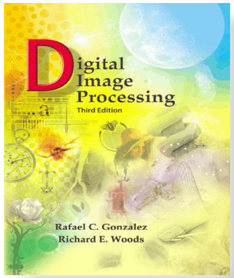


Intensity Transformations & Spatial Filtering

- Spatial Domain
  - Apply spatial filters directly

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

- Frequency / Transform Domain
  - Take the image to transform domain
  - Apply filter
  - Bring back image to spatial domain

- Spatial Processing (Computationally efficient & less processing resources)
  - Intensity Transformation (point processing)
    - Contrast manipulation & Image thresholding
  - Spatial Filtering (neighbourhood processing)
    - Image sharpening

Spatial Filtering

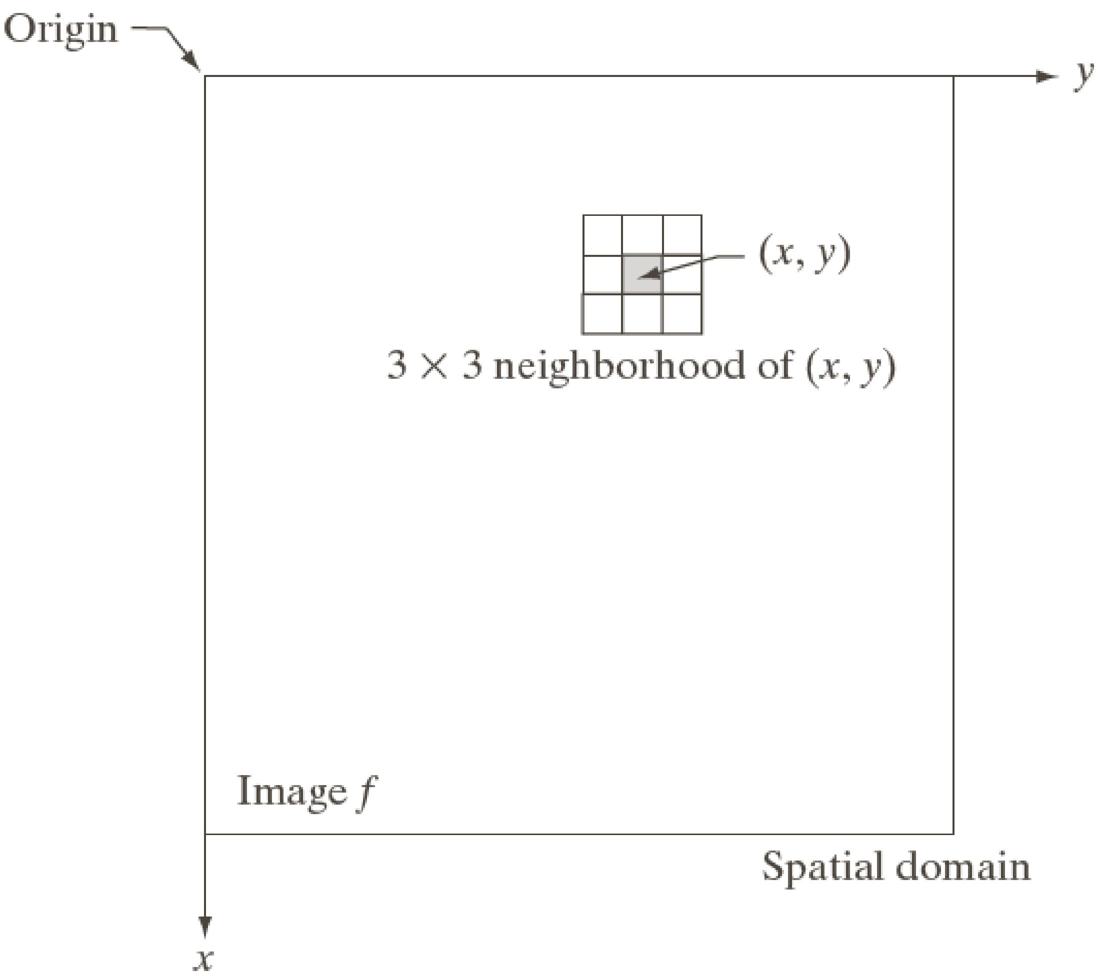
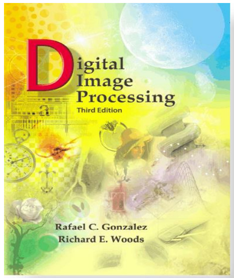


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

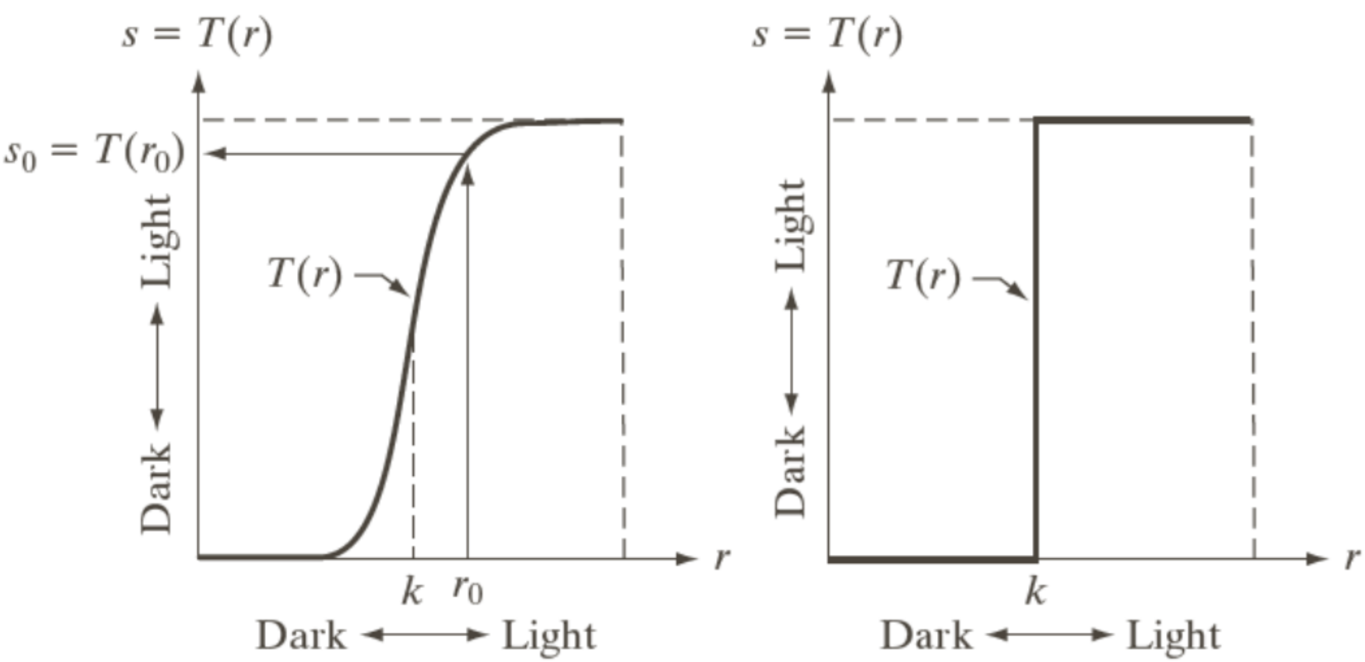


- Spatial Filtering: 3x3
  - Spatial Mask
  - Kernel
  - Template
  - Window
- Intensity Transformation: 1x1

$$s = T(r)$$

INTENSITY TRANSFORMATION

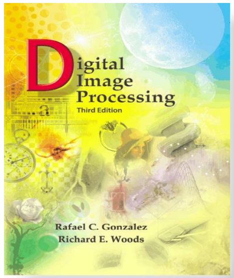
- ✓ Contrast stretching function
- ✓ Thresholding function



a b

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

- Image Enhancement
  - Process of manipulating an image so that the result is more suitable than the original for a specific application
  - Enhancement techniques are problem specific (X-ray images vs Infrared images)
  - Visual interpretation
  - Machine perception (Character recognition)



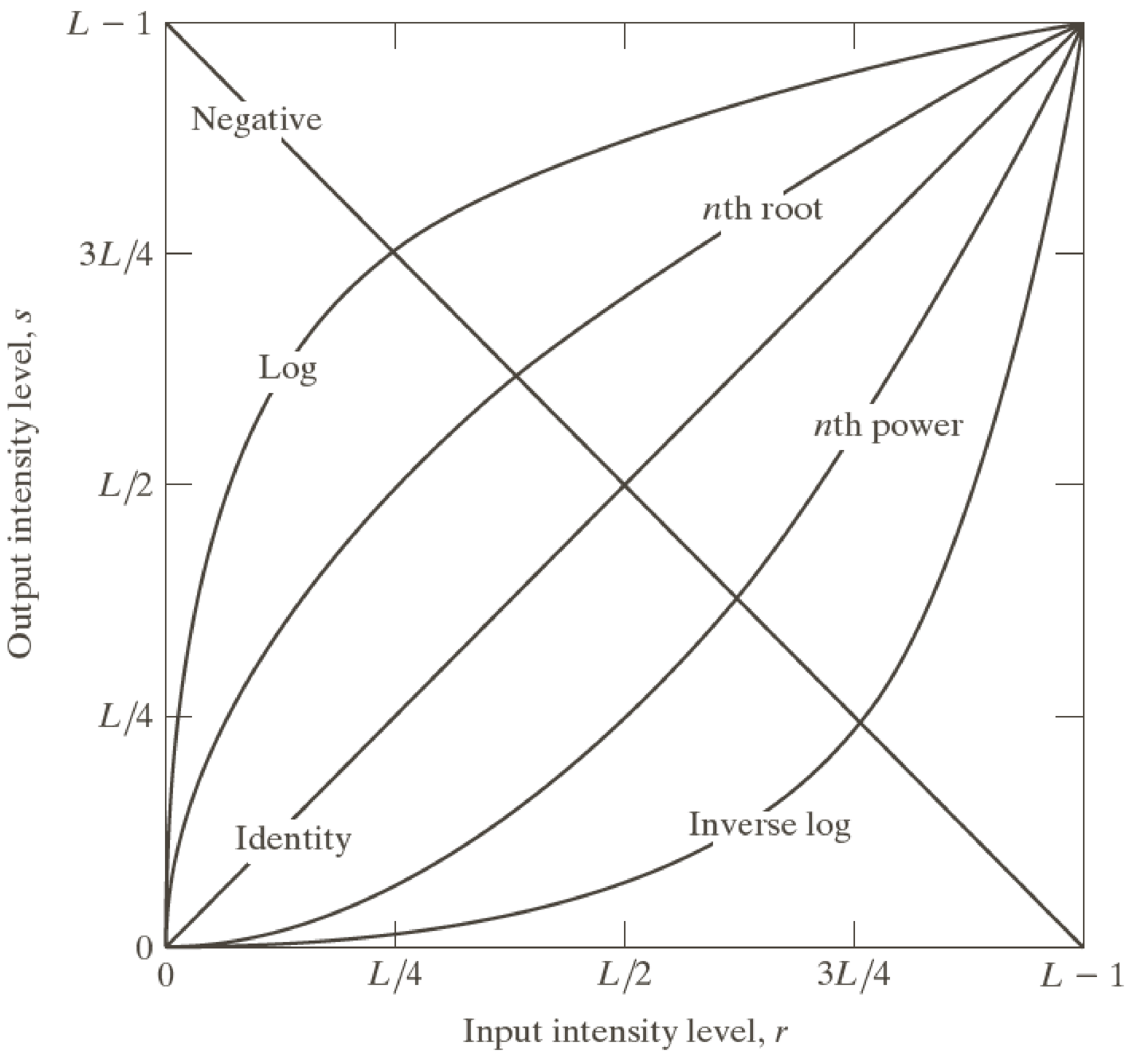
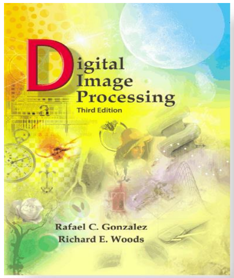


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Basic Intensity Transformation Functions



- Image Negatives

$$s = L - 1 - r$$

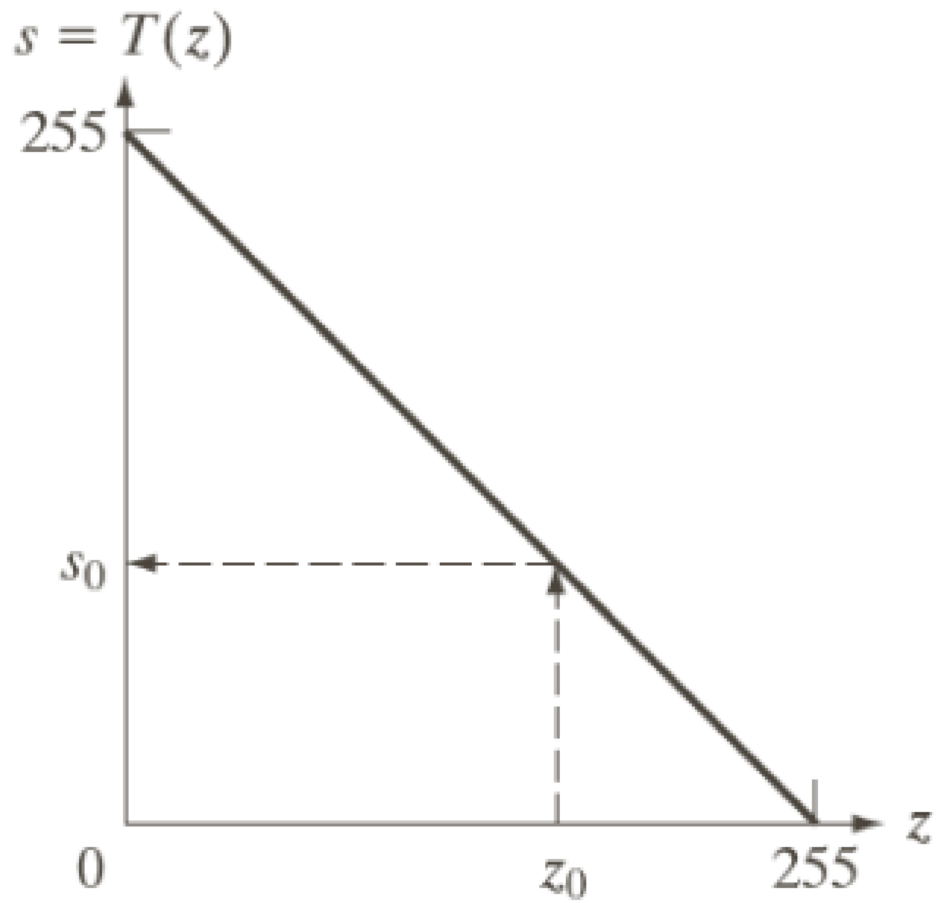
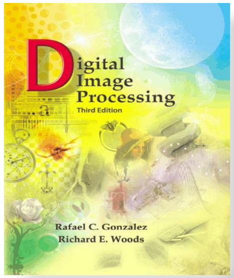
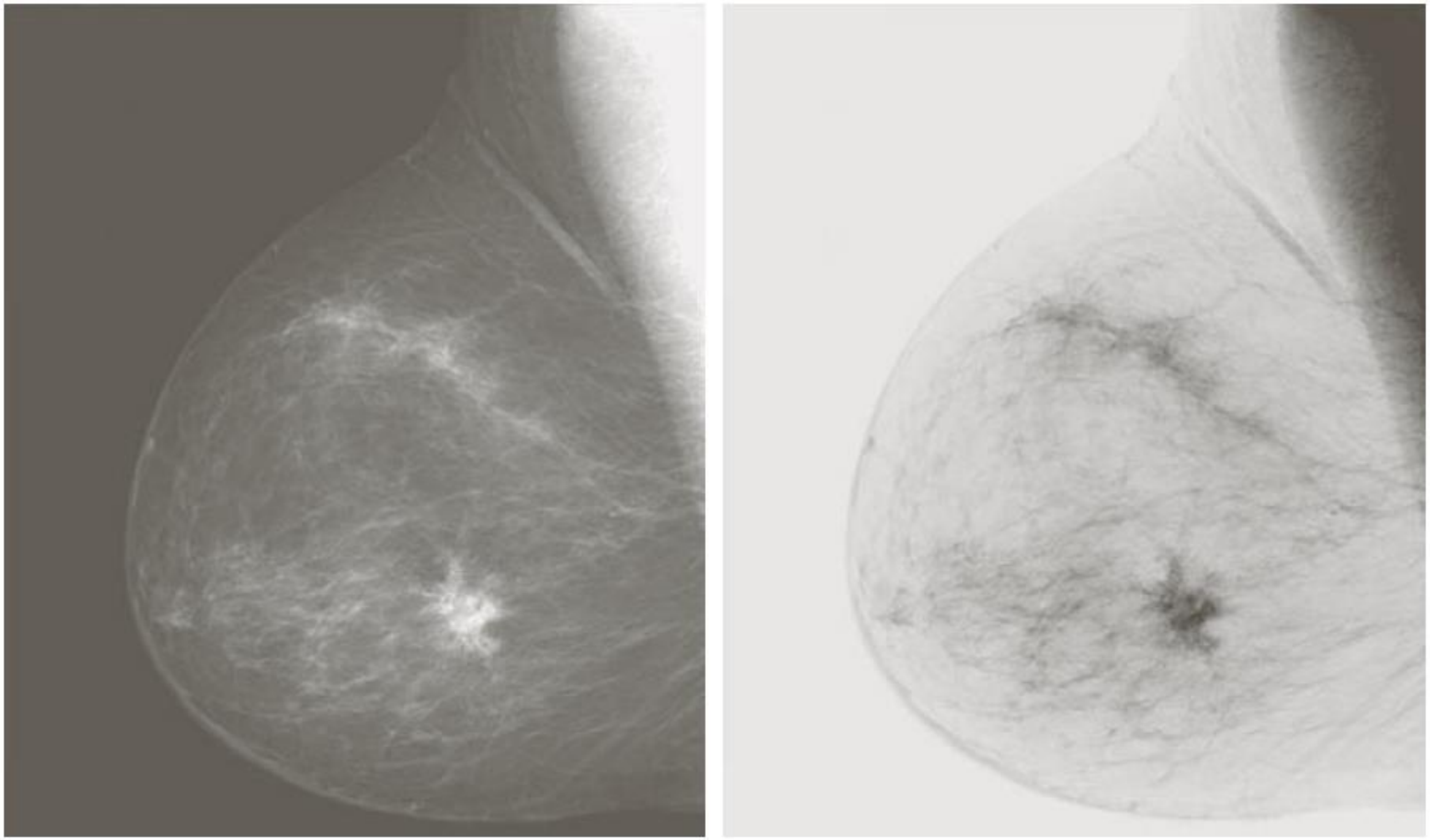


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.



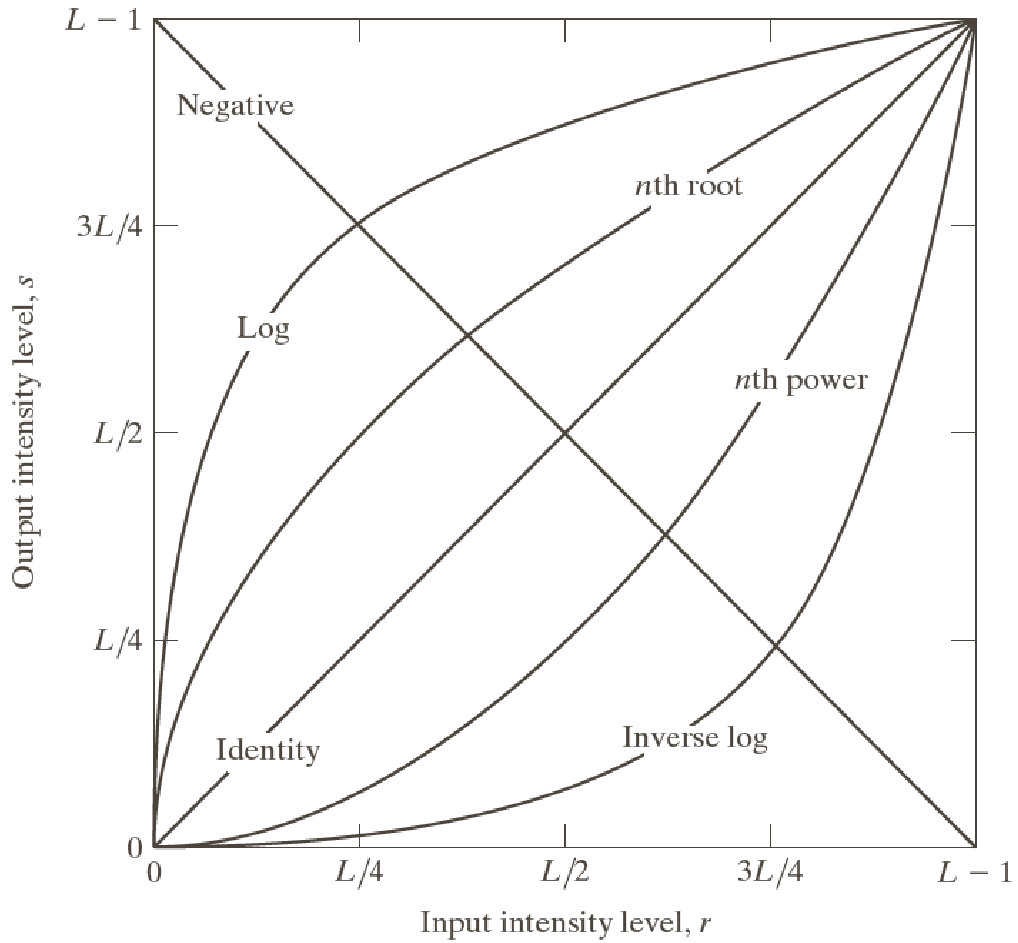


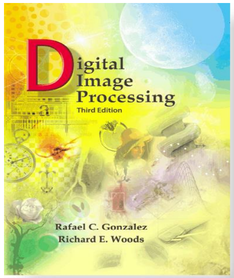
a b

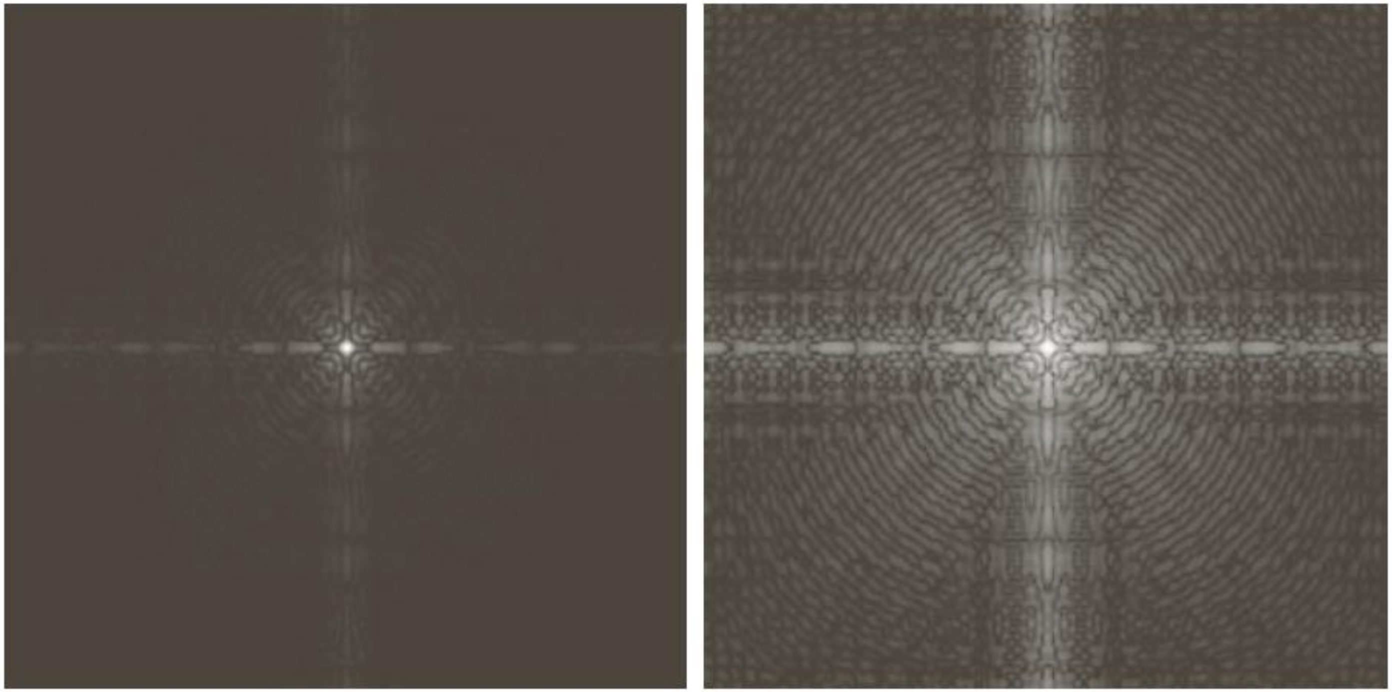
FIGURE 3.4
(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).
(Courtesy of G.E. Medical Systems.)

- Log Transformations

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





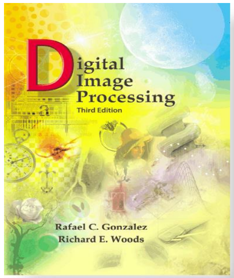


a b

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

$0 \ to \ 1.5 \times 10^6$

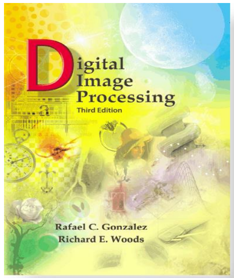
$0 \ to \ 6.2$



- Power-Law (Gamma) Transformations

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

- Gamma Correction
  - The devices used to capture, print and display obey a power-law
  - The process to correct the power-law response phenomena



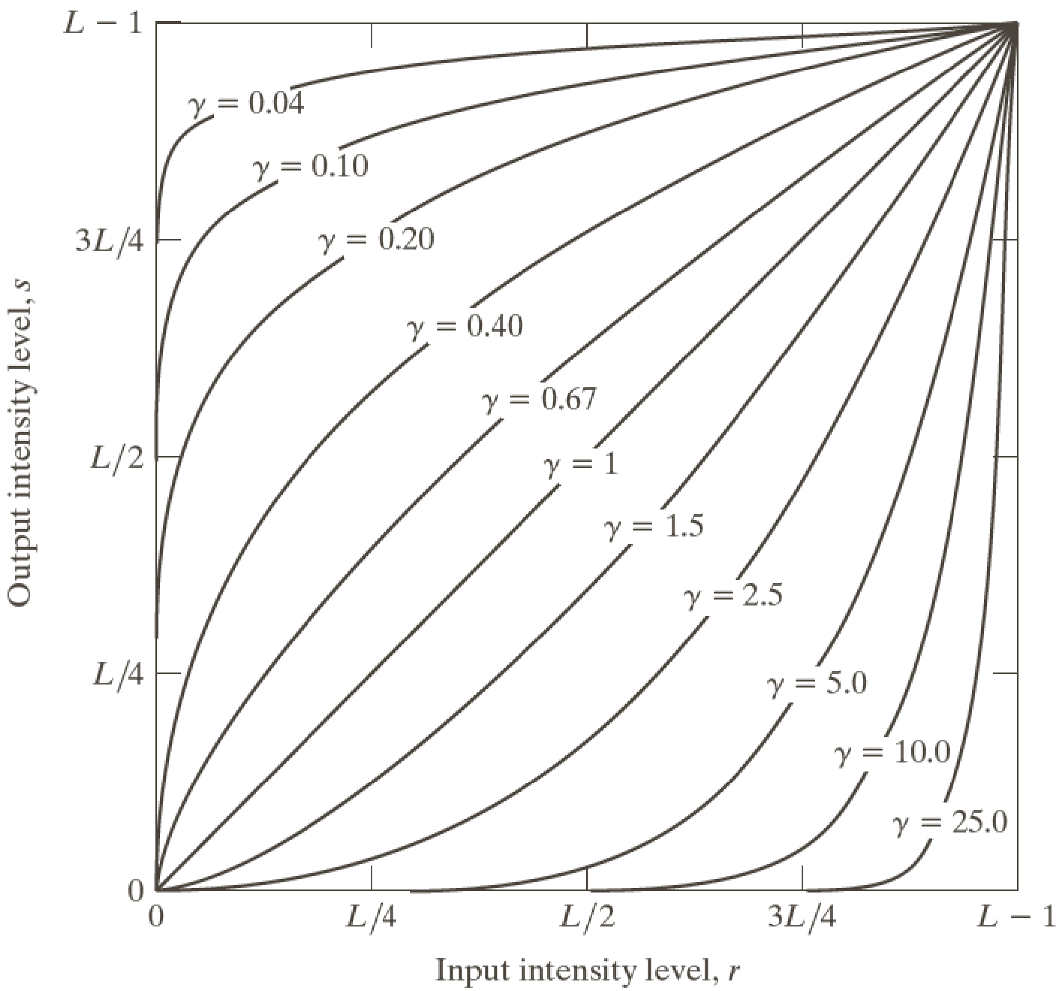
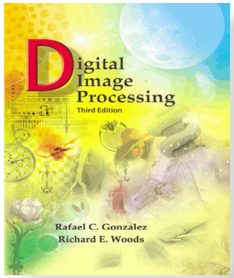
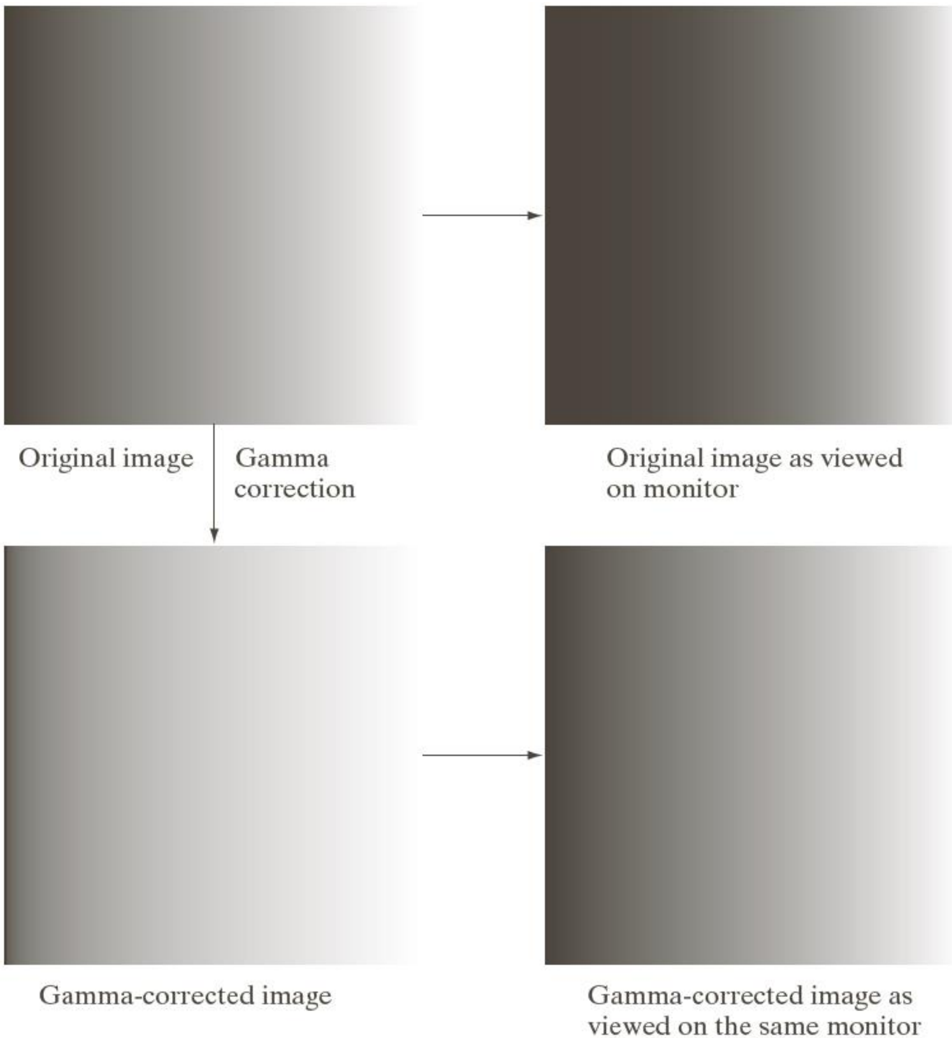


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.



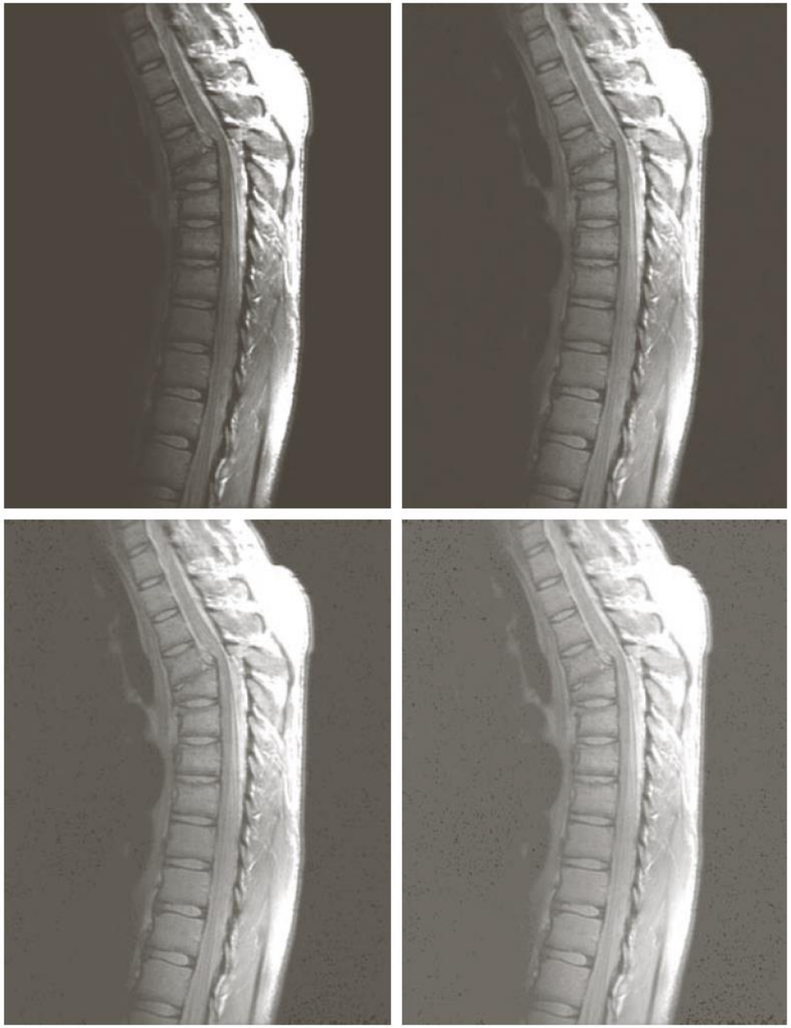


a b
c d

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

Chapter 3

Intensity Transformations & Spatial Filtering



a b
c d

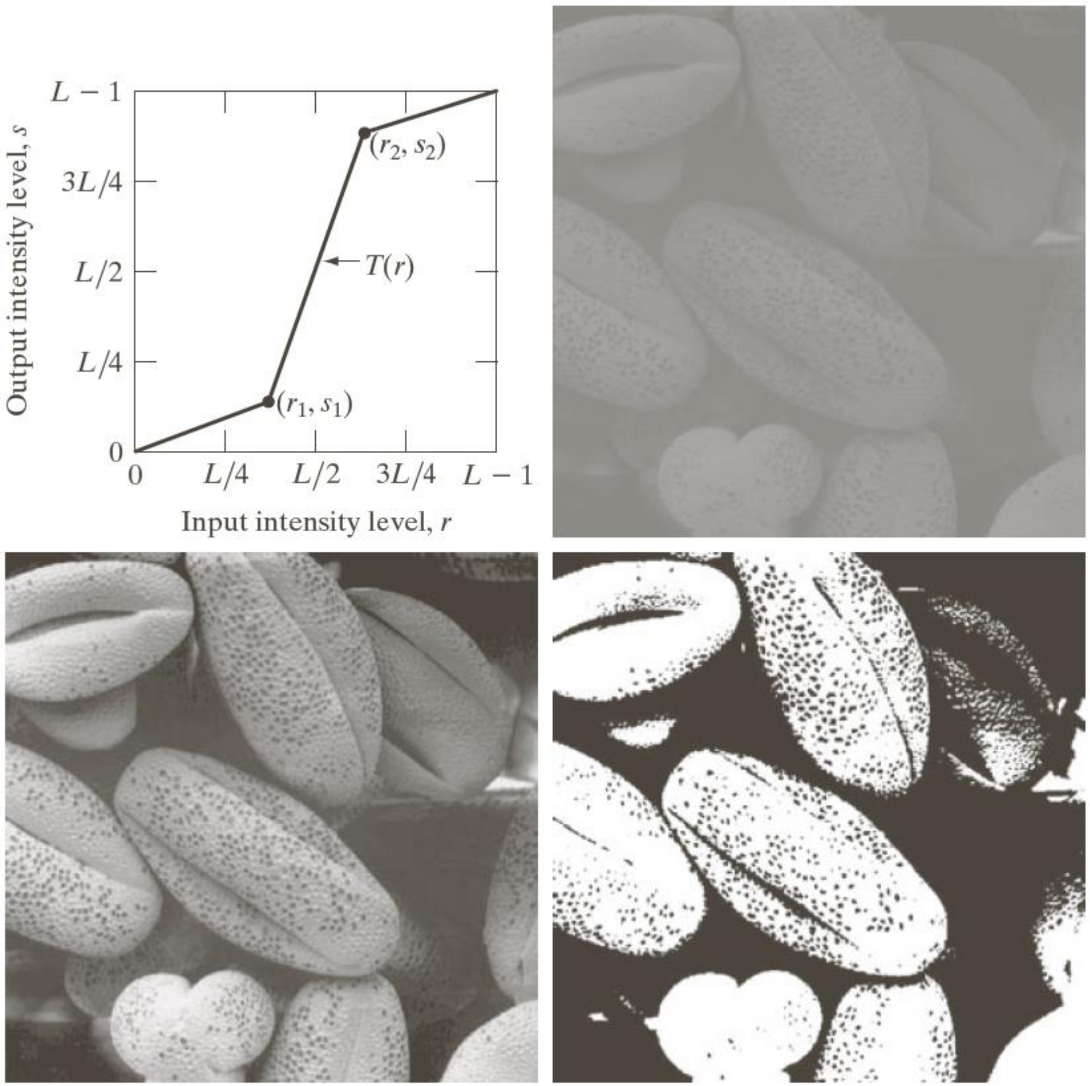
FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt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

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.)

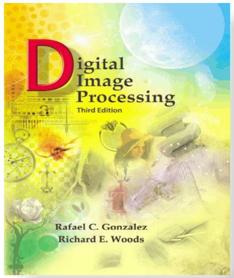
- Piecewise Linear Transformation
  - Contrast Stretching
    - Poor illumination
    - Lack of dynamic range of image sensor
    - Wrong setting of lens aperture
  - Intensity Level Slicing
  - Bit-Plane Slicing

Chapter 3
Intensity Transformations & Spatial Filtering



a	b
c	d

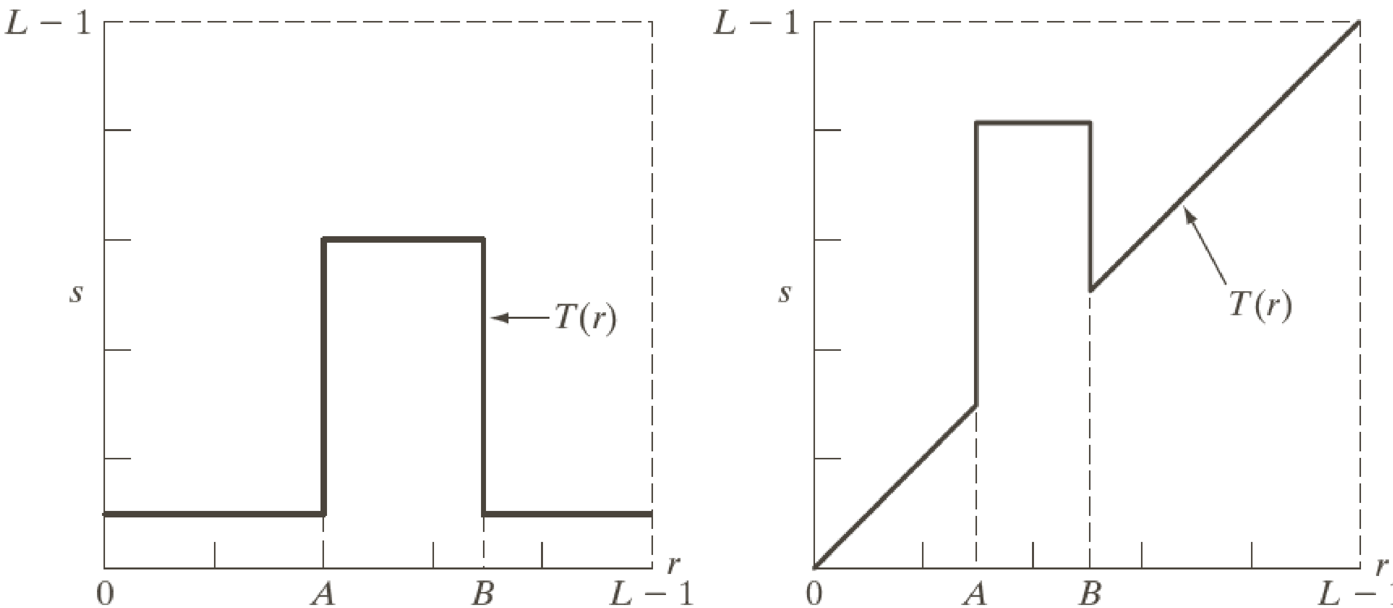
FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

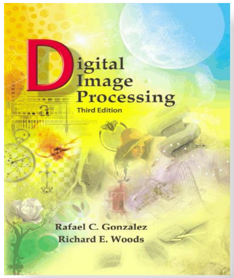


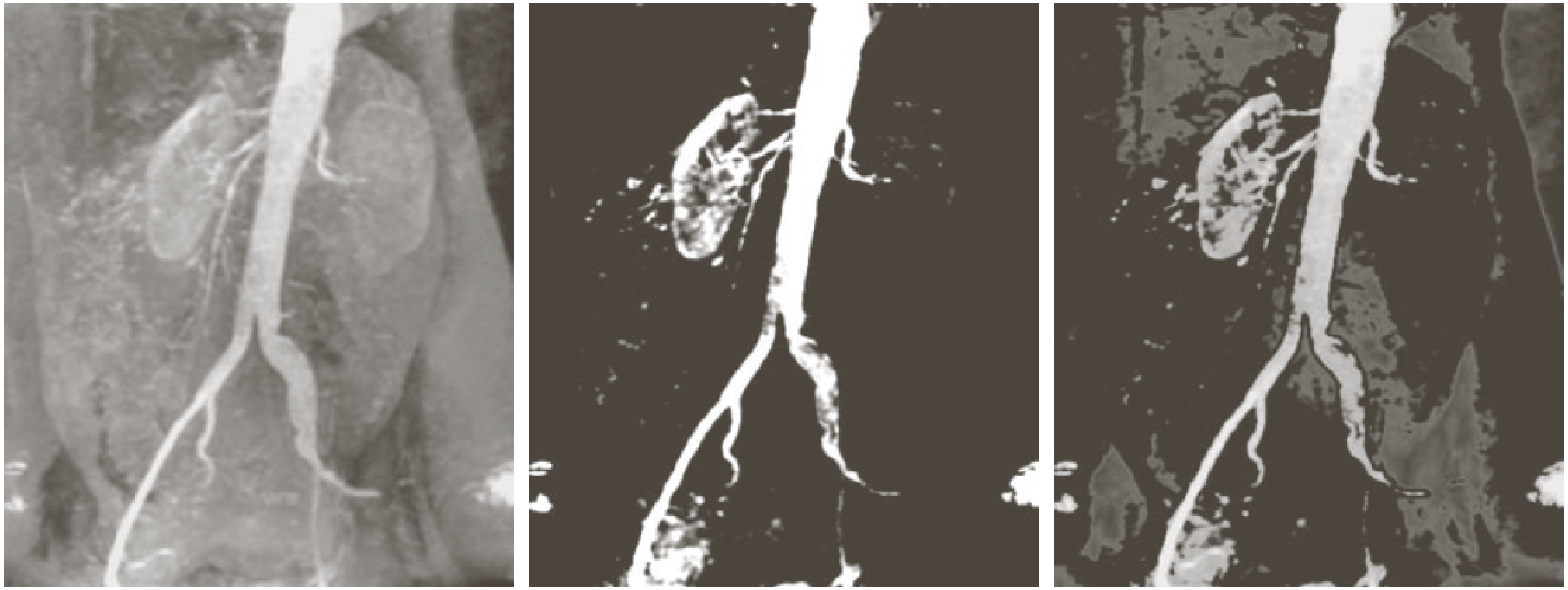
Chapter 3
Intensity Transformations & Spatial Filtering

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.)

