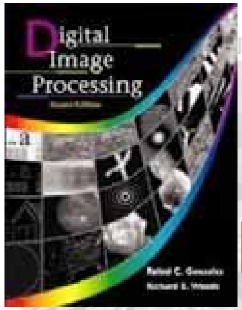
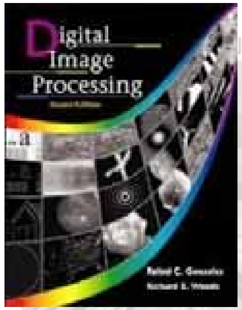


Chapter 3
Image Enhancement in the Spatial Domain

- Image Enhancement: process an image to make it more suitable for ***certain specific application.***
  - In spatial domain
  - In frequency domain (chapter 4)



3.1 Background

- Spatial domain process on image can be described as

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

Where $f(x,y)$ is the input image, $g(x,y)$ is the output image, T is an operator

T operates on the neighbors of (x, y) (a square or rectangular sub-image centered at (x,y) to yield the output $g(x, y)$.

3.1 Background

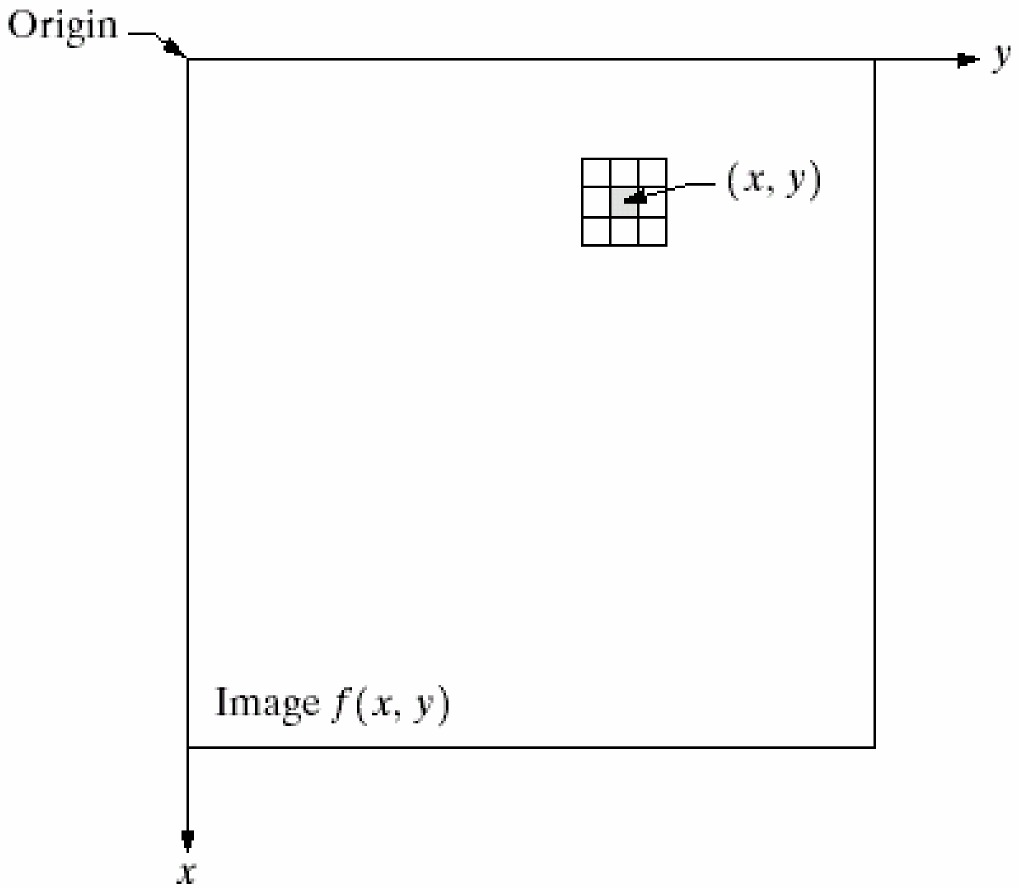


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.

3.1 Background

- The simplest form of T is the neighborhood is of size 1×1. g depends on the value of f at (*x*, *y*) which is a *gray level transformation* as

  $$s=T(r)$$

  Where r and s are the gray-level of f(x, y) and g(x, y) at any point (x, y).

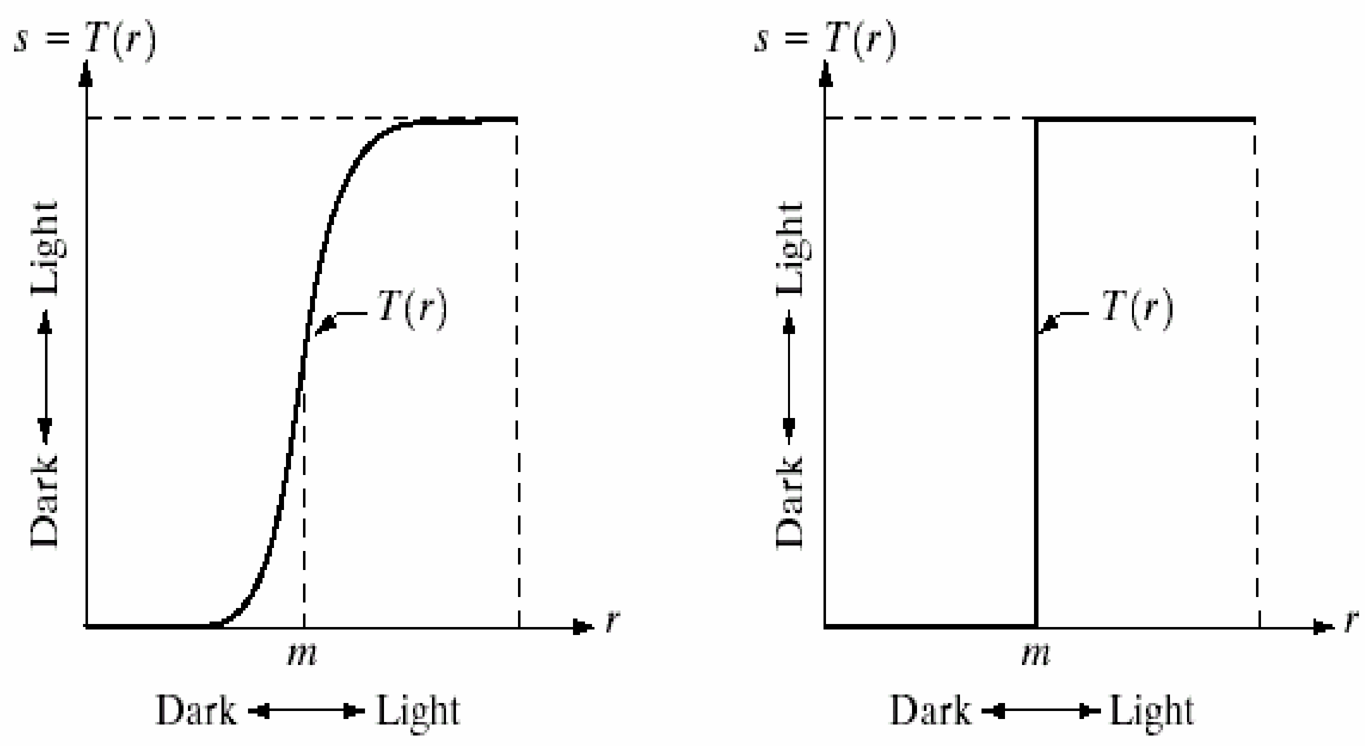
  Fig. 3.2(a) provides *Contrast stretching*

  Fig. 3.2(b) provides *Thresholding*.

  Enhancement of any point depend on that point only

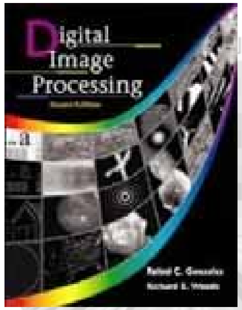
  - *Point processing*
- Larger neighborhood provides more flexibility
  - *Mask processing or filtering*

3.1 Background



a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.



3.2 Basic gray level transformations

- Three basic functions used in image enhancement
  - Linear (negative and identity transformation)

$$s=L-1-r$$

  - Logarithmic (log and inverse log)

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

  - Power law (*n*th power and *n*th root transformation)

$$s=cr^{\gamma} \quad or \quad s=c(r+\varepsilon)^{\gamma}$$

3.2 Basic gray level transformations

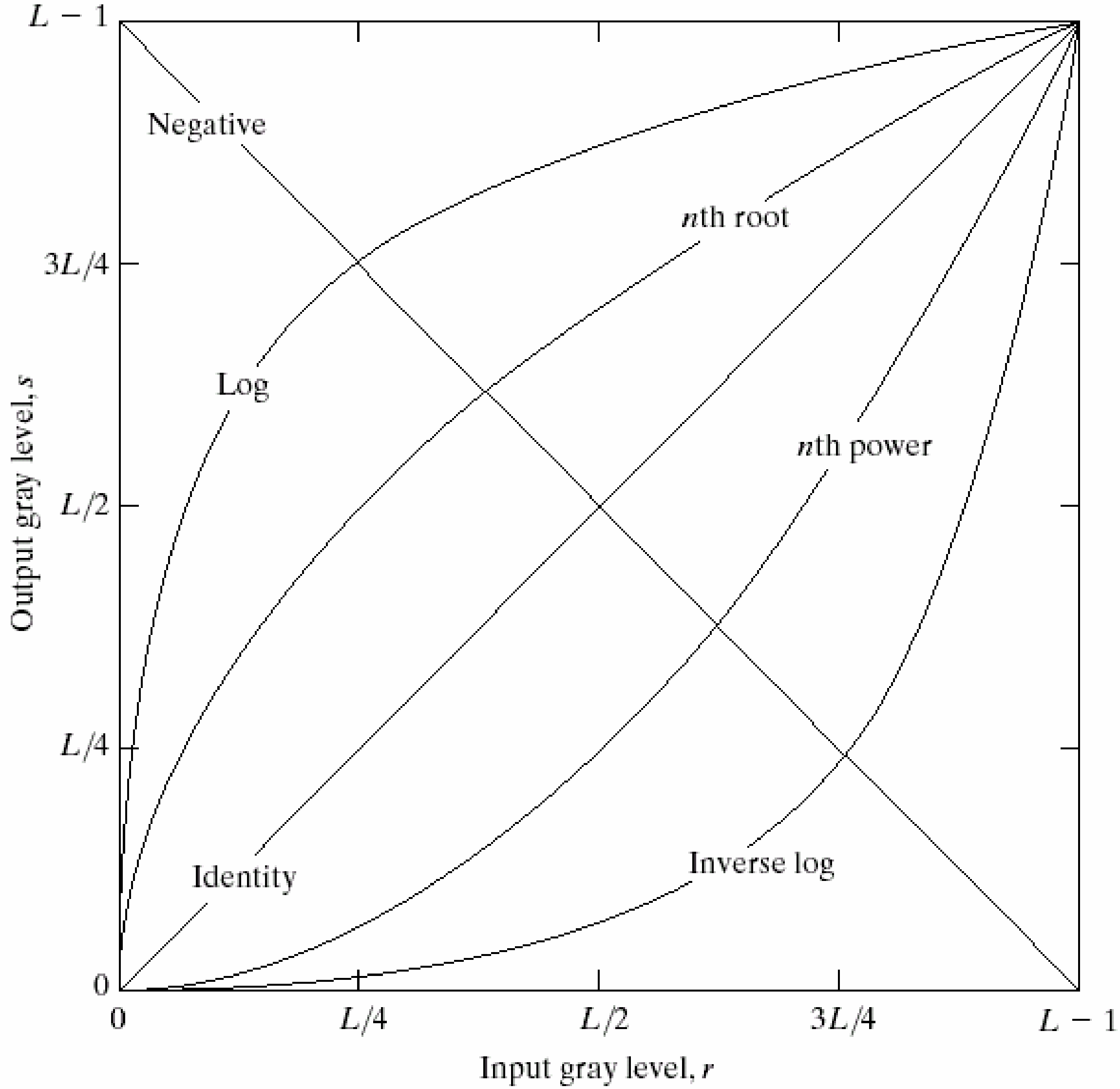
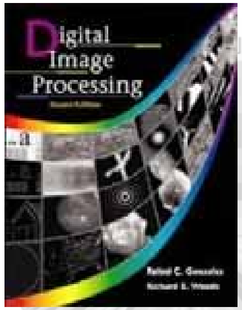
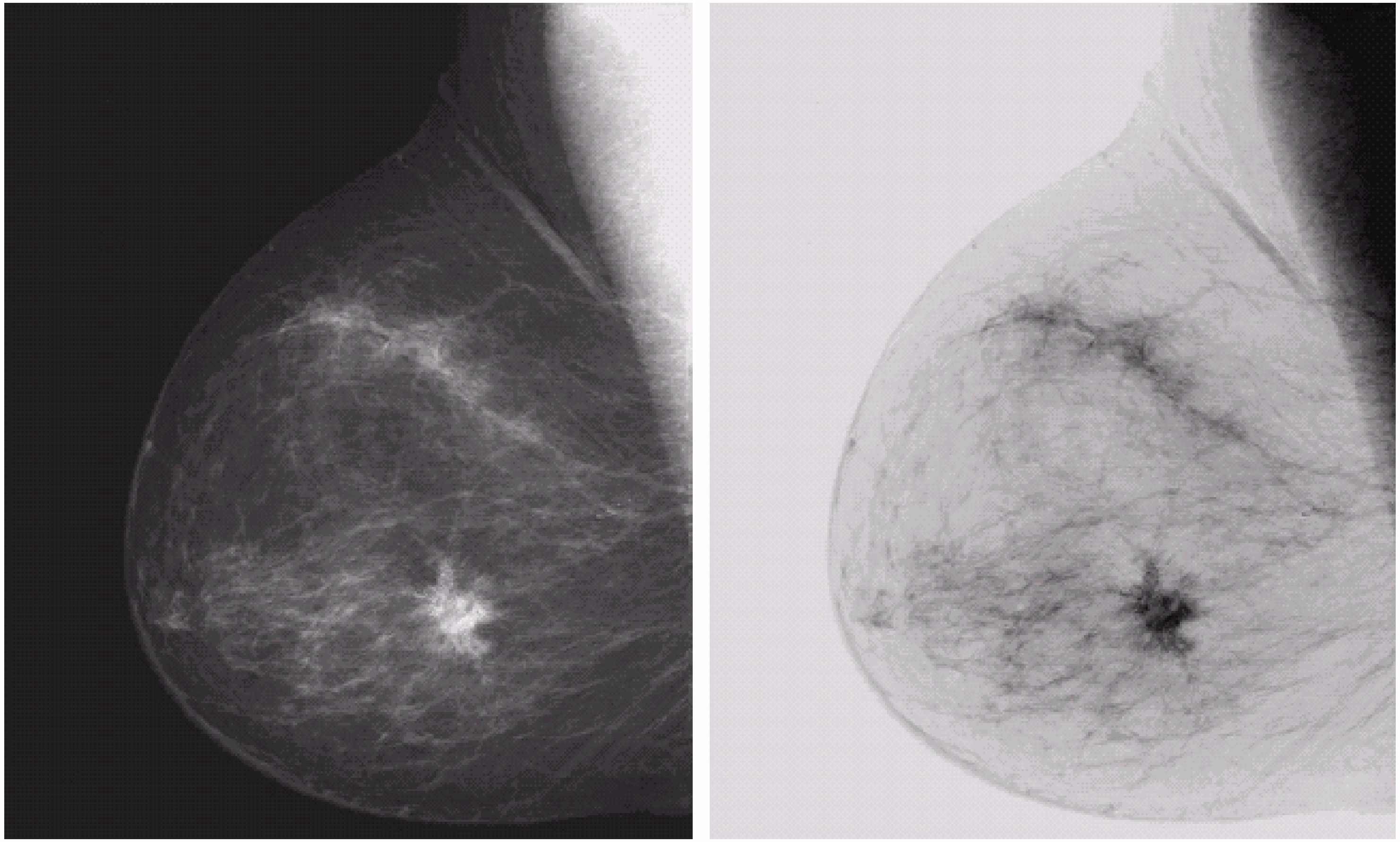


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



3.2 Basic gray level transformations



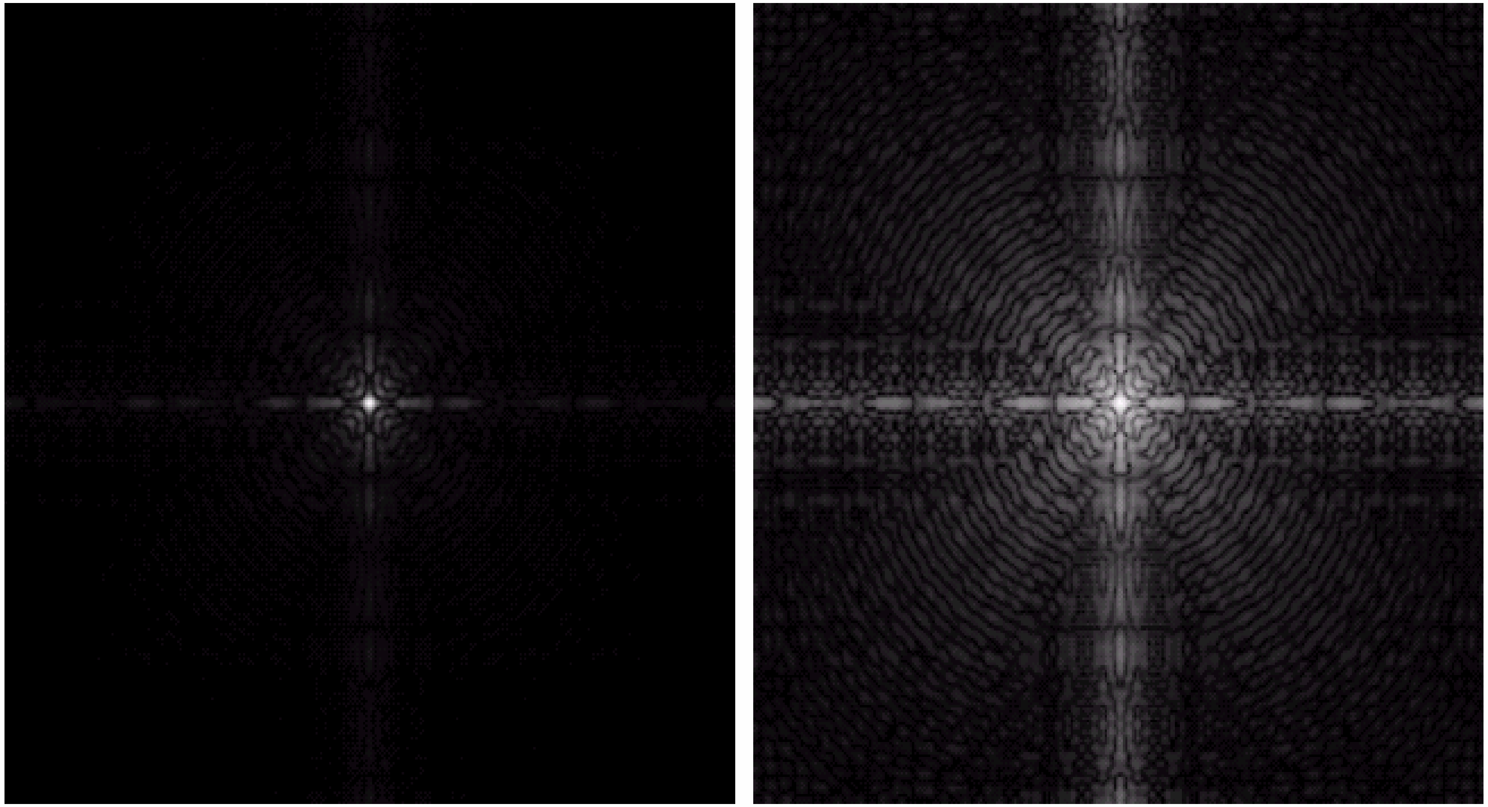
a b

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.)

