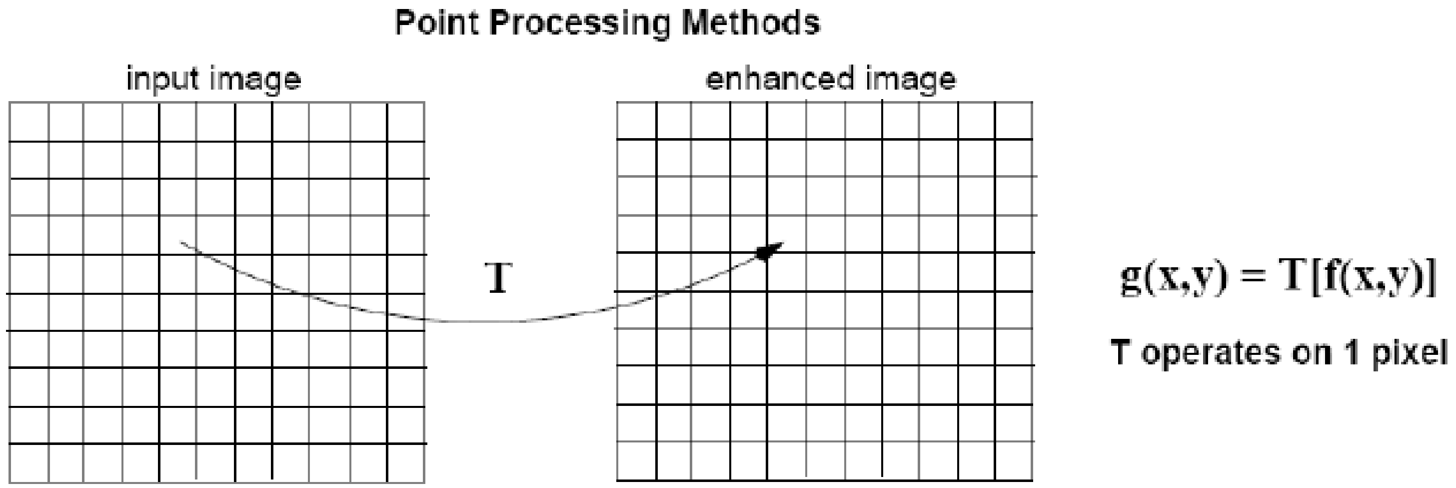
- Spatial domain processing
  - Direct manipulation of pixel

- Transform domain processing
  - Transform the image in suitable domain
  - Perform suitable operation
  - Inverse transform

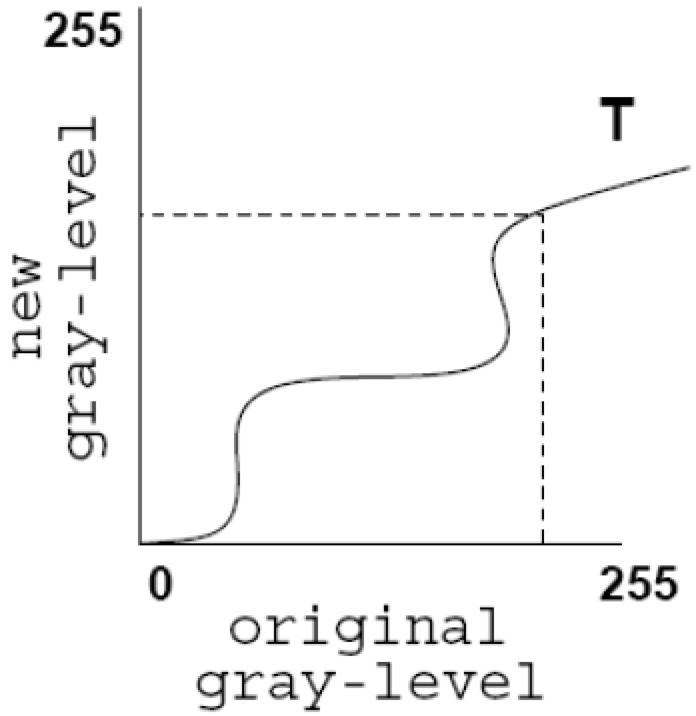
Which one is computationally efficient?

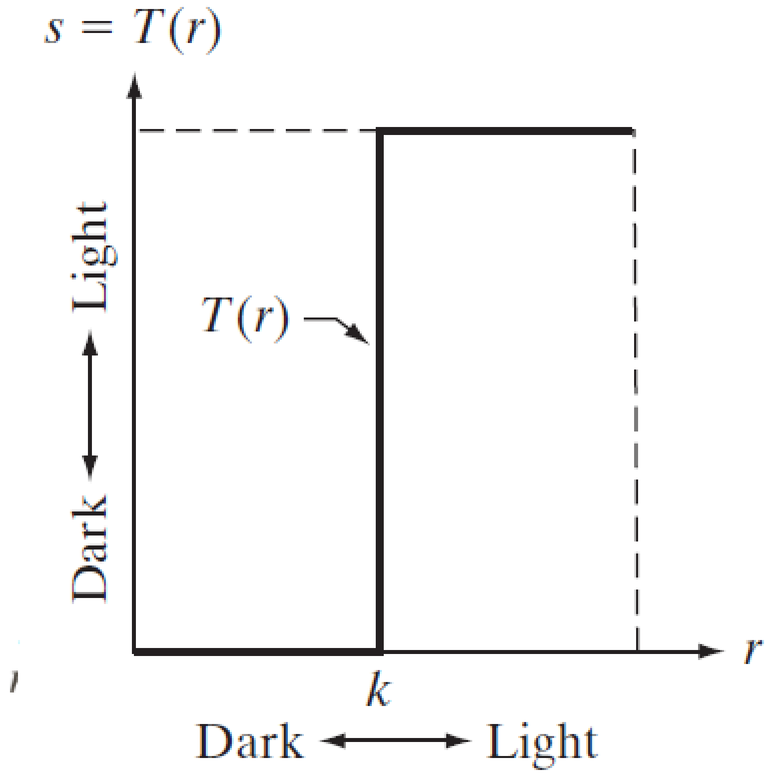
Spatial Domain processing:

- Intensity transformations:



Convert a given pixel value to a new pixel value based on some predefined function.





Which operation is this?


Some Notations

- '*r*' is Input image.
- *'s'* is Output image.
- $s(x, y)$ and $r(x, y)$ represents arbitrary pixel value of position (x, y) in output and input image respectively.
- Number of intensity levels in an image is 0 to $L - 1$. Here, $L = 2^k$. k is the number of bits used to store a pixel.
- $s(x, y) = T[r(x, y)]$ where T is intensity transformation function.
- This can be represented as $s = T(r)$.

Spatial Filtering

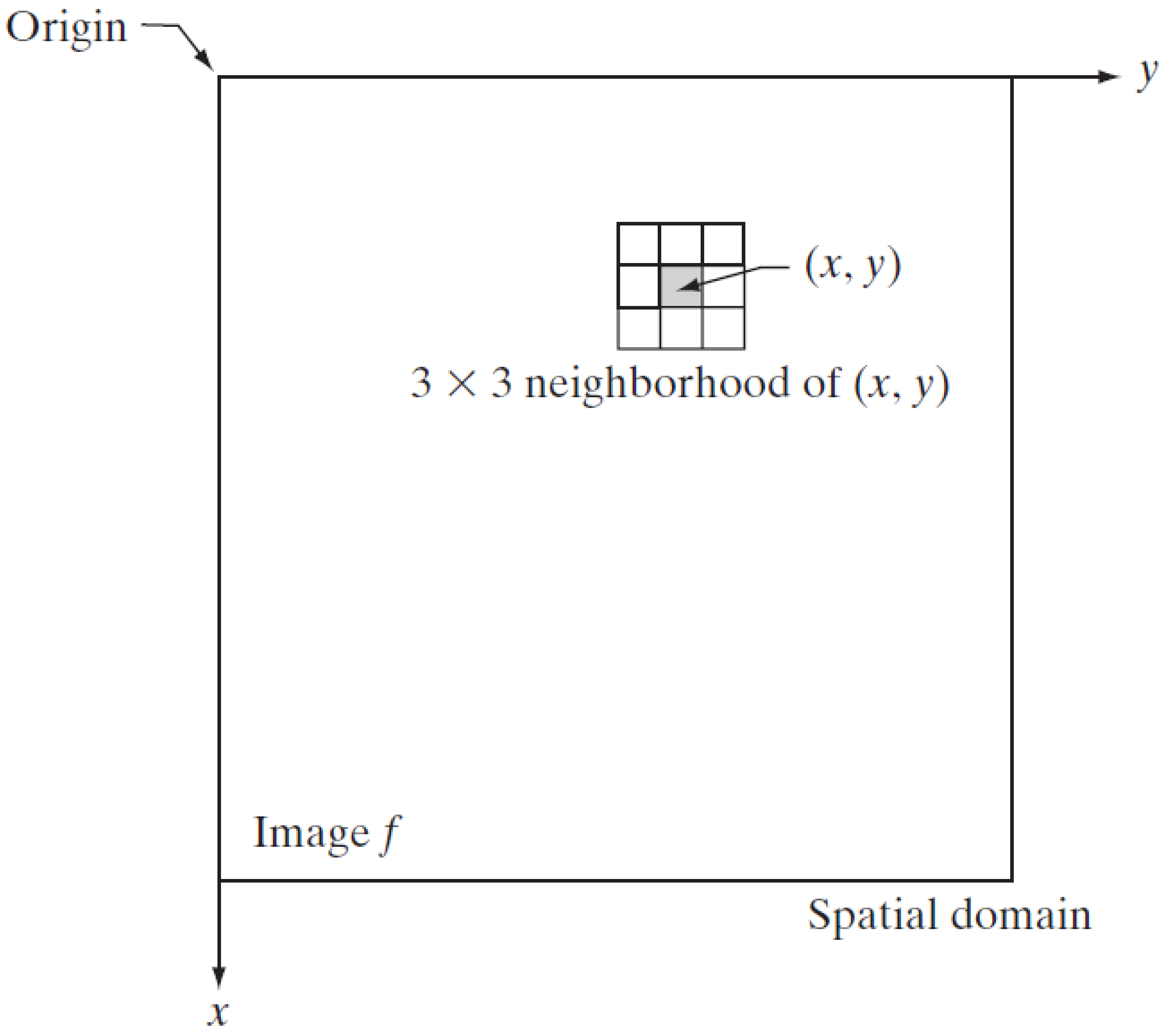
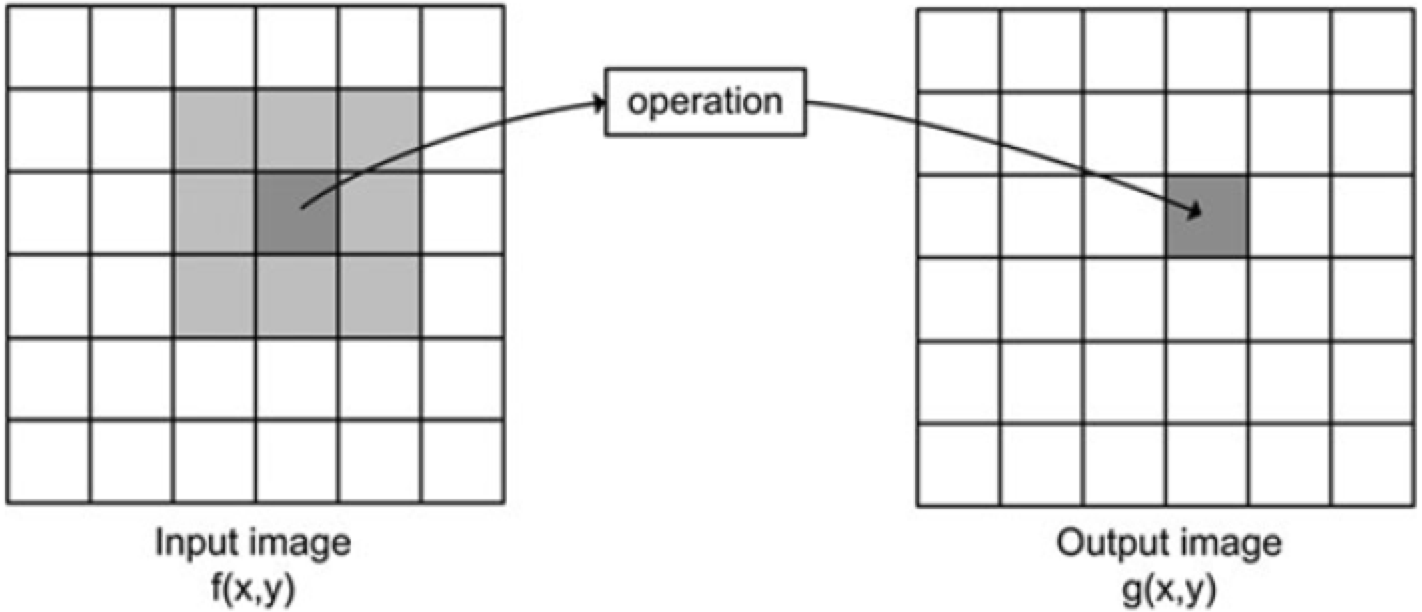
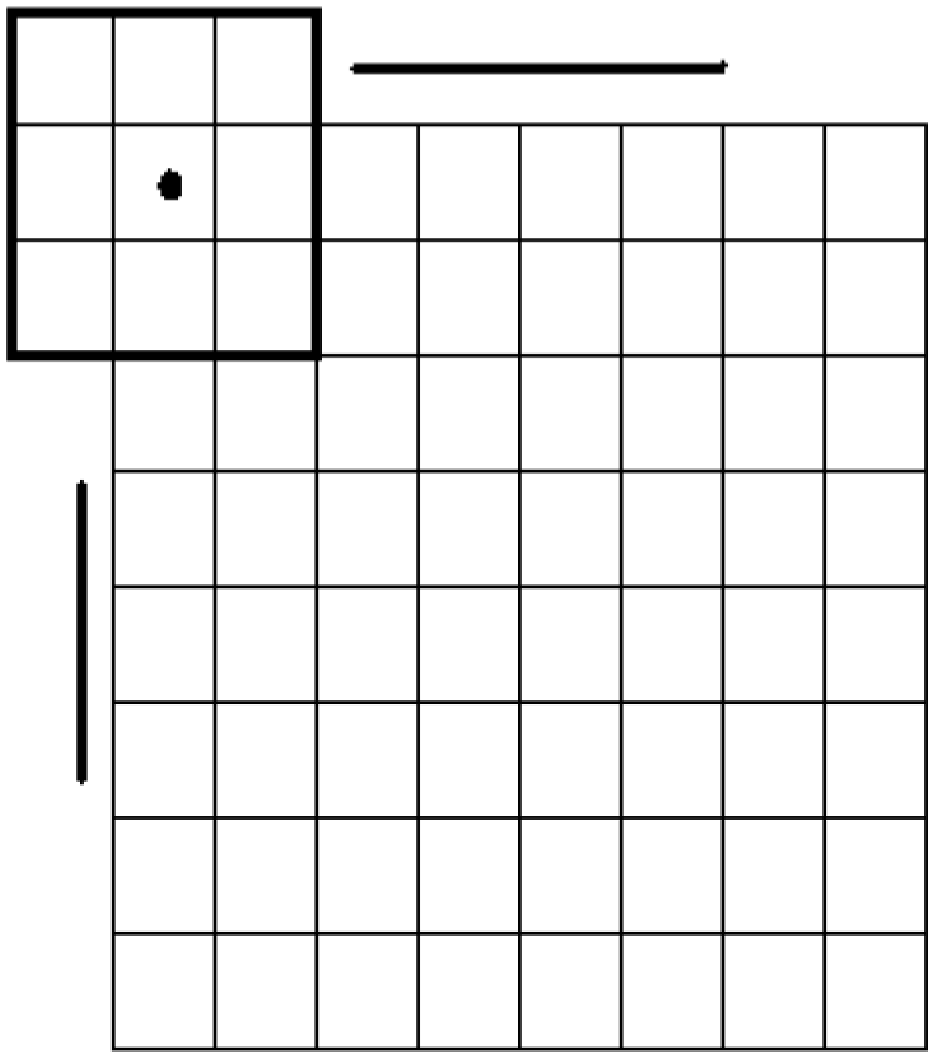


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Spatial Filtering



Averaging??



Padding?

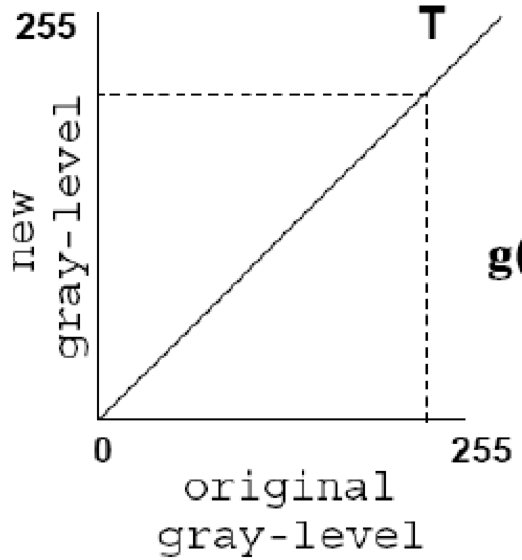
Image Enhancement:

- Image Enhancement is process of manipulating an image so that result is more suitable than the original to a *specific* application.

Basic intensity Transformation functions:

- Used for image enhancement
- Basic types:
  - Linear (identity and negative transformations)
  - Logarithmic(Log and inverse log transformations)
  - Power law(nth power and nth root transformations)

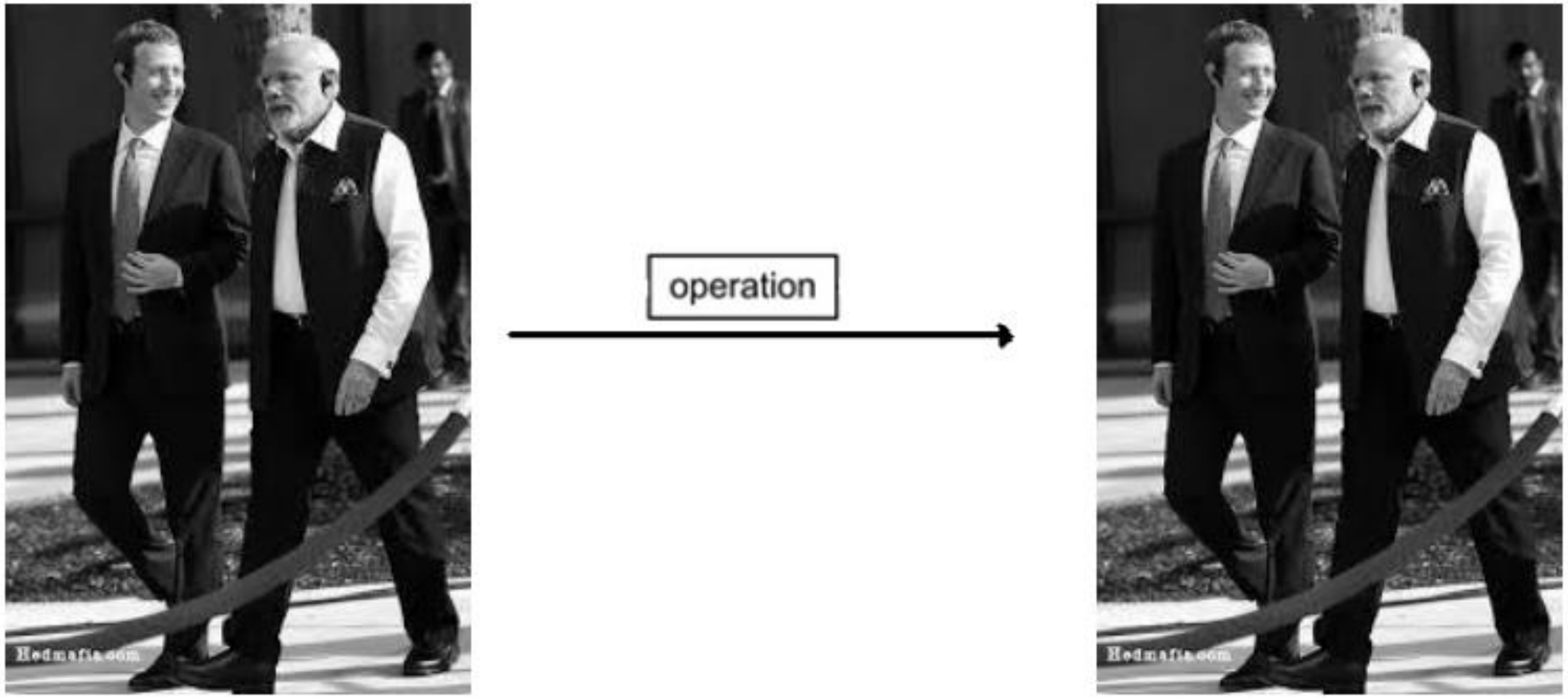
Identity transformation:



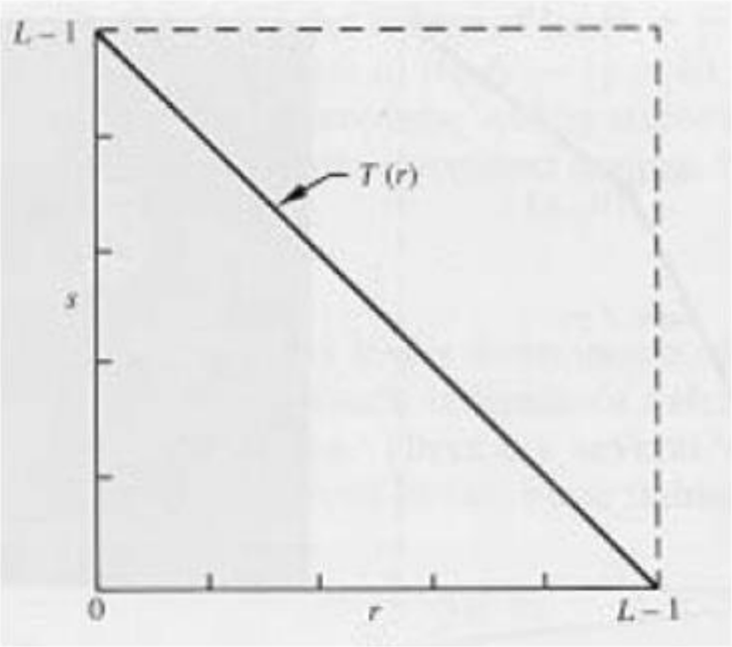
$$g(x,y) = f(x,y)$$

What is expected result??

Identity Transformation:



Negative transformation:



$$s = (L-1)-r$$

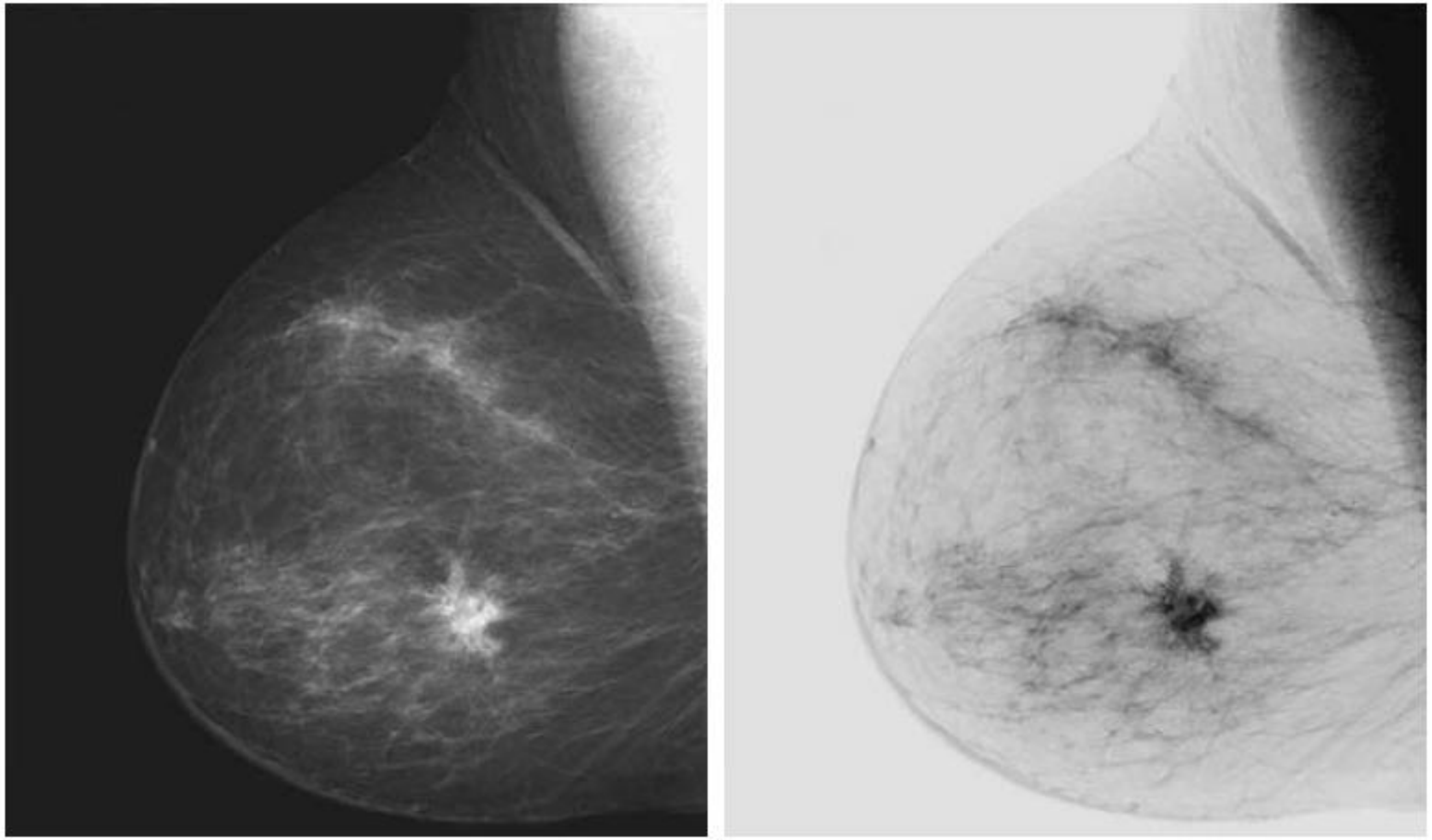
What is expected result??


Hedmafia.com


Hedmafia.com

Image negative:

- Used for enhancing white or gray detail embedded in dark regions of image.



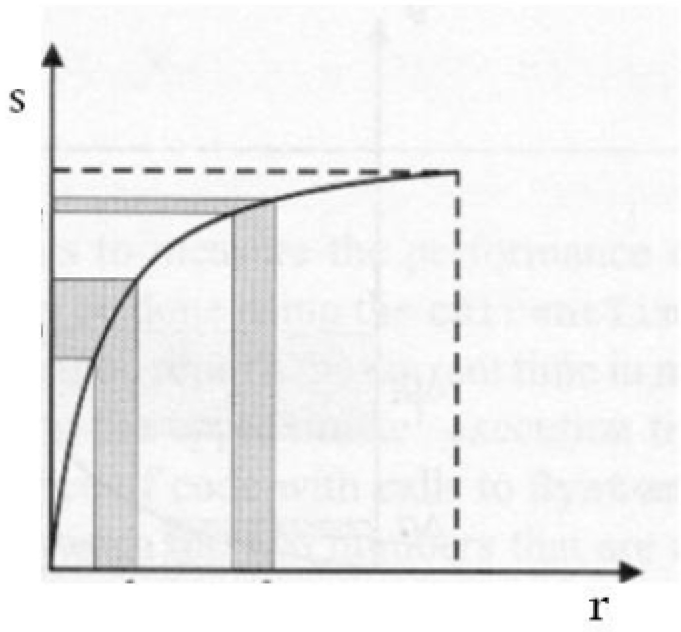
a b

FIGURE 3.4
(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).
(Courtesy of G.E. Medical Systems.)

Log Transformations:

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

where c is a constant.



Narrow range of low intensity to wider range of output.

Wider range of high intensity to narrow range of output.

Dynamic range compression.

a b

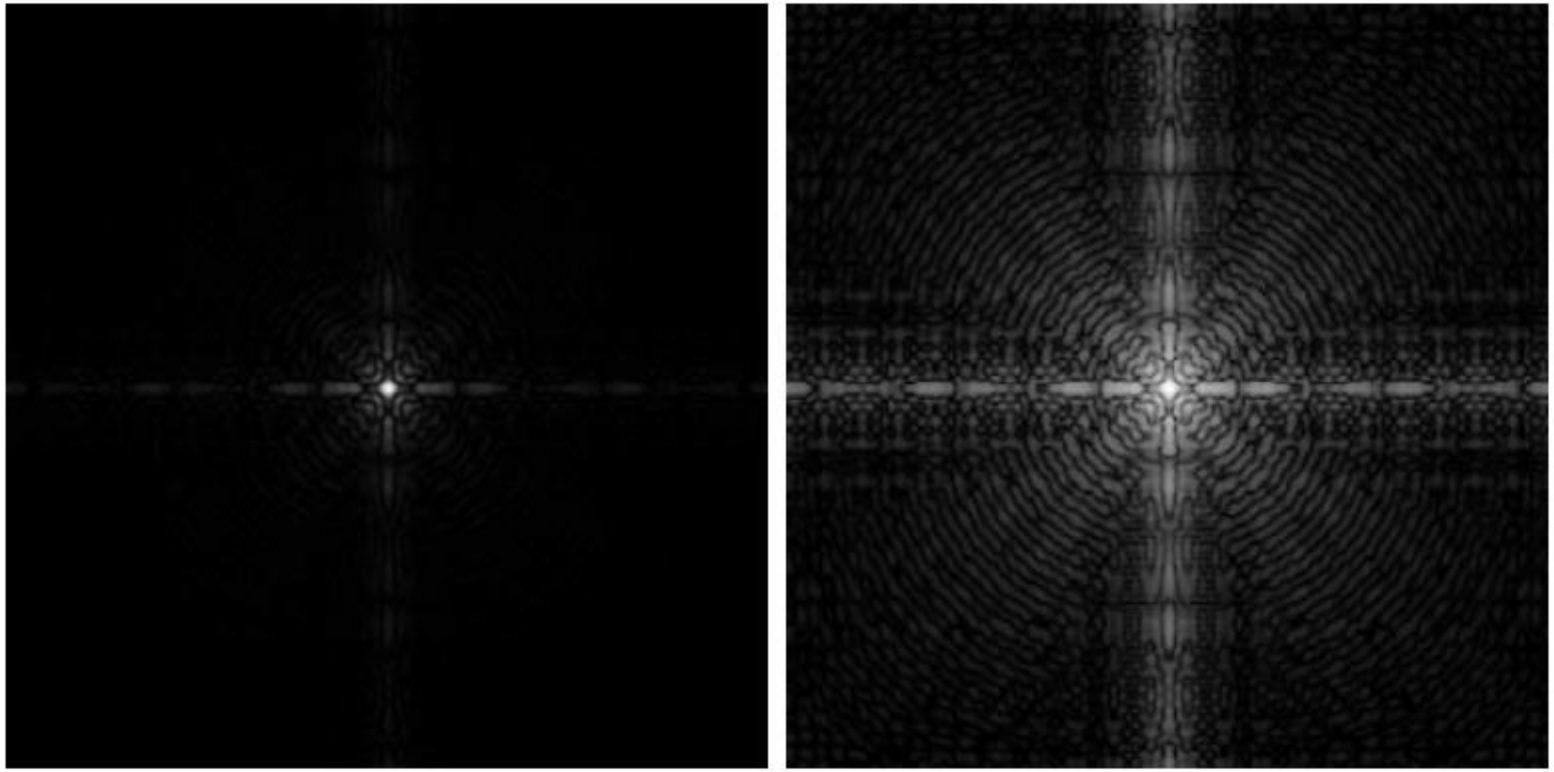
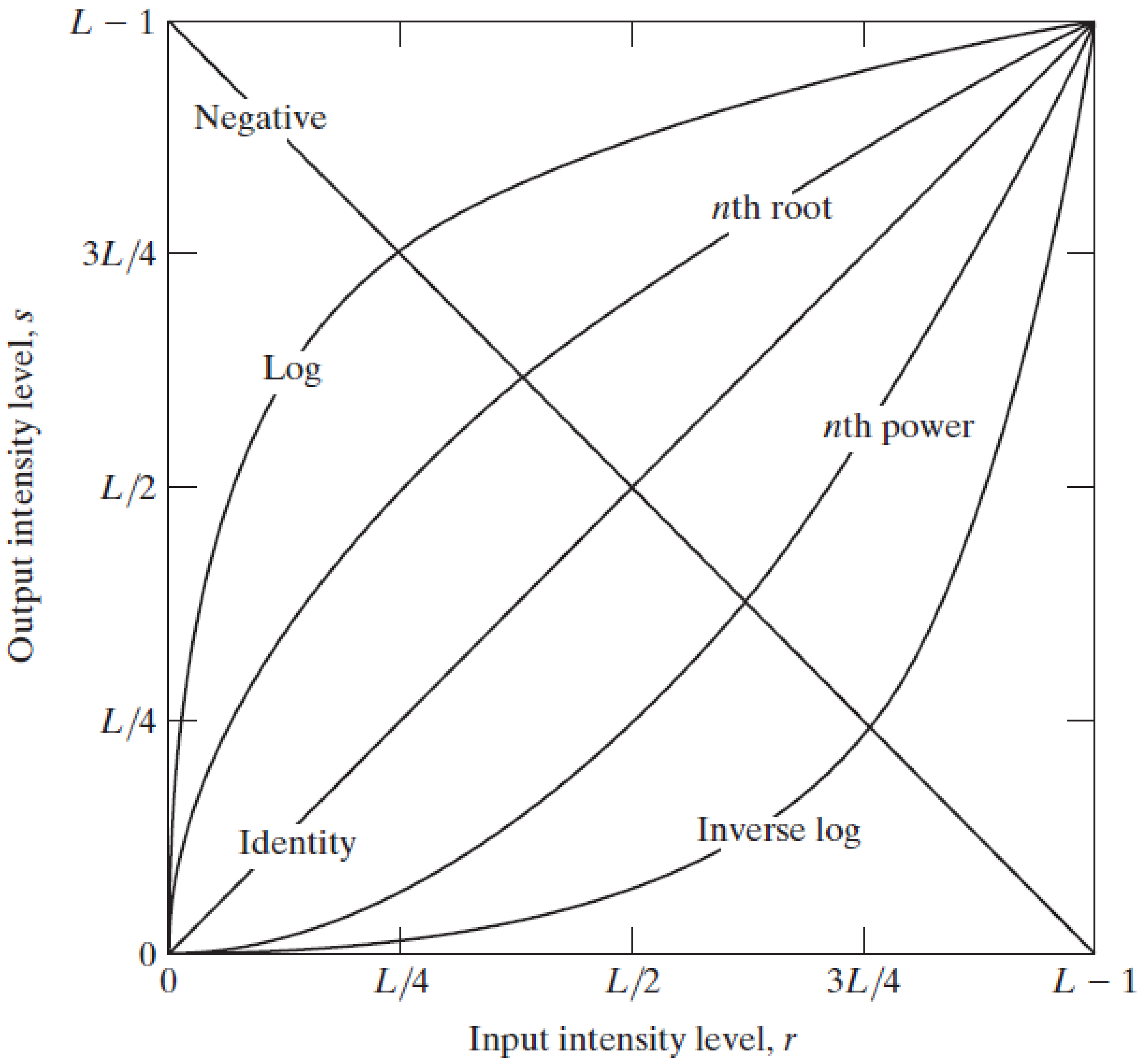


FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation in Eq. (3.2-2) with $c = 1$.

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



Power-Law Transformations:

$$s = cr^{\gamma}$$

Where c and γ are positive constant.

