Chapter-3

Intensity Transformation & Spatial Filtering

(4th Edition)

Part-1: Point Processing

Contents

- Image enhancement using Point Processing
  - What is Point Processing?
  - Negative images
  - Thresholding
  - Logarithmic transformation
  - Power Law transforms
  - Grey level slicing
  - Bit Plane slicing

Spatial Domain Image Enhancement

- Most spatial domain enhancement operations can be reduced to the form

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

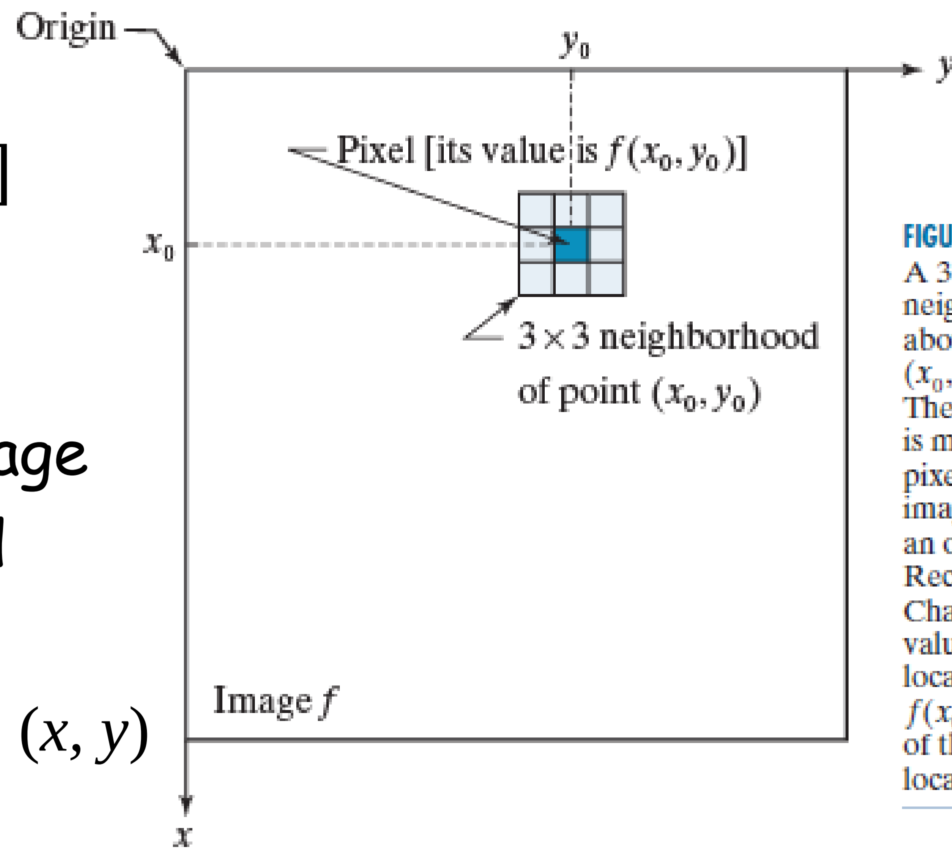


FIGURE 3.1 A 3×3 neighborhood about a point (x_0, y_0) in an image. The neighborhood is moved from pixel to pixel in the image to generate an output image. Recall from Chapter 2 that the value of a pixel at location (x_0, y_0) is $f(x_0, y_0)$, the value of the image at that location.

- $f(x, y)$: Input image
- $g(x, y)$: Processed image
- T : Operator defined over some neighbourhood of (x, y)

3.1 Basic Operations

- Mathematical representation of intensity transformation:

$$g(x, y) = T[f(x, y)] \qquad \text{OR} \quad [\, s = T(r) \,]$$

  - $f(x, y)$: Input image
  - $g(x, y)$: Processed image
  - T: Operator on f, defined over some neighborhood of (x, y)
- **Note:** $T[\,]$ may have one input at (x, y) only or multiple inputs in neighbors of (x, y) depending on $T[\,]$. For example,
  - **Contrast Enhancement:** Uses one pixel value at (x, y) only for input
  - **Smoothing Filter:** Uses several pixels around (x, y) as inputs

Point Processing

- Point processing operations take the form

$$s = T(r)$$

- where, s refers to the processed image pixel value and r refers to the original image pixel value

- The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself

- In this case, T is referred to as a *grey level transformation function* or a *point processing operation*