3.2 Basic gray level transformations

a b

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



3.2 Basic gray level transformations

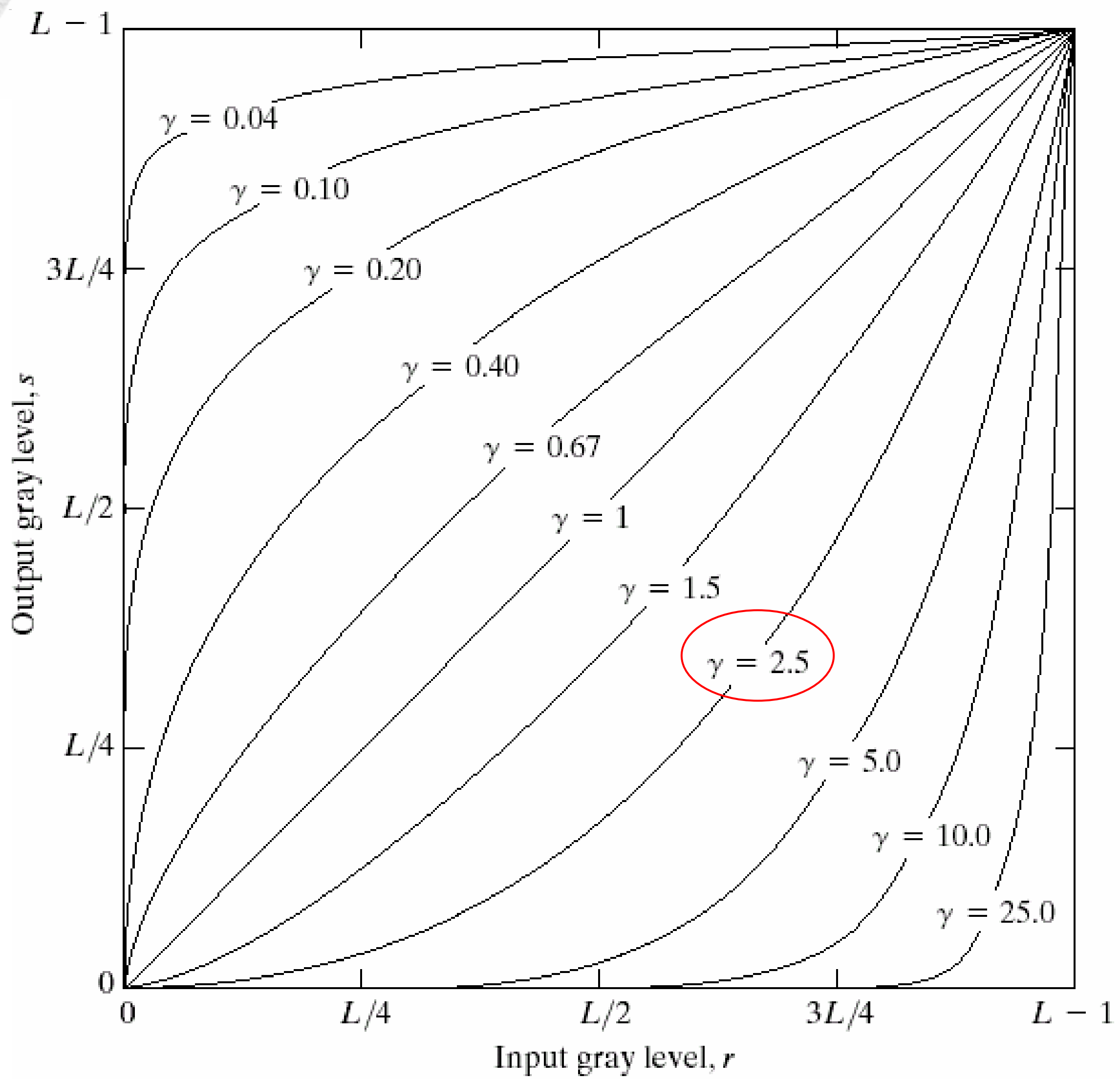
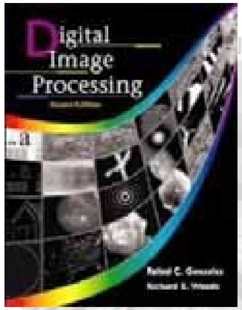


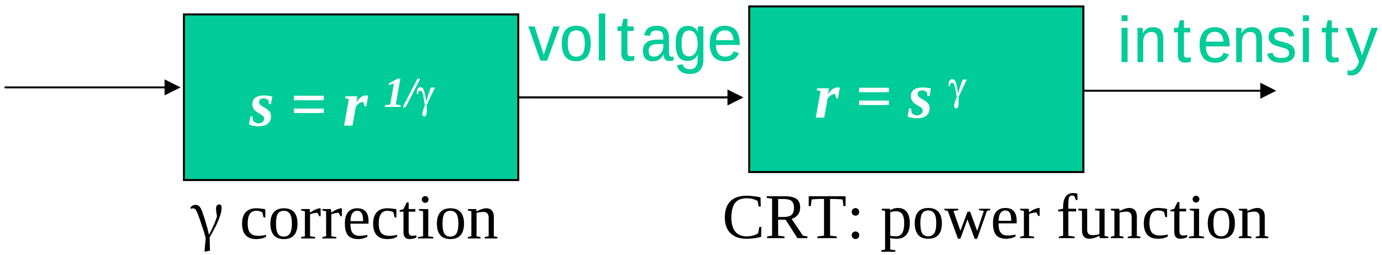
FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

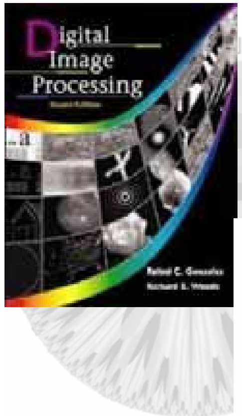


3.2 Basic gray level transformations

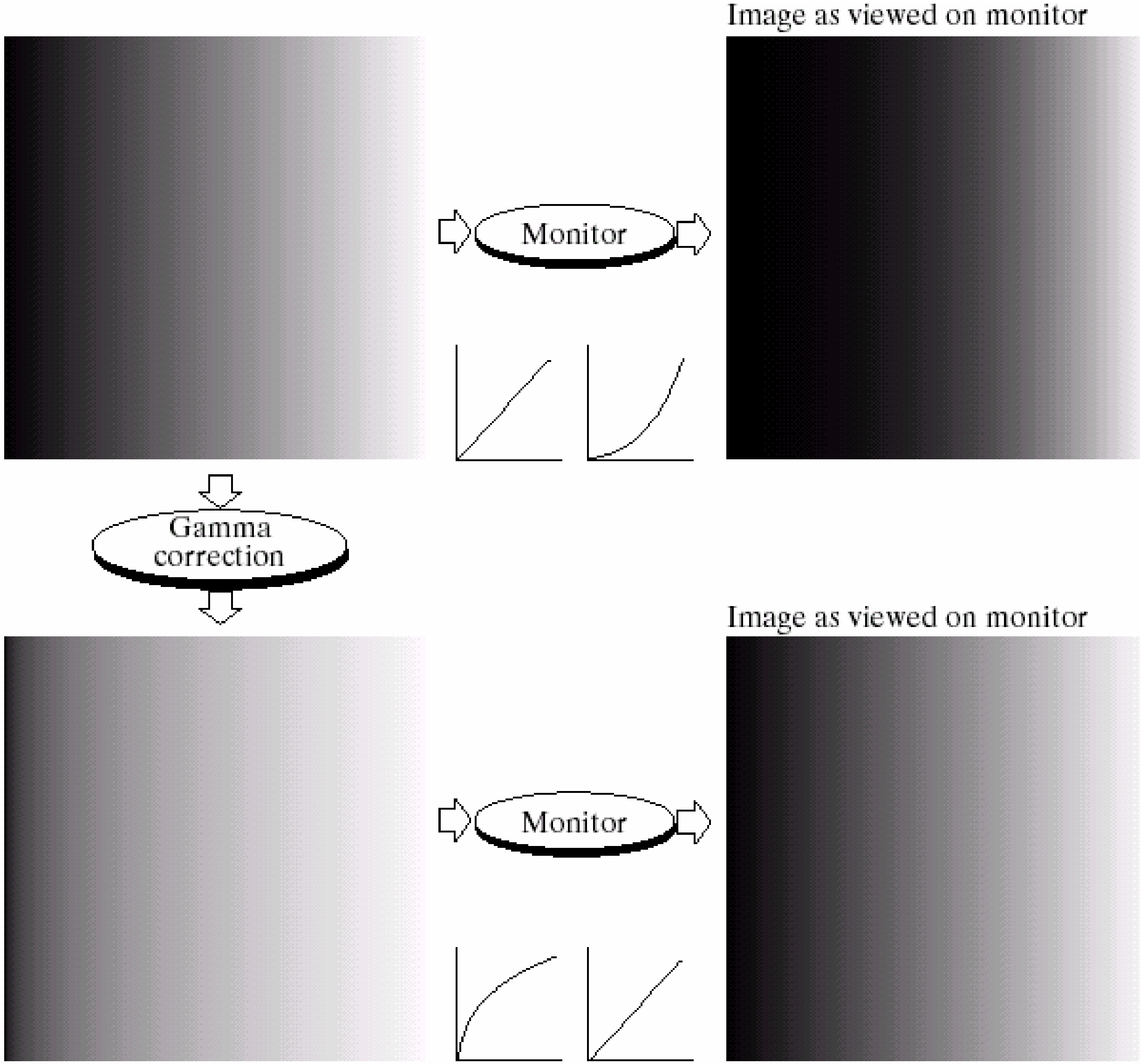
- γ correction
  - The CRT devices have an intensity-to-voltage response which is a ***power function***.
  - γ ranges from 1.8 to 2.5
  - Without γ correction, the monitor output will become darker than the original input
  - Prepare the input image before inputting it into the CRT monitor by performing the transforming

$$s = r^{1/\gamma} = r^{1/2.5} = r^{0.4}$$



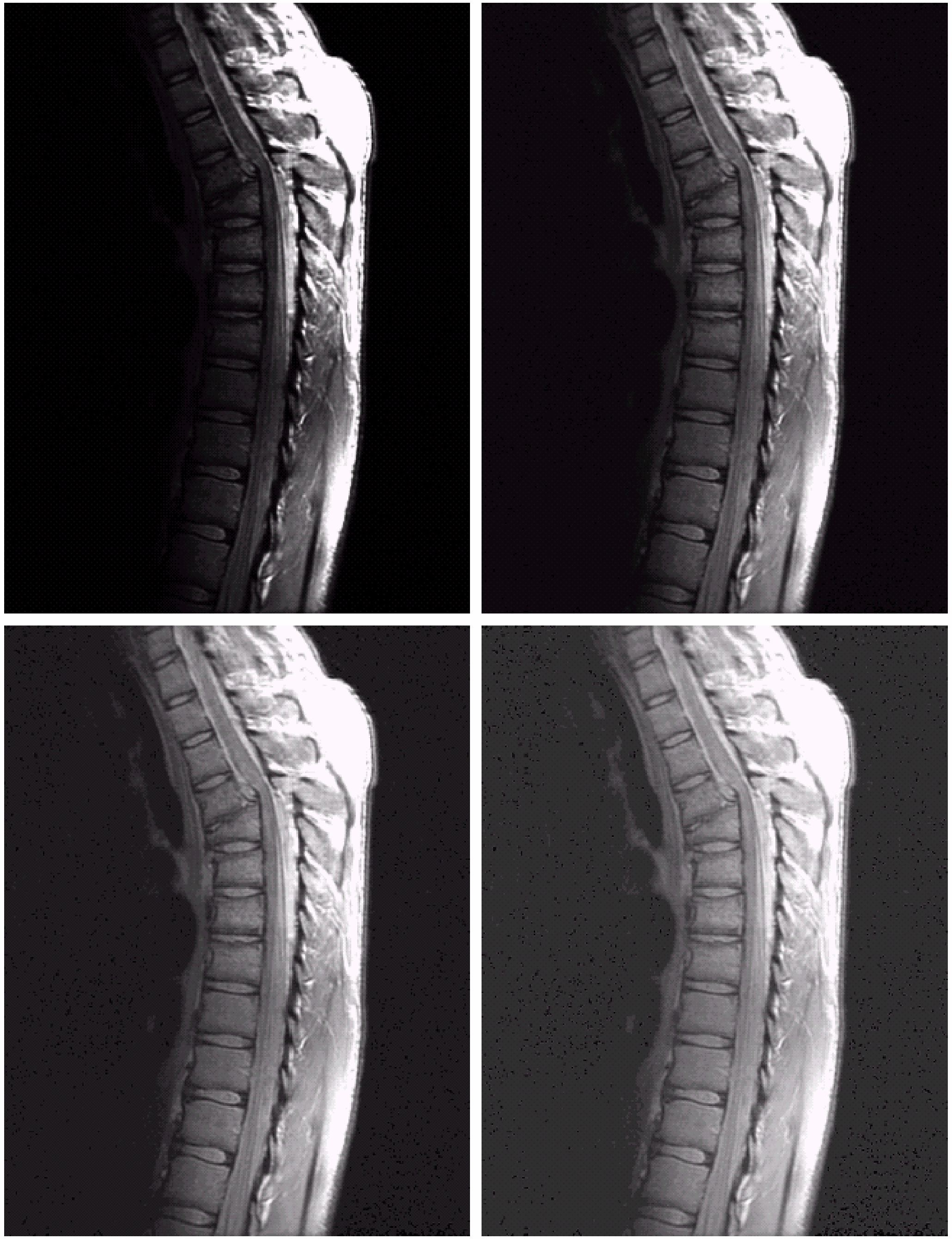


3.2 Basic gray level transformations



a b
c d

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

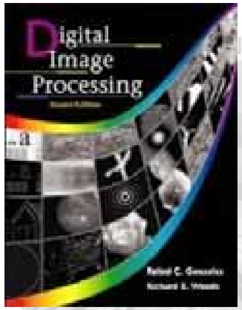


a b
c d

FIGURE 3.8 (a) Magnetic resonance (MR) image 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 for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

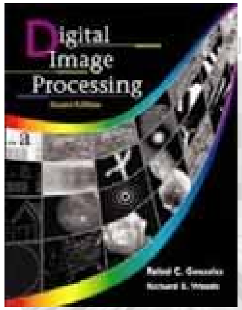
a b
c d

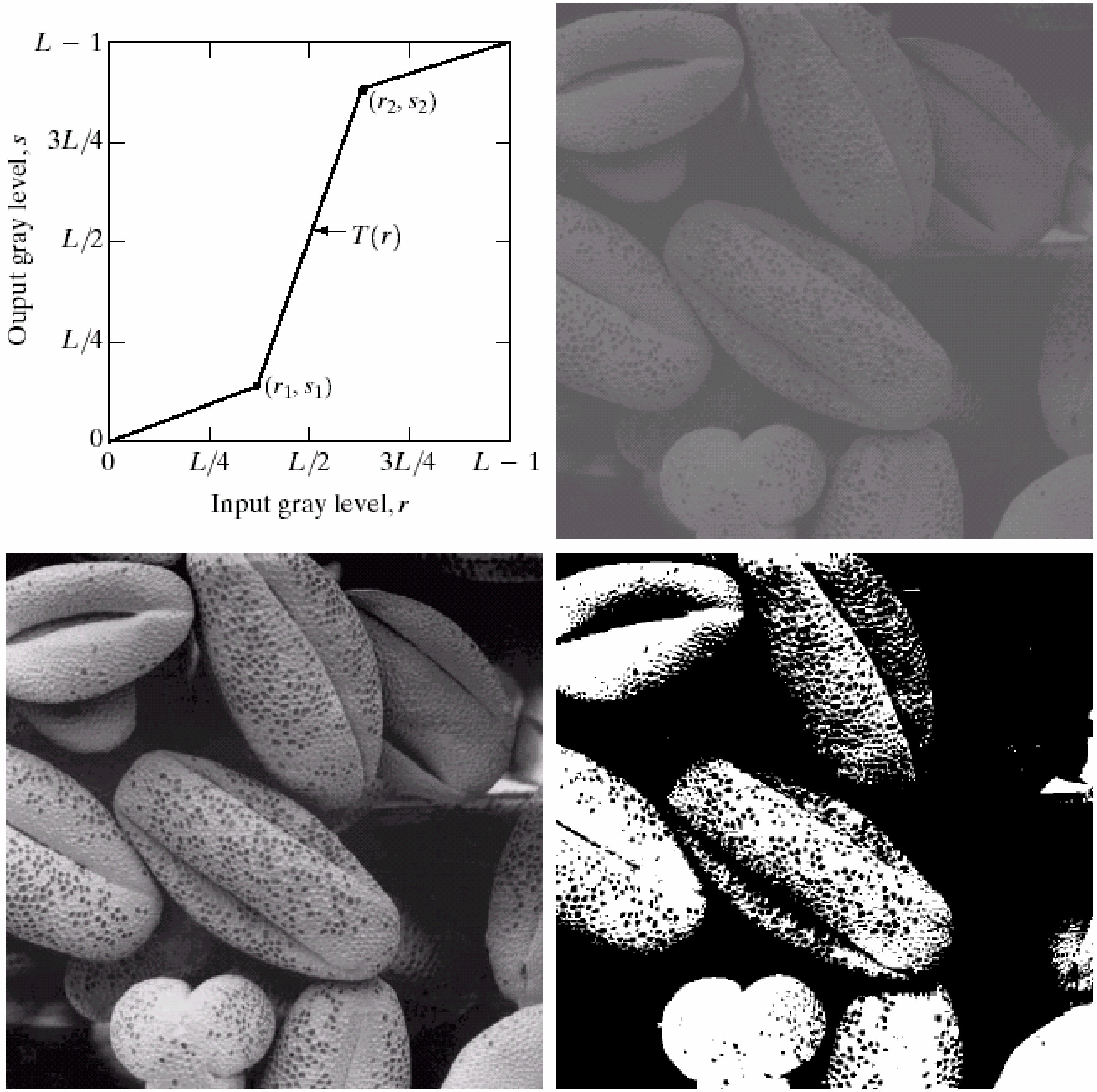
FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)



3.2 Basic gray level transformations

- Piecewise-Linear Transformation Functions
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing

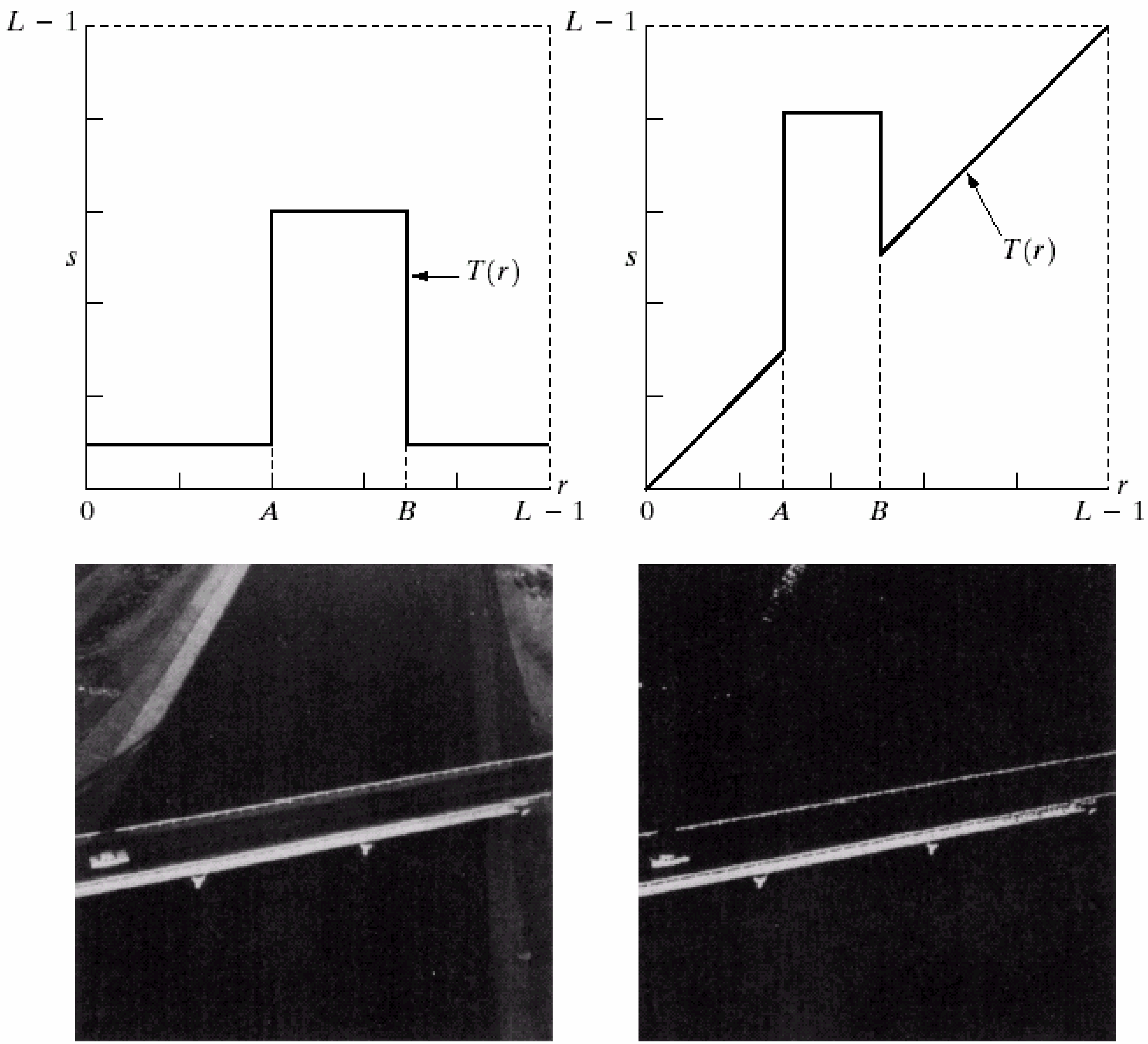




a b
c d

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.)