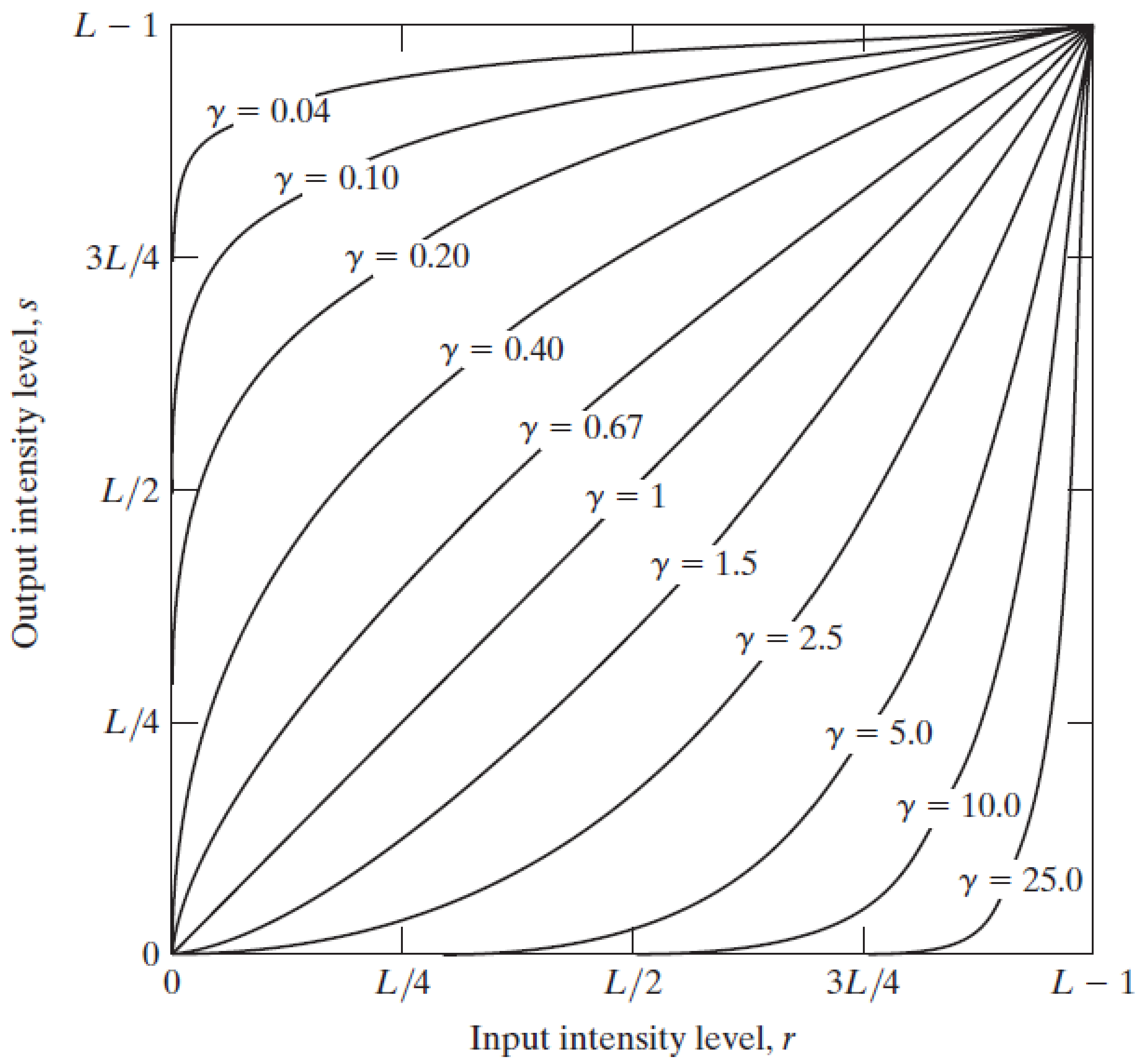


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

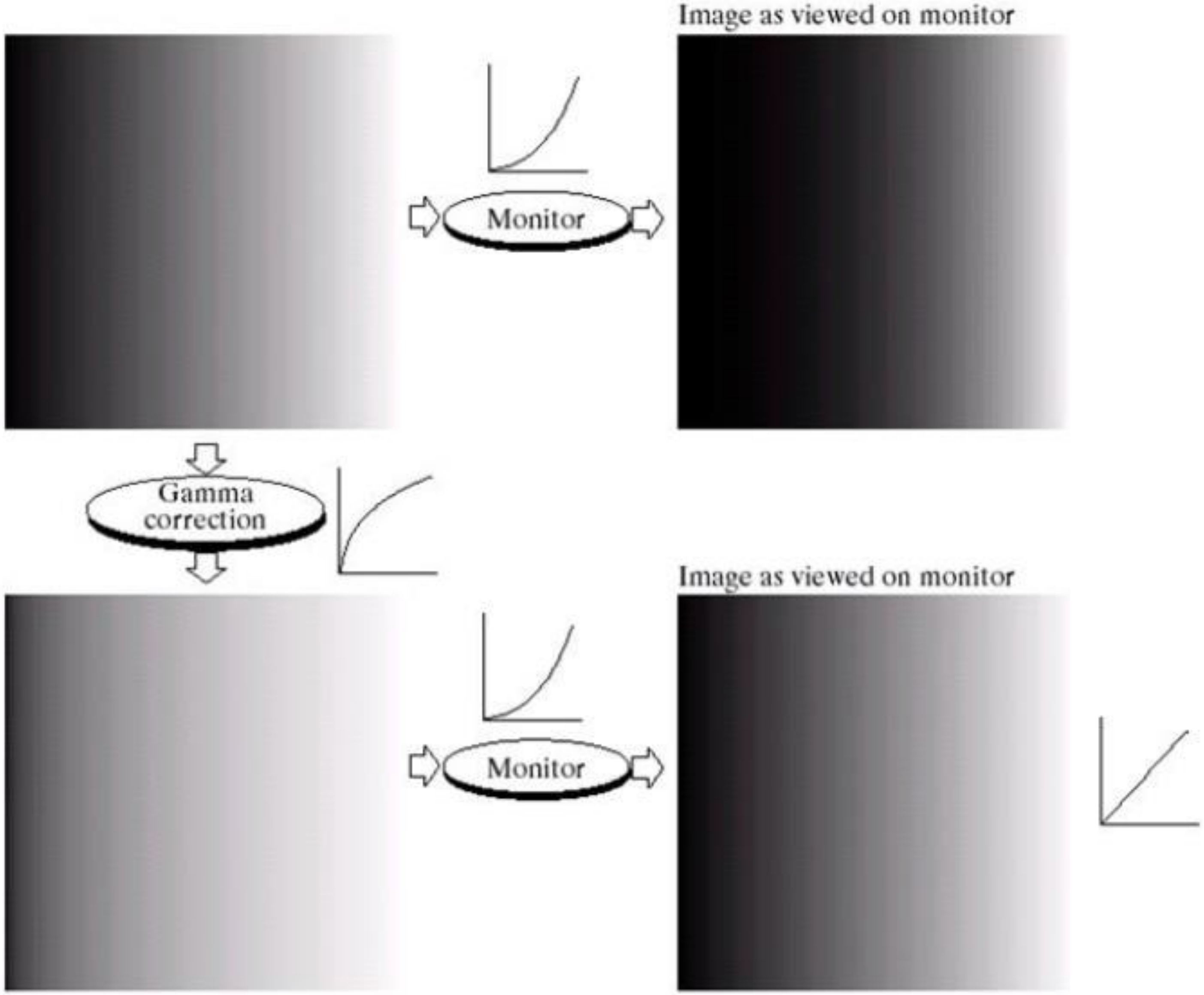
Gamma correction:

- A cathode ray tube (CRT), for example, converts a video signal to light in a nonlinear way. The light intensity *I* is proportional to a power (γ) of the source voltage *VS*
- For a computer CRT, γ is about 1.8 to 2.5
- Viewing images properly on monitors requires γ-correction

$$s = cr^{1/2.5} = cr^{0.4}$$

a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.



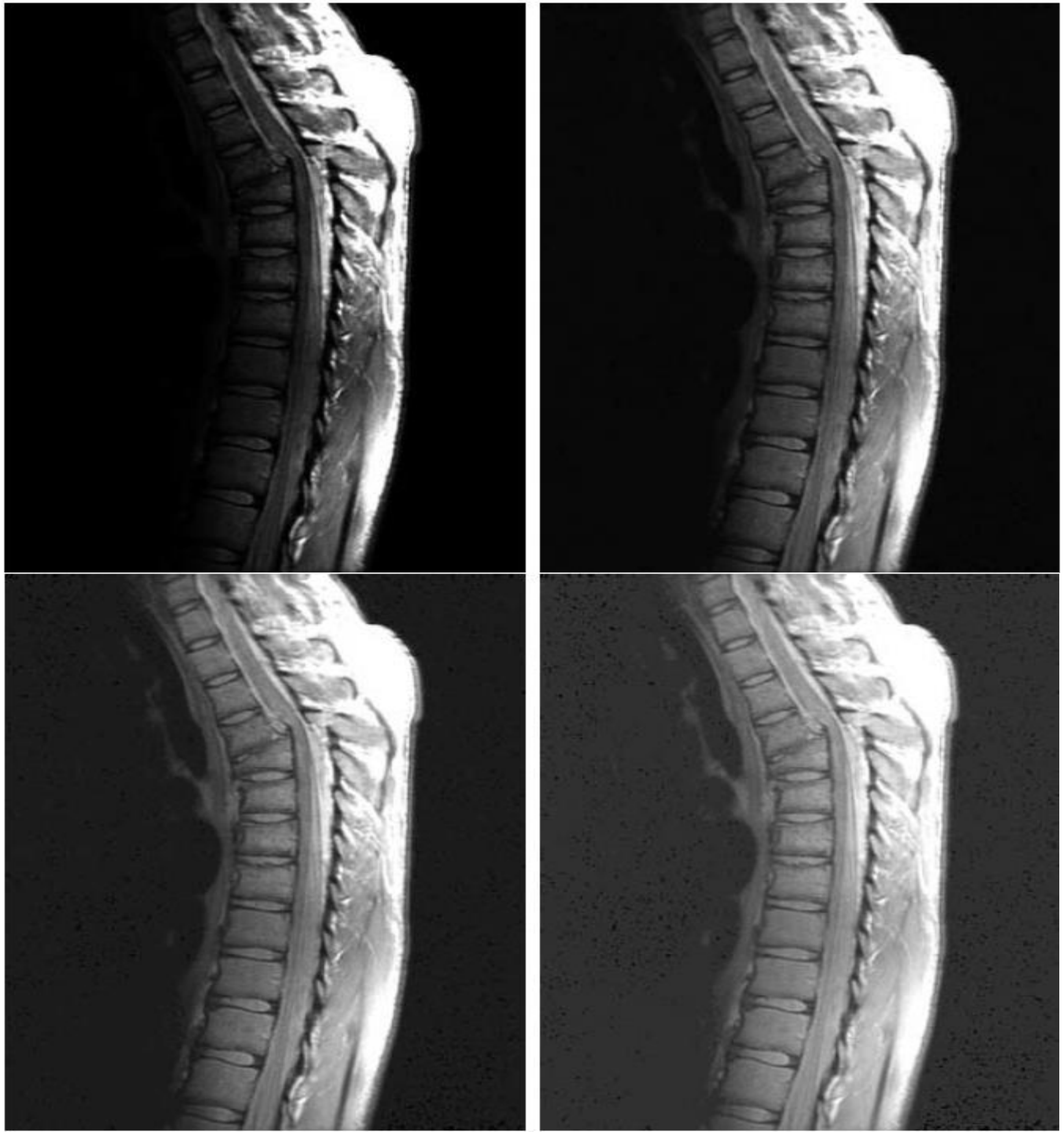


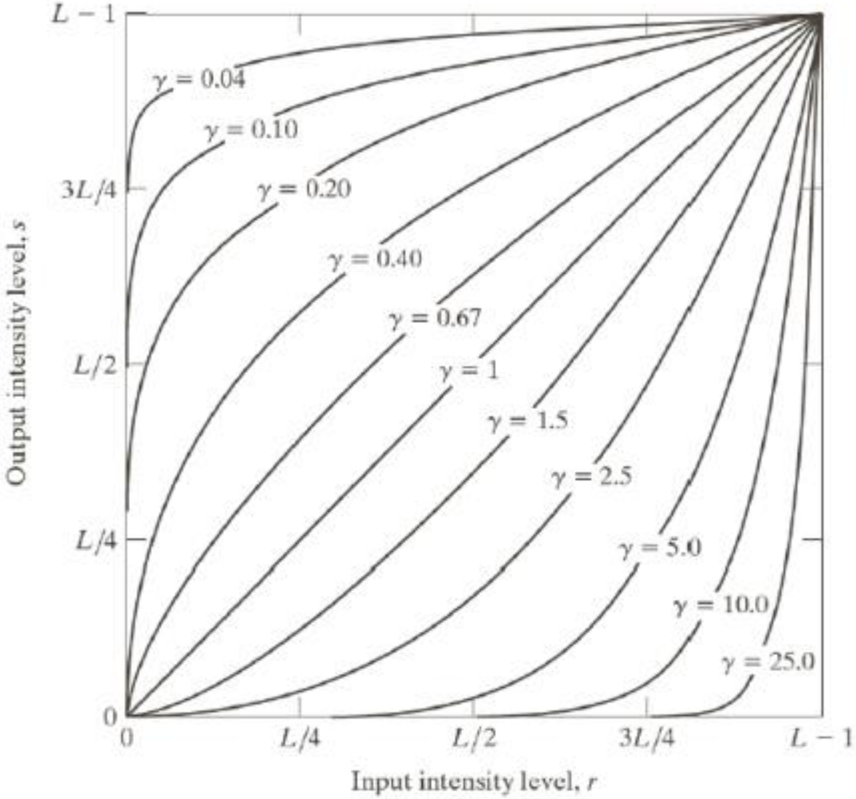
FIGURE 3.8
(a) Magnetic resonance image (MRI) of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt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

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.)

γ = ?

γ = ?

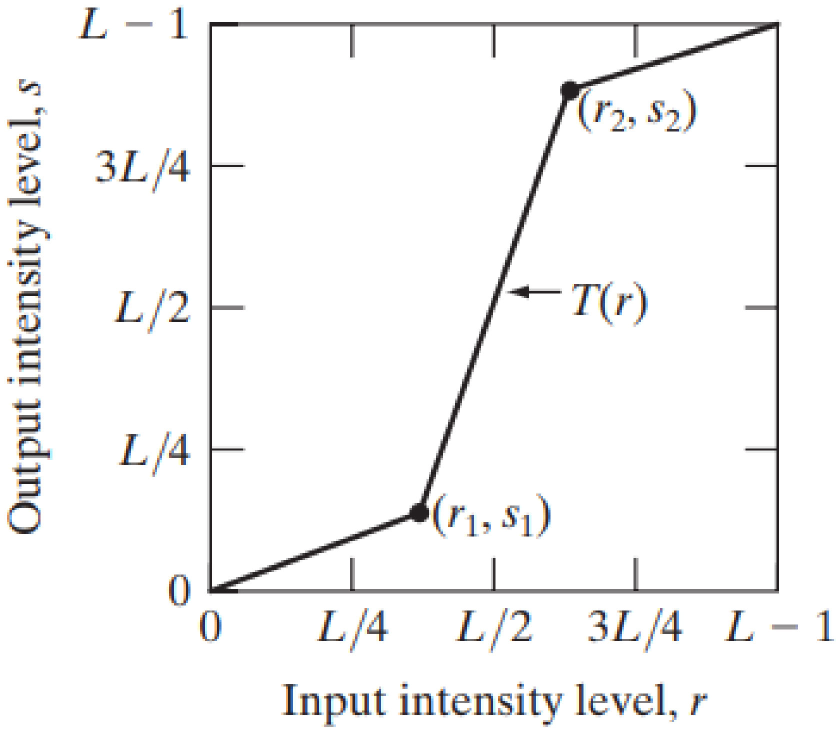


L − 1
3L/4
L/2
L/4
0
γ = 0.04
γ = 0.10
γ = 0.20
γ = 0.40
γ = 0.67
γ = 1
γ = 1.5
γ = 2.5
γ = 5.0
γ = 10.0
γ = 25.0
0
L/4
L/2
3L/4
L − 1
Output intensity level, s
Input intensity level, r

Piecewise Linear Transformation

- Contrast Stretching
- Intensity Level Slicing
- Bit Plane Slicing

Contrast Stretching



- *(r1, s1)* and *(r2, s2)* control the shape of transformation.
- *What if r1 = r2, s1 = 0 and s2 = L -1??*
- It's Thresholding Function.
- What if *(r1, s1)* = *(r_{min} , 0)* and *(r2, s2)* = *(r_{max} , L - 1) ??*
- It's Contrast Stretching.

Contrast stretching is a process that expands the range of intensity levels in an image so that it spans the available full intensity range.

a b
c d

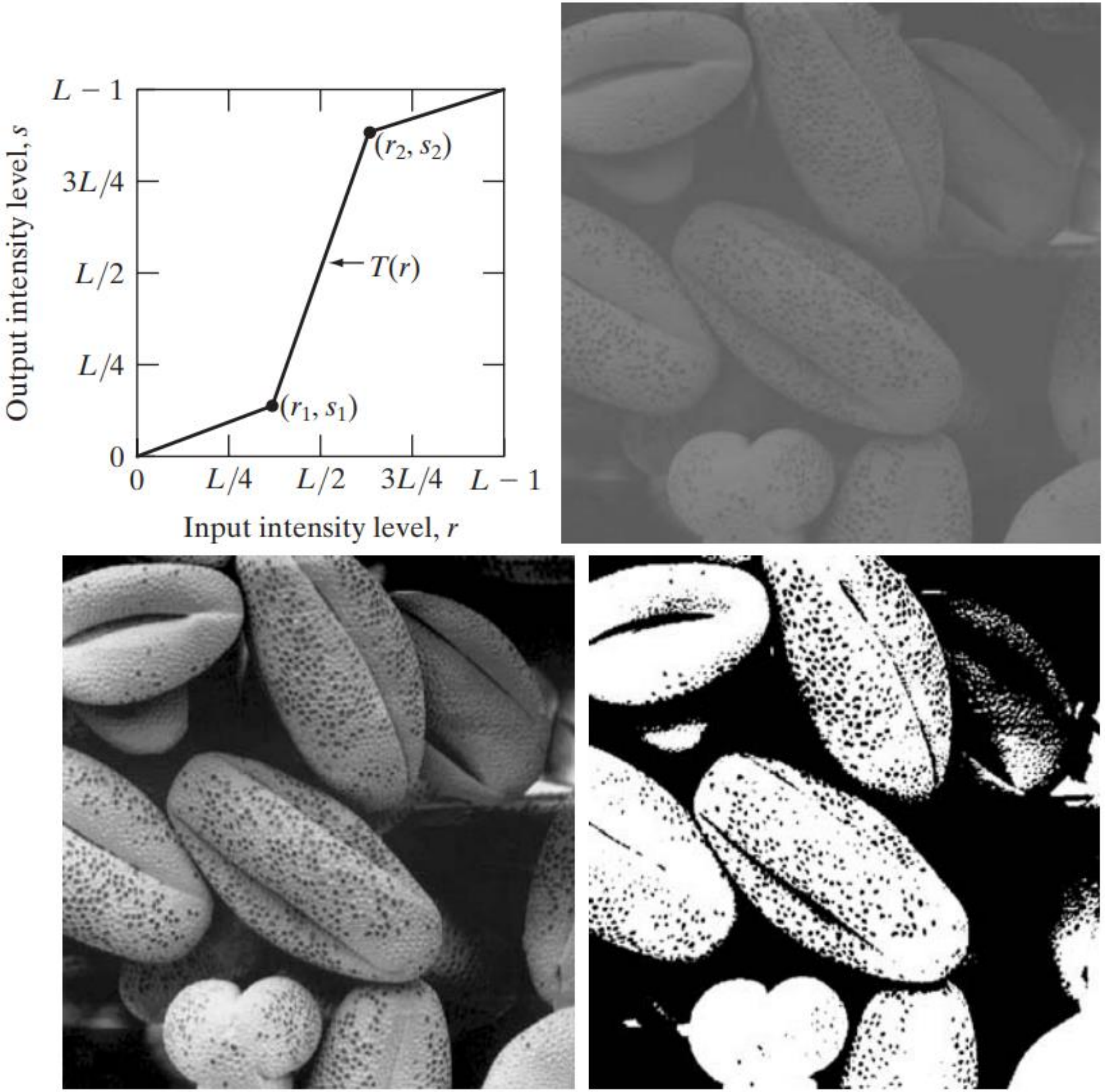


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

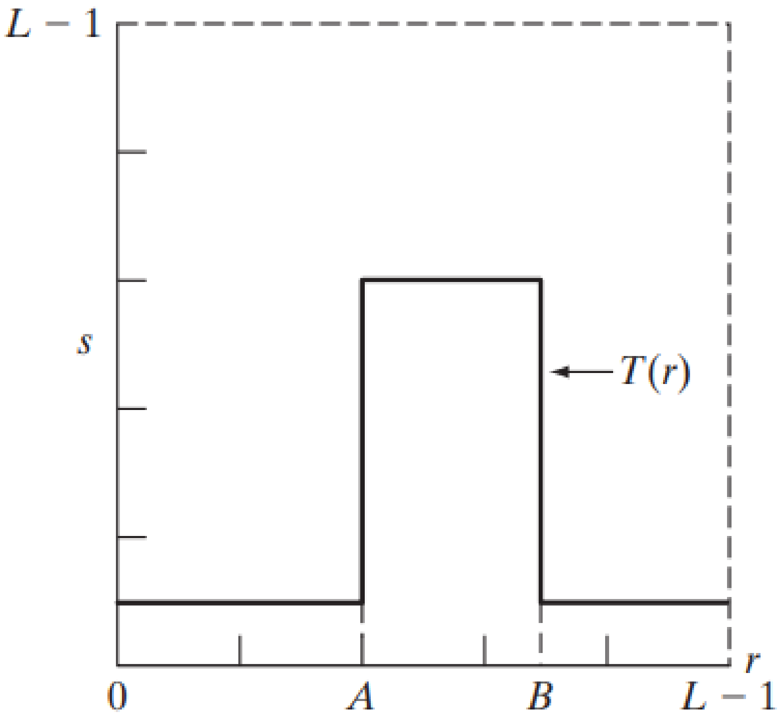
Intensity Level Slicing

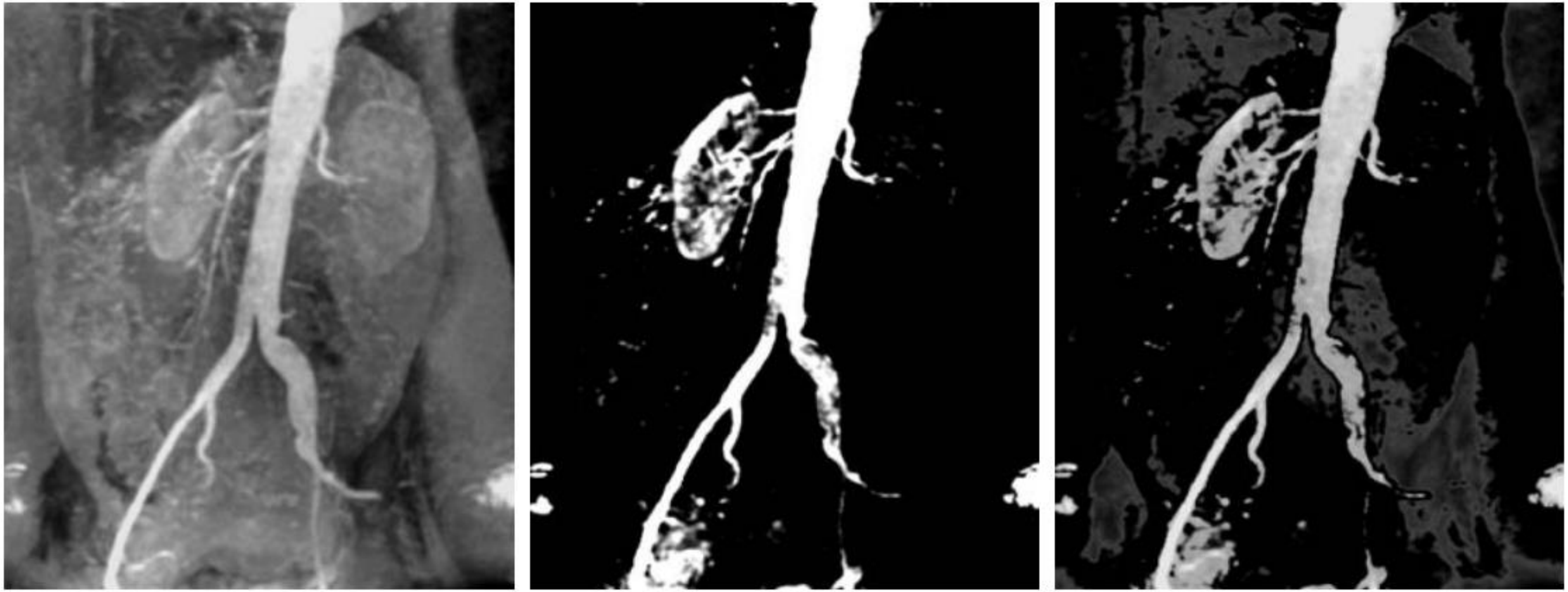
- Highlighting a specific range of intensities in an image.
- Two basic themes:

1. All the values of interest ➔ one intensity value(say white)
   All other values ➔ another intensity value(say black).
2. All the values of interest are brighten or darken
   keeping all other intensity levels unchanged.

a b

FIGURE 3.11 (a) This transformation highlights intensity range $[A, B]$ and reduces all other intensities to a lower level. (b) This transformation highlights range $[A, B]$ and preserves all other intensity levels.





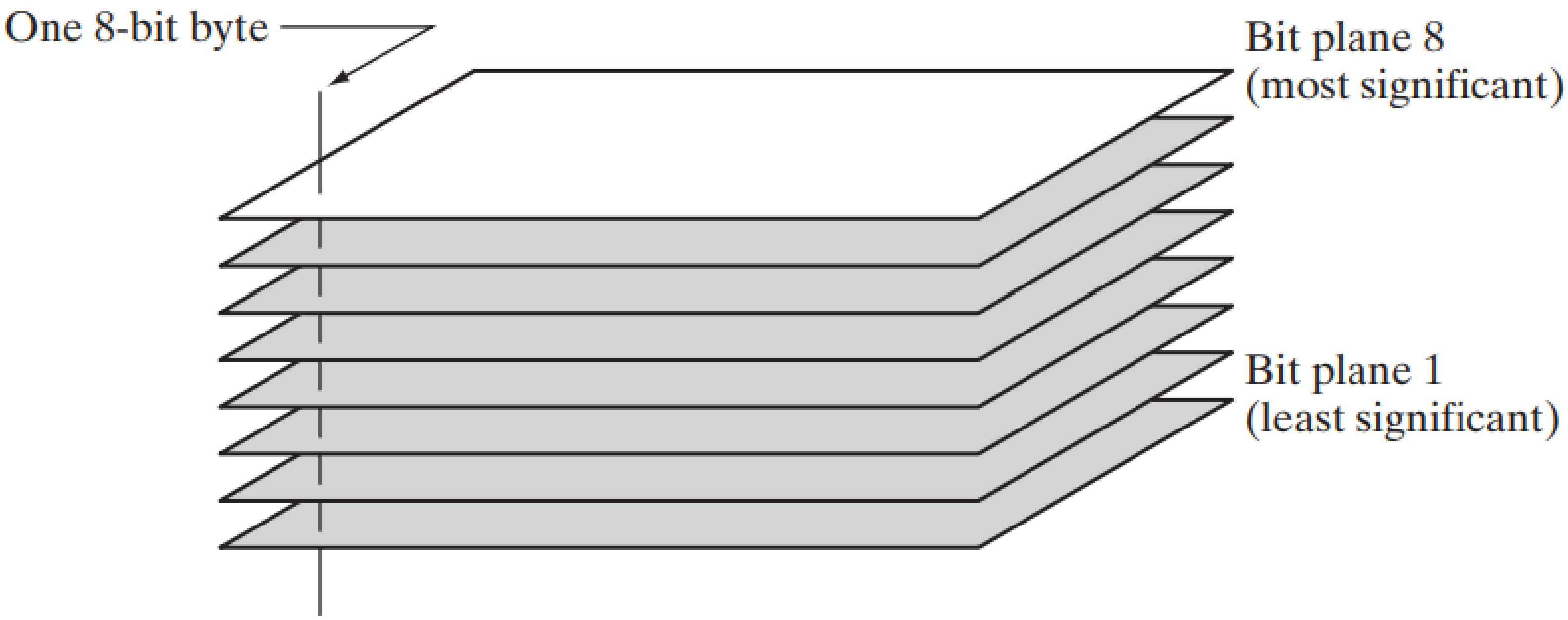
a b c

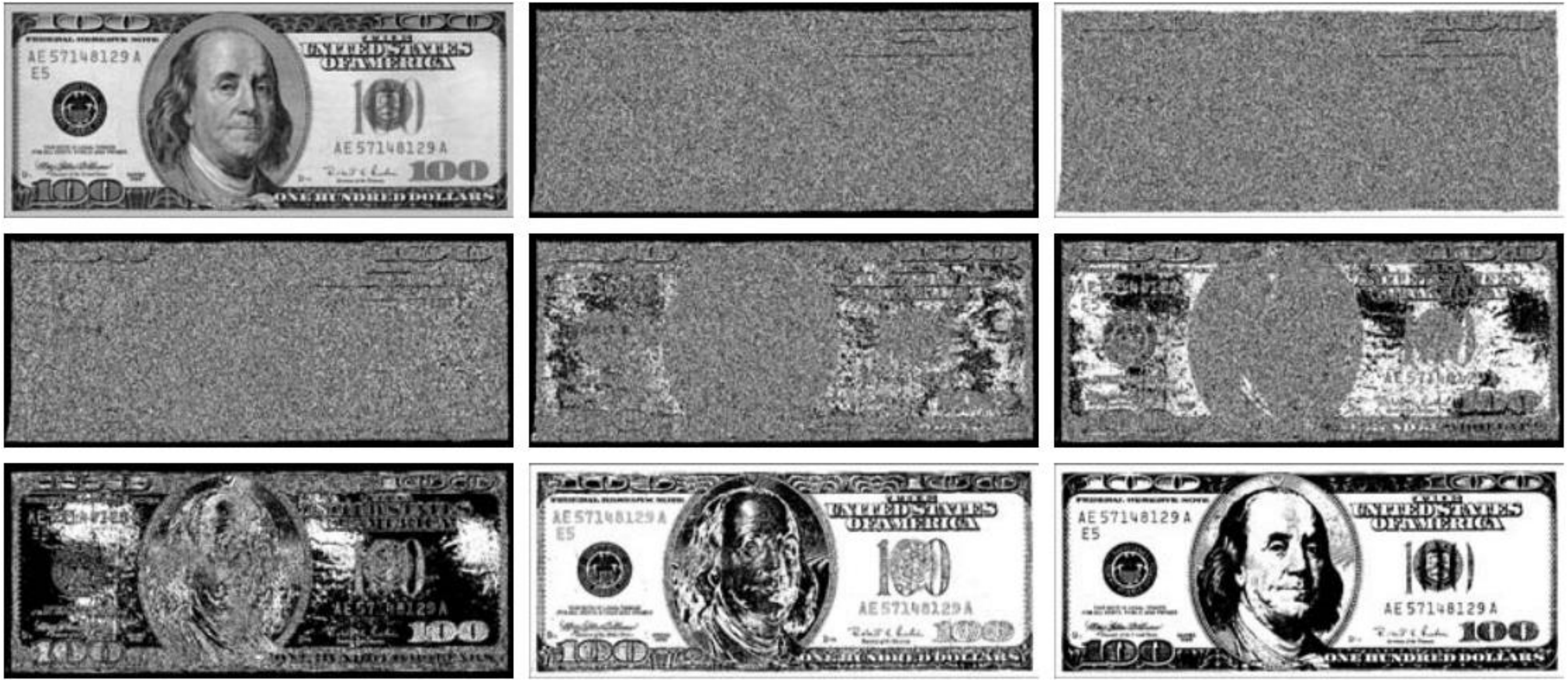
FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so 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.)

Bit – Plane Slicing

- 256 intensity levels → 8 bits to represent a pixel.
- It highlights the contribution made to the total image appearance by specific

FIGURE 3.13 Bit-plane representation of an 8-bit image.





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.

Bit Plane Slicing

- How to obtain 8-bit plane?
- Set all intensity values from 0-127 to 0 and 128 to 255 to 1.
- How to obtain others?
- 7 bit plane: 64-127 and 192-255 ➔1 o/w 0
- Border value =(194)d=(11000010)b

4	3	2	1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Original Image

Bit Planes 8 & 7

Bit Planes 8, 7 & 6

Bit Planes 8, 7, 6 & 5

Reconstruction:

- Nth plane bit multiplied by $2^{(n-1)}$ constant.
- Addition of all such planes.
- Bit plane slicing is useful in compression and storage

Image Histogram

- A histogram is a graph that shows frequency of the desired element. Usually histogram have bars that represent frequency of occurring data in the whole data set.
- A histogram has two axis, the x axis contains the event whose frequency we have to count and y contains the respective frequency.
- An image histogram is a plot of the gray- level frequencies. (i.e., the number of pixels in the image that have a particular gray level.)

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

frequencies

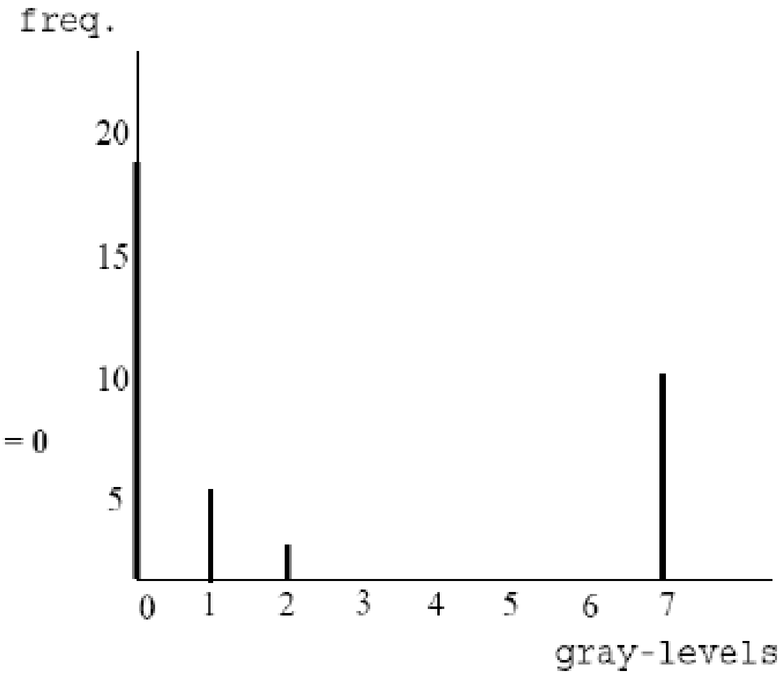
f(0)= 18

f(1)= 6

f(2)= 2

f(3)=f(4)=f(5)=f(6) = 0

f(7) = 10



- Divide frequencies by total number of pixels (m x n image size)to represent as probabilities.

$$p_k = n_k / N = n_k / mn$$

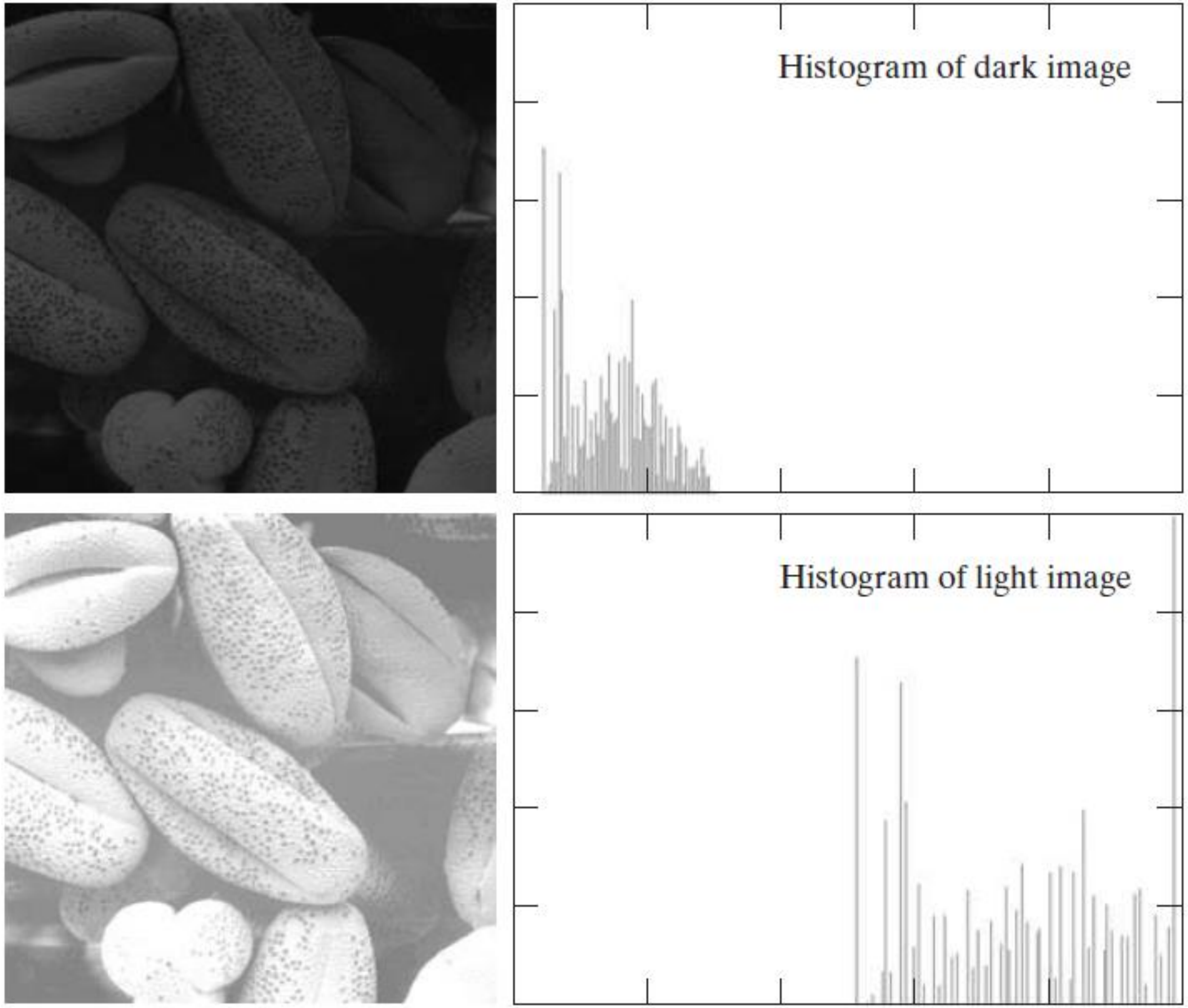
$$P(0) = \frac{f(0)}{36} = \frac{1}{2} \qquad P(1) = \frac{f(1)}{36} = \frac{1}{6}$$

$$P(2) = \frac{f(2)}{36} = \frac{1}{18} \qquad P(3) = P(4) = P(5) = P(6) = 0$$

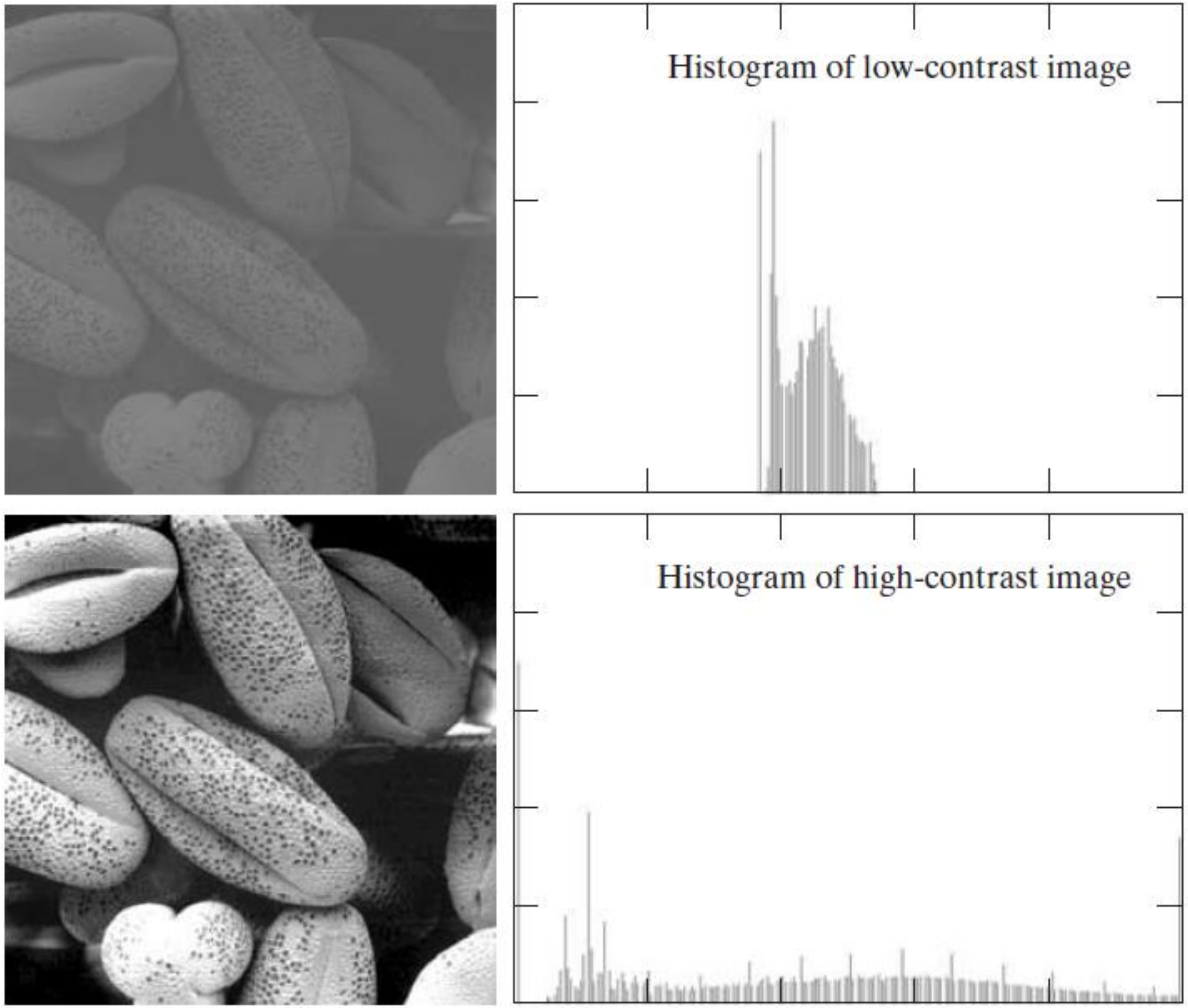
$$P(7) = \frac{f(7)}{36} = \frac{5}{18}$$

Sum of all these probabilities?