3.1.1 Intensity Transformations

Contrast Stretching

Thresholding

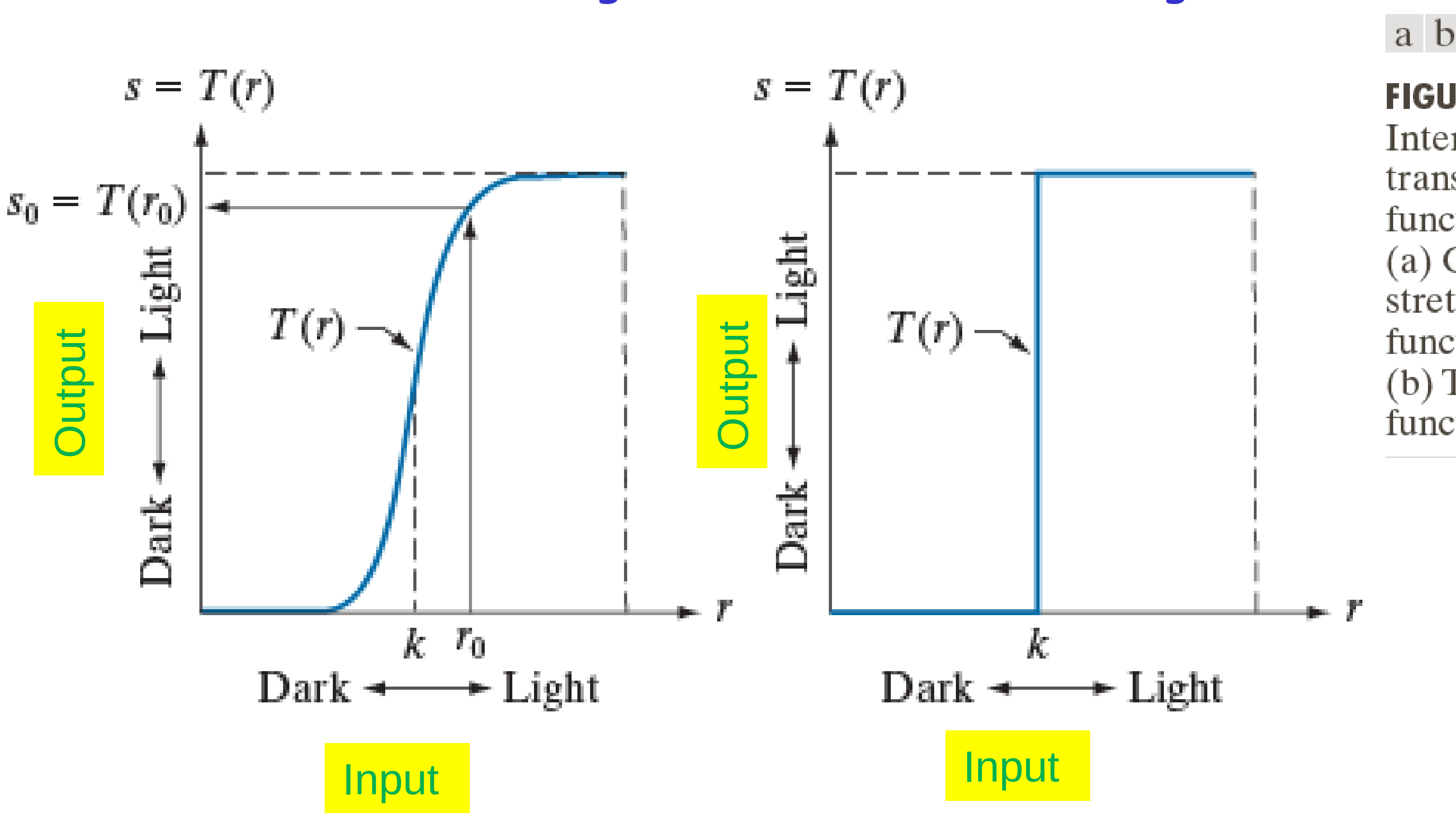


FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

3.2 Basic Intensity Transformations

- There are many kinds of grey level transformations
- Three most common ones are

- **Linear**
  - Negative/Identity
- **Logarithmic**
  - Log/Inverse log
- **Power law**
  - n^{th} power/n^{th} root

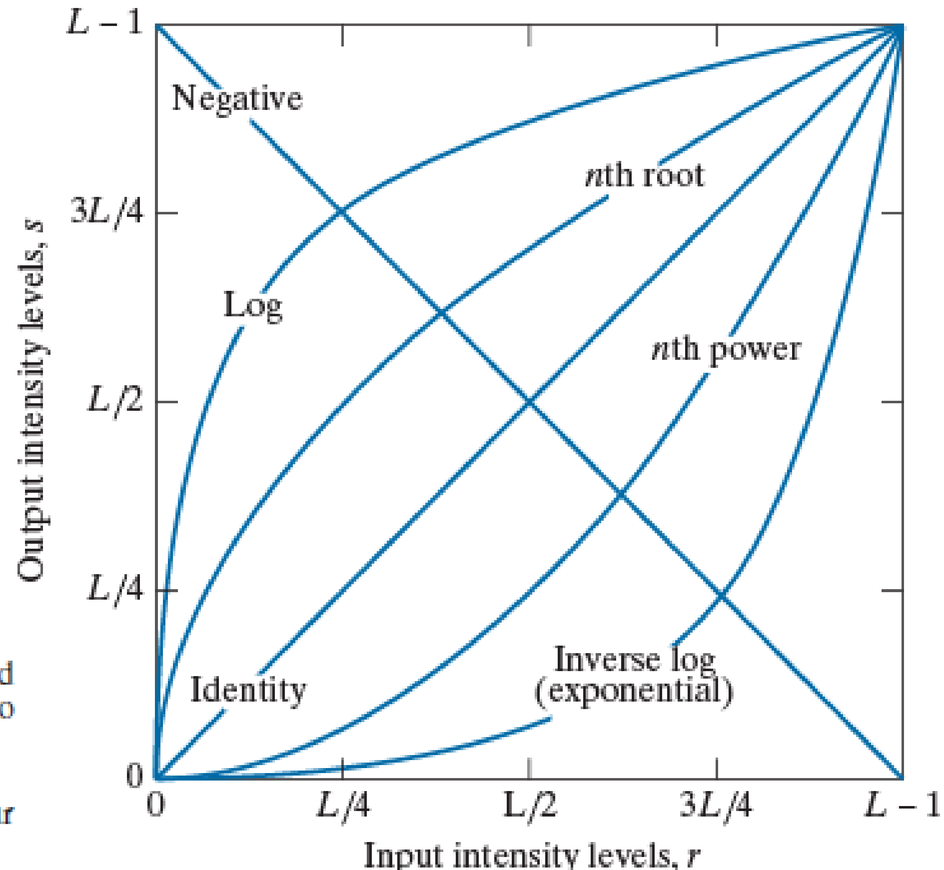
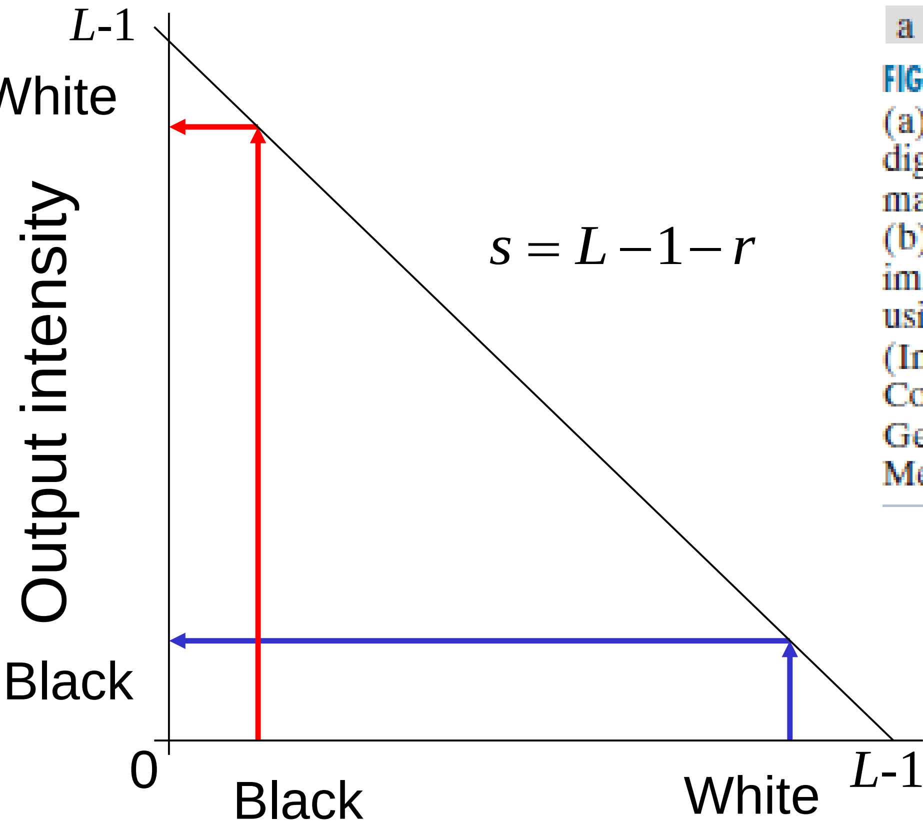


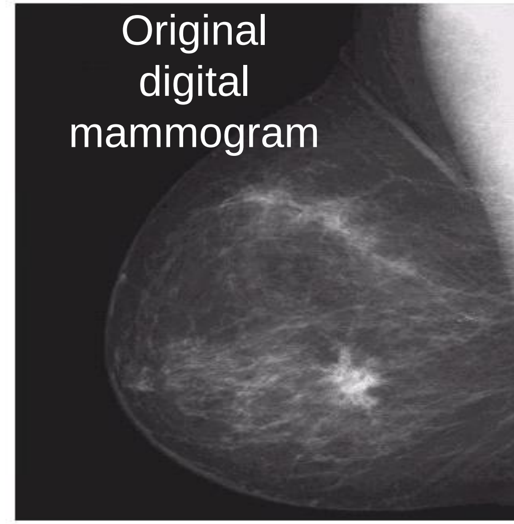
FIGURE 3.3 Some basic intensity transformation functions. Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.