3.2 Basic gray level transformations



a b
c d

FIGURE 3.11 (a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

3.2 Basic gray level transformations

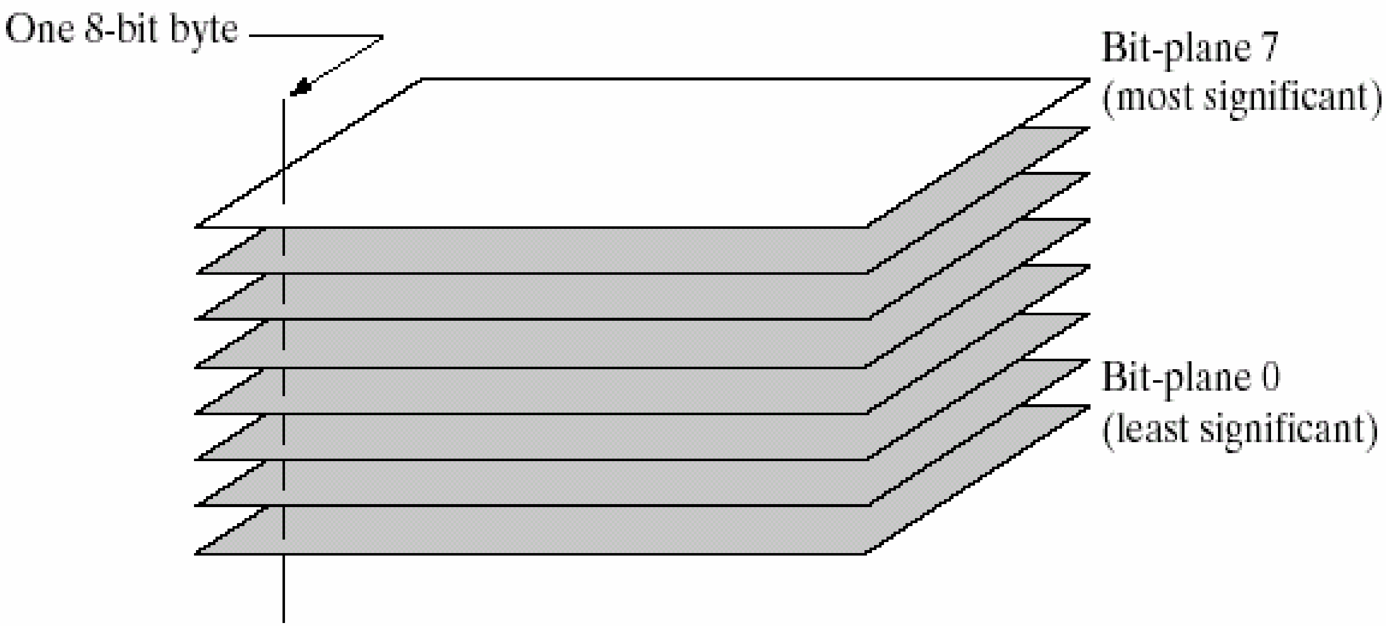


FIGURE 3.12 Bit-plane representation of an 8-bit image.

3.2 Basic gray level transformations

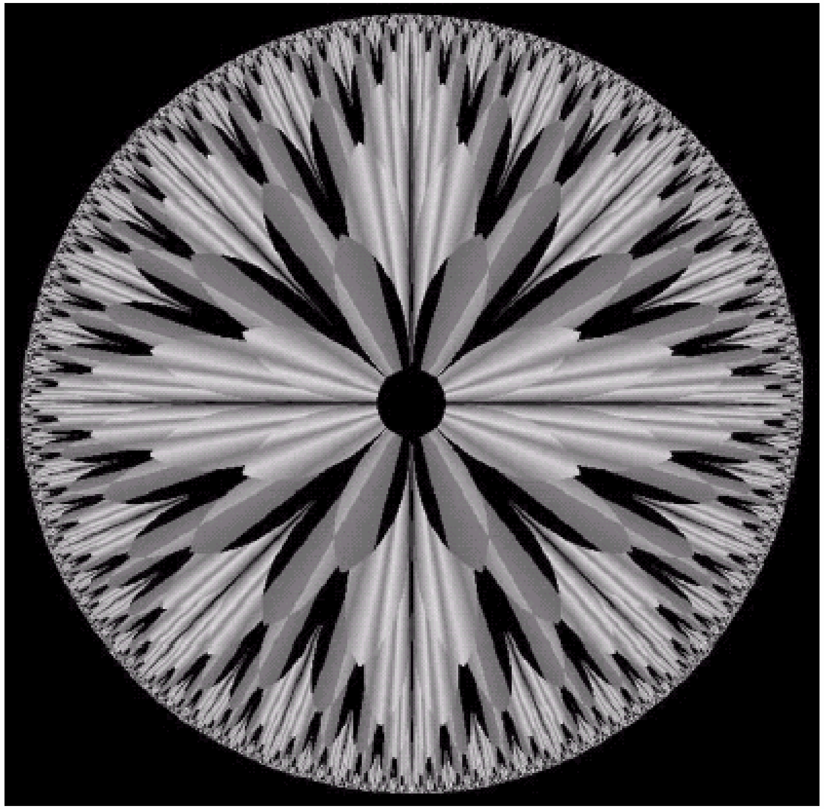


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

3.2 Basic gray level transformations

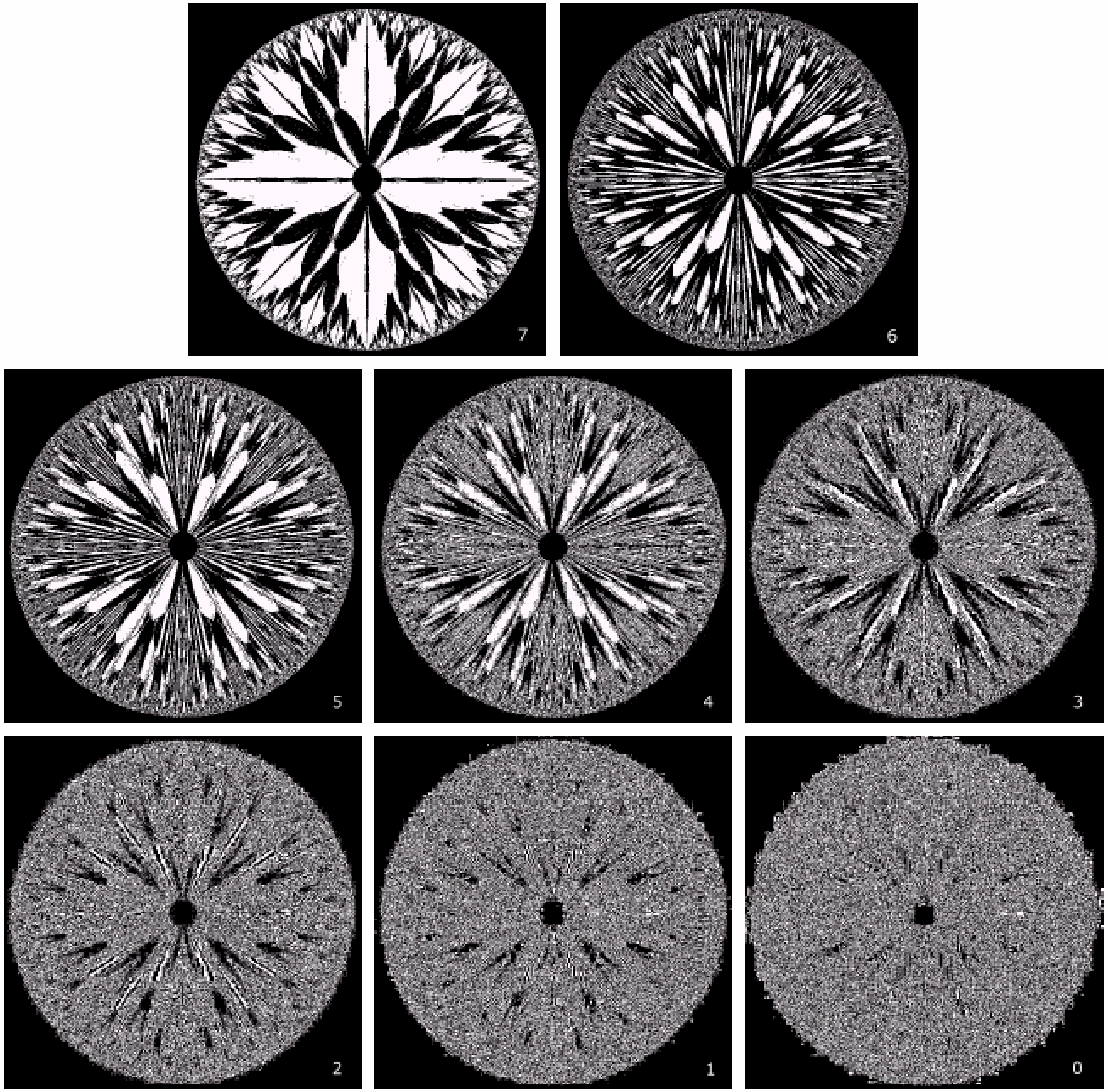
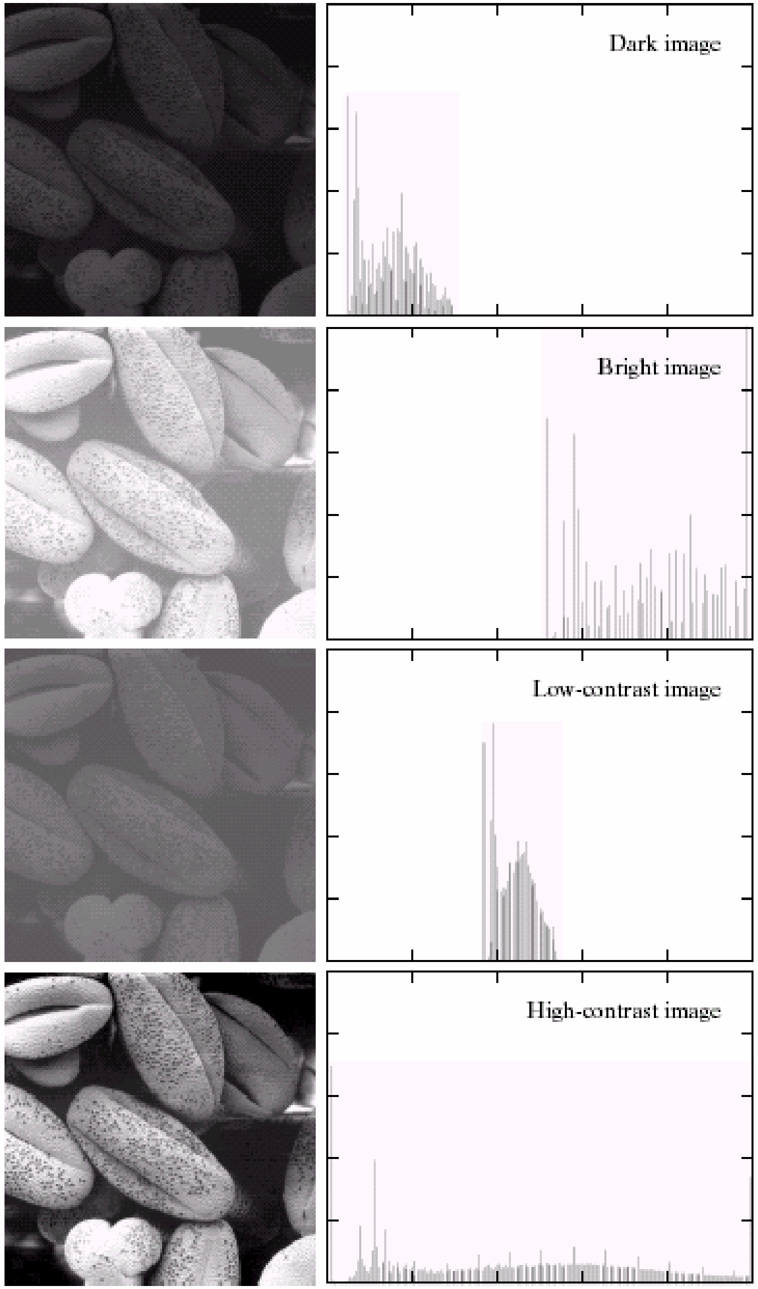


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

3.3 Histogram Processing

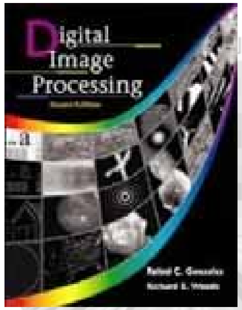
- The histogram of a digital image with gray-levels in the range [0, L-1] is a discrete function $h(r_k)=n_k$ where r_k is the *kth* level and n_k is the number of pixels having the gray-level r_k.
- A normalized histogram $h(r_k)=n_k/n$, n is the total number of pixels in the image.

3.3 Histogram Processing



a b

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

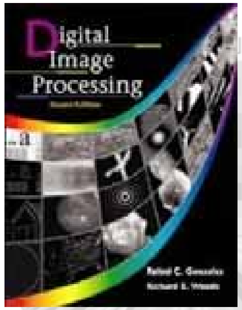


3.3 Histogram Processing

- Histogram equalization is to find a transformation s=T(r) $0 \le r \le 1$ that satisfying the following conditions:
  - T(r) is single-valued and monotonically increasing in the interval $0 \le r \le 1$
  - $0 \le T(r) \le 1$ for $0 \le r \le 1$
  - The T(r) is single-valued so that its inverse function exists.
  - The inverse transform from s to r is denoted as

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

    The inverse function may not be single valued, $s = Q^{-1}(r)$ may not exist



3.3 Histogram Processing

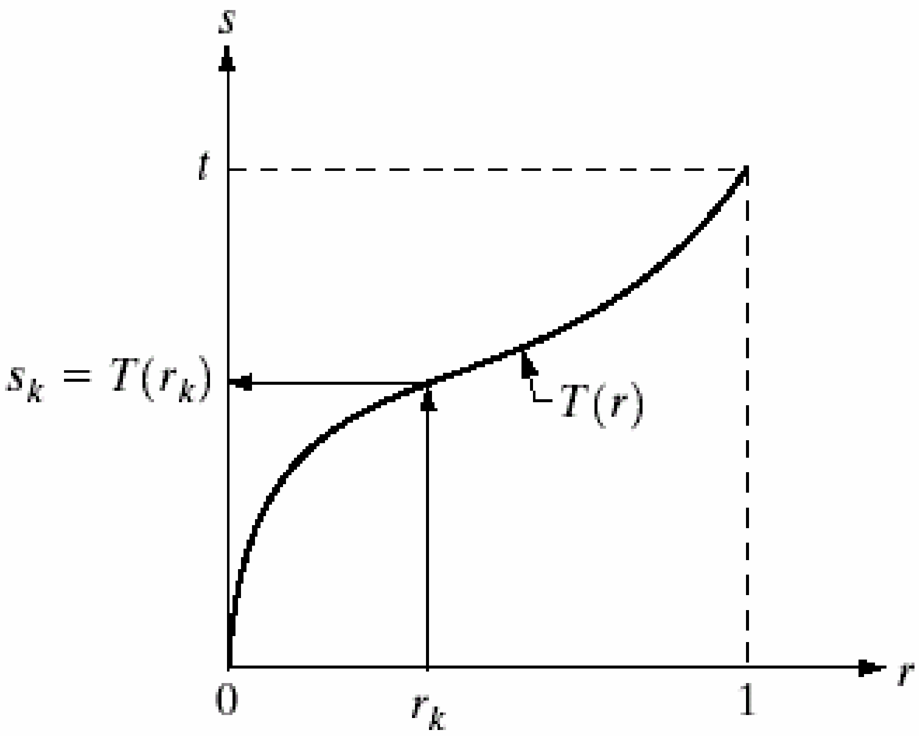
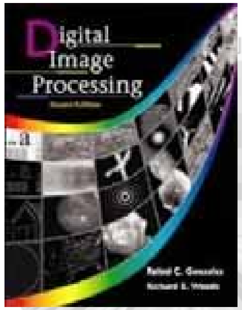


FIGURE 3.16 A gray-level transformation function that is both single valued and monotonically increasing.



3.3 Histogram Processing

- The gray-level is an image may be viewed as a random variable, so we let $p_r(r)$ and $p_s(s)$ denote the probability density function of random variables *r* and *s*.
- If $p_r(r)$ and $T(r)$ are known and $T^{-1}(s)$ is single-valued and monotonically increased function then

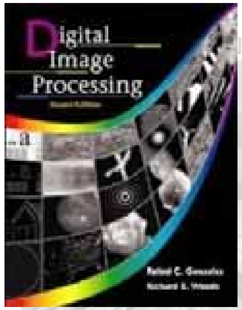
  $$p_s(s) = p_r(r)|dr/ds|$$

- If we assume the inverse transformation function as

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

  where *w* is a dummy variable,

  $s=T(r)$ is a ***cumulative distribution function*** (CDF) of the random variable *r*.