Plot of these probabilities is probability density function of input image pr(r)



x-axis – values of intensities
y-axis – their occurrence



x-axis – values of intensities
y-axis – their occurrence

Histogram Processing

- Histogram manipulation can be used for image enhancement.
- Information inherent in histogram is also useful in other image processing applications, such as image compression and segmentation.
- Histogram Equalization:
- To improve the contrast of an image.
- To transform an image in such a way that the transformed image has nearly *uniform distribution* of pixel values.

Histogram Equalization

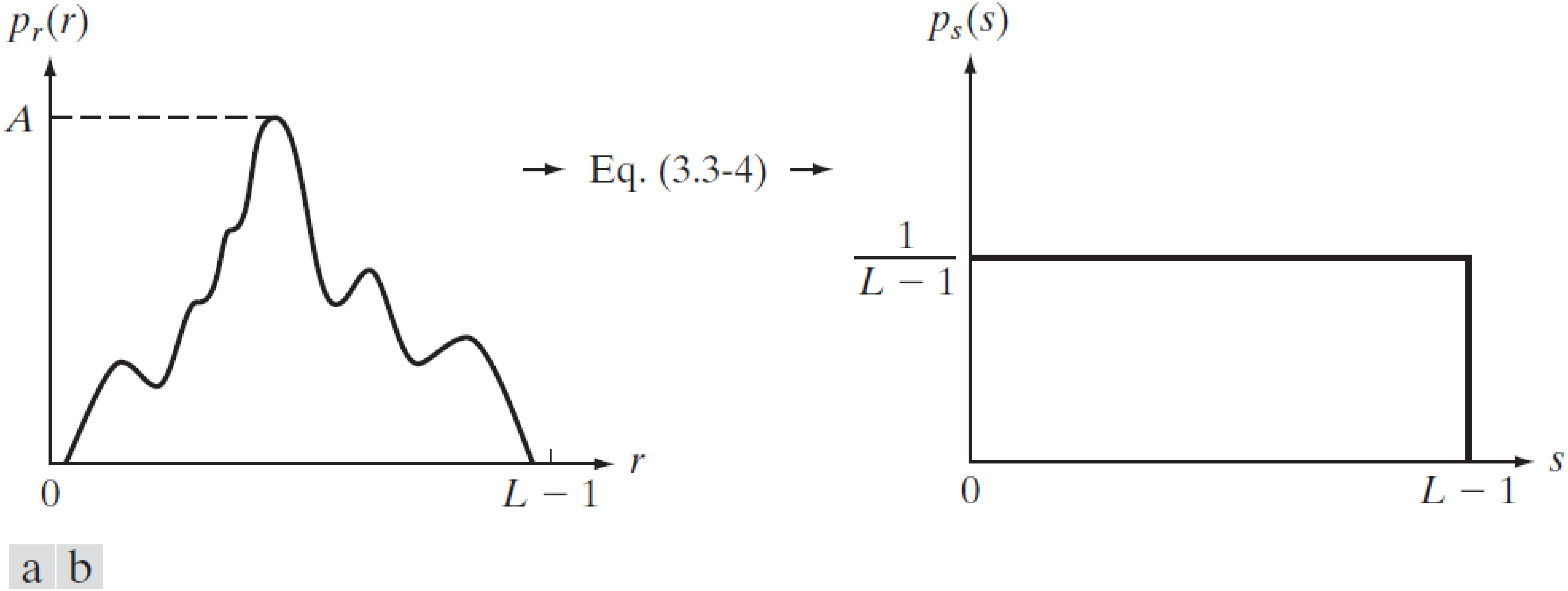


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Histogram Equalization

- Intensity Mapping form

$$s = T(r) \quad 0 \le r \le L - 1 \qquad (3.3\text{-}1)$$

- Conditions:

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

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

- In some formulations to be discussed later, we use the inverse

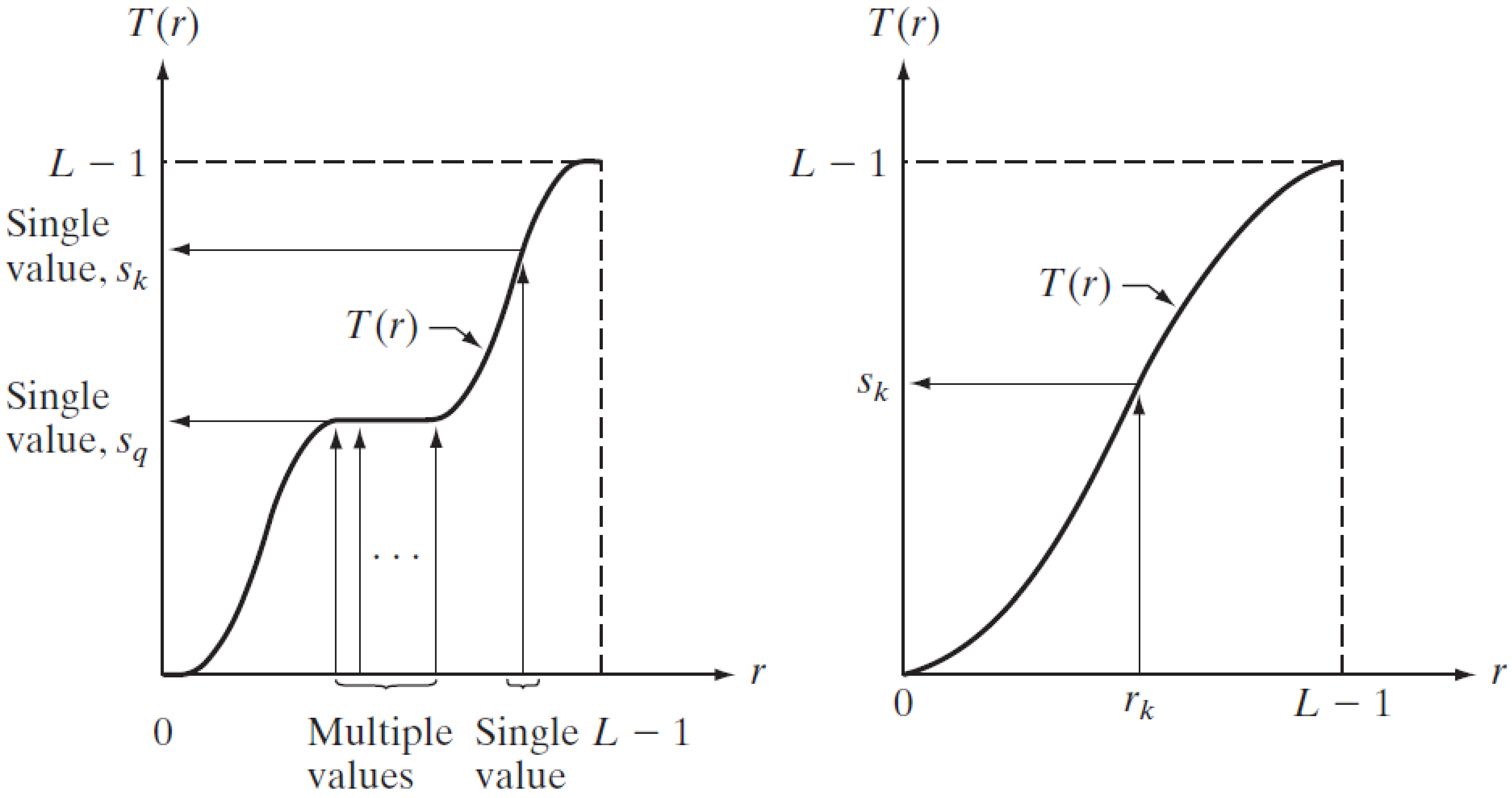
$$r = T^{-1}(s) \quad 0 \le s \le L - 1 \qquad (3.3\text{-}2)$$

in which case we change condition (a) to

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

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.



- A fundamental descriptor of a random variable is its PDF.
- Let $p_r(r)$ and $p_s(s)$ denote the PDFs of *r* and *s*, respectively.
- A fundamental result from basic probability theory is that if $p_r(r)$ and *T(r)* are known, and *T(r)* is continuous and differentiable over the range of values of interest, then PDF of the transformed (mapped) variable *s* can be obtained using the simple formula

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \qquad (3.3\text{-}3)$$

- Consider the following transform function :

$$s = T(r) = (L - 1)\int_0^r p_r(w)\, dw \qquad (3.3\text{-}4)$$

Where *w* is a dummy variable of integration and the right side of the equation is CDF of random variable *r.*

- *Does it satisfy condition a) and b)?*

- To find the $p_s(s)$ corresponding to the transformation function just discussed we use eq. (3).

We know from Leibniz's rule that the derivative of a definite integral w.r.t its upper limit is simply the integrand evaluated at that limit.

$$\frac{ds}{dr} = \frac{dT(r)}{dr}$$

$$= (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] \tag{3.3-5}$$

$$= (L-1)p_r(r)$$

Substituting this result for dr/ds in Eq. (3.3-3), and keeping in mind that all probability values are positive, yields

$$\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \\ &= p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| \\ &= \frac{1}{L-1} \qquad 0 \leq s \leq L-1 \end{aligned} \tag{3.3-6}$$

Example

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2} & \text{for } 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

From Eq. (3.3-4),

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$

$$\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| = \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right| \\ &= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right| \\ &= \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1} \end{aligned}$$

- For discreet values,

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0, 1, 2, \ldots, L - 1 \tag{3.3-7}$$

The discrete form of the transformation in Eq. (3.3-4) is

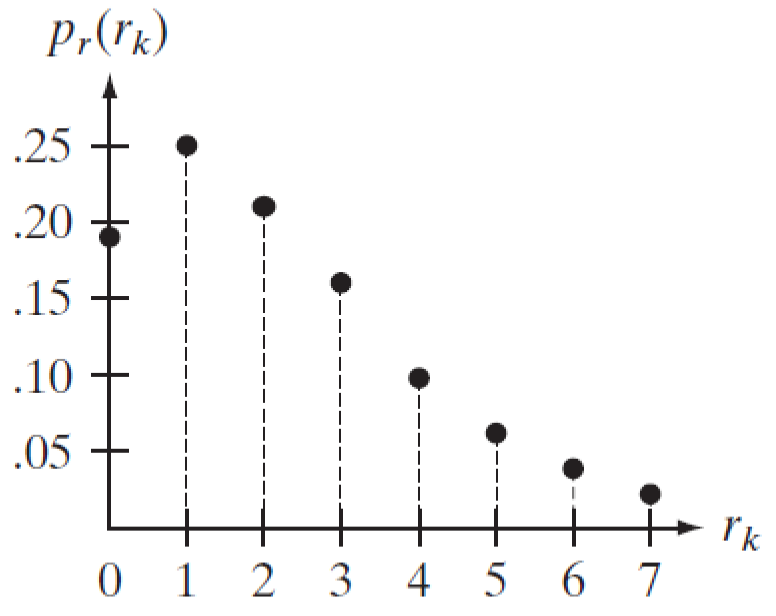
$$\begin{aligned} s_k = T(r_k) &= (L - 1)\sum_{j=0}^{k} p_r(r_j) \\ &= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, \ldots, L - 1 \end{aligned} \tag{3.3-8}$$

Thus, a processed (output) image is obtained by mapping each pixel in the input image with intensity r_k into a corresponding pixel with level s_k in the output image, using Eq. (3.3-8). The transformation (mapping) $T(r_k)$ in this equation is called a *histogram equalization* or *histogram linearization* transformation. It is not difficult to show (Problem 3.10) that this transformation satisfies conditions (a) and (b) stated previously in this section.

Histogram Equalization Numerical

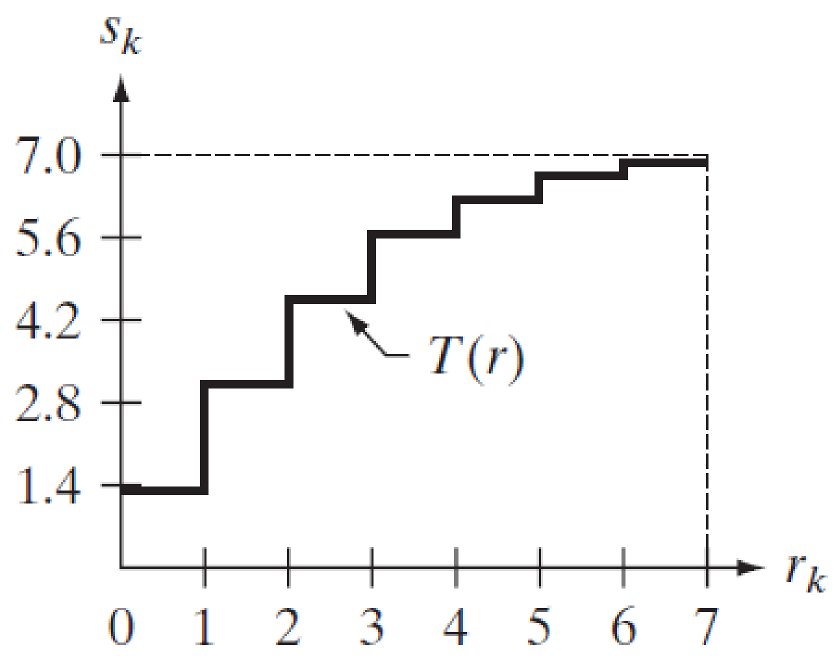
Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution shown in Table 3.1, where the intensity levels are integers in the range $[0, L - 1] = [0, 7]$.

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



$$s_k = T(r_k) = (L - 1)\sum_{j=0}^{k} p_r(r_j)$$

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$



Similarly,

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

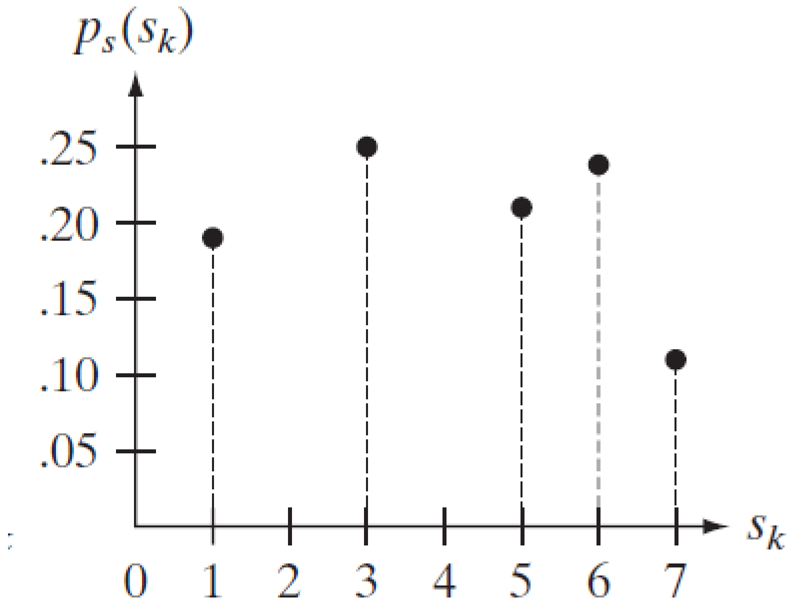
and $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, $s_7 = 7.00$.

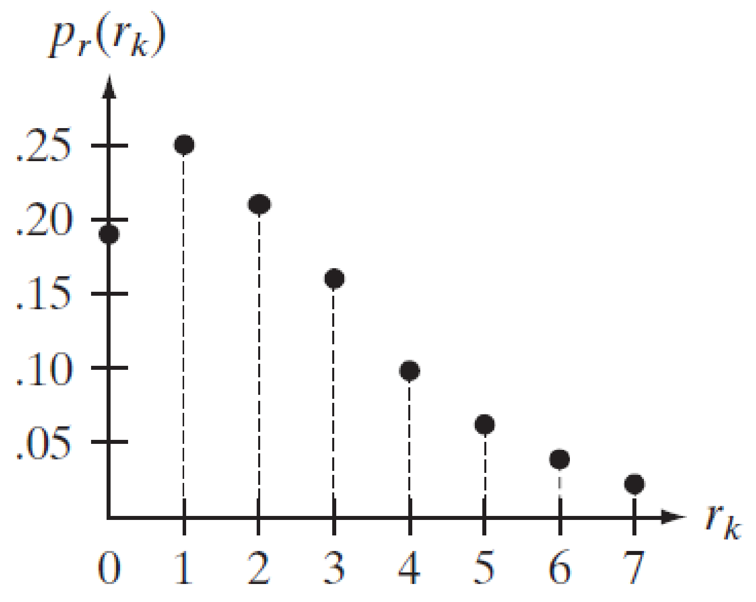
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

At this point, the s values still have fractions because they were generated by summing probability values, so we round them to the nearest integer:

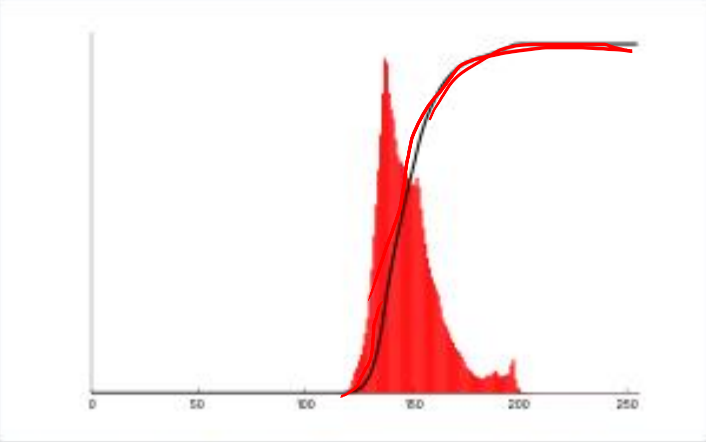
$$s_0 = 1.33 \rightarrow 1 \qquad s_4 = 6.23 \rightarrow 6$$
$$s_1 = 3.08 \rightarrow 3 \qquad s_5 = 6.65 \rightarrow 7$$
$$s_2 = 4.55 \rightarrow 5 \qquad s_6 = 6.86 \rightarrow 7$$
$$s_3 = 5.67 \rightarrow 6 \qquad s_7 = 7.00 \rightarrow 7$$

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



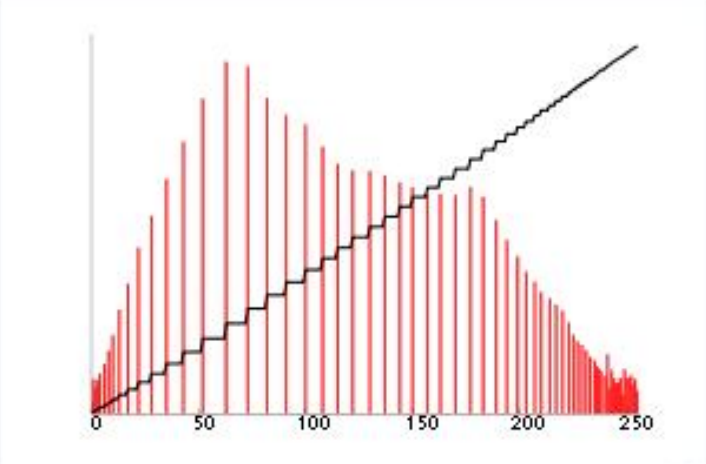


Before Histogram Equalization

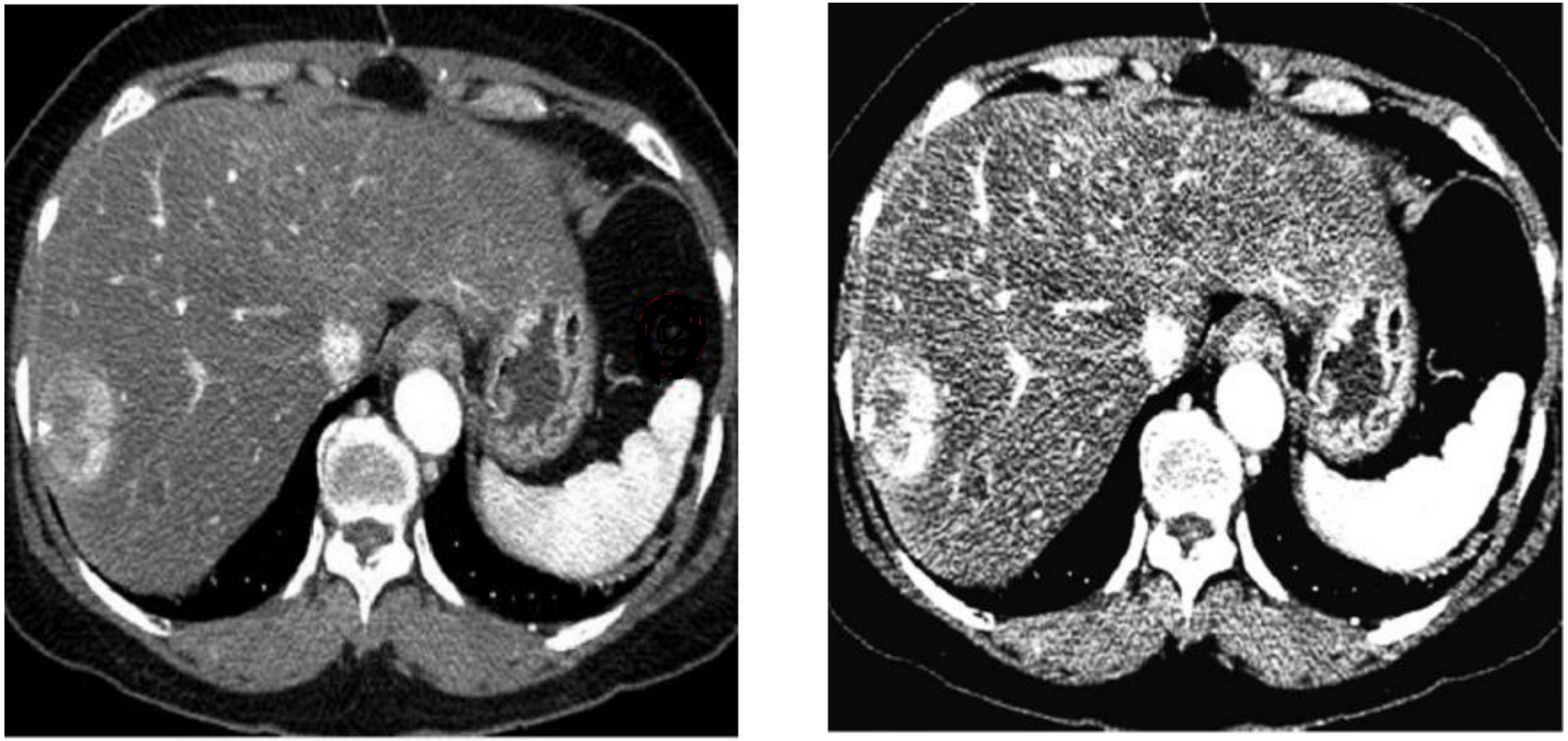


Corresponding histogram (red) and cumulative histogram (black)

After Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)



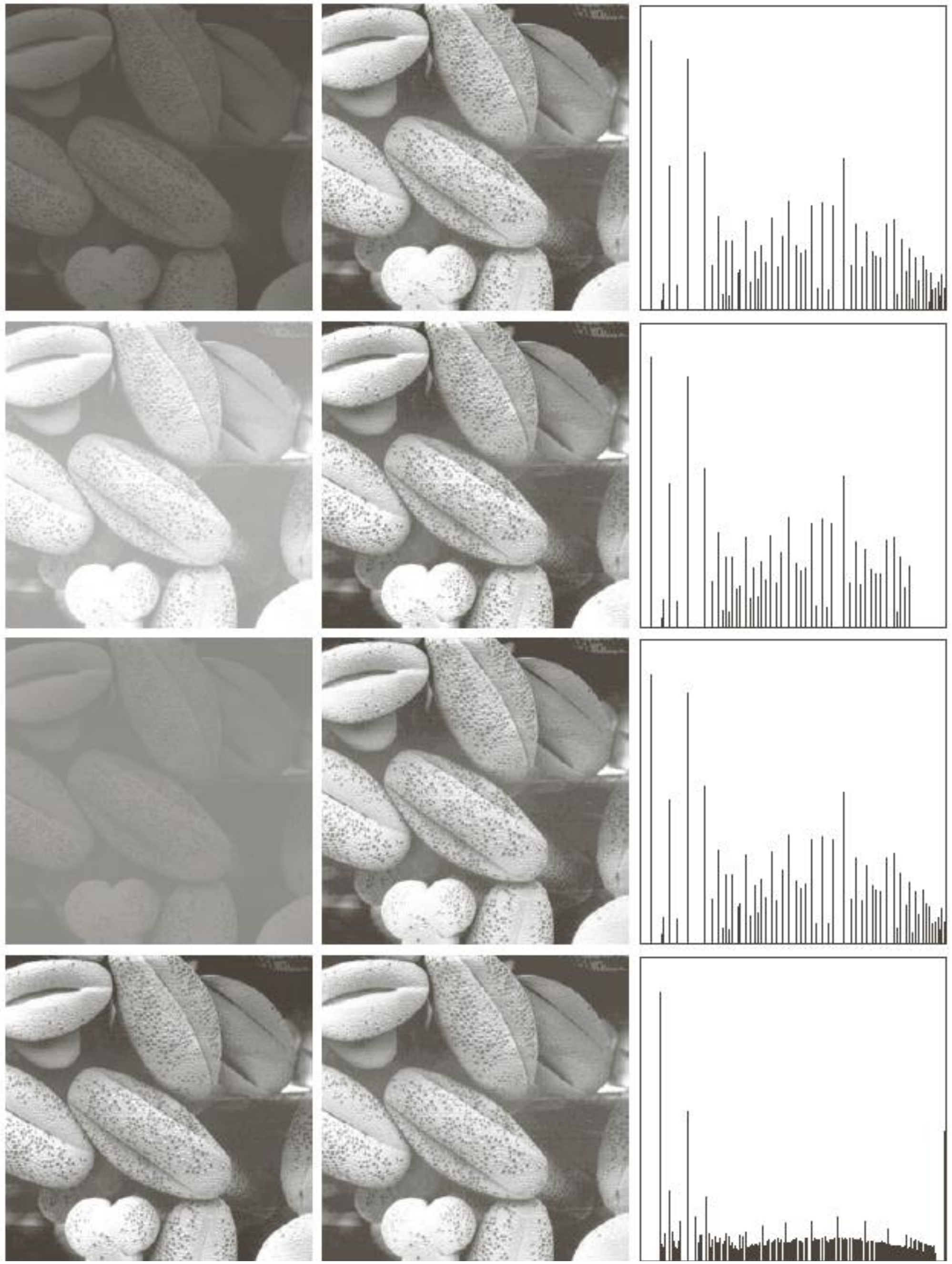


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

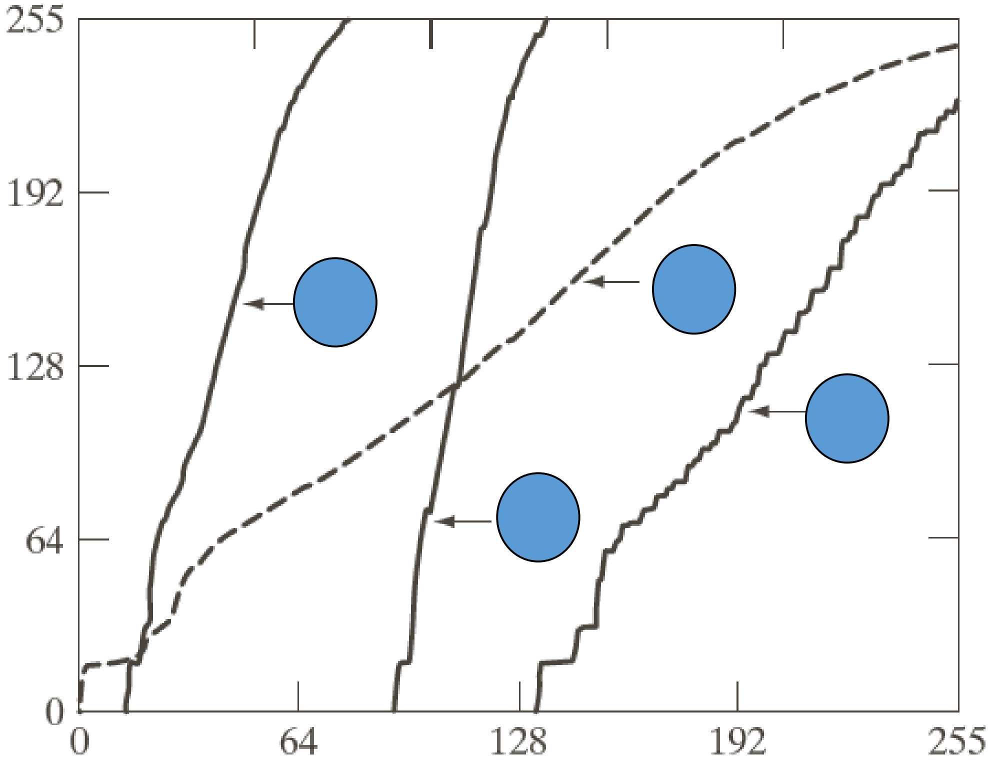
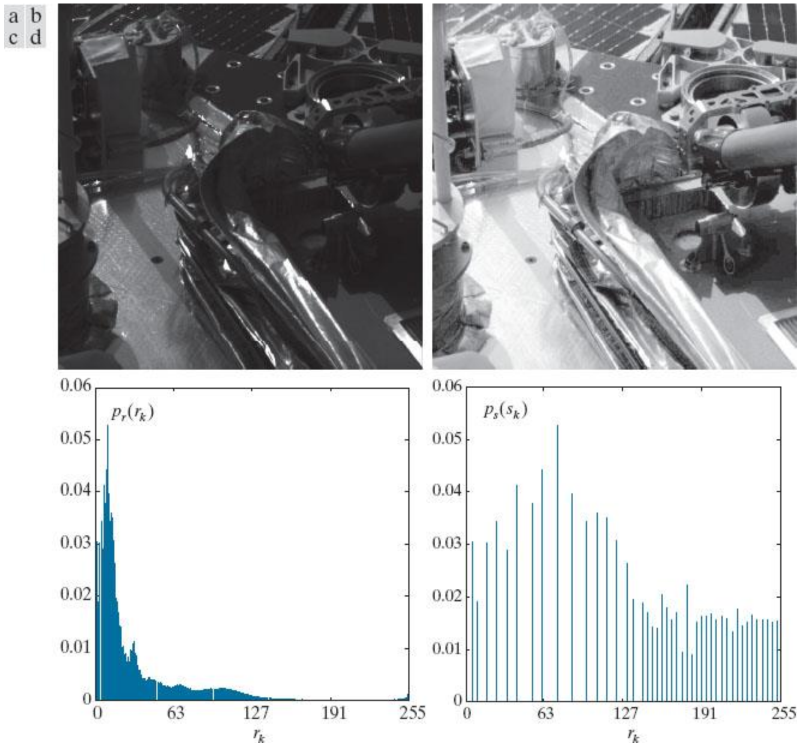


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

Figure 3.22

(a) Image from Phoenix Lander. (b) Result of histogram equalization. (c) Histogram of image (a). (d) Histogram of image (b). (Original image courtesy of NASA.)



Question

Question

Is histogram equalization always good?

No

Ques. Suppose that the intensity values in an image has PDF:

$$p_r(r) = \begin{cases} \dfrac{6r+2}{3(L-1)^2+2(L-1)} & \text{for } 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

Find the transformation fn $s = T(r)$ obtained through HE in continous variables. Also show that $T(r)$ has uniform PDF.

Sol. $$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

$$= (L-1)\int_0^r \frac{6w+2}{3(L-1)^2+2(L-1)}\,dw$$

$$= \frac{1}{3(L-1)+2}\left(\frac{6}{2}\left[w^2\right]_0^r + 2\left|w\right|_0^r\right)$$

$$= \frac{3r^2+2r}{3(L-1)+2}$$

$$\frac{ds}{dr} = \frac{d}{dr}\left[\frac{3r^2+2r}{3(L-1)+2}\right]$$

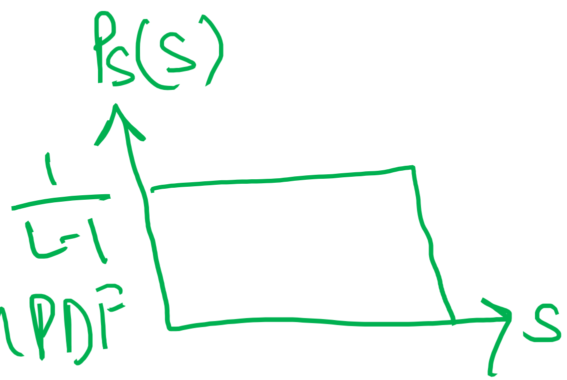
$$= \frac{6r+2}{3(L-1)+2} \quad \text{--- (1)}$$

we know

$$p_s(s) = p_r(r) \times \frac{dr}{ds} \quad \text{put (1)}$$

$$= \frac{\cancel{6r+2}}{3(L-1)^2+2(L-1)} \times \frac{3(L-1)+2}{\cancel{6r+2}}$$

$$= \frac{1}{L-1}$$



∴ we show that $T(r)$ i.e s has uniform PDF

Histogram Matching

- Histogram Equalization yields an image whose pixels are (in theory) uniformly distributed among all gray levels.
- 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 histogram specification.

Histogram Matching

Histogram matching (histogram specification)

— generate a processed image that has a specified histogram

Histogram Matching

Histogram matching (histogram specification)

— generate a processed image that has a specified histogram

Let $p_r(r)$ and $p_z(z)$ denote the continous probability density functions of the variables r and z. $p_z(z)$ is the specified probability density function.

Let s be the random variable with the probability

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

Define a random variable z with the probability

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

Histogram Matching

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

Histogram Matching

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

$$z = G^{-1}(s) = G^{-1}\left[T(r)\right]$$

Histogram Matching: Procedure

Histogram Matching: Procedure

- Obtain $p_r(r)$ from the input image and then obtain the values of s

- Use the specified PDF and obtain the transformation function G(z)

- Mapping from s to z

Histogram Matching: Procedure

- Obtain $p_r(r)$ from the input image and then obtain the values of s

$$s = (L-1)\int_0^r p_r(w)dw$$

- Use the specified PDF and obtain the transformation function G(z)

- Mapping from s to z

Histogram Matching: Procedure

- Obtain $p_r(r)$ from the input image and then obtain the values of s

$$s = (L-1)\int_0^r p_r(w)dw$$

- Use the specified PDF and obtain the transformation function G(z)

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

- Mapping from s to z

Histogram Matching: Procedure

- Obtain $p_r(r)$ from the input image and then obtain the values of s

$$s = (L-1)\int_0^r p_r(w)dw$$

- Use the specified PDF and obtain the transformation function G(z)

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

- Mapping from s to z

$$z = G^{-1}(s)$$

Histogram Matching: Example

Histogram Matching: Example

Assuming continuous intensity values, suppose that an image has the intensity PDF

Find the transformation function that will produce an image whose intensity PDF is

Histogram Matching: Example

Assuming continuous intensity values, suppose that an image has the intensity PDF

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2}, & \text{for } 0 \le r \le L\text{-}1 \\ 0\,, & \text{otherwise} \end{cases}$$

Find the transformation function that will produce an image whose intensity PDF is

Histogram Matching: Example

Assuming continuous intensity values, suppose that an image has the intensity PDF

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2}, & \text{for } 0 \le r \le L\text{-}1 \\ 0\ , & \text{otherwise} \end{cases}$$

Find the transformation function that will produce an image whose intensity PDF is

$$p_z(z) = \begin{cases} \dfrac{3z^2}{(L-1)^3}, & \text{for } 0 \le z \le (L\text{-}1) \\ 0, & \text{otherwise} \end{cases}$$

Histogram Matching: Example

Histogram Matching: Example

Find the histogram equalization transformation for the input image

Find the histogram equalization transformation for the specified histogram

The transformation function

Histogram Matching: Example

Find the histogram equalization transformation for the input image

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\int_0^r \frac{2w}{(L-1)^2}dw$$

Find the histogram equalization transformation for the specified histogram

The transformation function

Histogram Matching: Example

Find the histogram equalization transformation for the input image

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\int_0^r \frac{2w}{(L-1)^2}dw = \frac{r^2}{L-1}$$

Find the histogram equalization transformation for the specified histogram

$$G(z) = (L-1)\int_0^z p_z(t)dt = (L-1)\int_0^z \frac{3t^2}{(L-1)^3}dt = \frac{z^3}{(L-1)^2} = s$$

The transformation function

Histogram Matching: Example

Find the histogram equalization transformation for the input image

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\int_0^r \frac{2w}{(L-1)^2} dw = \frac{r^2}{L-1}$$

Find the histogram equalization transformation for the specified histogram

$$G(z) = (L-1)\int_0^z p_z(t)dt = (L-1)\int_0^z \frac{3t^2}{(L-1)^3} dt = \frac{z^3}{(L-1)^2} = s$$

The transformation function

Histogram Matching: Example

Find the histogram equalization transformation for the input image

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\int_0^r \frac{2w}{(L-1)^2}dw = \frac{r^2}{L-1}$$

Find the histogram equalization transformation for the specified histogram

$$G(z) = (L-1)\int_0^z p_z(t)dt = (L-1)\int_0^z \frac{3t^2}{(L-1)^3}dt = \frac{z^3}{(L-1)^2} = s$$

The transformation function

$$z = \left[(L-1)^2 s\right]^{1/3} = \left[(L-1)^2 \frac{r^2}{L-1}\right]^{1/3} = \left[(L-1)r^2\right]^{1/3}$$

Histogram Matching: Discrete Cases

Histogram Matching: Discrete Cases

- Obtain $p_r(r_j)$ from the input image and then obtain the values of s_k, round the value to the integer range [0, L-1].

- Use the specified PDF and obtain the transformation function $G(z_q)$, round the value to the integer range [0, L-1].

- Mapping from s_k to z_q

Histogram Matching: Discrete Cases

- Obtain $p_r(r_j)$ from the input image and then obtain the values of s_k, round the value to the integer range [0, L-1].

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

- Use the specified PDF and obtain the transformation function $G(z_q)$, round the value to the integer range [0, L-1].

- Mapping from s_k to z_q

Histogram Matching: Discrete Cases

- Obtain $p_r(r_j)$ from the input image and then obtain the values of s_k, round the value to the integer range [0, L-1].

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

- Use the specified PDF and obtain the transformation function $G(z_q)$, round the value to the integer range [0, L-1].