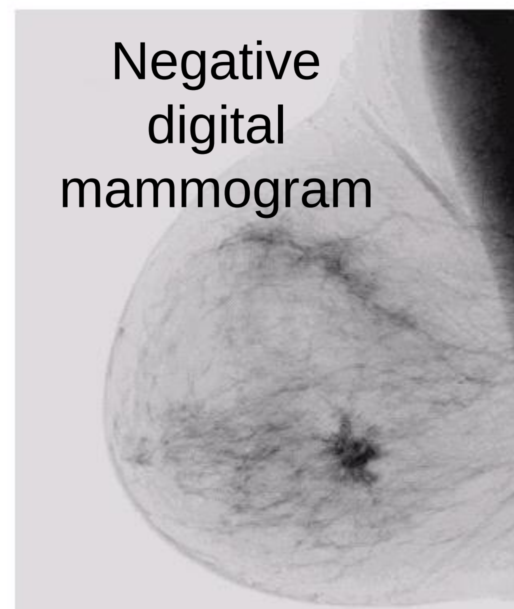
3.2.1 Point Processing Example: Negative Images



a b

FIGURE 3.4 (a) A digital mammogram. (b) Negative image obtained using Eq. (3-3). (Image (a) Courtesy of General Electric Medical Systems.)

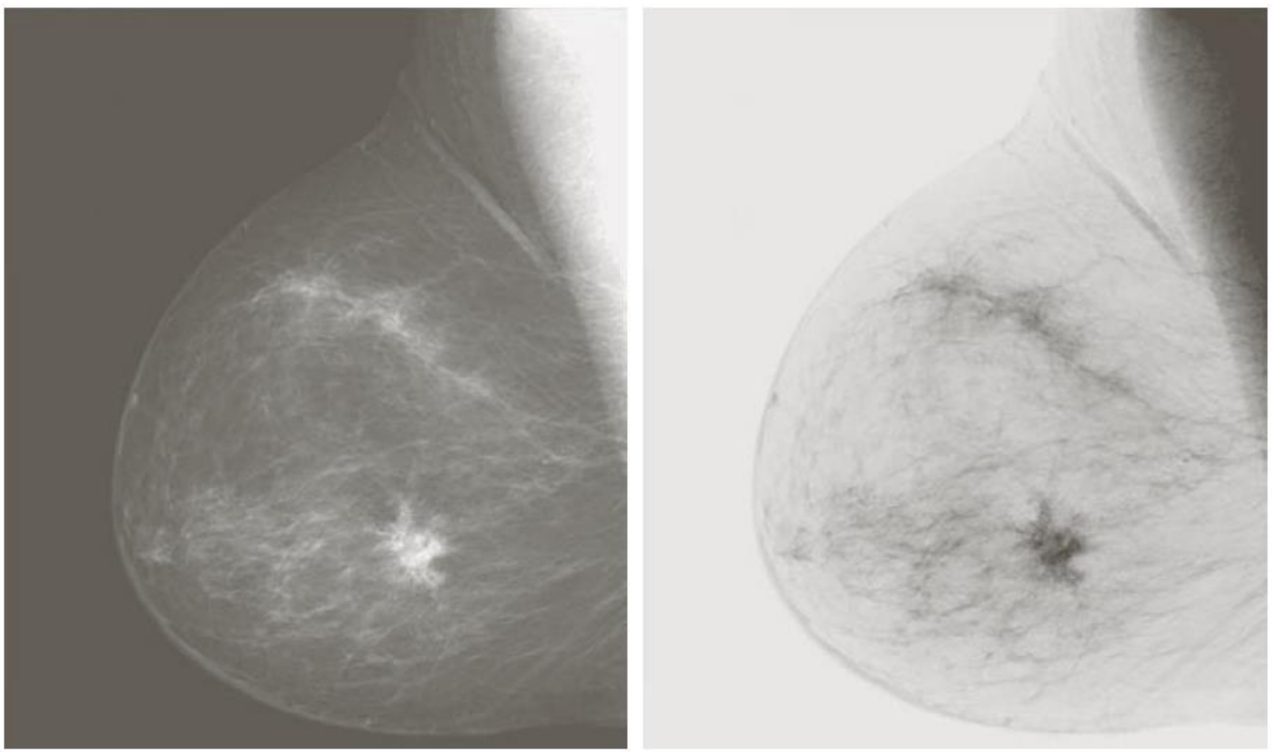




L = Number of gray levels

Image Negatives (contd.)

- Let the range of grey level be $[0, L\text{-}1]$, then
  - For enhancing white or gray detail embedded in dark regions, use transformation $$s = L - 1 - r$$



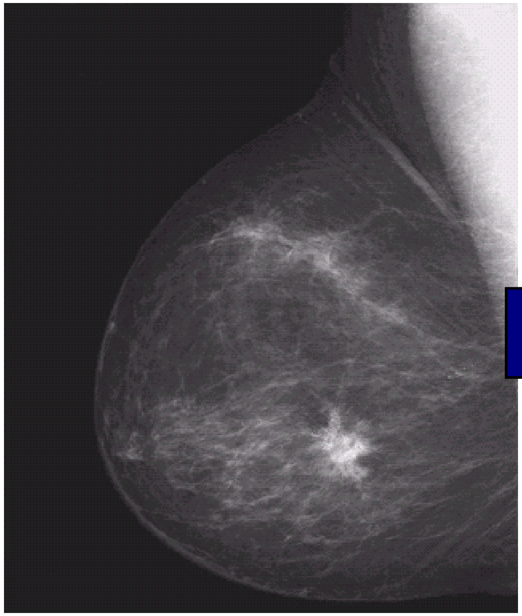
a b

FIGURE 3.4 (a) A digital mammogram. (b) Negative image obtained using Eq. (3-3). (Image (a) Courtesy of General Electric Medical Systems.)

Image Negatives (contd.)

- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image
  - Note the clarity of the tissues in the negative image of the mammogram below

Original Image



$s = 1.0 - r$

Negative Image

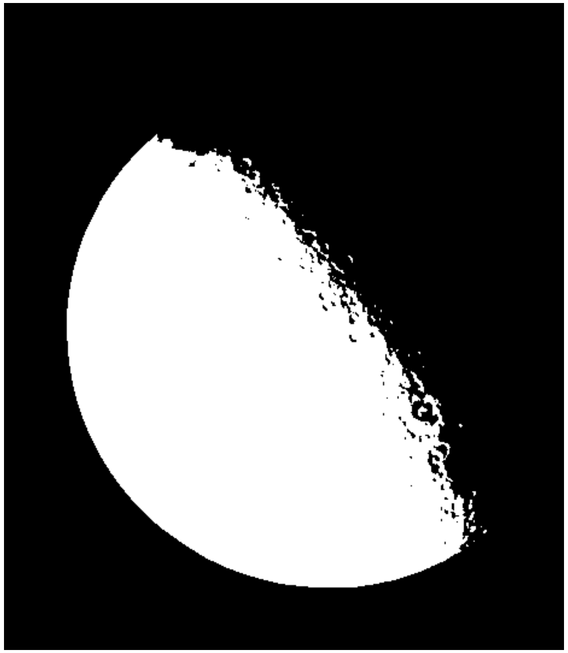
Point Processing Example: Thresholding

- Thresholding transformations are useful for segmentation to isolate an object of interest from background

Gray scale

$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$

Binary



3.2.2 Logarithmic Transformations

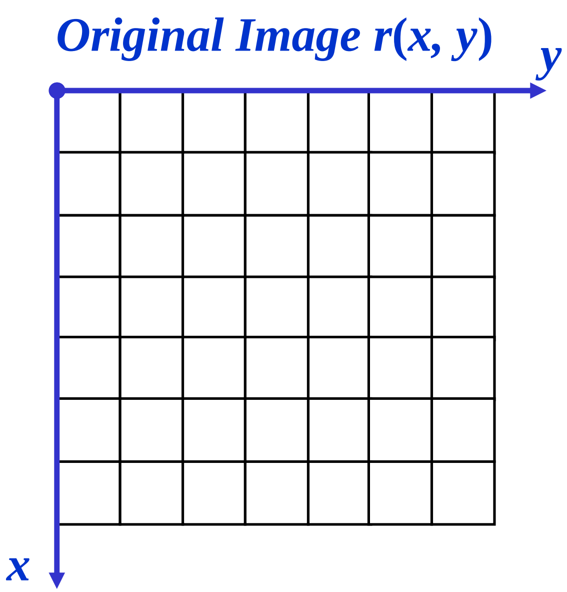
- General form :

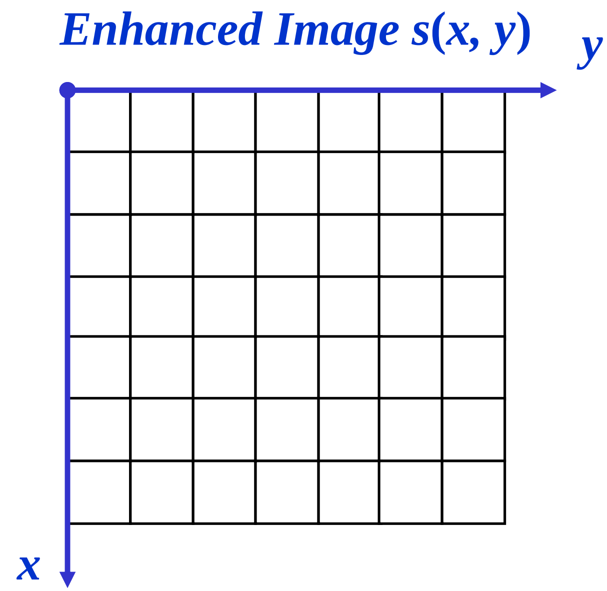
$$s = c\log(1+r)$$

- Maps a narrow range of low input grey level values into a wider range of output values

- Inverse log transformation: Performs the opposite transformation

Logarithmic Transformations (cont…)





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

- We usually set c to 1
- Grey levels must be in the range [0.0, 1.0]

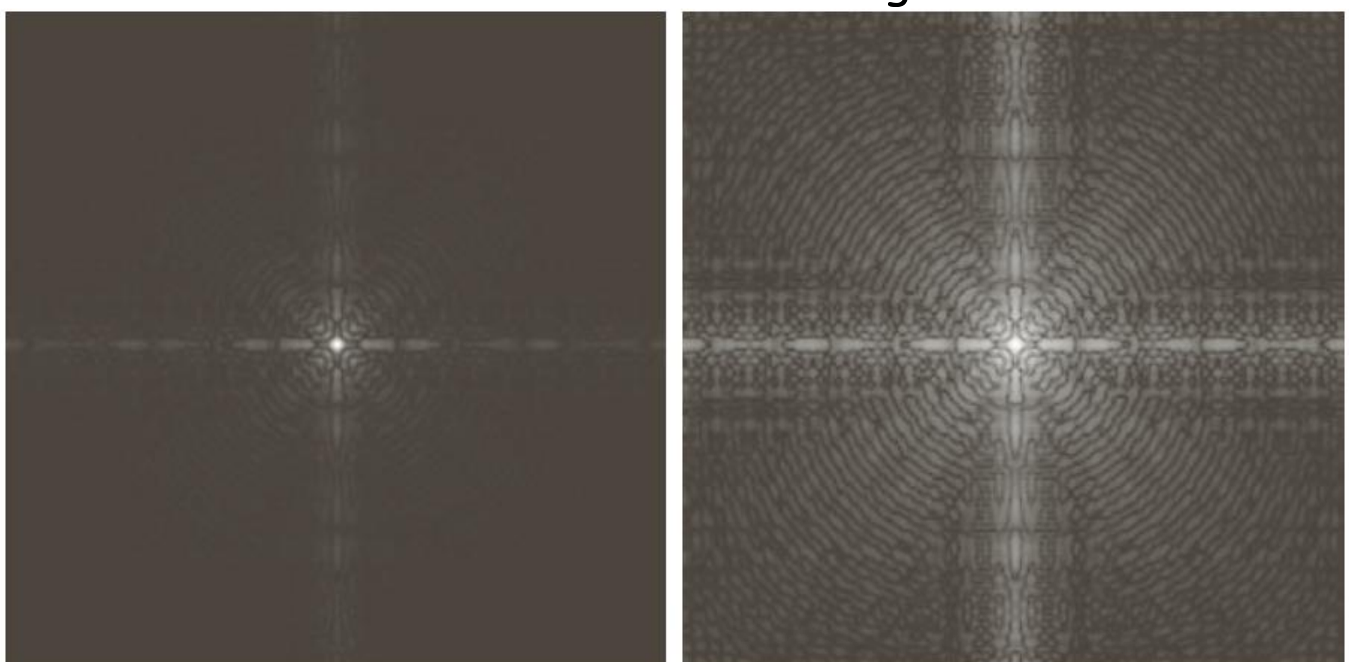
Log Transformations

$$s = c \log(1 + r) \quad \text{and} \quad r \geq 0$$

- c : constant
- Expands values of dark pixels while compressing the higher-level values

Fourier spectrum
Range of values: 0 to 1.5×10^6

log Transformation of Fourier spectrum
Range of values: 0 to 6.2



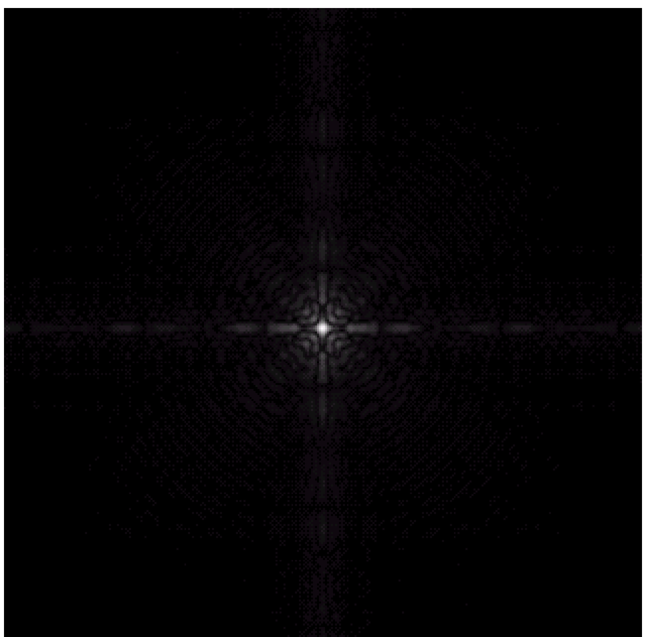
a b

FIGURE 3.5 (a) Fourier spectrum displayed as a grayscale image. (b) Result of applying the log transformation in Eq. (3-4) with $c = 1$. Both images are scaled to the range [0, 255].