3.3 Histogram Processing

- Given transformation T(r), we may find the $p_s(s)$ as

$$ds/dr = dT(r)/dr = d\left[\int_0^r p_r(w)dw\right]/dr = p_r(r)$$

and then $p_s(s) = p_r(r)|dr/ds| = 1$ for $0 \le s \le 1$

- $p_s(s)$ is a ***uniform prob. distribution***
- T(r) depends on $p_r(r)$

3.3 Histogram Processing

- For discrete case

$$p_r(r_k)=n_k/n \quad \text{for } k=0,1\ldots.L-1$$

- The discrete version of the transformation function is

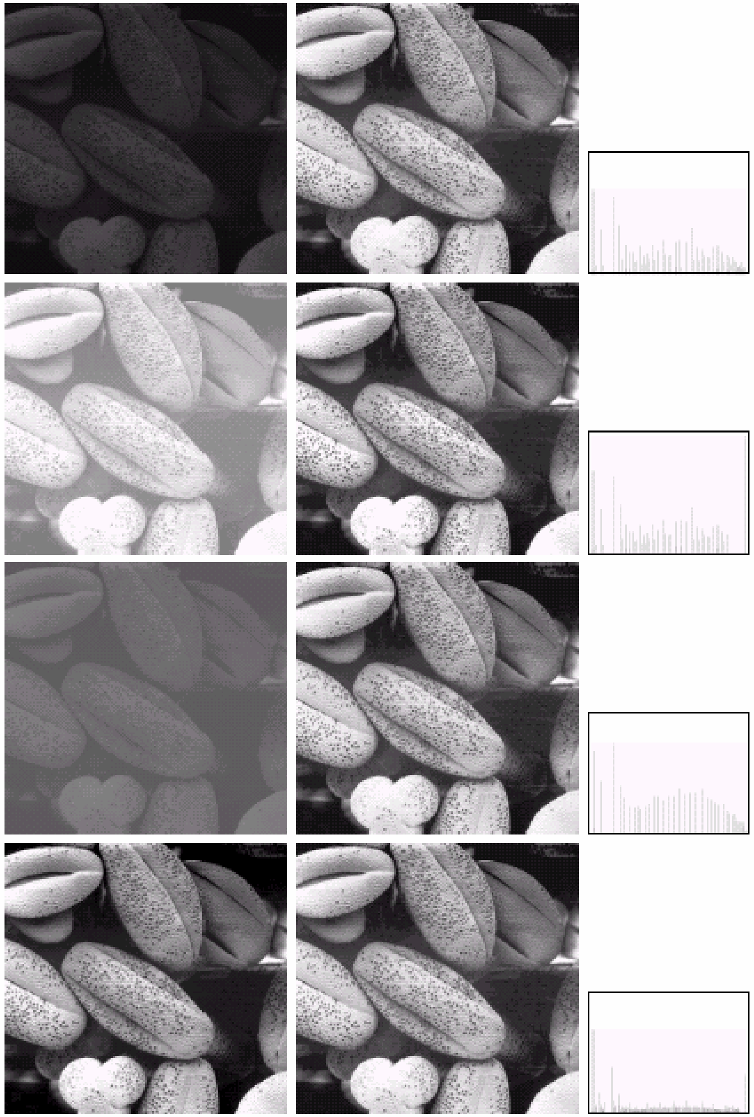
$$s_k = T(r_k) = \sum_{j=0}^{k} p(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$$

- The above mapping is called ***histogram equalization***
- The inverse transform

$$r_k=T^{-1}(s_k) \quad \text{for } k=0,1,\ldots L-1$$

It exists if none of the levels, r_k, $k=0,1,\ldots L-1$ are missing from the input images

3.3 Histogram Processing-histogram equalization

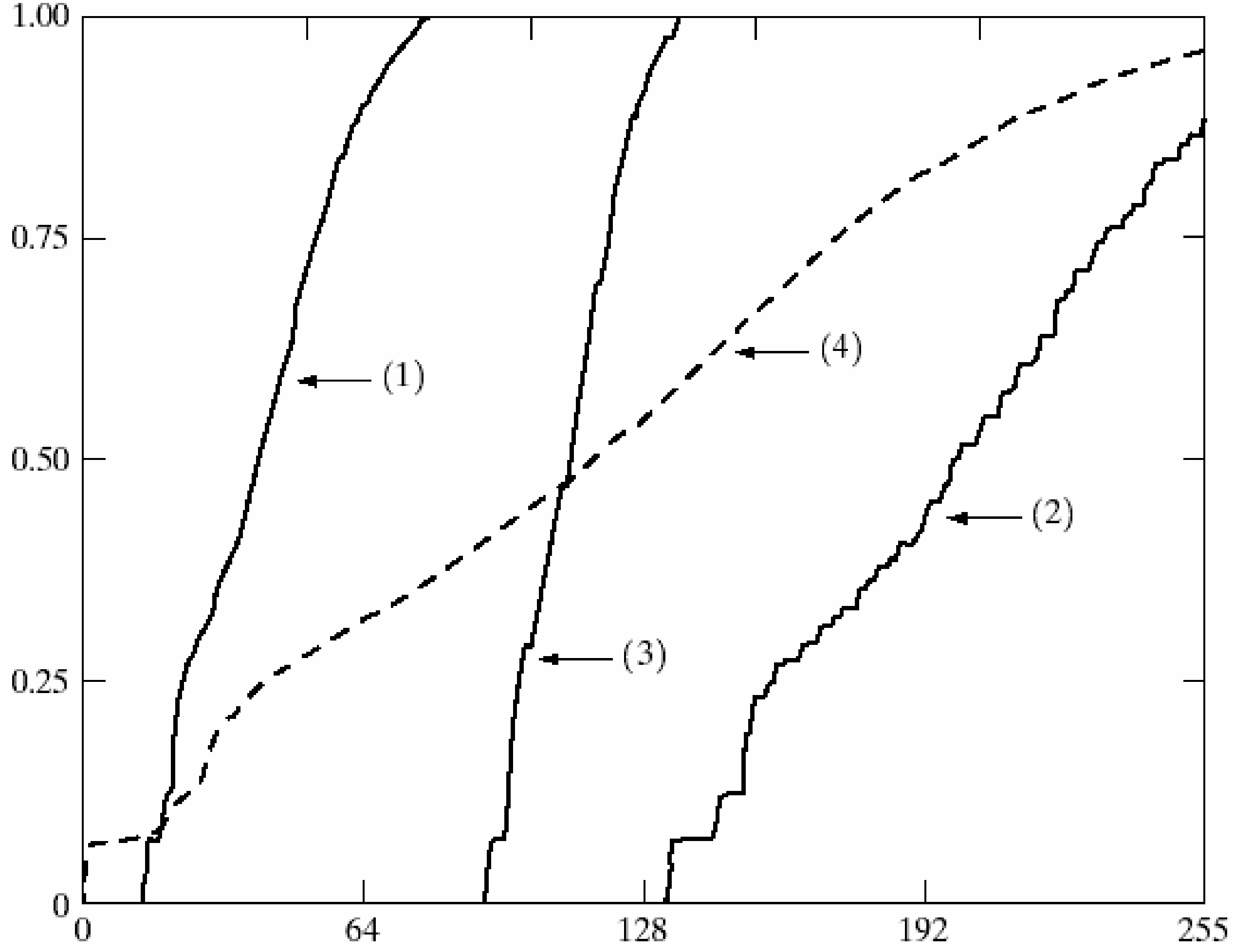


a b c

FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

3.3 Histogram Processing-histogram equalization

FIGURE 3.18 Transformation functions (1) through (4) were obtained from the histograms of the images in Fig.3.17(a), using Eq. (3.3-8).



3.3 Histogram Processing-histogram matching

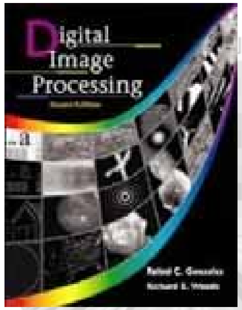
- Given the input image with $p_r(r)$, and the specific output image with $p_z(z)$, find the transfer function between the r and z.
- Let s=T(r)= $\int_0^r p_r(w)dw$

  where w is a dummy variable
- Define a random variable z with the property

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

  where t is a dummy variable
- From the above equations G(z)=T(r) we have

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



3.3 Histogram Processing-histogram matching

- **For discrete case:**
- From given histogram $p_r(r_k)$, k=0, 1,….L-1

  $$s_k=T(r_k)=\sum_{j=0}^{k} p(r_j)=\sum_{j=0}^{k}\frac{n_j}{n}$$
- From given histogram $p_z(z_i)$, i=0, 1,…L-1

  $$v_k=G(z_k)=\sum_{i=0}^{k} p_z(z_i)=s_k$$
- Finally, we have $G(z_k)=T(r_k)$

  and $z_k=G^{-1}(s_k)=G^{-1}[T(r_k)]$

Notes: $r\xrightarrow[T]{}s$ $\quad z\xrightarrow[G]{}v$ $\quad s=v$, hence $r\xrightarrow[T]{}s=v\xrightarrow[G^{-1}]{}z$

3.3 Histogram Processing-histogram matching

1. Obtain the histogram of each given image
2. Pre-compute a mapped s_k for each r_k, *i.e.*, $s_k = T(r_k)$
3. Obtain the transformation function G from given $p(z)$ using

   $$v_k = G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k$$

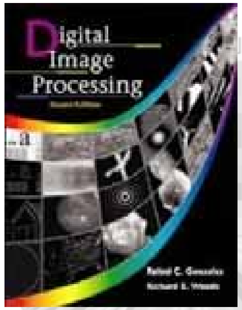
4. Precompute z_k for each value s_k using iterative scheme as follows:

   To find $z_k = G^{-1}(s_k) = G^{-1}(v_k)$, however, it may not exist such z_k.

   Since we are dealing with integer, we find the closest z_k. we can get to satisfy $G(z_k) - s_k = 0$

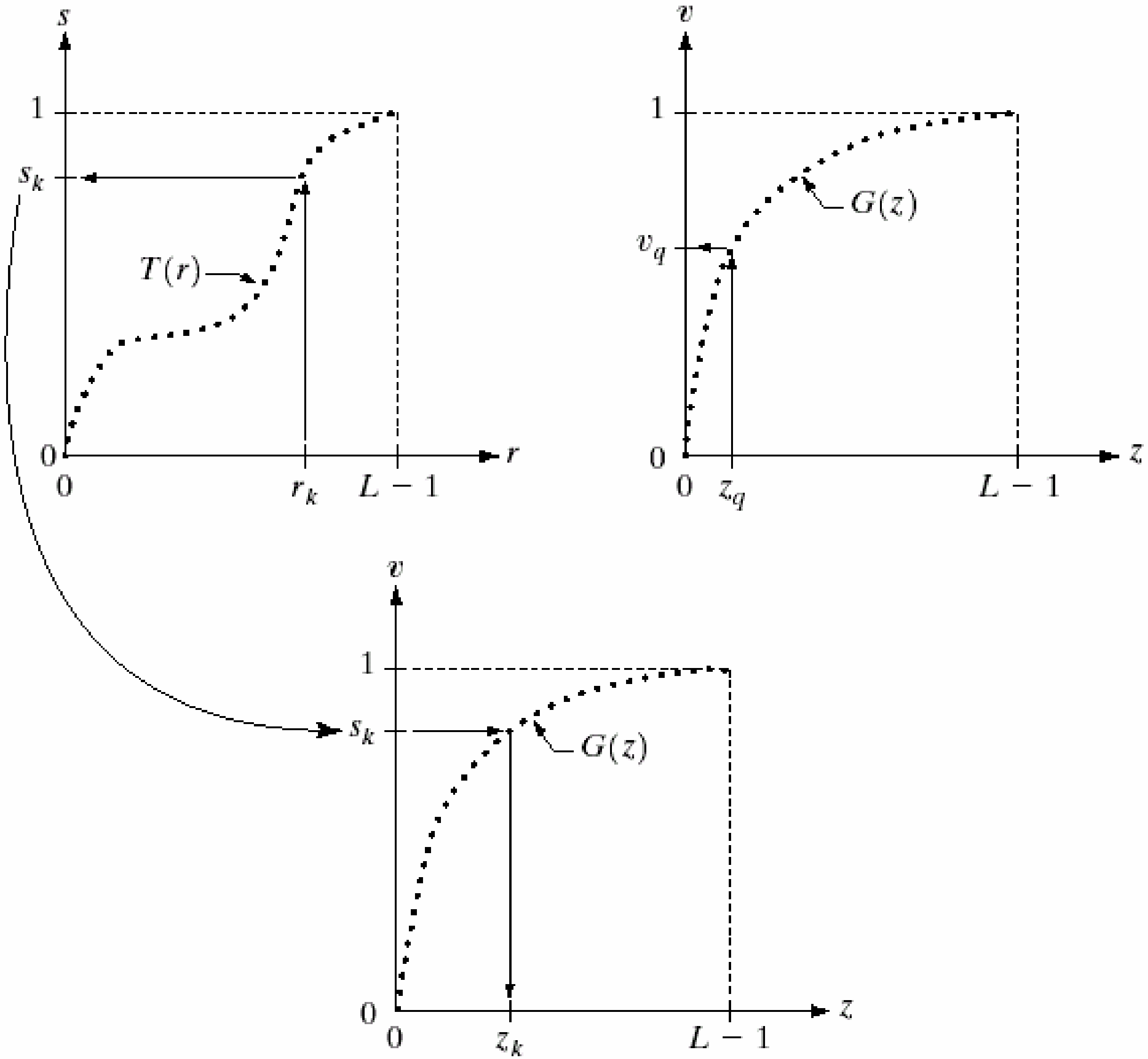
5. For each pixel in the original image, if the value of that pixel is r_k, map this value to its corresponding levels s_k; then map level s_k into the final level z_k.

3.3 Histogram Processing-histogram matching

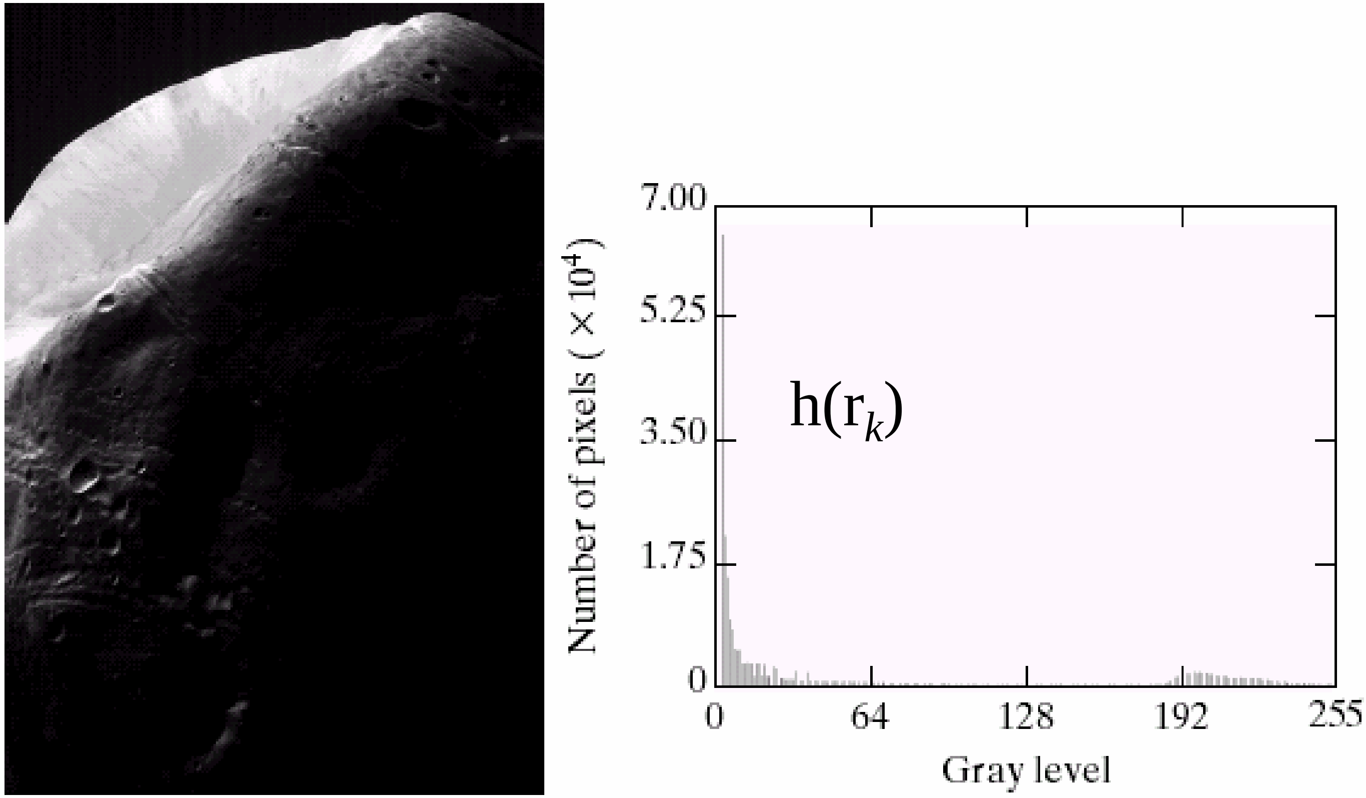


a b
c

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



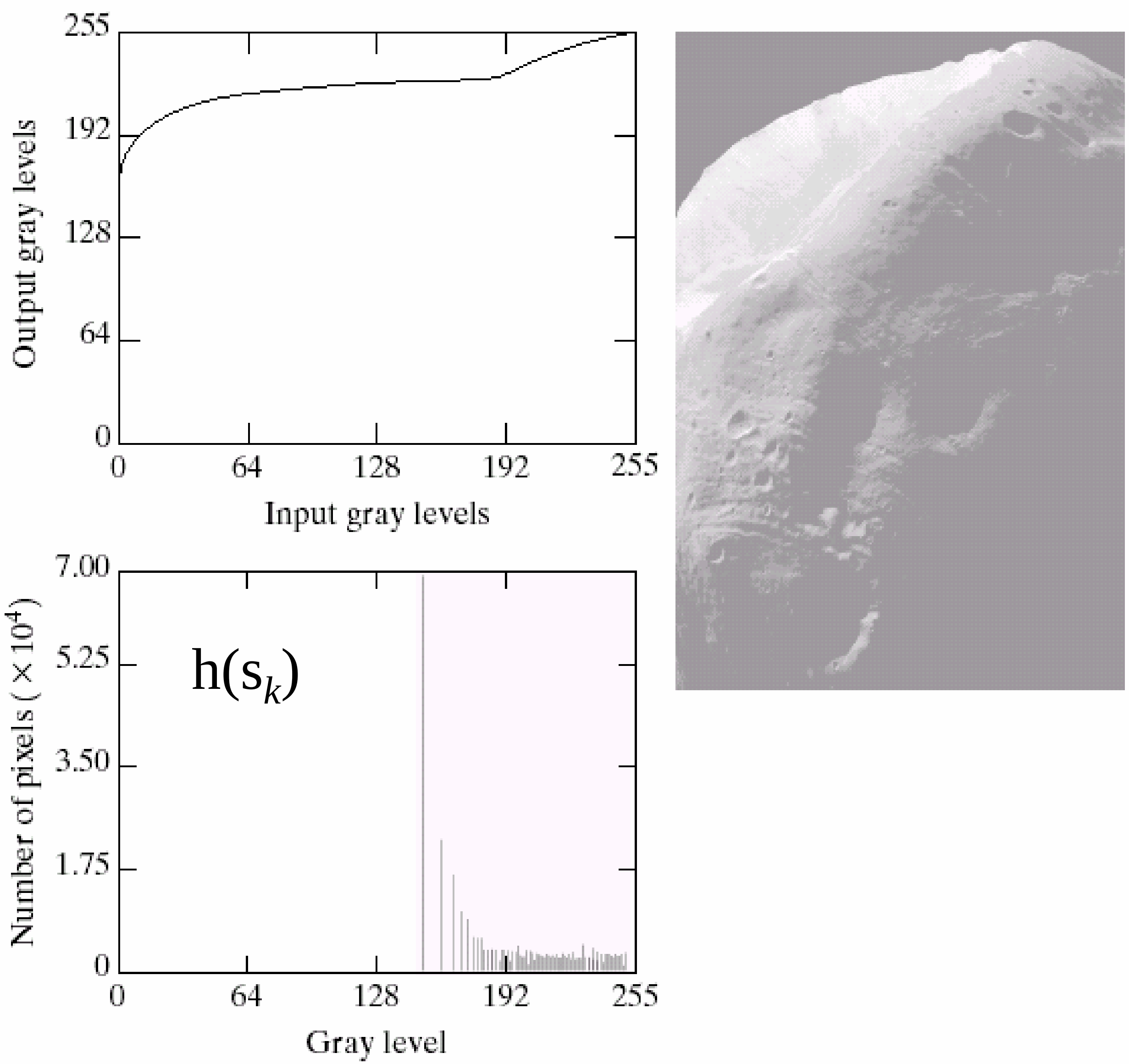
3.3 Histogram Processing-histogram matching



a b

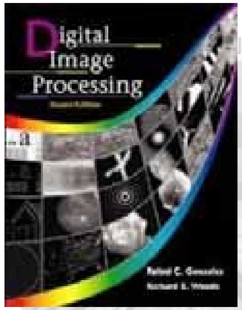
FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA's ***Mars Global Surveyor.*** (b) Histogram. (Original image courtesy of NASA.)