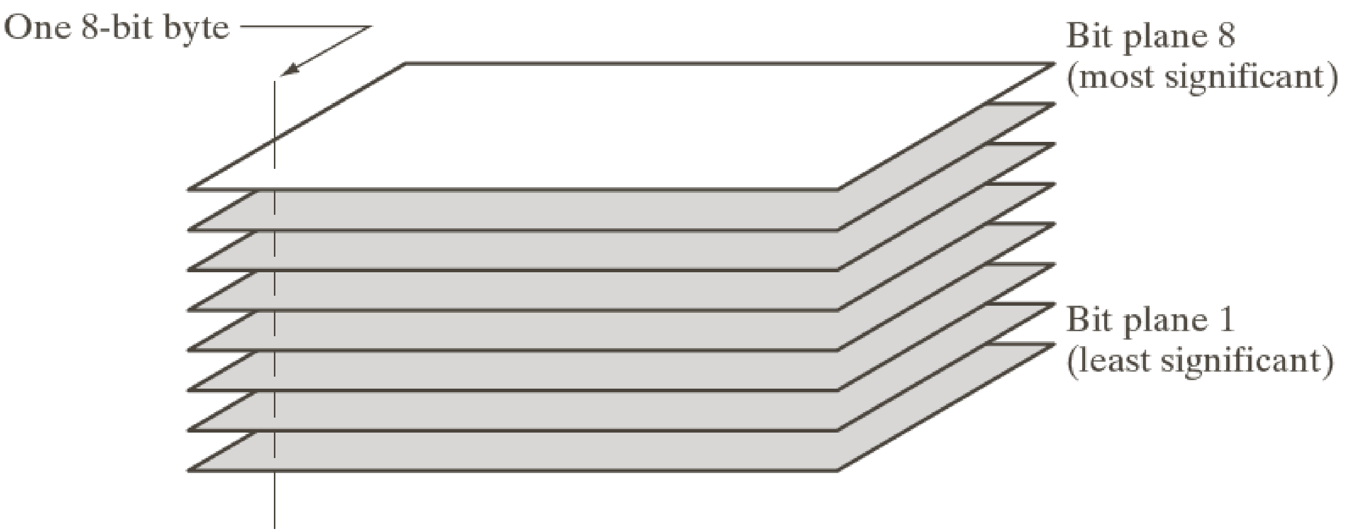
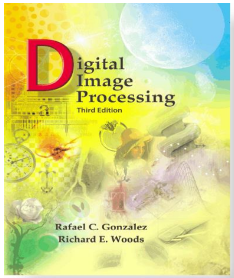


FIGURE 3.13 Bit-plane representation of an 8-bit image.



Chapter 3
Intensity Transformations & Spatial Filtering

a	b	c
d	e	f
g	h	i

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

a b c

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

HISTOGRAM PROCESSING: EQUALIZATION

- Histogram Processing

$$h(r_k) = n_k$$

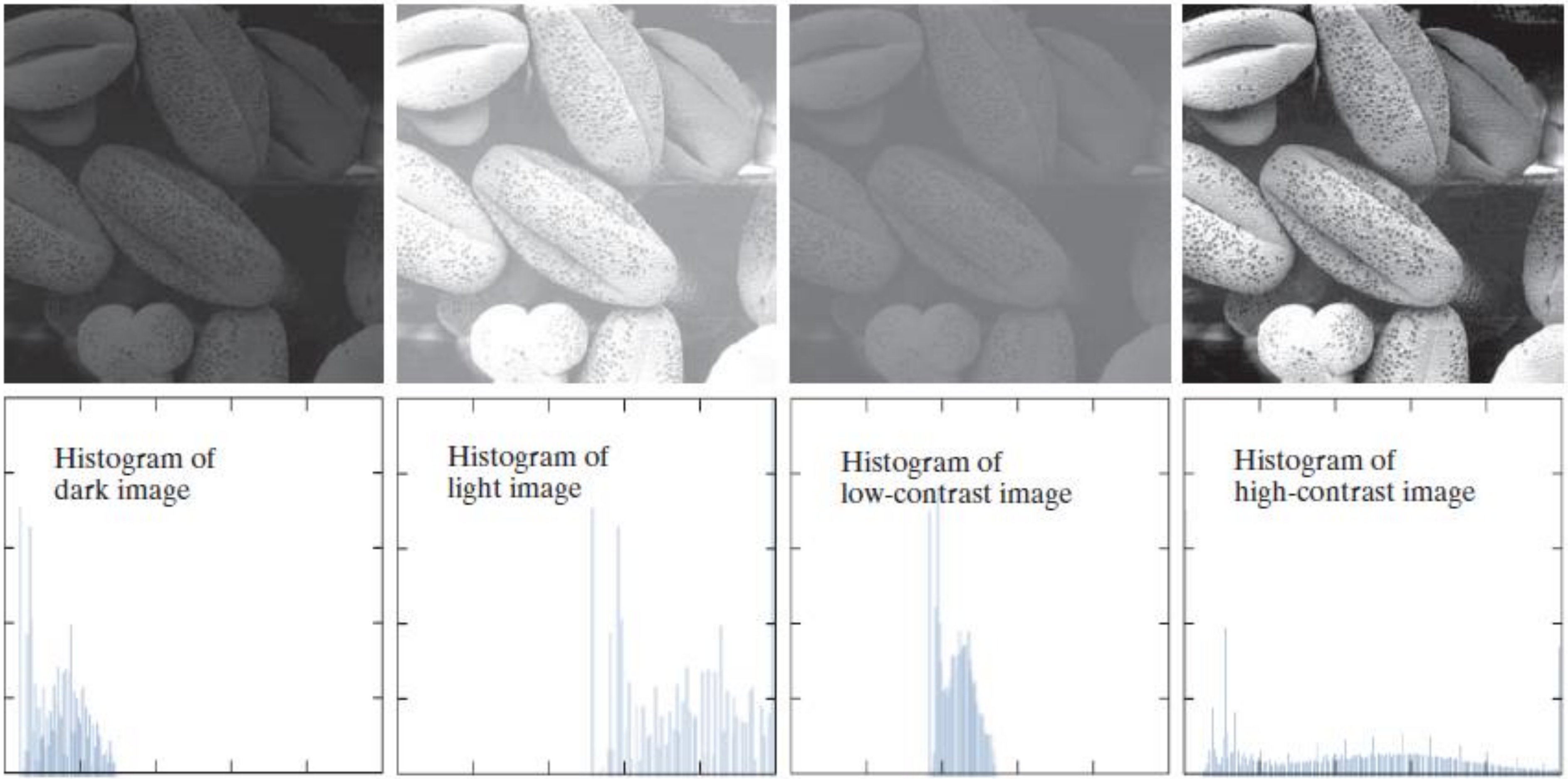
$$r_k : k\text{th intensity value}, \; k \in [0, L-1]$$

$$n_k : \#\text{ of pixels with value } r_k$$

$$p(r_k) = n_k / MN$$

$$M \times N : \text{image dimension}$$

Chapter 3



a b c d

FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

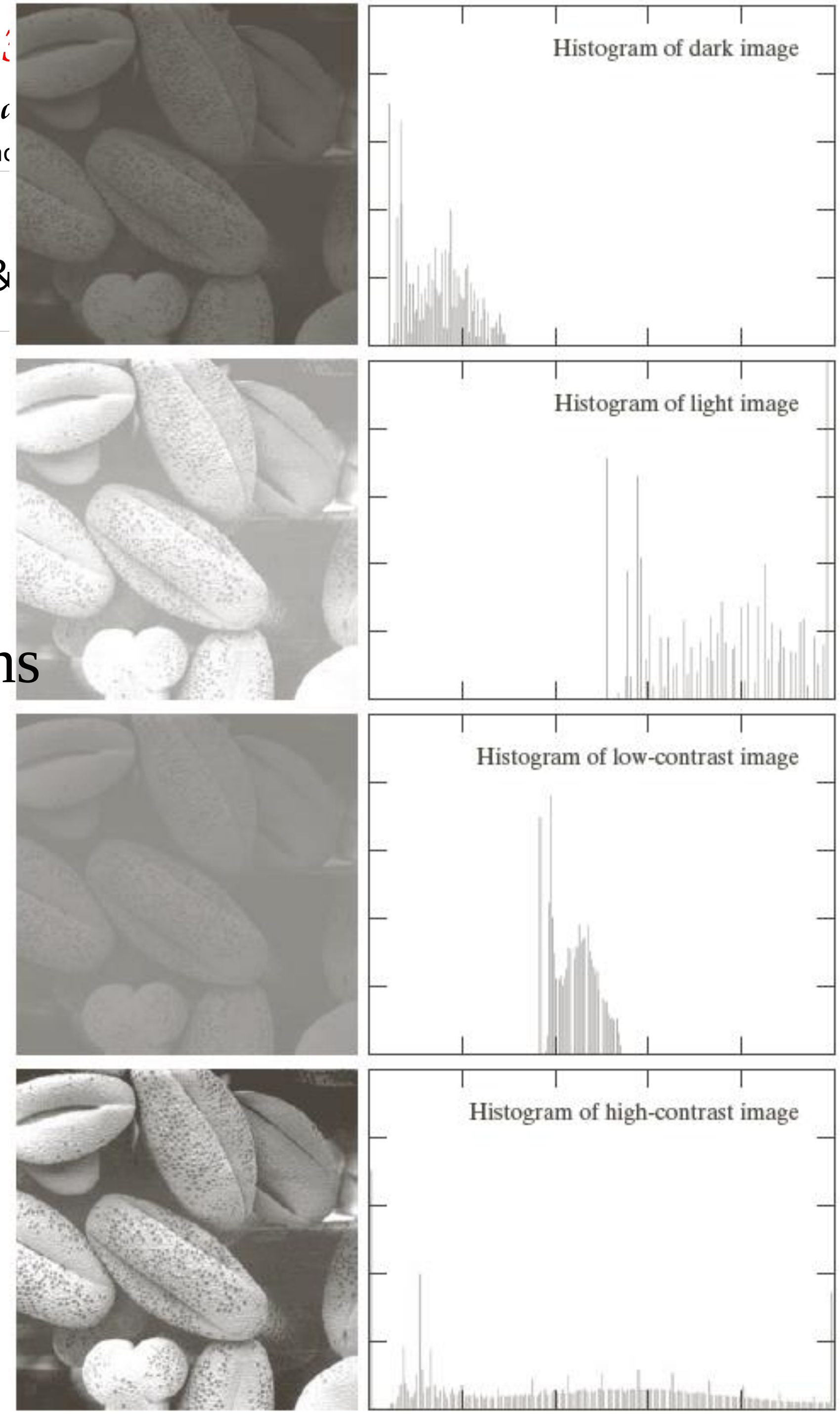
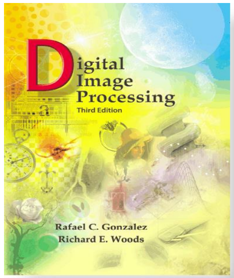


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

Advantages of Histogram Processing

- Simple to compute
- Suitable for fast H/W implementations
- Fully automatic
- No user inputs are required
- No parameter specifications required



Histogram Equalization

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

$T(r)$ should have the following properties:

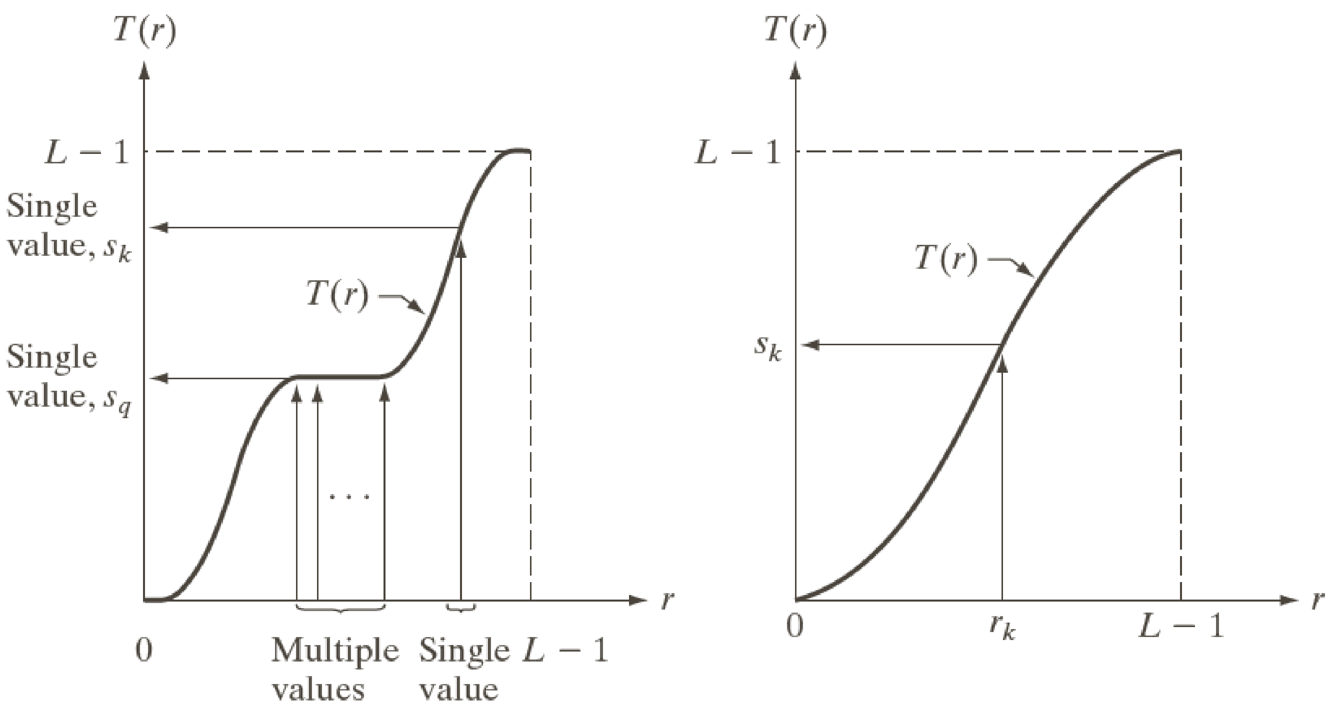
a) Monotonically increasing in $0 \le r \le L-1$

- Strictly monotonically increasing if

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

is to be used

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



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.

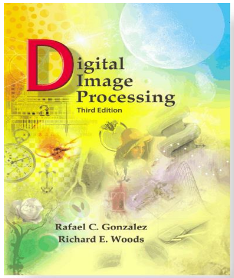
- Histogram Equalization

$$s = T(r)$$

$T(r)$ is continuous & differentiable

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$p_r(r)$ and $p_s(s)$ are PDF of i/p & o/p values



- Histogram Equalization

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

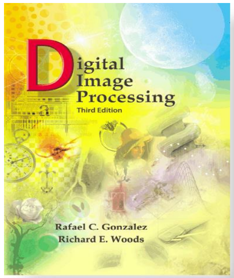
RHS is the CDF of r

$T(r)$ satisfies (a) & (b)

$$\frac{d(s)}{dr} = \frac{d(T(r))}{dr} = \frac{d}{dr}\left[(L-1)\int_0^r p_r(w)dw\right]$$

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

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



- Histogram Equalization

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$= p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right|$$

$$= \frac{1}{L-1}, \quad 0 \le s \le L-1$$

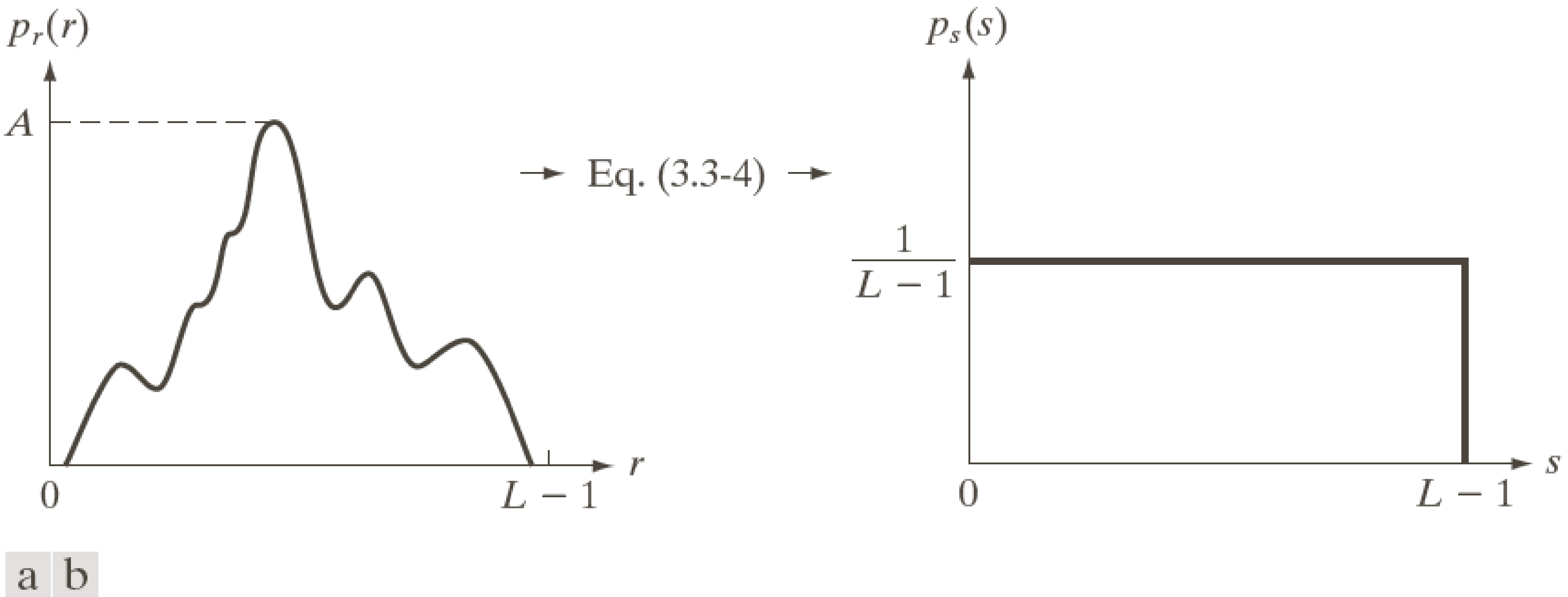
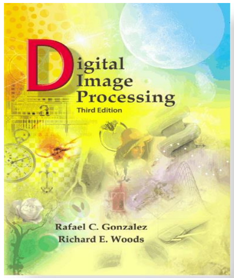


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

- Histogram Equalization Example

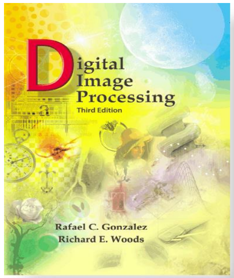
$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2} & \text{for } 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$$



- Histogram Equalization Example

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

$$= \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$



- Histogram Equalization Example

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

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

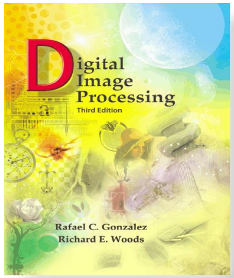
$$= \frac{1}{L-1}, \quad 0 \le s \le L-1$$

- Histogram Equalization – Discrete Probability

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

$$= \frac{L-1}{MN}\sum_{j=0}^{k} n_j \text{ for } k = 0,1,2,\ldots,L-1$$

$$\text{where } p_r(r_k) = \frac{n_k}{MN}$$



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

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

- Histogram Equalization – Example

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33\ [1]$$

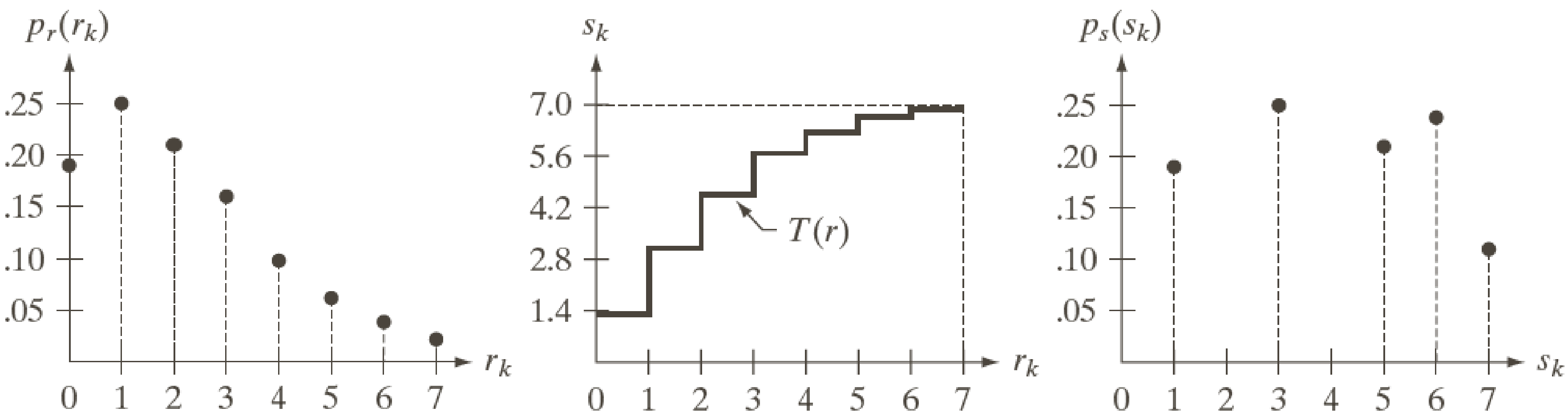
$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08\ [3]$$

$$s_2 = 4.55\ [4],\ s_3 = 5.67\ [5],\ s_4 = 6.23\ [6],$$

$$s_5 = 6.65\ [7],\ s_6 = 6.86\ [7],\ s_7 = 7.00\ [7]$$

Chapter 3

Intensity Transformations & Spatial Filtering



a b c

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

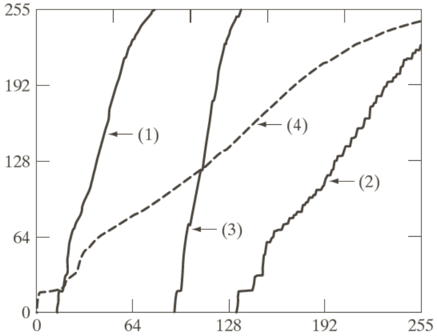
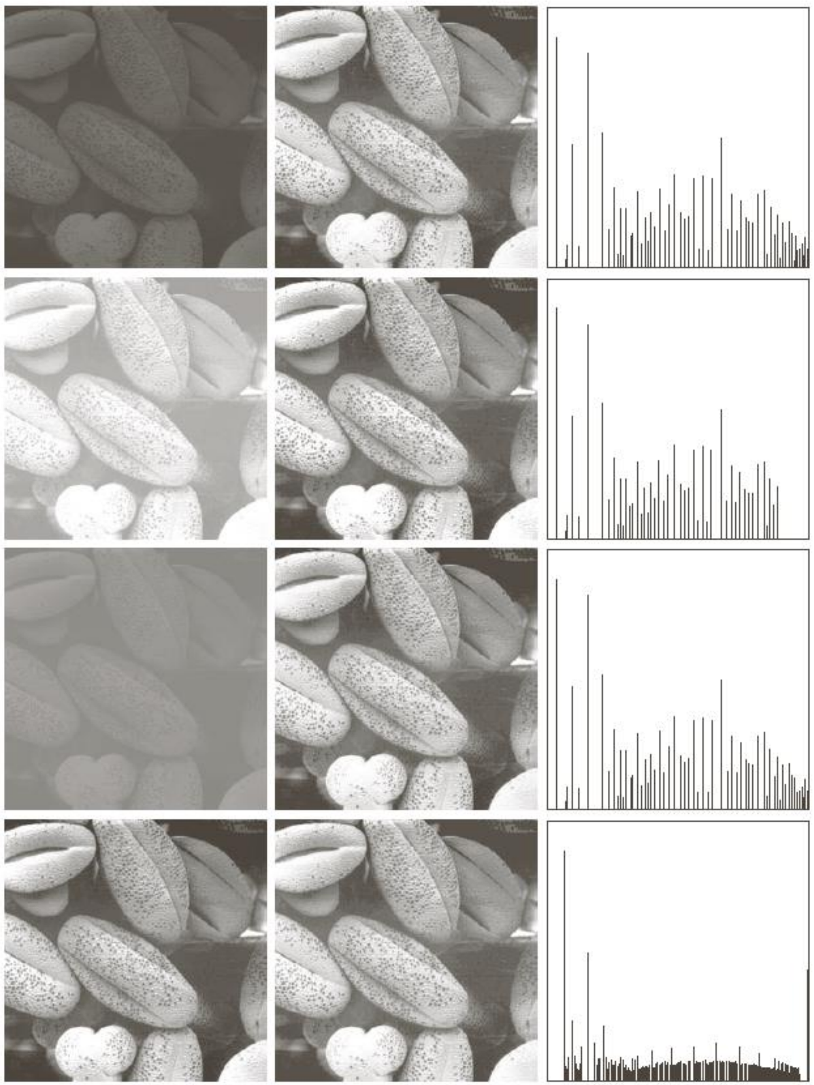


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.



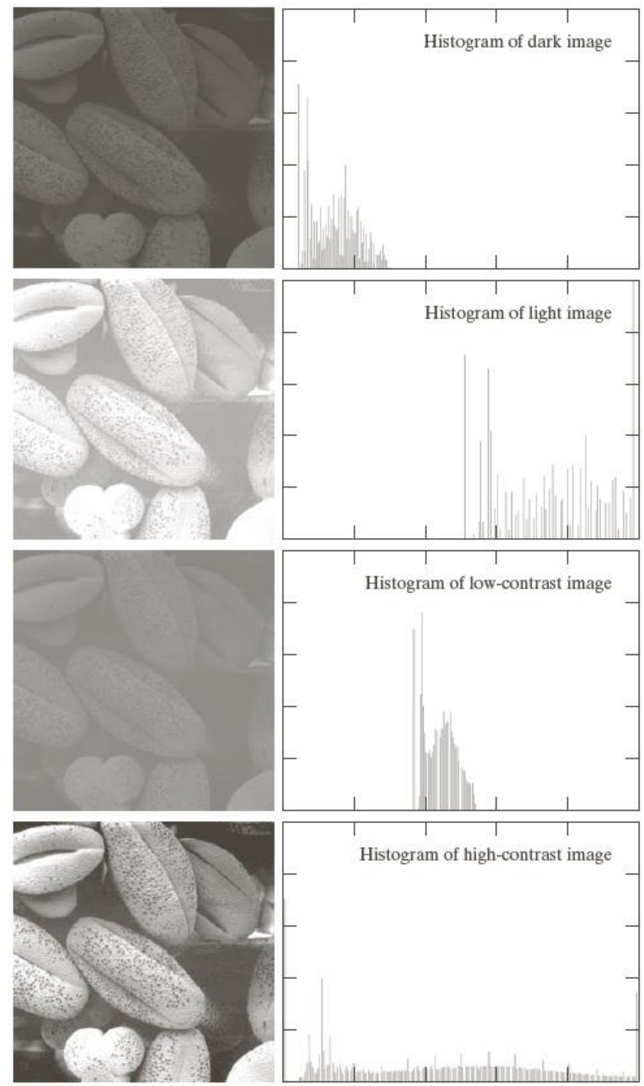


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

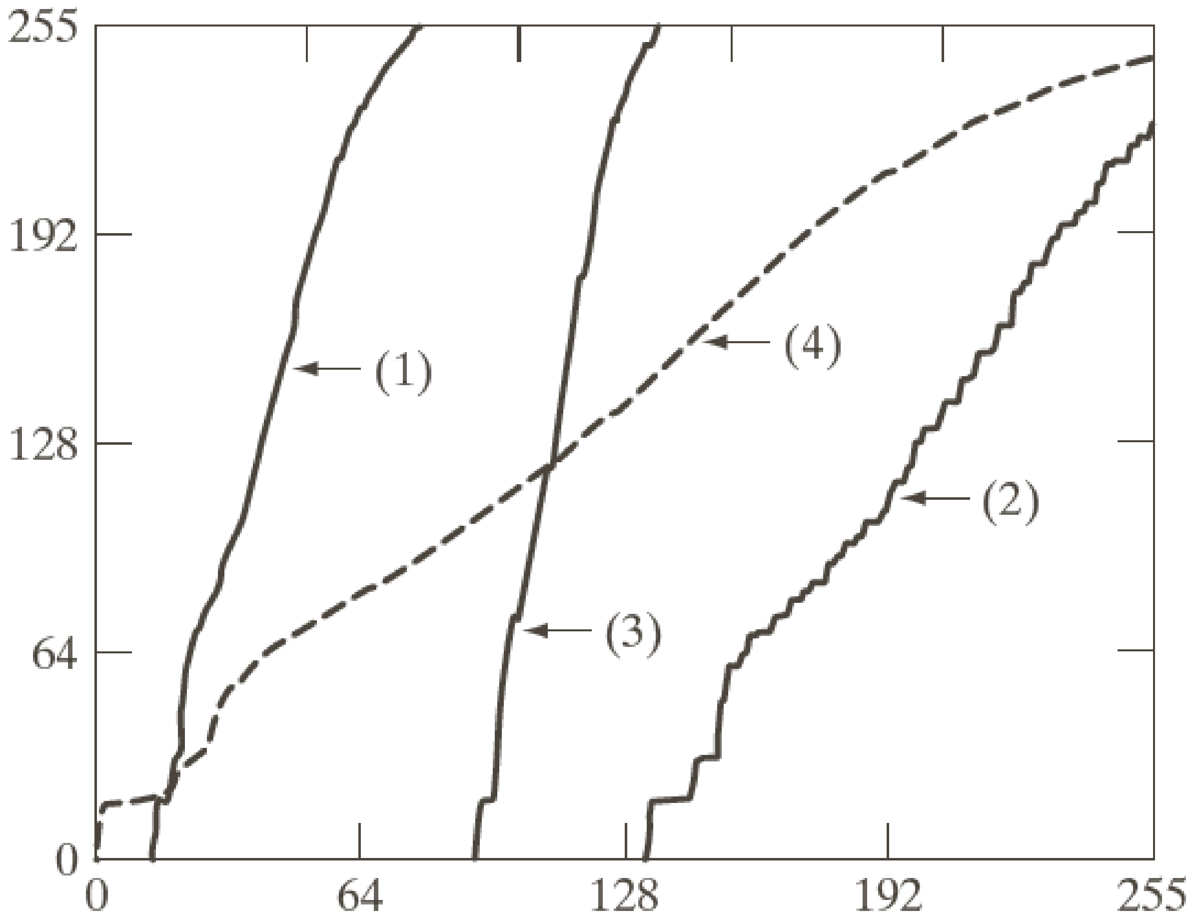
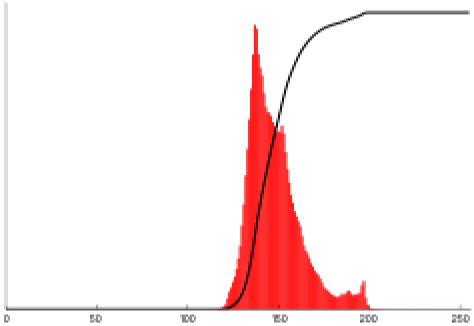
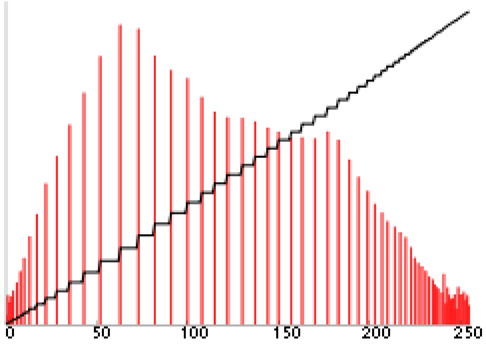


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

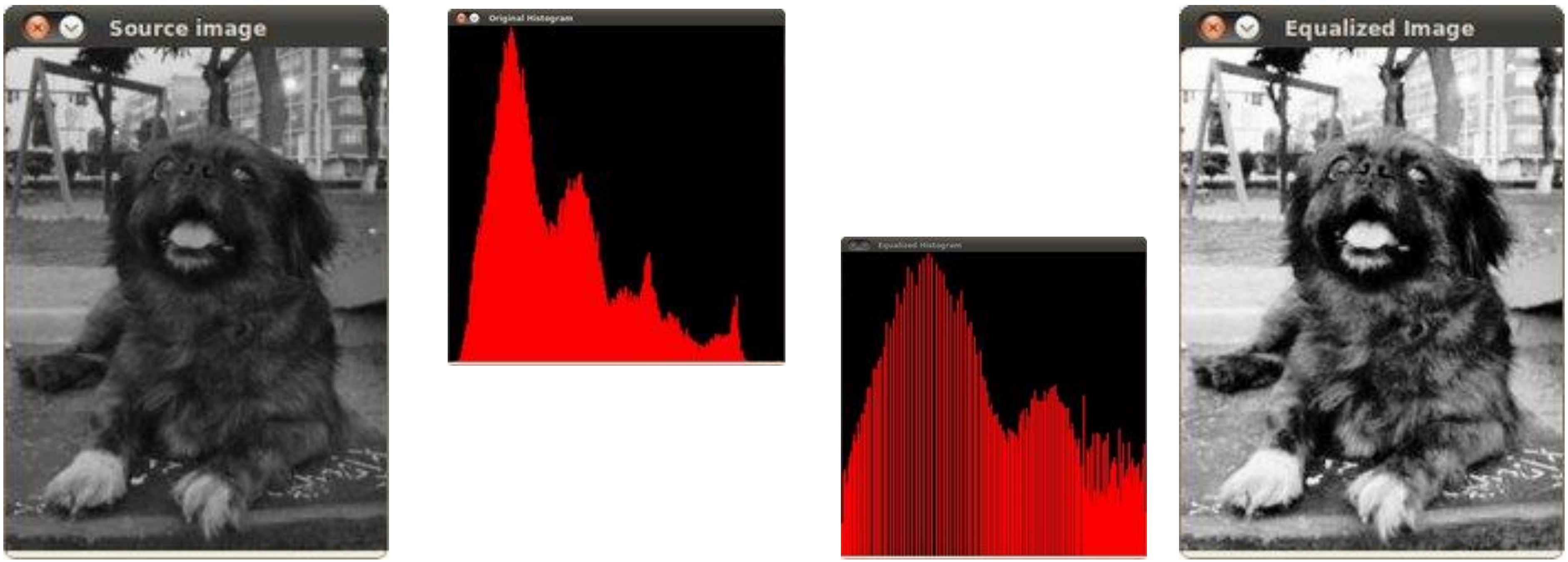


Histogram & Cumulative Histogram



Before & After Histogram Equalization

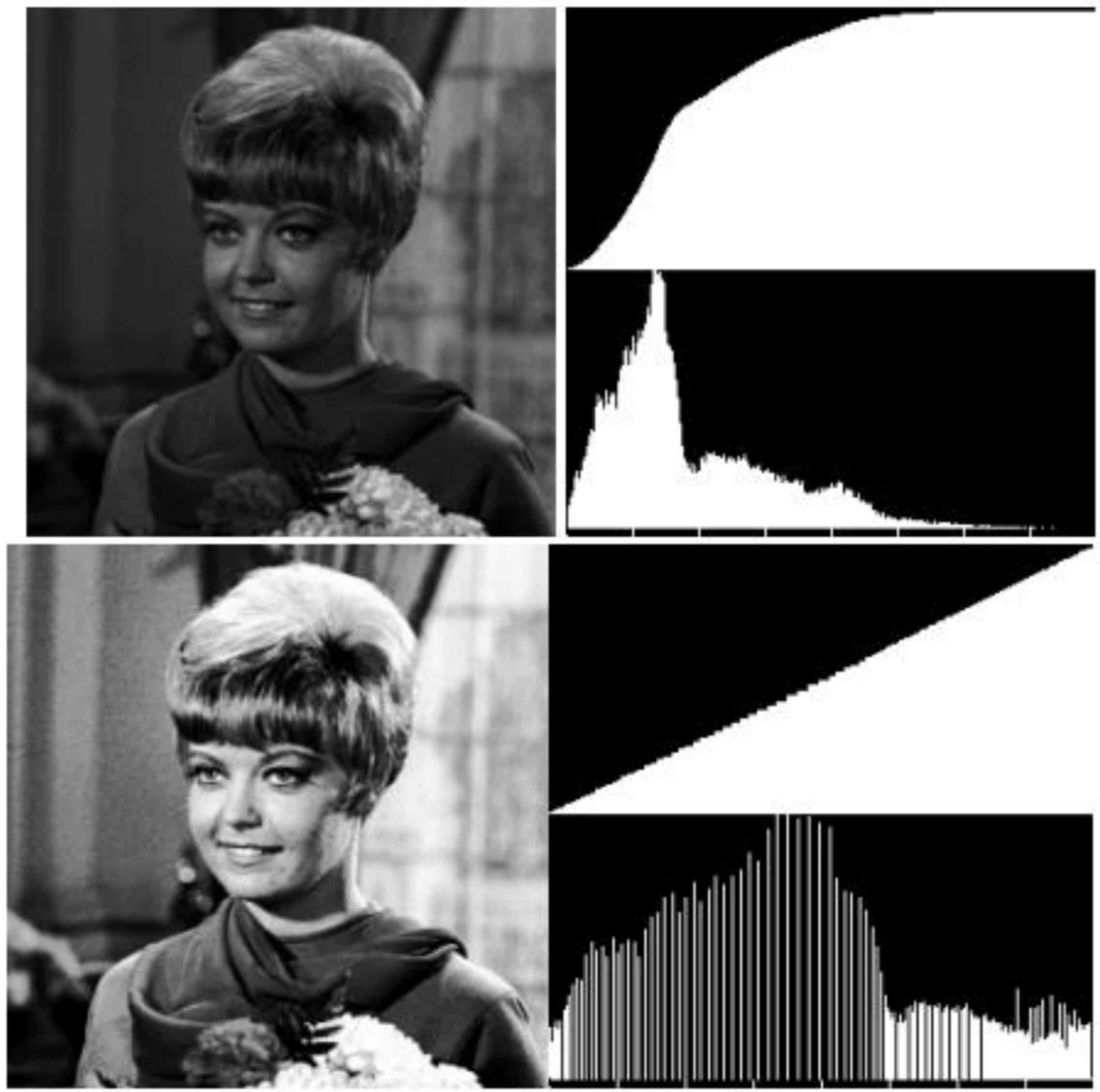
Source: https://en.wikipedia.org/wiki/Histogram_equalization



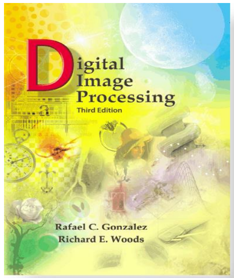
Before & After Histogram Equalization

Source: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html

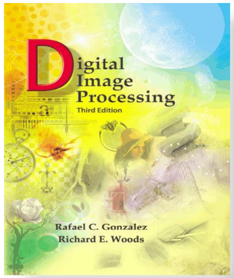
Before & After Histogram Equalization



Source: http://fourier.eng.hmc.edu/e161/lectures/HistogramEqualization.pdf

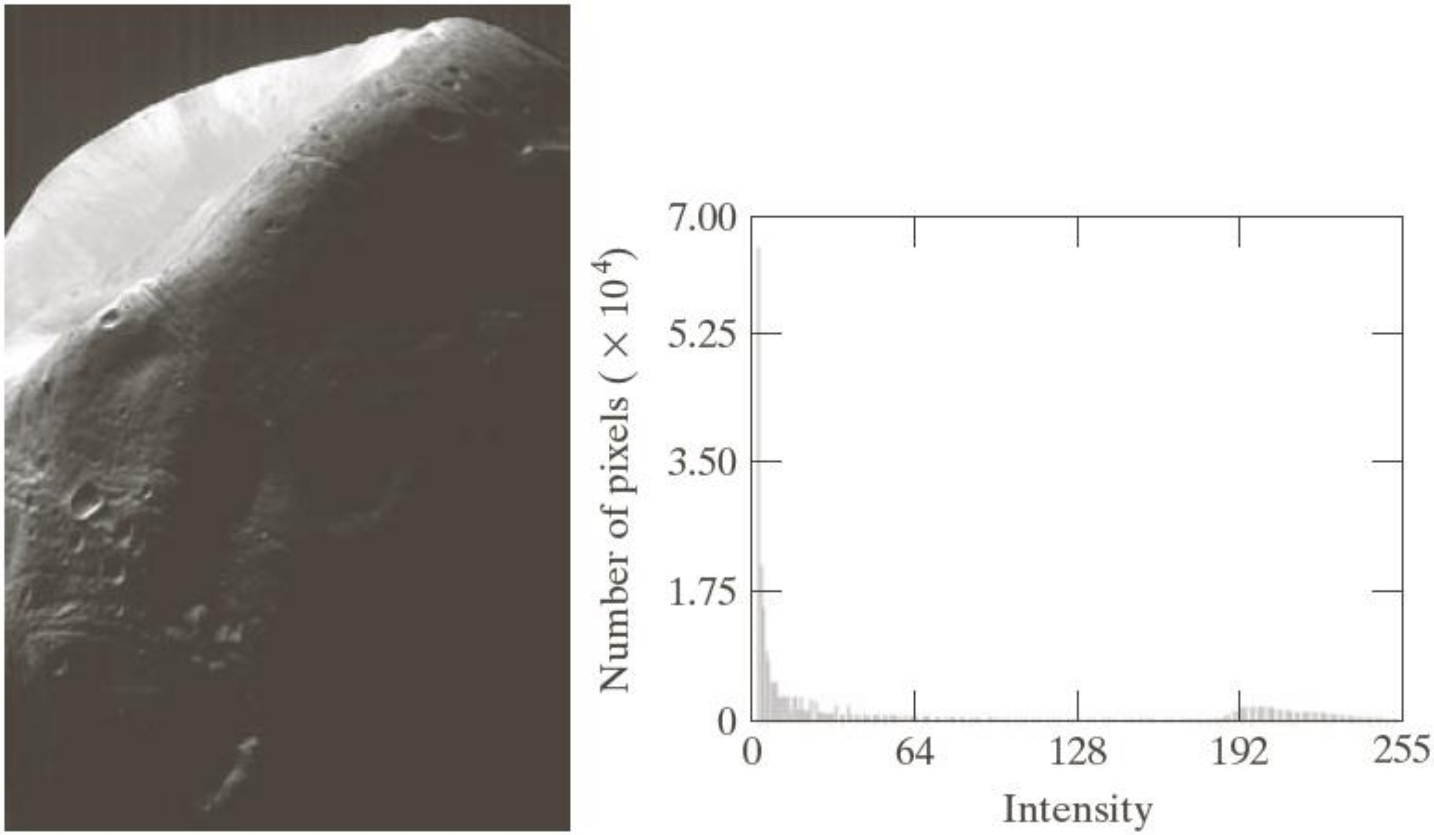


HISTOGRAM PROCESSING: SPECIFICATION



- Histogram Equalization – Failure Example
  - If the histogram is heavily skewed, equalization may not produce good result
  - Then we need to find transformation to a ‘desired’ histogram

Chapter 3
Intensity Transformations & Spatial Filtering

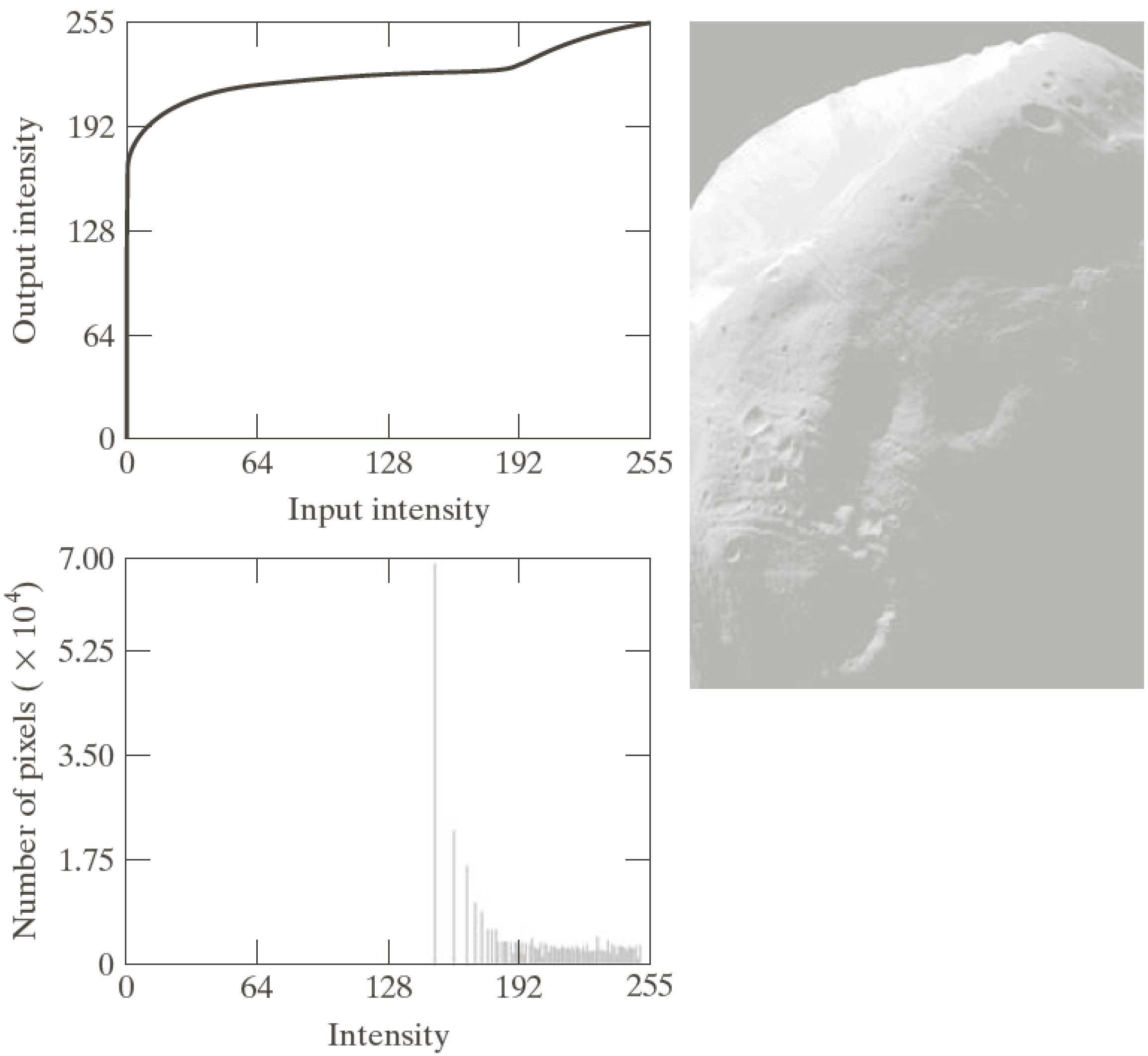


a b

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

Chapter 3

Intensity Transformations & Spatial Filtering

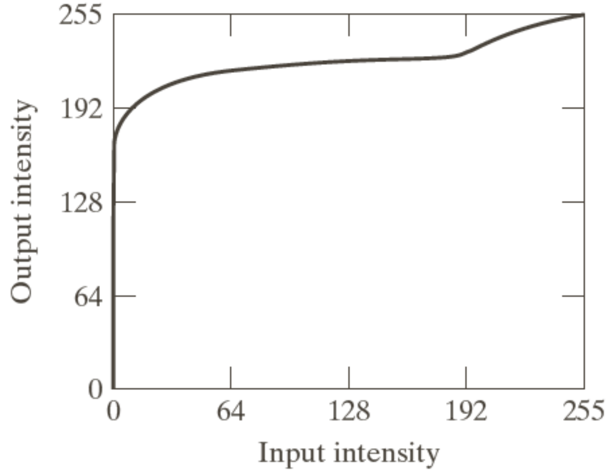


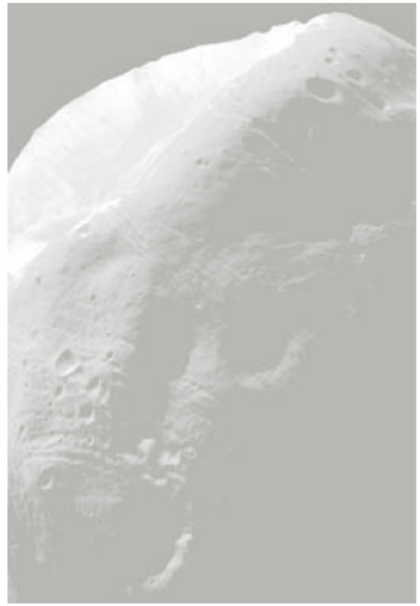
a b
c

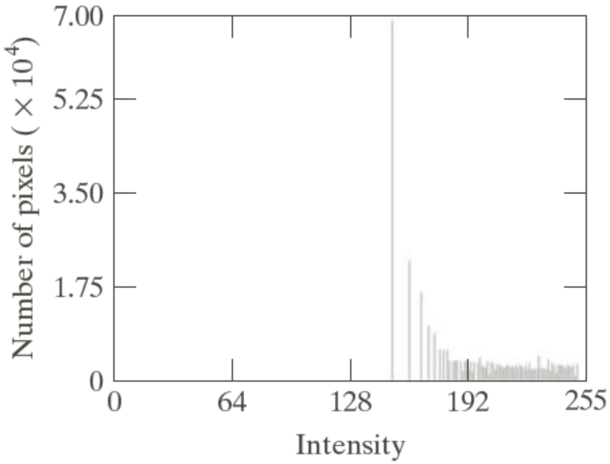
FIGURE 3.24
(a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).