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) = s_k$$

- Mapping from s_k to z_q

Histogram Matching: Discrete Cases

- Obtain $p_r(r_j)$ from the input image and then obtain the values of s_k, round the value to the integer range [0, L-1].

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

- Use the specified PDF and obtain the transformation function $G(z_q)$, round the value to the integer range [0, L-1].

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) = s_k$$

- Mapping from s_k to z_q

$$z_q = G^{-1}(s_k)$$

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image *'r.'*

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image '*r.*'

Gray Level r_k
r0 = 0
r1 = 1
r2 = 2
r3 = 3
r4 = 4
r5 = 5
r6 = 6
r7 = 7

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image '*r.*'

Gray Level r_k	n_k
r0 = 0	790
r1 = 1	1023
r2 = 2	850
r3 = 3	656
r4 = 4	329
r5 = 5	245
r6 = 6	122
r7 = 7	81

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image *'r.'*

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF
r0 = 0	790	790/4096 = 0.19
r1 = 1	1023	1023/4096 = 0.25
r2 = 2	850	850/4096 = 0.21
r3 = 3	656	656/4096 = 0.16
r4 = 4	329	329/4096 = 0.08
r5 = 5	245	245/4096 = 0.06
r6 = 6	122	122/4096 = 0.03
r7 = 7	81	81/4096 = 0.02

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image *'r.'*

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF	$CDF = \sum_{j=0}^{k} p_r(r_j)$
r0 = 0	790	790/4096 = 0.19	0.19
r1 = 1	1023	1023/4096 = 0.25	0.44
r2 = 2	850	850/4096 = 0.21	0.65
r3 = 3	656	656/4096 = 0.16	0.81
r4 = 4	329	329/4096 = 0.08	0.89
r5 = 5	245	245/4096 = 0.06	0.95
r6 = 6	122	122/4096 = 0.03	0.98
r7 = 7	81	81/4096 = 0.02	1

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image *'r.'*

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF	$CDF = \sum_{j=0}^{k} p_r(r_j)$	$s_k = (L-1) * CDF$
r0 = 0	790	790/4096 = 0.19	0.19	7 x 0.19 = 1.33
r1 = 1	1023	1023/4096 = 0.25	0.44	7 x 0.44 = 3.08
r2 = 2	850	850/4096 = 0.21	0.65	7 x 0.65 = 4.55
r3 = 3	656	656/4096 = 0.16	0.81	7 x 0.81 = 5.67
r4 = 4	329	329/4096 = 0.08	0.89	7 x 0.89 = 6.23
r5 = 5	245	245/4096 = 0.06	0.95	7 x 0.95 = 6.65
r6 = 6	122	122/4096 = 0.03	0.98	7 x 0.98 = 6.86
r7 = 7	81	81/4096 = 0.02	1	7 x 1 = 7

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image *'r.'*

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF	$CDF = \sum_{j=0}^{k} p_r(r_j)$	$s_k = (L - 1) * CDF$	Gray Level s_k
r0 = 0	790	790/4096 = 0.19	0.19	7 x 0.19 = 1.33	s0 = 1
r1 = 1	1023	1023/4096 = 0.25	0.44	7 x 0.44 = 3.08	s1 = 3
r2 = 2	850	850/4096 = 0.21	0.65	7 x 0.65 = 4.55	s2 = 5
r3 = 3	656	656/4096 = 0.16	0.81	7 x 0.81 = 5.67	s3 = 6
r4 = 4	329	329/4096 = 0.08	0.89	7 x 0.89 = 6.23	s4 = 6
r5 = 5	245	245/4096 = 0.06	0.95	7 x 0.95 = 6.65	s5 = 7
r6 = 6	122	122/4096 = 0.03	0.98	7 x 0.98 = 6.86	s6 = 7
r7 = 7	81	81/4096 = 0.02	1	7 x 1 = 7	s7 = 7

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

Step 1: To find scaled histogram equalized values *'s'* for original image '*r.*'

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF	$CDF = \sum_{j=0}^{k} p_r(r_j)$	$s_k = (L-1) * CDF$	Gray Level s_k	n_k
r0 = 0	790	790/4096 = 0.19	0.19	7 x 0.19 = 1.33	s0 = 1	790
r1 = 1	1023	1023/4096 = 0.25	0.44	7 x 0.44 = 3.08	s1 = 3	1023
r2 = 2	850	850/4096 = 0.21	0.65	7 x 0.65 = 4.55	s2 = 5	850
r3 = 3	656	656/4096 = 0.16	0.81	7 x 0.81 = 5.67	s3 = 6	656 +329 = 985
r4 = 4	329	329/4096 = 0.08	0.89	7 x 0.89 = 6.23	s4 = 6	
r5 = 5	245	245/4096 = 0.06	0.95	7 x 0.95 = 6.65	s5 = 7	245 + 122 + 81 = 448
r6 = 6	122	122/4096 = 0.03	0.98	7 x 0.98 = 6.86	s6 = 7	
r7 = 7	81	81/4096 = 0.02	1	7 x 1 = 7	s7 = 7	

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Gray Level z_i
z0 = 0
z1 = 1
z2 = 2
z3 = 3
z4 = 4
z5 = 5
z6 = 6
z7 = 7

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be 'z'.

Gray Level z_i	$p_z(z_i) = n_i/MN$ PDF
z0 = 0	0.00
z1 = 1	0.00
z2 = 2	0.00
z3 = 3	0.15
z4 = 4	0.20
z5 = 5	0.30
z6 = 6	0.20
z7 = 7	0.15

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be 'z'.

Gray Level z_i	$p_z(z_i) = n_i/MN$ PDF	$CDF = \sum_{j=0}^{i} p_z(z_i)$
z0 = 0	0.00	0.00
z1 = 1	0.00	0.00
z2 = 2	0.00	0.00
z3 = 3	0.15	0.15
z4 = 4	0.20	0.35
z5 = 5	0.30	0.65
z6 = 6	0.20	0.85
z7 = 7	0.15	1

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Gray Level z_i	$p_z(z_i) = n_i/MN$ PDF	$CDF = \sum_{j=0}^{i} p_z(z_i)$	$G(z) = (L-1) * CDF$
z0 = 0	0.00	0.00	7 x 0.00 = 0.00
z1 = 1	0.00	0.00	7 x 0.00= 0.00
z2 = 2	0.00	0.00	7 x 0.00 = 0.00
z3 = 3	0.15	0.15	7 x 0.15 = 1.05
z4 = 4	0.20	0.35	7 x 0.35 = 2.45
z5 = 5	0.30	0.65	7 x 0.65 = 4.55
z6 = 6	0.20	0.85	7 x 0.85 = 5.95
z7 = 7	0.15	1	7 x 1 = 7

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *‘z’*.

Gray Level z_i	$p_z(z_i) = n_i/MN$ PDF	$CDF = \sum_{j=0}^{i} p_z(z_i)$	$G(z) = (L-1) * CDF$	$G(z)$
z0 = 0	0.00	0.00	7 x 0.00 = 0.00	0
z1 = 1	0.00	0.00	7 x 0.00= 0.00	0
z2 = 2	0.00	0.00	7 x 0.00 = 0.00	0
z3 = 3	0.15	0.15	7 x 0.15 = 1.05	1
z4 = 4	0.20	0.35	7 x 0.35 = 2.45	2
z5 = 5	0.30	0.65	7 x 0.65 = 4.55	5
z6 = 6	0.20	0.85	7 x 0.85 = 5.95	6
z7 = 7	0.15	1	7 x 1 = 7	7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of $G(z)$ is closest to s_k

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k
r0 = 0
r1 = 1
r2 = 2
r3 = 3
r4 = 4
r5 = 5
r6 = 6
r7 = 7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k
r0 = 0	s0 = 1
r1 = 1	s1 = 3
r2 = 2	s2 = 5
r3 = 3	s3 = 6
r4 = 4	s4 = 6
r5 = 5	s5 = 7
r6 = 6	s6 = 7
r7 = 7	s7 = 7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$
r0 = 0	s0 = 1	0
r1 = 1	s1 = 3	0
r2 = 2	s2 = 5	0
r3 = 3	s3 = 6	1
r4 = 4	s4 = 6	2
r5 = 5	s5 = 7	5
r6 = 6	s6 = 7	6
r7 = 7	s7 = 7	7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$	z_i
r0 = 0	s0 = 1	0	3
r1 = 1	s1 = 3	0	4
r2 = 2	s2 = 5	0	5
r3 = 3	s3 = 6	1	6
r4 = 4	s4 = 6	2	6
r5 = 5	s5 = 7	5	7
r6 = 6	s6 = 7	6	7
r7 = 7	s7 = 7	7	7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$	z_i	n_i
r0 = 0	s0 = 1	0	3	790
r1 = 1	s1 = 3	0	4	1023
r2 = 2	s2 = 5	0	5	850
r3 = 3	s3 = 6	1	6	985
r4 = 4	s4 = 6	2	6	
r5 = 5	s5 = 7	5	7	448
r6 = 6	s6 = 7	6	7	
r7 = 7	s7 = 7	7	7	

Histogram Matching - Example

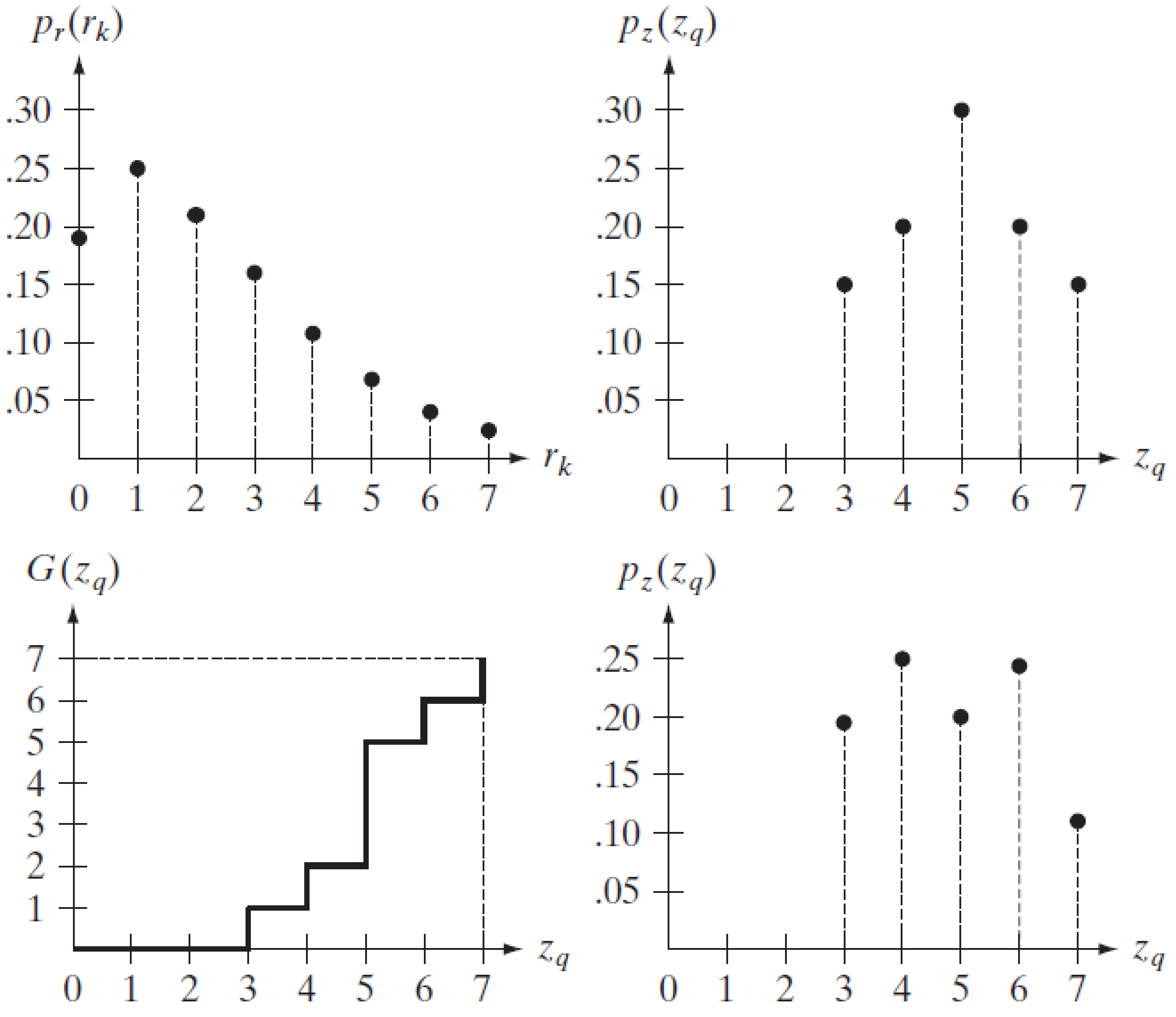
- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$	z_i	n_i	$p_z(z_i) = n_i/MN$ PDF
r0 = 0	s0 = 1	0	3	790	790/4096 = 0.19
r1 = 1	s1 = 3	0	4	1023	1023/4096 = 0.25
r2 = 2	s2 = 5	0	5	850	850/4096 = 0.21
r3 = 3	s3 = 6	1	6	985	985/4096 = 0.24
r4 = 4	s4 = 6	2	6		
r5 = 5	s5 = 7	5	7	448	448/4096 = 0.11
r6 = 6	s6 = 7	6	7		
r7 = 7	s7 = 7	7	7		

Example: Histogram Matching

a	b
c	d

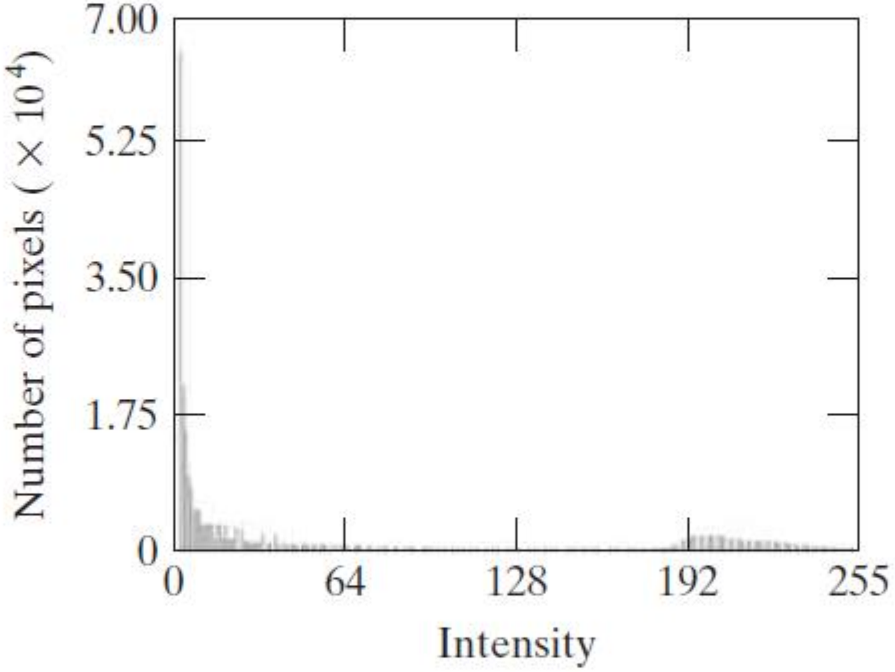
FIGURE 3.22
(a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).



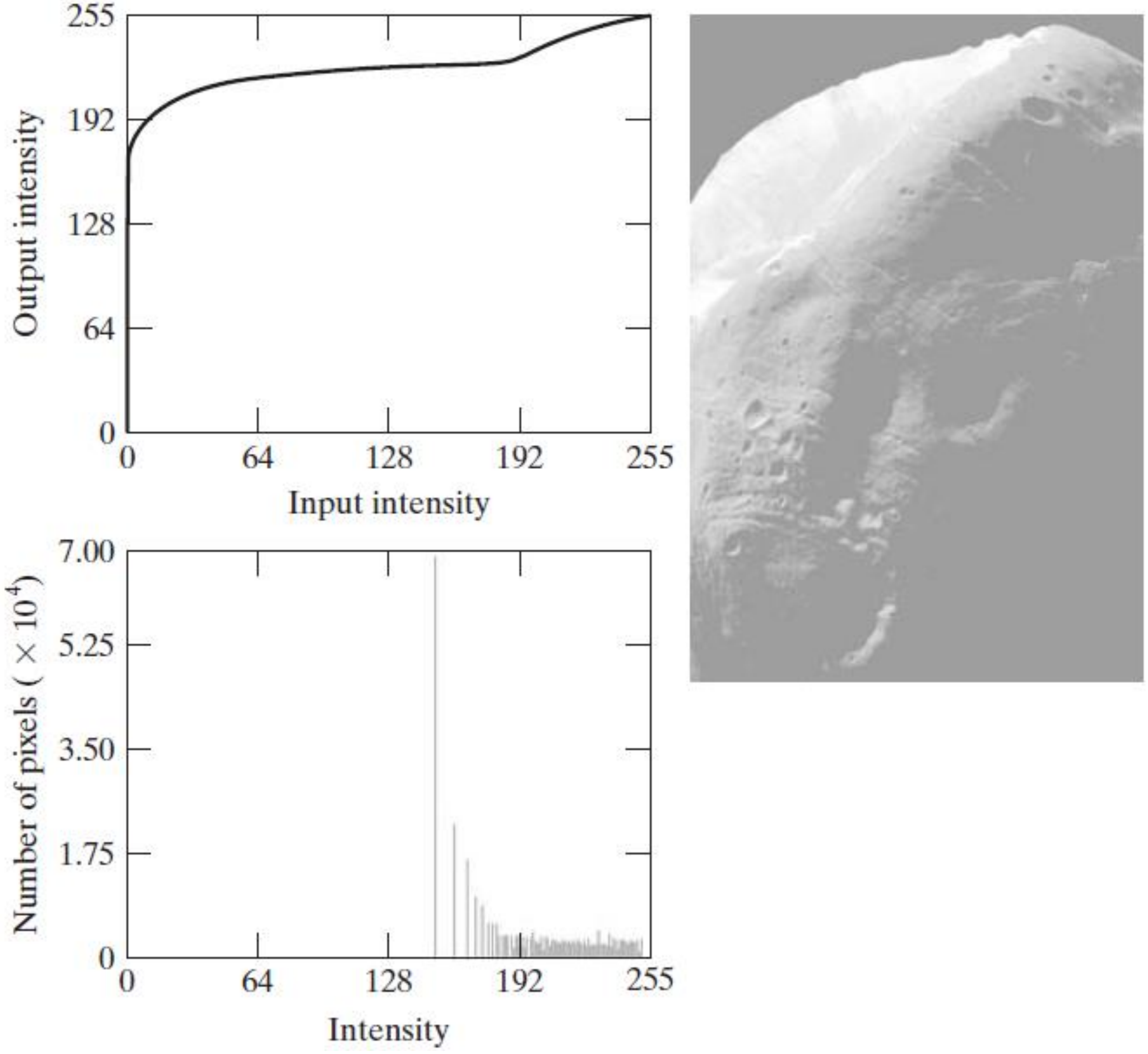
Example: Histogram Matching

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.)



Example: 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).

a c
b
d

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

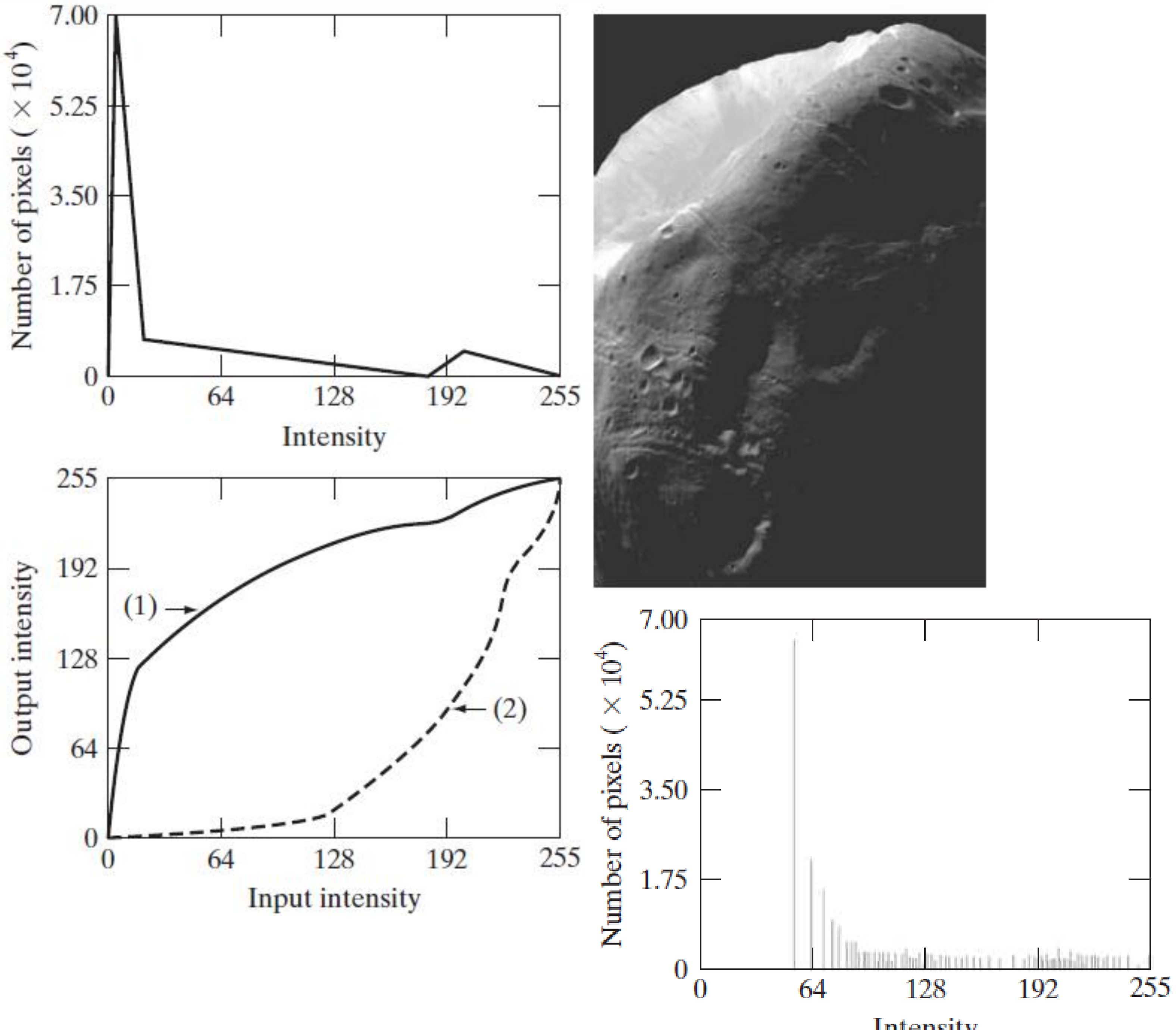


Figure 3.24

(a) An image, and (b) its histogram.

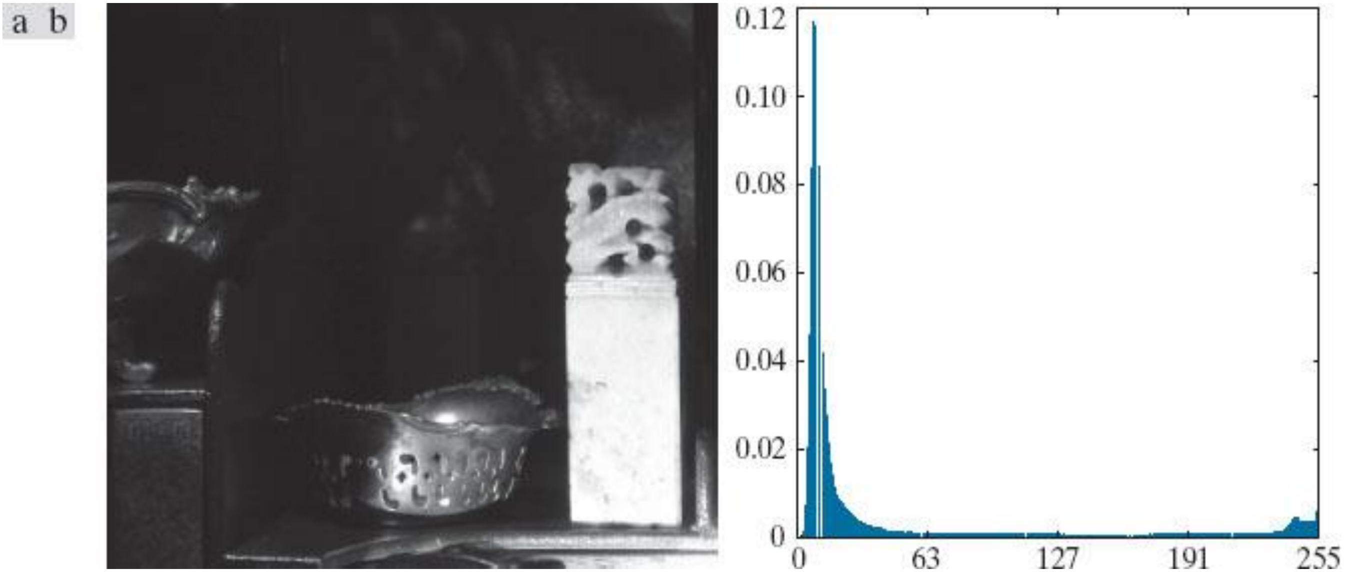


Figure 3.25

(a) Histogram equalization transformation obtained using the histogram in Fig. 3.24(b). (b) Histogram equalized image. (c) Histogram of equalized image.

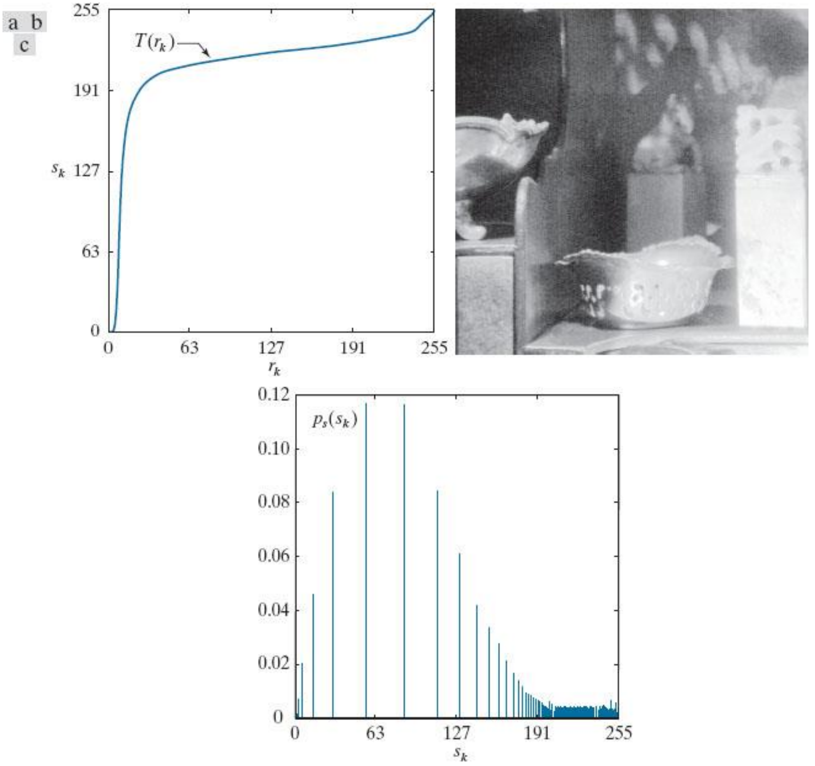
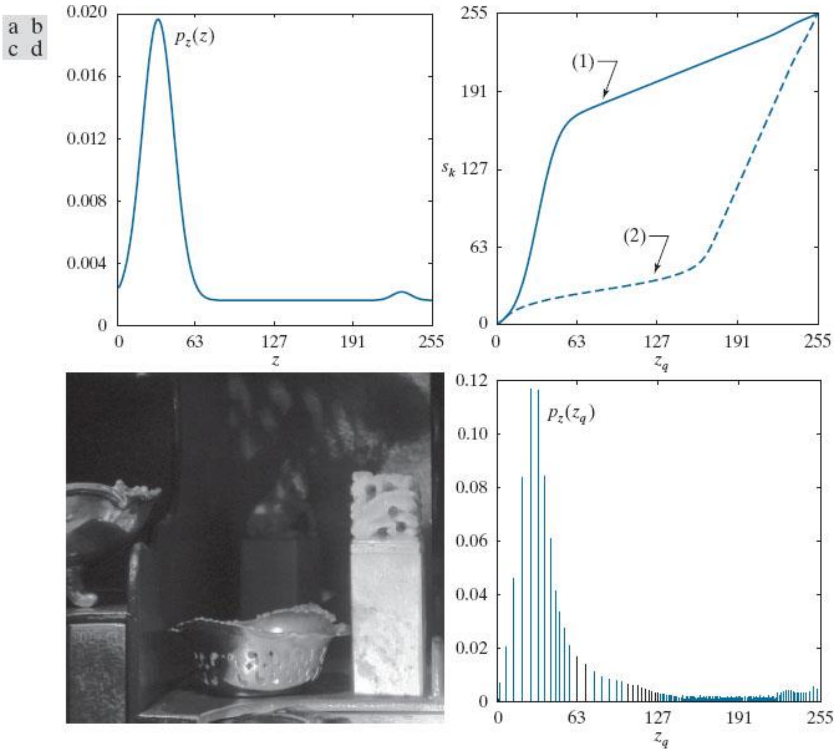


Figure 3.26

Histogram specification. (a) Specified histogram. (b) Transformation $G(z_q)$, labeled (1), and $G^{-1}(s_k)$, labeled (2). (c) Result of histogram specification. (d) Histogram of image (c).



Histogram Matching (Specification)

- Trial and error process
- No specific rules!!!

Histogram Matching - Example

- Perform Histogram Specification. Draw Original and matched Histogram.
- Original Image:

Gray Level	0	1	2	3	4	5	6	7
No. of Pixels	6	4	5	7	8	10	12	12

- Specified Image:

Gray Level	0	1	2	3	4	5	6	7
No. of Pixels	0	0	0	0	10	20	22	12

Example: Histogram Matching

Step 1: To find scaled histogram equalized values *‘s’* for original image ‘*r.’*

Example: Histogram Matching

Step 1: To find scaled histogram equalized values *'s'* for original image '*r.*'

Gray Level r_k
r0 = 0
r1 = 1
r2 = 2
r3 = 3
r4 = 4
r5 = 5
r6 = 6
r7 = 7

Example: Histogram Matching

Step 1: To find scaled histogram equalized values *‘s’* for original image ‘*r.*’

Gray Level r_k	n_k
r0 = 0	6
r1 = 1	4
r2 = 2	5
r3 = 3	7
r4 = 4	8
r5 = 5	10
r6 = 6	12
r7 = 7	12

Example: Histogram Matching

Step 1: To find scaled histogram equalized values *‘s’* for original image ‘*r.*’

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF
r0 = 0	6	6/64 = 0.094
r1 = 1	4	0.0625
r2 = 2	5	0.0781
r3 = 3	7	0.109
r4 = 4	8	0.125
r5 = 5	10	0.156
r6 = 6	12	0.1875
r7 = 7	12	0.1875

Example: Histogram Matching

Step 1: To find scaled histogram equalized values *‘s’* for original image ‘*r.*’

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF	$CDF = \sum_{j=0}^{k} p_r(r_j)$
r0 = 0	6	6/64 = 0.094	0.094
r1 = 1	4	0.0625	0.1565
r2 = 2	5	0.0781	0.2346
r3 = 3	7	0.109	0.343
r4 = 4	8	0.125	0.4686
r5 = 5	10	0.156	0.6246
r6 = 6	12	0.1875	0.8121
r7 = 7	12	0.1875	0.9996

Example: Histogram Matching

Step 1: To find scaled histogram equalized values *'s'* for original image '*r.*'

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF	$CDF = \sum_{j=0}^{k} p_r(r_j)$	$s_k = (L - 1) * CDF$
r0 = 0	6	6/64 = 0.094	0.094	7 x 0.094=0.658
r1 = 1	4	0.0625	0.1565	1.09
r2 = 2	5	0.0781	0.2346	1.6
r3 = 3	7	0.109	0.343	2.4
r4 = 4	8	0.125	0.4686	3.2
r5 = 5	10	0.156	0.6246	4.3
r6 = 6	12	0.1875	0.8121	5.7
r7 = 7	12	0.1875	0.9996	6.9

Example: Histogram Matching

Step 1: To find scaled histogram equalized values *'s'* for original image '*r.*'

Gray Level r_k	n_k	$p_r(r_k) = n_k/MN$ PDF	$CDF = \sum_{j=0}^{k} p_r(r_j)$	$s_k = (L - 1) * CDF$	Gray Level s_k
r0 = 0	6	6/64 = 0.094	0.094	7 x 0.094=0.658	s0 = 1
r1 = 1	4	0.0625	0.1565	1.09	s1 = 1
r2 = 2	5	0.0781	0.2346	1.6	s2 = 2
r3 = 3	7	0.109	0.343	2.4	s3 = 2
r4 = 4	8	0.125	0.4686	3.2	s4 = 3
r5 = 5	10	0.156	0.6246	4.3	s5 = 4
r6 = 6	12	0.1875	0.8121	5.7	s6 = 6
r7 = 7	12	0.1875	0.9996	6.9	s7 = 7

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *‘z’*.

Gray Level z_i
z0 = 0
z1 = 1
z2 = 2
z3 = 3
z4 = 4
z5 = 5
z6 = 6
z7 = 7

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Gray Level z_i	n_k	$p_z(z_i) = n_i/MN$ PDF
z0 = 0	0	0.00
z1 = 1	0	0.00
z2 = 2	0	0.00
z3 = 3	0	0.00
z4 = 4	10	0.156
z5 = 5	20	0.3125
z6 = 6	22	0.3438
z7 = 7	12	0.1875

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *‘z’*.

Gray Level z_i	n_k	$p_z(z_i) = n_i/MN$ PDF	$CDF = \sum_{j=0}^{i} p_z(z_i)$
z0 = 0	0	0.00	0.00
z1 = 1	0	0.00	0.00
z2 = 2	0	0.00	0.00
z3 = 3	0	0.00	0.00
z4 = 4	10	0.156	0.156
z5 = 5	20	0.3125	0.4685
z6 = 6	22	0.3438	0.8123
z7 = 7	12	0.1875	0.9998

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Gray Level z_i	n_k	$p_z(z_i) = n_i/MN$ PDF	$CDF = \sum_{j=0}^{i} p_z(z_i)$	$G(z) = (L-1) * CDF$
z0 = 0	0	0.00	0.00	7 x 0.00 = 0.00
z1 = 1	0	0.00	0.00	7 x 0.00= 0.00
z2 = 2	0	0.00	0.00	7 x 0.00 = 0.00
z3 = 3	0	0.00	0.00	7 x 0.00 = 0.00
z4 = 4	10	0.156	0.156	7 x 0.156 = 1.092
z5 = 5	20	0.3125	0.4685	7 x 0.4685 = 3.27
z6 = 6	22	0.3438	0.8123	7 x 0.8123 = 5.68
z7 = 7	12	0.1875	0.9998	7 x 0.9998 = 6.998

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Gray Level z_i	n_k	$p_z(z_i) = n_i/MN$ PDF	$CDF = \sum_{j=0}^{i} p_z(z_i)$	$G(z) = (L-1) * CDF$	$G(z)$
z0 = 0	0	0.00	0.00	7 x 0.00 = 0.00	0
z1 = 1	0	0.00	0.00	7 x 0.00= 0.00	0
z2 = 2	0	0.00	0.00	7 x 0.00 = 0.00	0
z3 = 3	0	0.00	0.00	7 x 0.00 = 0.00	0
z4 = 4	10	0.156	0.156	7 x 0.156 = 1.092	1
z5 = 5	20	0.3125	0.4685	7 x 0.4685 = 3.27	3
z6 = 6	22	0.3438	0.8123	7 x 0.8123 = 5.68	6
z7 = 7	12	0.1875	0.9998	7 x 0.9998 = 6.998	7

Example: Histogram Matching

Step 2: Compute all values of transformation function G(z) for the specified image histogram. Let this specified image be *'z'*.