3.3 Histogram Processing-histogram matching



a b
c

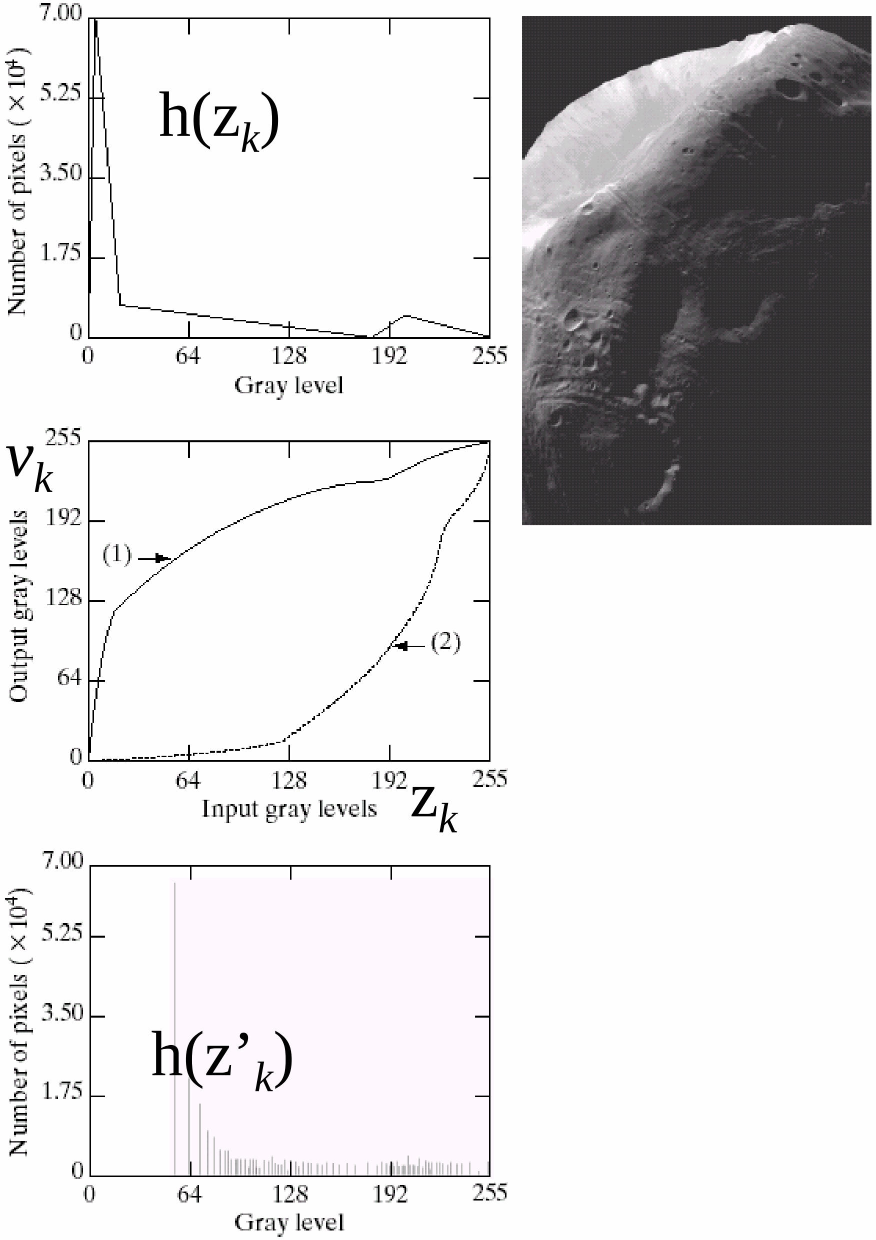
FIGURE 3.21
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).



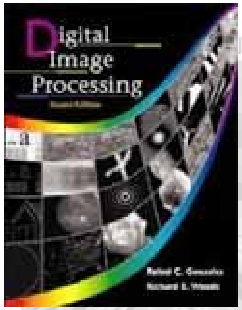
3.3 Histogram Processing-histogram matching

a c
b
d

FIGURE 3.22
(a) Specified histogram.
(b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17).
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

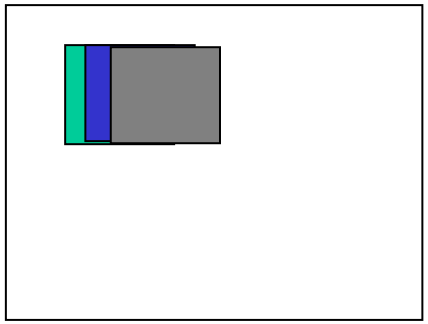


1. $r \xrightarrow{T} s \quad v \xrightarrow{G^{-1}} z \quad s=v$
2. *Curve 1:* $v_k = G(z_k) = s_k$
3. *Curve 2:* $z'_k = G^{-1}(s_k)$

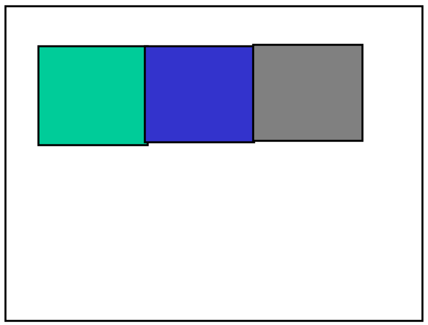


3.3.3 Local Enhancement

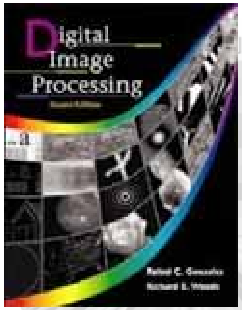
- Transformation function based on ***gray-level distribution*** in the ***neighborhood*** of every pixel in the image.
- At each location, the histogram of points in the neighborhood (or inside a ***region***) is computed and histogram equalization is applied.
- Enhancement applied for ***overlapped regions*** is better than ***non-overlapped regions***.



Over-lapped

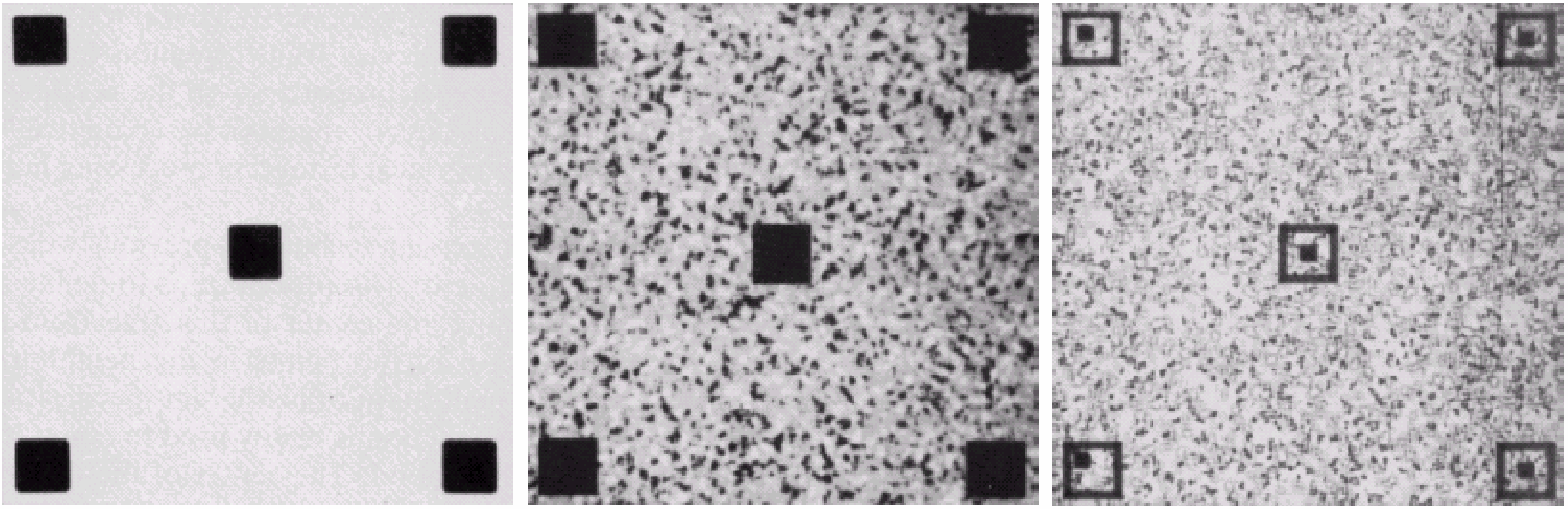


Non-Overlapped



3.3.3 Histogram Processing-local enhancement

Devise transformation functions based on the gray-level histogram distribution in the neighborhood of every pixel in the image.



a b c

FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

3.3.4 Histogram Processing-Image Enhancement Using Histogram Statistics

- Let $p(r_i)$ be an estimate of the probability of occurrence of gray-level r_i.
- The nth moment of r about it mean is

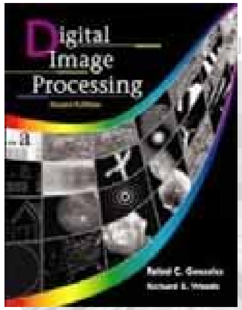
$$\mu_n(\mathrm{r}) = \sum_{i=0}^{L-1} (r_i - m)^n \, p(r_i)$$

- where m is the mean value of r

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

- The second moment is given by

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



3.3.4 Histogram Processing-Image Enhancement Using Histogram Statistics

- Let S_{xy} denote a neighborhood of specified size centered at (x, y), the mean value $m_{S_{xy}}$ of the pixel in S_{xy} is

$$m_{S_{xy}} = \sum_{s,t \in S_{xy}} r_{s,t}\, p(r_{s,t})$$

where $r_{s,t}$ is the gray level at (s, t)

The gray level variance is

$$\sigma_{S_{xy}} = \sum_{s,t \in S_{xy}} (r_{s,t} - m_{S_{xy}})^2\, p(r_{s,y})$$

3.3.4 Histogram Processing-Image Enhancement Using Histogram Statistics

- ***Example SEM image*** (Fig. 3.24): To enhance the ***dark areas*** while leaving the ***light area*** as unchanged.
- Consider the pixel at (x, y) as a candidate for enhancement by measuring whether an area is relatively light or dark by comparing the ***local average gray level*** $m_{S_{xy}}$ to the ***global mean*** M_G.

$$m_{S_{xy}} < K_0 M_G \quad \text{where} \quad 0 < K_0 < 1$$

and the ***local standard deviation*** $\sigma_{S_{xy}}$ to the global ***standard deviation*** σ_G.

$$\sigma_{S_{xy}} < K_2 \sigma_G \quad \text{where } K_2 > 0.$$

Set $K_2 > 1$ for *light area* and $K_2 < 1$ for *dark area.*

3.3.4 Histogram Processing-Image Enhancement Using Histogram Statistics

- Finally, we set the lower limit on the ***local standard deviation*** $\sigma_{S_{xy}}$ as $K_1\sigma_G < \sigma_{S_{xy}}$ with $K_1 < K_2$

Summary

- The original image $f(x, y)$ and enhanced image $g(x, y)$.

$$g(x, y) = E \cdot f(x, y) \text{ if } m_{S_{xy}} < K_0 M_G \text{ AND}$$

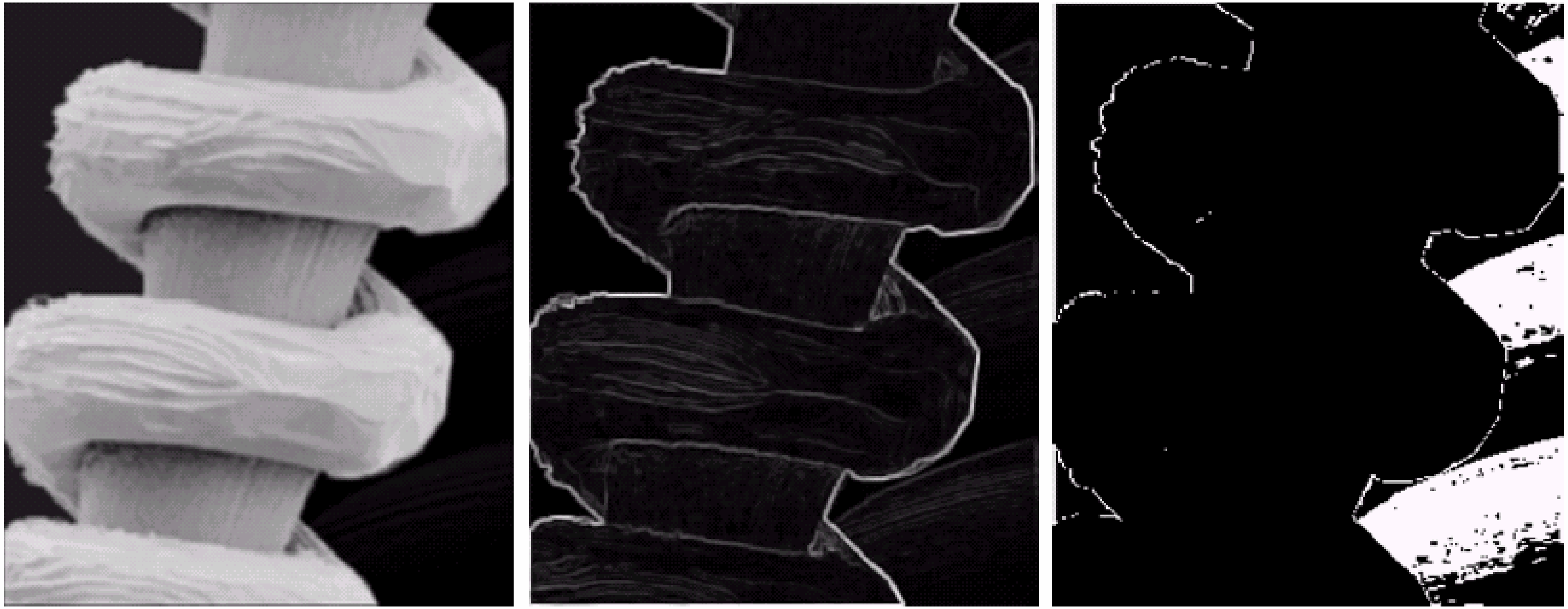
$$K_1\sigma_G < \sigma_{S_{xy}} < K_2\,\sigma_G \text{ with } K_1 < K_2$$

$$g(x, y) = f(x, y) \text{ otherwise}$$

3.3.4 Histogram Processing-Image Enhancement Using Histogram Statistics

FIGURE 3.24 SEM image of a tungsten filament and support, magnified approximately 130×. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene).

3.3.4 Histogram Processing-Image Enhancement Using Histogram Statistics



a b c

m_{Sxy} σ_{Sxy} $Ef(x,y)$

FIGURE 3.25 (a) Image formed from all local means obtained from Fig. 3.24 using Eq. (3.3-21). (b) Image formed from all local standard deviations obtained from Fig. 3.24 using Eq. (3.3-22). (c) Image formed from all multiplication constants used to produce the enhanced image shown in Fig. 3.26.

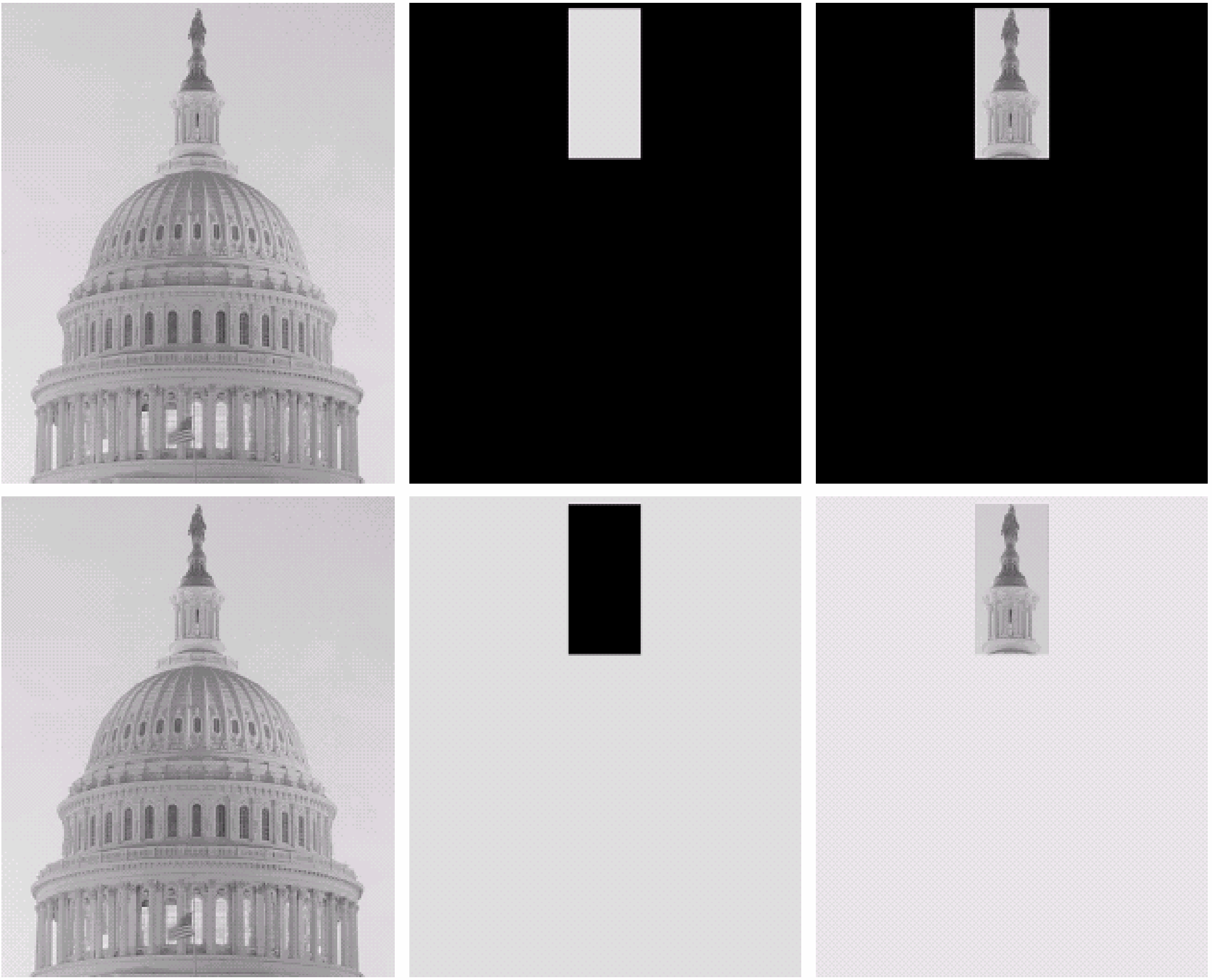
3.3.4 Histogram Processing-Image Enhancement Using Histogram Statistics

FIGURE 3.26 Enhanced SEM image. Compare with Fig. 3.24. Note in particular the enhanced area on the right side of the image.