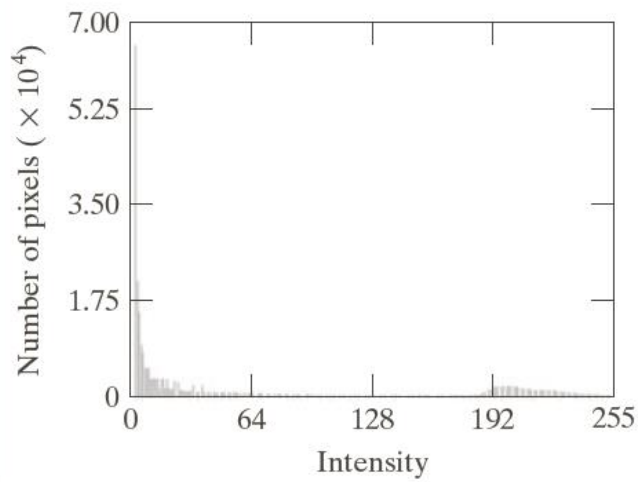
3

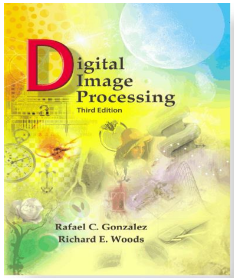
& Spatial Filtering











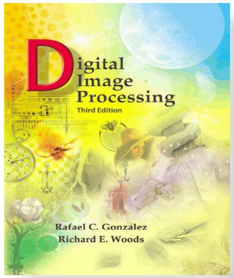
- Histogram Matching

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

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

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

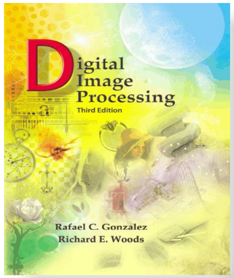
where $p_r(r)$ and $p_z(z)$ are i/p & o/p PDFs



- Histogram Matching Example

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

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



- Histogram Matching Example

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

$$= \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$

- Histogram Matching Example

$$G(z) = (L-1)\int_0^z p_z(w)\,dw$$

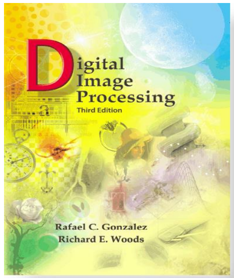
$$= \frac{3}{(L-1)^2}\int_0^z w^2\,dw = \frac{z^3}{(L-1)^2}$$

- Histogram Matching Example

$$G(z) = s \text{ or } \frac{z^3}{(L-1)^2} = s$$

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

$$= \left((L-1) r^2\right)^{1/3}$$



- Histogram Matching – Discrete Probability

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

$$= \frac{L-1}{MN}\sum_{j=0}^{k} n_j \text{ for } k = 0,1,2,\dots,L-1$$

$$\text{where } p_r(r_k) = \frac{n_k}{MN}$$

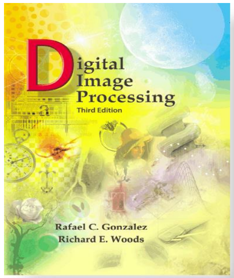
- Histogram Matching – Discrete Probability

$$G(z_q) = (L-1)\sum_{j=0}^{q} p_z(z_j)$$

for a value of q such that

$$G(z_q) = s_k \quad \text{or}$$

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



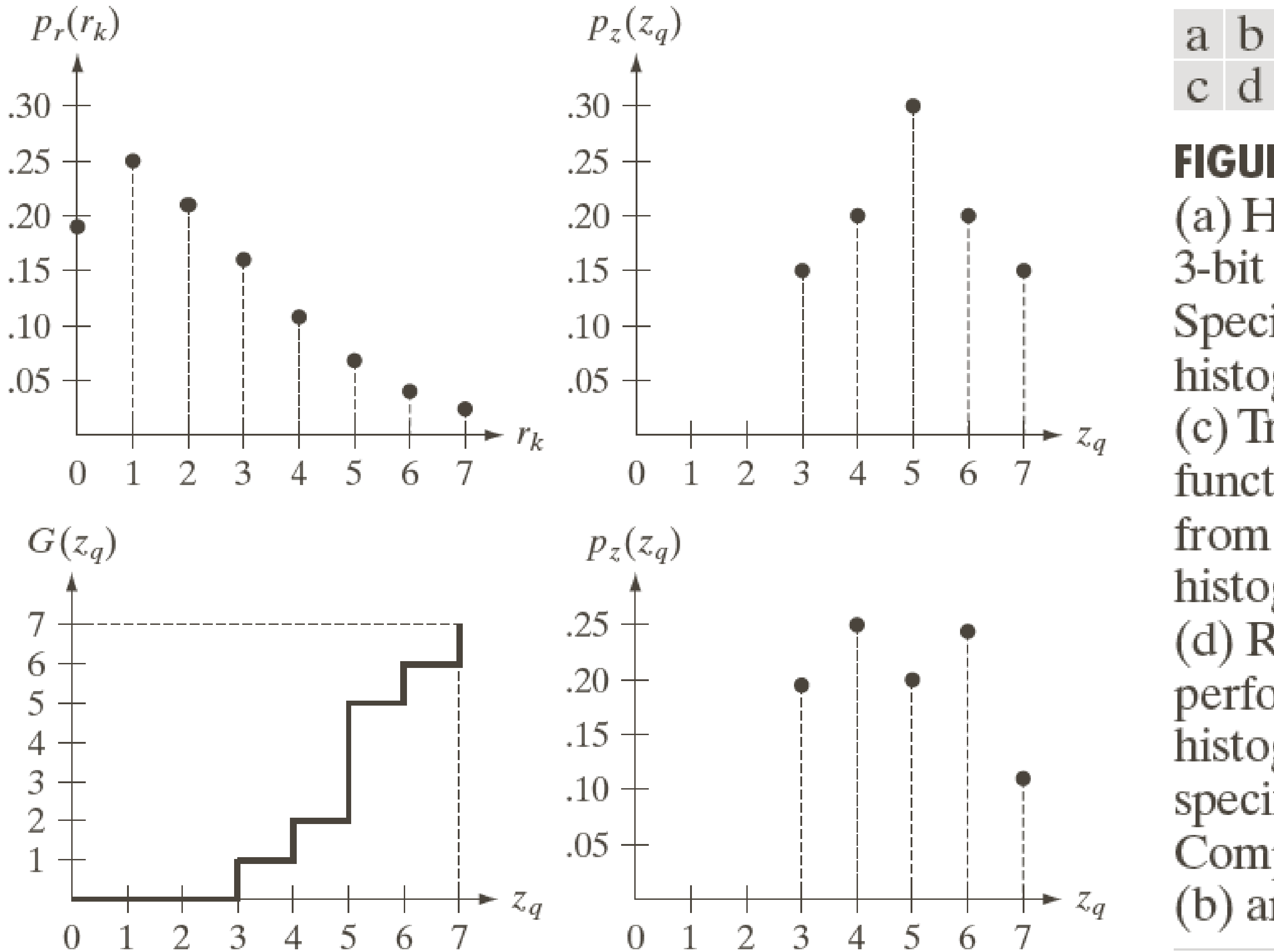


FIGURE 3.22
(a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

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

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

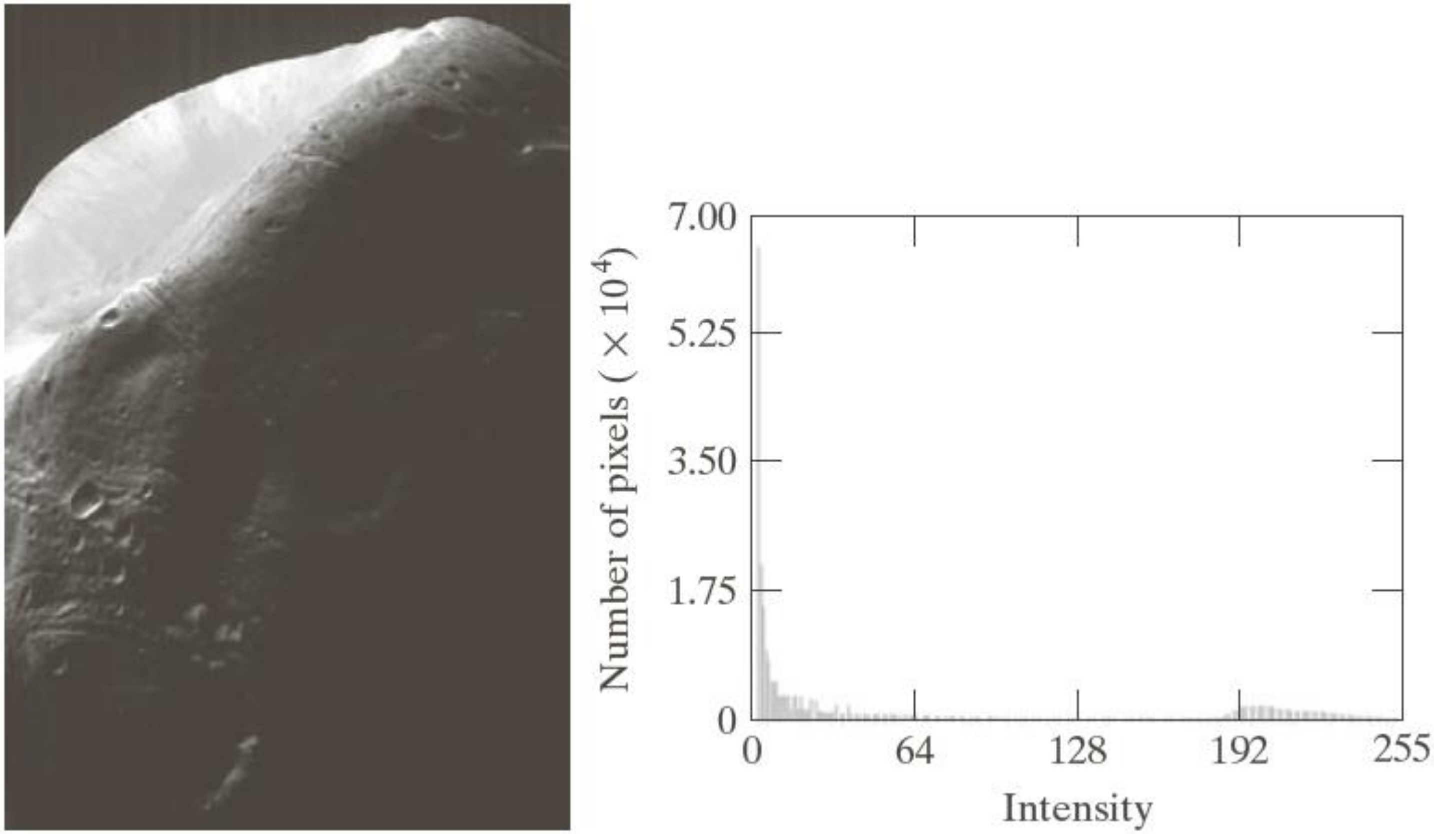
TABLE 3.3 All possible values of the transformation function G scaled, rounded, and ordered with respect to z.

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

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

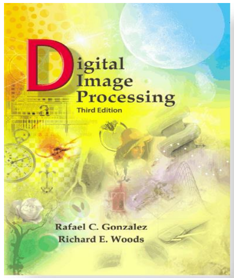
TABLE 3.4
Mappings of all the values of s_k into corresponding values of z_q.

r_k	$T(r_k) = s_k$	$T(r_k) = s_k$	$G(z_q)$	$G(z_q)$	z_q
0	1.33	1		0	0
1	3.08	3		0	1
2	4.55	5		0	2
3	5.67		1	1.05	3
4	6.23	6	2	2.45	4
5	6.65		5	4.55	5
6	6.86	7	6	5.95	6
7	7.00		7	7	7



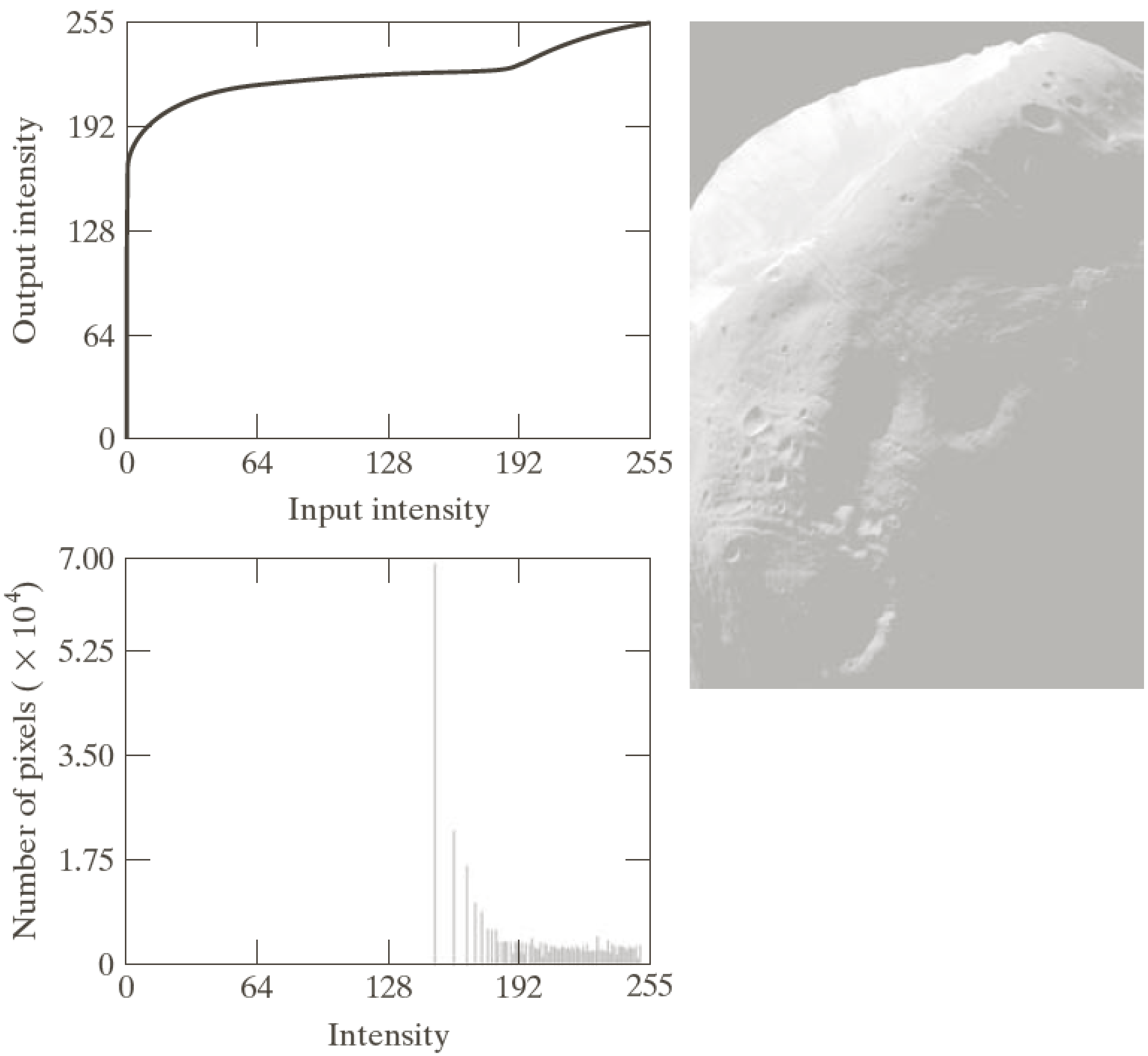
a b

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



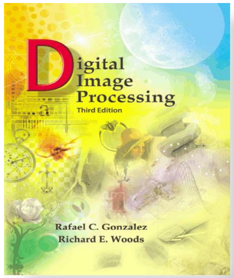
Chapter 3

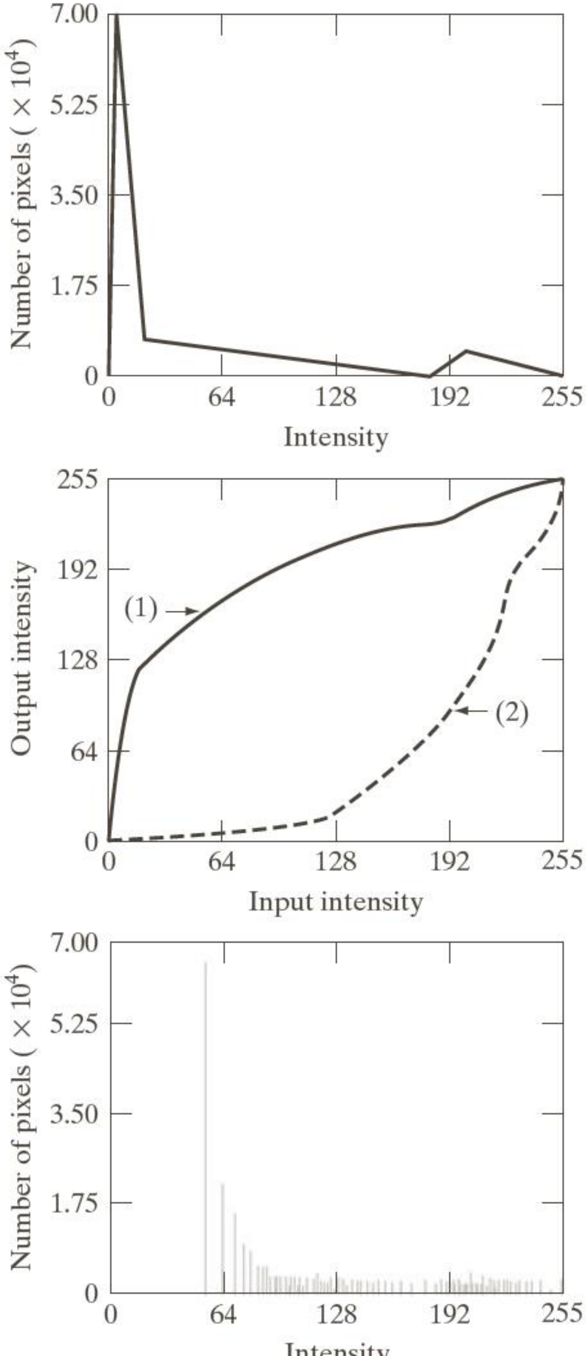
Intensity Transformations & Spatial Filtering

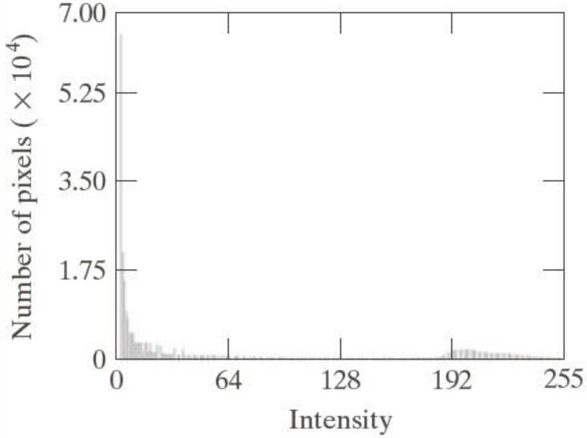


a b
c

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

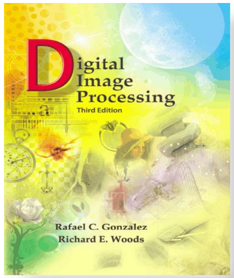




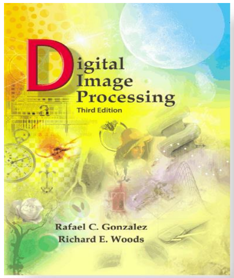


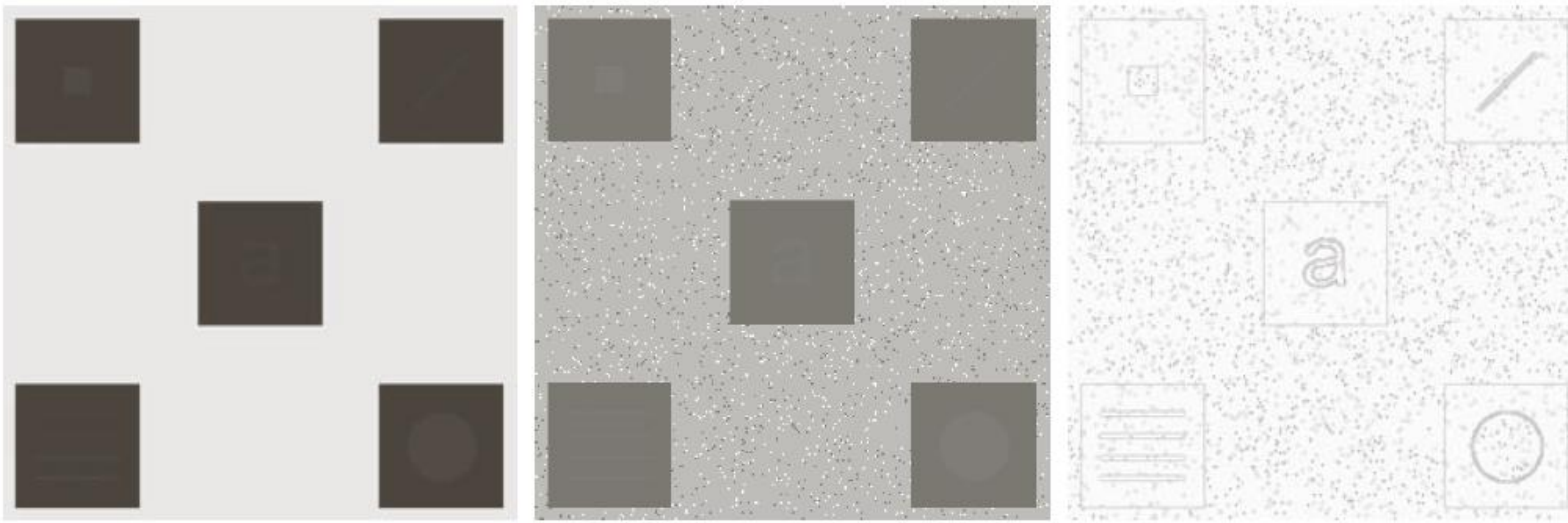
a c
b
d

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



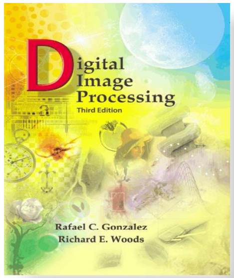
- Local Histogram Processing
  - Enhance details in local areas
- Local Processing Steps
  - Define a Neighborhood
  - Move its center from pixel to pixel
  - Apply histogram equalization / matching @ center
  - Non-overlapping computation is fast but blocky





a b c

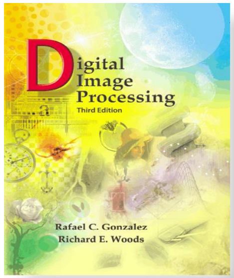
FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.



- Histogram Statistics (Continuous)

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

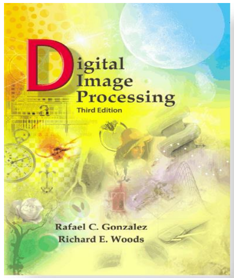
$$\mu_2(r) = \sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 \, p(r_i)$$



- Histogram Statistics (Discrete)

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

$$\mu_2(r) = \sigma^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(f(x,y) - m\right)^2$$



- Histogram Statistics (Discrete): Example

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

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

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

$$m = \sum_{i=0}^{3} r_i p(r_i)$$

$$= 0*0.24 + 1*0.28 + 2*0.28 + 3*0.20$$

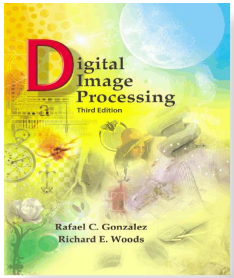
$$= 1.44$$

- Histogram Statistics – Local Mean & Variance

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

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

$$\text{where } S_{xy} \text{ is a sub-image at } (x, y)$$

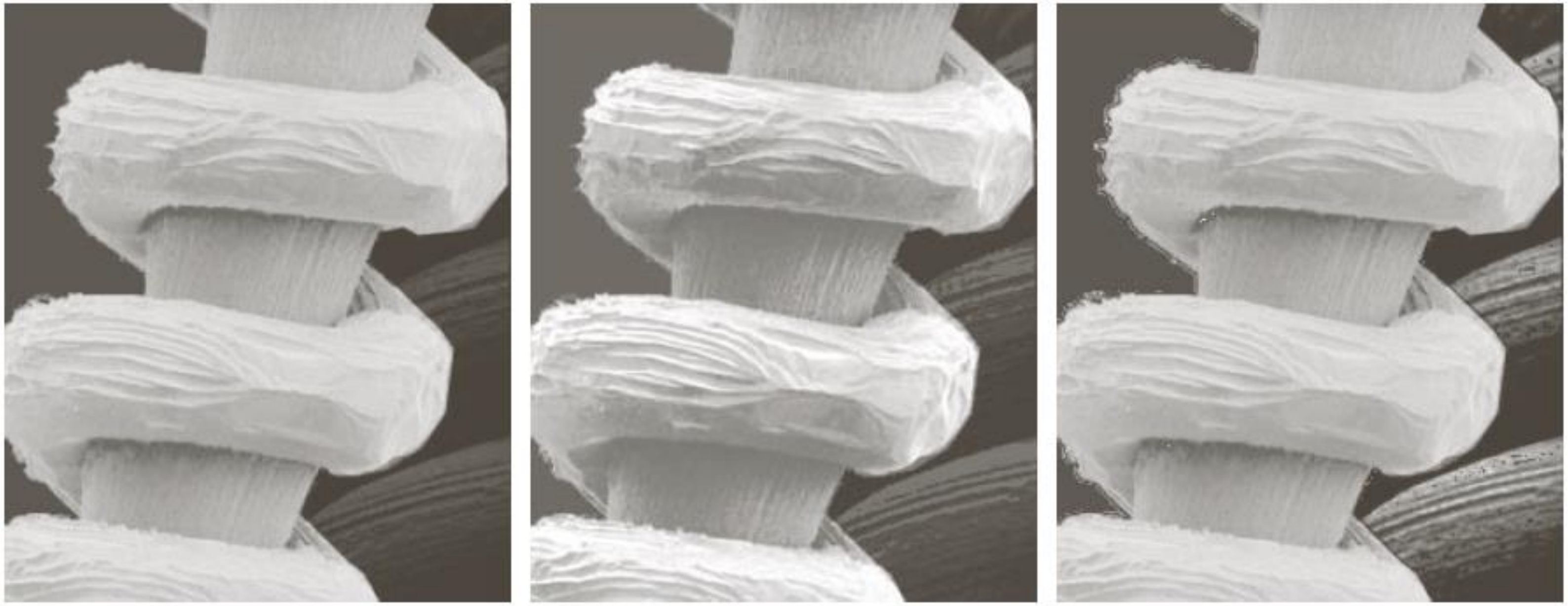


- Enhancement with Local Mean & Variance

$$g(x,y)=\begin{cases} E.f(x,y), \text{ if } m_{S_{xy}} \le k_0 m_G \ \& \\ \qquad\qquad k_1\sigma_G \le \sigma_{S_{xy}} \le k_2\sigma_G \\ f(x,y), \quad \text{otherwise} \end{cases}$$

where m_G is the global mean and

σ_G is the global standard deviation



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.)

$$E = 4.0, k_0 = 0.4, k_1 = 0.02, k_2 = 0.4$$

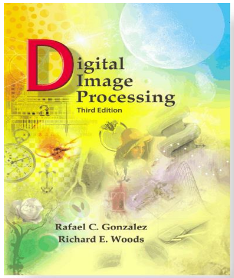
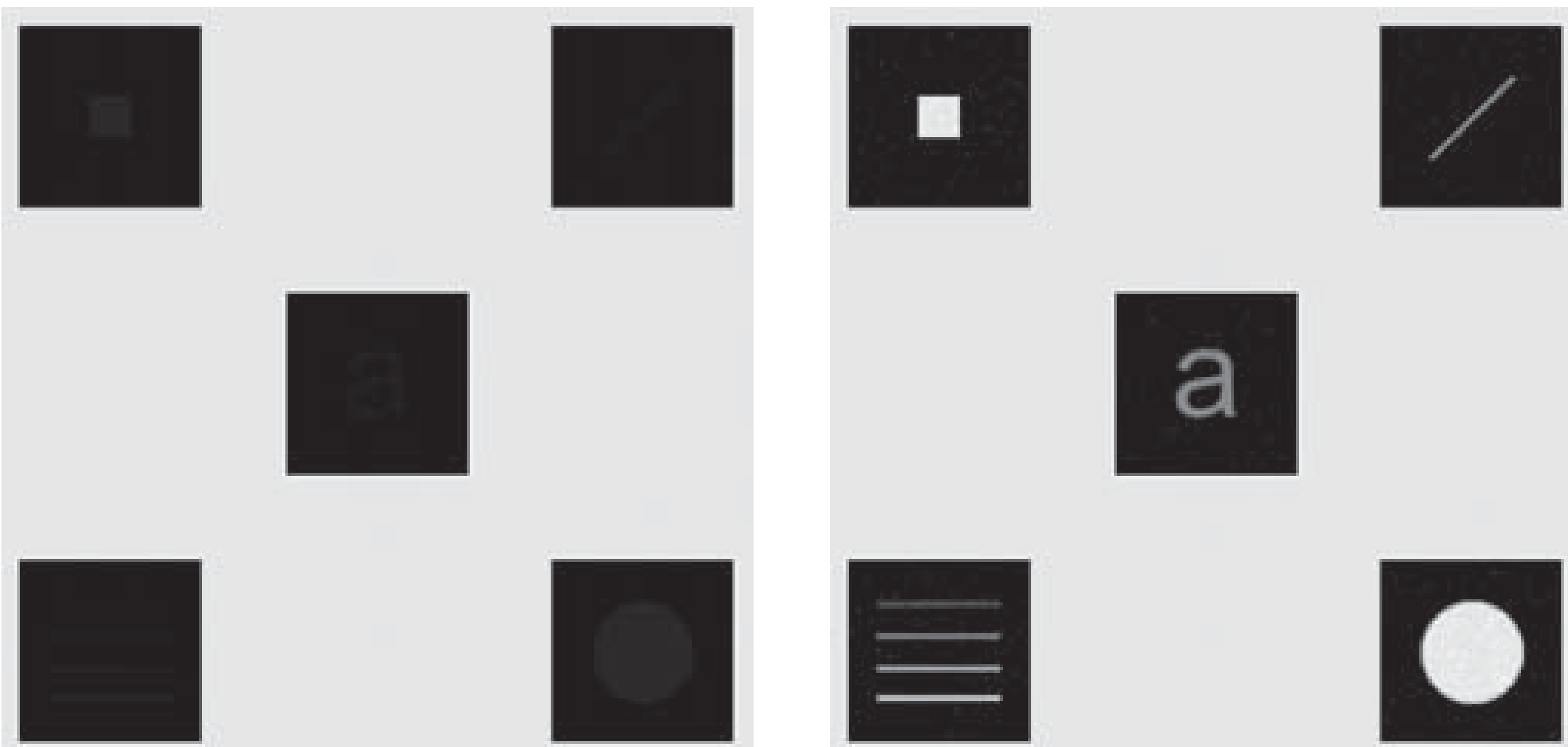
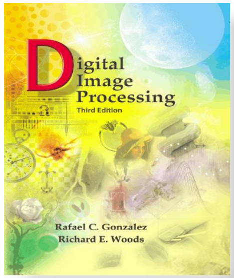


Image Enhancement using Histogram Statistics:

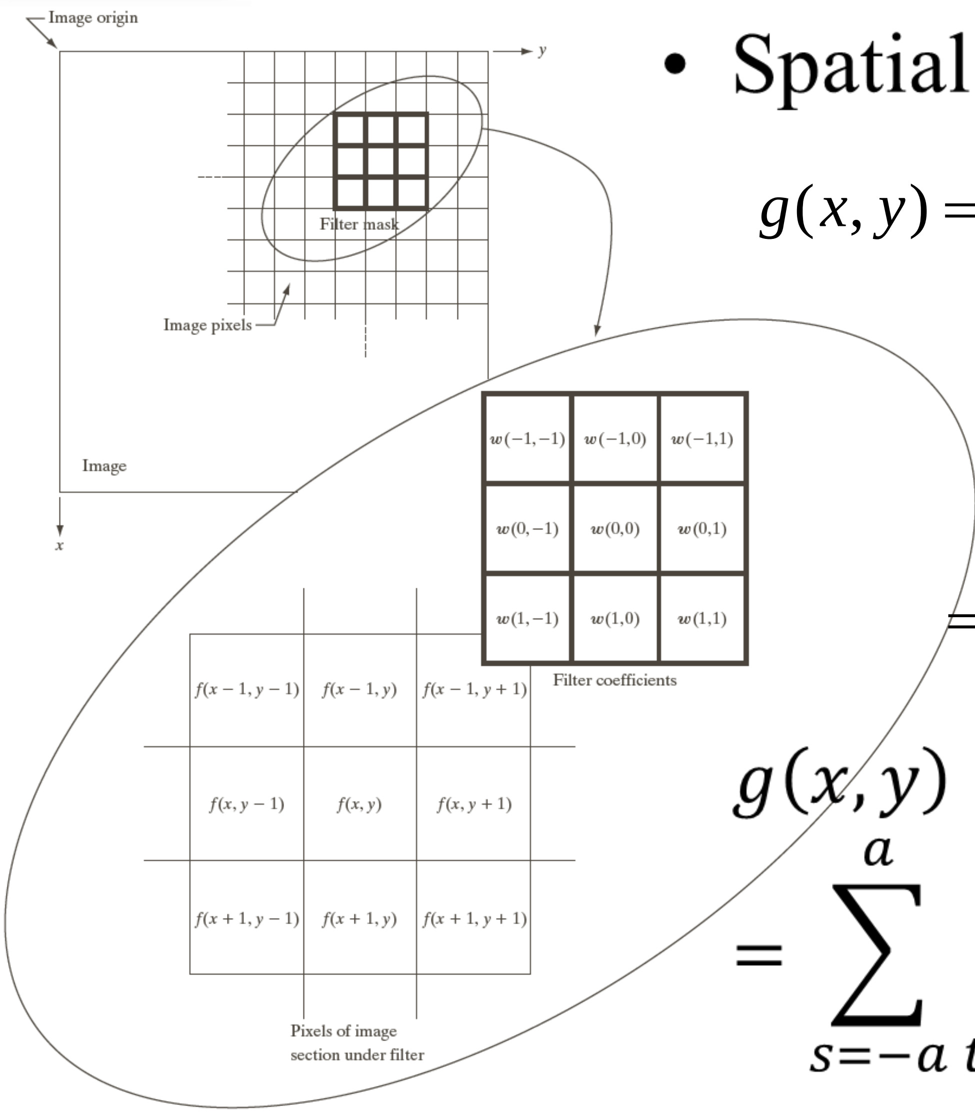
$E = 22.8,\ K_0 = 0.1,\ K_2 = 0.1$





SPATIAL FILTERING

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



- Spatial Filtering

$$
\begin{aligned}
g(x,y) = {} & w(-1,-1)\,f(x-1,y-1) + \\
& w(-1,0)\,f(x-1,y) + \cdots + \\
& w(0,0)\,f(x,y) + \cdots + \\
& w(+1,+1)\,f(x+1,y+1) \\
= {} & \sum_{\delta x=-1}^{+1} \sum_{\delta y=-1}^{+1} w(\delta x,\delta y)\,f(x+\delta x, y+\delta y)
\end{aligned}
$$

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

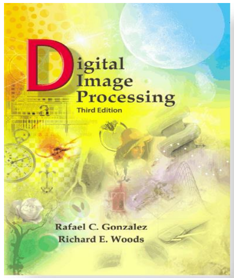


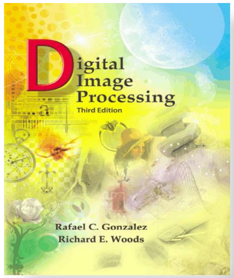
FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

- Vector Notation
  - Consider a 3X3 mask
  - Compute the response

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$$

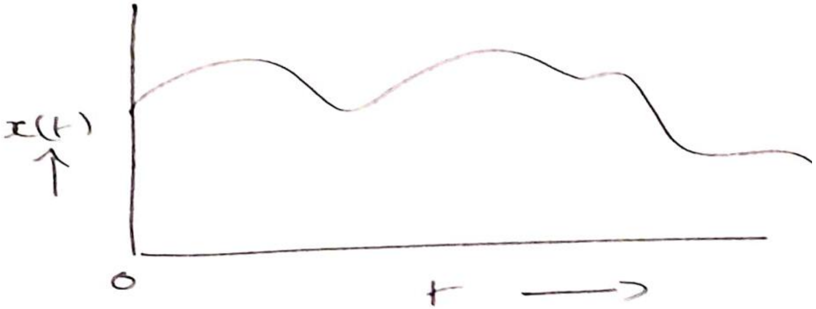
$$= \sum_{k=1}^{9} w_k z_k$$



- Spatial Filtering
  - Correlation
    - Process of moving the filter mask over the image and compute the sum-of-products at every location
  - Convolution
    - Same as correlation except the filter is first rotated by 180°

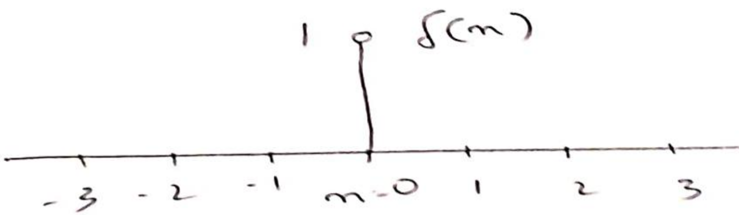
Appropriately pad enough 0's on all sides so that the mask can cover the function (image) at every position