Logarithmic Transformations (cont…)

- Log functions are particularly useful when the input grey level values may have an extremely large range of values, i.e., large dynamic range.
- In this example the Fourier transform of an image is put through a log transform to reveal more detail

Range of values: 0 to 1.5×10^6

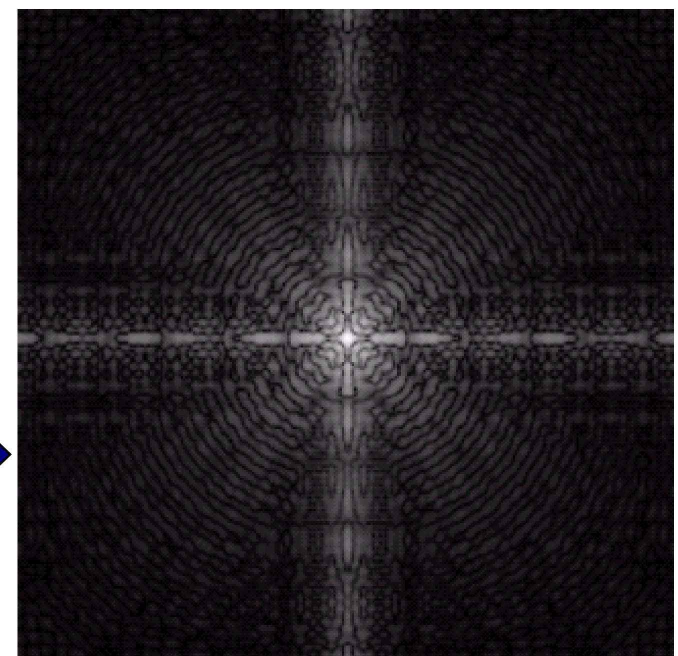


a b

FIGURE 3.5 (a) Fourier spectrum displayed as a grayscale image. (b) Result of applying the log transformation in Eq. (3-4) with $c = 1$. Both images are scaled to the range [0, 255].

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

Range of values: 0 to 6.2



3.2.3 Power Law Transformations

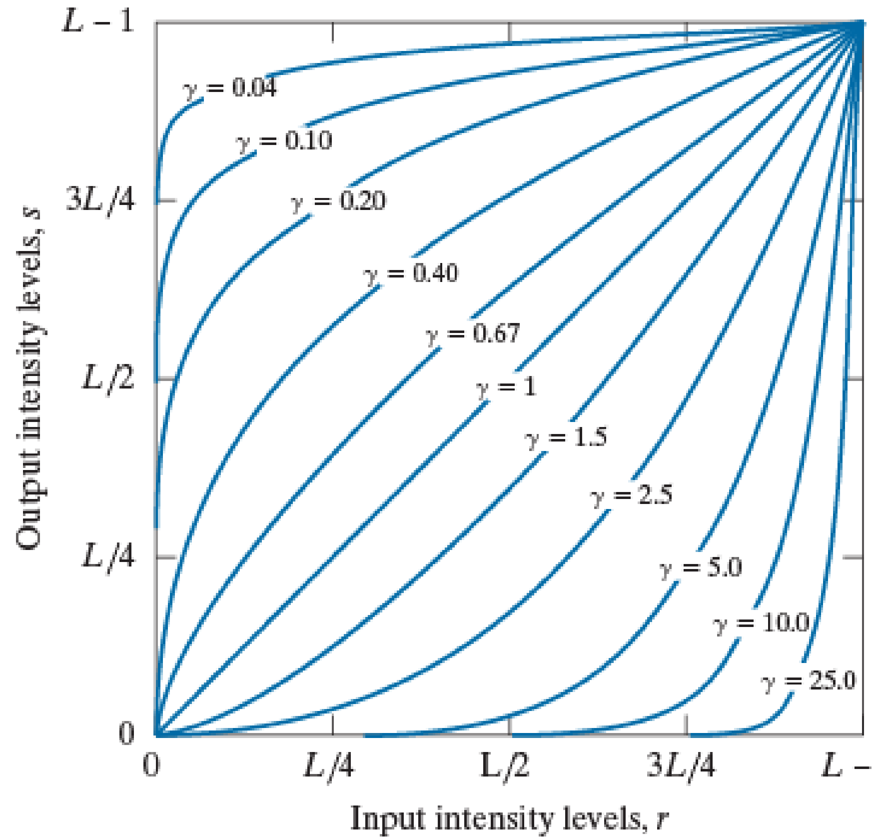
- Power law transformations have the following form

$$s = cr^{\gamma} \quad \text{where } c, \gamma \text{ : positive constants}$$

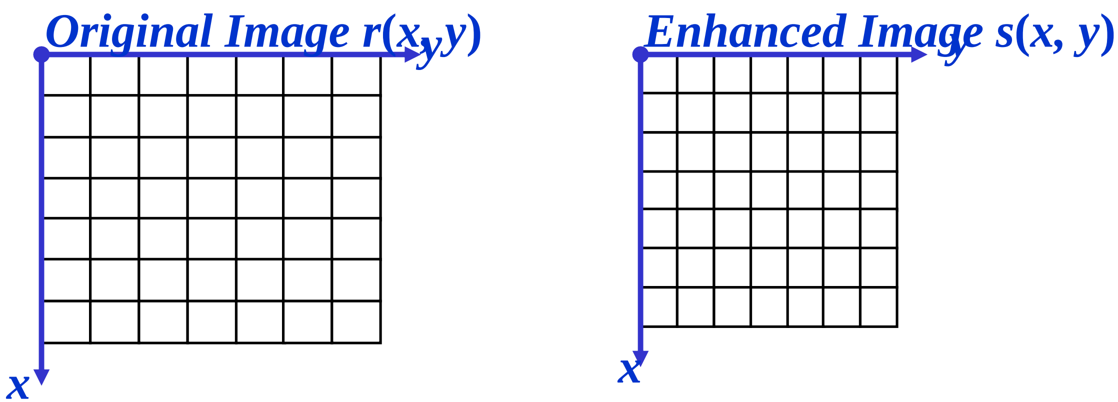
- Fractional γ: Map a narrow range of dark input values into a wider range of output values or vice versa

- Varying γ gives a whole family of curves

FIGURE 3.6 Plots of the gamma equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.