Gray Level z_i	n_k	$p_z(z_i) = n_i/MN$ PDF	$CDF = \sum_{j=0}^{i} p_z(z_i)$	$G(z) = (L - 1) * CDF$	$G(z)$
z0 = 0	0	0.00	0.00	7 x 0.00 = 0.00	0
z1 = 1	0	0.00	0.00	7 x 0.00= 0.00	0
z2 = 2	0	0.00	0.00	7 x 0.00 = 0.00	0
z3 = 3	0	0.00	0.00	7 x 0.00 = 0.00	0
z4 = 4	10	0.156	0.156	7 x 0.156 = 1.092	1
z5 = 5	20	0.3125	0.4685	7 x 0.4685 = 3.27	3
z6 = 6	22	0.3438	0.8123	7 x 0.8123 = 5.68	6
z7 = 7	12	0.1875	0.9998	7 x 0.9998 = 6.998	7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of $G(z)$ is closest to s_k

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k
r0 = 0
r1 = 1
r2 = 2
r3 = 3
r4 = 4
r5 = 5
r6 = 6
r7 = 7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k
r0 = 0	s0 = 1
r1 = 1	s1 = 1
r2 = 2	s2 = 2
r3 = 3	s3 = 2
r4 = 4	s4 = 3
r5 = 5	s5 = 4
r6 = 6	s6 = 6
r7 = 7	s7 = 7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$
r0 = 0	s0 = 1	0
r1 = 1	s1 = 1	0
r2 = 2	s2 = 2	0
r3 = 3	s3 = 2	0
r4 = 4	s4 = 3	1
r5 = 5	s5 = 4	3
r6 = 6	s6 = 6	6
r7 = 7	s7 = 7	7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$	z_i
r0 = 0	s0 = 1	0	4
r1 = 1	s1 = 1	0	4
r2 = 2	s2 = 2	0	4
r3 = 3	s3 = 2	0	4
r4 = 4	s4 = 3	1	5
r5 = 5	s5 = 4	3	5
r6 = 6	s6 = 6	6	6
r7 = 7	s7 = 7	7	7

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$	z_i	n_i
r0 = 0	s0 = 1	0	4	22
r1 = 1	s1 = 1	0	4	
r2 = 2	s2 = 2	0	4	
r3 = 3	s3 = 2	0	4	
r4 = 4	s4 = 3	1	5	18
r5 = 5	s5 = 4	3	5	
r6 = 6	s6 = 6	6	6	12
r7 = 7	s7 = 7	7	7	12

Histogram Matching - Example

- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Downward Comparisions

Upward Comparisions

Gray Level r_k	Gray Level s_k	$G(z)$	z_i	n_i	z_i
r0 = 0	s0 = 1	0	4	22	4
r1 = 1	s1 = 1	0	4		4
r2 = 2	s2 = 2	0	4		5
r3 = 3	s3 = 2	0	4		5
r4 = 4	s4 = 3	1	5	18	5
r5 = 5	s5 = 4	3	5		5
r6 = 6	s6 = 6	6	6	12	6
r7 = 7	s7 = 7	7	7	12	7

Histogram Matching - Example

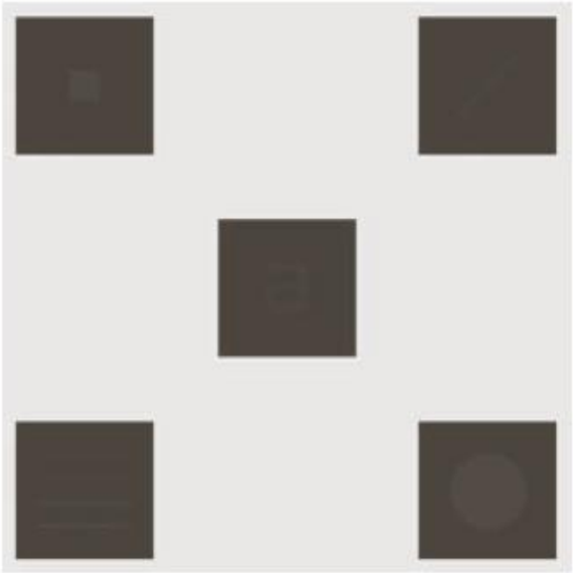
- Step 3: Create mappings from *s to z* : We find the value of z_q so that the value of *G(z)* is closest to s_k

Gray Level r_k	Gray Level s_k	$G(z)$	z_i	n_i	z_i	n_i
r0 = 0	s0 = 1	0	4	22	4	10
r1 = 1	s1 = 1	0	4		4	
r2 = 2	s2 = 2	0	4		5	30
r3 = 3	s3 = 2	0	4		5	
r4 = 4	s4 = 3	1	5	18	5	
r5 = 5	s5 = 4	3	5		5	
r6 = 6	s6 = 6	6	6	12	6	12
r7 = 7	s7 = 7	7	7	12	7	12

Local Histogram Processing

- Define a neighborhood and move its center from pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed. Either histogram equalization or histogram specification transformation function is obtained.
- Map the intensity of the pixel centered in the neighborhood.
- Move to the next location and repeat the procedure

Local Histogram Processing: Example

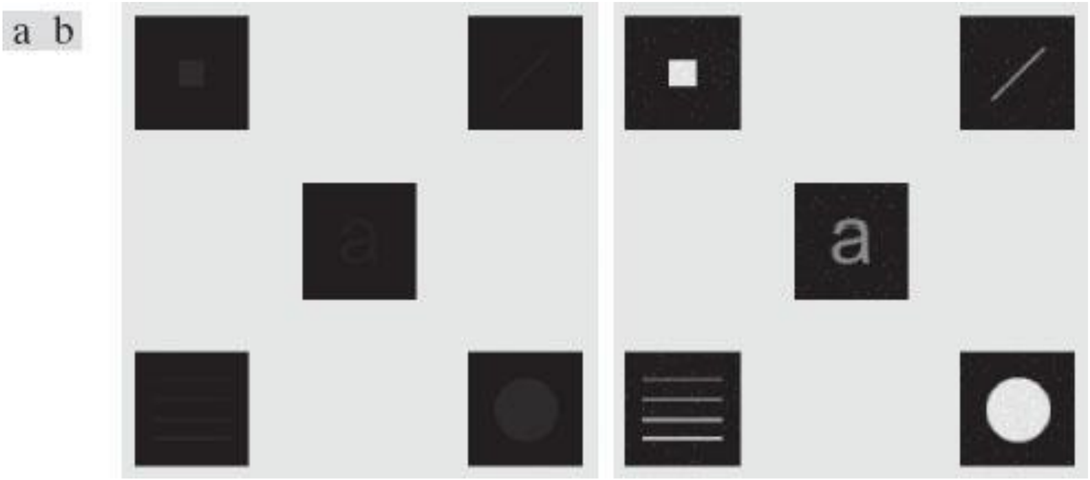


a b c

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.

Figure 3.33

(a) Original image. (b) Result of local enhancement based on local histogram statistics. Compare (b) with Fig. 3.32(c).



Using Histogram Statistics for Image Enhancement

$$m = \sum_{i=0}^{L-1} r_i p(r_i)$$

$$u_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$$

$$\sigma^2 = u_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$$

Using Histogram Statistics for Image Enhancement

Average Intensity

$$m = \sum_{i=0}^{L-1} r_i p(r_i) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

$$u_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$$

$$\sigma^2 = u_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$$

Using Histogram Statistics for Image Enhancement

Average Intensity

$$m = \sum_{i=0}^{L-1} r_i\, p(r_i) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

$$u_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n\, p(r_i)$$

Variance

$$\sigma^2 = u_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2\, p(r_i) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [f(x,y) - m]^2$$

Using Histogram Statistics for Image Enhancement

Average Intensity

$$m = \sum_{i=0}^{L-1} r_i \, p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

$$u_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n \, p(r_i)$$

Variance

$$\sigma^2 = u_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 \, p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[f(x,y) - m \right]^2$$

Using Histogram Statistics for Image Enhancement

Average Intensity

$$m = \sum_{i=0}^{L-1} r_i p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

$$u_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$$

Variance

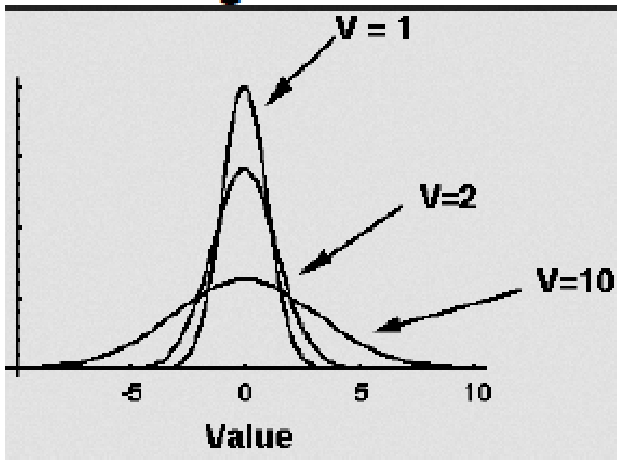
$$\sigma^2 = u_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - m]^2$$

Histogram statistics:

the nth moment of r about its mean is defined as

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n \, p(r_i)$$

Mean is a measure of average intensity and the variance is a measure of contrast in an image.



EXAMPLE 3.11: Computing histogram statistics.

■ Before proceeding, it will be useful to work through a simple numerical example to fix ideas. Consider the following 2-bit image of size 5×5:

$$\begin{matrix} 0 & 0 & 1 & 1 & 2 \\ 1 & 2 & 3 & 0 & 1 \\ 3 & 3 & 2 & 2 & 0 \\ 2 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 2 & 2 \end{matrix}$$

The pixels are represented by 2 bits; therefore, $L = 4$ and the intensity levels are in the range $[0, 3]$. The total number of pixels is 25, so the histogram has the components

$$p(r_0) = \frac{6}{25} = 0.24; \; p(r_1) = \frac{7}{25} = 0.28;$$

$$p(r_2) = \frac{7}{25} = 0.28; \; p(r_3) = \frac{5}{25} = 0.20$$

where the numerator in $p(r_i)$ is the number of pixels in the image with intensity level r_i. We can compute the average value of the intensities in the image using Eq. (3.3-18):

$$\begin{aligned} m &= \sum_{i=0}^{3} r_i \, p(r_i) \\ &= (0)(0.24) + (1)(0.28) + (2)(0.28) + (3)(0.20) \\ &= 1.44 \end{aligned}$$

Letting $f(x, y)$ denote the preceding 5×5 array and using Eq. (3.3-20), we obtain

$$\begin{aligned} m &= \frac{1}{25} \sum_{x=0}^{4} \sum_{y=0}^{4} f(x, y) \\ &= 1.44 \end{aligned}$$

Using Histogram Statistics for Image Enhancement

- Global mean and variance are measured over entire image. Used for gross adjustment of overall intensity and variance.
- Local mean and variance are measured locally. Used for local adjustment of intensity and contrast.
- *(x, y)* Co-ordinates of pixel.
- S_{xy} : neighbourhood (sub image), centered on *(x, y)*.
- $r_0 \ \ldots\ldots r_{L-1}$: possible intensity values.
- $p_{s_{xy}}$: Histogram of pixels in the region S_{xy}

Local average intensity

$$m_{s_{xy}} = \sum_{i=0}^{L-1} r_i \, p_{s_{xy}}(r_i)$$

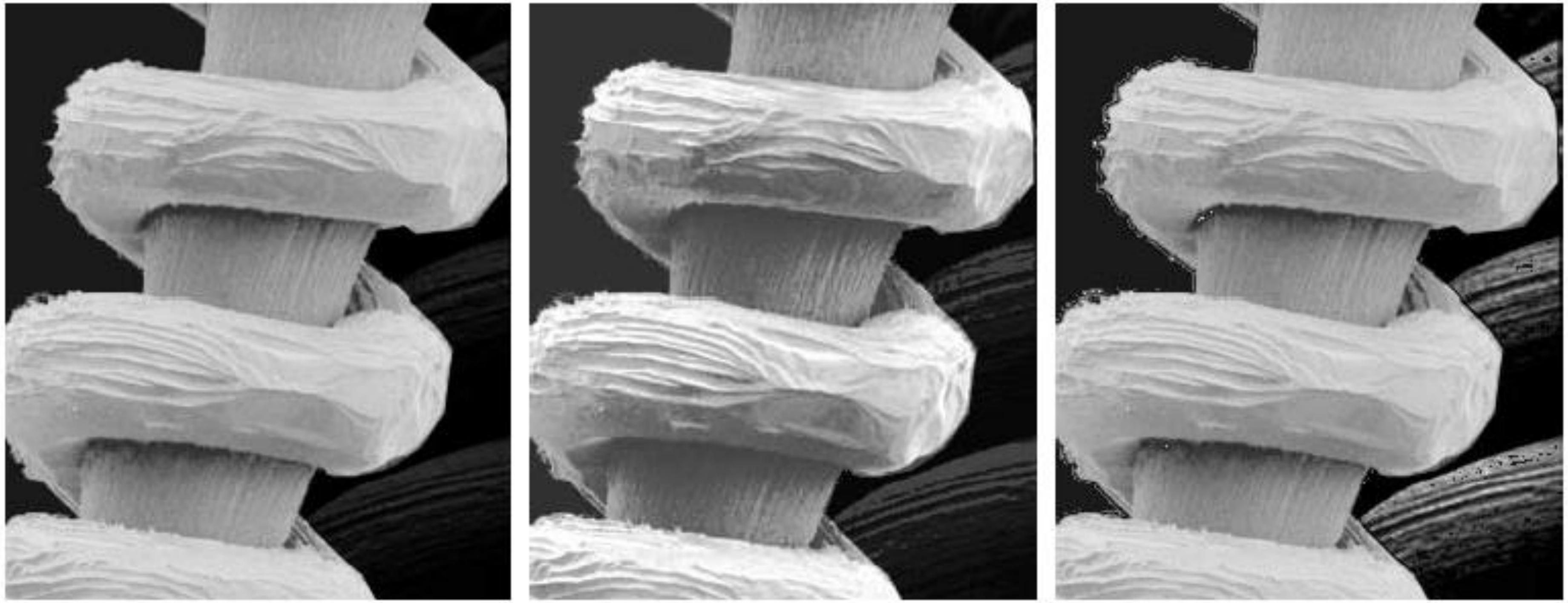
s_{xy} denotes a neighborhood

Local variance

$$\sigma^2_{s_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{s_{xy}})^2 \, p_{s_{xy}}(r_i)$$

m_G : global mean; σ_G : global standard deviation

$k_0 = 0.4$; $k_1 = 0.02$; $k_2 = 0.4$; $E = 4$



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Image Enhancement using Local histogram statistics

- Measure whether an area is relatively light or dark at *(x,y)*. Compare the local average gray level $m_{S_{xy}}$ to the global mean m_G.
- *(x,y)* is a candidate for enhancement if $m_{S_{xy}} \leq k_0 m_G$
- Enhance areas that have low contrast
- Compare the local standard deviation $\sigma_{S_{xy}}$ to the global standard deviation σ_G
- *(x,y)* is a candidate for enhancement if $\sigma_{S_{xy}} \leq k_2 \sigma_G$
- Restrict lowest values of contrast
- *(x,y)* is a candidate for enhancement if $k_1 \sigma_G \leq \sigma_{S_{xy}}$

We summarize the preceding approach as follows. Let $f(x, y)$ represent the value of an image at any image coordinates (x, y), and let $g(x, y)$ represent the corresponding enhanced value at those coordinates. Then,

$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } m_{S_{xy}} \leq k_0 m_G \text{ AND } k_1 \sigma_G \leq \sigma_{S_{xy}} \leq k_2 \sigma_G \\ \\ f(x, y) & \text{otherwise} \end{cases} \tag{3.3-24}$$

We summarize the preceding approach as follows. Let $f(x, y)$ represent the value of an image at any image coordinates (x, y), and let $g(x, y)$ represent the corresponding enhanced value at those coordinates. Then,

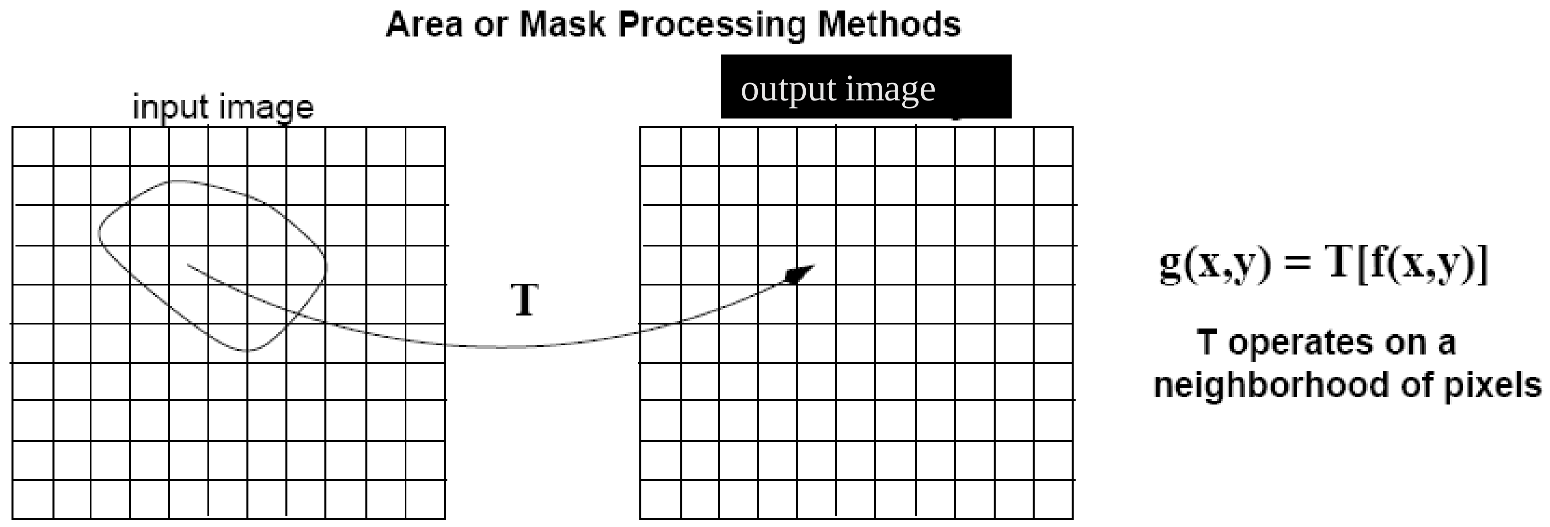
$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } m_{S_{xy}} \leq k_0 m_G \text{ AND } k_1 \sigma_G \leq \sigma_{S_{xy}} \leq k_2 \sigma_G \\ f(x, y) & \text{otherwise} \end{cases} \tag{3.3-24}$$

for $x = 0, 1, 2, \dots, M - 1$ and $y = 0, 1, 2, \dots, N - 1$, where, as indicated above, E, k_0, k_1, and k_2 are specified parameters, m_G is the global mean of the input image, and σ_G is its standard deviation. Parameters $m_{S_{xy}}$ and $\sigma_{S_{xy}}$ are the local mean and standard deviation, respectively. As usual, M and N are the row and column image dimensions.

The Mechanics of Spatial Filtering
(II Part of Image Enhancement)

Spatial Filtering Methods (or Mask Processing Methods)

- Some operations work with the values of the image pixels in the neighborhood and the corresponding values of the sub-image that has the same dimensions as the neighborhood.

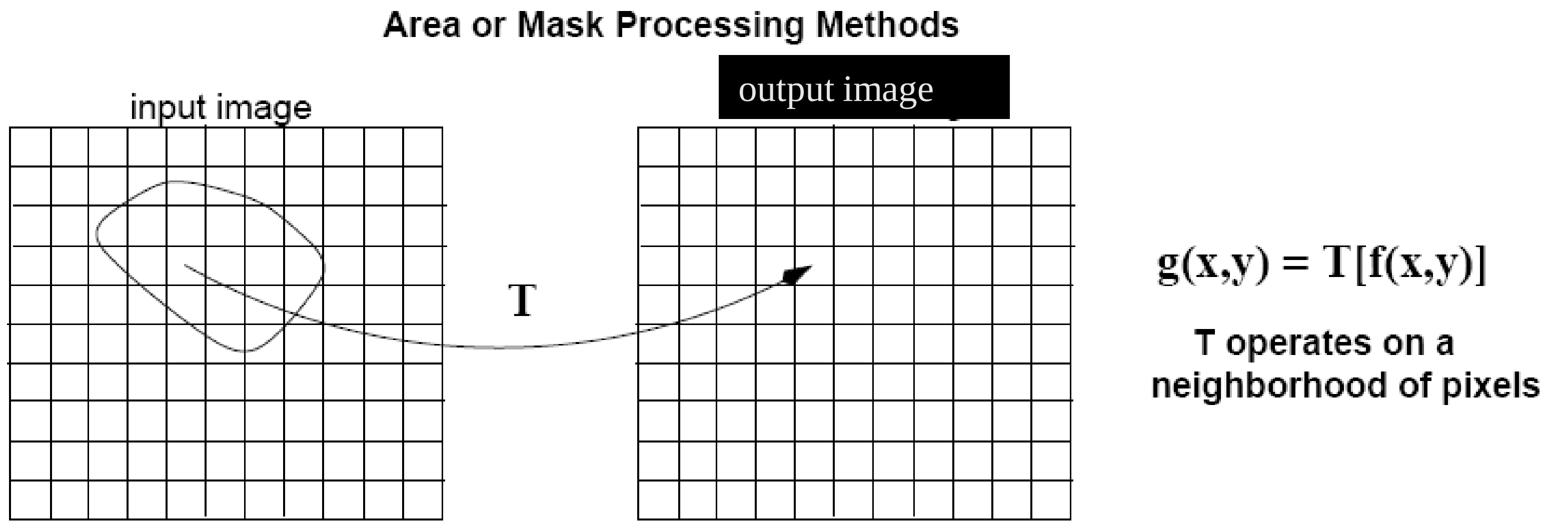


Spatial Filtering

- The word “filtering” has been borrowed from the frequency domain.
- Filters are classified as:
  - Low-pass (i.e., preserve low frequencies)
  - High-pass (i.e., preserve high frequencies)
  - Band-pass (i.e., preserve frequencies within a band)
  - Band-reject (i.e., reject frequencies within a band)

Spatial Filtering (cont'd)

- Spatial filtering is defined by:
  - **(1)** A neighborhood and a sub-image
  - **(2)** An operation that is performed on the pixels inside the neighborhood



Basics of Spatial filtering

Sometime we need to manipulate values obtained from neighboring pixels

Example: How can we compute an average value of pixels in a 3x3 region center at a pixel z?

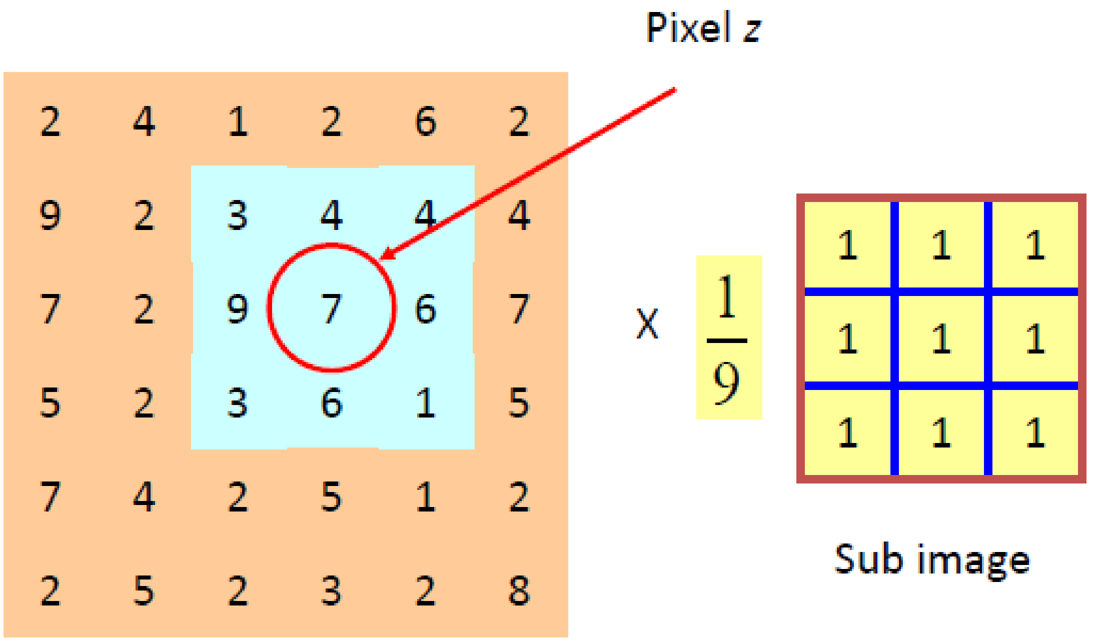
Pixel z

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

Image

Basics of Spatial Filtering contd.

Step 1: Select a pixel z.
Step 2: Select predefined neighborhood.
Step 3: select a sub image which will give u suitable result.
Step 4: Multiply elements of sub image with corresponding elements of image neighborhood



$$\rightarrow \quad y = \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 9 + \frac{1}{9}\cdot 7 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 1$$

Question: How to compute the 3x3 average values at every pixels?

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

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

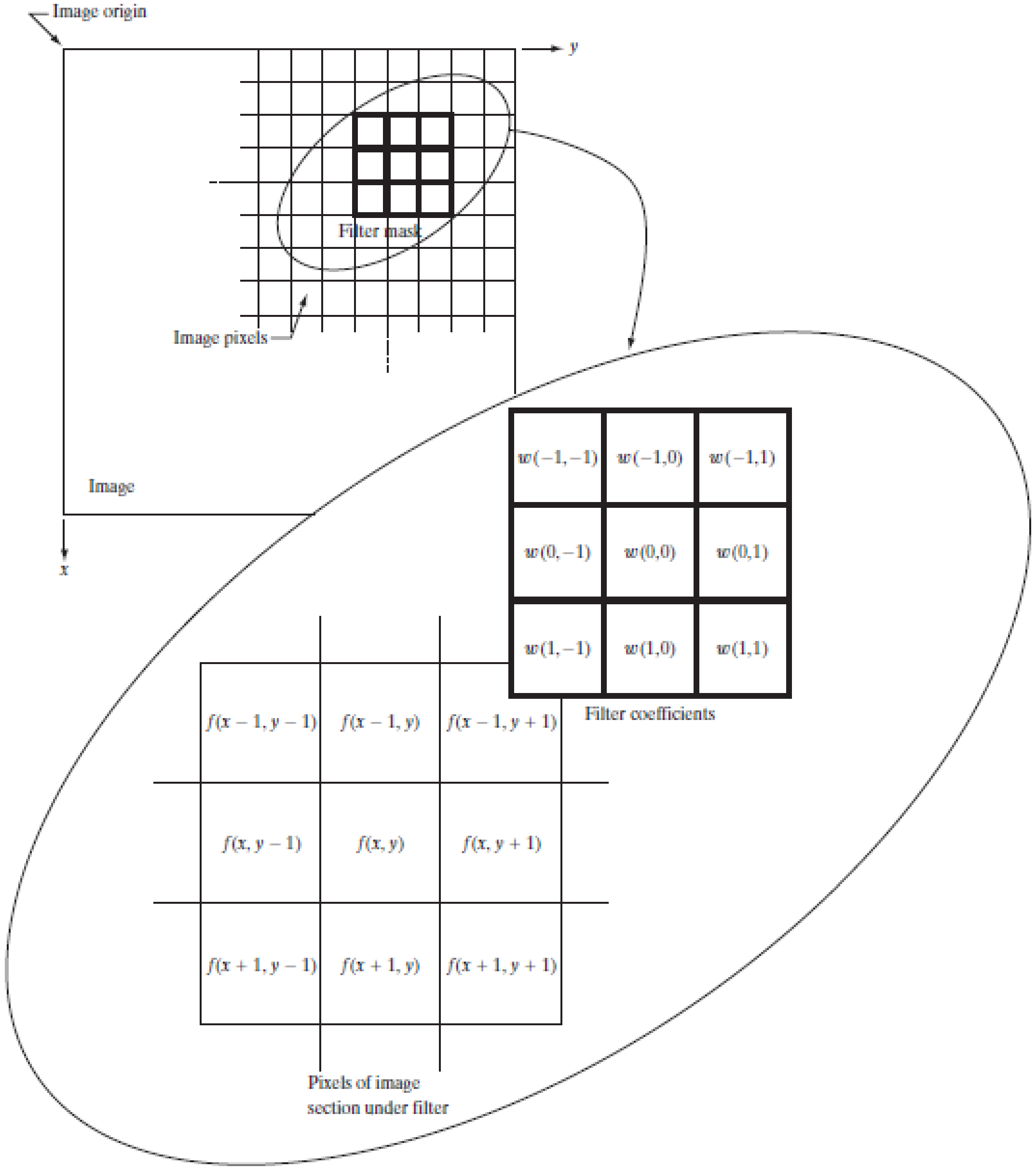
2	4	1	2	6	2
9	2	3	4	4	4
7	2	9	7	6	7
5	2	3	6	1	5
7	4	2	5	1	2
2	5	2	3	2	8

Solution: Imagine that we have a 3x3 window that can be placed everywhere on the image

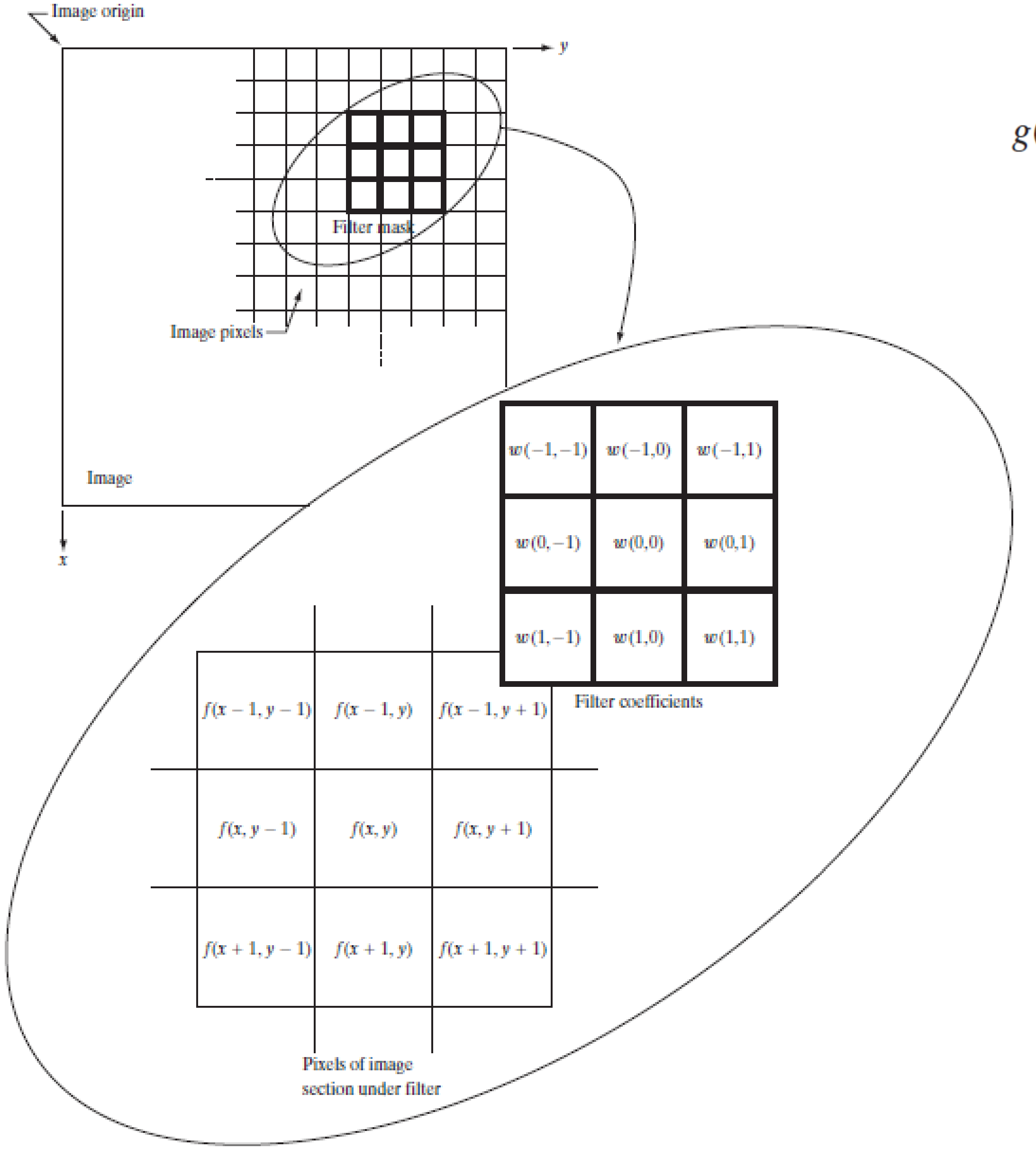
Spatial filtering:

- Select a pixel(x, y)
- Select neighborhood of predefined size of that pixel.
- Select a sub image of same size as selected neighborhood
- Multiply element of image neighborhood and corresponding element of sub-image.
- Sum of that product: linear operation.
- Replace value of (x,y) with result.

Spatial Filtering - Operation

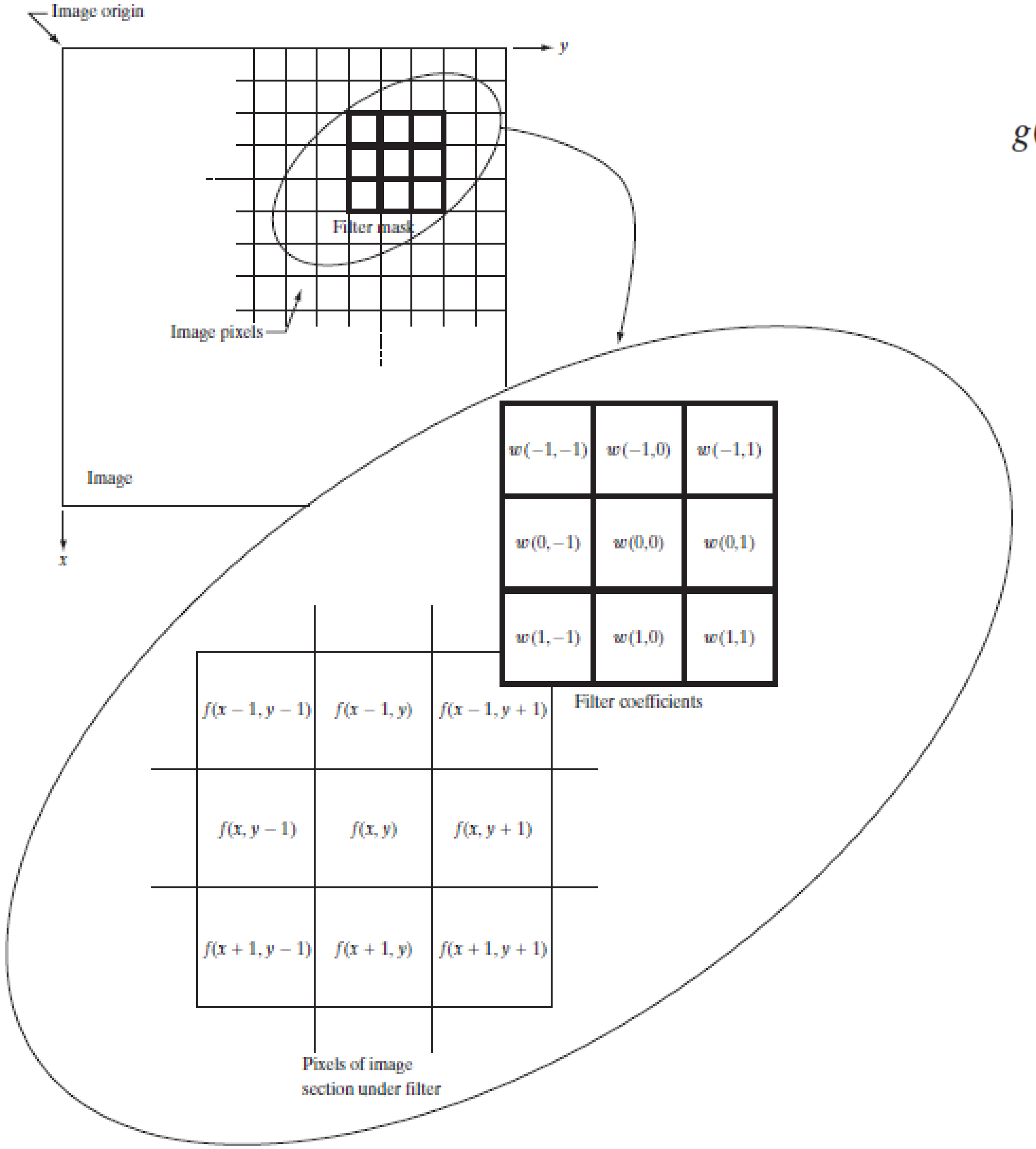


Spatial Filtering - Operation



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

Spatial Filtering - Operation

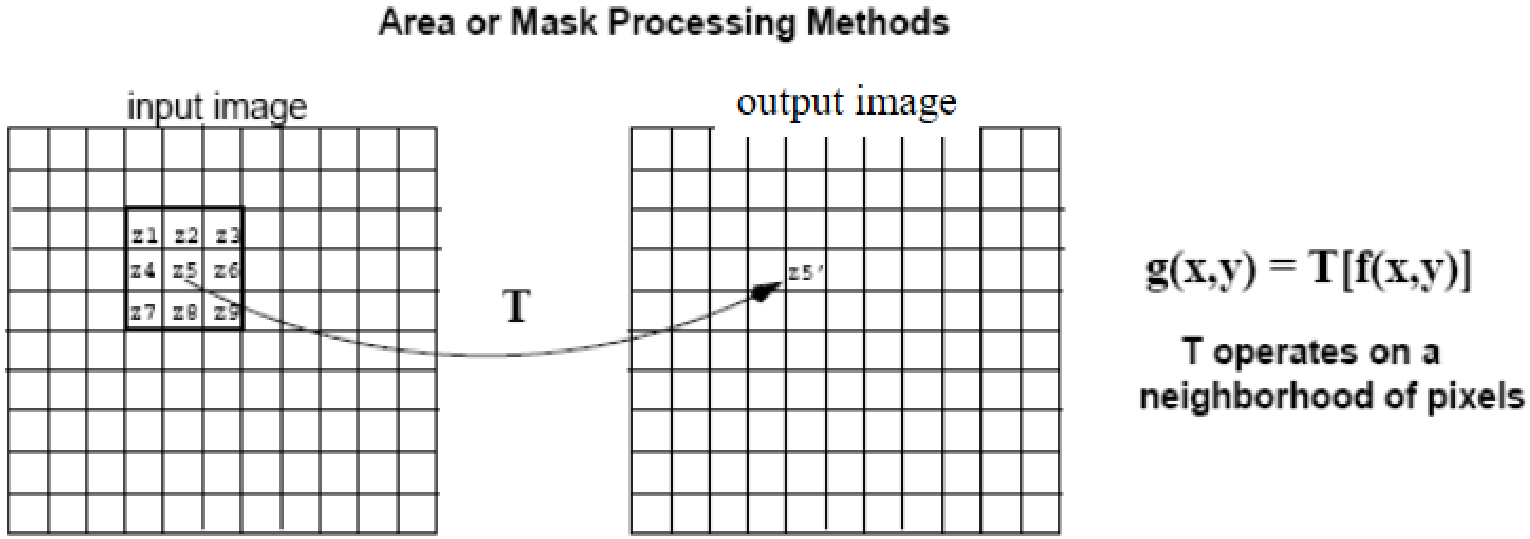


$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1)$$

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

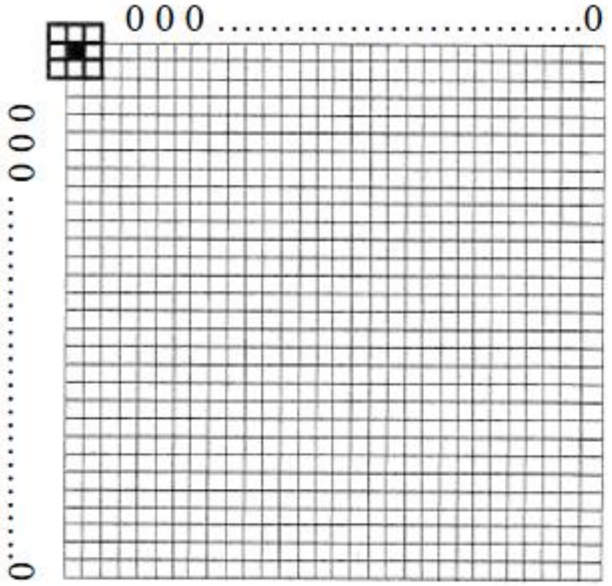
Spatial Filtering-Operation

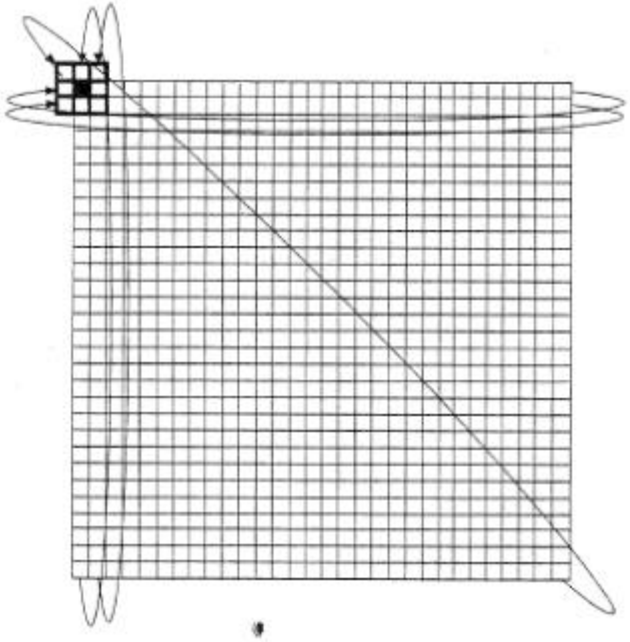
W1	W2	W3
W4	W5	W6
W7	W8	W9



A filtered image is generated as the center of the mask moves to every pixel in the input image.