Analog Signal

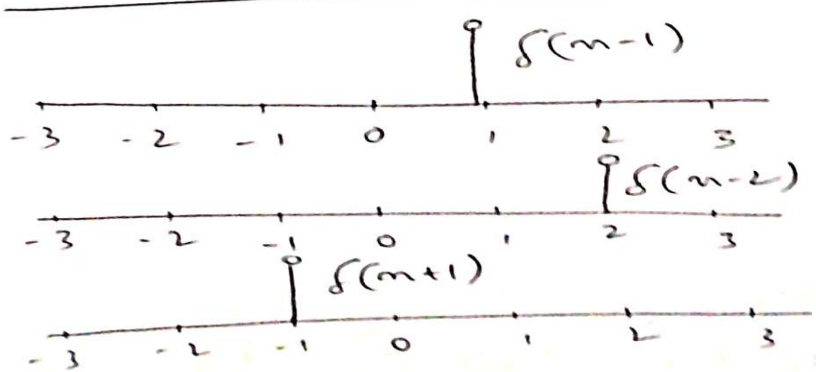


Unit Sample

$$\delta(n) = 1 \;;\; n = 0$$
$$= 0 \;;\; n \neq 0$$

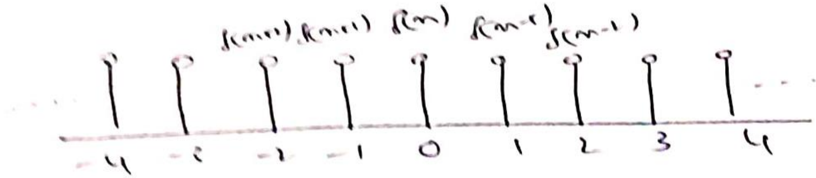


Shifted unit Samples



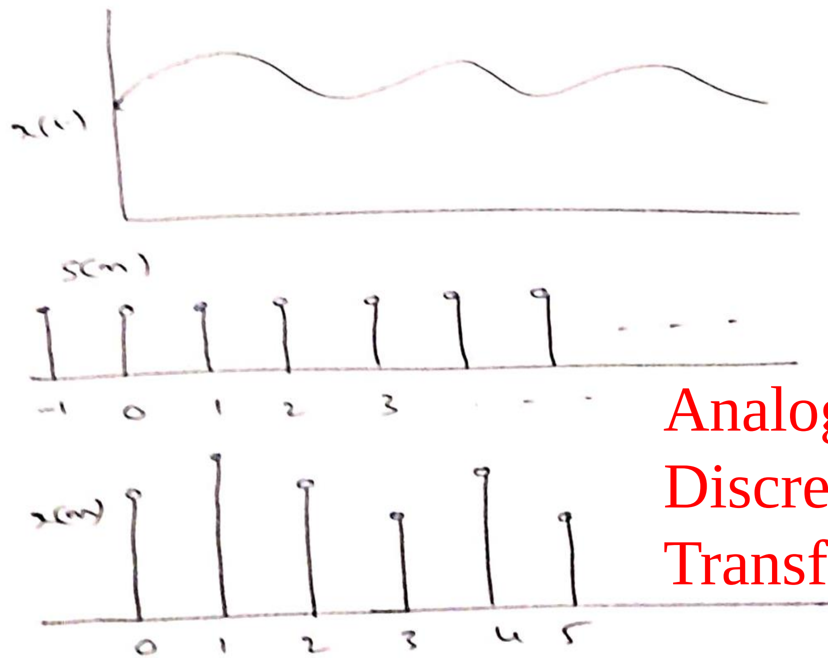
Unit Sample Sequence

$$S(n) = \sum_{k=-\infty}^{\infty} \delta(n-k)$$



Discrete Signal

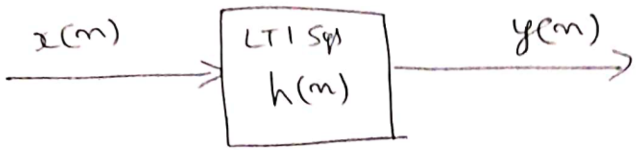
$$x(n) = x(t) \times S(n)$$



$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

Analog to Discrete Transformation

Concepts of Convolution & correlation

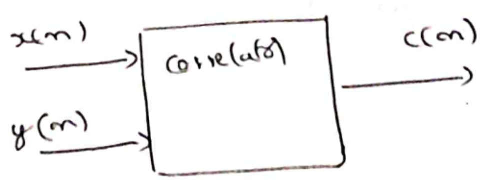


I/p	O/p
$\delta(n)$	$h(n)$
$\delta(n-k)$	$h(n-k)$
$x(k)\,\delta(n-k)$	$x(k)\,h(n-k)$
$\sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$	$\sum_{k=-\infty}^{\infty} x(k)\,h(n-k)$
$x(n)$	$y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k)$

$$y(n) = x(n) * h(n)$$

($*$ — convolution operation)

Correlation



$$c(n) = \sum_{k=-\infty}^{\infty} x(k)\,y(n+k)$$

Convolution in Time & Frequency domain

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\,h(n-k)$$

$$= x(n) * h(n)$$

$$FT\left[y(n)\right] = FT\left[x(n) * h(n)\right]$$

$$Y(\omega) = X(\omega)\,H(\omega)$$

$$\Rightarrow FT\left[x(n) * h(n)\right] = FT\left[x(n)\right] FT\left[h(n)\right]$$

$$Y(\omega) = X(\omega)\,H(\omega)$$

$$x(n) * h(n) \overset{FT}{\Longleftrightarrow} X(\omega)\,H(\omega)$$

||ly

$$X(\omega) * H(\omega) \overset{FT}{\Longleftrightarrow} x(n)\,h(n)$$

Convolution & Correlation

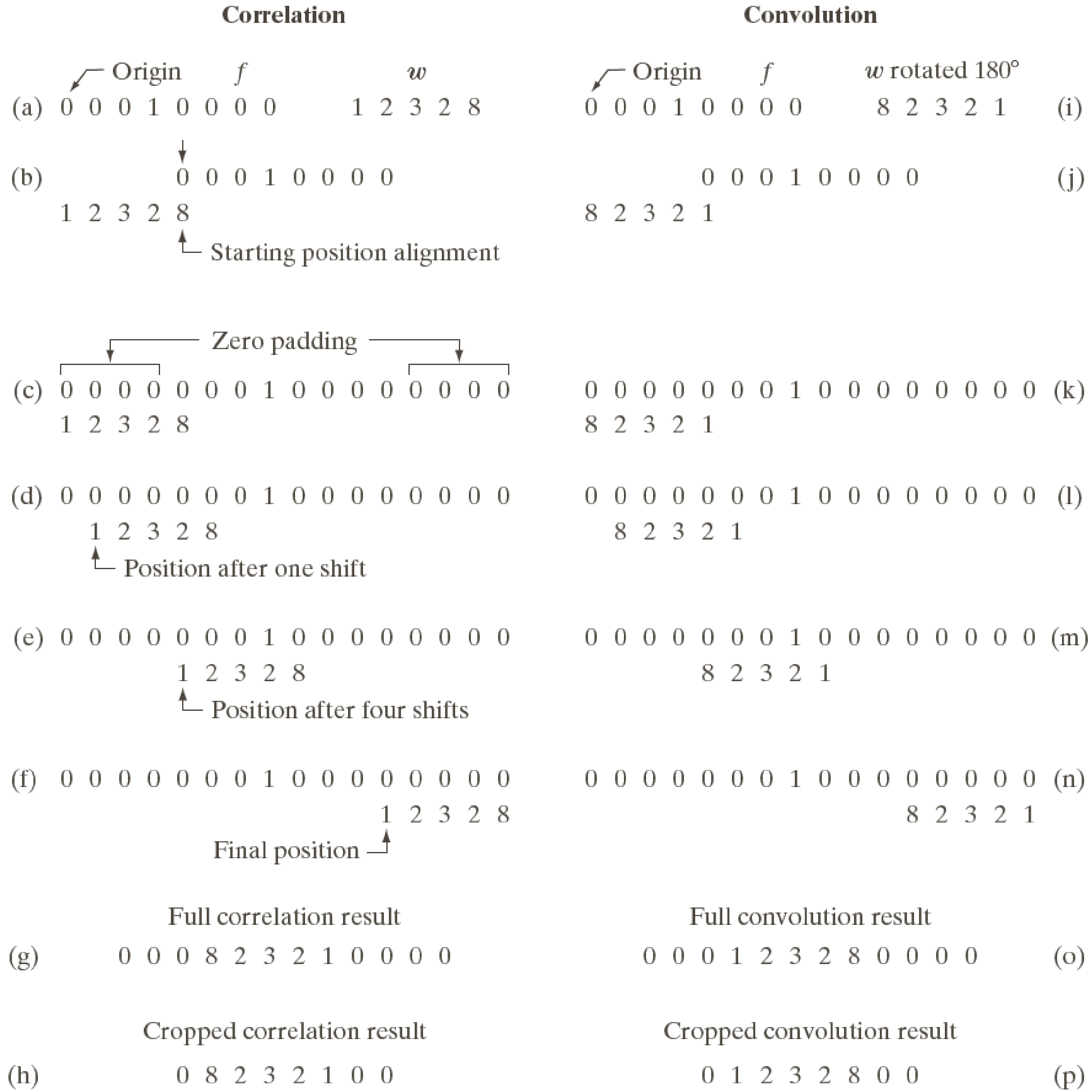
1D: Convolution & Correlation

FIGURE 3.29 Illustration of 1-D correlation and convolution of a kernel, *w*, with a function *f* consisting of a discrete unit impulse. Note that correlation and convolution are functions of the variable *x*, which acts to *displace* one function with respect to the other. For the extended correlation and convolution results, the starting configuration places the right-most element of the kernel to be coincident with the origin of *f*. Additional padding must be used.

	Correlation	Convolution	
	Origin, *f*, *w*	Origin, *f*, *w* rotated 180°	
(a)	0 0 0 1 0 0 0 0 1 2 4 2 8	0 0 0 1 0 0 0 0 8 2 4 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 4 2 8 Starting position alignment	0 0 0 1 0 0 0 0 8 2 4 2 1 Starting position alignment	(j)
(c)	Zero padding 0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Starting position	Zero padding 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Starting position	(k)
(d)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Position after 1 shift	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Position after 1 shift	(l)
(e)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Position after 3 shifts	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Position after 3 shifts	(m)
(f)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Final position	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Final position	(n)
	Correlation result	**Convolution result**	
(g)	0 8 2 4 2 1 0 0	0 1 2 4 2 8 0 0	(o)
	Extended (full) correlation result	**Extended (full) convolution result**	
(h)	0 0 0 8 2 4 2 1 0 0 0 0	0 0 0 1 2 4 2 8 0 0 0 0	(p)

Correlation & Convolution for 1D signals

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



Extended 1D Cerrelation & 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*.

2D Cropped Correlation & Convolution

FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D kernel with an image consisting of a discrete unit impulse. The 0's are shown in gray to simplify visual analysis. Note that correlation and convolution are functions of x and y. As these variable change, they *displace* one function with respect to the other. See the discussion of Eqs. (3-36) and (3-37) regarding full correlation and convolution.

(a) Origin, f

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

w

1	2	3
4	5	6
7	8	9

(b) Padded f

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

(c) Initial position for w

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

(d) Correlation result

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

(e) Full correlation result

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

(f) Rotated w

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

(g) Convolution result

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

(h) Full convolution result

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

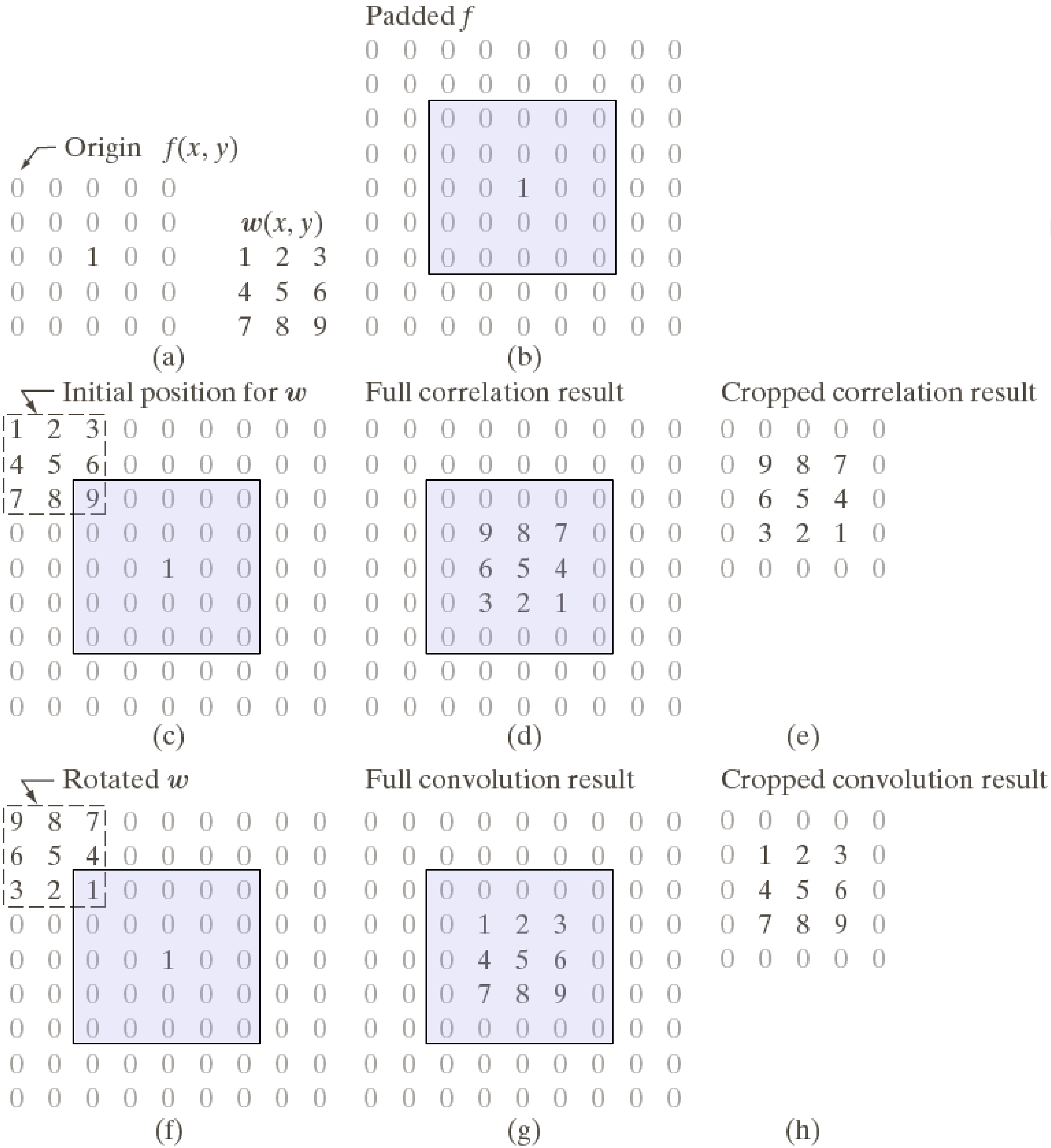
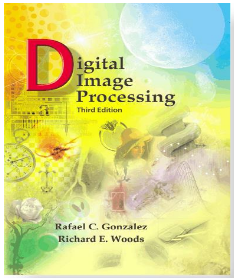


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Extended
2D Correlation
& Convolution



- Spatial Filtering
  - Correlation

$$m \times n \text{ mask}$$

$$a = (m-1)/2, b = (n-1)/2$$

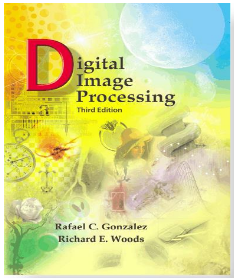
$$w(x,y) \circ f(x,y) = \sum_{s=-a}^{+a} \sum_{t=-b}^{+b} w(s,t) f(x+s, y+t)$$

$$= f(x,y) \circ w(x,y)$$

  - Convolution

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

$$= f(x,y) \bullet w(x,y)$$



Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \,☆\, (g + h) = (f \,☆\, g) + (f \,☆\, h)$

Separable Filter kernels

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

is separable because it can be expressed as the outer product of the vectors

$$\mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{r} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

That is,

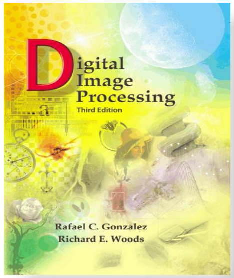
$$\mathbf{c}\,\mathbf{r}^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = w$$

A separable kernel of size $m \times n$ can be expressed as the outer product of two vectors, $\mathbf{v}$ and $\mathbf{w}$:

$$w = \mathbf{v}\mathbf{w}^T \qquad (3\text{-}41)$$

Computational efficiency =

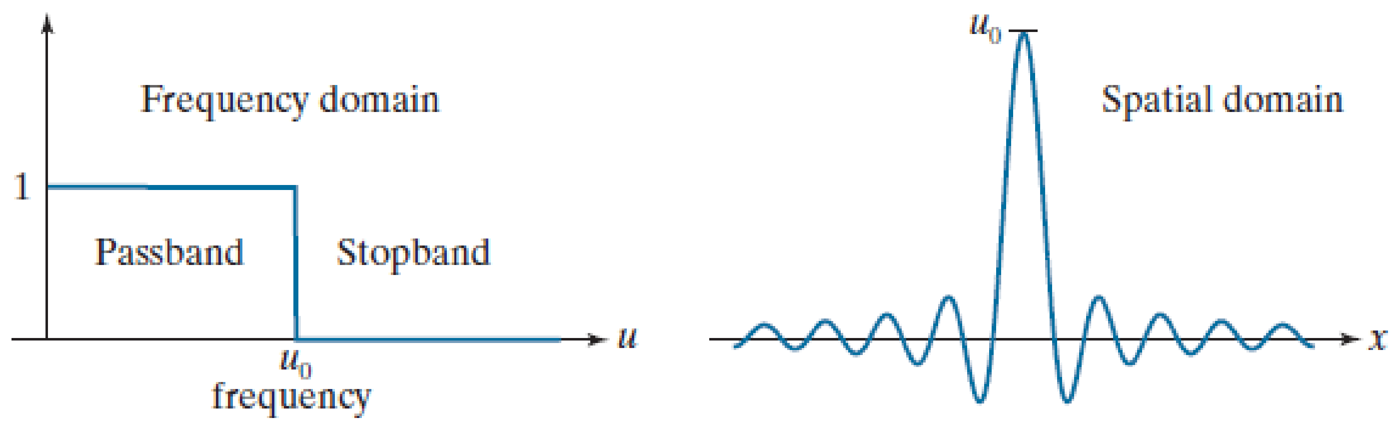
$$C = \frac{MNmn}{MN(m+n)} = \frac{mn}{m+n}$$



Filtering in Frequency and Spatial Domains

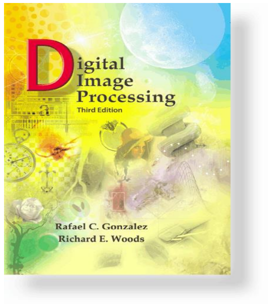
a b

FIGURE 3.32
(a) Ideal 1-D low-pass filter transfer function in the frequency domain.
(b) Corresponding filter kernel in the spatial domain.

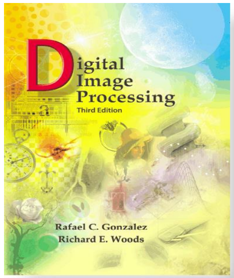


Construction of Lowpass Spatial Kernels

- Based on mathematical properties
- Sampling the 2D/1D spatial function whose shape has desired property
- 1D spatial filter with desired freq response using digital filter design techniques.



SMOOTHING FILTER



- Smoothing Filters
  - Average

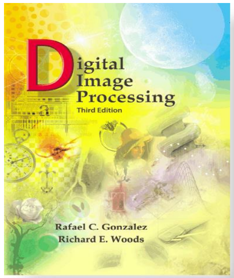
$$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$

$m \times n$ mask

$a = (m-1)/2, b = (n-1)/2$

  - Weighted Average

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



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

Box Filter

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

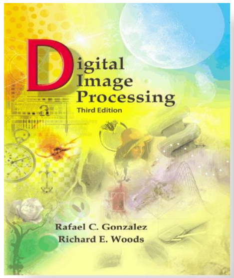
Gaussian Filter

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.

Why Normalization?

- Avg value of the uniform intensity regions remains same
- Prevents to introduce bias in the process of filtering



Smoothing Spatial Filters (Lowpass Filters)

- Blurring

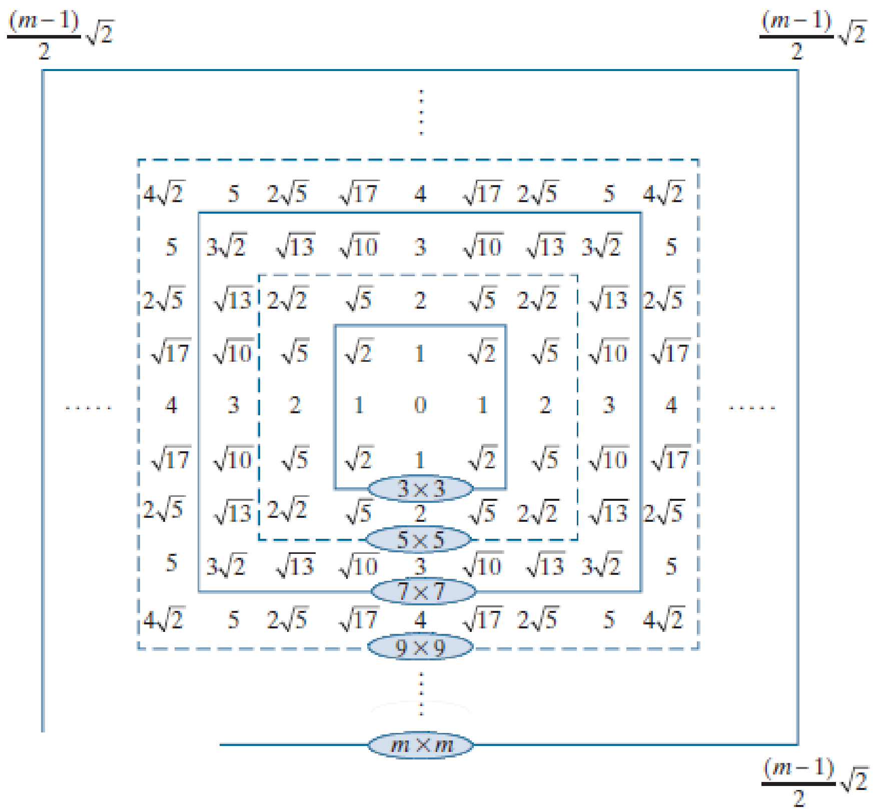
  Degree of blurring depends on the size of the kernel and values of the kernel coefficients

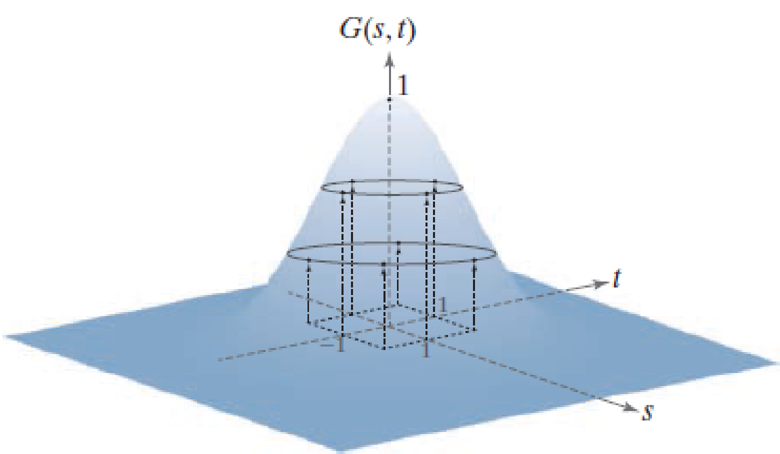
  - Preprocessing tasks
    - removal of small details (irrelevant) from an image
    - Removing false contours
    - Bridging of small gaps in lines and curves
    - Anti-aliasing before sampling
- Noise reduction
- Box filter
  - ✓ Simple & separable LP
  - ✓ Reduce effect of smoothing on edges
  - ✓ Blurring is not uniform (limited to perpendicular directions)

Gaussian Filter

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

$$G(r) = Ke^{-\frac{r^2}{2\sigma^2}}$$





$\frac{1}{4.8976} \times$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

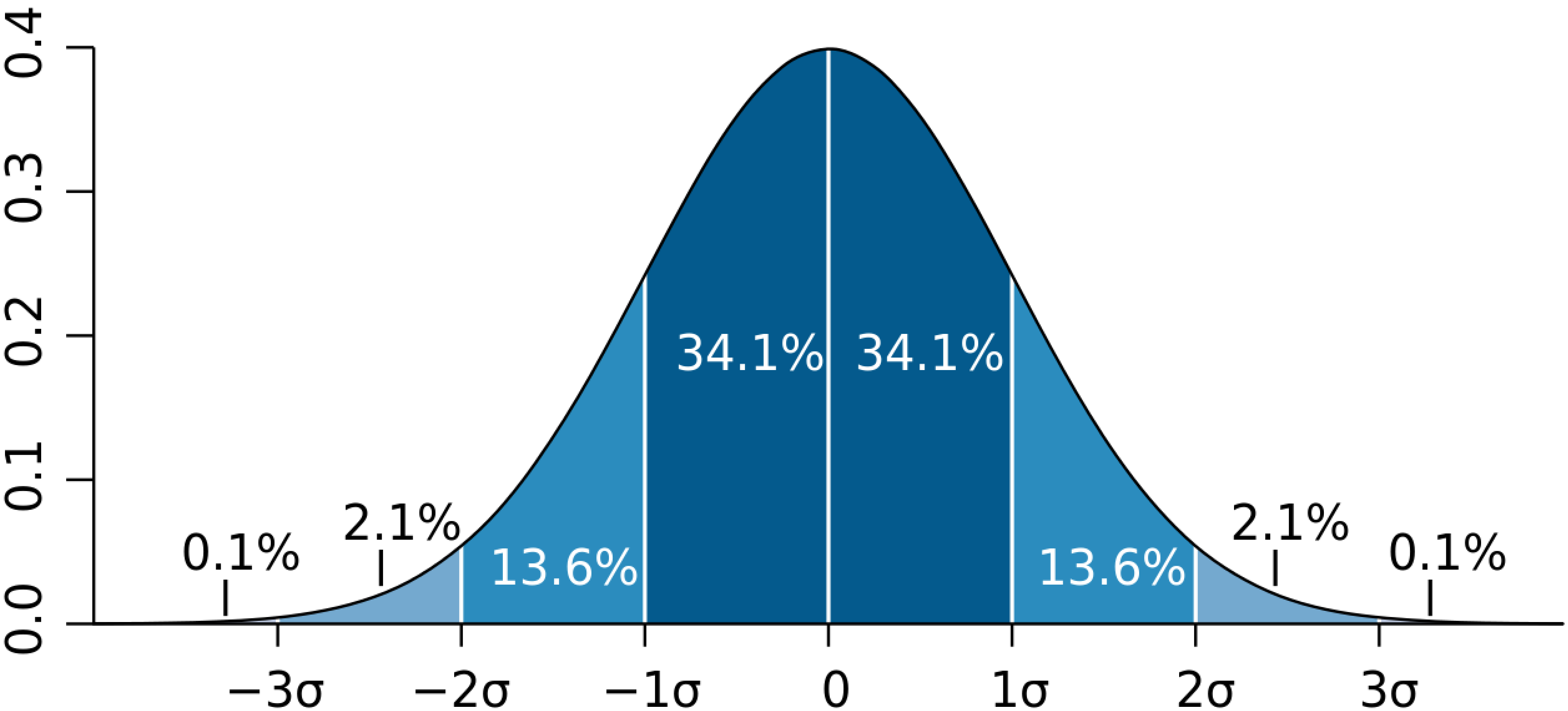
Gaussian (Weighted average) Filter

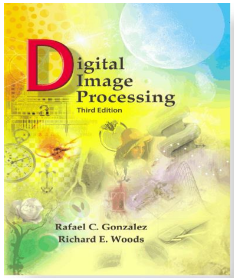
- Separable, circularly-symmetric and isotropic kernel
- Uniform smoothing in all directions
- Product or convolution of two Gaussians is a Gaussian
- Size of a Gaussian kernel: $6\boldsymbol{\sigma} \times 6\boldsymbol{\sigma}$

Image Padding

- Zero padding results black borders
- Mirror padding: Borders contain the image details
- Replicate padding: Borders contain constant intensity

Standard Gaussian Distribution

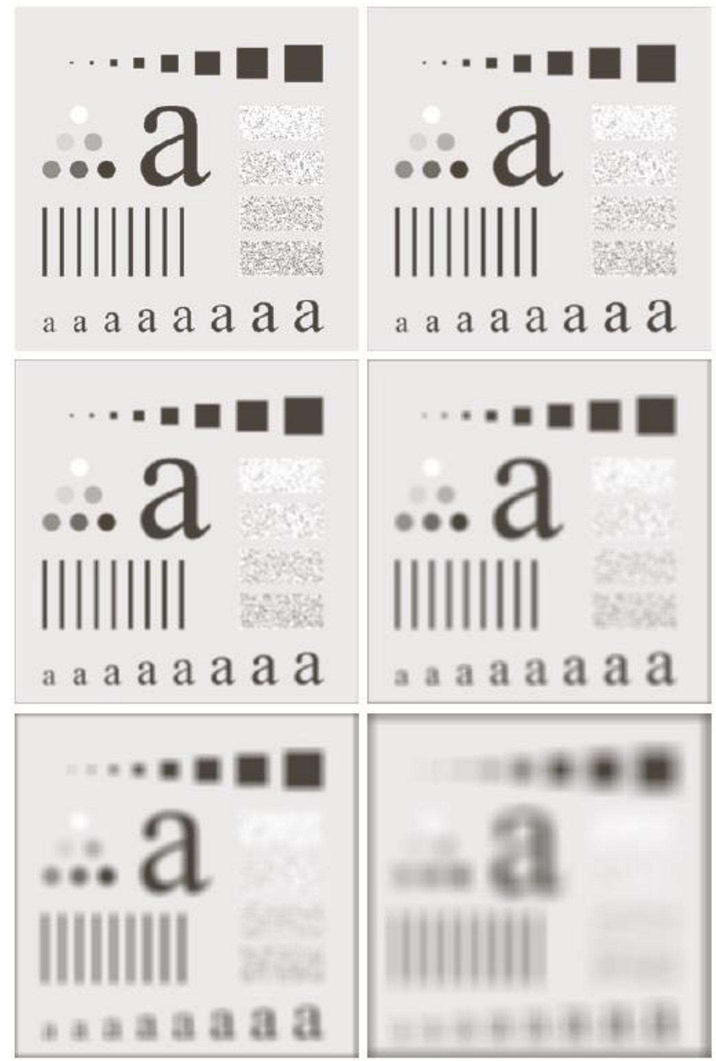




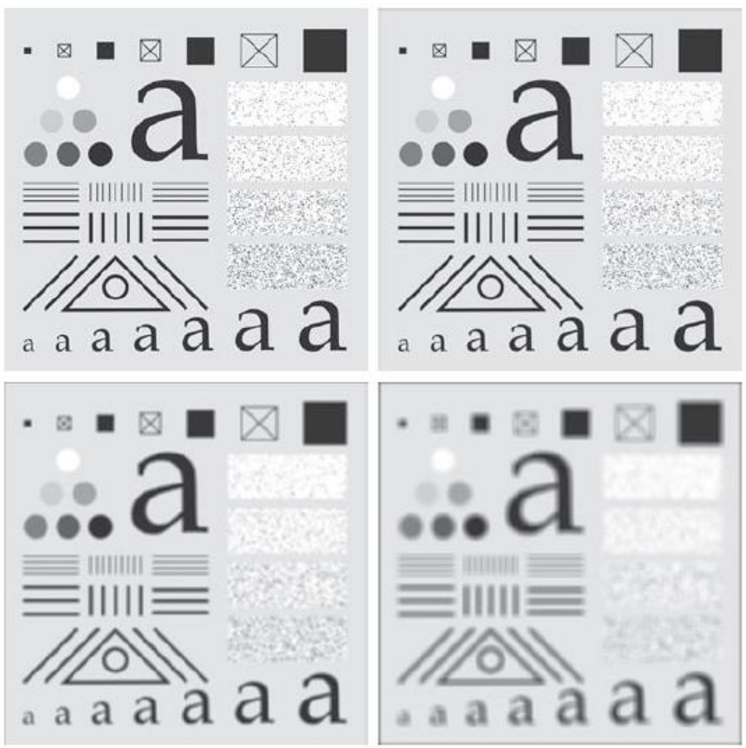
Chapter 3
Intensity Transformations & Spatial Filtering

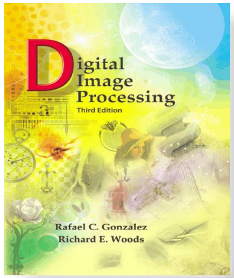
FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart.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

a b
c d
e f

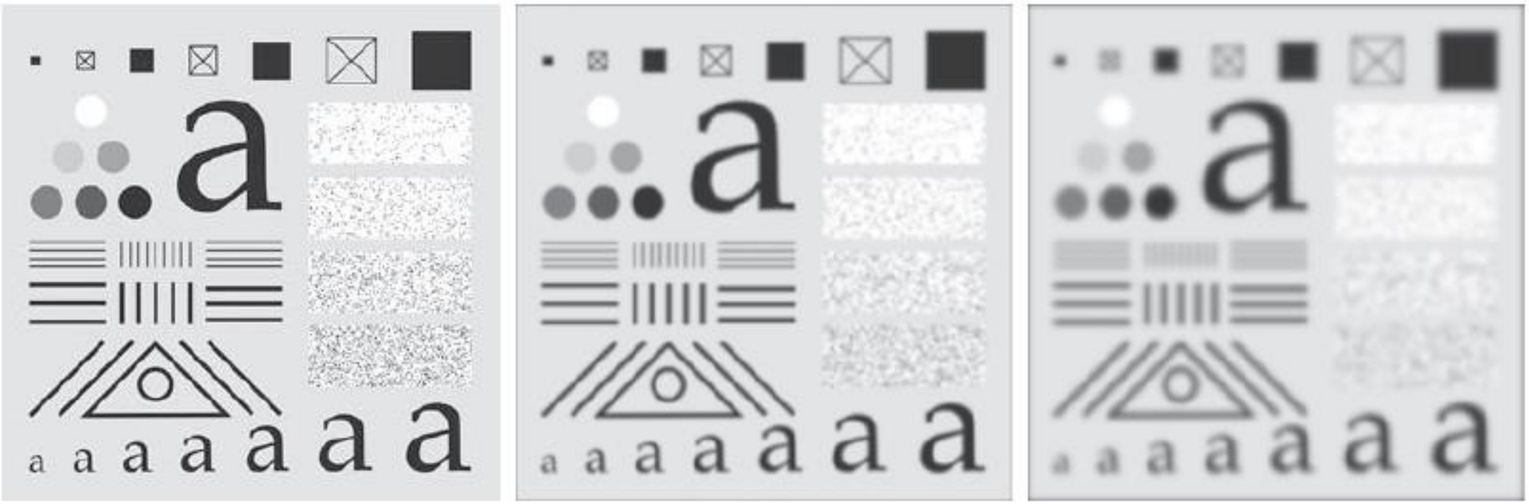


Mask size
3, 11, 21





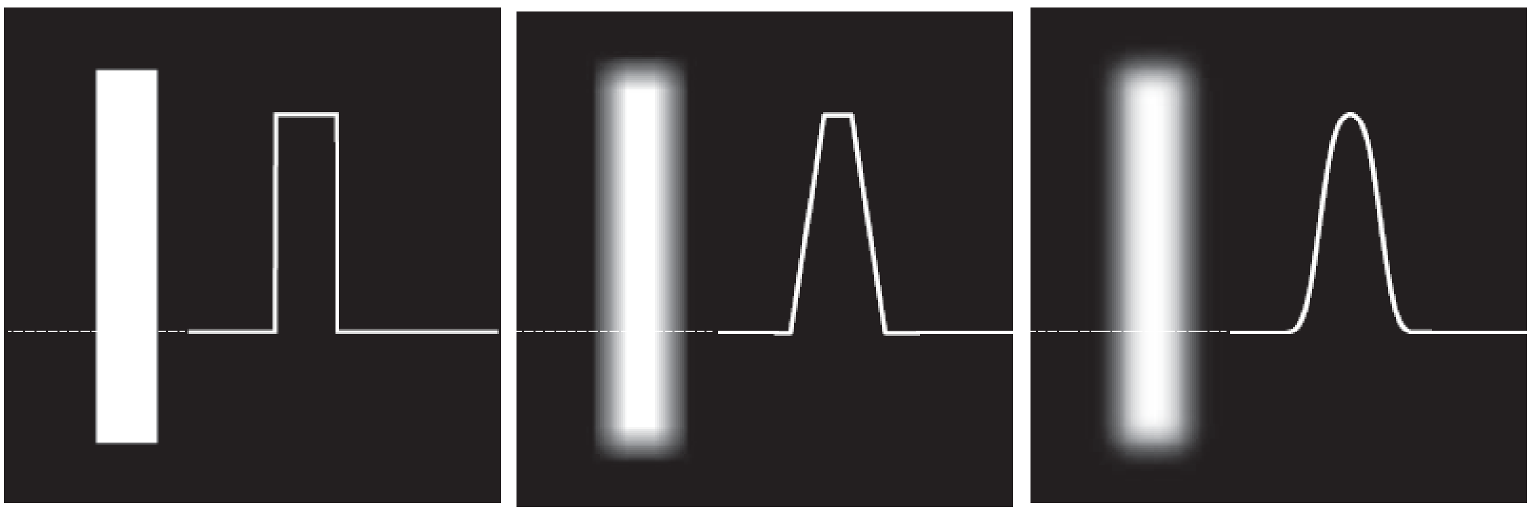
Gaussian Smoothing



Gaussian Mask = 21, 43

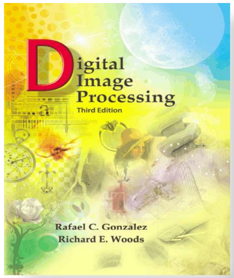
Gaussian Mask = 43, 85

Smoothing using Box and Gaussian Filters

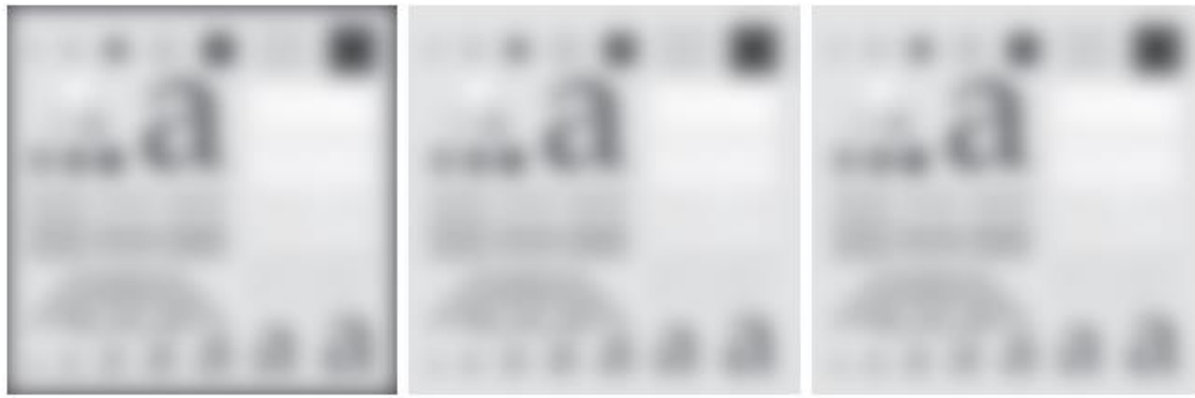


a b c

FIGURE 3.38 (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes 1024×1024 and 768×128 pixels, respectively.



Gaussian Filter O/P for different paddings



a b c

FIGURE 3.39 Result of filtering the test pattern in Fig. 3.36(a) using (a) zero padding, (b) mirror padding, and (c) replicate padding. A Gaussian kernel of size 187×187, with $K = 1$ and $\sigma = 31$ was used in all three cases.

Gaussian Filter O/P w.r.t image and kernel size

a b c

FIGURE 3.40 (a) Test pattern of size 4096×4096 pixels. (b) Result of filtering the test pattern with the same Gaussian kernel used in Fig. 3.39. (c) Result of filtering the pattern using a Gaussian kernel of size 745×745 elements, with $K = 1$ and $\sigma = 124$. Mirror padding was used throughout.

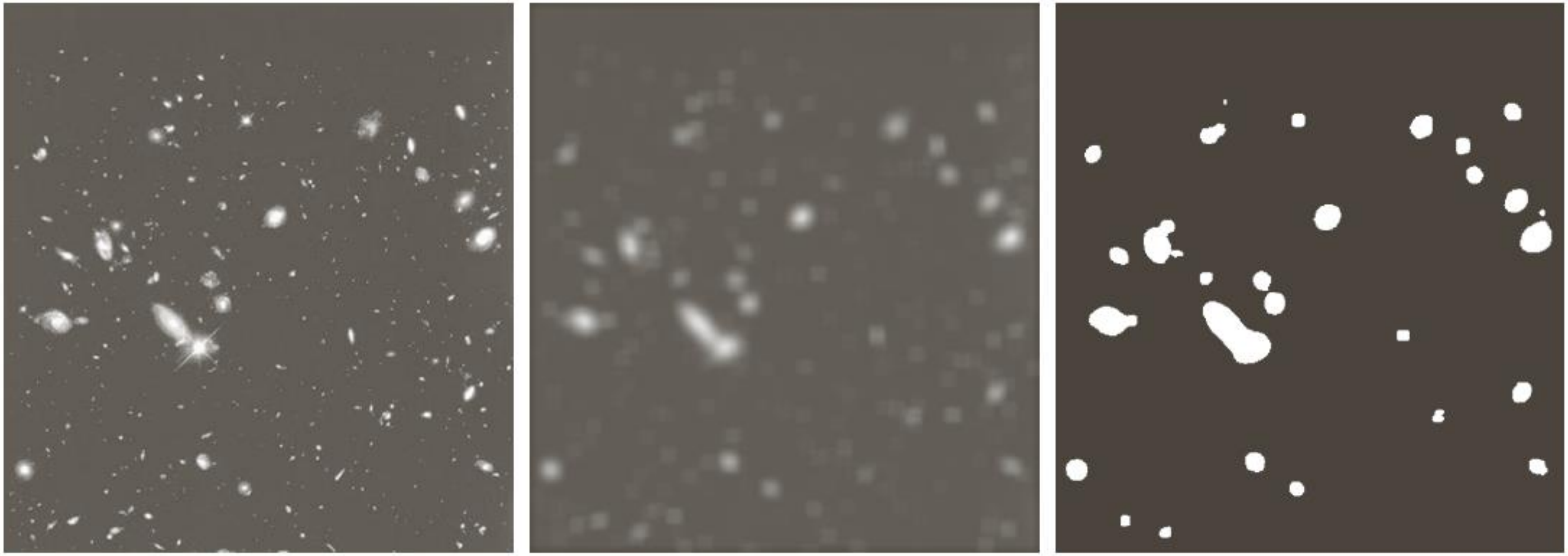
LPF and thresholding for region extraction

a b c

FIGURE 3.41 (a) A 2566 × 2758 Hubble Telescope image of the *Hickson Compact Group*. (b) Result of lowpass filtering with a Gaussian kernel. (c) Result of thresholding the filtered image (intensities were scaled to the range [0, 1]). The Hickson Compact Group contains dwarf galaxies that have come together, setting off thousands of new star clusters. (Original image courtesy of NASA.)

Gaussian Kernel 151X151, Threshold = 40% of Max intensity

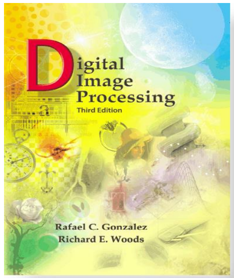
LPF and thresholding for region extraction



a b c

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

Threshold = 0.25*Max Intensity value



Shading estimation using LPF

a b c

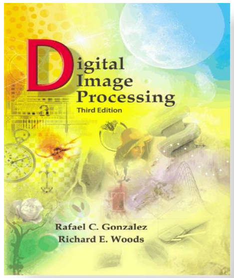
FIGURE 3.42 (a) Image shaded by a shading pattern oriented in the −45° direction. (b) Estimate of the shading patterns obtained using lowpass filtering. (c) Result of dividing (a) by (b). (See Section 9.8 for a morphological approach to shading correction).