3.4 Enhancement using Arithmetic/Logic Operation

- Operations on Pixel-by-pixel basis
- *AND* and *OR* operations are used for masking
  - Selecting subimages in an image
- Subtraction and Addition are used for image enhancement

3.4 Enhancement using Arithmetic/Logic Operation

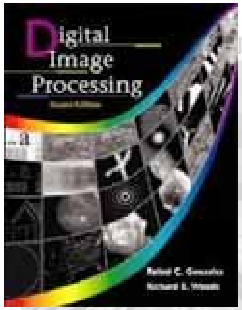


a	b	c
d	e	f

FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

3.4.1 Enhancement using Arithmetic Operation – Image subtraction

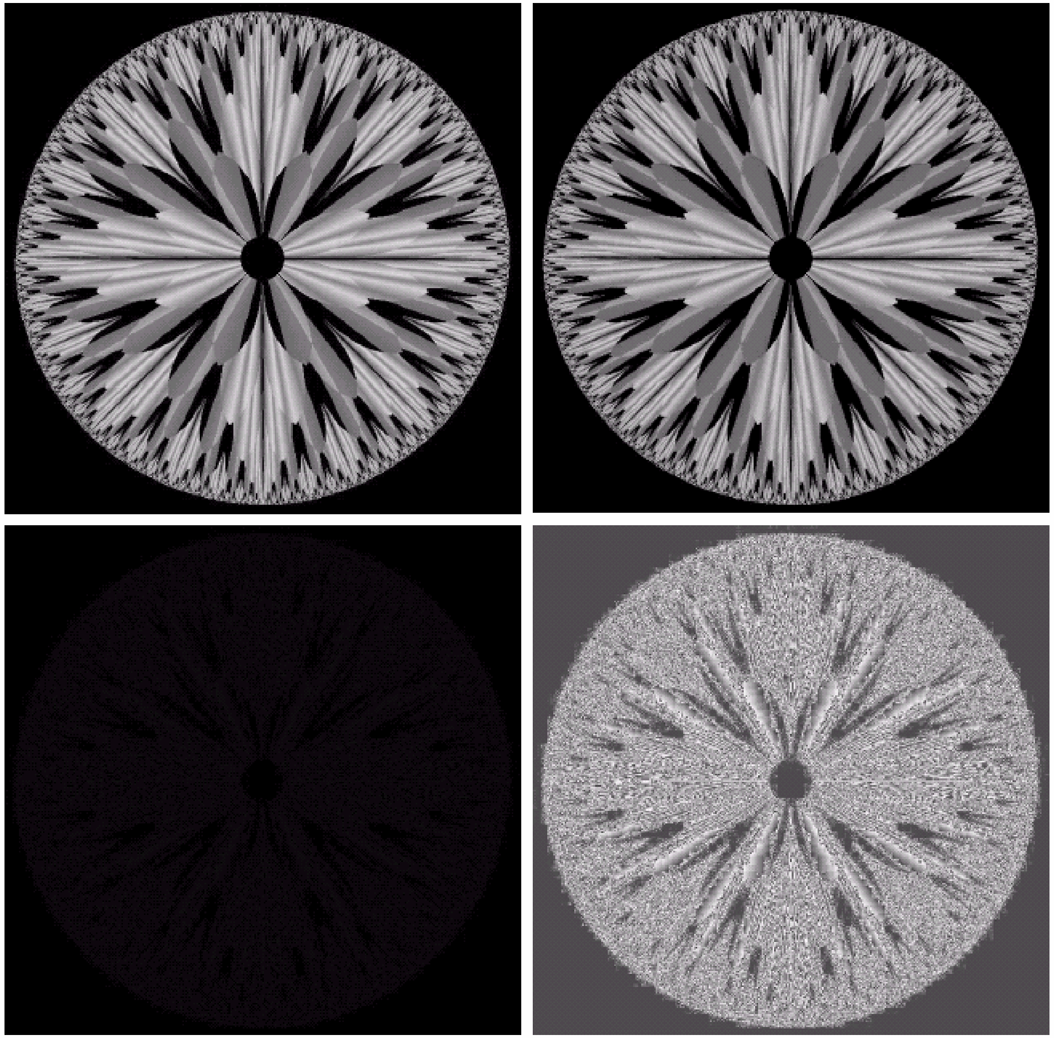
- $g(x, y) = f(x, y) - h(x, y)$
- *Higher-order bit plane – visual relevant detail*
- *Lower-order bit plane – fine detail or imperceptible detail*



3.4.1 Enhancement using Arithmetic Operation – Image subtraction

a b
c d

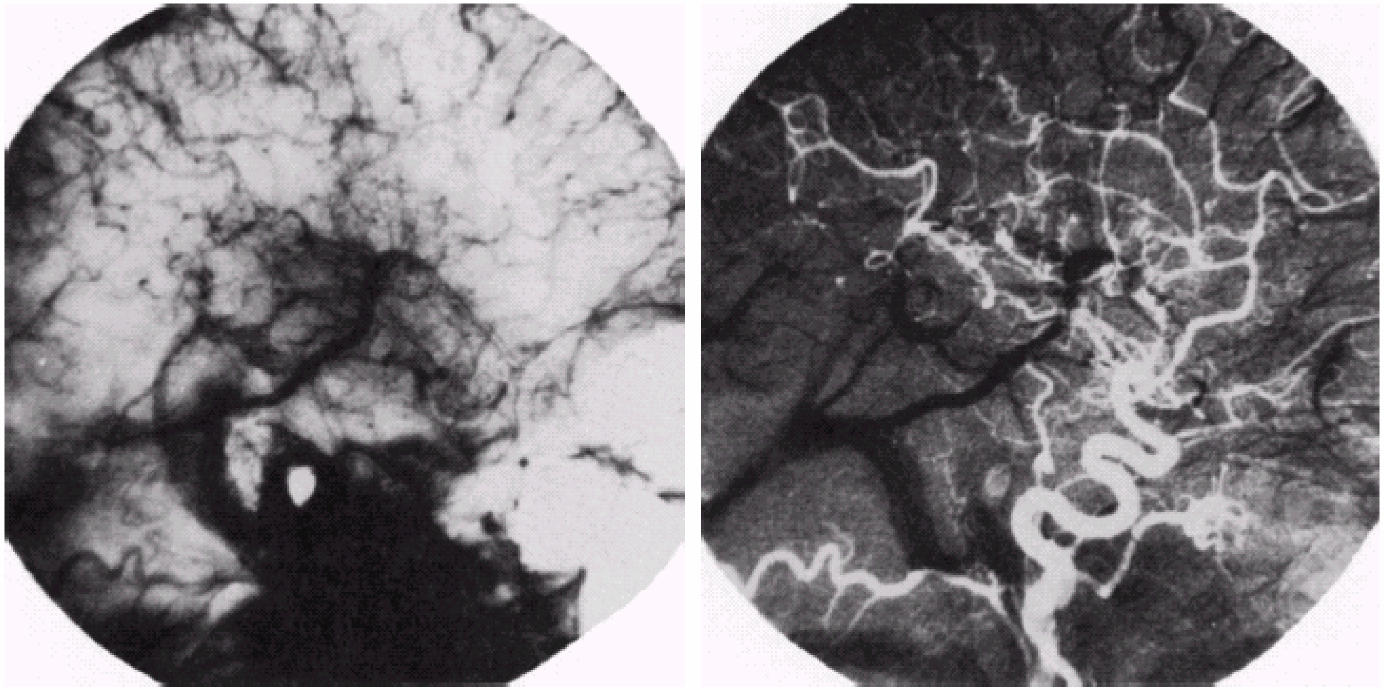
FIGURE 3.28 (a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).



3.4.1 Enhancement using Arithmetic Operation – Image subtraction

- ***Example***: Mask mode radiograph
  - h(x, y) is a ***mask***, or an X- ray image before injection
  - f(x, y) image after injection a contrast medium into bloodstream.
  - f(x,y) – h(x,y) enhanced detail
- Image scaling to interval of [0, 255]

3.4.1 Enhancement using Arithmetic Operation – image subtraction



a b

FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

$h(x, y)$ $f(x, y) - h(x, y)$

3.4.2 Enhancement using Arithmetic Operation – image averaging

- *Noisy image g(x,y) is* $g(x, y) = f(x, y) + \eta(x, y)$

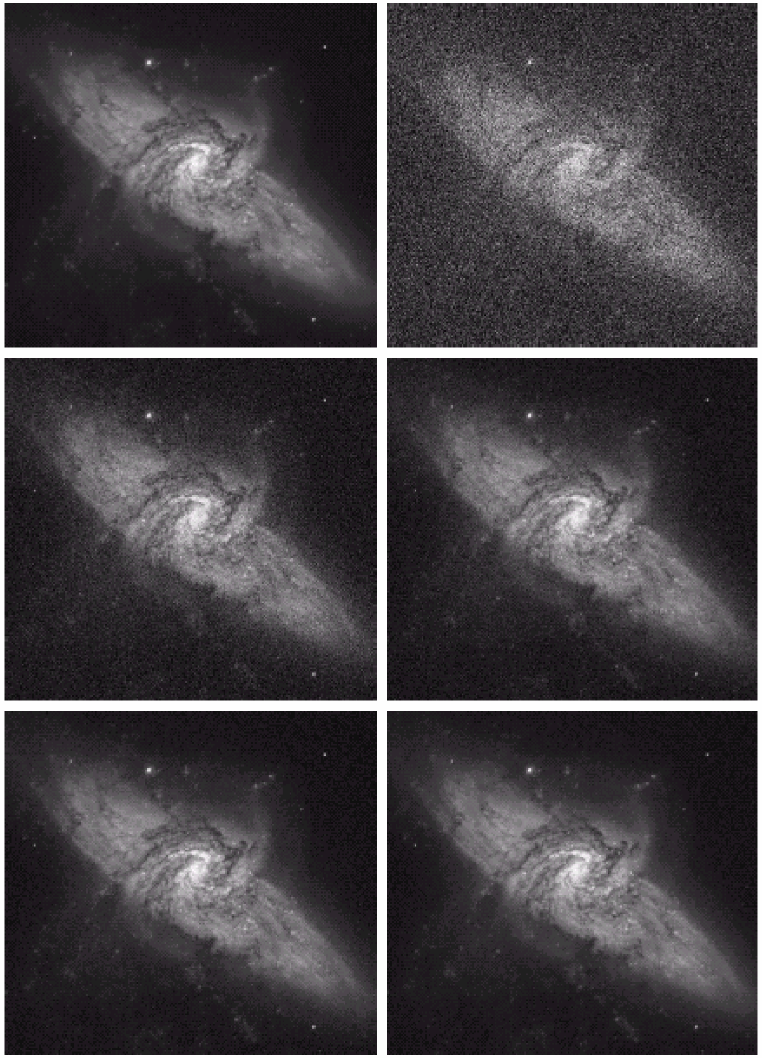
$$\bar{g}(x, y) = \frac{1}{k}\sum_{i=1}^{K} g_i(x, y)$$

$$E\{\bar{g}(x, y)\} = f(x, y)$$

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

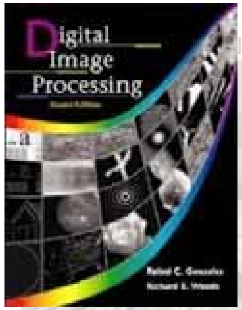
- As K increases, noise $\sigma^2_{\bar{g}(x,y)}$ decreases

3.4.2 Enhancement using Arithmetic Operation – image averaging

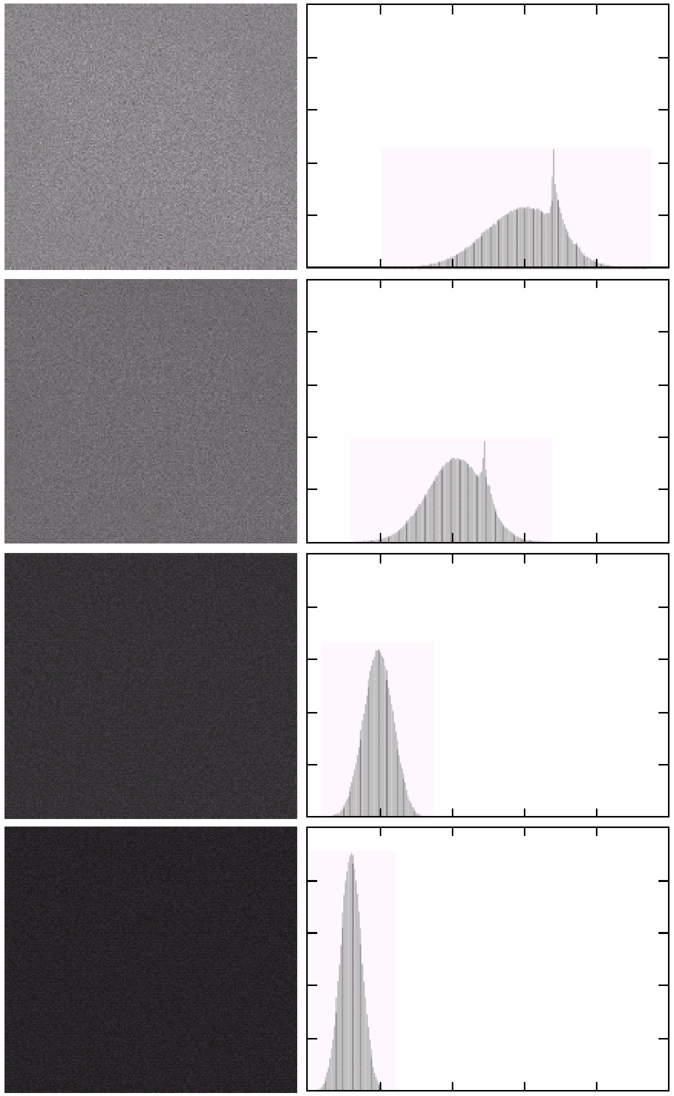


a b
c d
e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

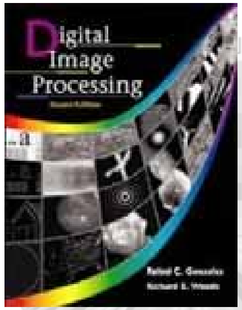


3.4.2 Enhancement using Arithmetic Operation – image averaging



a b

FIGURE 3.31 (a) From top to bottom: Difference images between Fig. 3.30(a) and the four images in Figs. 3.30(c) through (f), respectively. (b) Corresponding histograms.



3.5 Spatial Filtering

- ***Spatial filtering***: using a ***filter*** ***or*** ***kernel*** (*i.e.*, a ***subimage*** $w(x, y)$) to operate on the image $f(x,y)$.
- Filtering can also be applied in ***frequency domain*** (Chapter 4)
- The response R of the pixel at (x, y) after filtering is (Fig. 3.32 and 3.33)

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

  - The mask is centered at location (x, y)
  - Mask size is ***odd*** size (3x3, 5x5, 7x7,…)

3.5 Spatial Filtering

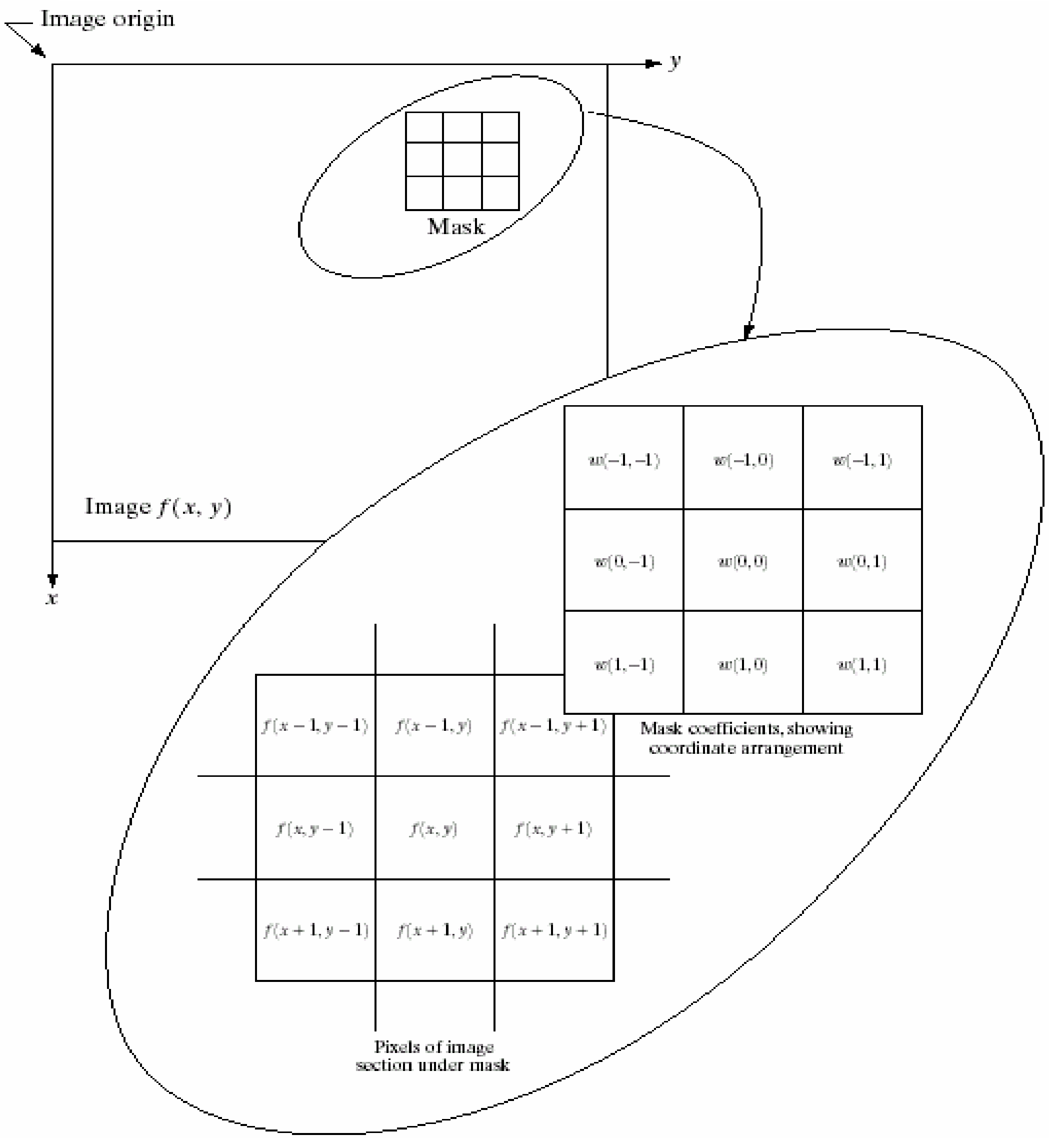


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

3.5 Spatial Filtering

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$R = w_1 z_1 + \ldots\ldots w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