Handling pixels close to boundaries





Linear vs Non-Linear

- A filtering method is linear when the output is a weighted sum of the input pixels.

W1	**W2**	**W3**
W4	**W5**	**W6**
W7	**W8**	**W9**

$$z5' = R = w1z1 + w2z2 + \cdots + w9z9$$

- Methods that do not satisfy the above property are called non-linear.
- e.g,

$$z5' = \max(z_k, k = 1,2,..,9)$$

Linear spatial filtering:

- Linear Filtering of an image f of size MxN filter mask of size mxn is given by the expression

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

where a = (m-1)/2 and b = (n-1)/2

To generate a complete filtered image this equation must be applied for x = 0, 1, 2, … , M-1 and y = 0, 1, 2, … , N-1

Linear Spatial Filtering Methods

- Correlation
- Convolution

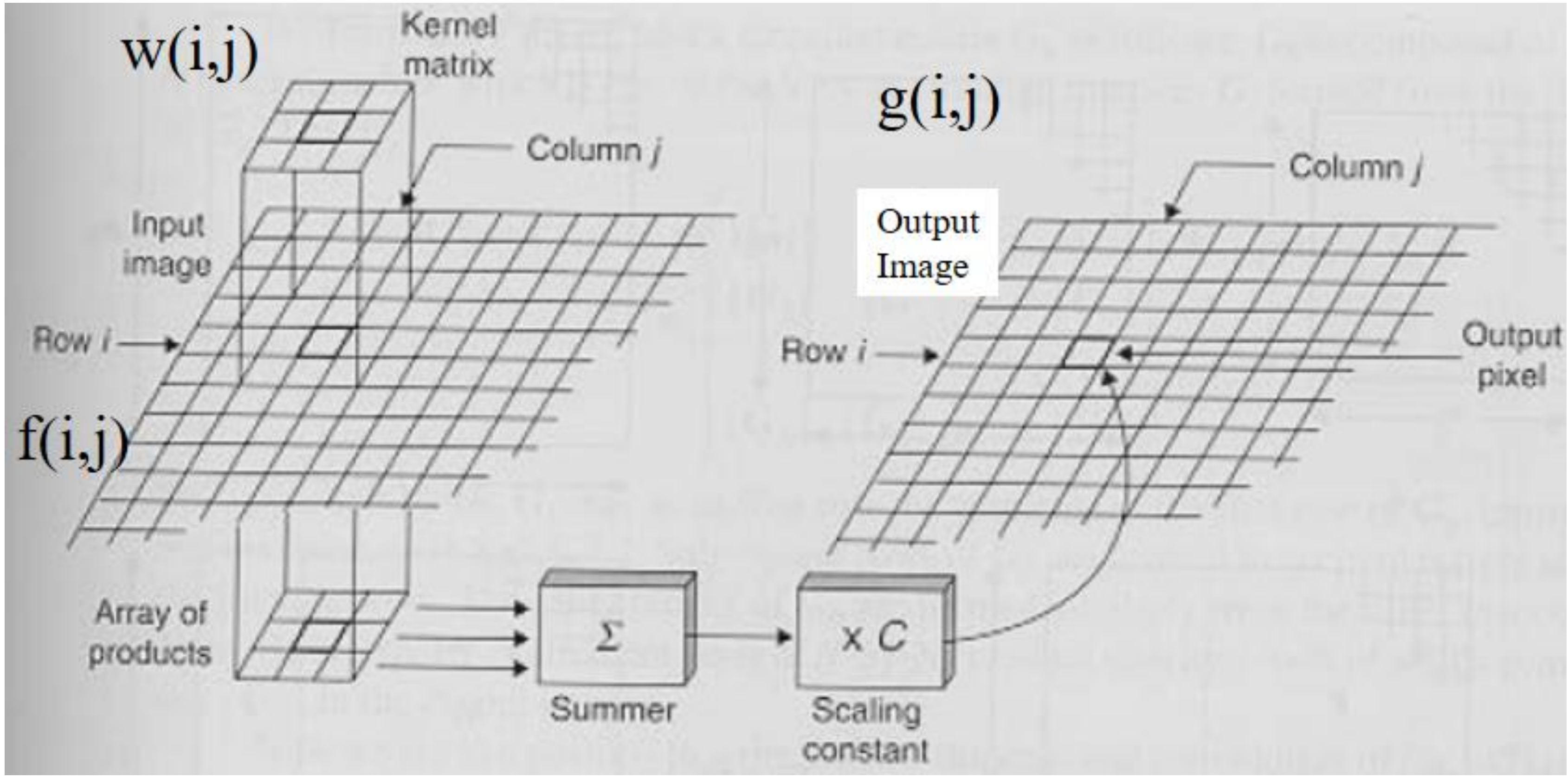


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

	Correlation		**Convolution**	
	Origin $\quad f \qquad\qquad w$		Origin $\quad f \qquad\qquad w$ rotated 180°	
(a)	0 0 0 1 0 0 0 0 $\qquad$ 1 2 3 2 8		0 0 0 1 0 0 0 0 $\qquad$ 8 2 3 2 1	(i)
(b)	$\qquad\qquad$ 0 0 0 1 0 0 0 0		$\qquad\qquad$ 0 0 0 1 0 0 0 0	(j)
	1 2 3 2 8		8 2 3 2 1	
	Starting position alignment			

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*.

Correlation		Convolution	
Origin f	w	Origin f	w rotated 180°
(a) 0 0 0 1 0 0 0 0	1 2 3 2 8	0 0 0 1 0 0 0 0	8 2 3 2 1 (i)
(b) 0 0 0 1 0 0 0 0 1 2 3 2 8 Starting position alignment		0 0 0 1 0 0 0 0 8 2 3 2 1	(j)
Zero padding (c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8		0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(k)

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*.

Correlation	Convolution
Origin f w	Origin f w rotated 180°
(a) 0 0 0 1 0 0 0 0 1 2 3 2 8	0 0 0 1 0 0 0 0 8 2 3 2 1 (i)
(b) 0 0 0 1 0 0 0 0 1 2 3 2 8 Starting position alignment	0 0 0 1 0 0 0 0 (j) 8 2 3 2 1
Zero padding (c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k) 8 2 3 2 1
(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	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l) 8 2 3 2 1

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*.

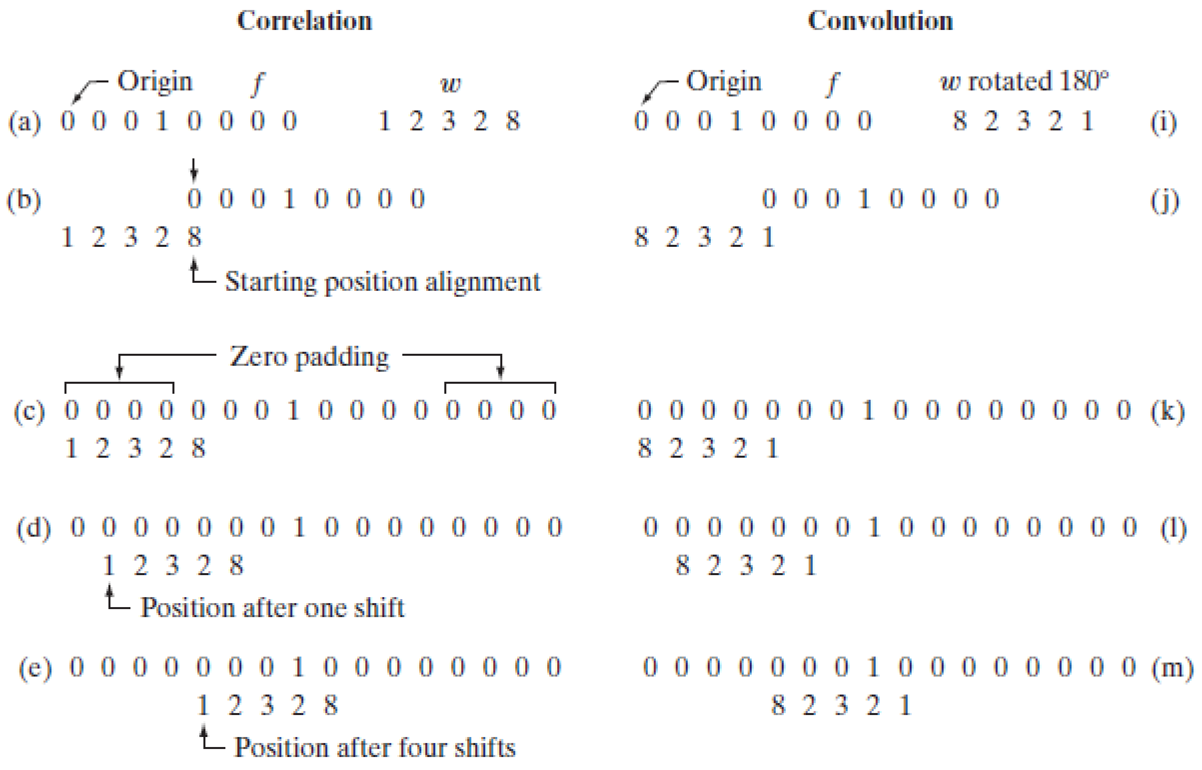


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

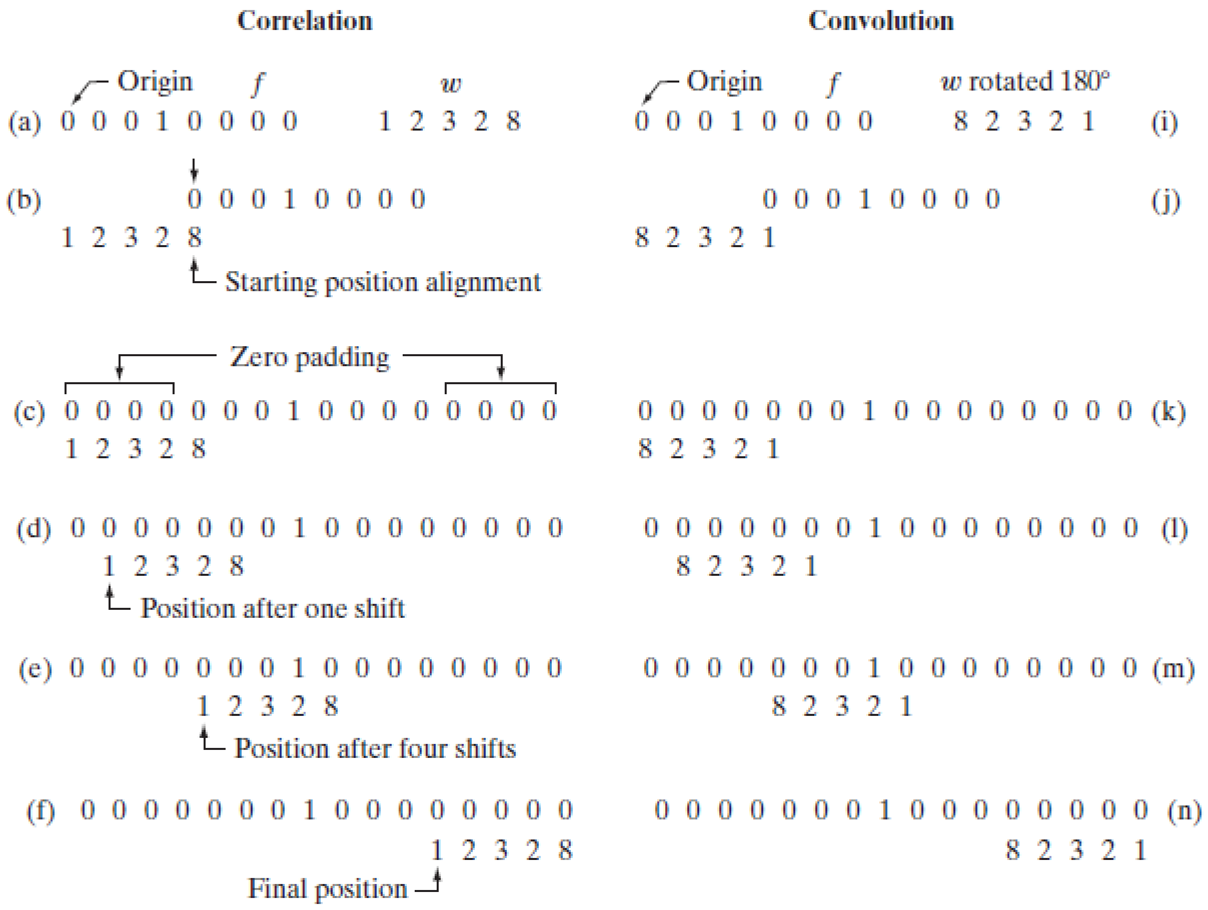


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

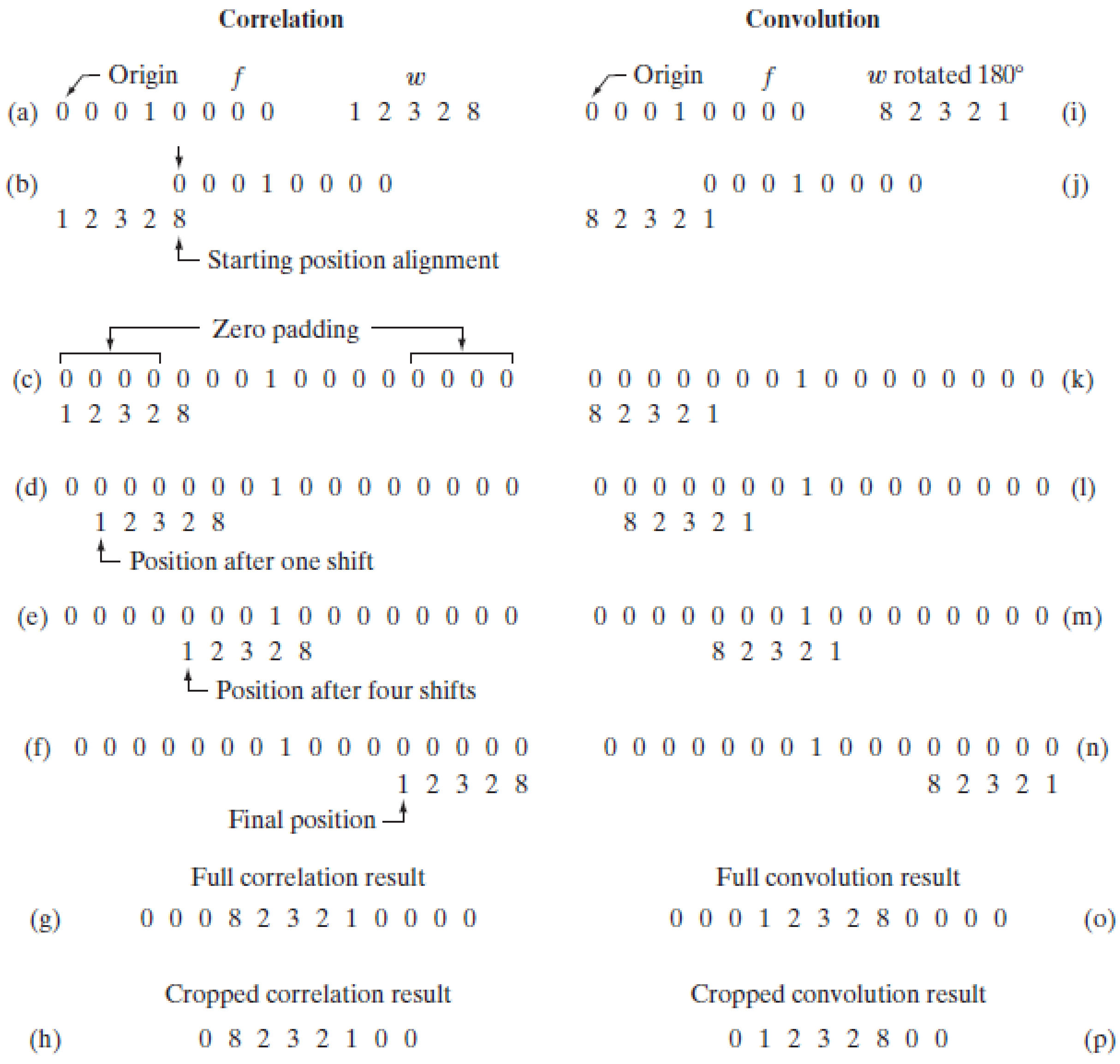


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Spatial Correlation

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t) \qquad (3.4\text{-}1)$$

Spatial Correlation

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t) \qquad (3.4\text{-}1)$$

$$w(x, y) \bigstar f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t) \qquad (3.4\text{-}2)$$

Spatial Correlation

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t) \qquad (3.4\text{-}1)$$

Spatial Convolution

$$w(x, y) \bigstar f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t) \qquad (3.4\text{-}2)$$

Spatial Correlation

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t) \qquad (3.4\text{-}1)$$

Spatial Convolution

$$w(x, y) \bigstar f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t) \qquad (3.4\text{-}2)$$

Origin $f(x, y)$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} & w(x, y) & \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

(a)

Origin $f(x, y)$

$$
\begin{array}{ccccc}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{c}
w(x, y) \\
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}
\end{array}
$$

(a)

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.

Rotation of 2D mask:

- Flip wrt one axes and then wrt other.

1	2	3
4	5	6
7	8	9

W

7	8	9
4	5	6
1	2	3

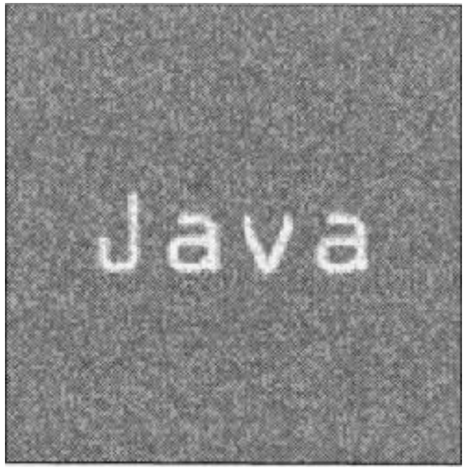
9	8	7
6	5	4
3	2	1

W1

w1(-1,-1)=w(1,1), w1(-1,0)=w(1,0) ,..., w1(1,1)=w(-1,-1)
w1(s,t)=w(-s,-t) where s and t varies from -1 to 1

Correlation contd.

- Often used in applications where we need to measure the similarity between images or parts of images (e.g., pattern matching).



Vector representation of Linear Filtering

- When interest lies in the characteristic response , R, of a mask either for correlation or convolution, it is convenient sometimes to write the sum of products as

$$
\begin{aligned}
R &= w_1 z_1 + w_2 z_2 + \ \dots \ + w_{mn} z_{mn} \\
&= \sum_{k=1}^{mn} w_k z_k \qquad\qquad (3.4\text{-}3) \\
&= \mathbf{w}^T \mathbf{z}
\end{aligned}
$$

- Where the *ws* are the coefficients of an *m* x *n* filter and the *zs* are the corresponding image intensities encompassed by the filter.

Generating Spatial Filter Masks

- Linear spatial filtering by 3 x 3 filter

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i \quad = \mathbf{w}^T \mathbf{z}$$

- Average value in 3 x 3 neighborhood

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

- Gaussian function

$$h(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Smoothing Spatial Filters

- Smoothing Filters are used for blurring and noise reduction.
- Smoothing Linear Filters:
  - The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
  - These filters sometimes are called *averaging filters*.
  - They also are referred to a *lowpass filters*.
  - The idea behind smoothing filters is straightforward. 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, this process results in an image with reduced “sharp” transitions in intensities. Because random noise typically consists of sharp transitions in intensity levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp intensity transitions, so averaging filters have the undesirable side effect that they blur edges

Smoothing spatial filters:

- Linear spatial filters for smoothing:
  averaging filters, lowpass filters
  - Noise reduction
  - Undesirable side effect: blur edges

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Standard average → Box filter

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

weighted average

Which one will have less blurring??

Spatial Smoothing Linear Filter

The general implementation for filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ is given

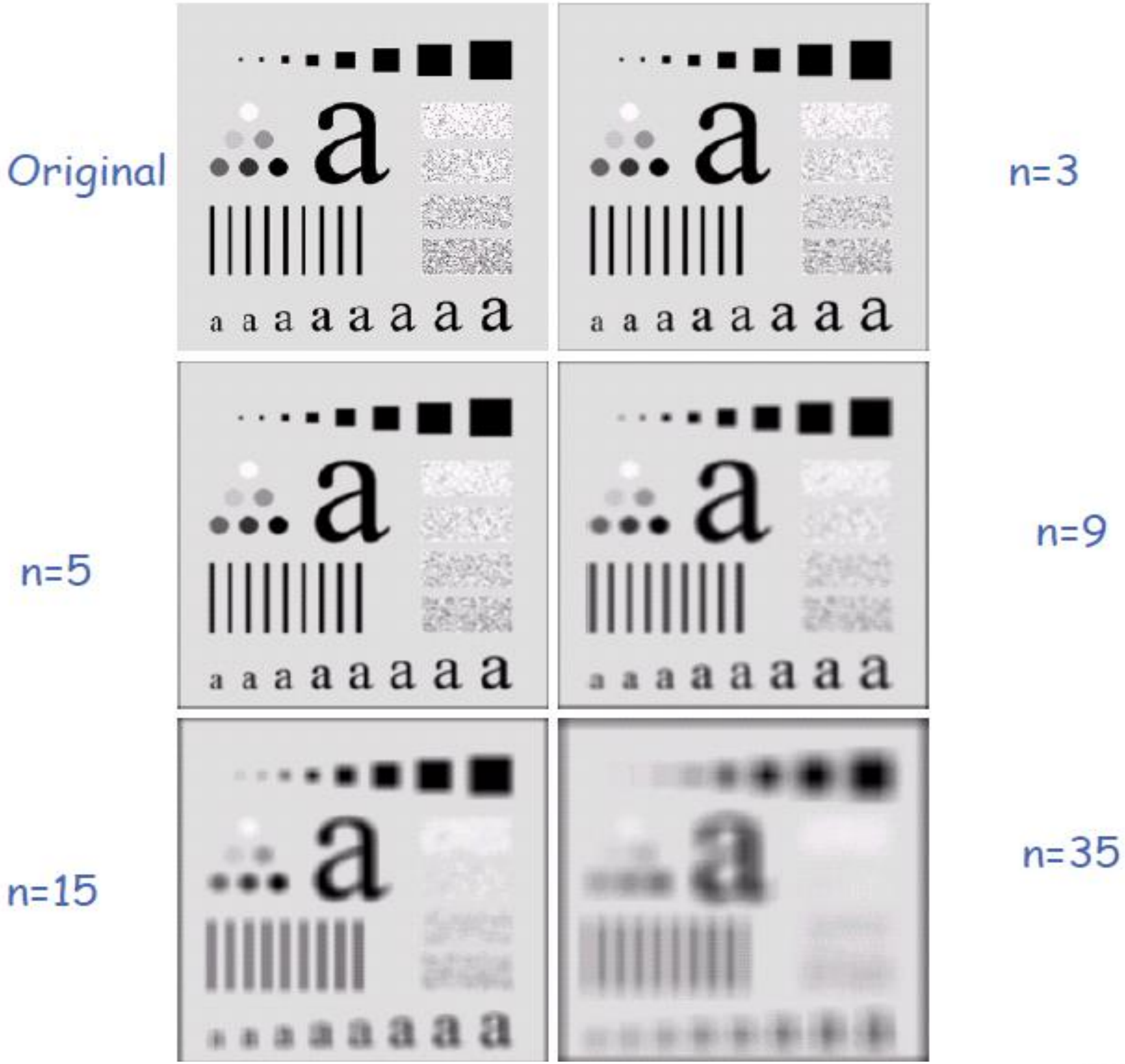
$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

where $m = 2a+1, \quad n = 2b+1.$

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

a b

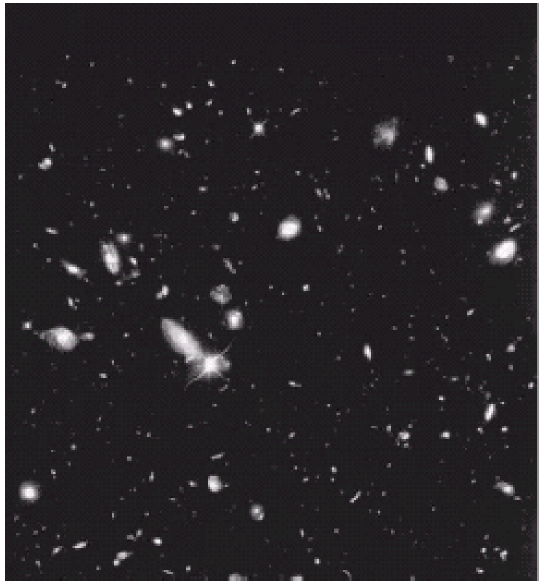
FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.



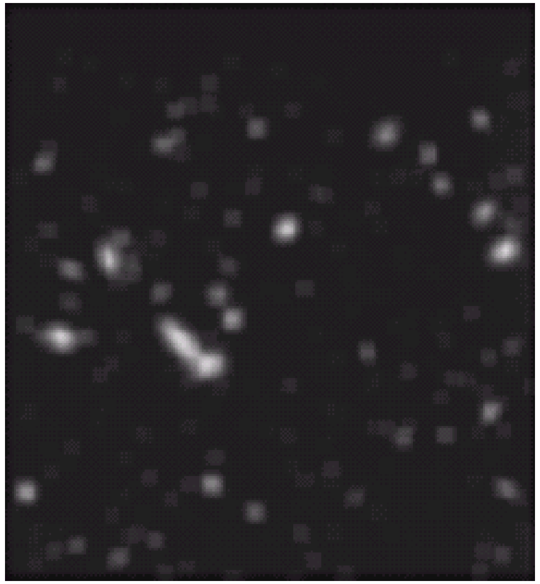
Result of smoothing with square averaging filter masks

Another Smoothing Example

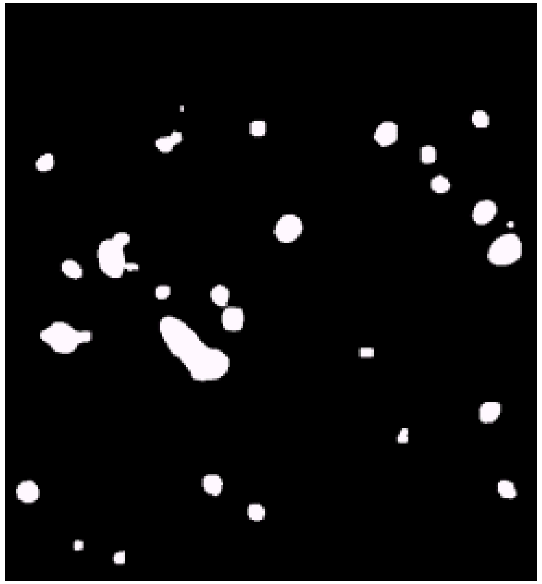
By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding



Original Image



Smoothed Image 15x15 mask

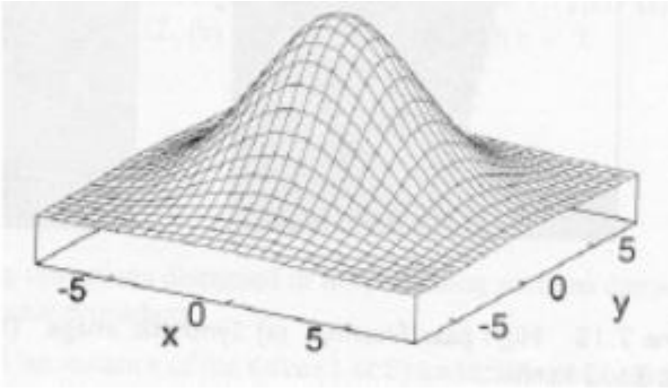


Thresholded Image

Smoothing Filters: Gaussian

- The weights are samples of the Gaussian functions.

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2 + y^2}{2\sigma^2}}$$



7×7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

σ = 1.4

Smoothing Filters: Gaussian contd.

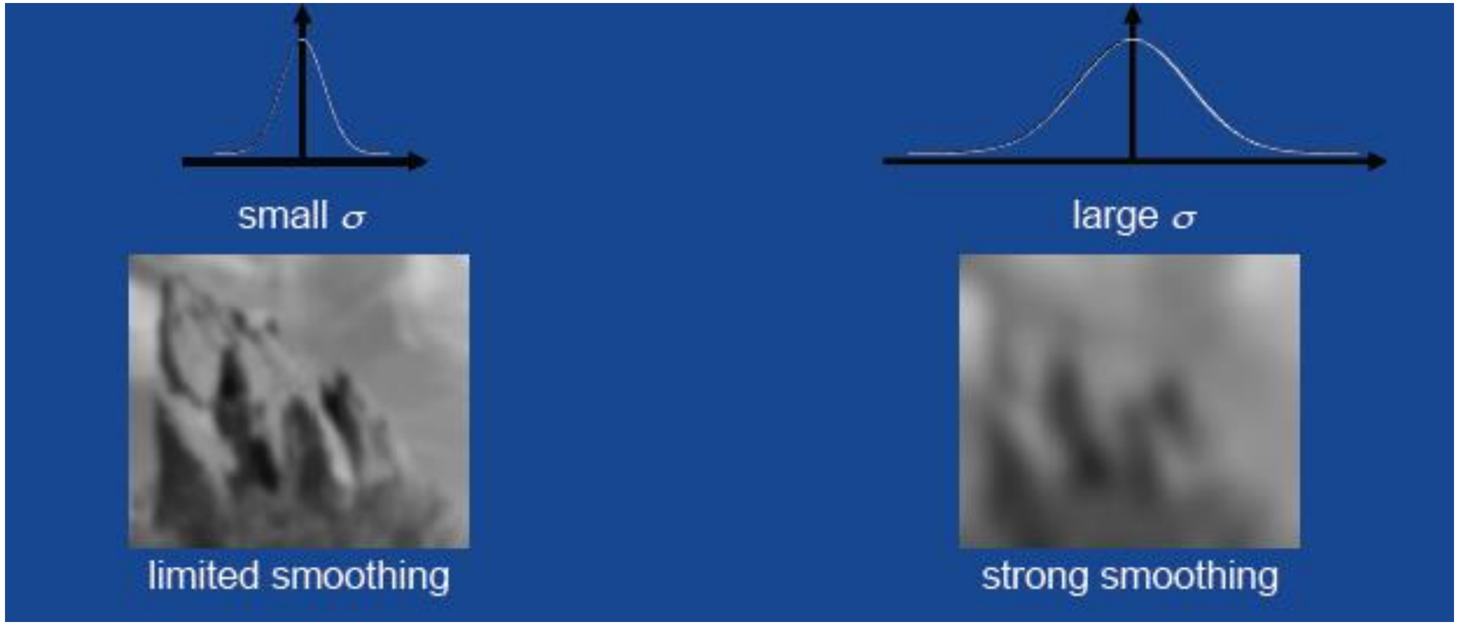
- **σ** controls the amount of smoothing
- As **σ** increases, more samples must be obtained to represent the Gaussian function accurately.

σ = 3

15 × 15 Gaussian mask

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2

Smoothing Filters: Gaussian contd.



Averaging vs Gaussian Blur

Averaging

Gaussian

Order-Statistic (Nonlinear Filters)

- Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels.
- Median Filter
  - Replaces the pixel value by the median of the gray levels in the neighborhood of that pixel
  - Effective for impulse noise (salt-and-pepper noise)
  - 3x3 neighborhood: 5th largest value
  - 5x5 neighborhood: 13th largest value
  - Isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$, are eliminated by an n x n median filter
- Max filter: select maximum value in the neighborhood
- Min filter: select minimum value in the neighborhood

Smoothing Filters: Median Filtering (non-linear)

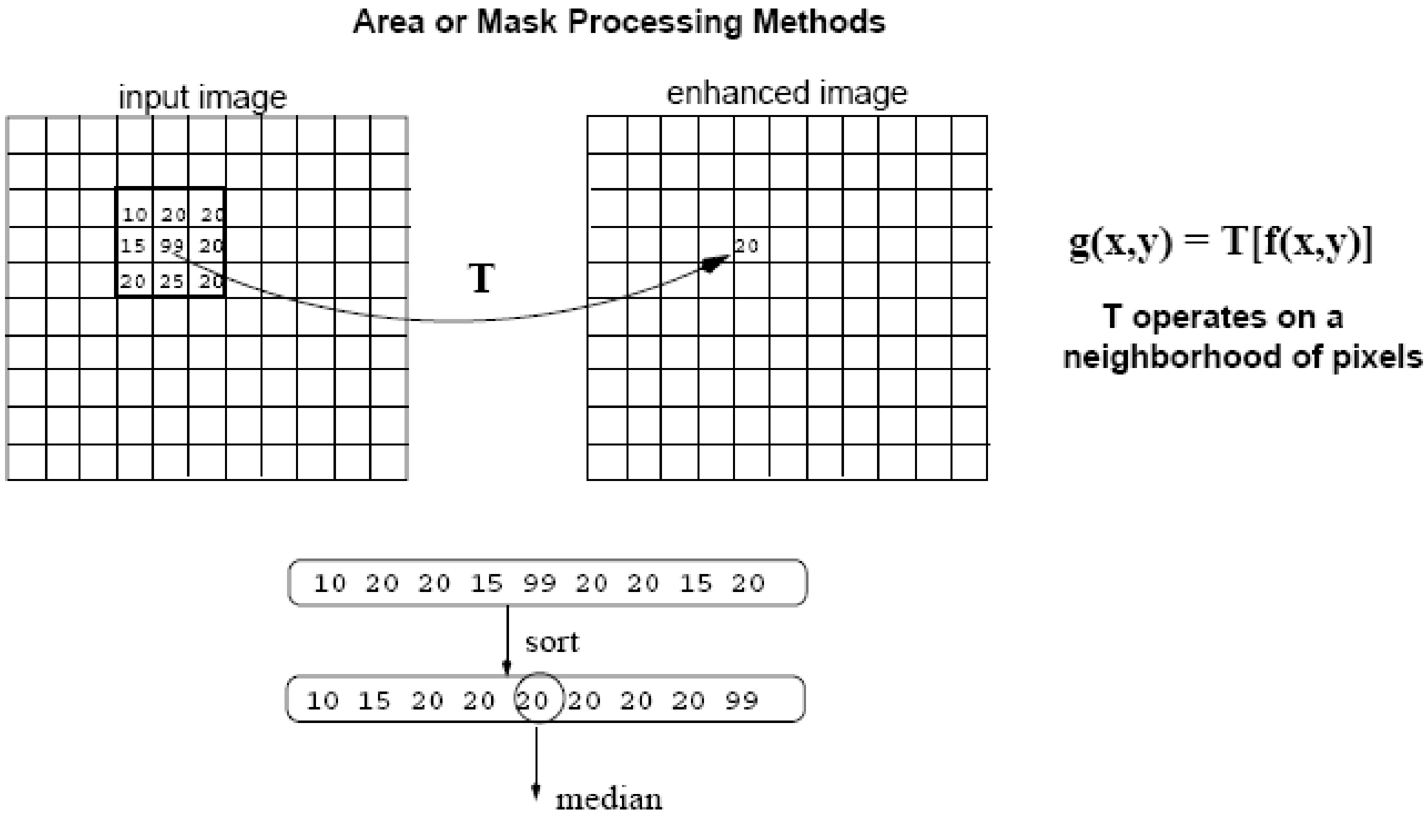
- Very effective for removing "salt and pepper" noise (i.e., random occurrences of black and white pixels).

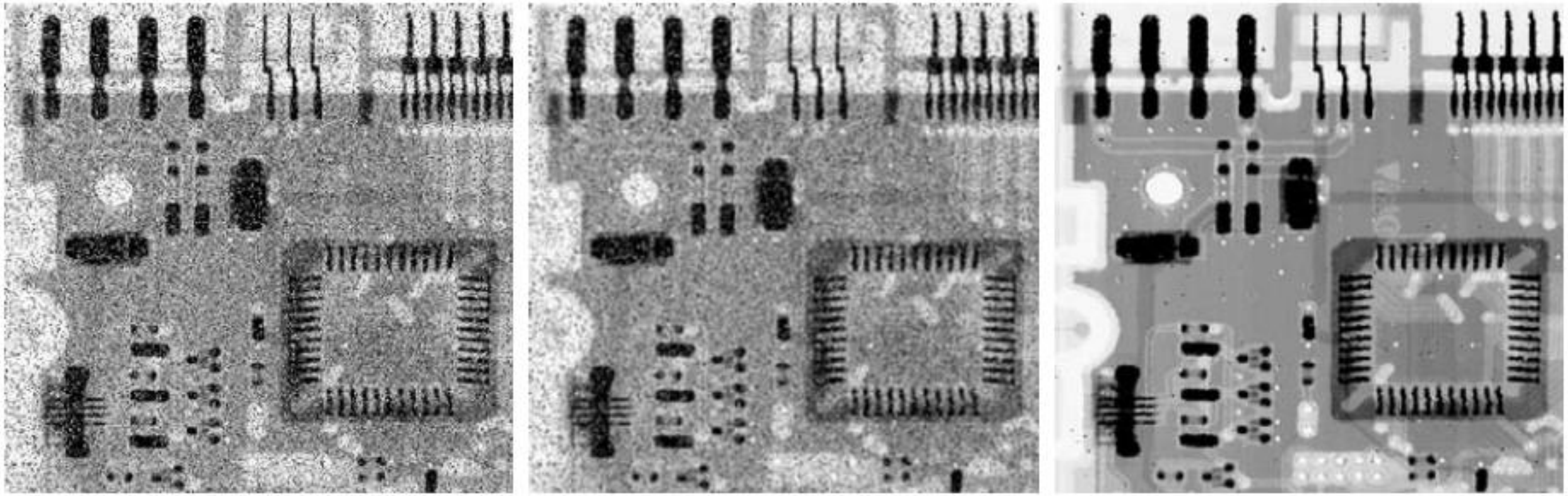
averaging

median filtering

Smoothing Filters: Median Filtering (cont'd)

- Replace each pixel by the median in a neighborhood around the pixel.





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.)

Sharpening Spatial Filters

- The principal objective of sharpening is to highlight transitions in intensity.
- In the last section, we saw that image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood.
- Because averaging is analogous to integration, it is logical to conclude that sharpening can be accomplished by spatial differentiation.
- Fundamentally, the strength of the response of a derivative operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied.
- Thus, image differentiation enhances edges and other discontinuities (such as noise) and de-emphasizes areas with slowly varying intensities.

Foundation

- The derivatives of a digital function are defined in terms of differences.
- There are various ways to define these differences. However, we require that
- any definition we use for a *first 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.
- Similarly, any definition of a second derivative
  (1) must be zero in constant areas;
  (2) must be nonzero at the onset *and* end of an intensity step or ramp; and
  (3) must be zero along ramps of constant slope.
- Because we are dealing with digital quantities whose values are finite, the maximum possible intensity change also is finite, and the shortest distance over which that change can occur is between adjacent pixels.