Size of square block = 128 × 128; Size of Gaussian kernel = 512 × 512

Product and Convolution of two 1-D Gaussian functions

TABLE 3.6 Mean and standard deviation of the product ($\times$) and convolution ($\star$) of two 1-D Gaussian functions, f and g. These results generalize directly to the product and convolution of more than two 1-D Gaussian functions (see Problem 3.25).

	f	g	$f \times g$	$f \star g$
Mean	m_f	m_g	$m_{f \times g} = \dfrac{m_f \sigma_g^2 + m_g \sigma_f^2}{\sigma_f^2 + \sigma_g^2}$	$m_{f \star g} = m_f + m_g$
Standard deviation	σ_f	σ_g	$\sigma_{f \times g} = \sqrt{\dfrac{\sigma_f^2 \sigma_g^2}{\sigma_f^2 + \sigma_g^2}}$	$\sigma_{f \star g} = \sqrt{\sigma_f^2 + \sigma_g^2}$



- Order Statistics Filters
  - Median Filter (50th percentile)
    - Replace center value by the median of values in the mask
    - Good for Impulse / Salt-and-Pepper Noise
  - Max Filter (100th percentile)
    - To detect brightest points
  - Min Filter (0th percentile)

Chapter 3

Intensity Transformations & Spatial Filtering

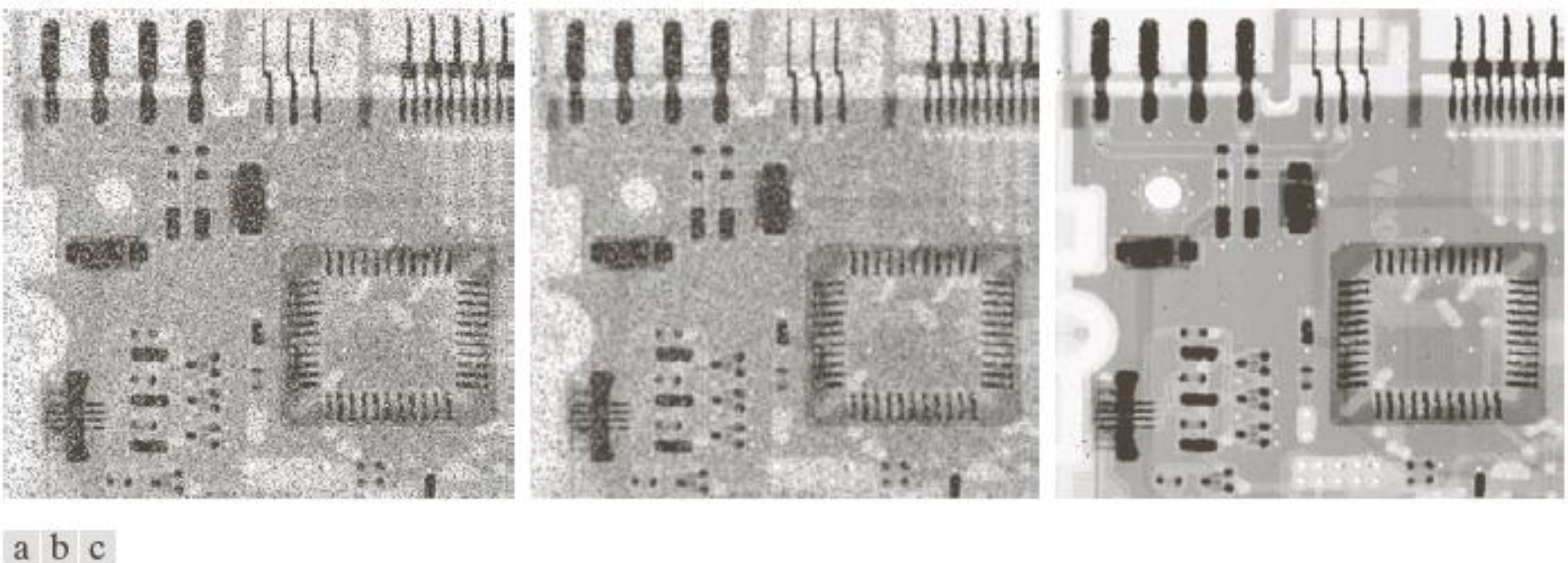


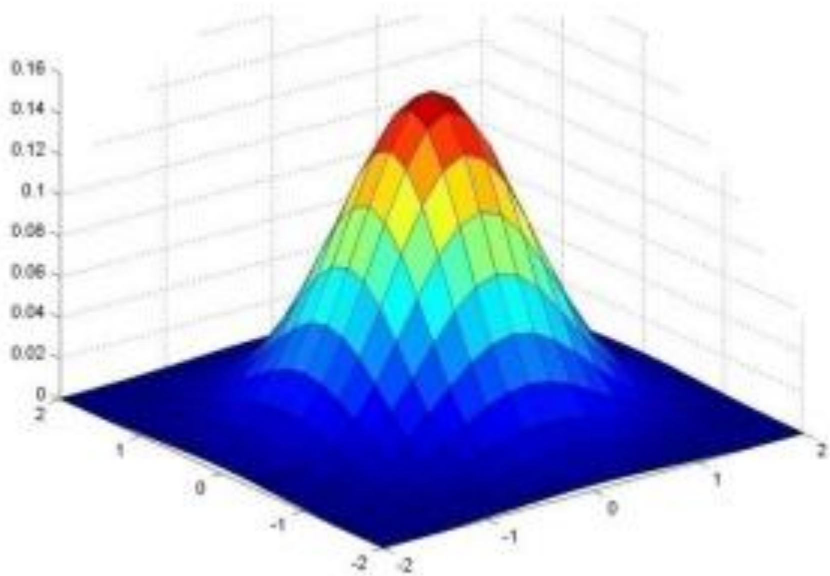
FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

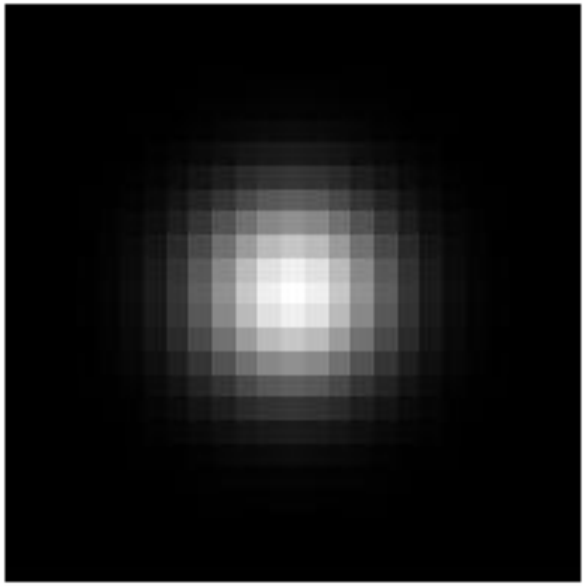
Gaussian Filter

SMOOTHING FILTER

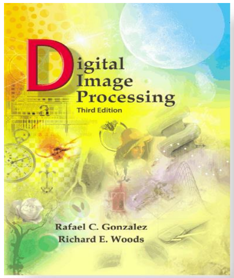
- Gaussian Kernel

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

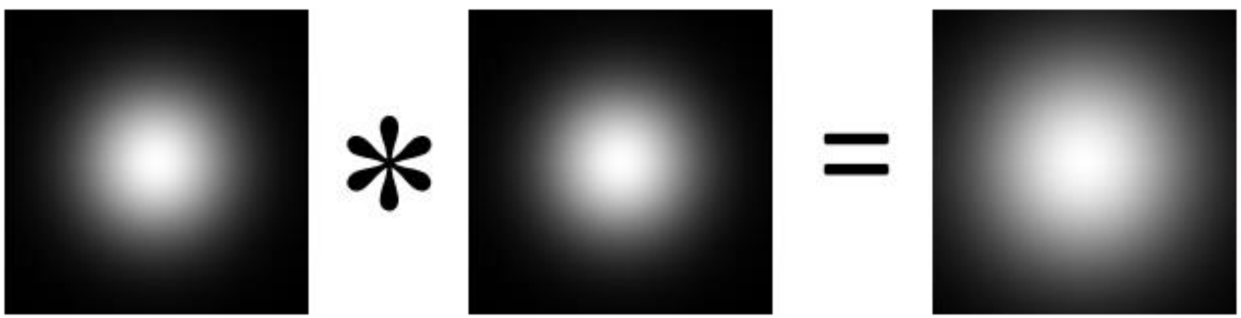




Source: http://www.cs.cornell.edu/courses/cs6670/2011sp/lectures/lec02_filter.pdf



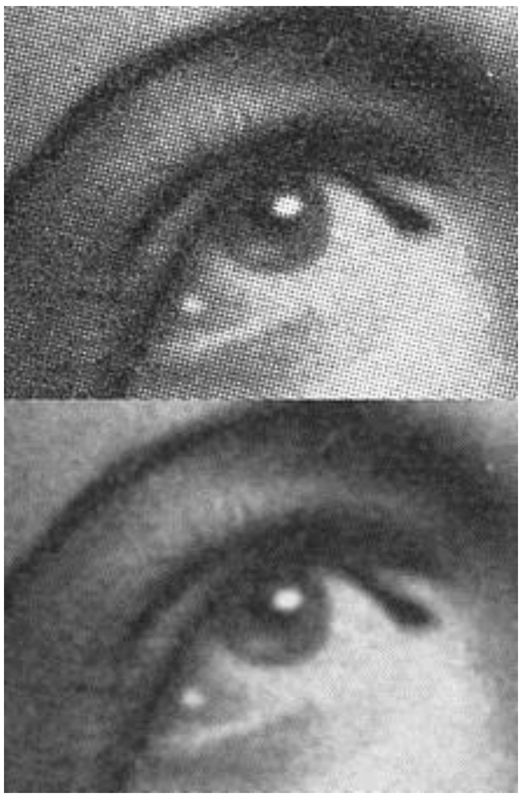
- Gaussian Filter
  - Removes "high-frequency" components from the image (low-pass filter)
  - Convolution with self is another Gaussian



  - Convolving two times with Gaussian kernel of width σ = convolving once with kernel of width $\sigma\sqrt{2}$

Source: http://www.cs.cornell.edu/courses/cs6670/2011sp/lectures/lec02_filter.pdf

- Gaussian Blur



Source: https://en.wikipedia.org/wiki/Gaussian_blur

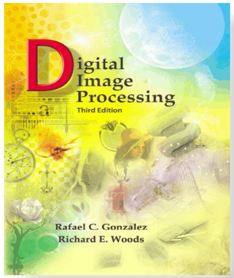
- Gaussian Kernels

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

1	2	1
2	4	2
1	2	1

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



- Gaussian Filters

Original Image

Gaussian filtered image, $\sigma = 2$

Source: https://in.mathworks.com/help/images/ref/imgaussfilt.html

- Gaussian Filters (σ = 3)

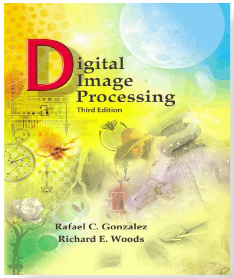
Source: http://www.cse.psu.edu/~rtc12/CSE486/lecture04.pdf

- Box Filter (3 x 3)

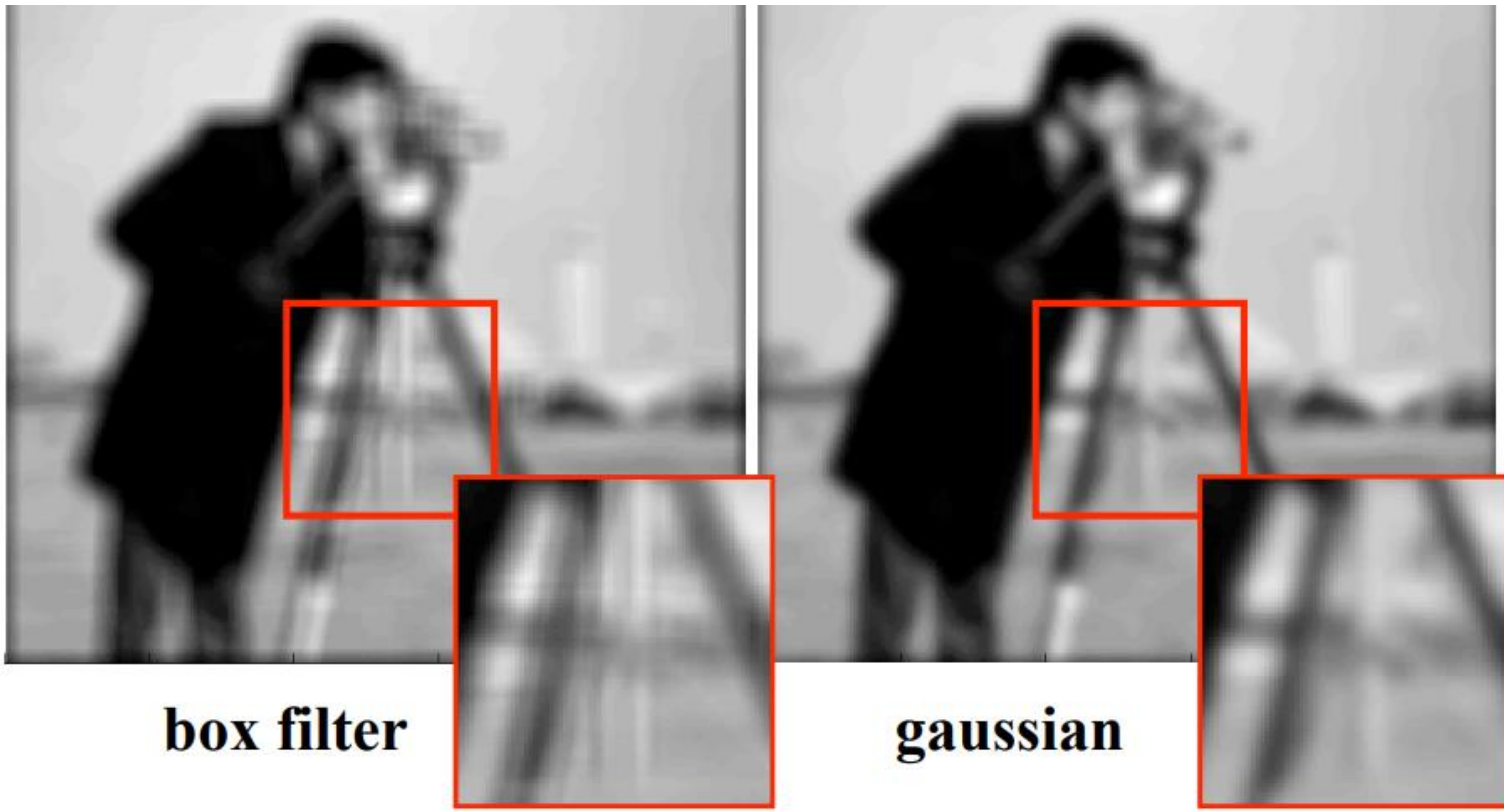
Source: http://www.cse.psu.edu/~rtc12/CSE486/lecture04.pdf

- Box vs. Gaussian Filters

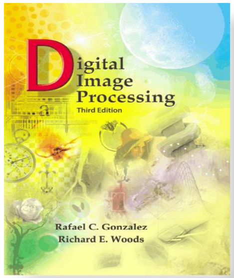
Source: http://www.cse.psu.edu/~rtc12/CSE486/lecture04.pdf



- Gaussian is a true low-pass filter – little high frequency artifacts

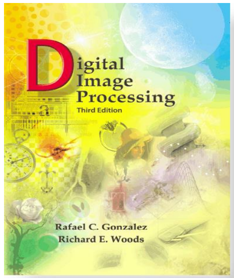


Source: http://www.cse.psu.edu/~rtc12/CSE486/lecture04.pdf

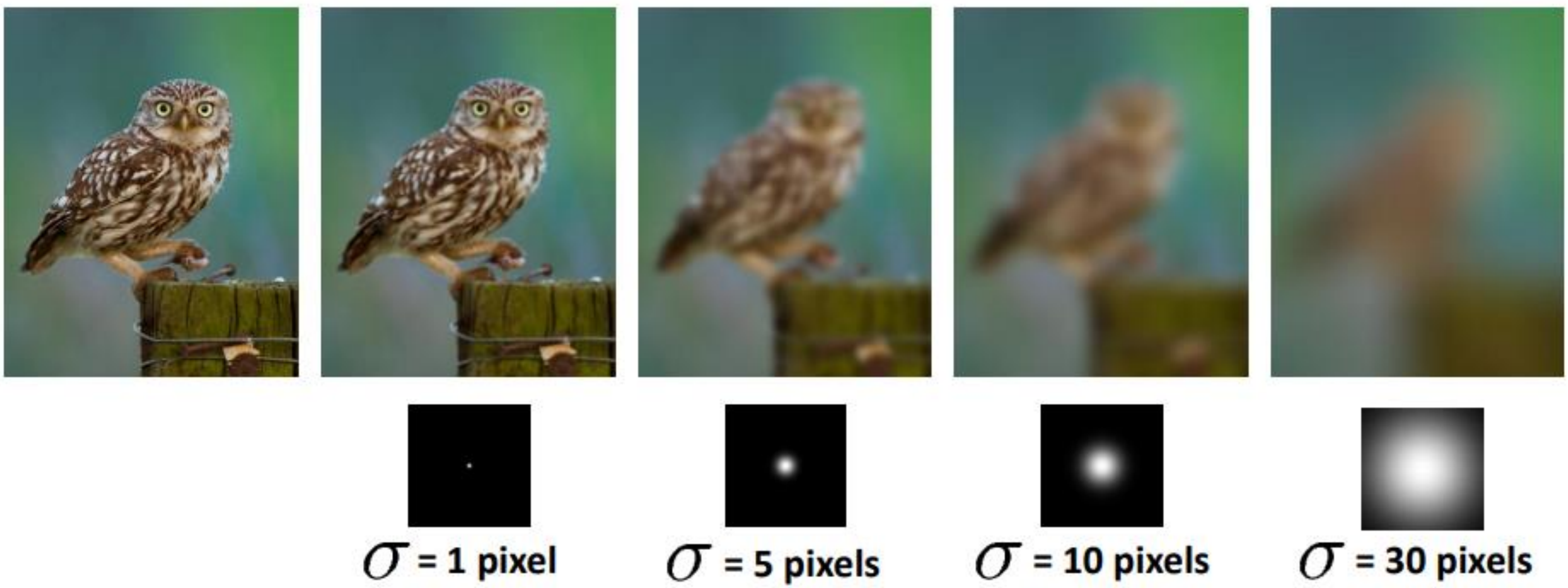


- Gaussian at Different Scales (σ = 1, 3, 10)

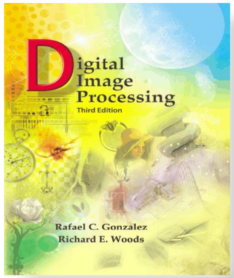
Source: http://www.cse.psu.edu/~rtc12/CSE486/lecture04.pdf



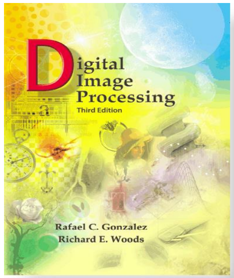
- Gaussian Filters



Source: http://www.cs.cornell.edu/courses/cs6670/2011sp/lectures/lec02_filter.pdf



SHARPENING FILTER



- Sharpening (High Pass Filter) Spatial Filters
  - First Derivative

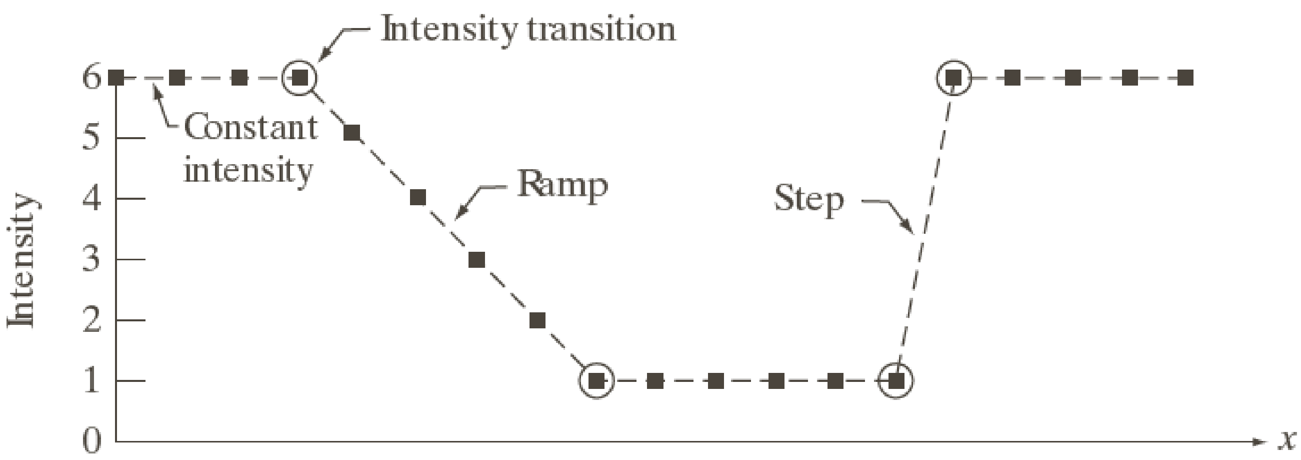
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

  - Second Derivative

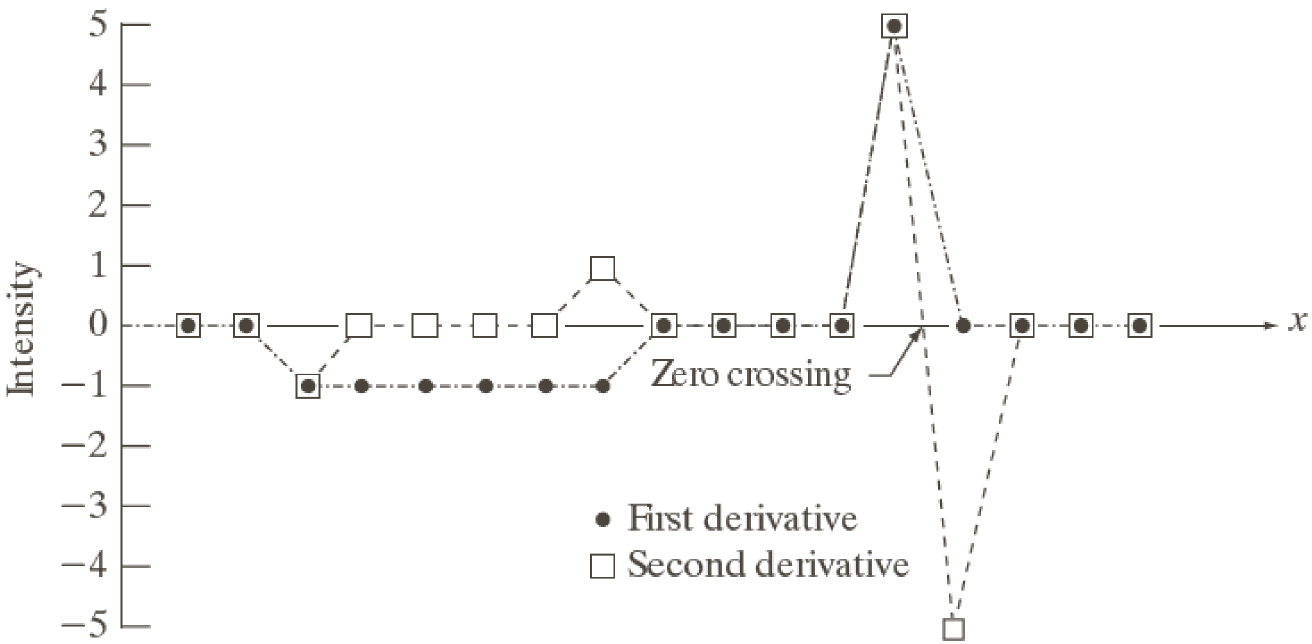
$$\partial^2 f / \partial x^2 = f'' = f'(x+1) - f'(x)$$
$$= f(x+2) - f(x+1) - f(x+1) + f(x)$$
$$\partial^2 f / \partial x^2 = f(x+1) + f(x-1) - 2f(x)$$

Properties of Derivatives

- First Derivative
  - Must be zero in the areas of constant intensity
  - Must be non-zero at the onset of an intensity ramp or step
  - Must be *non-zero* along ramps
- Second Derivative
  - Must be zero in the areas of constant intensity
  - Must be non-zero at the onset *and end* of a ramp or step
  - Must be *zero* along ramps of *constant slope*



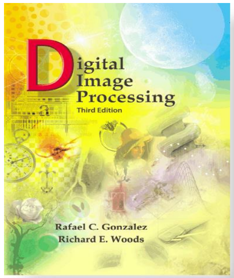
Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	



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.

LAPLACIAN



- Laplacian
  - Isotropic
  - Rotation Invariant

0	1	0
1	−4	1
0	1	0

$$\nabla^2 f = \partial^2 f / \partial x^2 + \partial^2 f / \partial y^2$$

$$\partial^2 f / \partial x^2 = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\partial^2 f / \partial y^2 = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

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

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

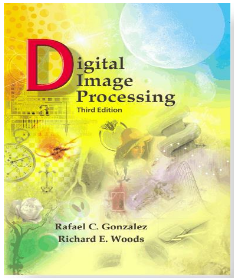
−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.37 (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.

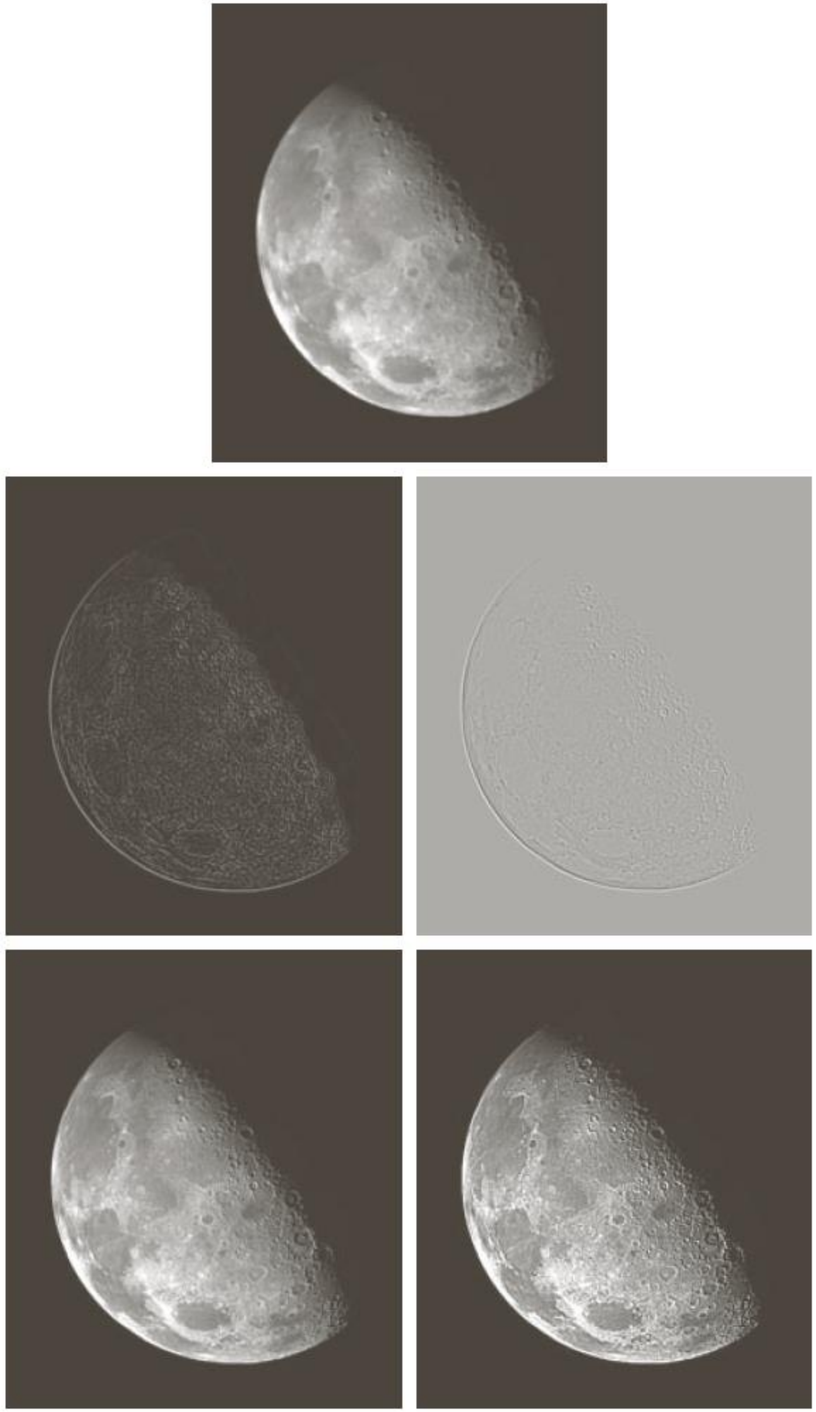
- Sharpening Filters

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



0	1	0
1	−4	1
0	1	0

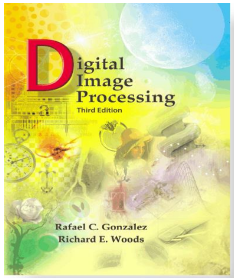
1	1	1
1	−8	1
1	1	1



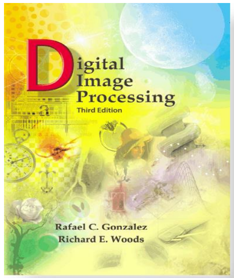
a
b c
d e

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

UNSHARP MASKING & HIGHBOOST FILTERING



- Unsharp Masking & Highboost Filtering
  - Blur (unsharp) the original image
  - Subtract blurred image from original to get a mask
  - Add the mask to the original



- Unsharp Masking & Highboost Filtering

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

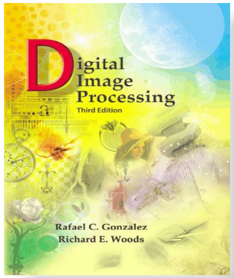
$$g(x,y) = f(x,y) + k * g_{mask}(x,y)$$

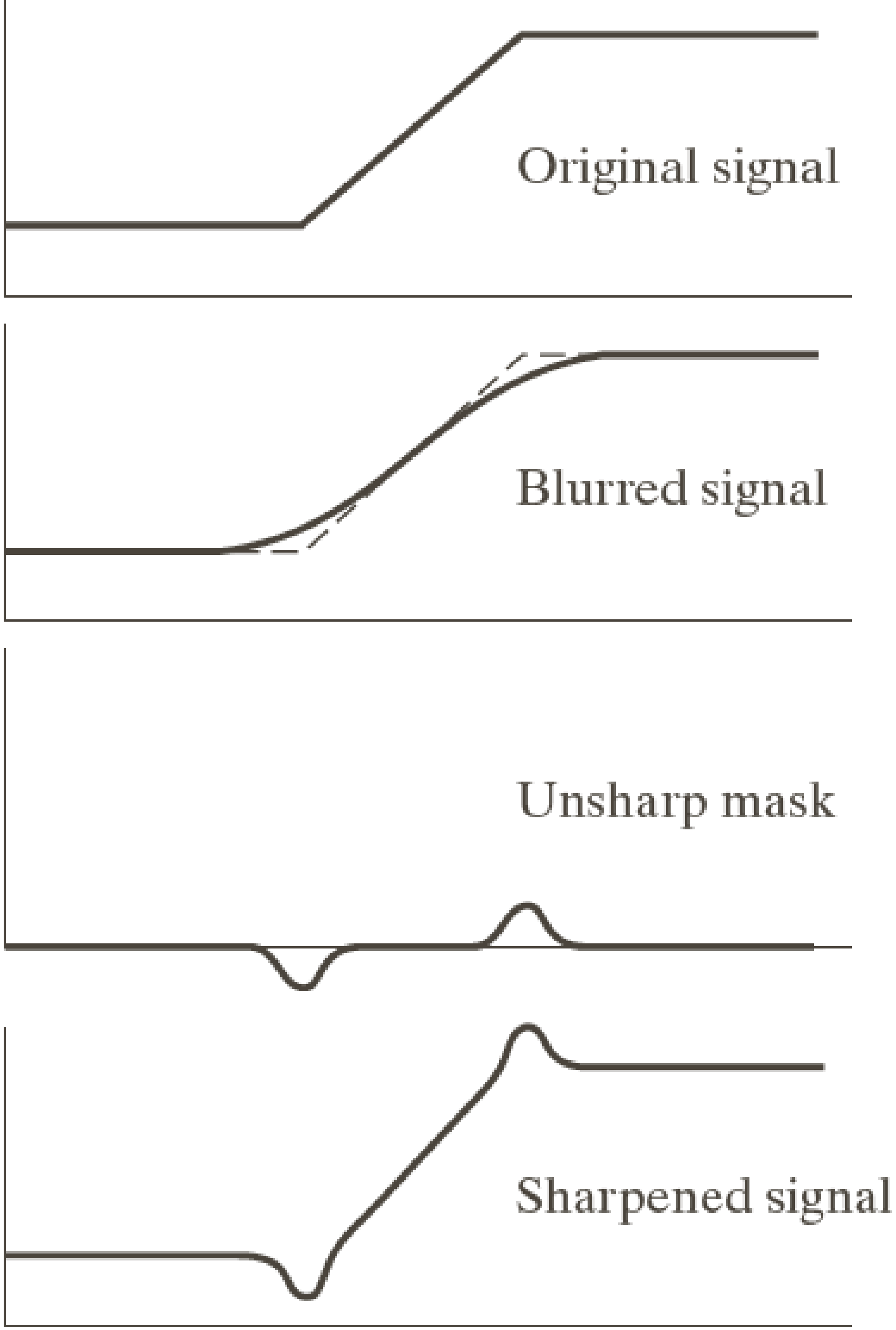
$$\bar{f}(x,y): \text{Blurred } f(x,y)$$

$$k = 1: \text{Unsharp Masking}$$

$$k > 1: \text{Highboost Filtering}$$

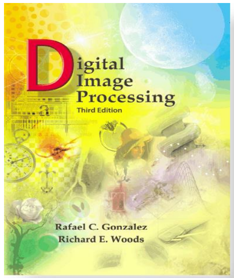
$$k < 1: \text{De-emphasized Unsharp Masking}$$





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

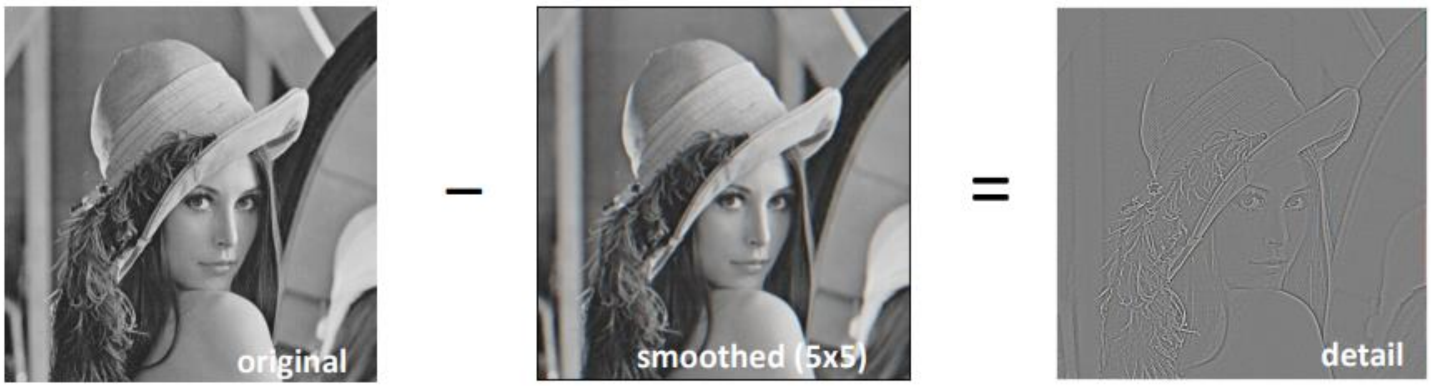
FIGURE 3.40
(a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

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

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

Source: https://en.wikipedia.org/wiki/Unsharp_masking

- What does blurring take away?



- Let us add it back:

Source: http://www.cs.cornell.edu/courses/cs6670/2011sp/lectures/lec02_filter.pdf

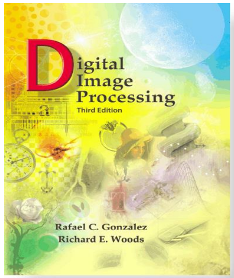
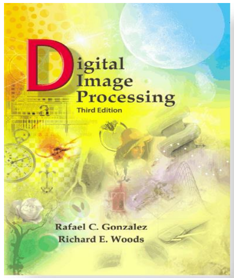


IMAGE GRADIENTS

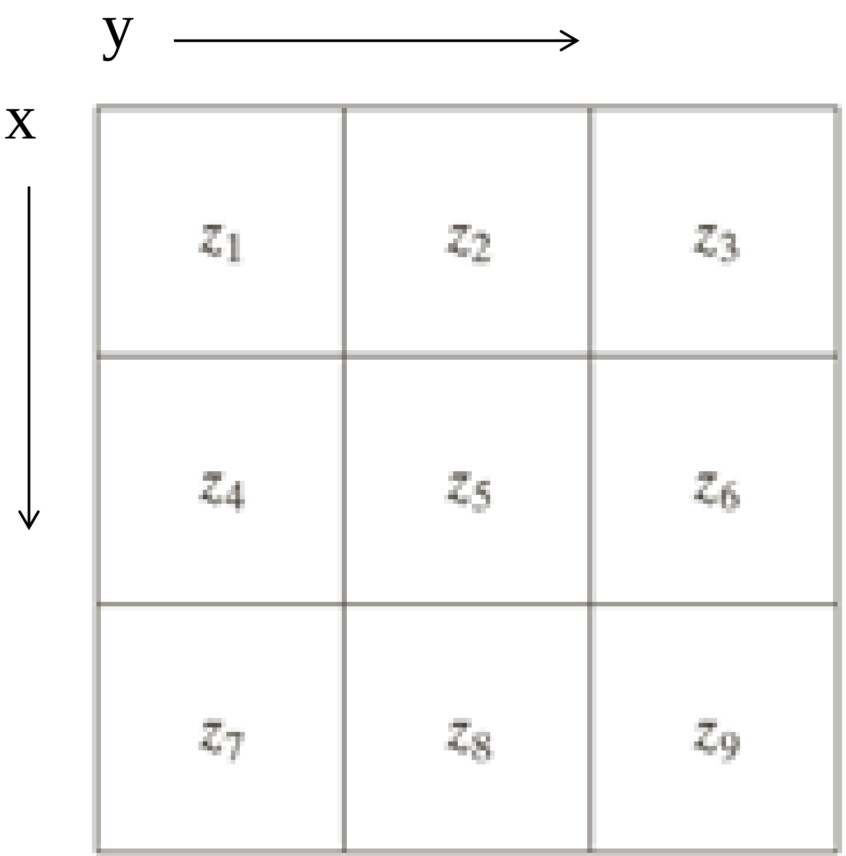


First Order Derivatives: Gradient Magnitude

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

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

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

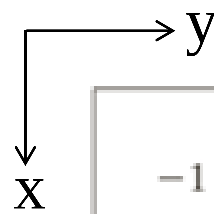


- Gradient Mask
  - Simple Approximation

$$g_x = z_8 - z_5$$

$$g_y = z_6 - z_5$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9



−1	0
0	1

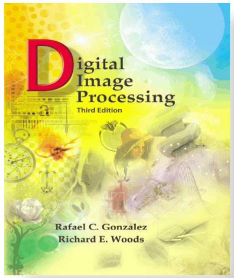
0	−1
1	0

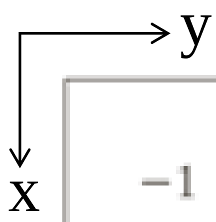
$$g_x = z_9 - z_5, g_y = z_8 - z_6$$

Gradient Mask:
Robert's Operator

$$M(x,y) = \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2}$$

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



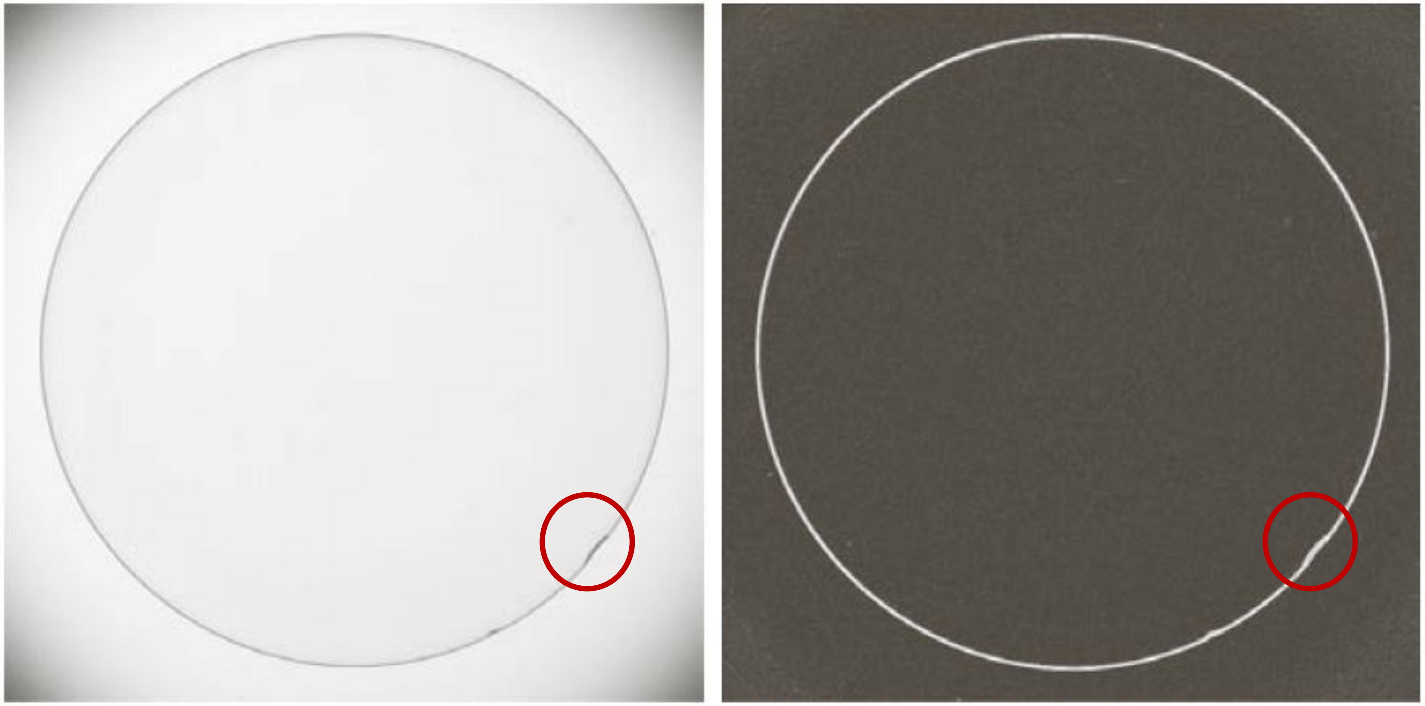


Gradient Mask: Sobel's Operator

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

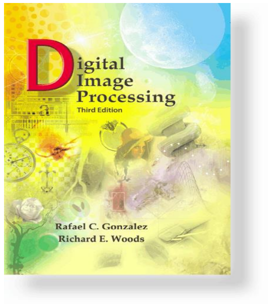
$$M(x,y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$



a b

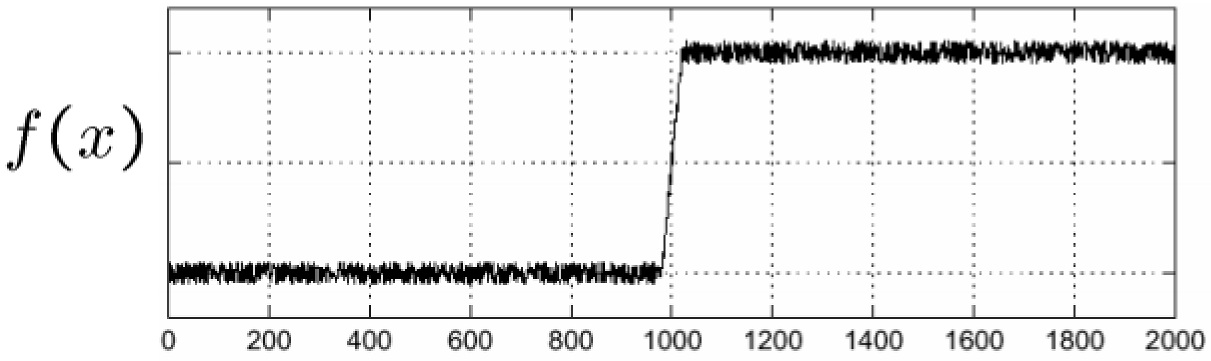
FIGURE 3.42 (a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Pete Sites, Perceptics Corporation.)