Power Law Transformations (cont…)

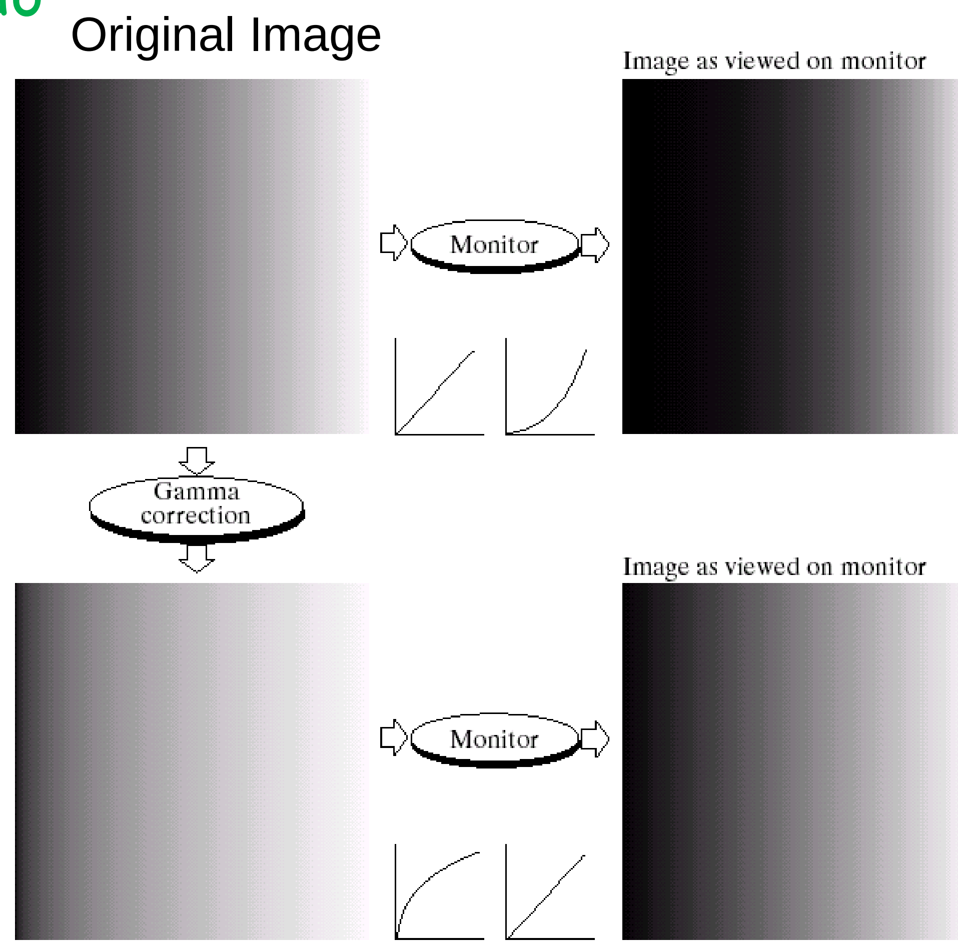


- Typical formula: $s = cr^{\gamma}$
- If 0-offset is needed: $s = c(r+\varepsilon)^{\gamma}$

- We usually set C to 1
- Grey levels must be in the range [0.0, 1.0]

Power Law-Gamma Correction

- Gamma correction is often used in computer monitors
- Problem: Display devices do not respond linearly to different intensities
- Can be corrected using a log transform



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).