Foundation

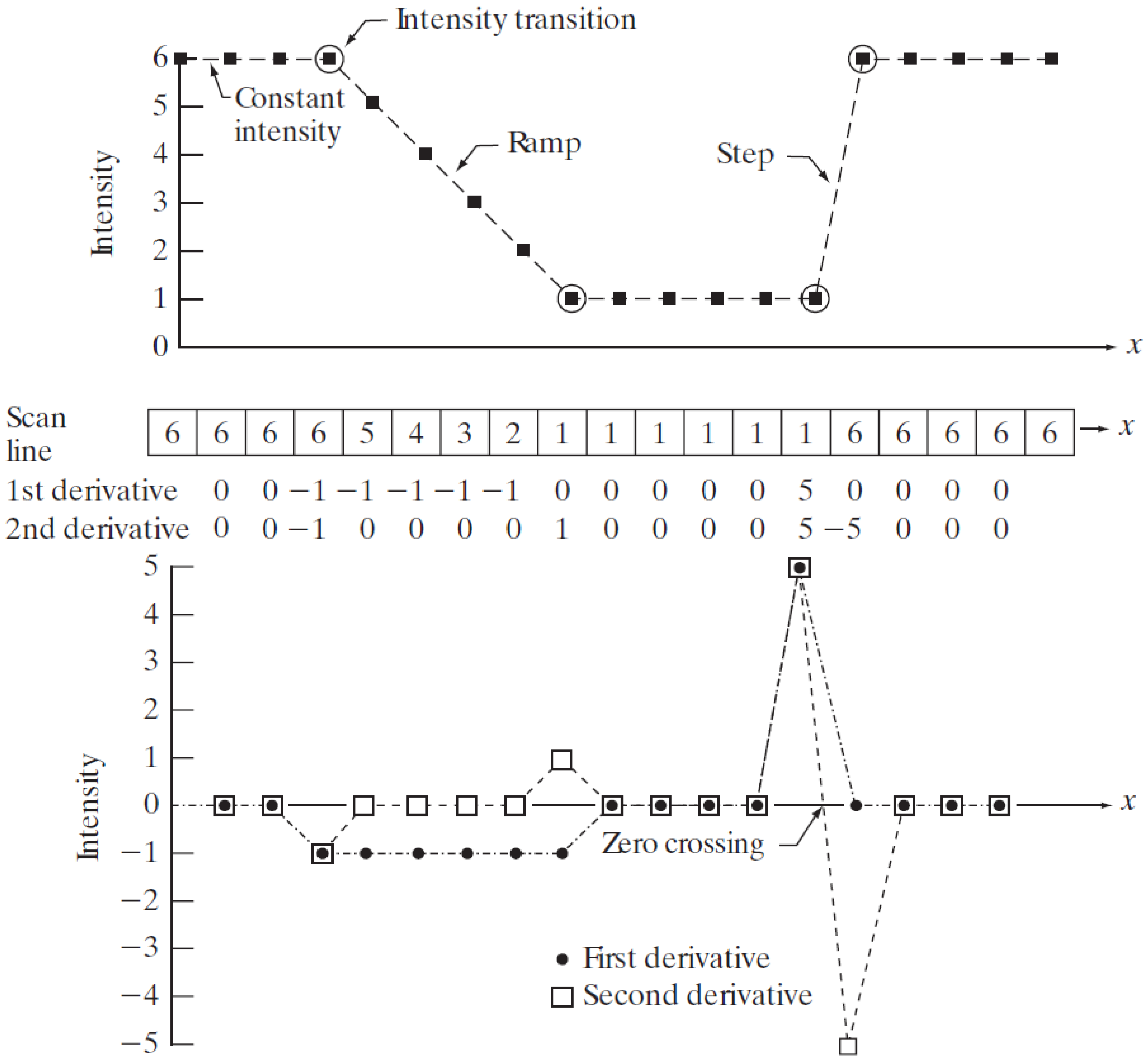
$$\frac{\partial f}{\partial x}(x) = f(x+1) - f(x)$$

first-order derivative

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

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

second-order derivative



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.

Using the second derivative for image sharpening – The Laplacian

- Isotropic filters: rotation invariant
- Simplest isotropic second-order derivative operator: Laplacian

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

2-D Laplacian operation

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

x-direction

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

y-direction

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

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.

Using the second derivative for image sharpening – The Laplacian

- Image enhancement (sharpening) by Laplacian operation

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

$$g(x,y)=f(x,y)+c\left[\nabla^2 f(x,y)\right]$$

where,

$f(x,y)$ is input image,

$g(x,y)$ is sharpenend images,

$c=$ -1 if $\nabla^2 f(x,y)$ corresponding to Fig. 3.37(a) or (b)

and $c=1$ if either of the other two filters is used.

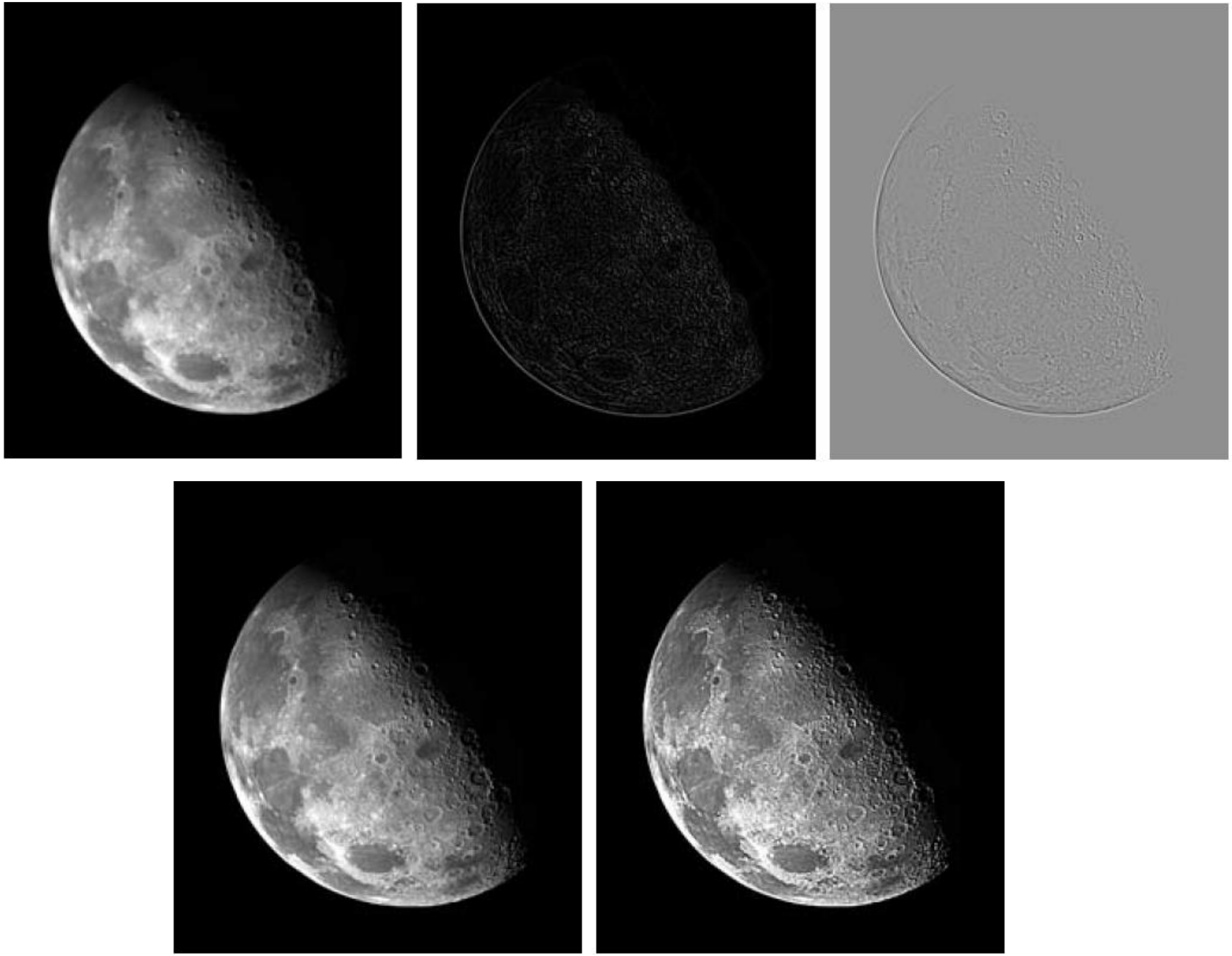


FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

Unsharp Masking and Highboost Filtering

Unsharp Masking and Highboost Filtering

- Unsharp masking

  Sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image

  e.g., printing and publishing industry

- Steps

  1. Blur the original image
  2. Subtract the blurred image from the original
  3. Add the mask to the original

Unsharp Masking and Highboost Filtering

Let $\bar{f}(x, y)$ denote the blurred image, unsharp masking is

$$g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

Then add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{mask}(x, y) \qquad k \geq 0$$

Unsharp Masking and Highboost Filtering

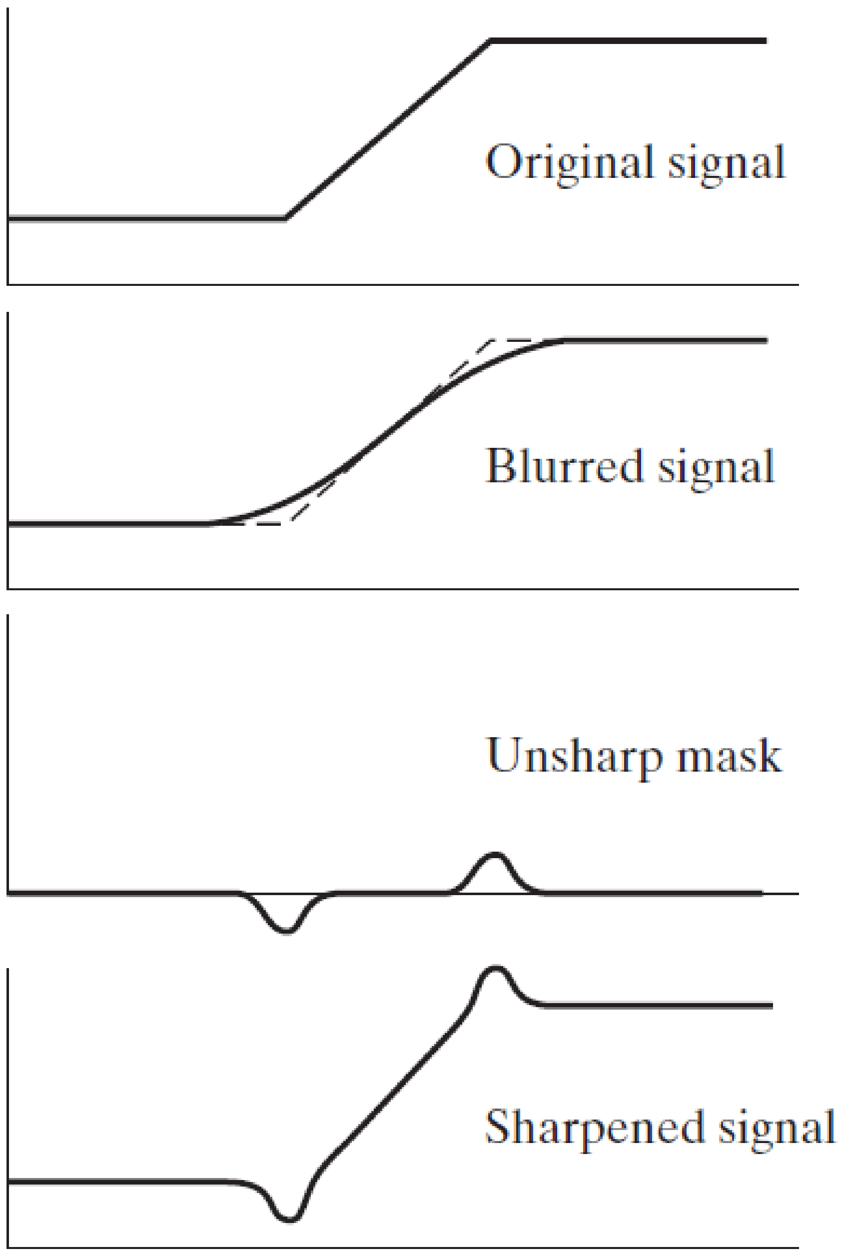
Let $\overline{f}(x, y)$ denote the blurred image, unsharp masking is

$$g_{mask}(x, y) = f(x, y) - \overline{f}(x, y)$$

Then add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{mask}(x, y) \qquad k \geq 0$$

when $k > 1$, the process is referred to as highboost 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).

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.

a) Unretouched “soft-tone” digital image of size 469x600 pixels.
b) Image blurred using 31 x 31 Gaussian lowpass filter with σ = 5.
c) Mask.
d) Result of unsharp masking using eqn 3.6-8 with *k* = 1
e) Results of highboost filtering with *k* = 2
f) Results of highboost filtering with *k* = 3

First Derivative – The Gradient

For function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^{\ 2} + g_y^{\ 2}}$$

Image Sharpening based on First-Order Derivatives

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x{}^2 + g_y{}^2}$$

Image Sharpening based on First-Order Derivatives

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^{\;2} + g_y^{\;2}}$$

$$M(x, y) \approx |g_x| + |g_y|$$

Image Sharpening based on First-Order Derivatives

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x,y) = \text{mag}(\nabla f) = \sqrt{g_x{}^2 + g_y{}^2}$$

$$M(x,y) \approx |g_x| + |g_y|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image Sharpening based on First-Order Derivatives

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^{\,2} + g_y^{\,2}}$$

$$M(x, y) \approx |g_x| + |g_y|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$M(x, y) = |z_8 - z_5| + |z_6 - z_5|$$

Image Sharpening based on First-Order Derivatives

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image Sharpening based on First-Order Derivatives

Roberts Cross-gradient Operators

$$M(x, y) \approx |z_9 - z_5| + |z_8 - z_6|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image Sharpening based on First-Order Derivatives

Roberts Cross-gradient Operators

$$M(x, y) \approx | z_9 - z_5 | + | z_8 - z_6 |$$

Sobel Operators

$$M(x, y) \approx | (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) | \\ + | (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) |$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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.

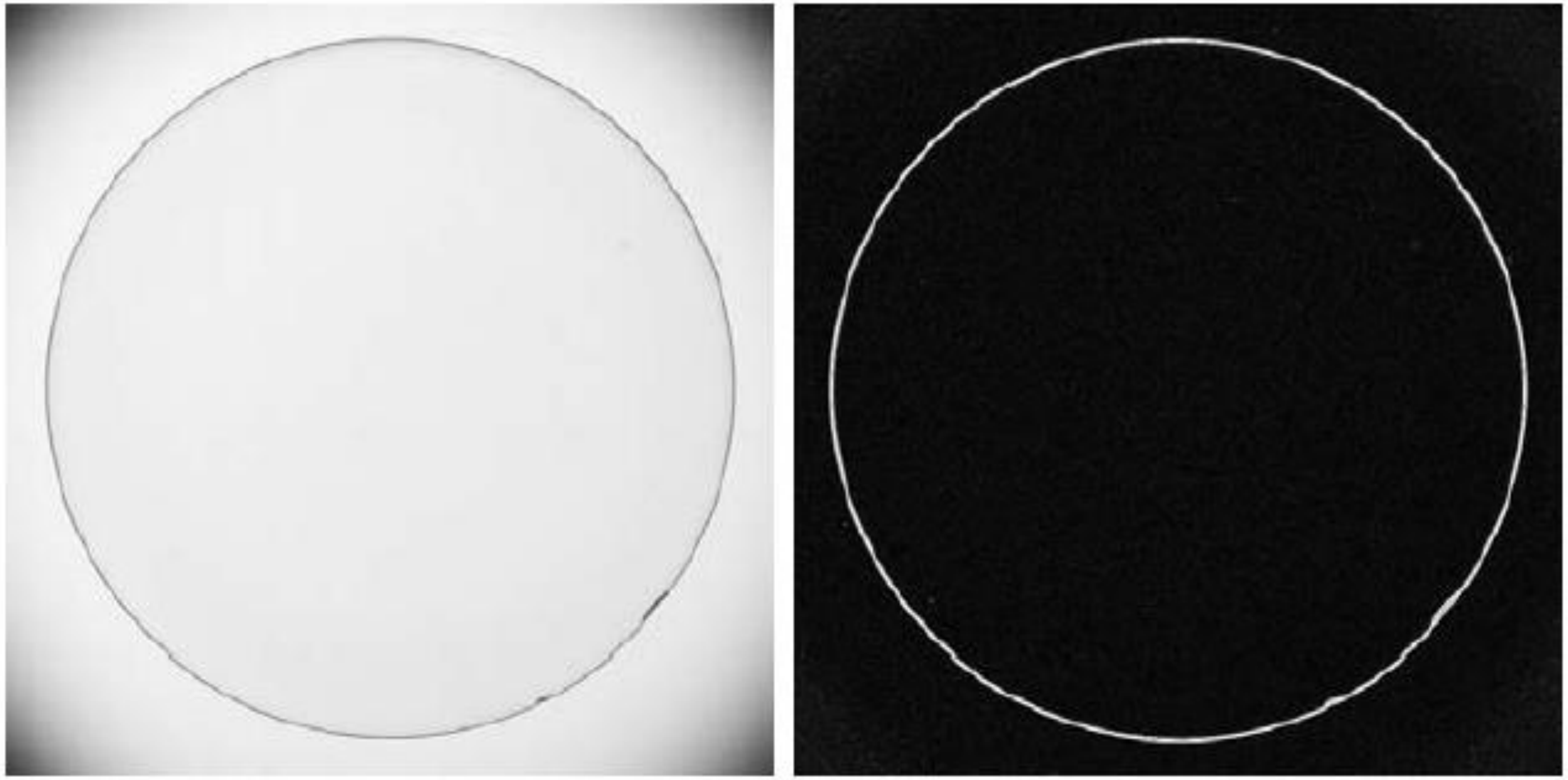
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−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



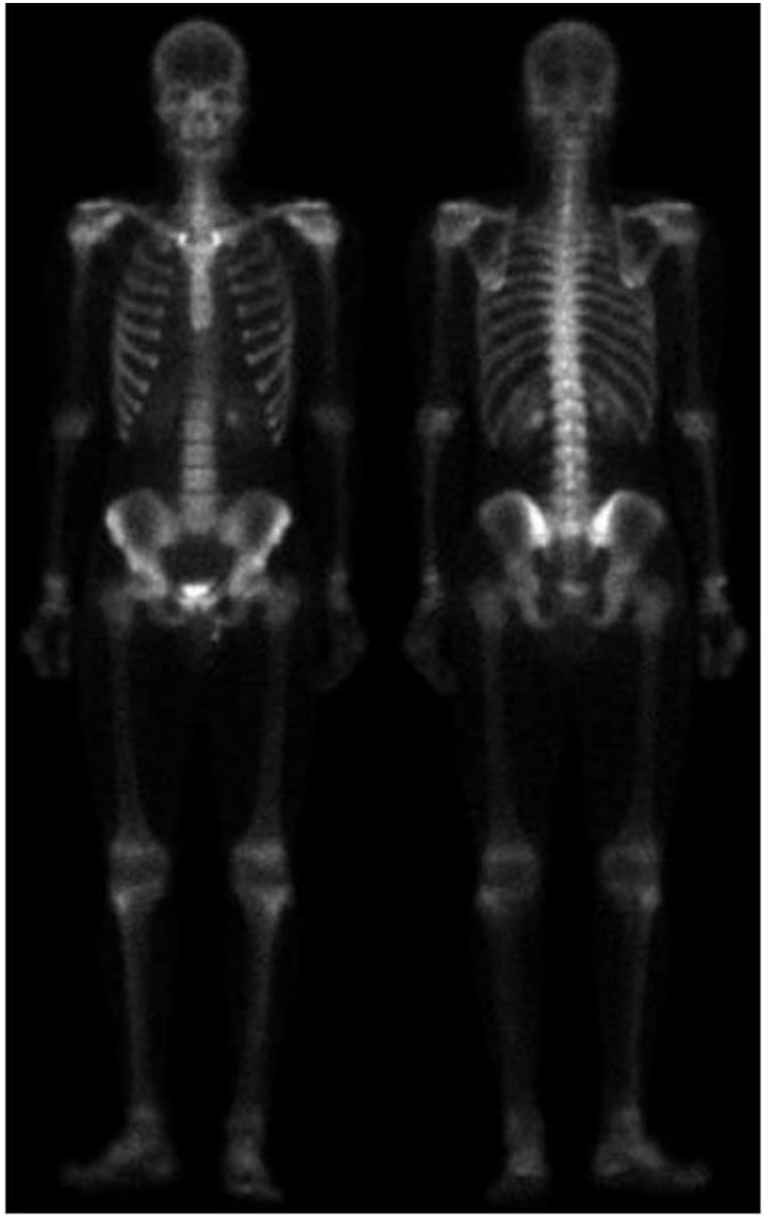
a b

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

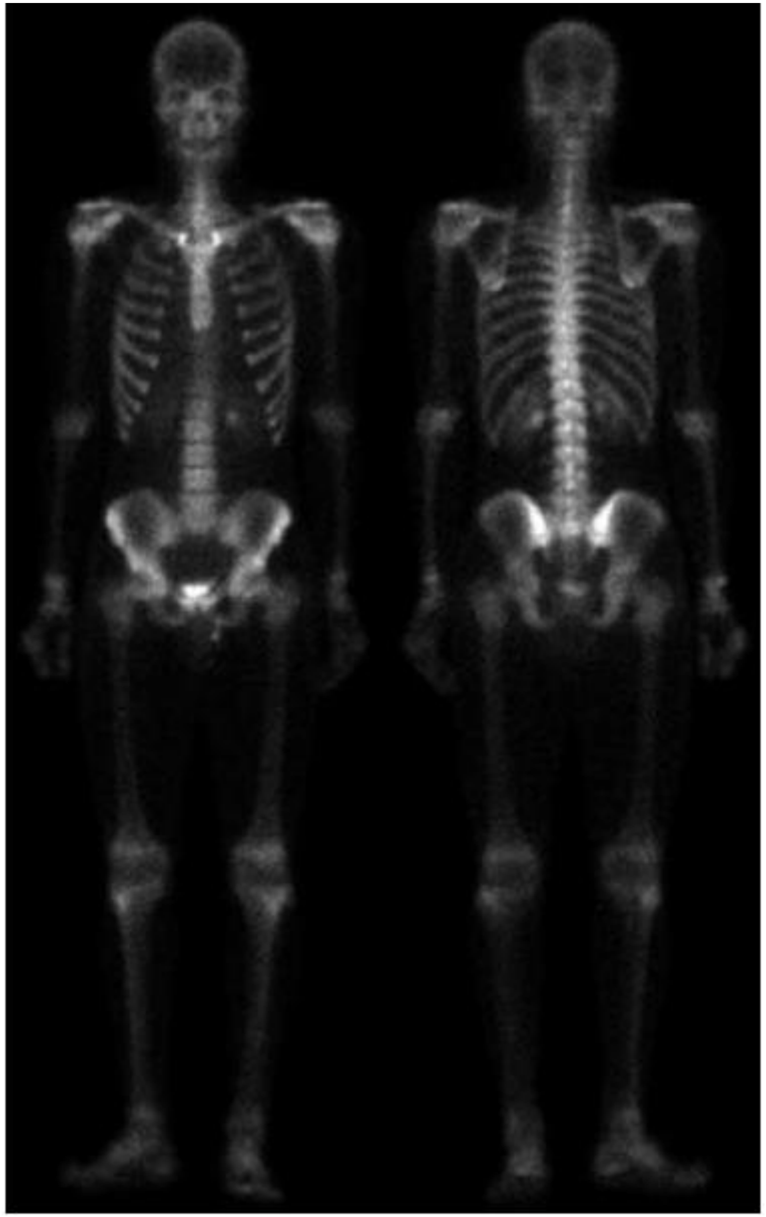
- With a few exceptions, like combining blurring with thresholding (Fig. 3.34), we have focused attention thus far on individual approaches.
- Frequently, a given task will require application of several complementary techniques in order to achieve an acceptable result.
- In this section we illustrate by means of an example how to combine several of the approaches developed thus far in this chapter to address a difficult image enhancement task.

[Reference](#)



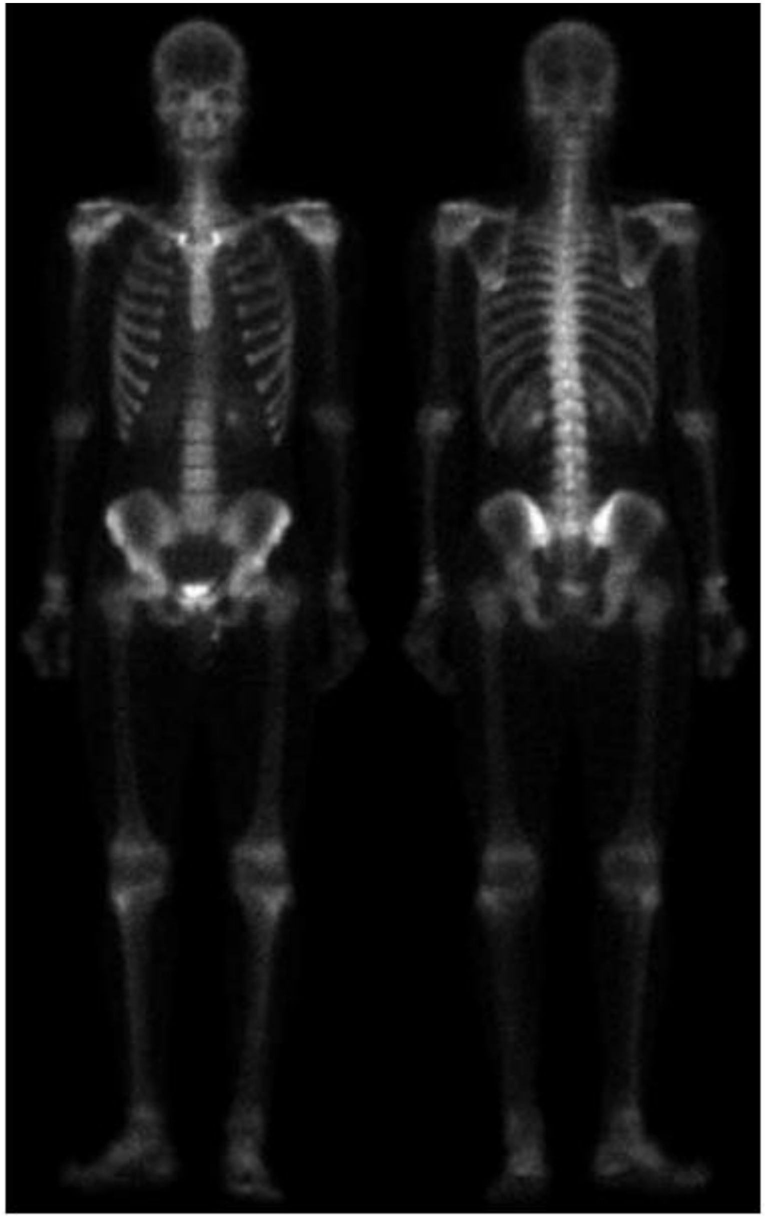
Reference

(a) Image of whole body bone scan

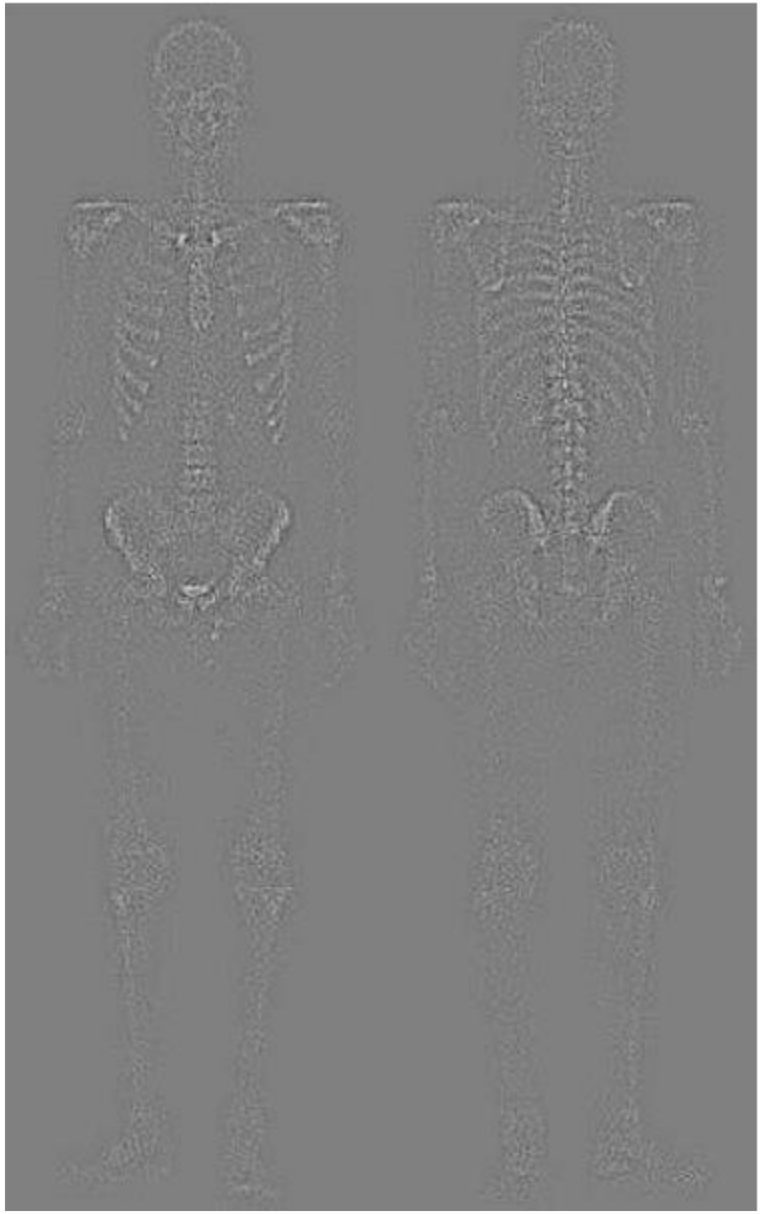


[Reference](#)

(a) Image of whole body bone scan

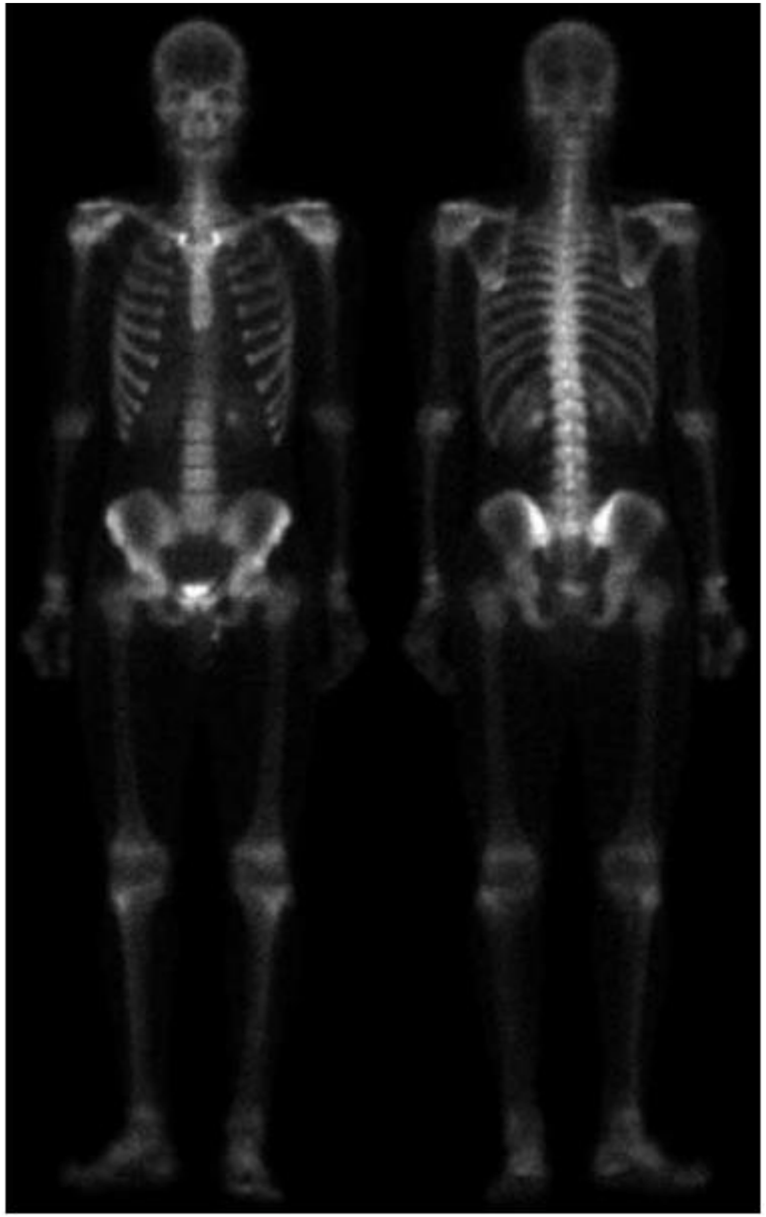


(b) Laplacian of (a)

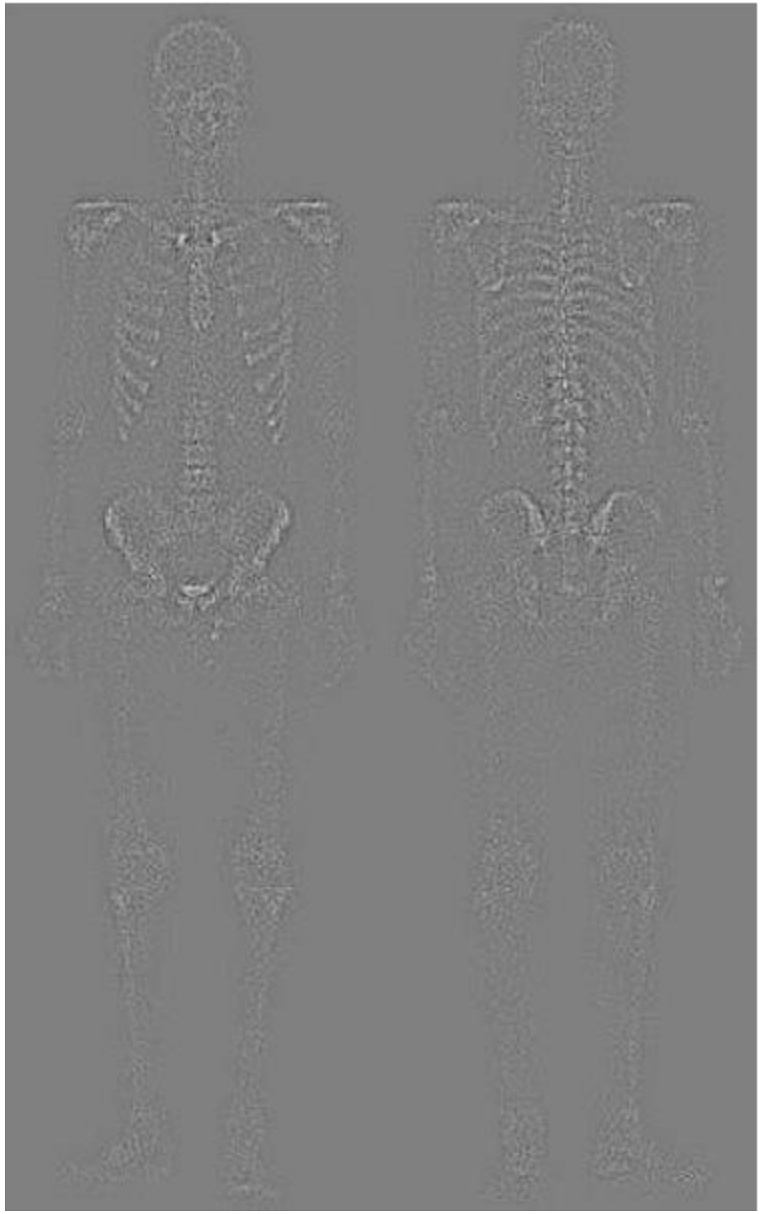


Reference

(a) Image of whole body bone scan



(b) Laplacian of (a)



Reference

(c) Sharpened image obtained by adding (a) and (b)

(c) Sharpened image obtained by adding (a) and (b)

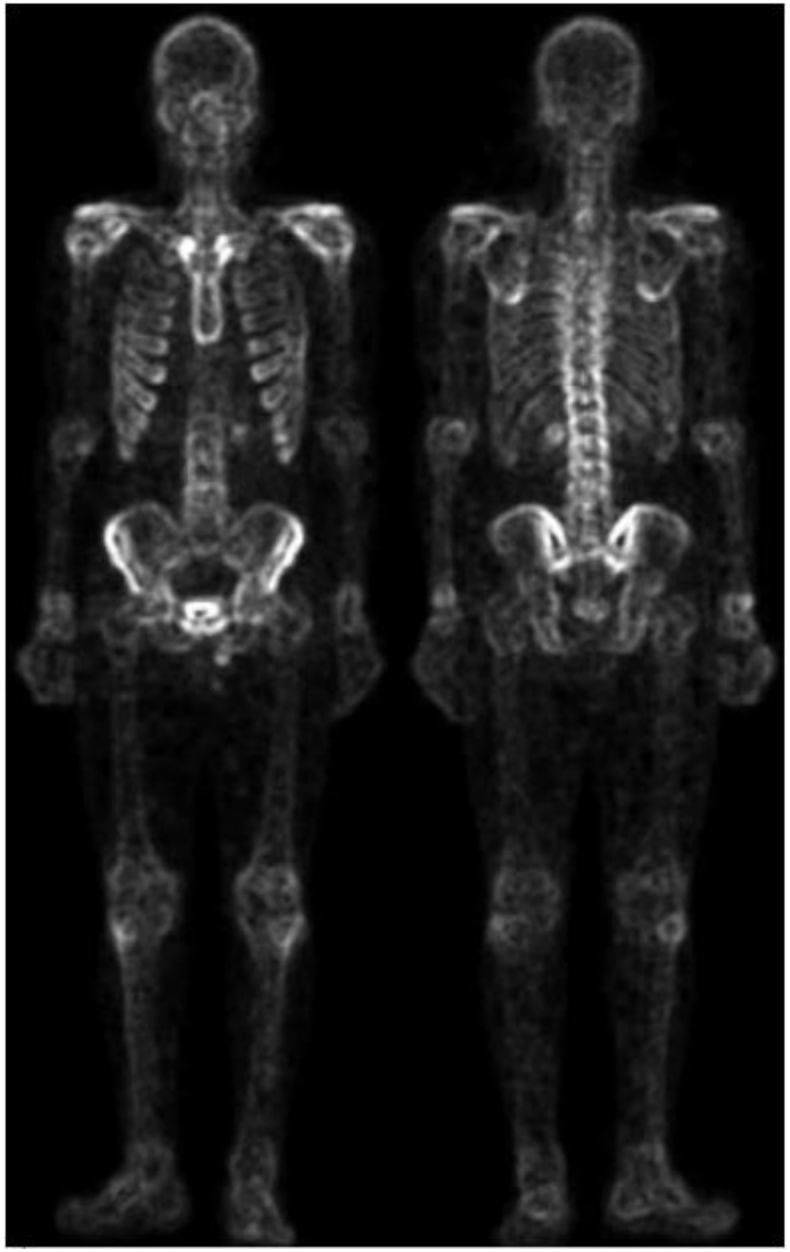
(c) Sharpened image obtained by adding (a) and (b)