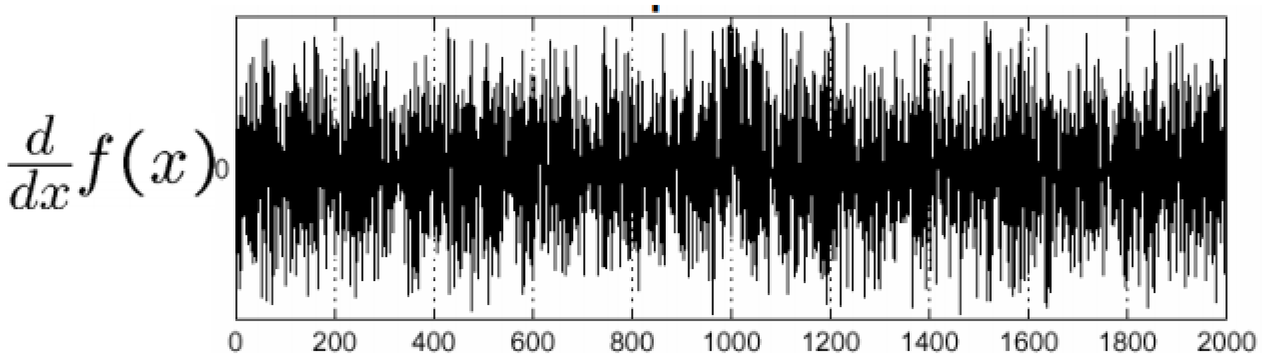
Sobel: Example Application



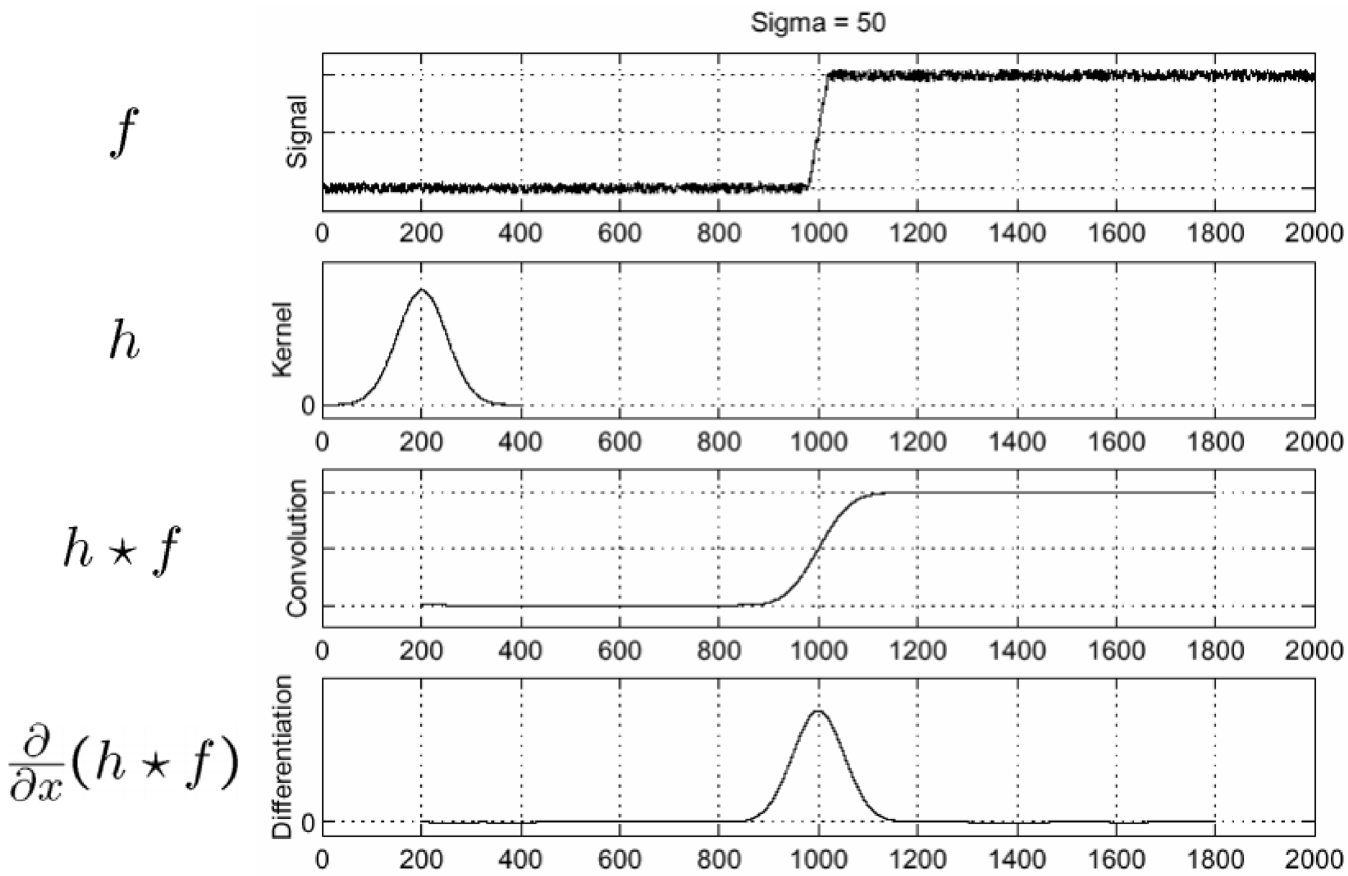
EDGE DETECTION

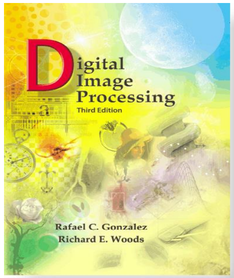
- Where is the edge?





- Smooth first



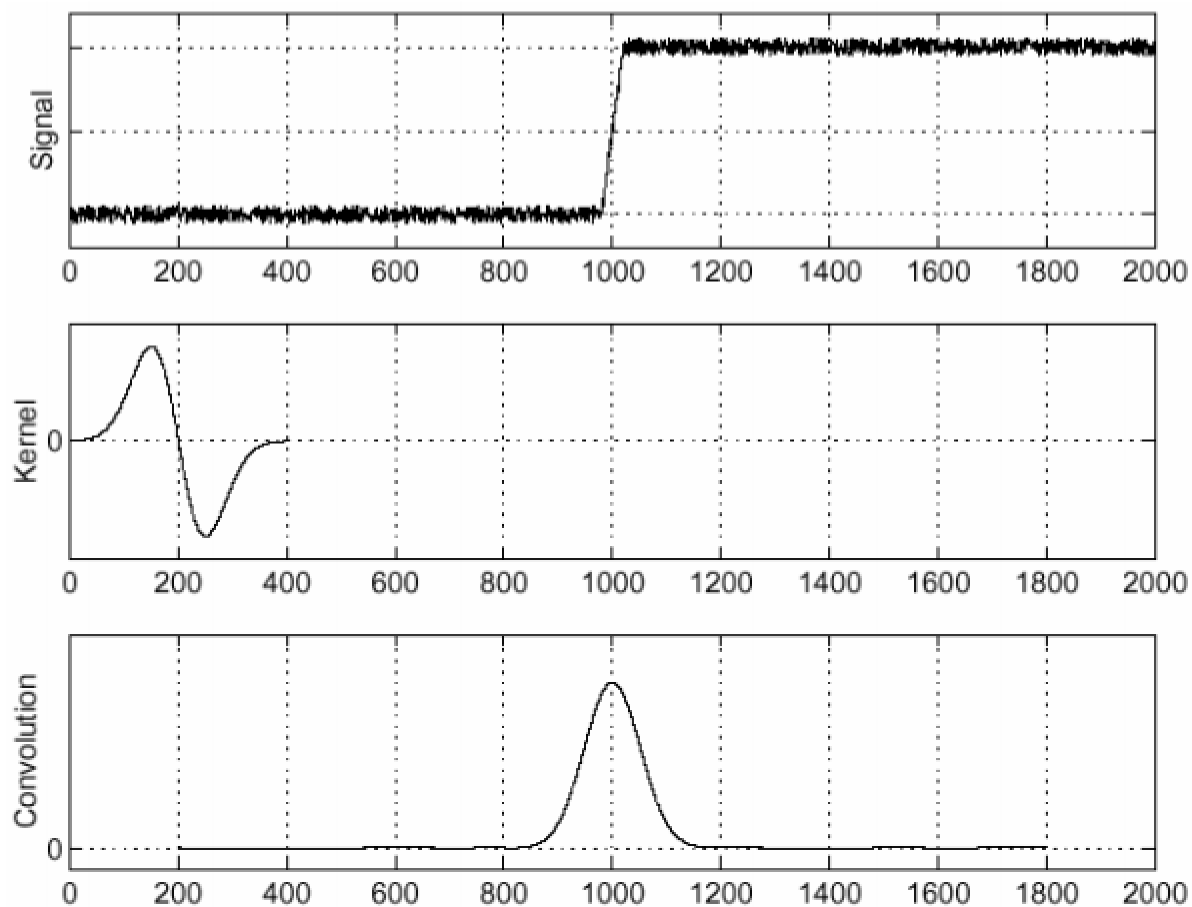


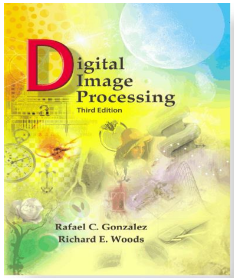
- Short-cut Computation

$$f$$

$$\frac{\partial}{\partial x}h$$

$$(\frac{\partial}{\partial x}h) \star f$$





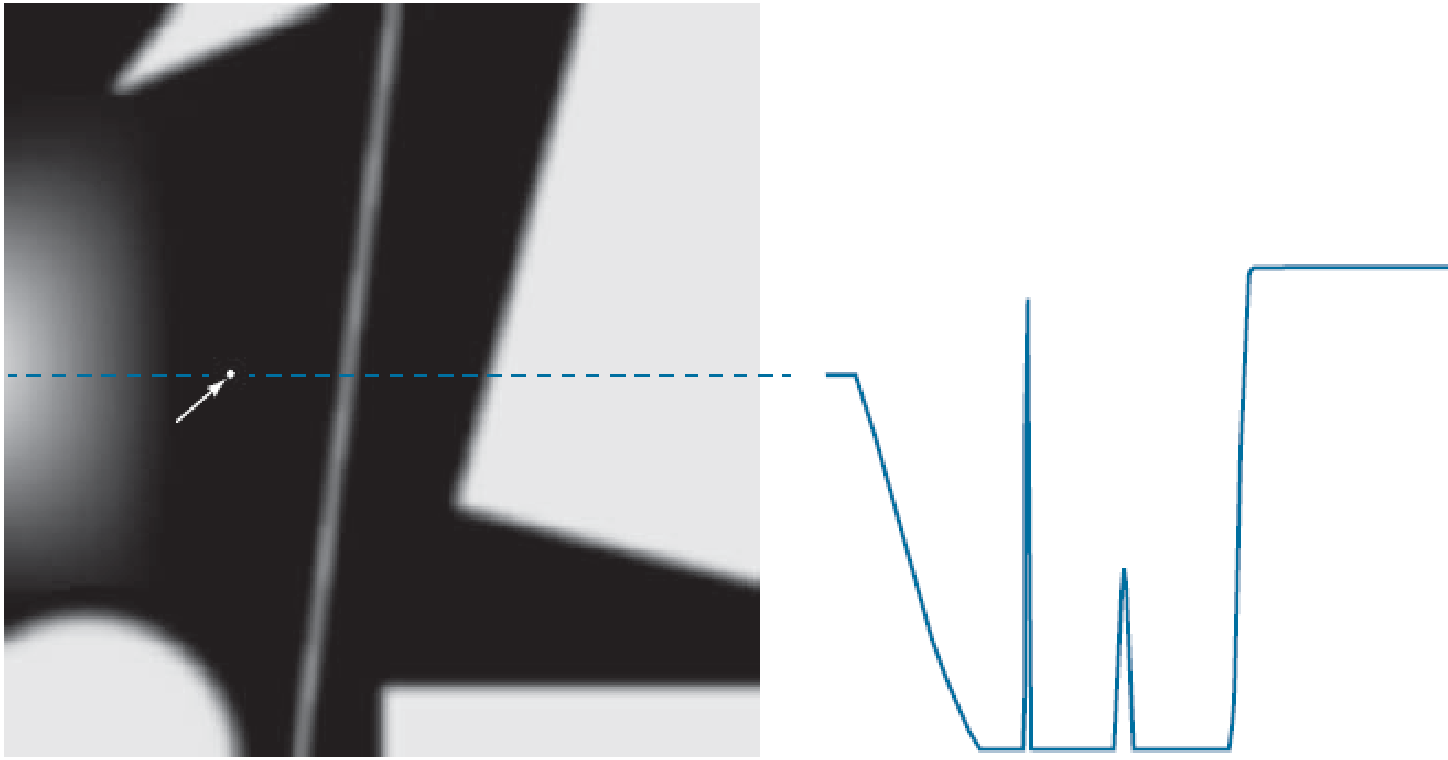
Approximations to Derivatives of Digital Functions

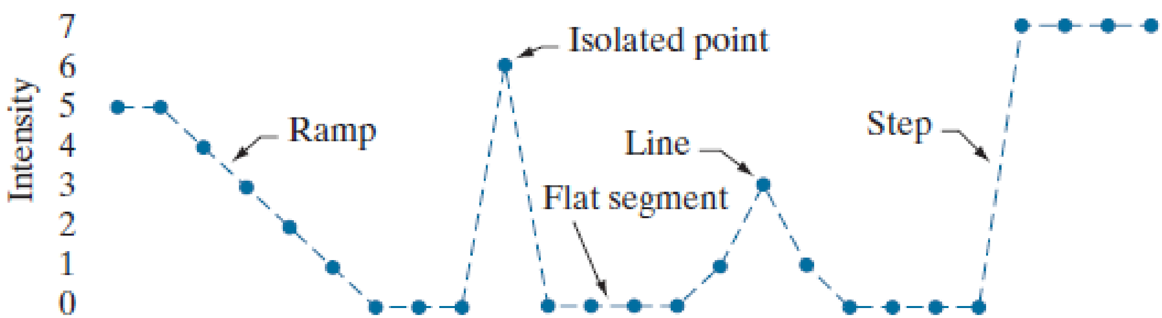
$$f(x+\Delta x) = f(x) + \Delta x \frac{\partial f(x)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \cdots$$
$$= \sum_{n=0}^{\infty} \frac{(\Delta x)^n}{n!}\frac{\partial^n f(x)}{\partial x^n}$$

$$f(x+1) = f(x) + \frac{\partial f(x)}{\partial x} + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \frac{1}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \cdots$$
$$= \sum_{n=0}^{\infty} \frac{1}{n!}\frac{\partial^n f(x)}{\partial x^n}$$

$$f(x-1) = f(x) - \frac{\partial f(x)}{\partial x} + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} - \frac{1}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \cdots$$
$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!}\frac{\partial^n f(x)}{\partial x^n}$$

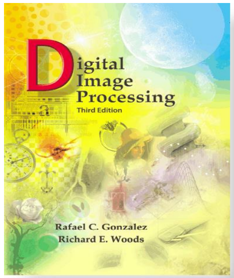
	$f(x+2)$	$f(x+1)$	$f(x)$	$f(x-1)$	$f(x-2)$
$2f'(x)$		1	0	−1	
$f''(x)$		1	−2	1	
$2f'''(x)$	1	−2	0	2	−1
$f''''(x)$	1	−4	6	−4	1





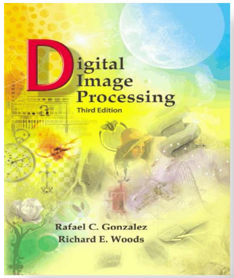
Intensity values	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	·	·
First derivative		–1	–1	–1	–1	–1	0	0	6	–6	0	0	0	1	2	–2	–1	0	0	0	7	0	0	0			
Second derivative		–1	0	0	0	0	1	0	6	–12	6	0	0	1	1	–4	1	1	0	0	7	–7	0	0			

- Isolated Point
- Flat Segments
- Edges
  - Step
  - Ramp
  - Roof (line)



Summary of 1st & 2nd derivative responses:

- (1) First-order derivatives generally produce thicker edges.
- (2) Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise.
- (3) Second-order derivatives produce a double-edge response at ramp and step transitions in intensity.
- (4) The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light.

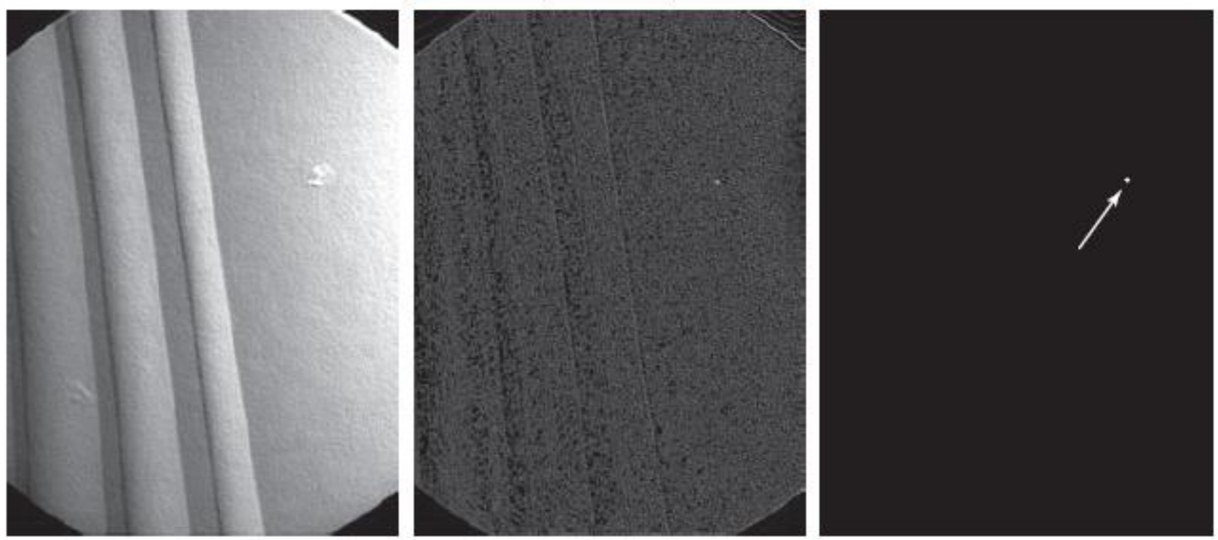


Detection of Isolated Points

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

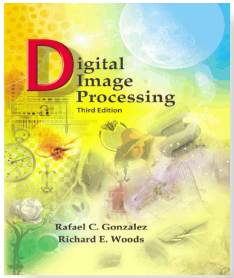
$$g(x,y) = \begin{cases} 1 & \text{if } |Z(x,y)| > T \\ 0 & \text{otherwise} \end{cases}$$

1	1	1
1	−8	1
1	1	1



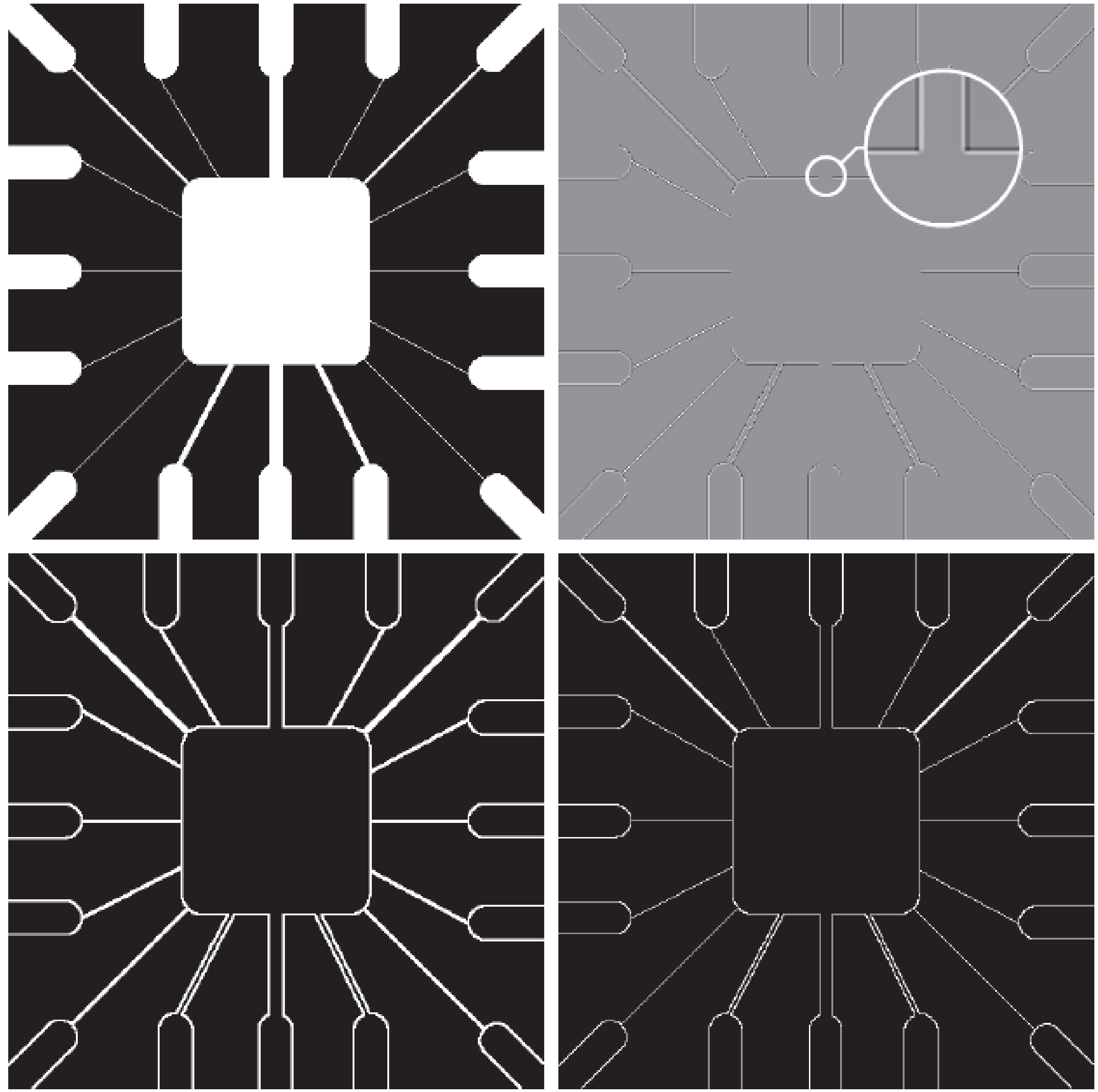
a
b c d

FIGURE 10.4 (a) Laplacian kernel used for point detection. (b) X-ray image of a turbine blade with a porosity manifested by a single black pixel. (c) Result of convolving the kernel with the image. (d) Result of using Eq. (10-15) was a single point (shown enlarged at the tip of the arrow). (Original image courtesy of X-TEK Systems, Ltd.)



a b
c d

FIGURE 10.5
(a) Original image.
(b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian.
(c) Absolute value of the Laplacian.
(d) Positive values of the Laplacian.



Line detection using Laplacian operator

Line detection in specified directions

Horizontal

−1	−1	−1
2	2	2
−1	−1	−1

+45°

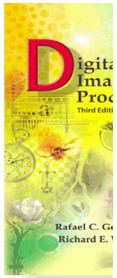
2	−1	−1
−1	2	−1
−1	−1	2

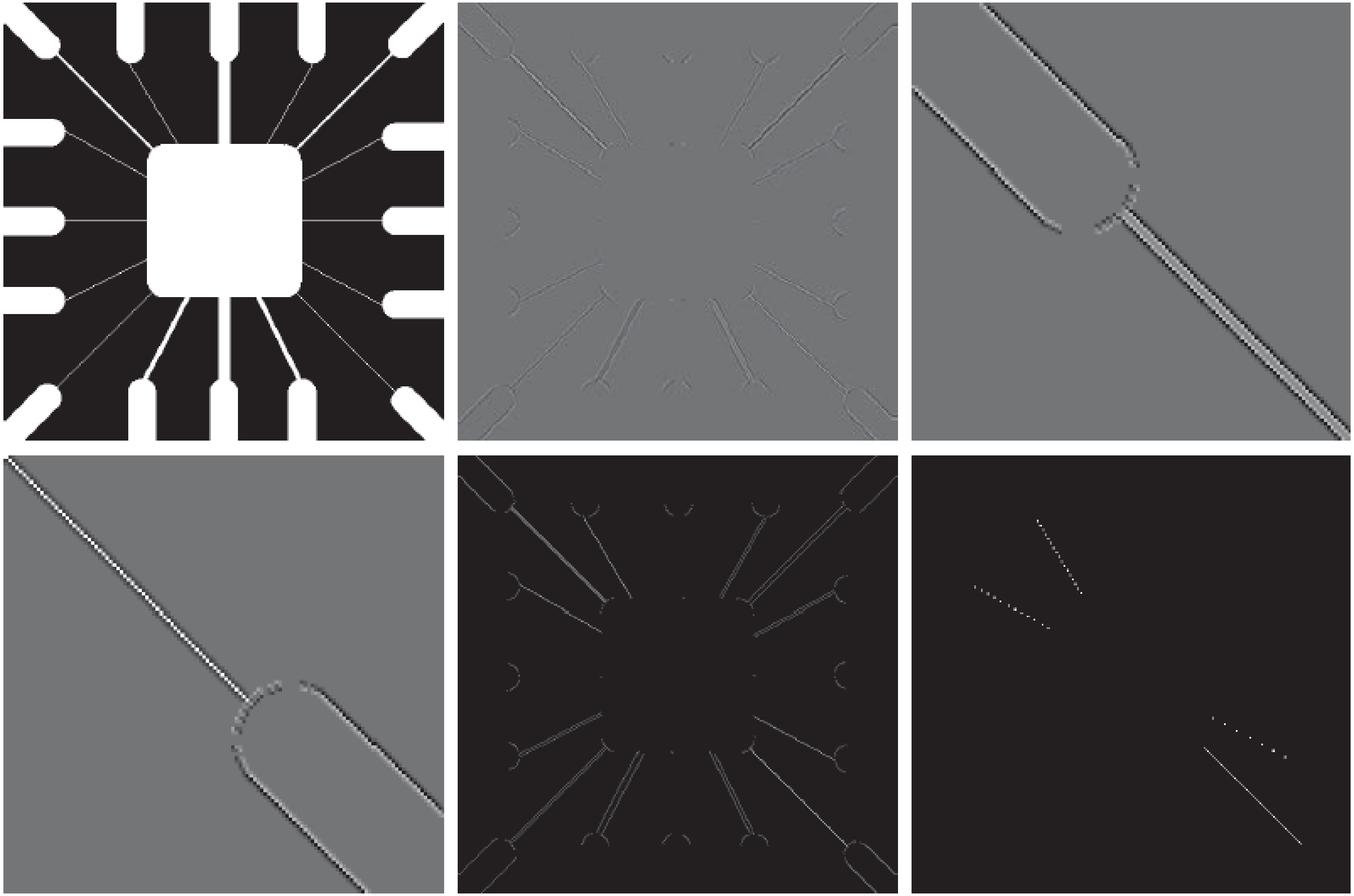
Vertical

−1	2	−1
−1	2	−1
−1	2	−1

−45°

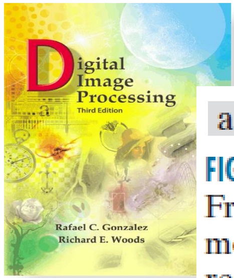
−1	−1	2
−1	2	−1
2	−1	−1



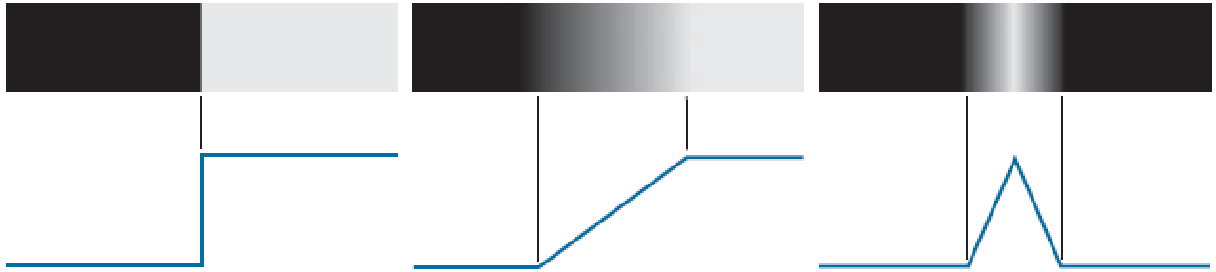


a	b	c
d	e	f

FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the +45° line detector kernel in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g > T$, where g is the image in (e) and $T = 254$ (the maximum pixel value in the image minus 1). (The points in (f) were enlarged to make them easier to see.)



Edge Models



a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

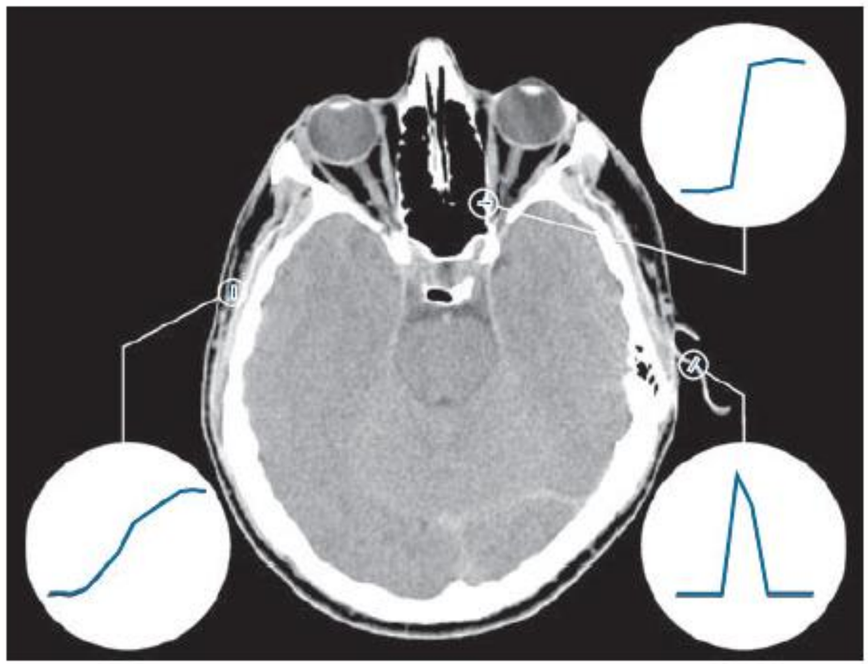
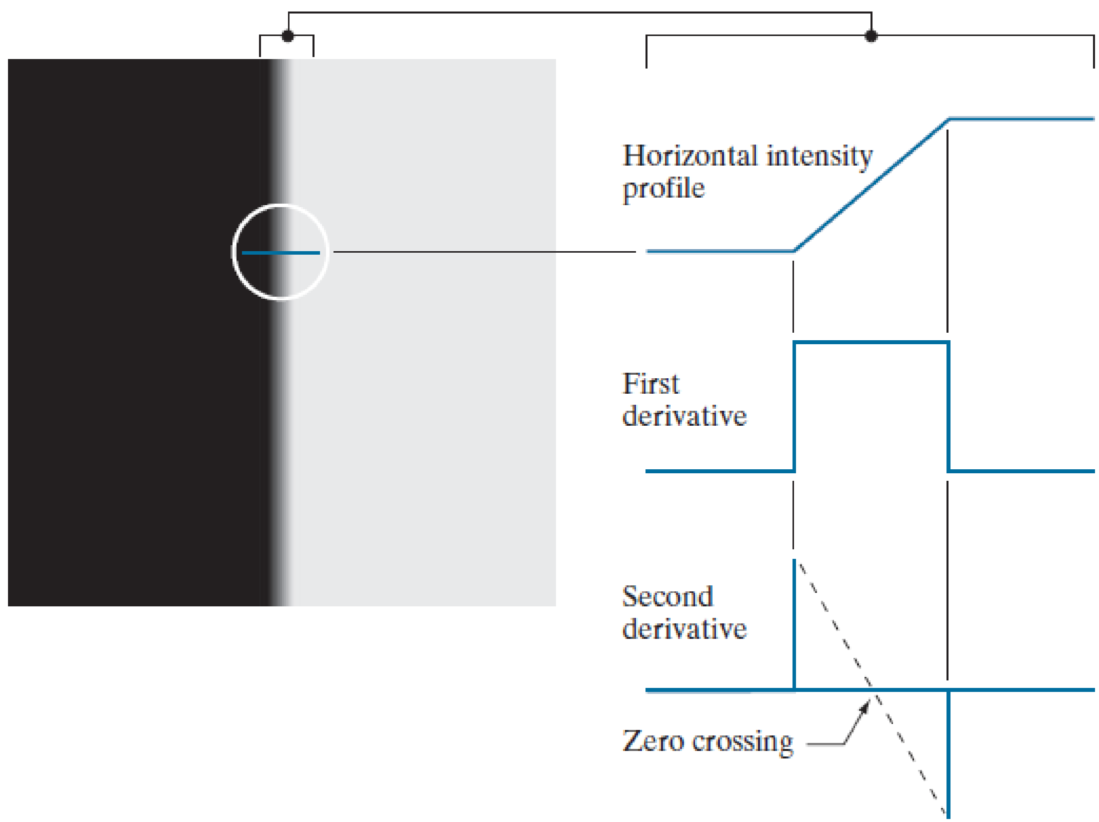


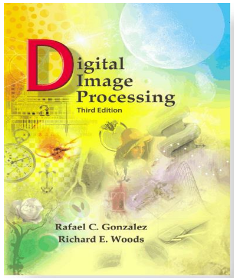
FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas enclosed by the small circles. The ramp and step profiles span 9 pixels and 2 pixels, respectively. The base of the roof edge is 3 pixels. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Modeling of Horizontal Edge

a b

FIGURE 10.10 (a) Two regions of constant intensity separated by an ideal ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, and its first and second derivatives.





Effect of noise on 1st and 2nd Derivatives

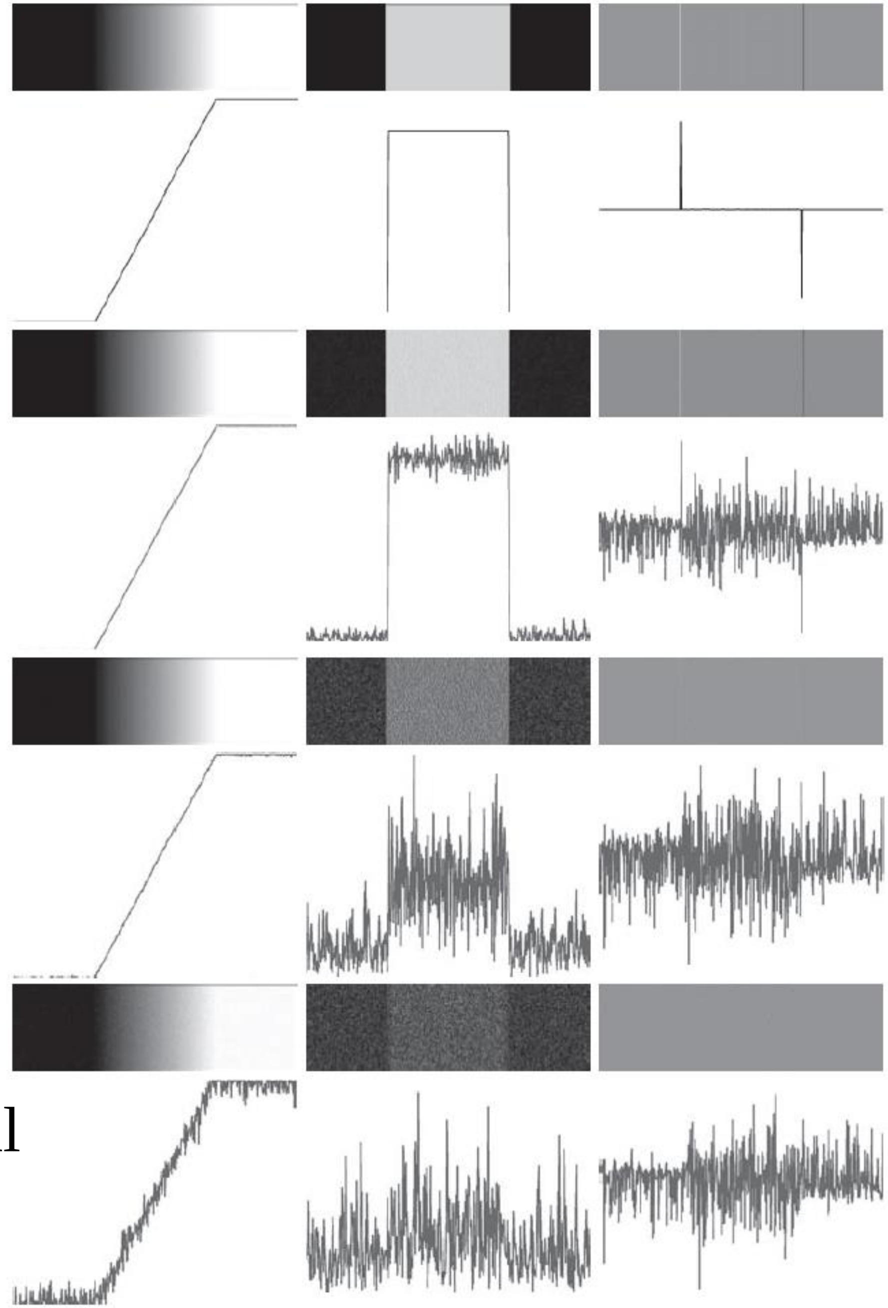
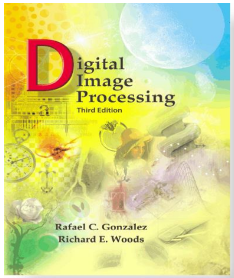


FIGURE 10.11 First column: 8-bit images with values in the range [0,255], and intensity profiles of a ramp edge corrupted by Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

Steps in Edge Detection

- Image smoothing for noise removal
- Detection of edge points
- Edge localization



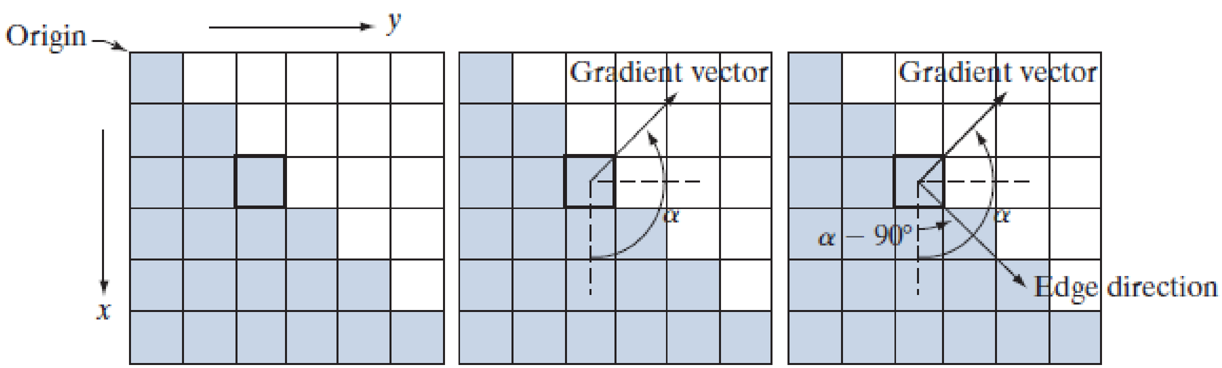
Edge Detection

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

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

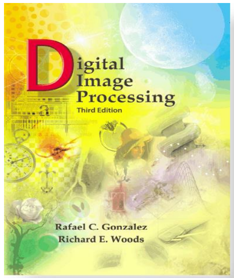
$$M(x,y) \approx |g_x| + |g_y|$$

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y(x,y)}{g_x(x,y)}\right]$$



a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge direction is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square represents one pixel. (Recall from Fig. 2.19 that the origin of our coordinate system is at the top, left.)



Gradient Operators

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

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

−1
1

−1	1

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5)$$

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6)$$

−1	0
0	1

0	−1
1	0

Roberts

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

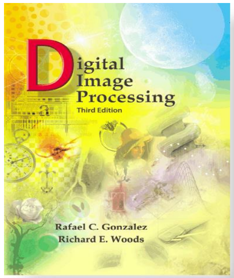
$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel



Kirsch Compass Kernels

a b c d
e f g h

FIGURE 10.15 Kirsch compass kernels. The edge direction of strongest response of each kernel is labeled below it.

−3	−3	5
−3	0	5
−3	−3	5

N

−3	5	5
−3	0	5
−3	−3	−3

NW

5	5	5
−3	0	−3
−3	−3	−3

W

5	5	−3
5	0	−3
−3	−3	−3

SW

5	−3	−3
5	0	−3
5	−3	−3

S

−3	−3	−3
5	0	−3
5	5	−3

SE

−3	−3	−3
−3	0	−3
5	5	5

E

−3	−3	−3
−3	0	5
−3	5	5

NE

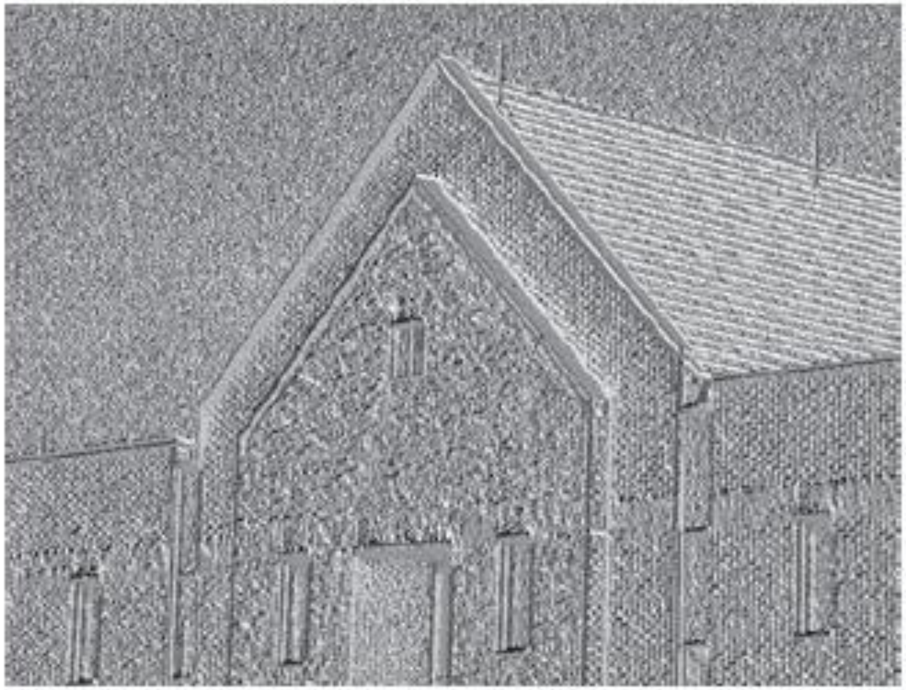
a b
c d

FIGURE 10.16 (a) Image of size 834×1114 pixels, with intensity values scaled to the range [0,1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel kernel in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the kernel in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

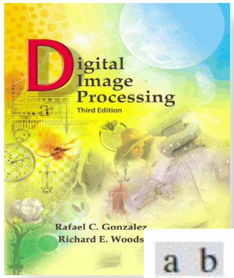
Edge map derived using Sobel operators

Gradient angle image derived using Sobel operators

FIGURE 10.17 Gradient angle image computed using Eq. (10-18). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

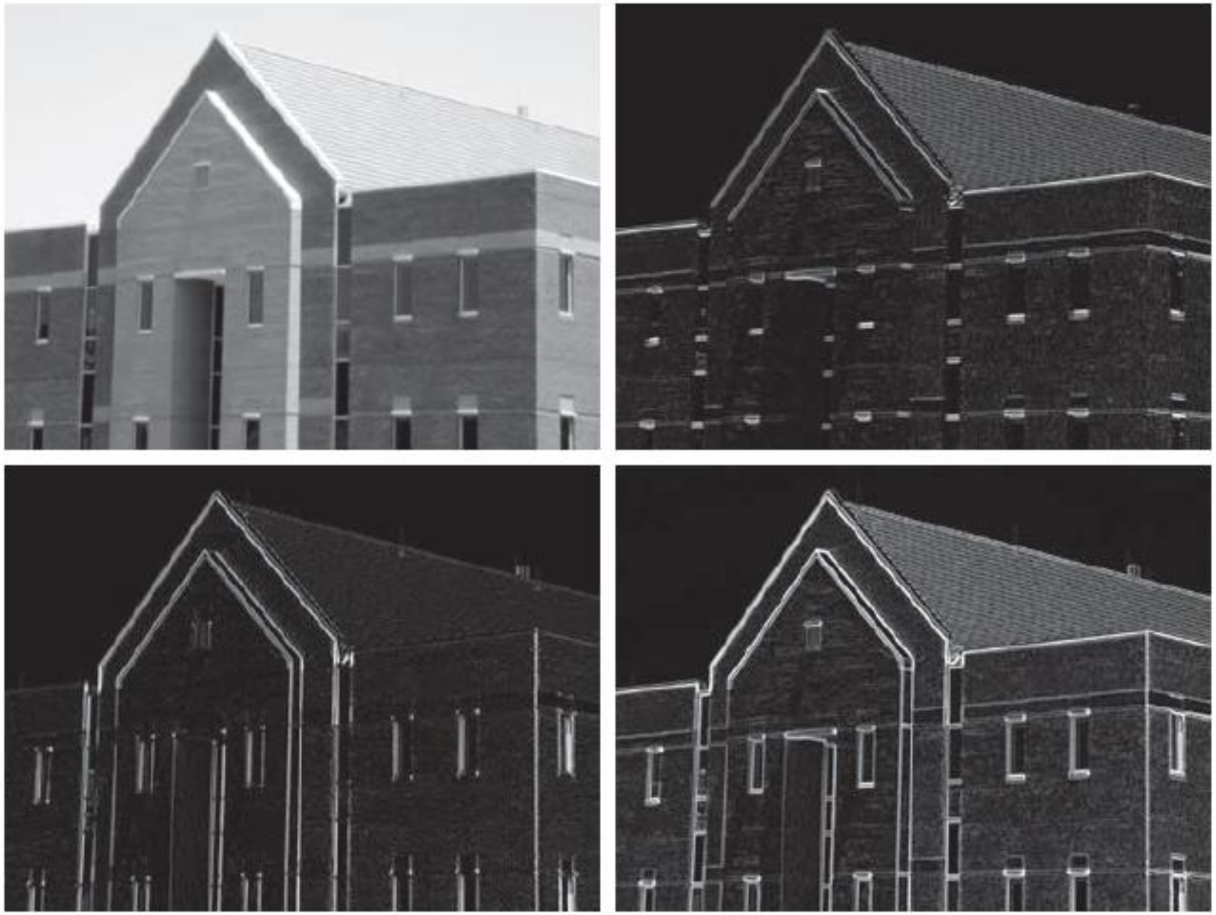


Chapter 3

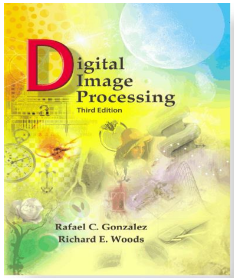


a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging kernel prior to edge detection.