3rd Edition

Power-Law Transformations : Gamma Correction Application

Desired image

Figure 3.7

After Gamma correction

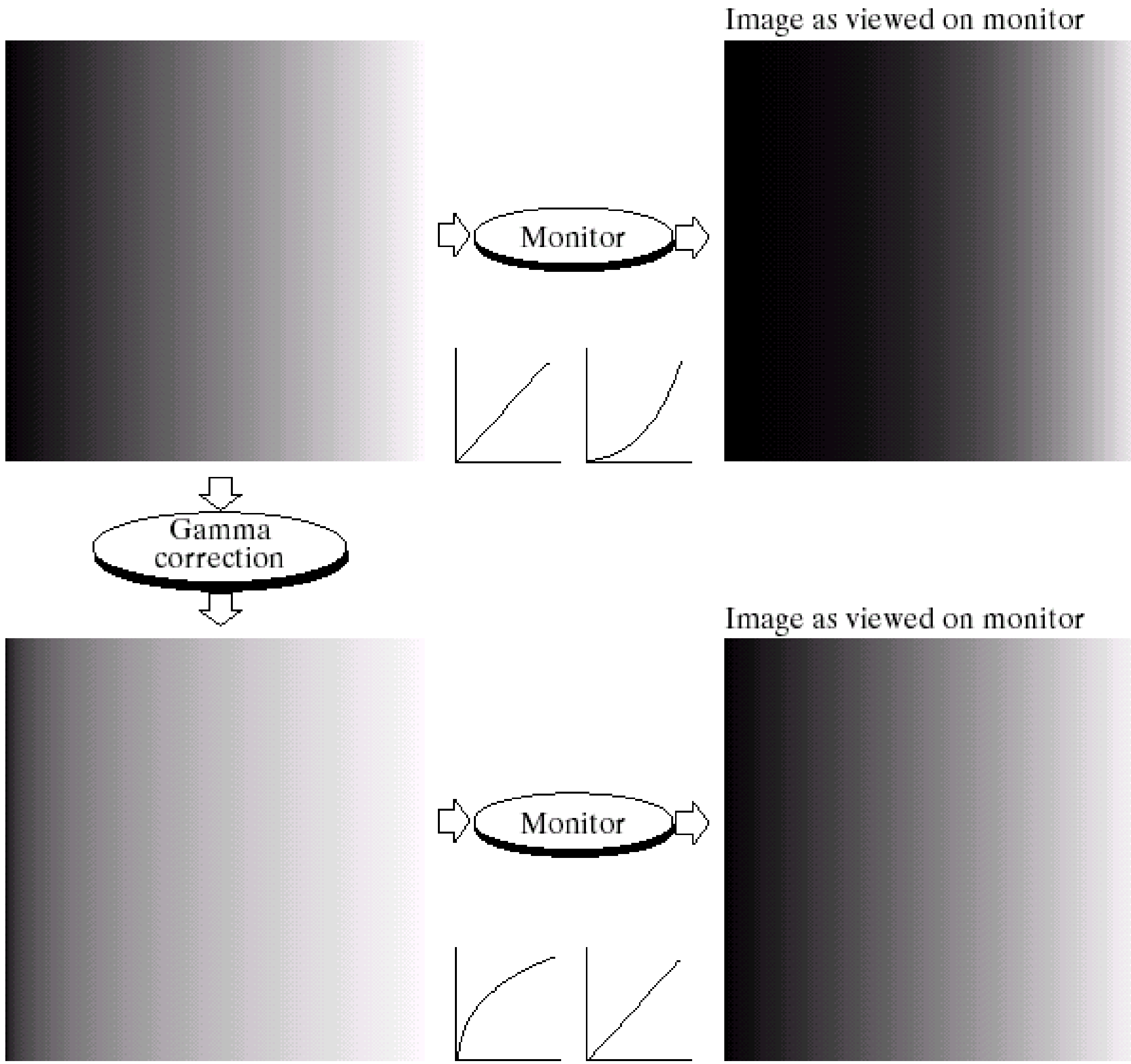


Image displayed at Monitor

Image displayed at Monitor

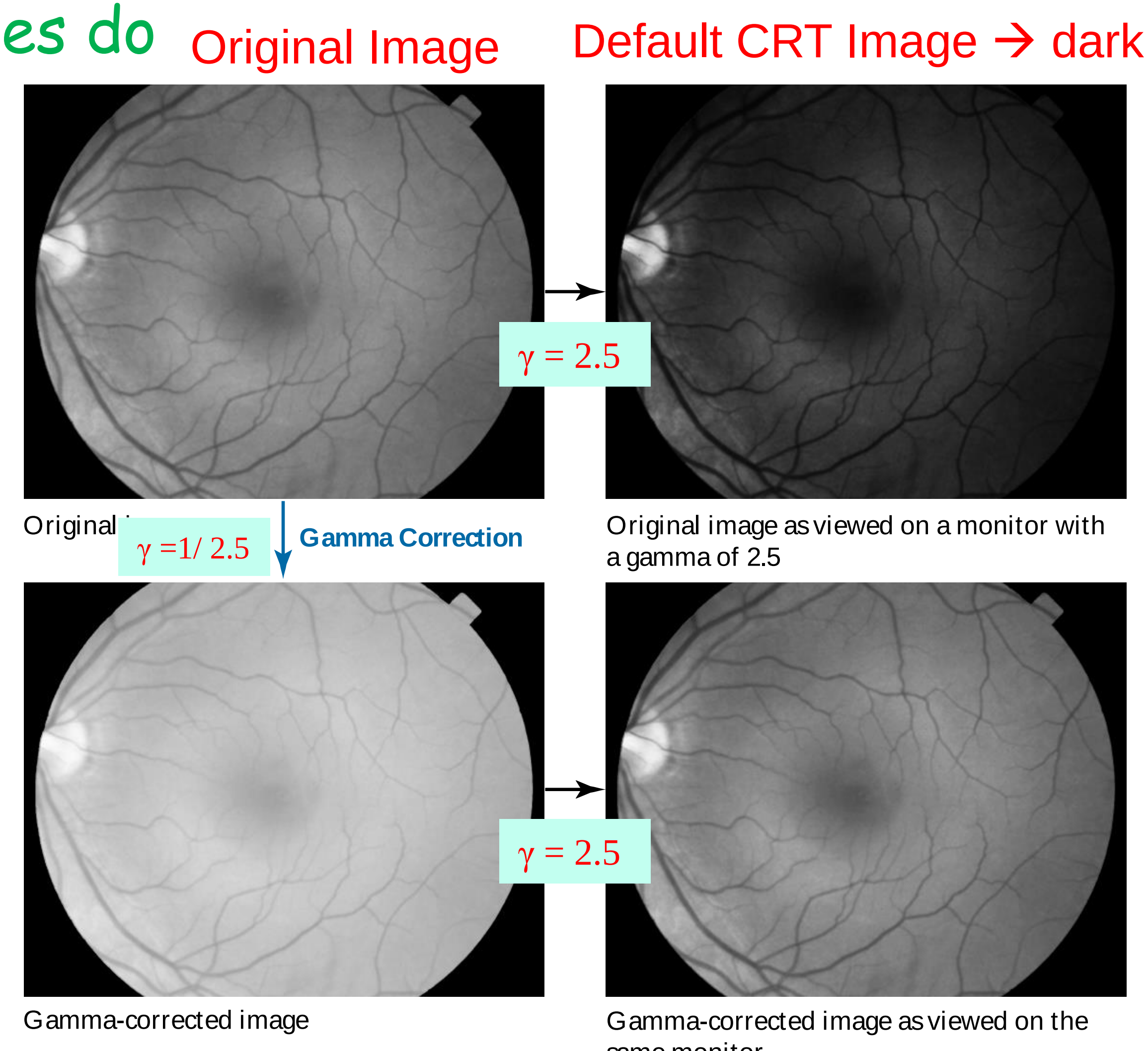
3rd Edition

Power Low-Gamma Correction

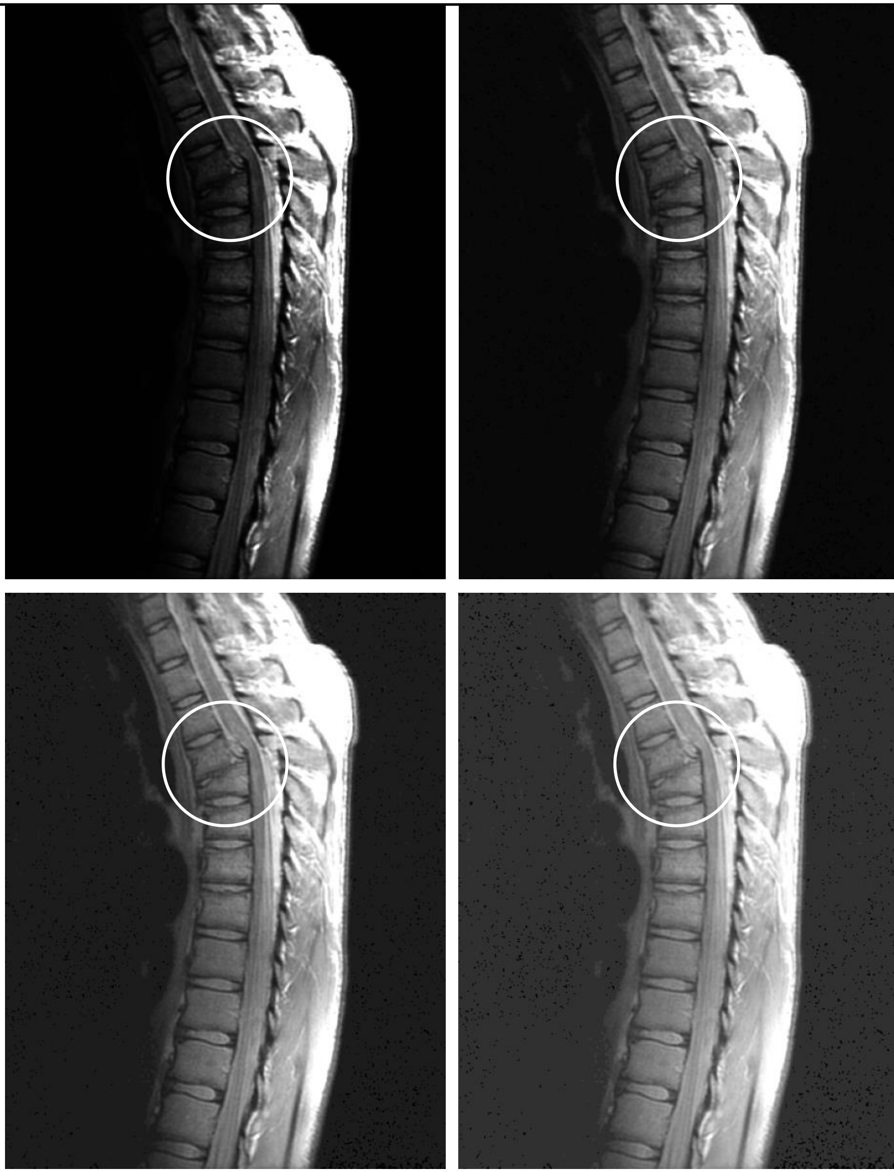
- Gamma correction is often used in computer monitors
- Problem: Display devices do not respond linearly to different intensities
- Can be corrected using a log transform
- (a) and (d) appear almost identical

a	b
c	d

FIGURE 3.7 (a) Image of a human retina. (b) Image as as it appears on a monitor with a gamma setting of 2.5 (note the darkness). (c) Gamma-corrected image. (d) Corrected image, as it appears on the same monitor (compare with the original image). (Image (a) courtesy of the National Eye Institute, NIH)



Power-Law Transformations : Gamma Correction Application



a b
c d

FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine (the region of the fracture is enclosed by the circle). (b)–(d) Results of applying the transformation in Eq. (3-5) 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.)