(d) Sobel Gradient of (a)

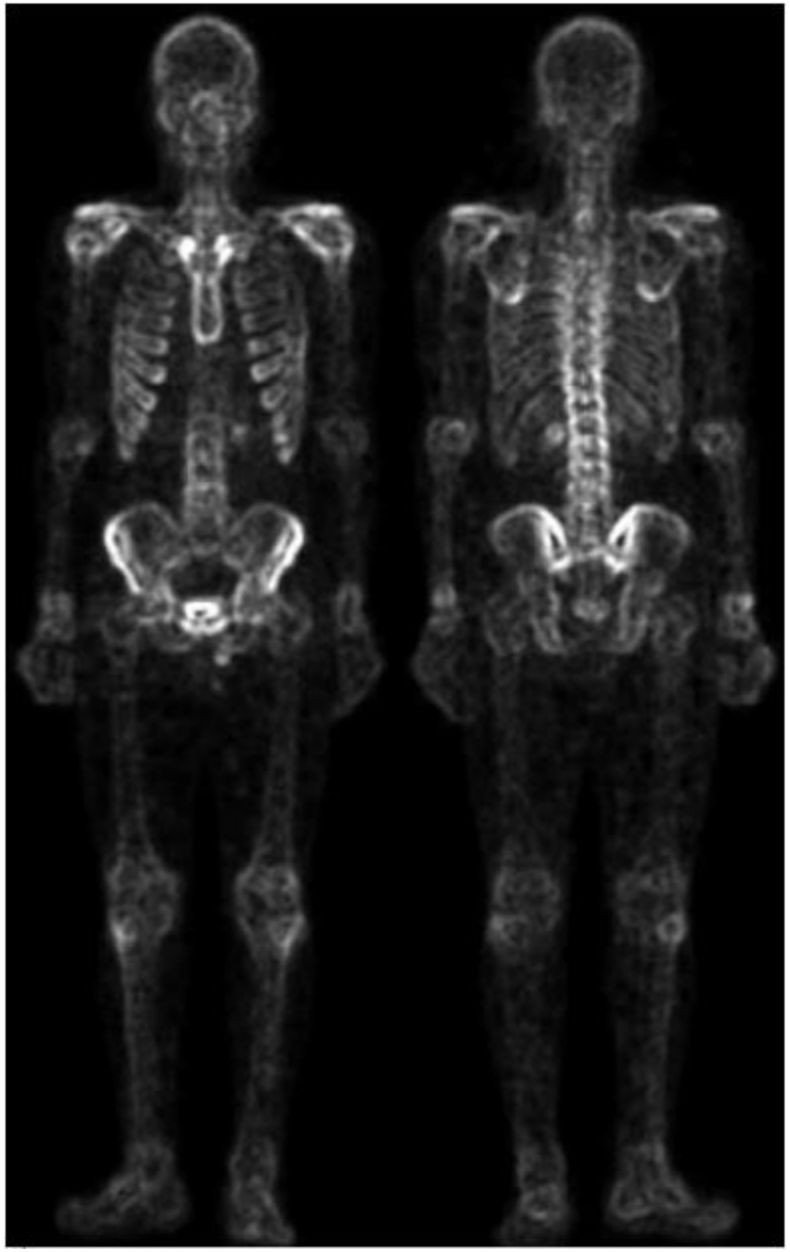
(c) Sharpened image obtained by adding (a) and (b)

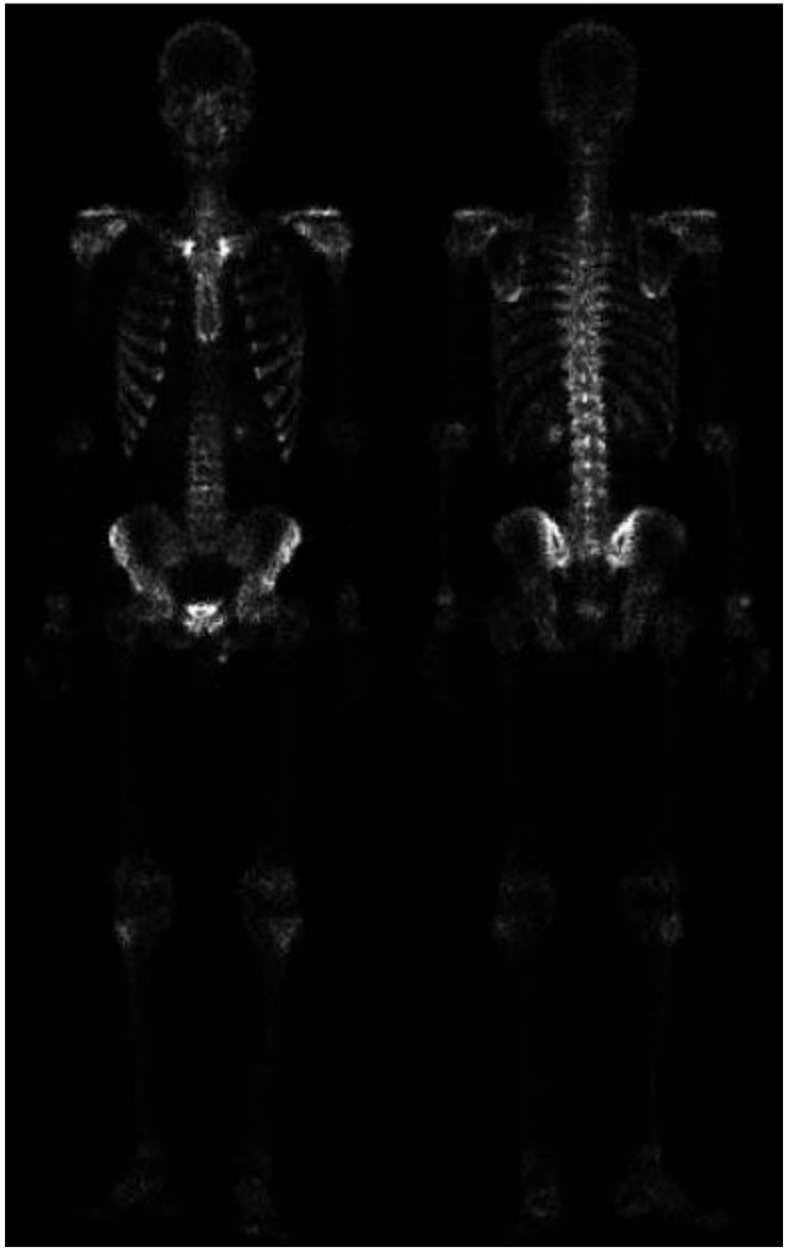
(d) Sobel Gradient of (a)

(e) Sobel image smoothed with a 5 x 5 averaging filter

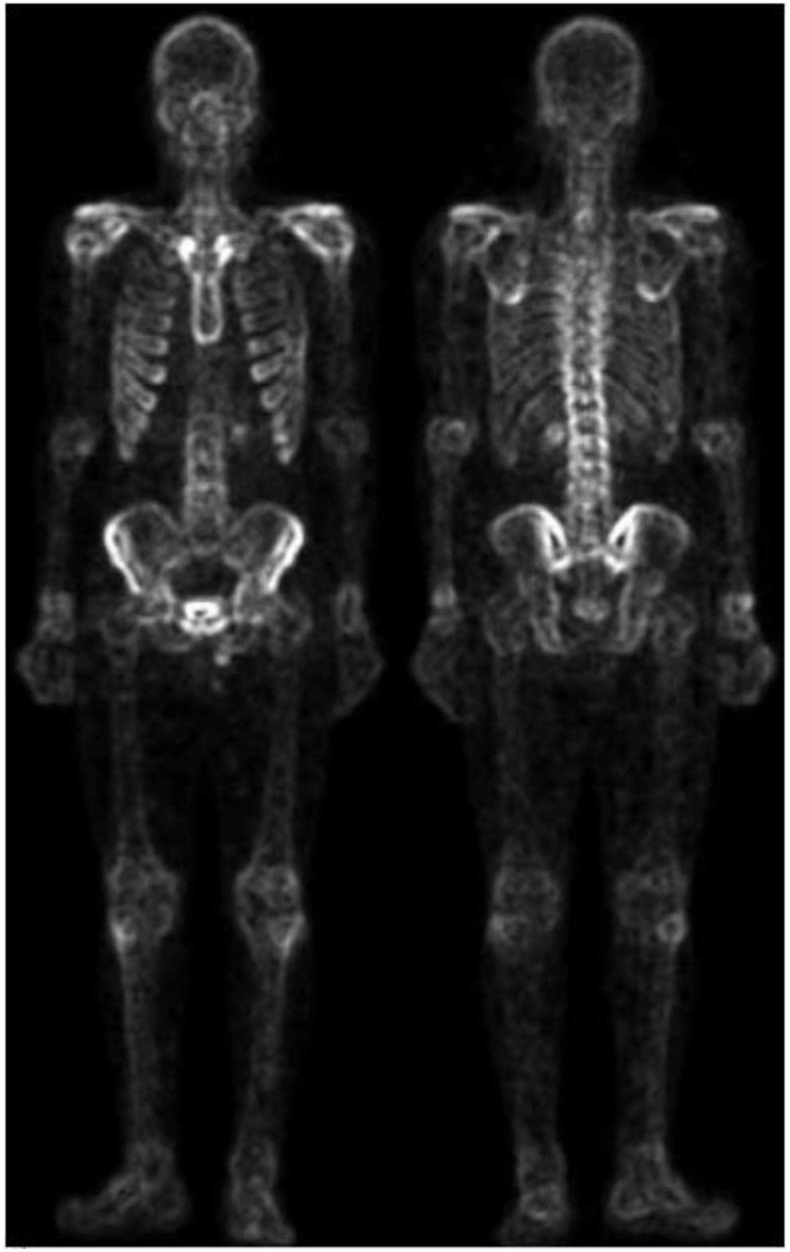


(e) Sobel image smoothed with a 5 x 5 averaging filter

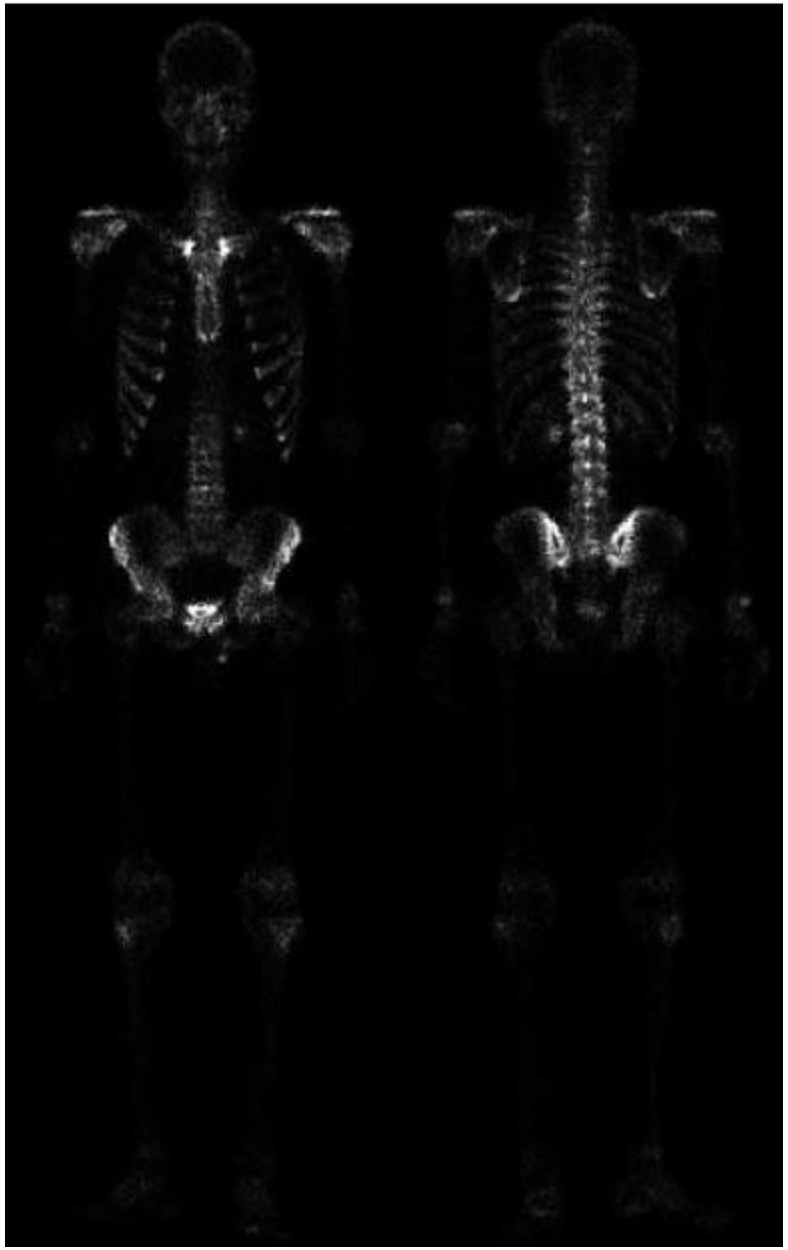




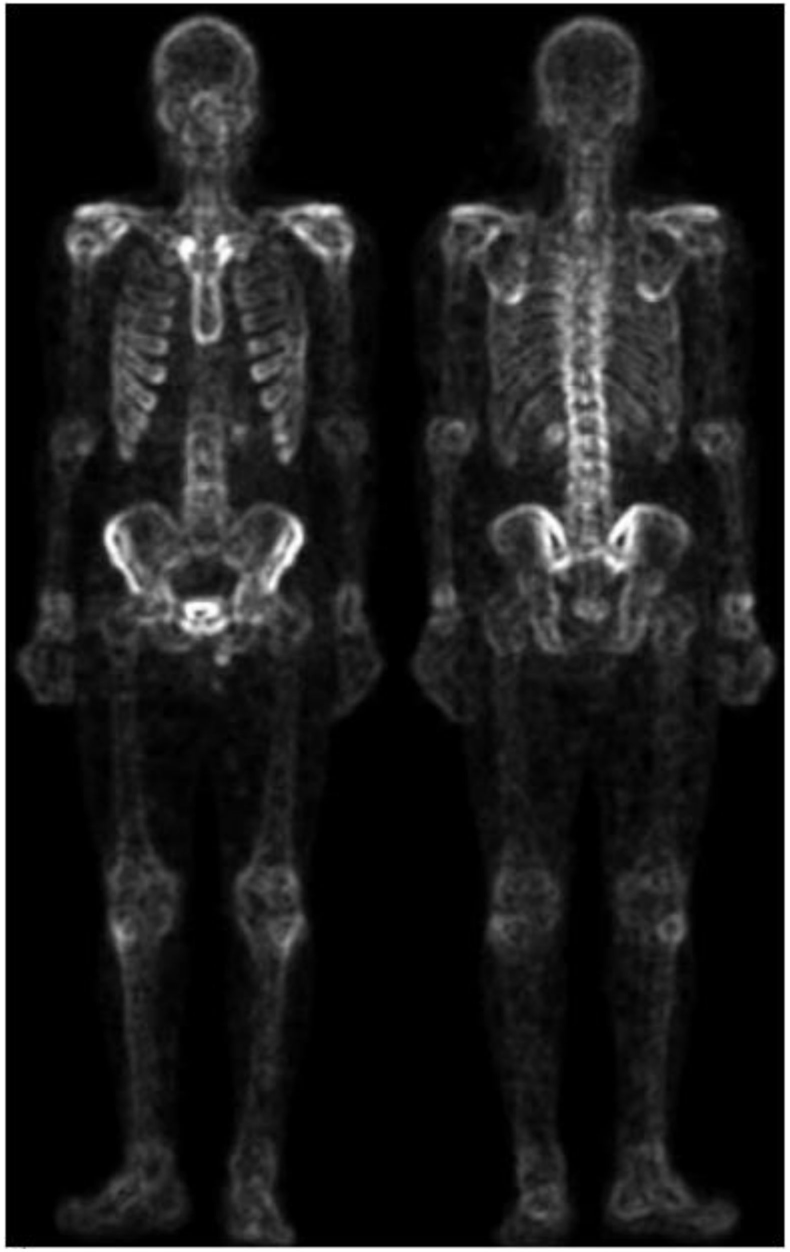
(e) Sobel image smoothed with a 5 x 5 averaging filter



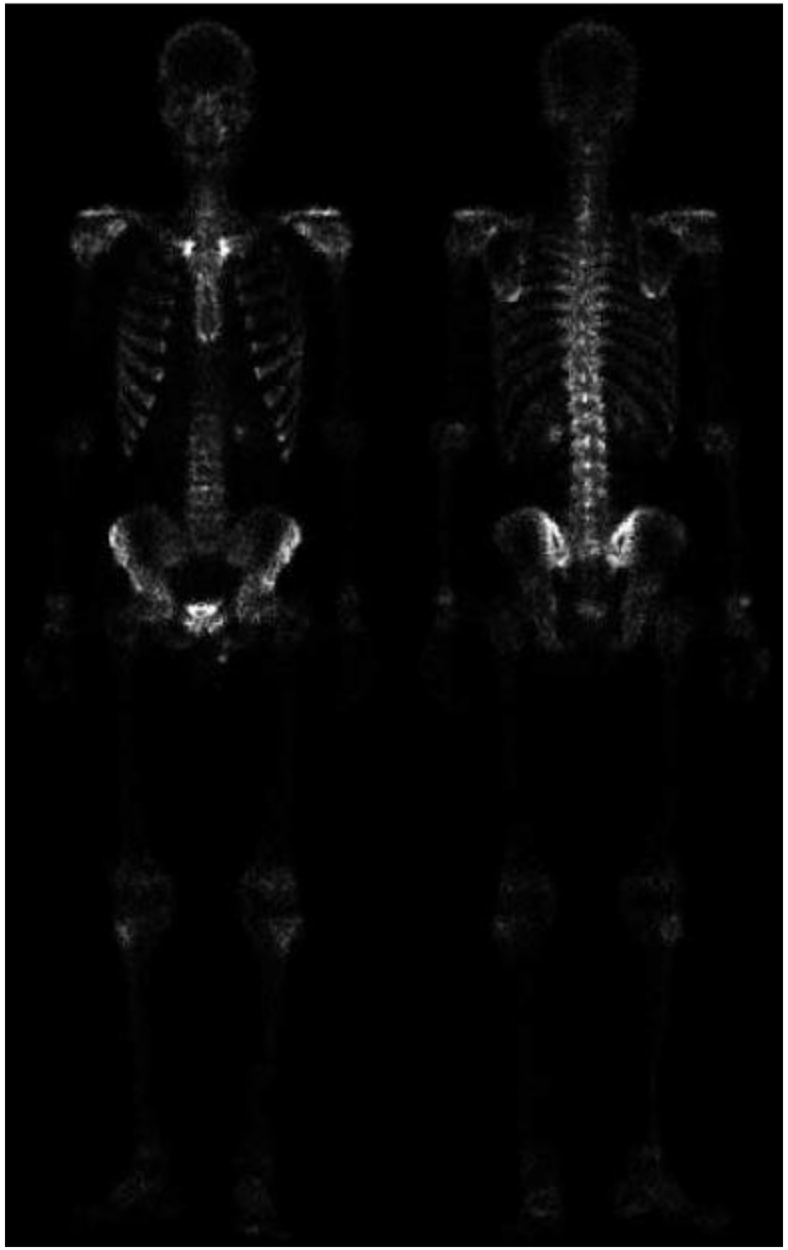
(f) Mask image formed by the product of (c) & (e)



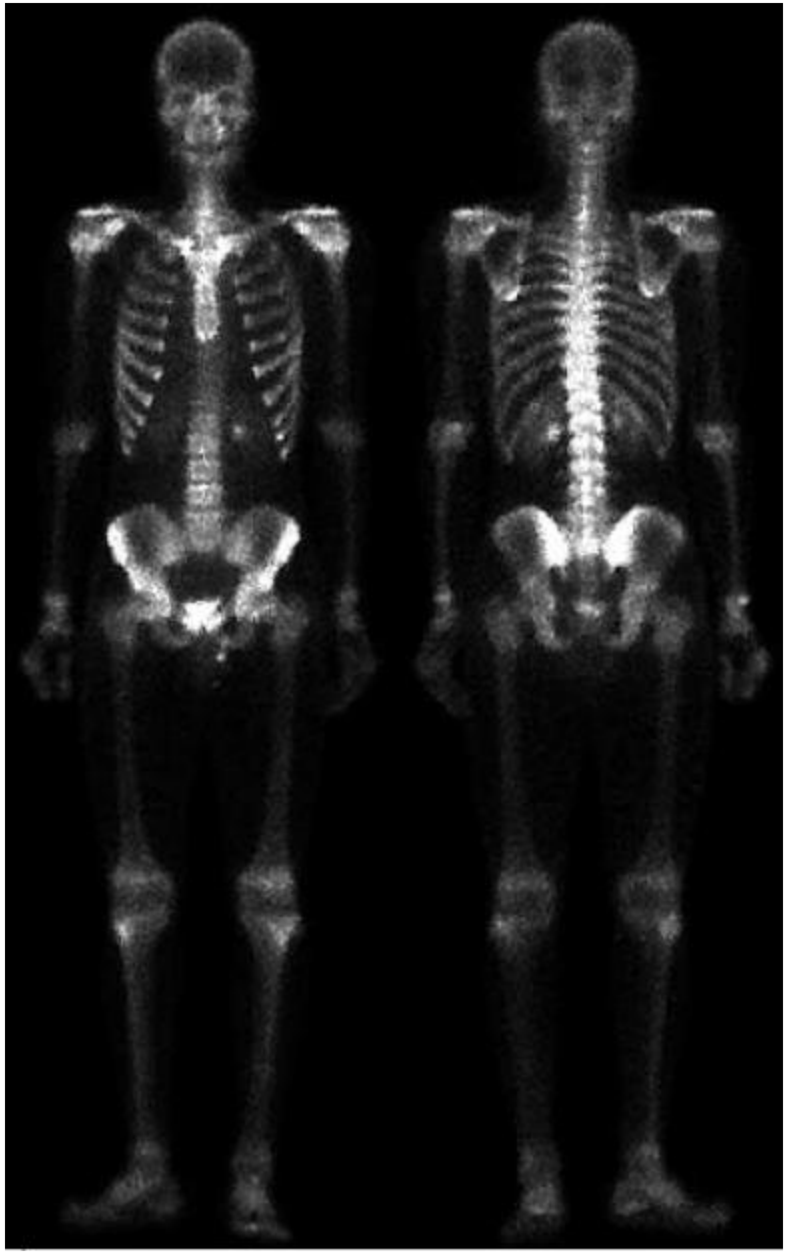
(e) Sobel image smoothed with a 5 x 5 averaging filter



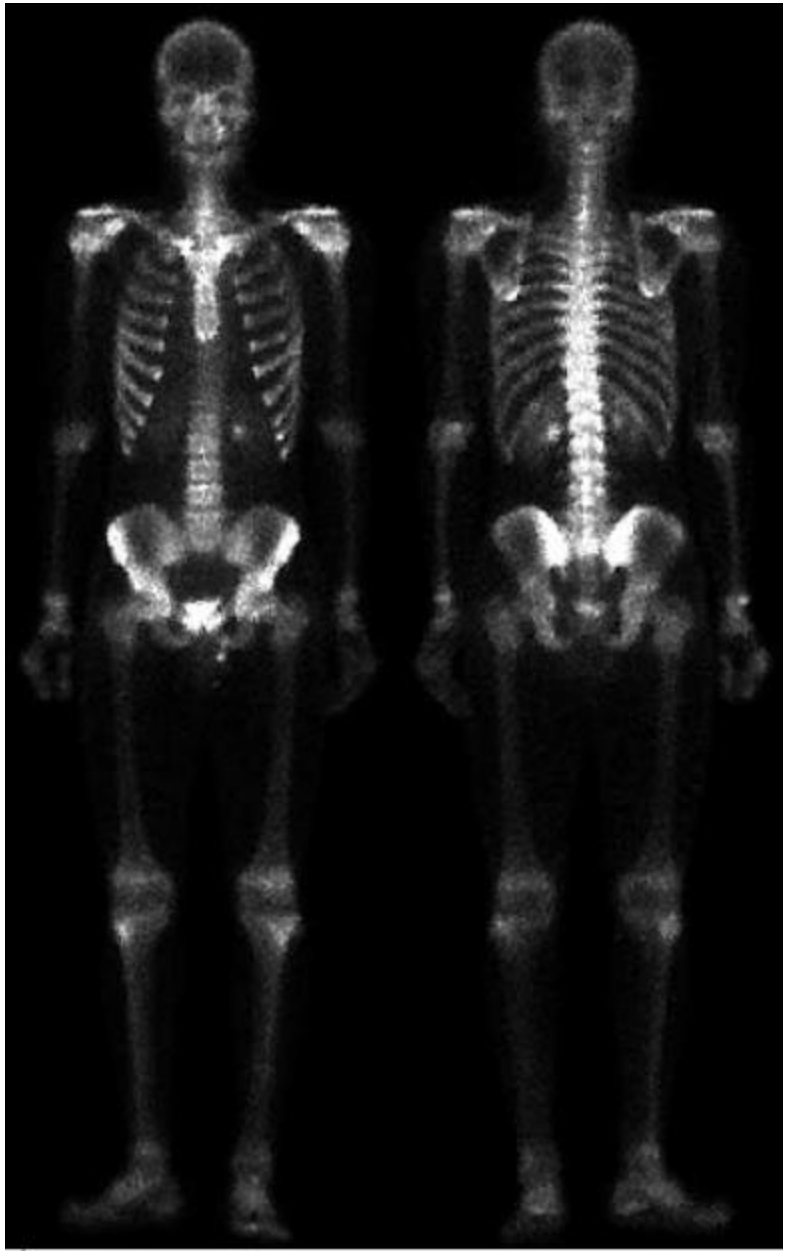
(f) Mask image formed by the product of (c) & (e)

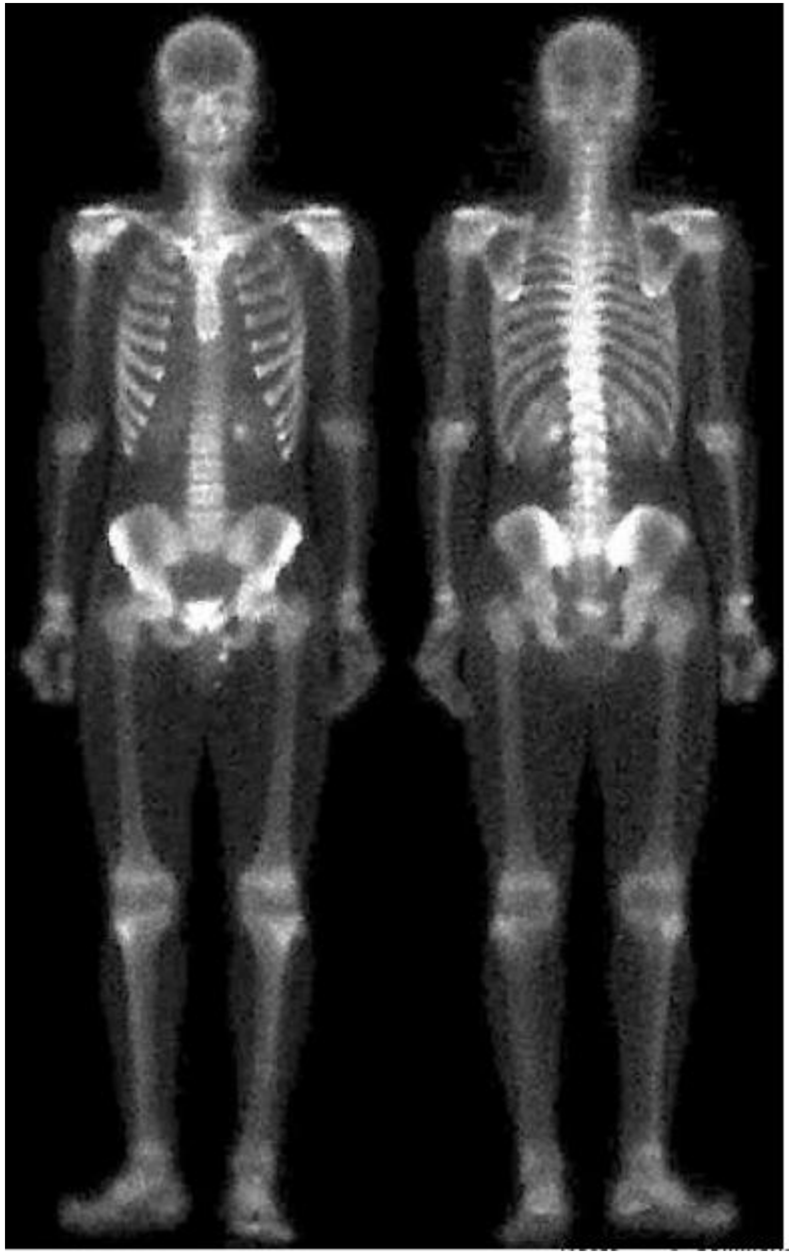


(g) Sharpened image obtained
by the sum of (a) and (f)

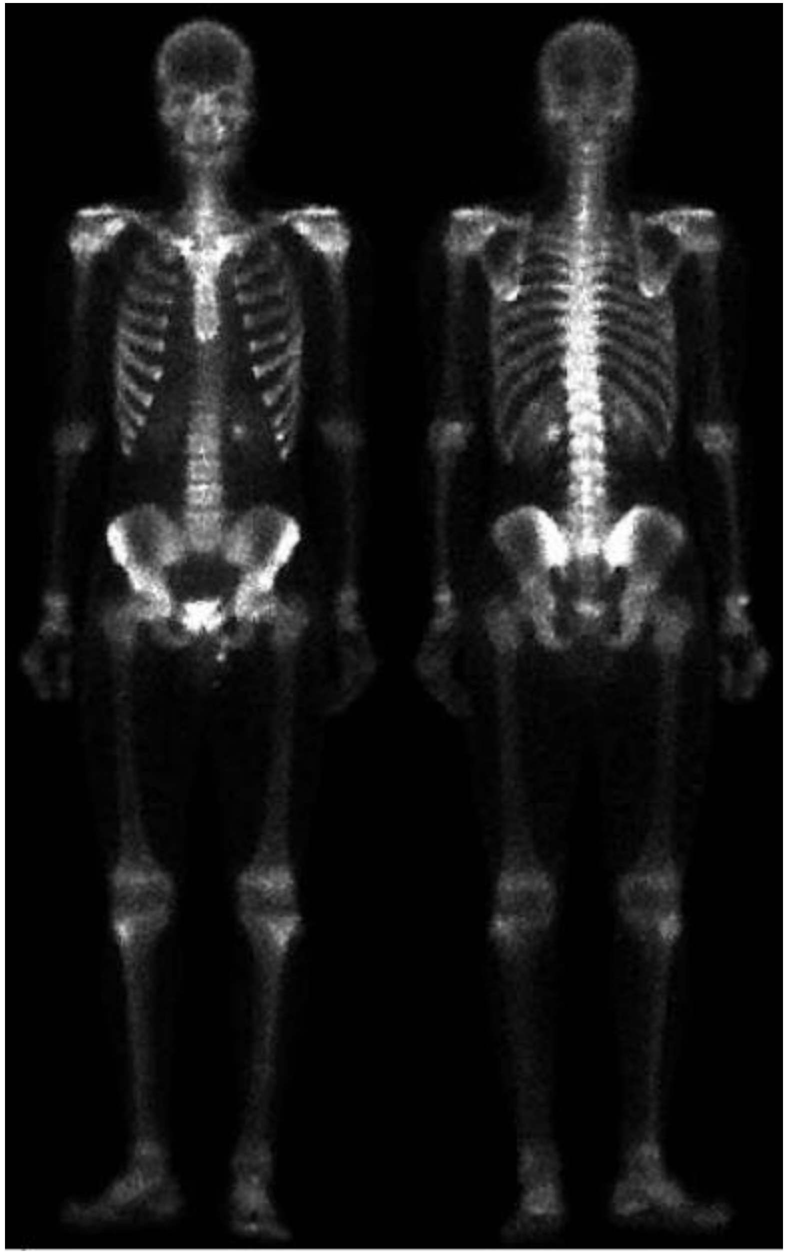


(g) Sharpened image obtained by the sum of (a) and (f)

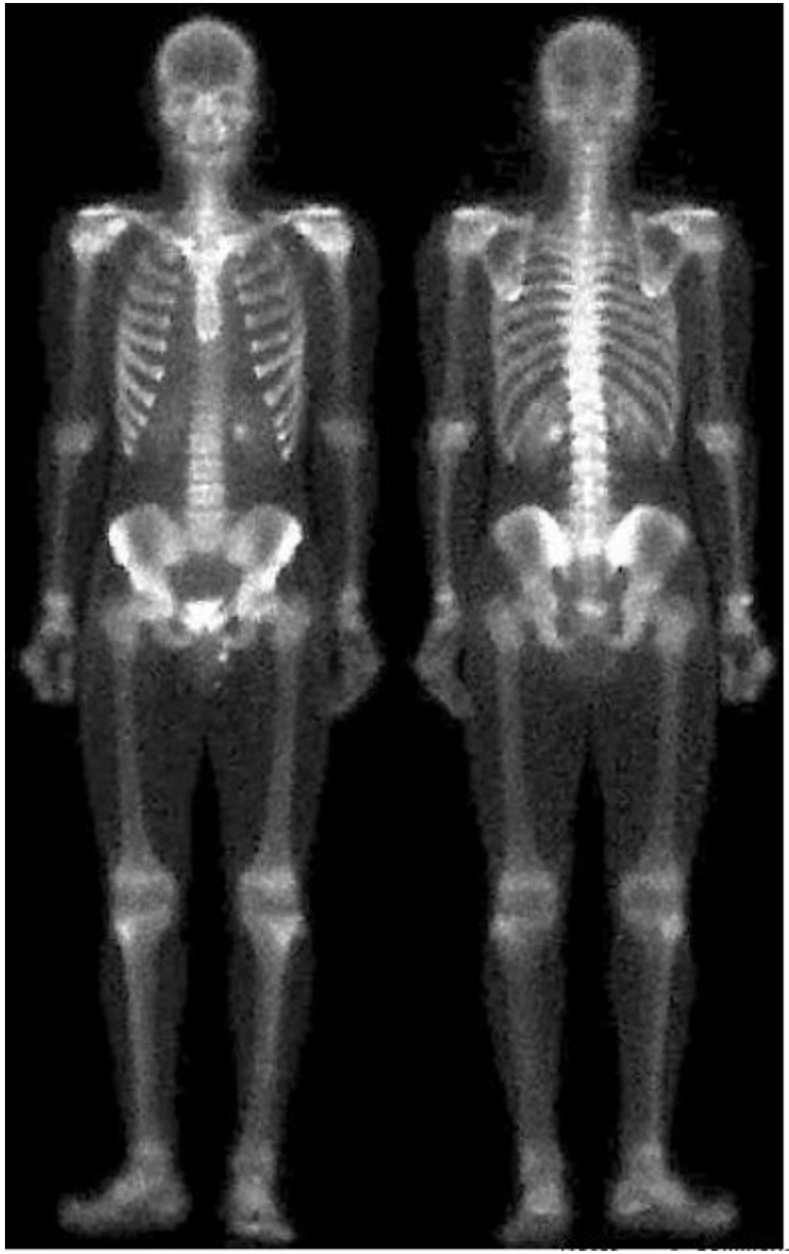




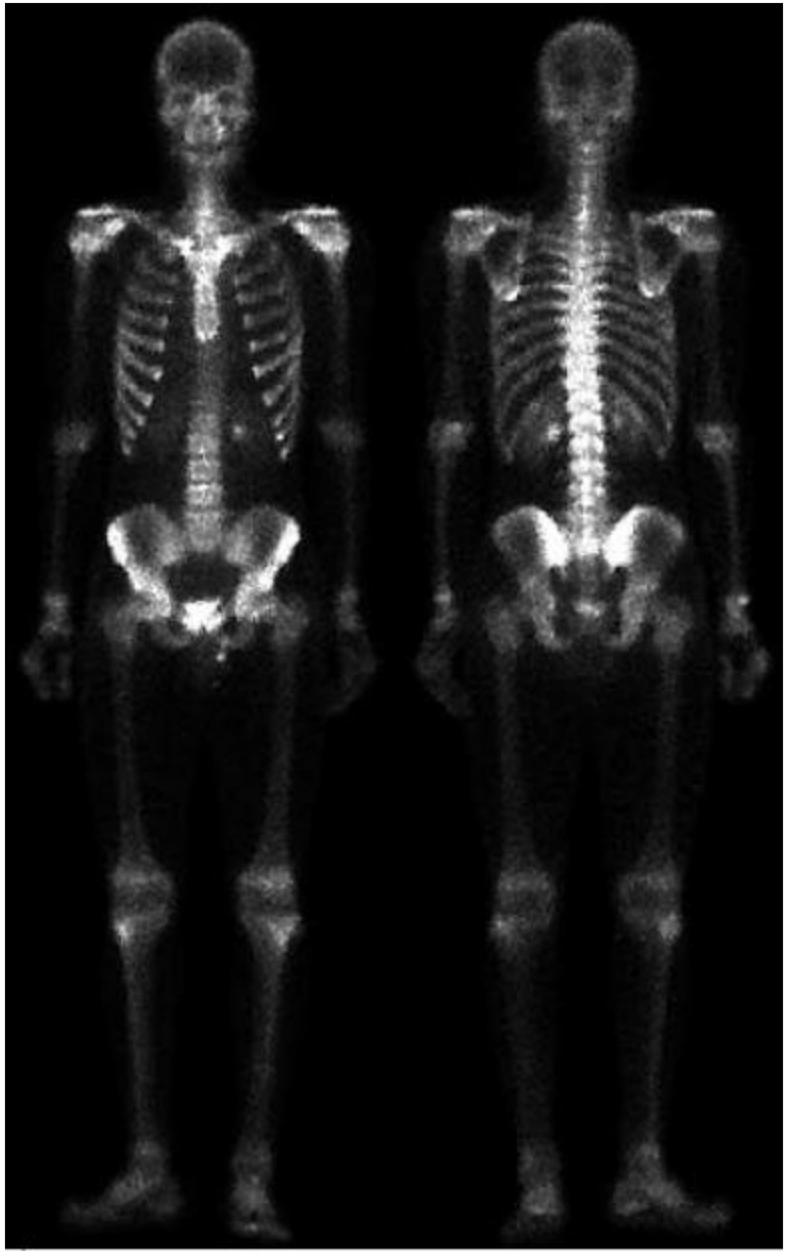
(g) Sharpened image obtained by the sum of (a) and (f)



(h) Final result obtained by applying power law transformation to (g)



(g) Sharpened image obtained by the sum of (a) and (f)



(h) Final result obtained by applying power law transformation to (g)