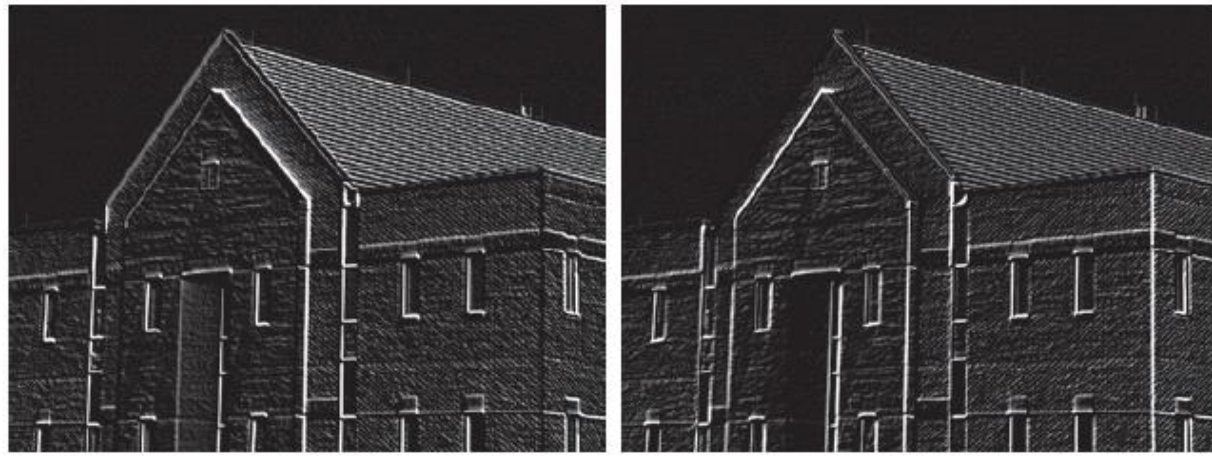
Edge-map derived using Sobel operators on smoothed image



Edgemaps for +45 and -45 using compass kernels

a b

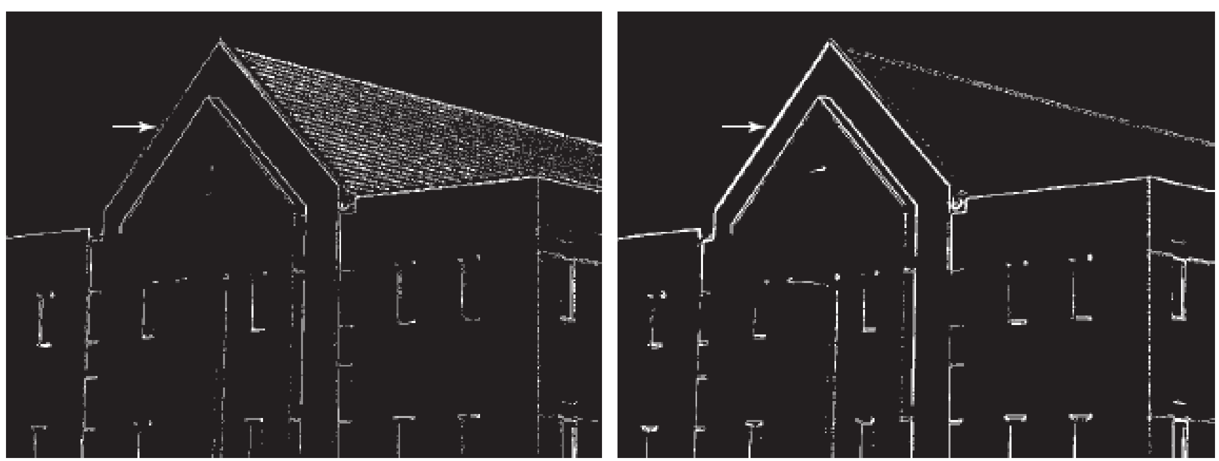
FIGURE 10.19 Diagonal edge detection. (a) Result of using the Kirsch kernel in Fig. 10.15(c). (b) Result of using the kernel in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).



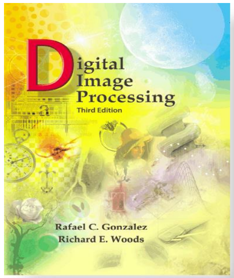
Combining the gradient with thresholding

a b

FIGURE 10.20 (a) Result of thresholding Fig. 10.16(d), the gradient of the original image. (b) Result of thresholding Fig. 10.18(d), the gradient of the smoothed image.



LAPLACIAN OF GAUSSIAN



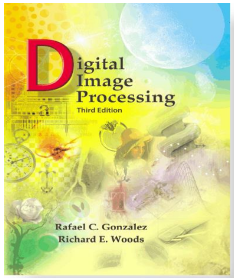
- LoG: Laplacian of Gaussian

$$\nabla^2 G \text{ where } \nabla^2 = \partial^2 / \partial x^2 + \partial^2 / \partial y^2 \text{ and } G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla^2 G(x,y) = \partial^2 G(x,y) / \partial x^2 + \partial^2 G(x,y) / \partial y^2$$

$$= \frac{1}{2\pi\sigma^2} \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] + \frac{1}{2\pi\sigma^2} \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right]$$

$$= \frac{1}{2\pi\sigma^2}\left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{1}{2\pi\sigma^2}\left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$



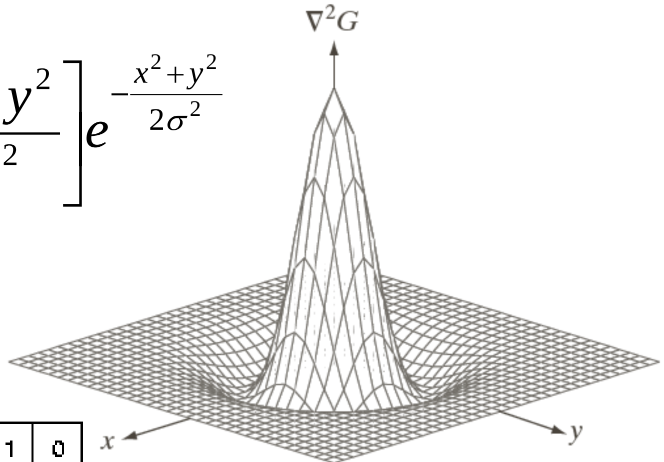
- Laplacian of Gaussian (LoG)

$$\nabla^2 G(x,y) = \frac{1}{2\pi\sigma^2}\left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

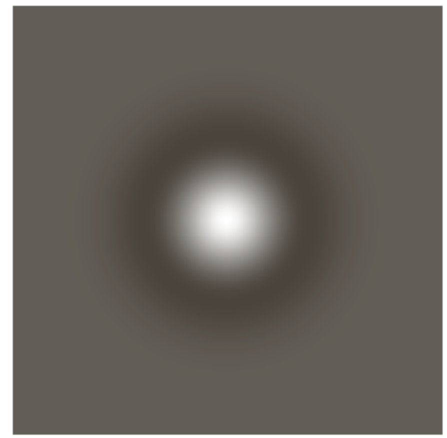
$$= -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

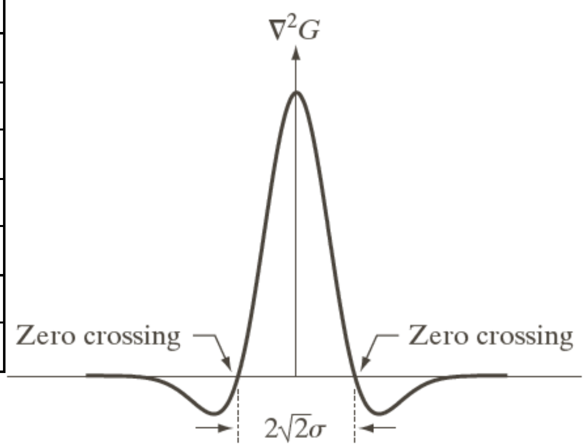
- Mexican Hat
  - positive central term
  - surrounded by an adjacent negative region
    - a function of distance
  - zero outer region

$$\nabla^2 G(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$



0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0





0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

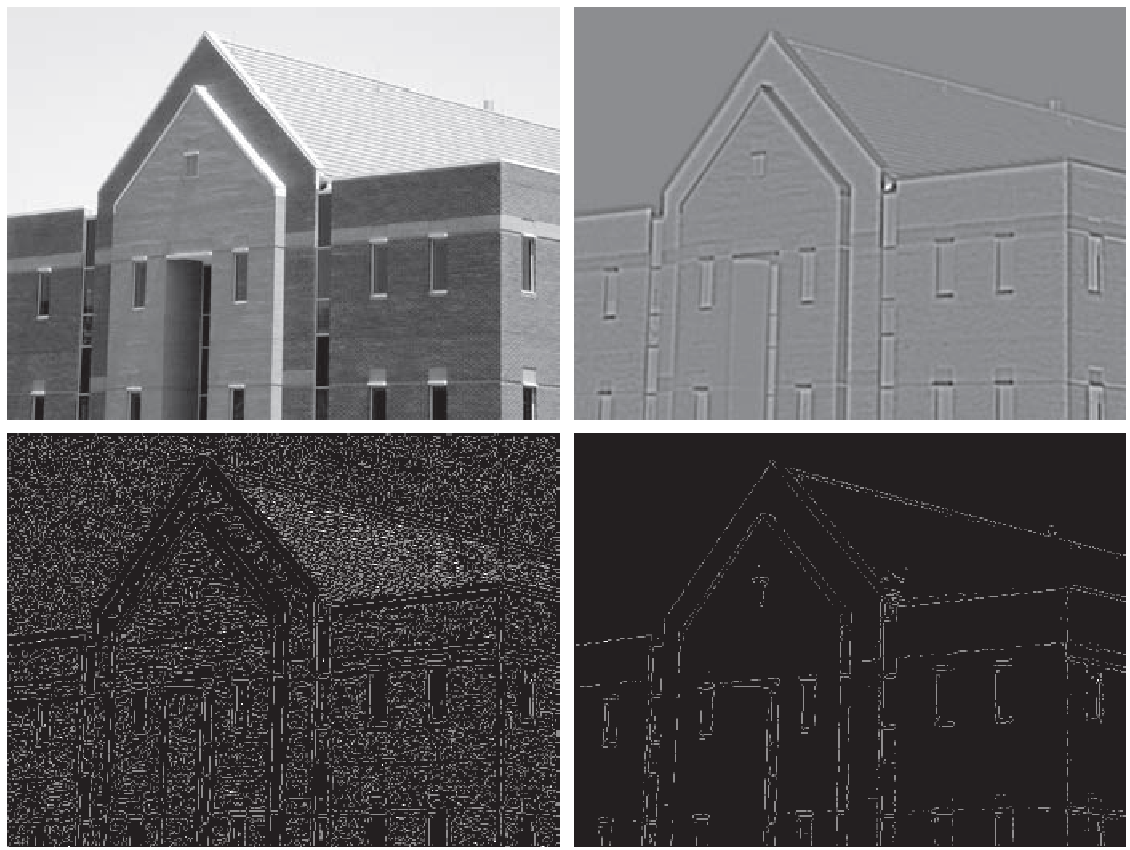
a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5 × 5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

$$g(x,y) = \left[\nabla^2 G(x,y)\right] \star f(x,y) \qquad g(x,y) = \nabla^2 \left[G(x,y) \star f(x,y)\right]$$

| a | b |
| c | d |

FIGURE 10.22 (a) Image of size 834×1114 pixels, with intensity values scaled to the range [0, 1]. (b) Result of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.



- Difference of Gaussian (DoG)

$$D_G(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \qquad \sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$$

a b

FIGURE 10.23 (a) Negatives of the LoG (solid) and DoG (dotted) profiles using a σ ratio of 1.75:1. (b) Profiles obtained using a ratio of 1.6:1.

High-Pass, Band-Reject and Band-Pass Filters from Low-Pass Filters

a	b
c	d

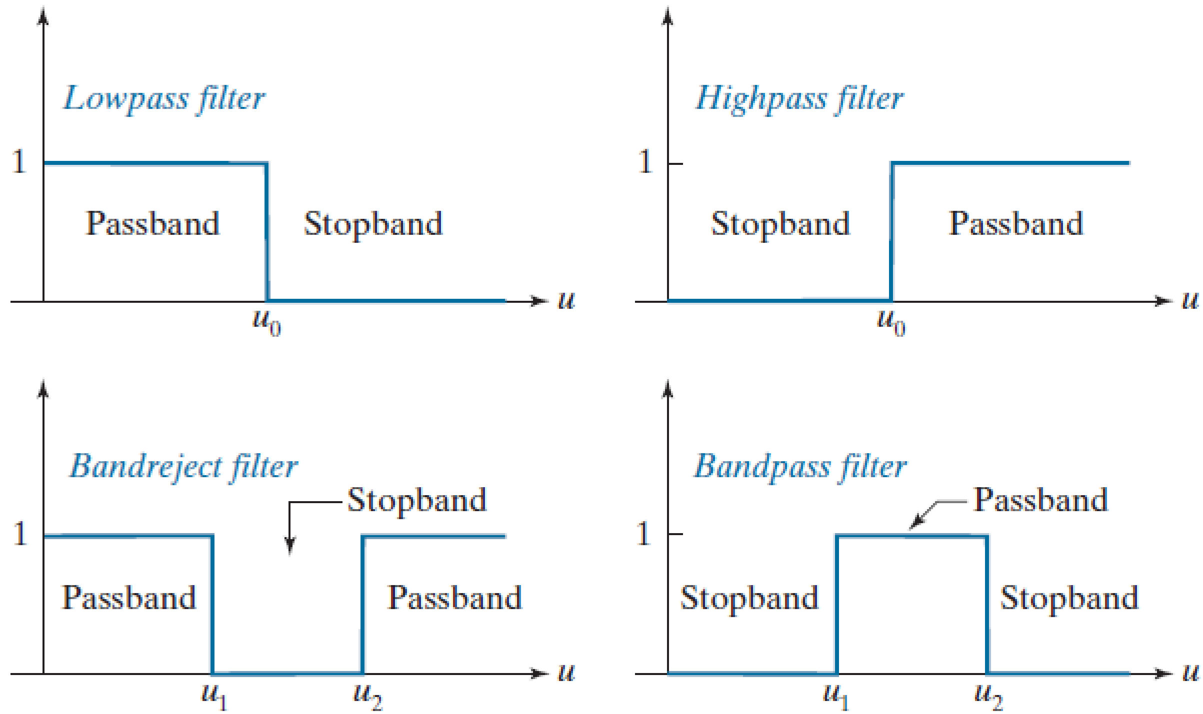
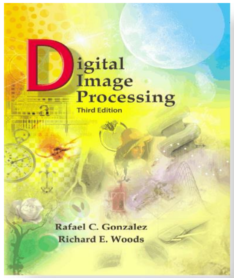


FIGURE 3.52 Transfer functions of ideal 1-D filters in the frequency domain (u denotes frequency).
(a) Lowpass filter.
(b) Highpass filter.
(c) Bandreject filter.
(d) Bandpass filter.
(As before, we show only positive frequencies for simplicity.)

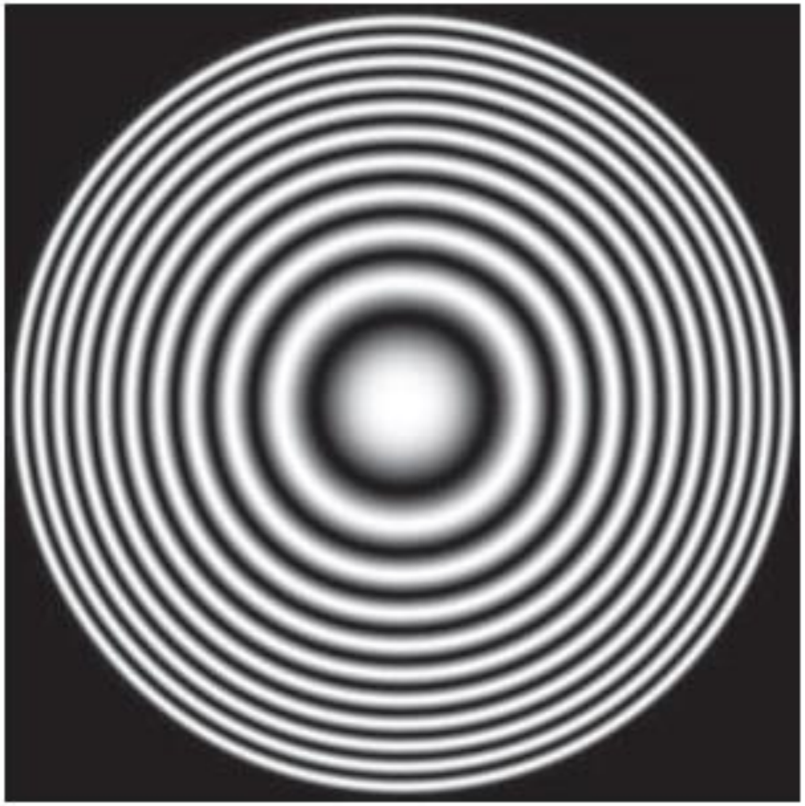


High-Pass, Band-Reject and Band-Pass Filters from Low-Pass Filters

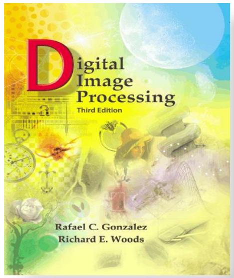
TABLE 3.7
Summary of the four principal spatial filter types expressed in terms of low-pass filters. The centers of the unit impulse and the filter kernels coincide.

Filter type	Spatial kernel in terms of lowpass kernel, *lp*
Lowpass	$lp(x, y)$
Highpass	$hp(x, y) = \delta(x, y) - lp(x, y)$
Bandreject	$br(x, y) = lp_1(x, y) + hp_2(x, y)$ $= lp_1(x, y) + [\delta(x, y) - lp_2(x, y)]$
Bandpass	$bp(x, y) = \delta(x, y) - br(x, y)$ $= \delta(x, y) - [lp_1(x, y) + [\delta(x, y) - lp_2(x, y)]]$

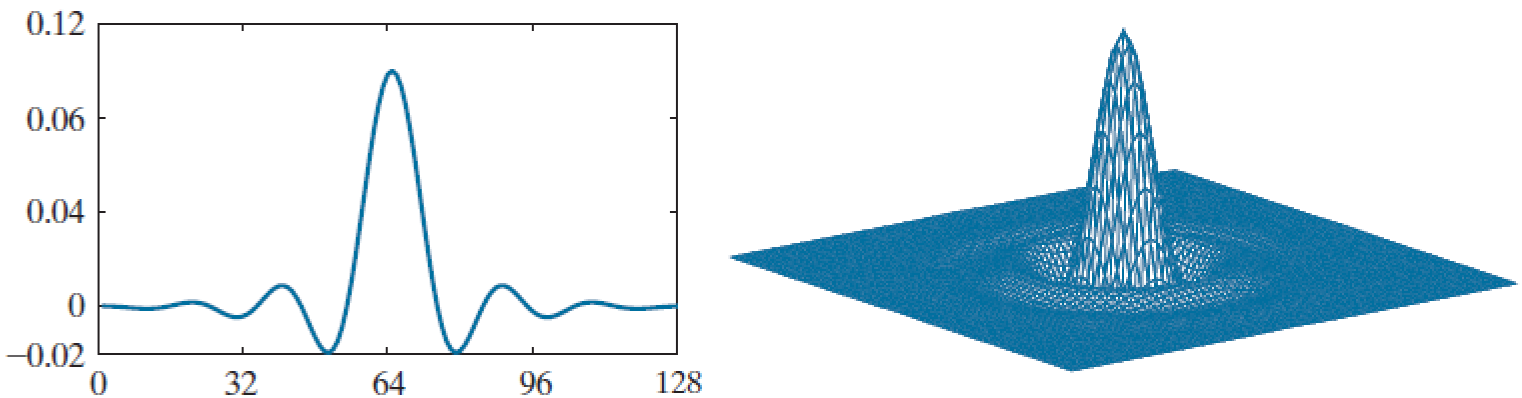
Zone Plate (597 X 597)



$$z(x,y)=\frac{1}{2}\left[1+\cos\left(x^2+y^2\right)\right]$$

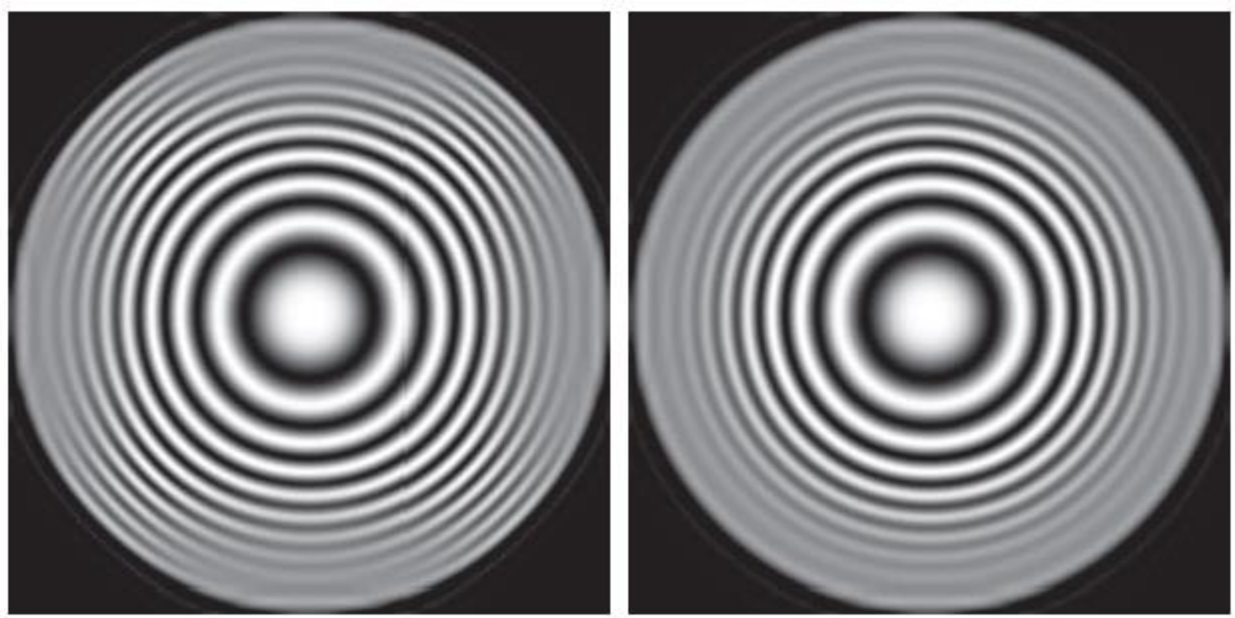


LPF of Zone Plate



a b

FIGURE 3.54
(a) A 1-D spatial lowpass filter function. (b) 2-D kernel obtained by rotating the 1-D profile about its center.



a b

FIGURE 3.55
(a) Zone plate image filtered with a separable lowpass kernel. (b) Image filtered with the isotropic lowpass kernel in Fig. 3.54(b).

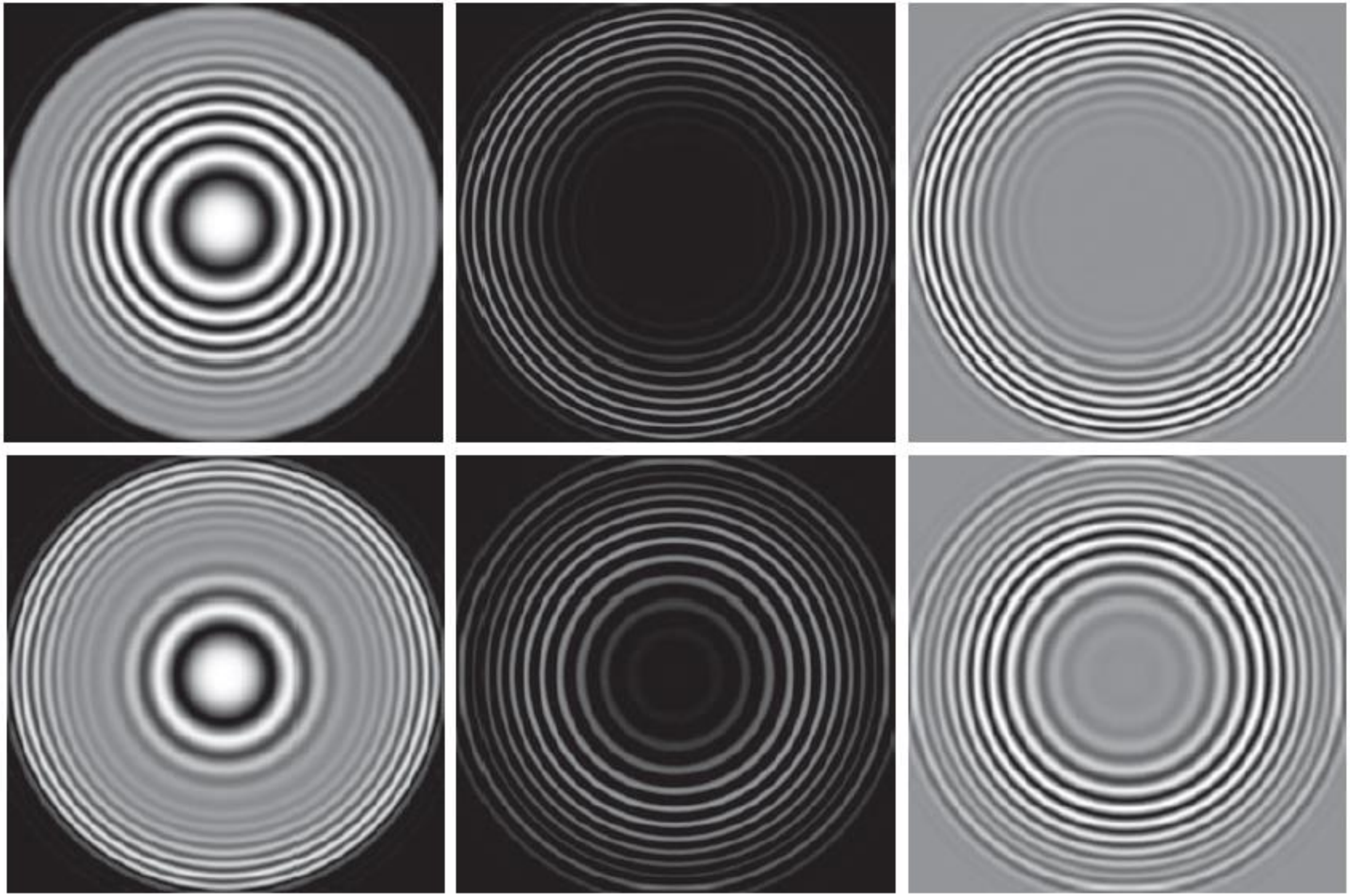
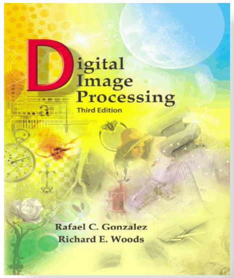
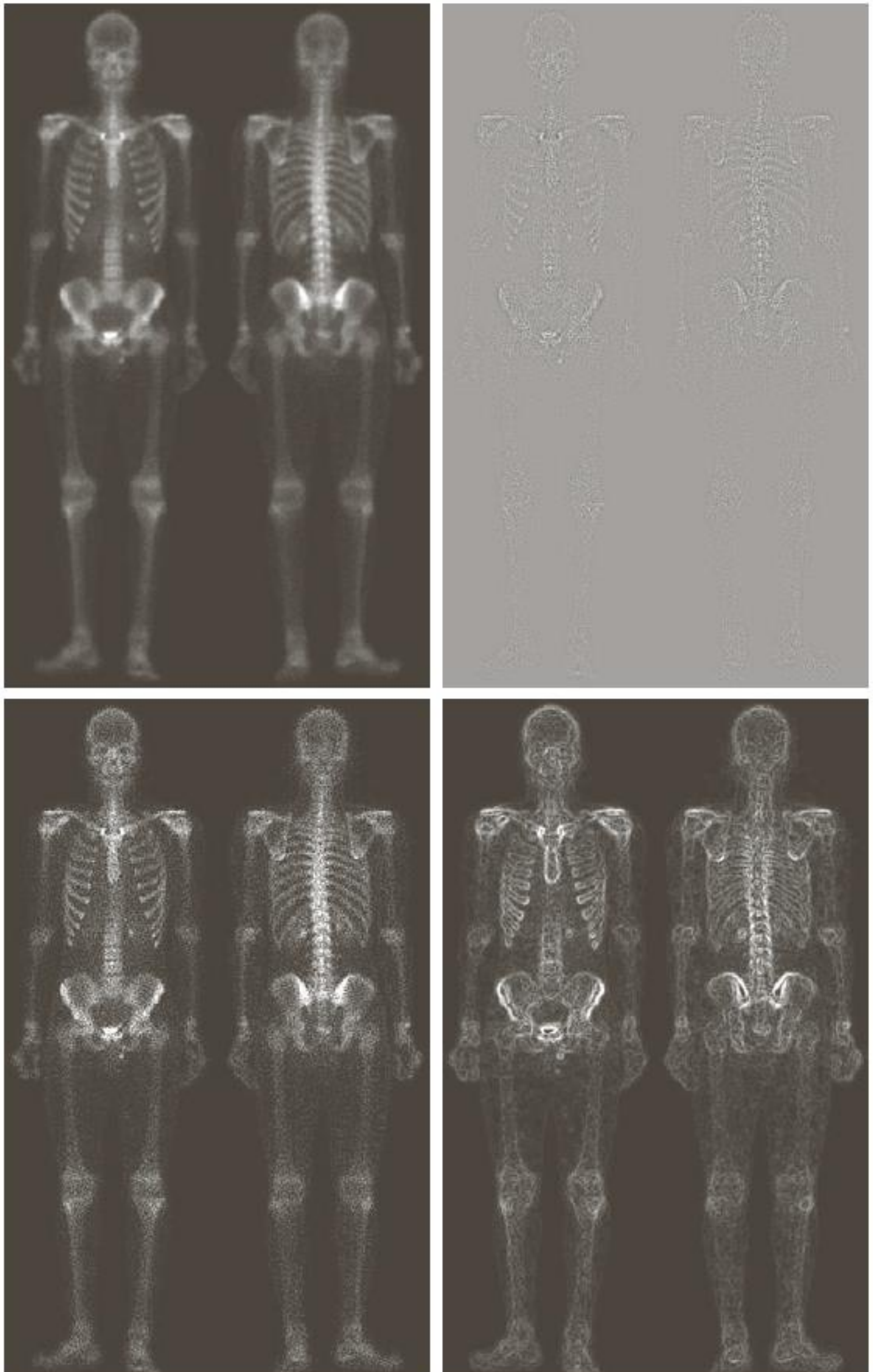


FIGURE 3.56
Spatial filtering of the zone plate image. (a) Lowpass result; this is the same as Fig. 3.55(b). (b) Highpass result. (c) Image (b) with intensities scaled. (d) Bandreject result. (e) Bandpass result. (f) Image (e) with intensities scaled.

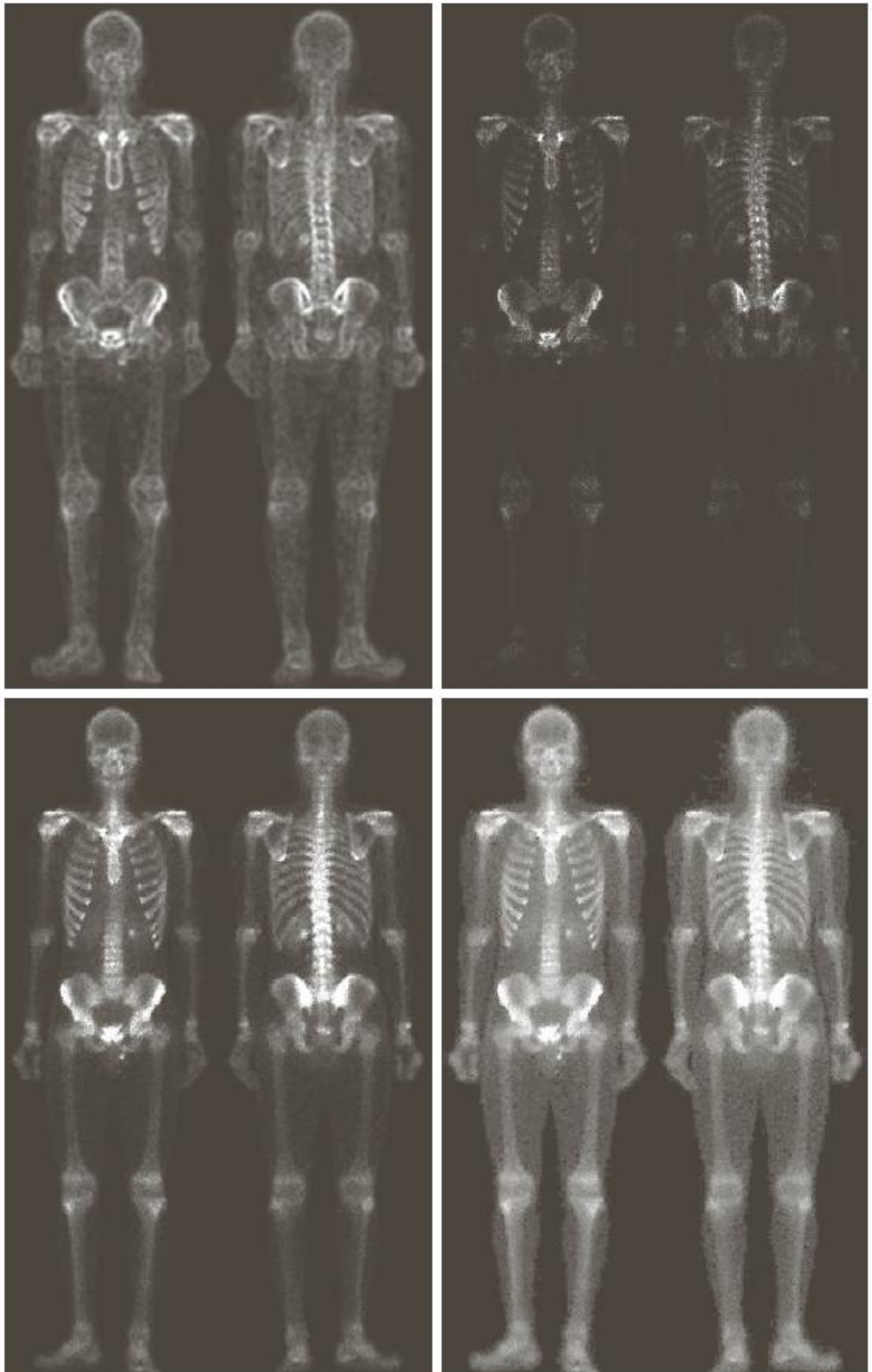


- Composite Spatial Enhancement Example:
  - Laplacian to enhance fine details
  - Gradient to enhance prominent edges
  - Smooth gradient image
  - Product of Laplacian and Gradient as mask
  - Add back to original image
  - Power-law transformation to stretch gray levels



a b
c d

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



e	f
g	h

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

- Using Fuzzy Sets for Enhancement :

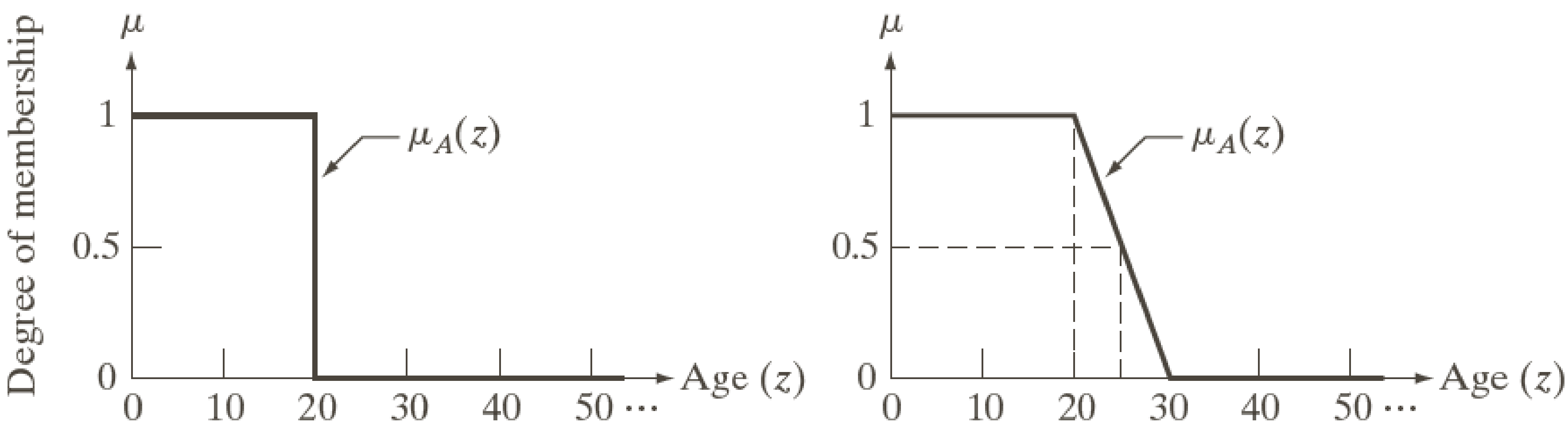
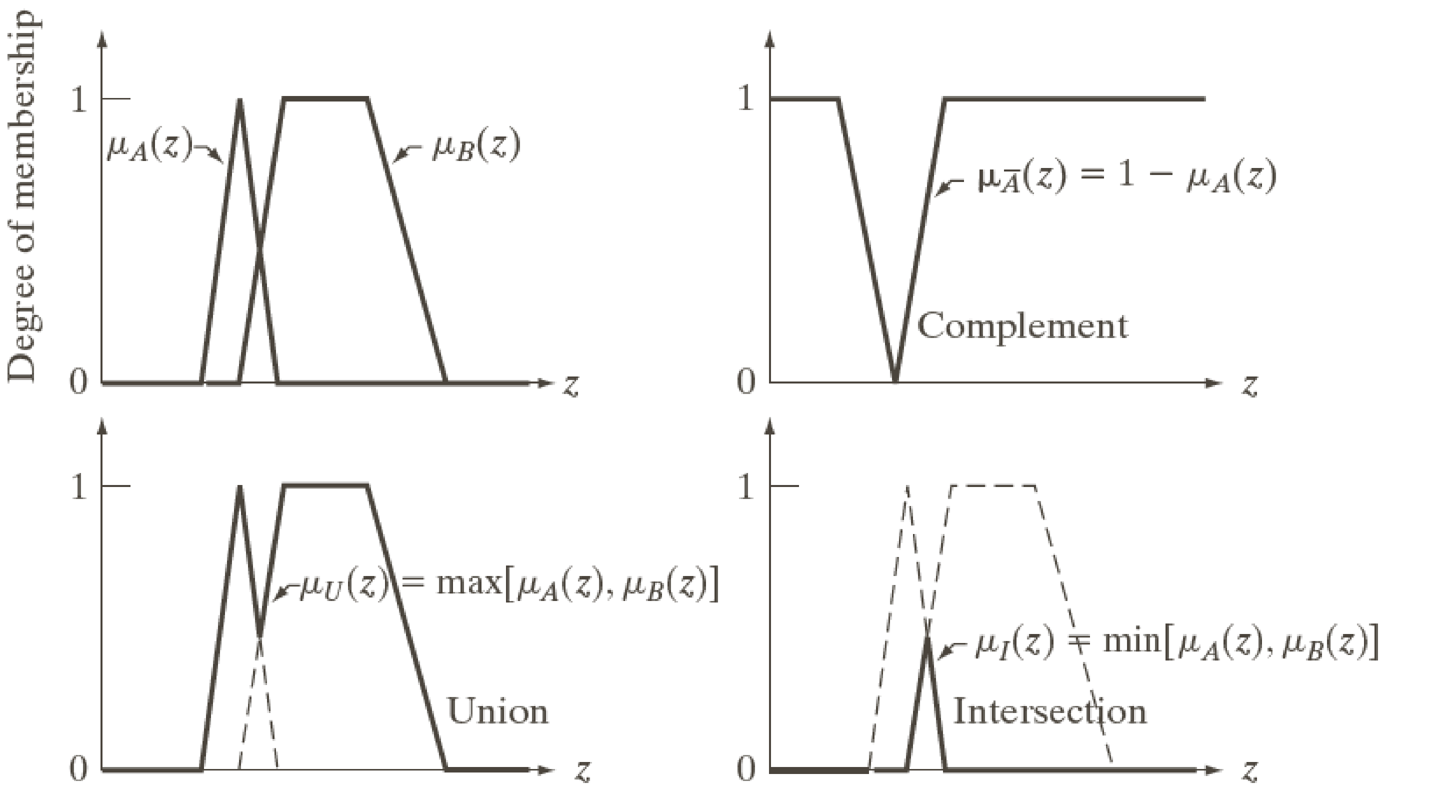


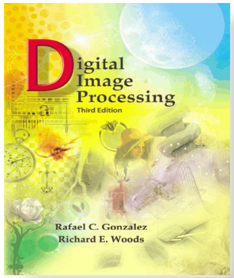
FIGURE 3.44 Membership functions used to generate (a) a crisp set, and (b) a fuzzy set.

Chapter 3
Intensity Transformations & Spatial Filtering



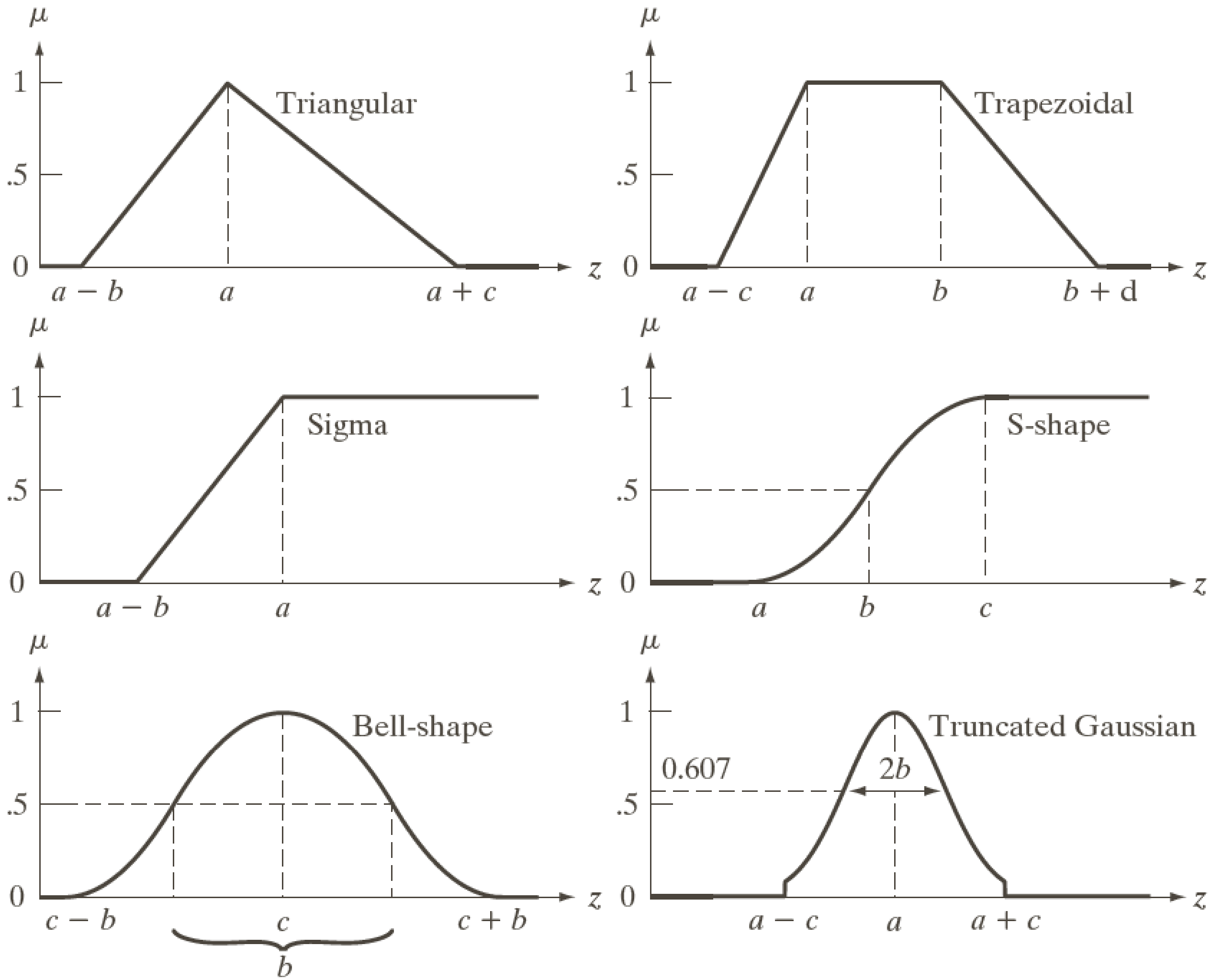
a b
c d

FIGURE 3.45 (a) Membership functions of two sets, A and B. (b) Membership function of the complement of A. (c) and (d) Membership functions of the union and intersection of the two sets.