3.6 Smoothing Spatial Filters

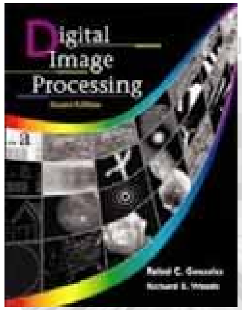
$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array} \qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

a b

FIGURE 3.34 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

Weighted Averaging

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



3.5 Spatial Filtering

- In general, linear filtering of an image *f* of size *MxN* with a filter mask of size *mxn* is given by

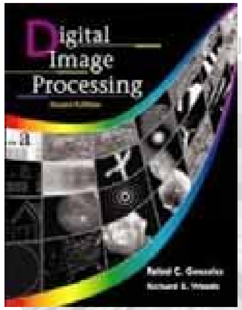
$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)f(x+s,y+t)$$

  where the output image is $g(x,y)$, $a=(m-1)/2$ *and* $b=(n-1)/2$, $x=0,\ldots M-1$ and $y=0,\ldots N-1$
- ***Linear filtering*** ⇔ ***convolution***
- ***Filter mask*** ⇔ ***convolution mask***
- When ***filter approach the border of the image***

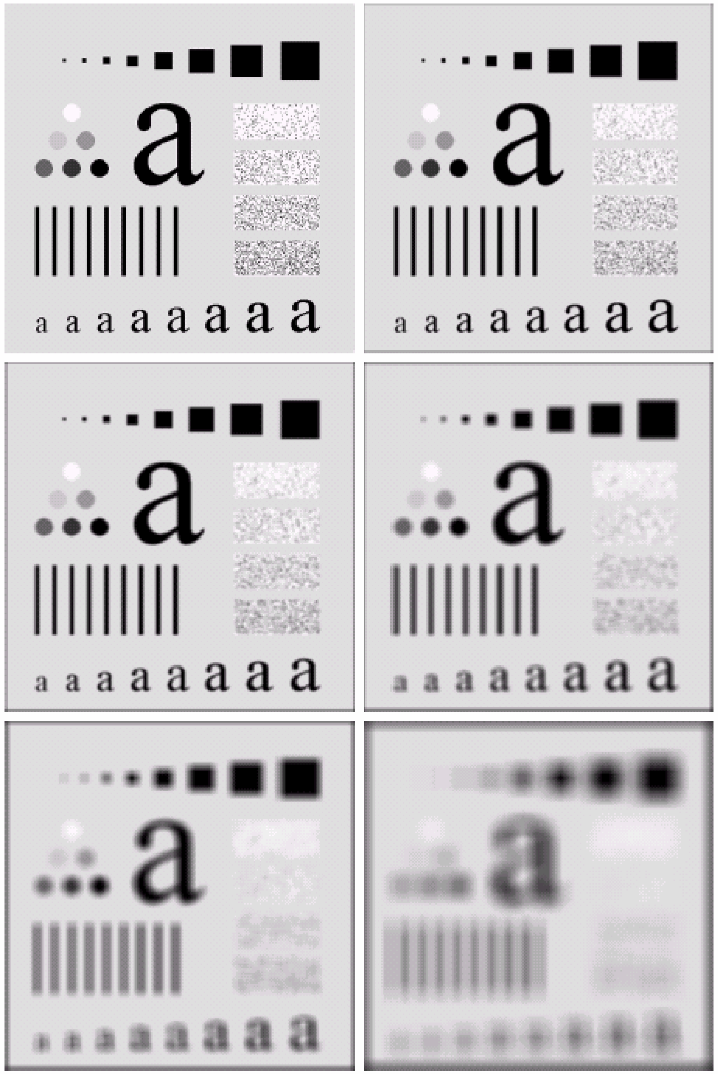
  1. Limit the excursions of the center of the mask to be at a distance no less than $(n-1)/2$ pixels from the border.

  2. "Padding" the image by adding rows and columns of 0's (or other value)

  3. "Padding" by replicating rows and columns.



3.6 Smoothing Spatial Filters

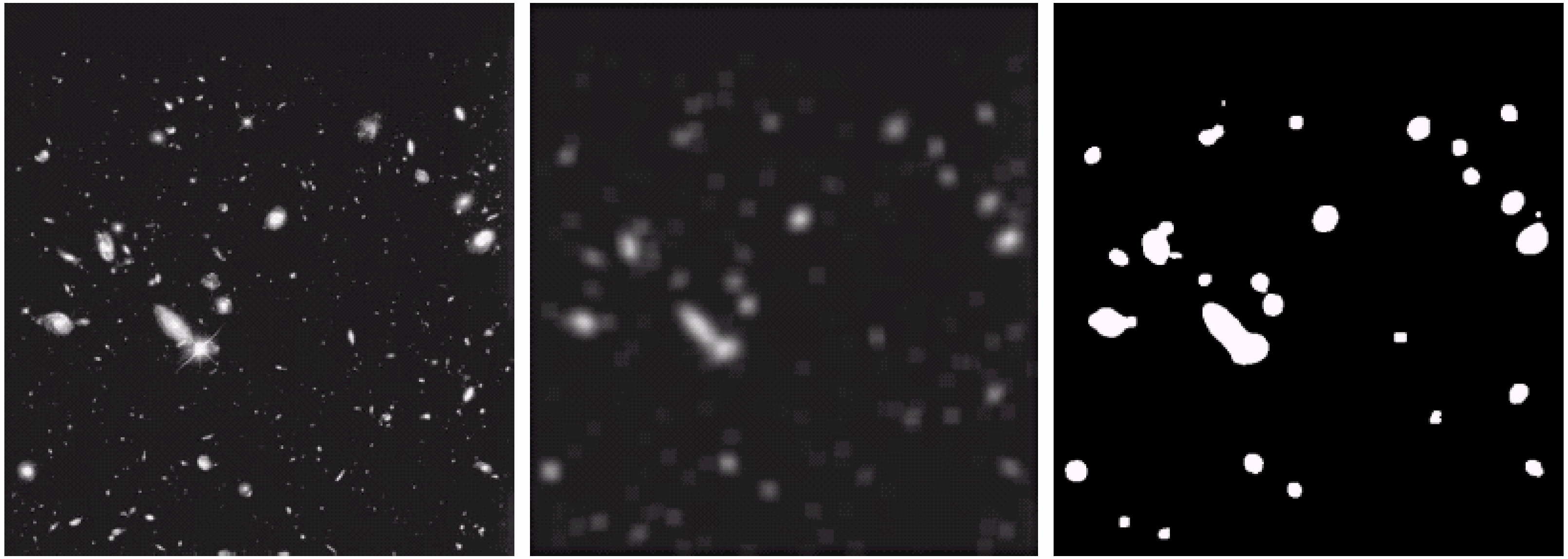


Filter mask sizes: 3x3, 5x5, 9x9, 15x15, 35x35.

a b
c d
e f

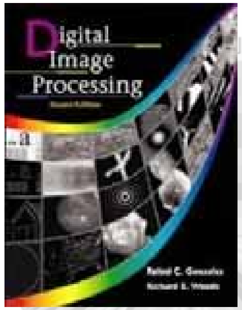
FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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 gray 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.

3.6 Smoothing Spatial Filters



a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)



3.6.2 Smoothing Spatial Non-linear Filters

- Median Filter
  - The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.
- For certain noise, such as ***impulse noise*** or ***salt-and-pepper*** noise, median filter is effective.
- The median, ξ, of a set of values is such that half of the values in the set are less than or equal to ξ, and half are greater than or equal to ξ.

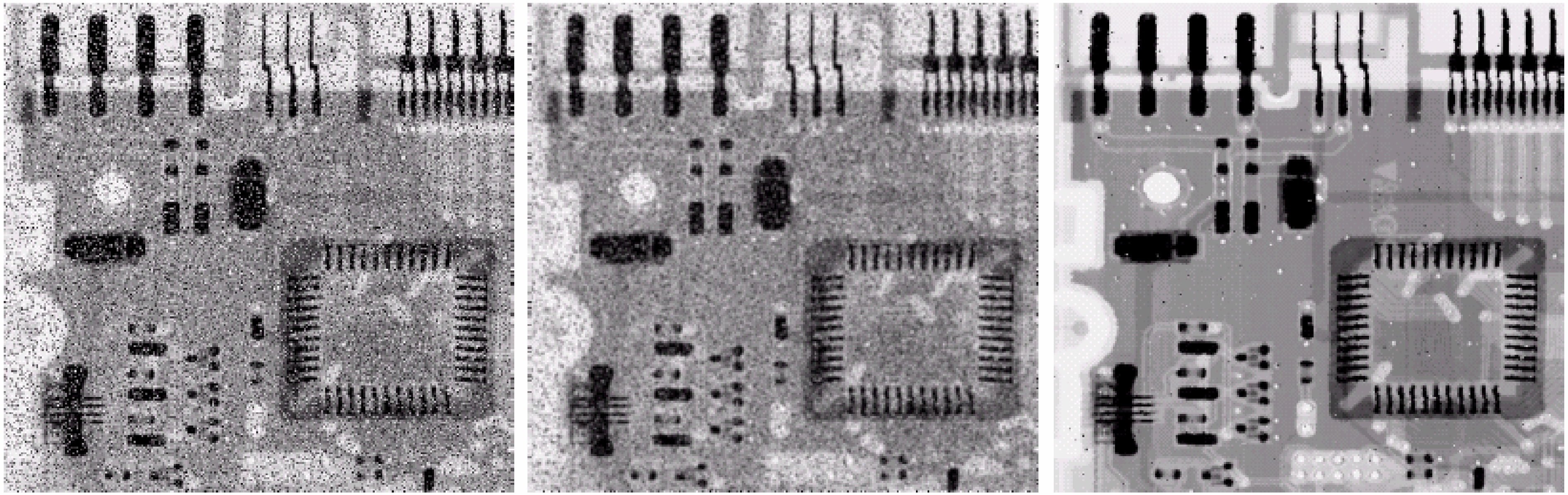
3.6.2 Smoothing Spatial Non-linear Filters

- ***Median filtering***:

1. Sort the values of the target pixel and its neighboring pixels to find the median.
2. Replace the target pixel with the median

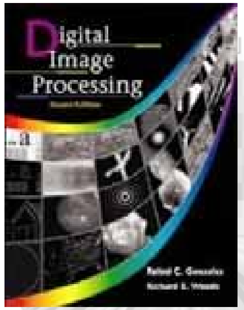
For example, the median is the 5^{th} largest value of the 3x3 neighborhood and 15th largest value in the 5x5 neighborhood.

3.6.2 Smoothing Spatial Non-linear Filters



a b c

FIGURE 3.37 (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.)



3.7 Sharpening Spatial Filters

- Image averaging = low-pass filtering = image blurring =spatial integration
- Image sharpening = high-pass filtering = spatial differentiation.
- It enhances the edges and the other discontinuities

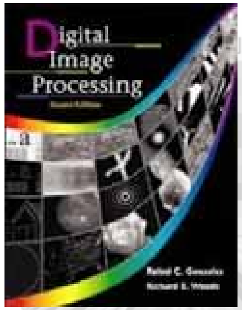
First order difference is

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

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

Second order difference

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



3 .7 Sharpening Spatial Filters

a b
c

FIGURE 3.38 (a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

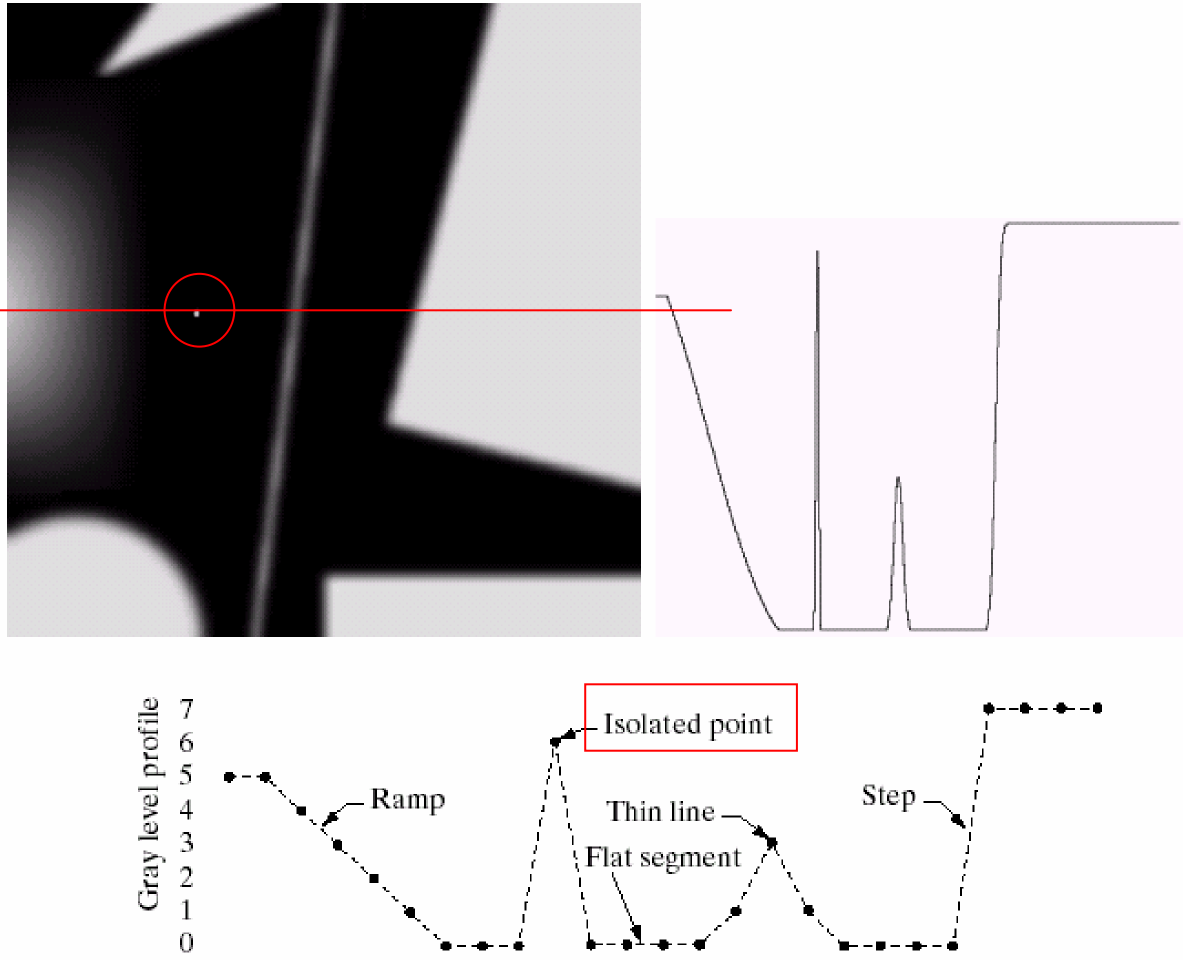


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First Derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0			
Second Derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0			

3.7.2 Second order derivative for enhancement

- Isotropic filter, rotational invariant — Laplacian

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

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

Use the Laplacian for image enhancement

$g(x,y) = f(x,y) - \nabla^2 f(x,y)$ if the center coefficient of the Laplacian mask is negative.

$g(x,y) = f(x,y) + \nabla^2 f(x,y)$ if the center coefficient of the Laplacian mask is positive

3.7.2 Second order derivative for enhancement

0	1	0
1	−4	1
0	1	0

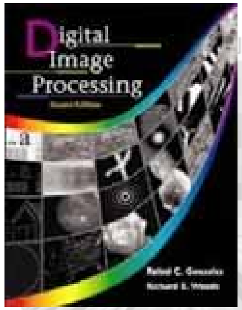
1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

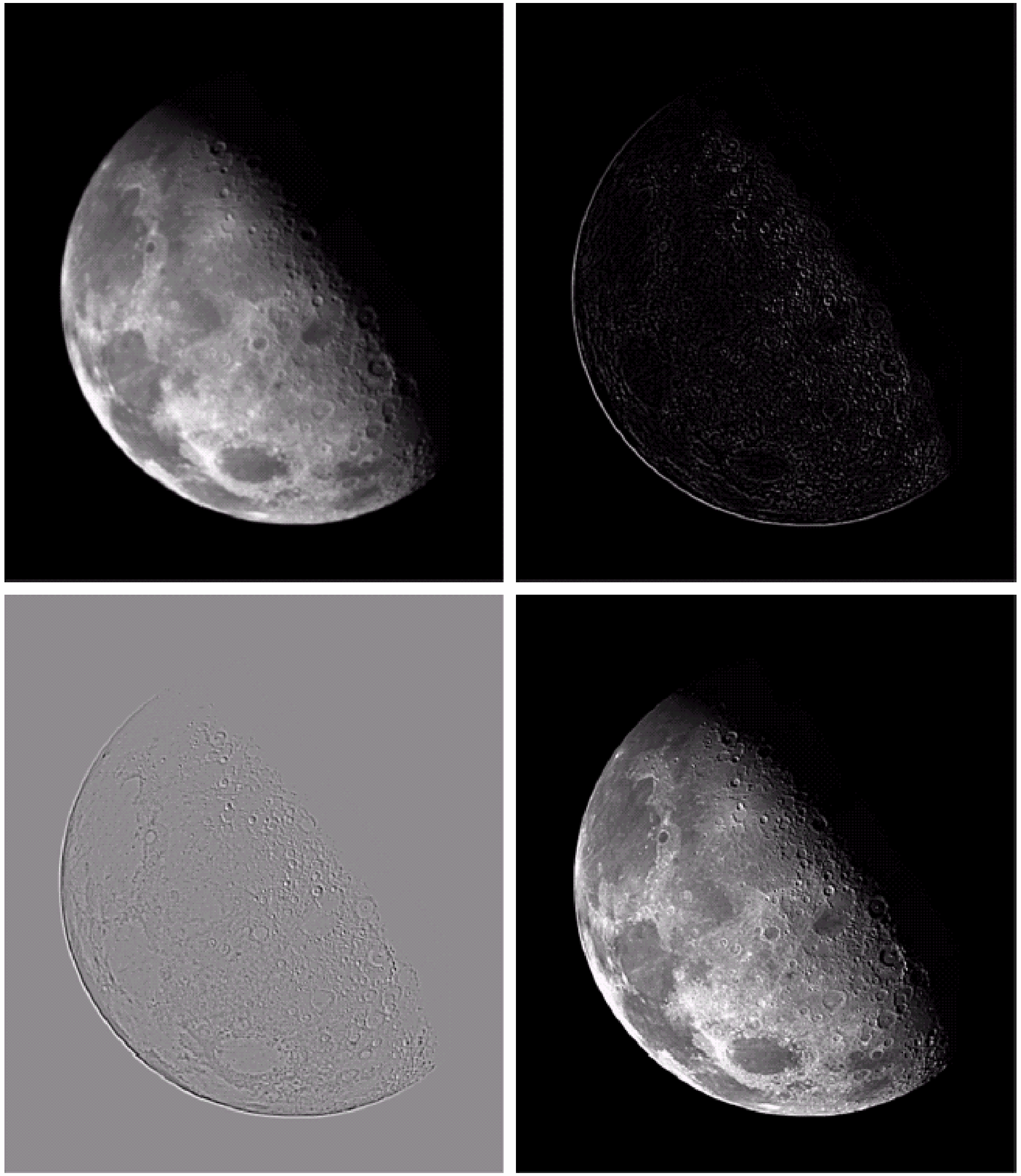
−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

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



3.7.2 Second order derivative for enhancement



a b
c d

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

3.7.2 Second order derivative for enhancement

- Simplification:

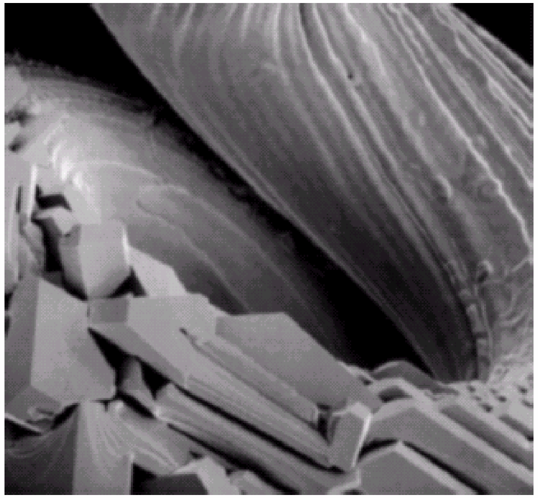
$$g(x, y)=f(x, y)-[f(x+1,y)+f(x-1, y)+f(x, y+1)+f(x, y-1)]+4f(x, y)$$

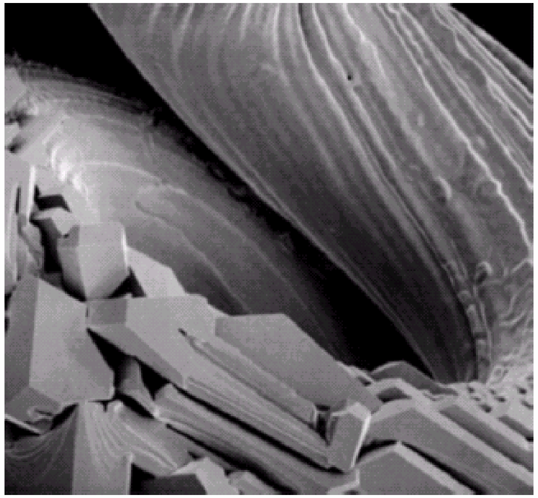
$$=5f(x,y)- [f(x+1,y)+f(x-1,y)+f(x, y+1)+f(x, y-1)]$$

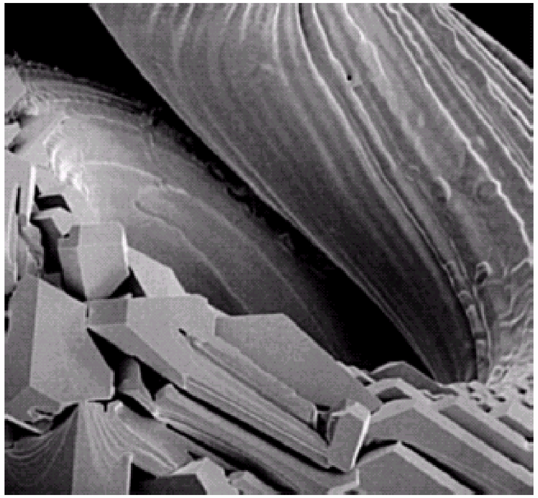
3.7.2 Second order derivative for enhancement

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1







a b c
d e

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

3.7.2 Second order derivative for enhancement

- *Image enhancement*

  $g(x,y)=f(x,y)-\nabla^2 f(x,y)$ *if the center of laplacian mask<0*

  $g(x,y)=f(x,y)+\nabla^2 f(x,y)$ *if the center of laplacian mask>0*

- *Unsharp masking*

  $f_s(x,y)=f(x,y)-f^*(x,y)$ *where* $f^*(x,y)$ *is the blurred image*

- *High boost filtering*

  $$f_{hb}(x,y)=Af(x,y)-f^*(x,y)=(A-1)f(x,y)+f(x,y)-f^*(x,y)$$
  $$=(A-1)f(x,y)+f_s(x,y)$$

- *Using Laplacian*

  $f_{hb}(x,y)=Af(x,y)-\nabla^2 f(x,y)$ *if the center coefficient of the Laplacian mask is negative.*

  $f_{hb}(x,y)=Af(x,y)+\nabla^2 f(x,y)$ *if the center coefficient of the Laplacian mask is positive*

3.7.2 Second order derivative for enhancement

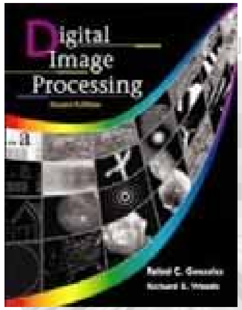
a

0	−1	0
−1	$A + 4$	−1
0	−1	0

b

−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

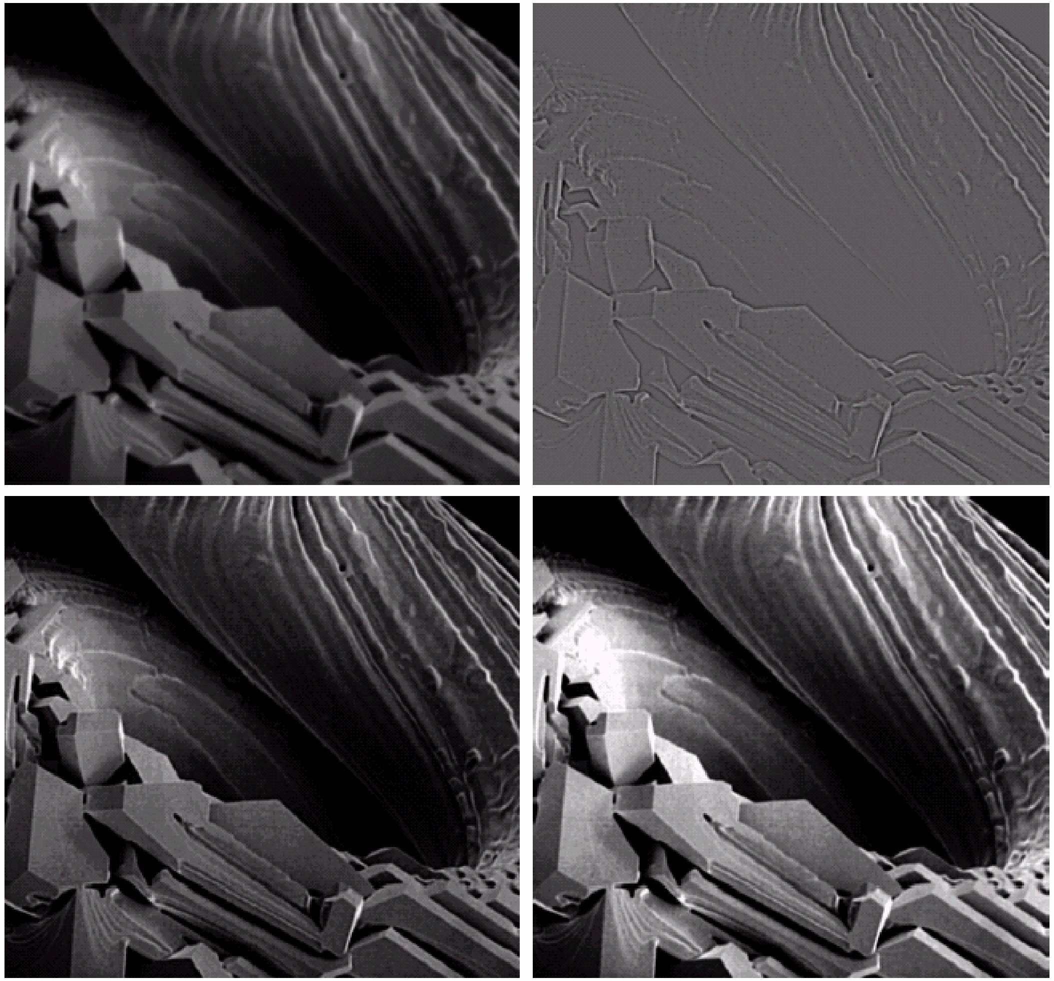
FIGURE 3.42 The high-boost filtering technique can be implemented with either one of these masks, with $A \geq 1$.

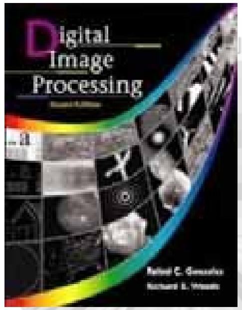


3.7.2 Second order derivative for enhancement

a b
c d

FIGURE 3.43 (a) Same as Fig. 3.41(c), but darker. (a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$. (c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.





3.7.3 First derivative for enhancement

- $\nabla f(x,y)=[G_x, G_y]=[\partial f/\partial x, \partial f/\partial y\ ]$
- $\nabla f(x,y)=mag\ (\nabla f)=[G_x{}^2, G_y{}^2]^{1/2}$

  $=[(\partial f/\partial x)^2, (\partial f/\partial y)^2\ ]^{1/2}$
- *Robert operator*

  $G_x=\partial f/\partial x=z_9-z_5\quad G_y=\partial f/\partial y=z_8-z_6$

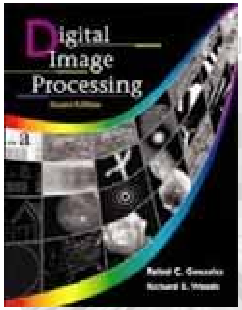
  $\nabla f(x,y)=[(z_9-z_5)^2+(z_8-z_6)^2\ ]^{1/2}$

  $=|z_9-z_5|+|z_8-z_6|$

Sobel operator

$\nabla f(x,y)=|z_7+2z_8+z_9)-(z_1+2z_2+z_3)|$

$+|(z_3+2z_6+z_9)-(z_1+2z_4+z_7)|$



3.7.3 First derivative for enhancement

a

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

b

−1	0
0	1

c

0	−1
1	0

d

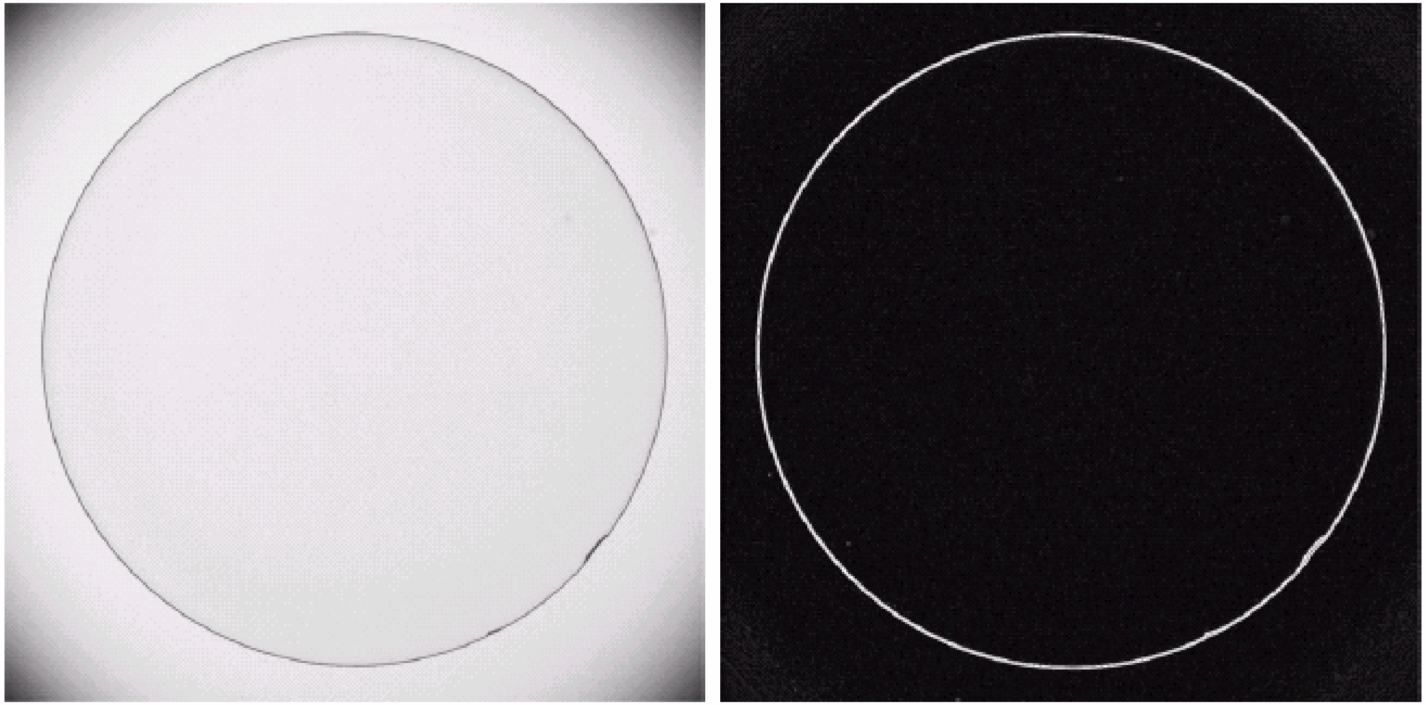
−1	−2	−1
0	0	0
1	2	1

e

−1	0	1
−2	0	2
−1	0	1

FIGURE 3.44 A 3×3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

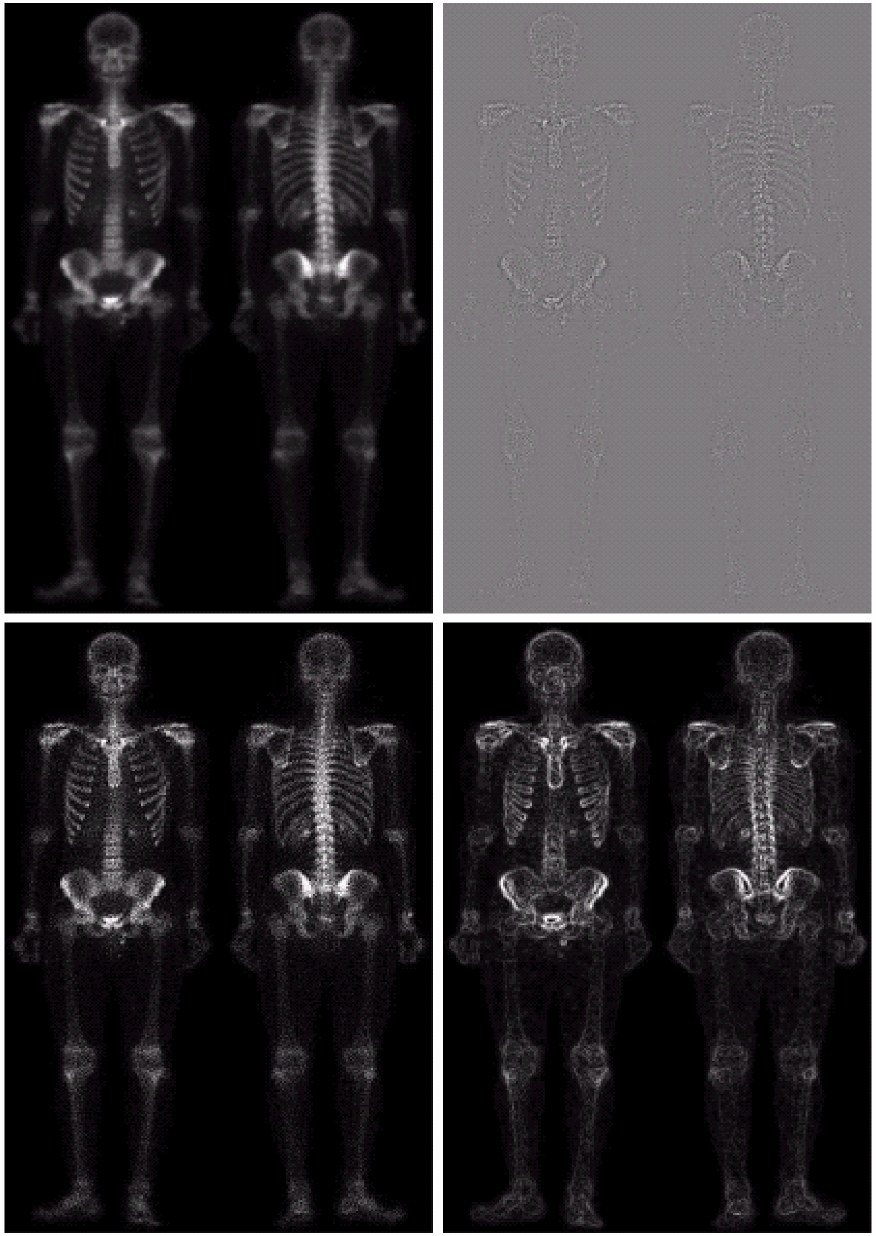
3.7.3 First derivative for enhancement



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

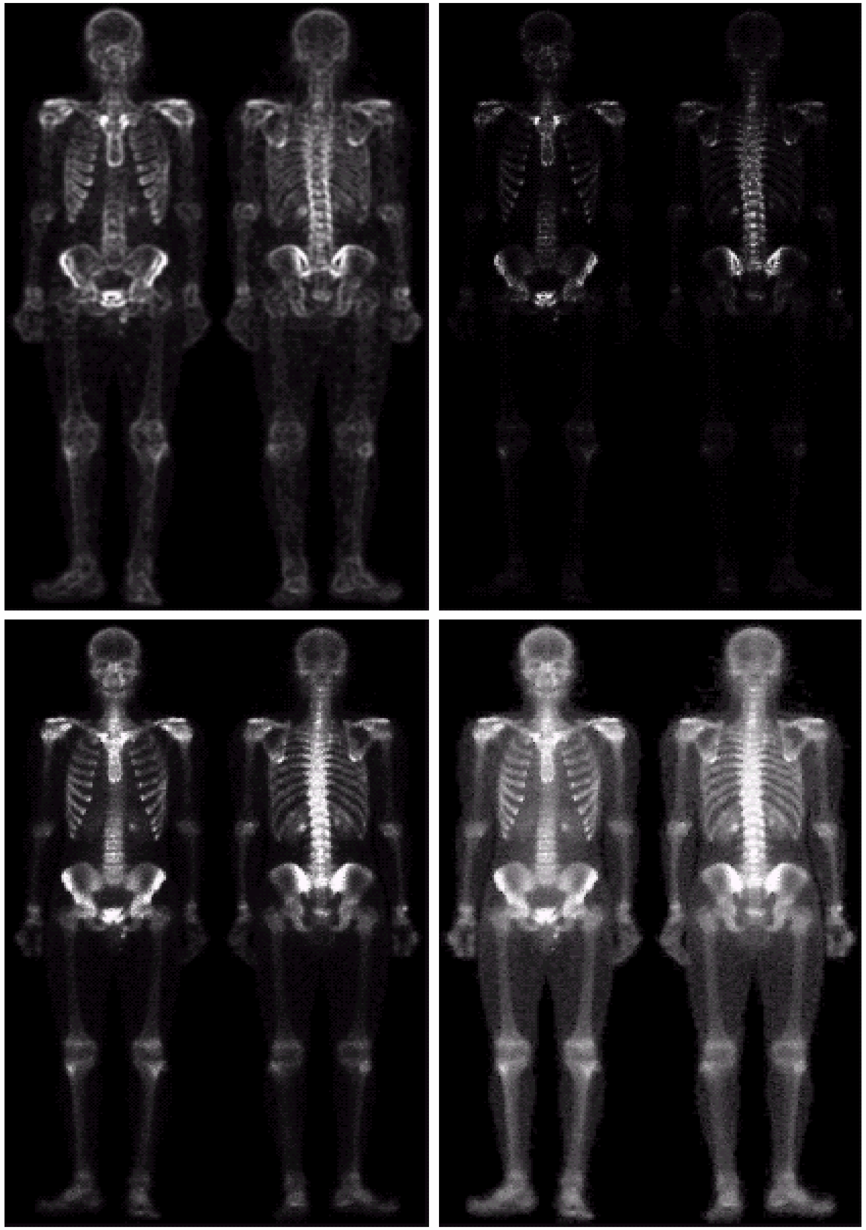
3.8 Combining Spatial Enhancement Methods



a b
c d

FIGURE 3.46 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

3.8 Combining Spatial Enhancement Methods



e	f
g	h

FIGURE 3.46 *(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.)