Chapter 3

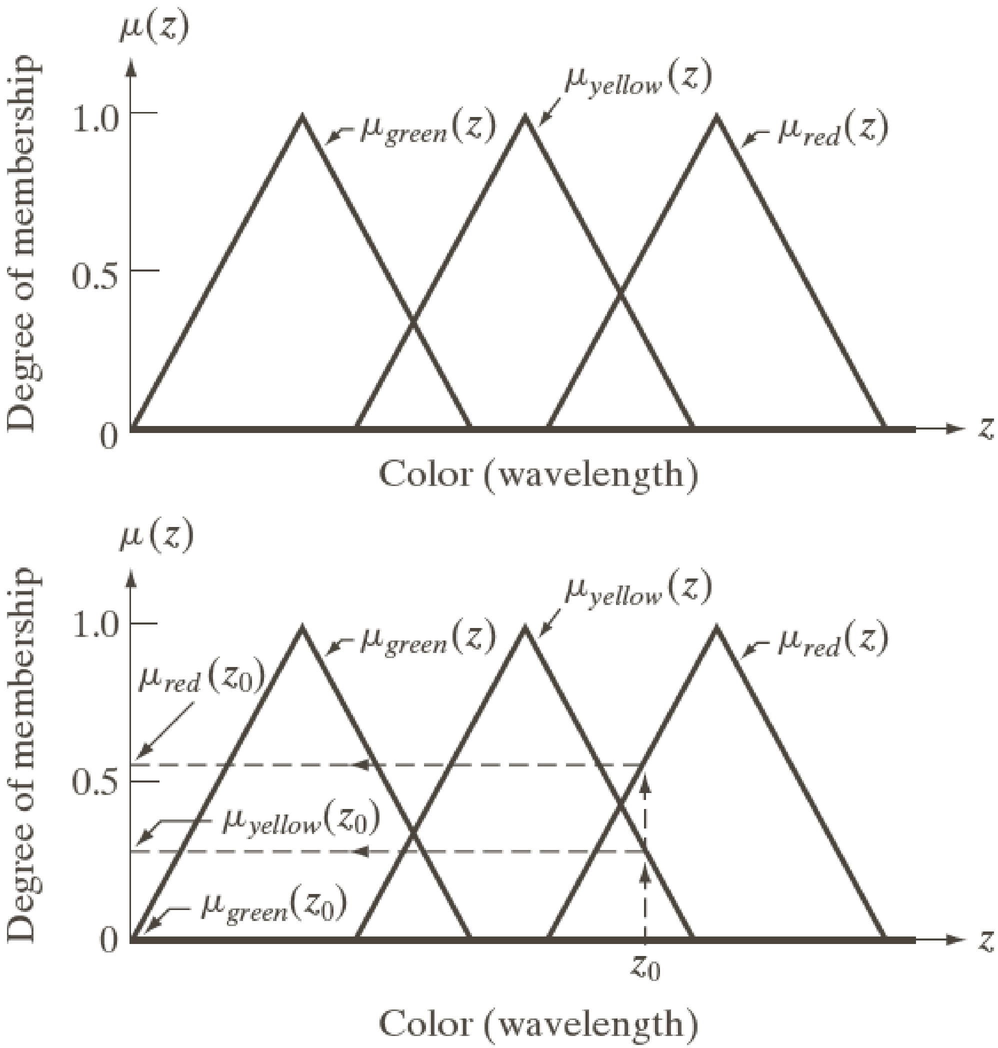
Intensity Transformations & Spatial Filtering



a	b
c	d
e	f

FIGURE 3.46 Membership functions corresponding to Eqs. (3.8-6)–(3.8-11).

Fuzzy Inference



a
b

FIGURE 3.47 (a) Membership functions used to fuzzify color. (b) Fuzzifying a specific color z_0. (Curves describing color sensation are bell shaped; see Section 6.1 for an example. However, using triangular shapes as an approximation is common practice when working with fuzzy sets.)

- Fuzzy IF-THEN Rules:

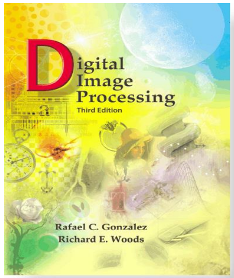
R_1: IF the color is *green*, THEN the fruit is *verdant*.

OR

R_2: IF the color is *yellow*, THEN the fruit is *half-mature*.

OR

R_3: IF the color is *red*, THEN the fruit is *mature*.



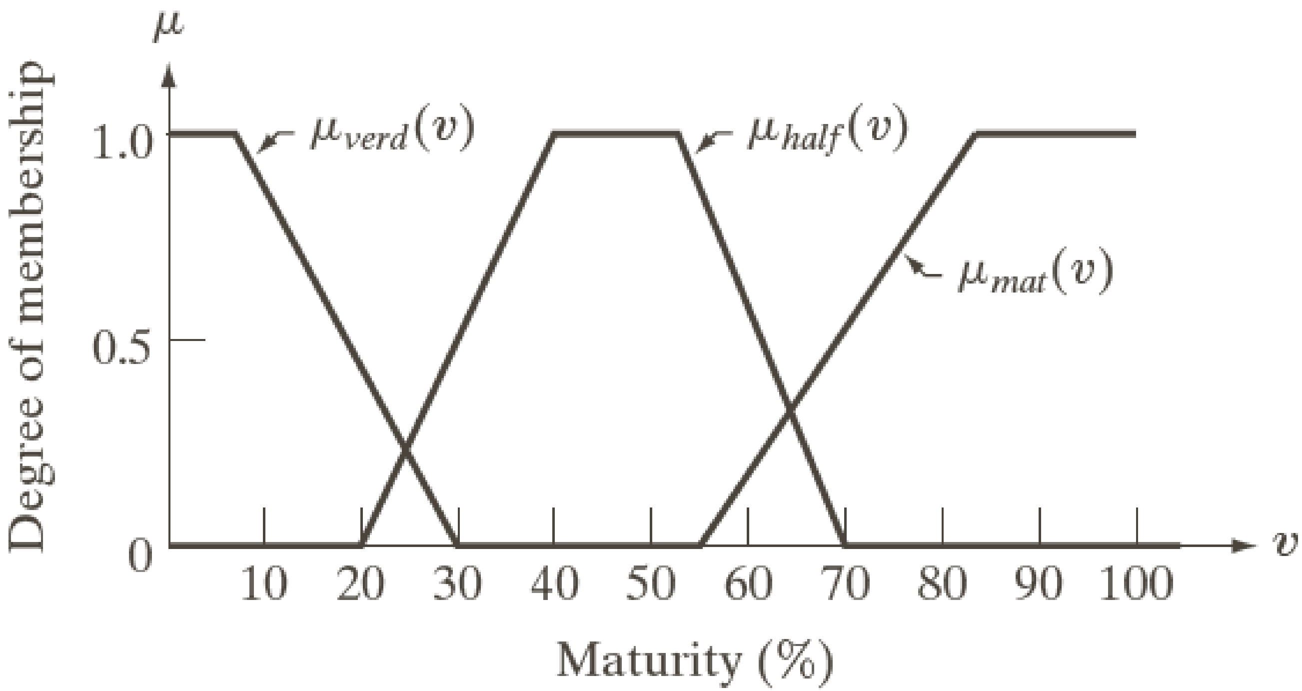


FIGURE 3.48 Membership functions characterizing the outputs *verdant*, *half-mature*, and *mature*.

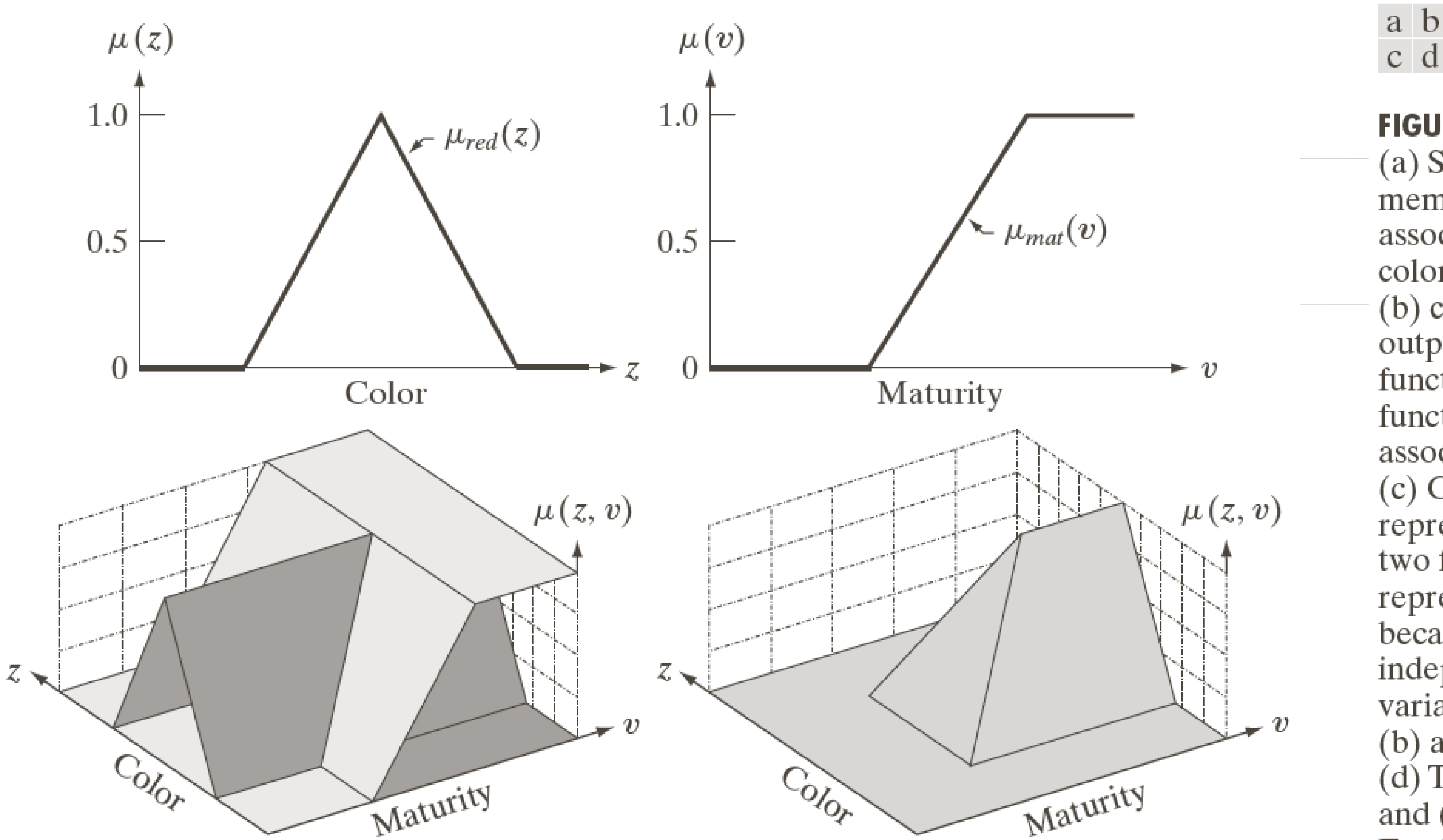
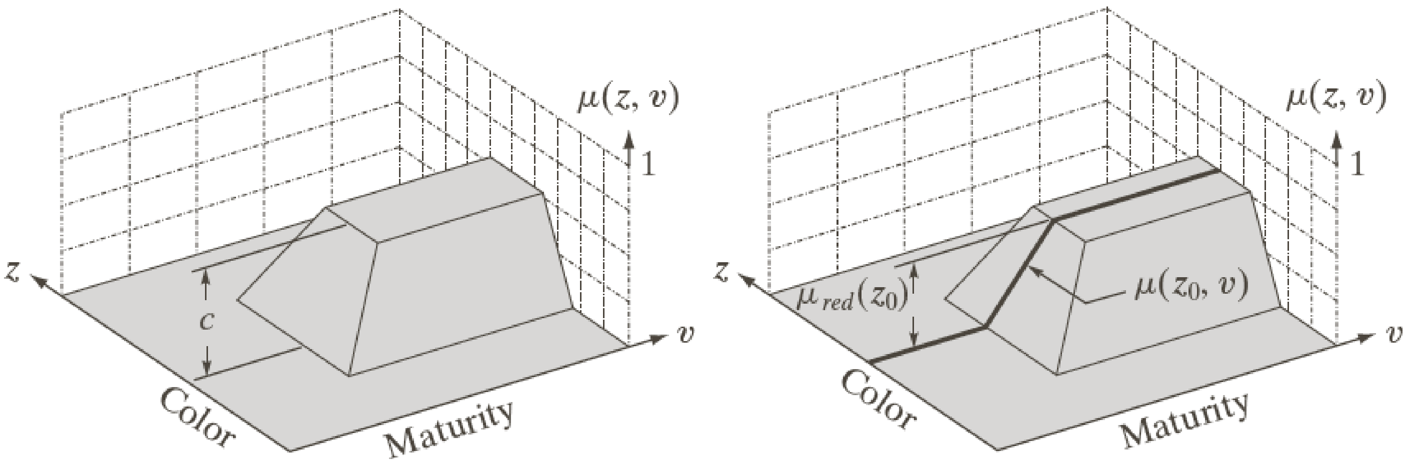


FIGURE 3.49 (a) Shape of the membership function associated with the color red, and (b) corresponding output membership function. These two functions are associated by rule R_3. (c) Combined representation of the two functions. The representation is 2-D because the independent variables in (a) and (b) are different. (d) The AND of (a) and (b), as defined in Eq. (3.8-5).

$$\mu_3(z,v) = \min\{\mu_{red}(z), \mu_{mat}(v)\}$$

Chapter 3
Intensity Transformations & Spatial Filtering



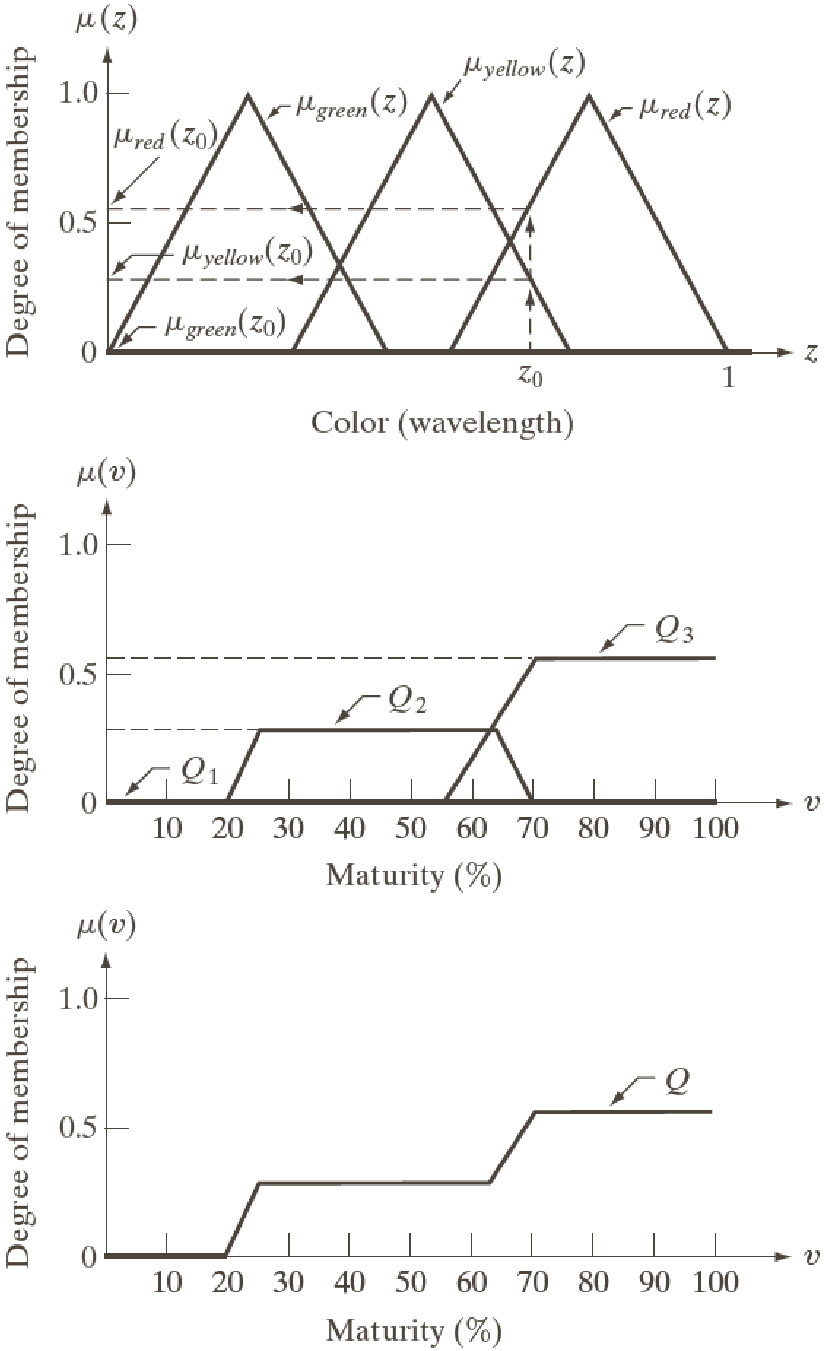
a b

FIGURE 3.50 (a) Result of computing the minimum of an arbitrary constant, c, and function $\mu_3(z, v)$ from Eq. (3.8-12). The minimum is equivalent to an AND operation. (b) Cross section (dark line) at a specific color, z_0.

$$Q_3(v) = \min\{\mu_{red}(z_0), \mu_3(z_0, v)\}$$

$$Q_2(v) = \min\{\mu_{yellow}(z_0), \mu_2(z_0, v)\}$$

$$Q_1(v) = \min\{\mu_{green}(z_0), \mu_1(z_0, v)\}$$



$$Q = Q_1 \; OR \; Q_2 \; OR \; Q_3$$

$$Q(v) = \max_r \{\min_s \{\mu_s(z_0), \mu_r(z_0, v)\}\}$$

Defuzzification

$$v_0 = \frac{\sum_{v=1}^{K} v Q(v)}{\sum_{v=1}^{K} Q(v)}$$

a
b
c

FIGURE 3.51 (a) Membership functions with a specific color, z_0, selected. (b) Individual fuzzy sets obtained from Eqs. (3.8-13)–(3.8-15). (c) Final fuzzy set obtained by using Eq. (3.8-16) or (3.8-17).

- Fuzzy Rule-based Logic:

R_1: IF (z, *green*) THEN (v, *verdant*).

OR

R_2: IF (z, *yellow*) THEN (v, *half-mature*).

OR

R_3: IF (z, *red*) THEN (v, *mature*).

- Fuzzy Rule-based Logic:

IF (z_1, A_{11}) AND (z_2, A_{12}) AND … AND (z_N, A_{1N}) THEN (v, B_1).

IF (z_1, A_{21}) AND (z_2, A_{22}) AND … AND (z_N, A_{2N}) THEN (v, B_2).

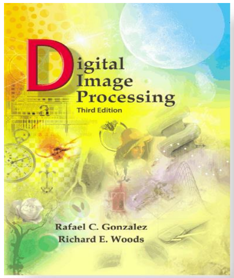
…

IF (z_1, A_{M1}) AND (z_2, A_{M2}) AND … AND (z_N, A_{MN}) THEN (v, B_M).

ELSE (v, B_E).

$$\lambda_i = \min\{\mu_{A_{ij}}(z_j) : j = 1,2,\ldots M\}$$

$$\lambda_E = \min\{1 - \lambda_i : i = 1,2,\ldots M\}$$



- Fuzzy Rule-based Logic:
  - Fuzzify the inputs
  - Perform any required fuzzy logical operations
  - Apply an implication method
  - Apply an aggregation method
  - Defuzzify the final output

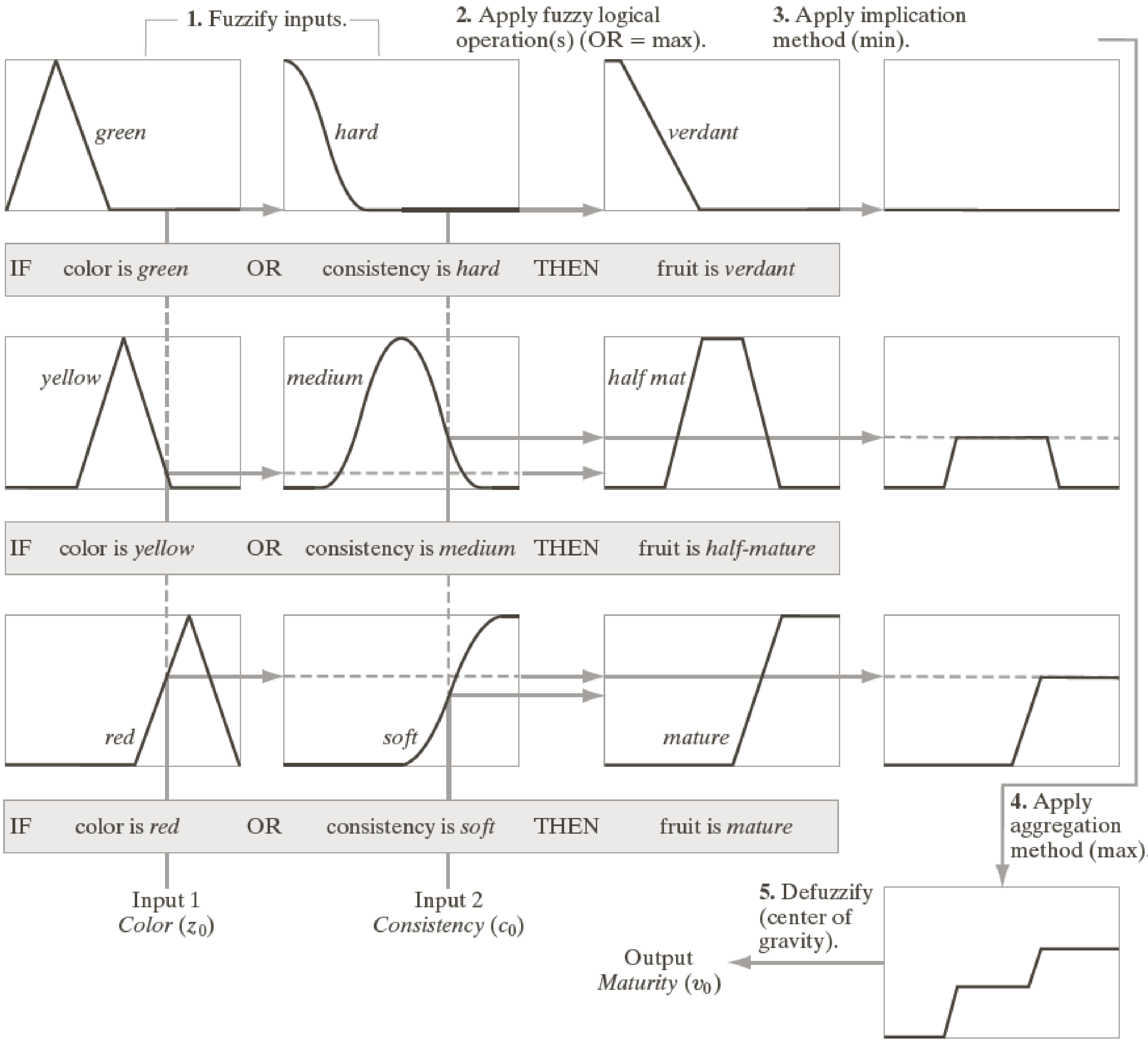


FIGURE 3.52 Example illustrating the five basic steps used typically to implement a fuzzy, rule-based system: (1) fuzzification, (2) logical operations (only OR was used in this example), (3) implication, (4) aggregation, and (5) defuzzification.

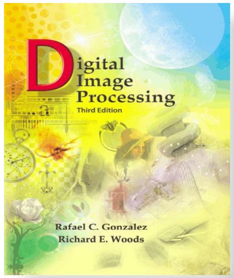
- Using Fuzzy Sets for Intensity Transformation:

IF a pixel is dark THEN make it darker.

IF a pixel is gray THEN make it gray.

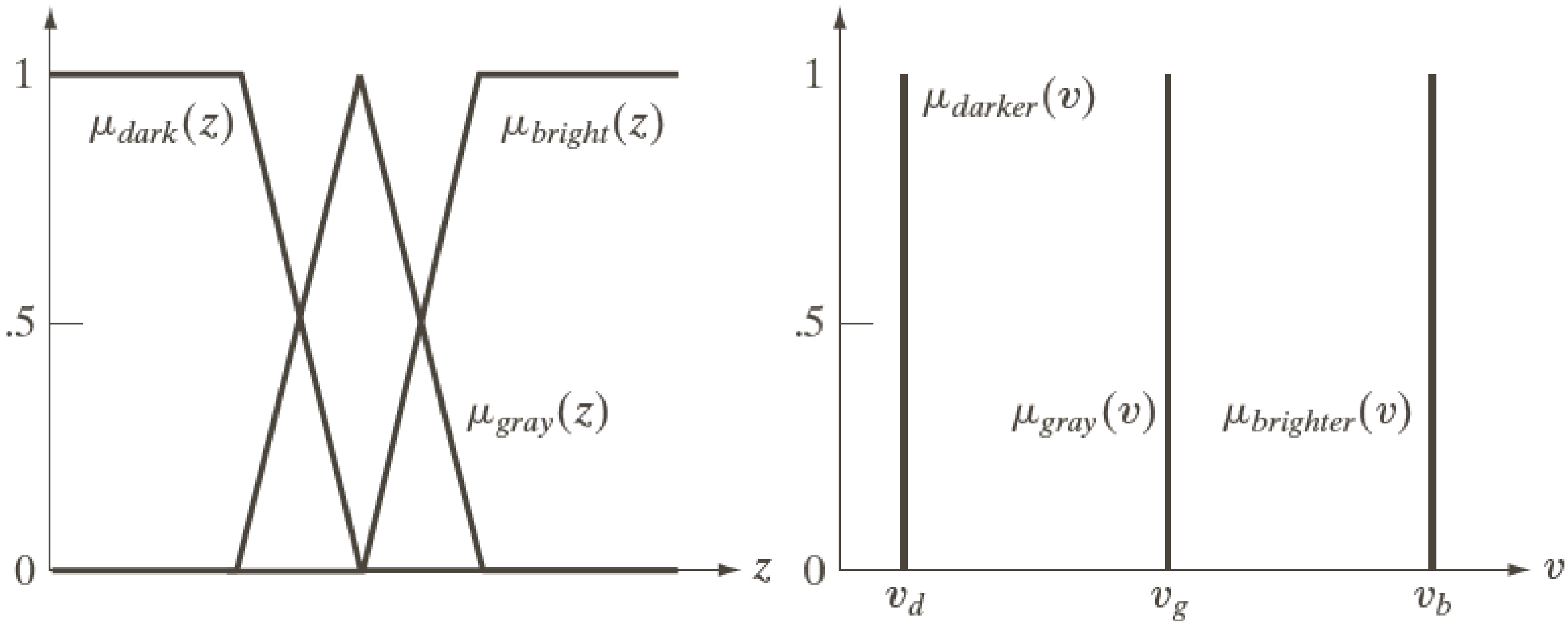
IF a pixel is bright THEN make it brighter.

$$v_0 = \frac{\mu_{dark}(z_0) \times v_d + \mu_{gray}(z_0) \times v_g + \mu_{bright}(z_0) \times v_b}{\mu_{dark}(z_0) + \mu_{gray}(z_0) + \mu_{bright}(z_0)}$$



a b

FIGURE 3.53 (a) Input and (b) output membership functions for fuzzy, rule-based contrast enhancement.

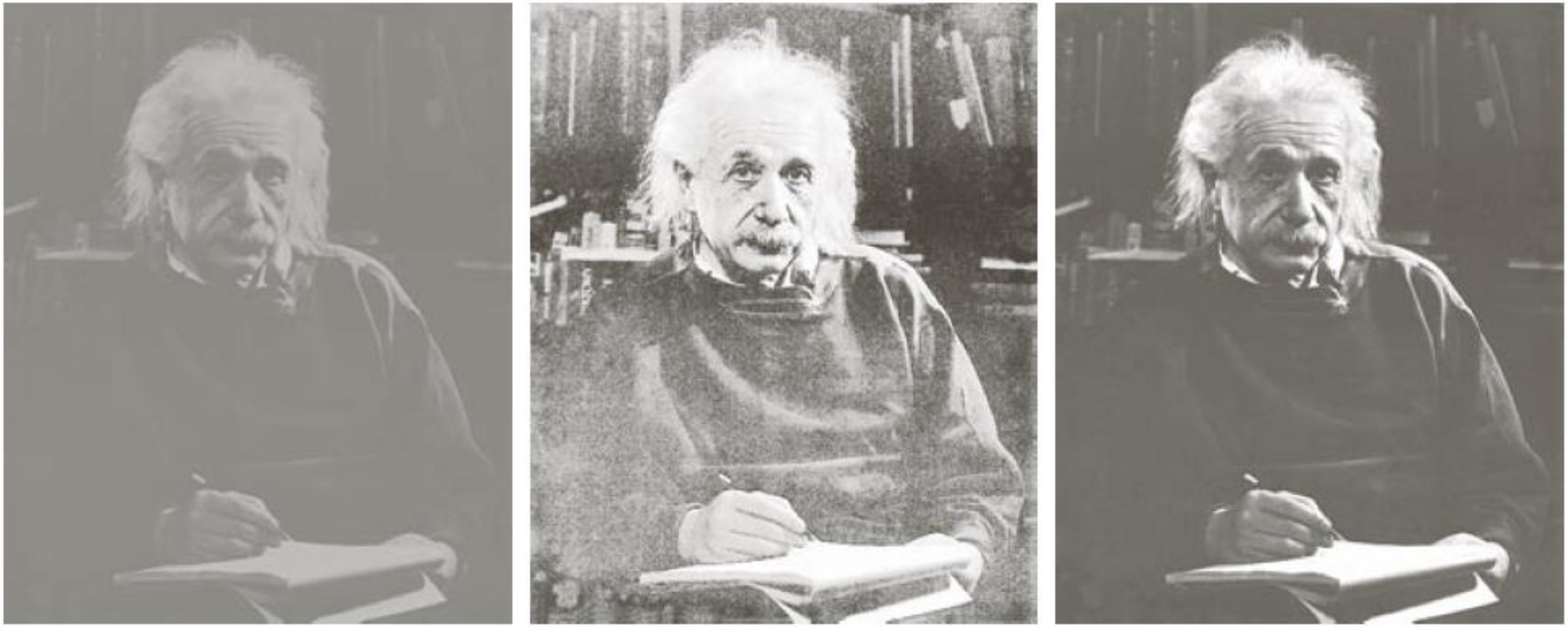


$$v_0 = \frac{\mu_{dark}(z_0) \times v_d + \mu_{gray}(z_0) \times v_g + \mu_{bright}(z_0) \times v_b}{\mu_{dark}(z_0) + \mu_{gray}(z_0) + \mu_{bright}(z_0)}$$

$$v_d = 0\,(\text{black}), v_g = 127\,(\text{mid-gray}), v_b = 255\,(\text{white})$$

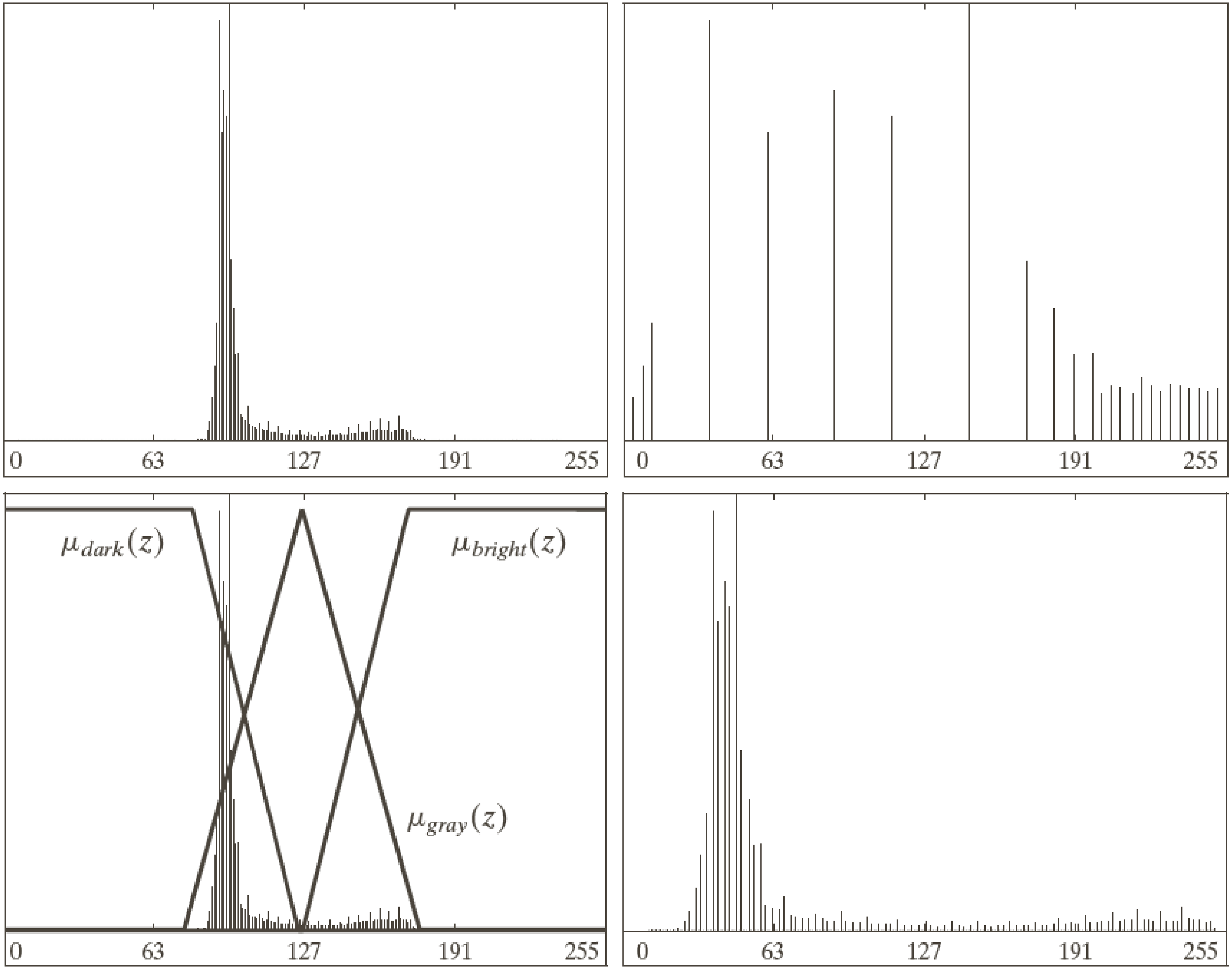
Chapter 3

Intensity Transformations & Spatial Filtering



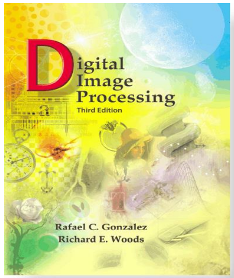
a b c

FIGURE 3.54 (a) Low-contrast image. (b) Result of histogram equalization. (c) Result of using fuzzy, rule-based contrast enhancement.



a b
c d

FIGURE 3.55 (a) and (b) Histograms of Figs. 3.54(a) and (b). (c) Input membership functions superimposed on (a). (d) Histogram of Fig. 3.54(c).



- Histogram Equalization over-exposes
  - Forces extreme darkness
  - Forces extreme whiteness
- Fuzzy Approach
  - Provides moderate shift to dark & white peaks
  - Computationally more expensive
- Histogram Specification can improve speed using the histogram obtained from the fuzzy approach

- Using Fuzzy Sets for Spatial Filtering:

  "If a pixel belongs to a uniform region, then make it *white*; else make it *black* where *black* & *white* are fuzzy sets"

IF d_2 is *zero* AND d_6 is *zero* THEN z_5 is *white*.

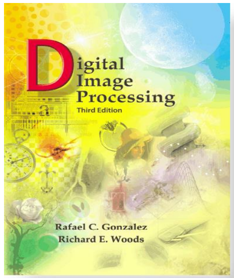
IF d_6 is *zero* AND d_8 is *zero* THEN z_5 is *white*.

IF d_8 is *zero* AND d_4 is *zero* THEN z_5 is *white*.

IF d_4 is *zero* AND d_2 is *zero* THEN z_5 is *white*.

$\mathbf{d_i = z_i - z_5}$

ELSE z_5 is *black*.



z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixel neighborhood

d_1	d_2	d_3
d_4	0	d_6
d_7	d_8	d_9

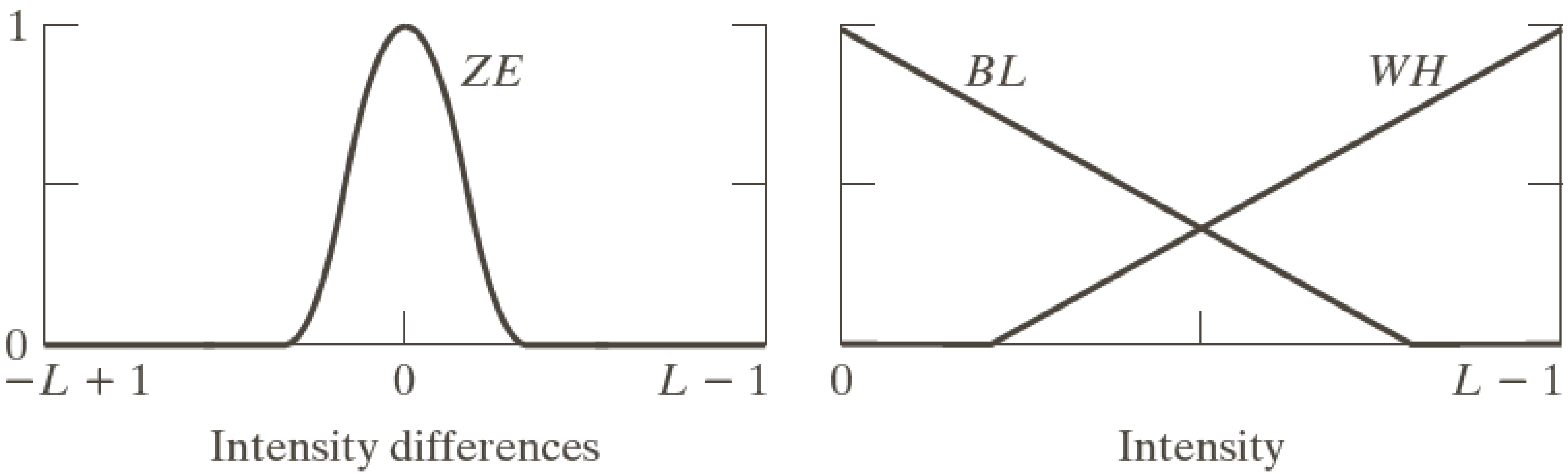
Intensity differences

a b

FIGURE 3.56 (a) A 3 × 3 pixel neighborhood, and (b) corresponding intensity differences between the center pixels and its neighbors. Only d_2, d_4, d_6, and d_8 were used in the present application to simplify the discussion.

a b

FIGURE 3.57 (a) Membership function of the fuzzy set *zero*. (b) Membership functions of the fuzzy sets *black* and *white*.



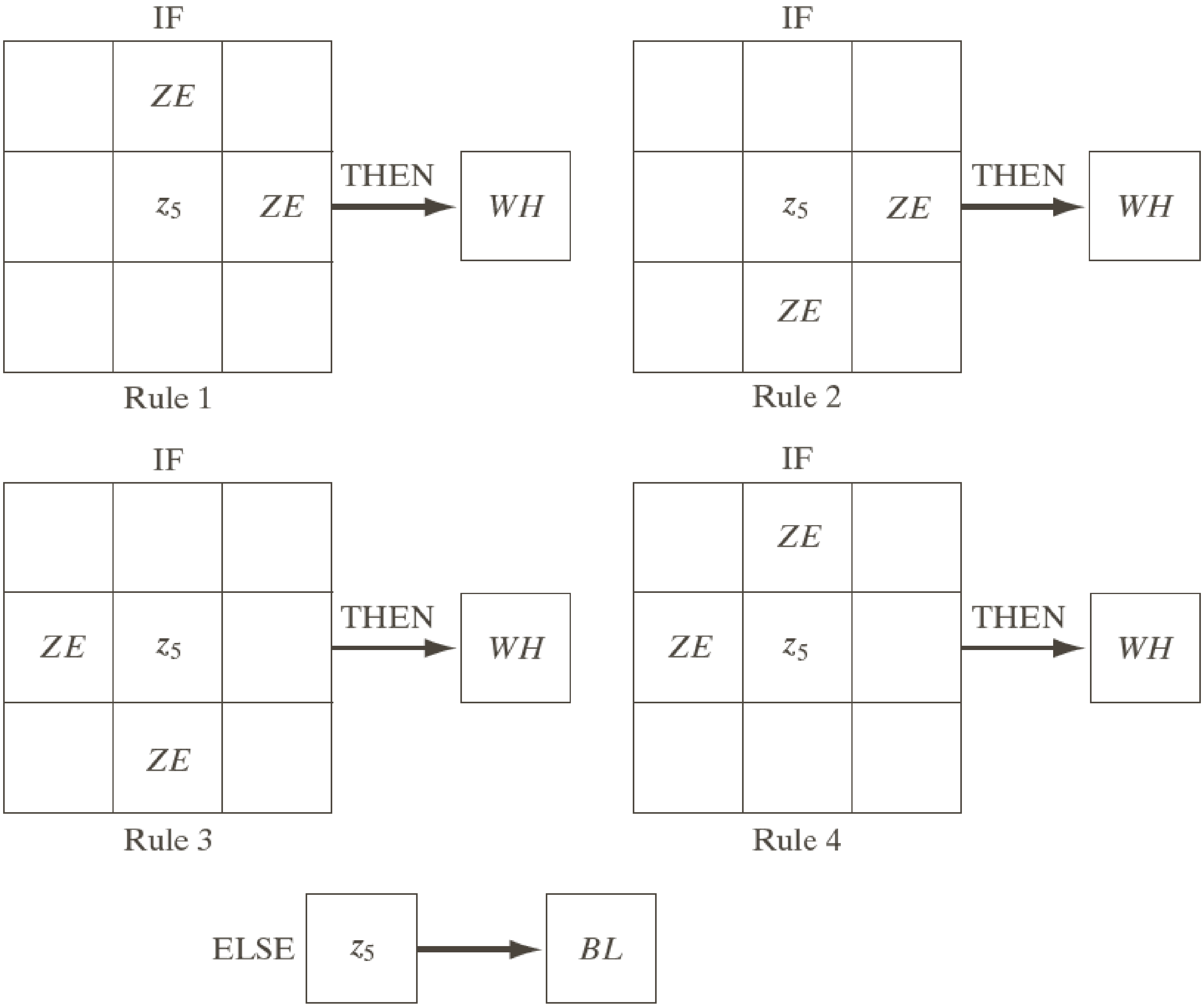


FIGURE 3.58 Fuzzy rules for boundary detection.

FIGURE 3.59 (a) CT scan of a human head. (b) Result of fuzzy spatial filtering using the membership functions in Fig. 3.57 and the rules in Fig. 3.58. (c) Result after intensity scaling. The thin black picture borders in (b) and (c) were added for clarity; they are not part of the data. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Scaling :

$$f_m = f - \min(f)$$

$$f_s = K[f_m / \max(f_m)]$$