MRI Image before and after Gamma Correction

Power Law Example (cont…)

- Correction of MRI image of a fractured human spine

- Different curves highlight different detail

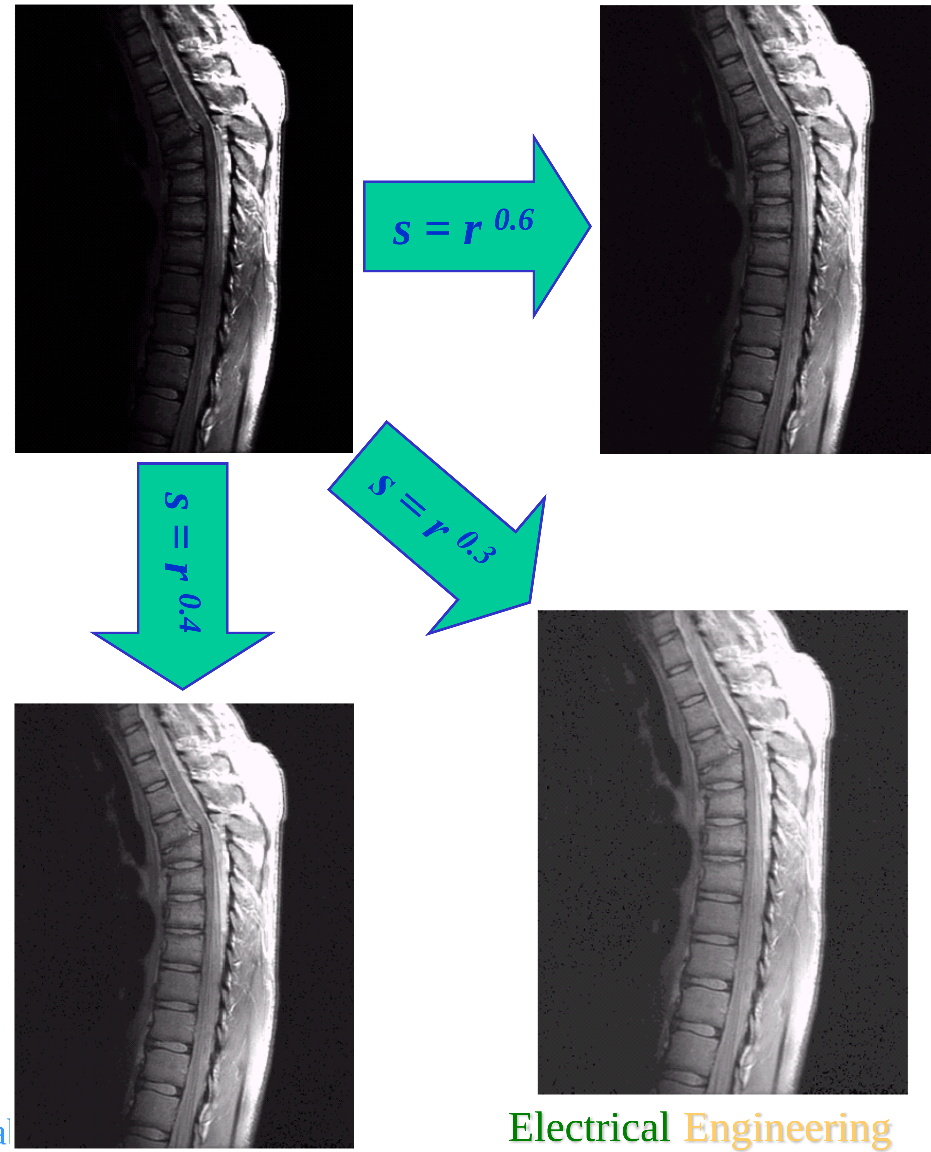


Figure 3.8

Power Law Example

a	b
c	d

FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine (the region of the fracture is enclosed by the circle). (b)–(d) Results of applying the transformation in Eq. (3-5) 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.)

Original MRI of fractured human spine

Power Law Example (cont…)

$\gamma = 0.6$

Figure 3.8(b)

Power Law Example (cont…)

$$\gamma = 0.4$$

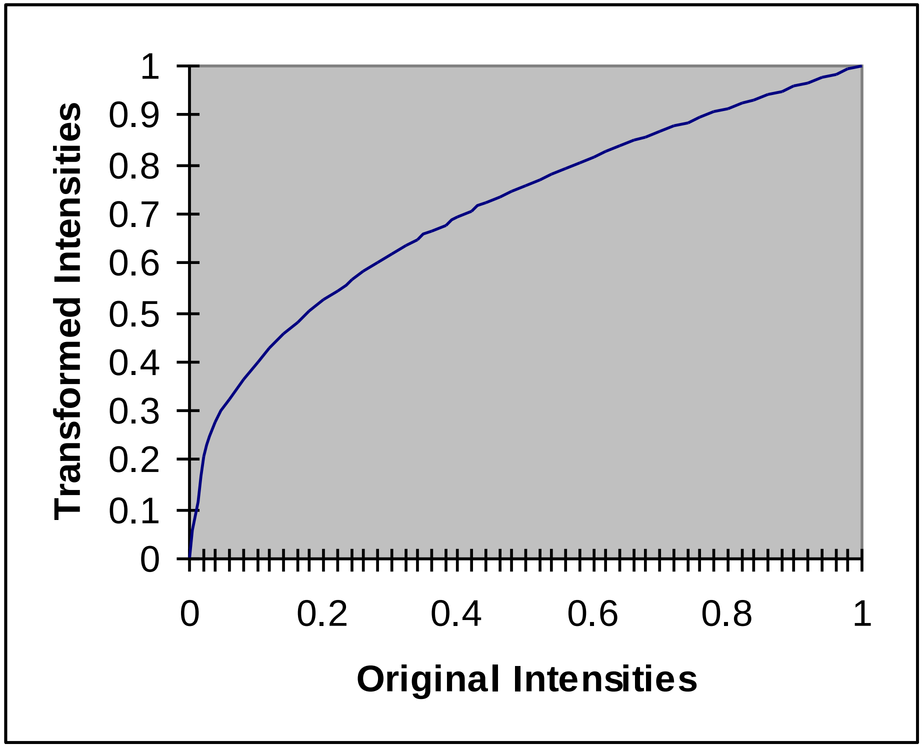


Figure 3.8(c)

Power Law Example (cont…)

$$\gamma = 0.3$$

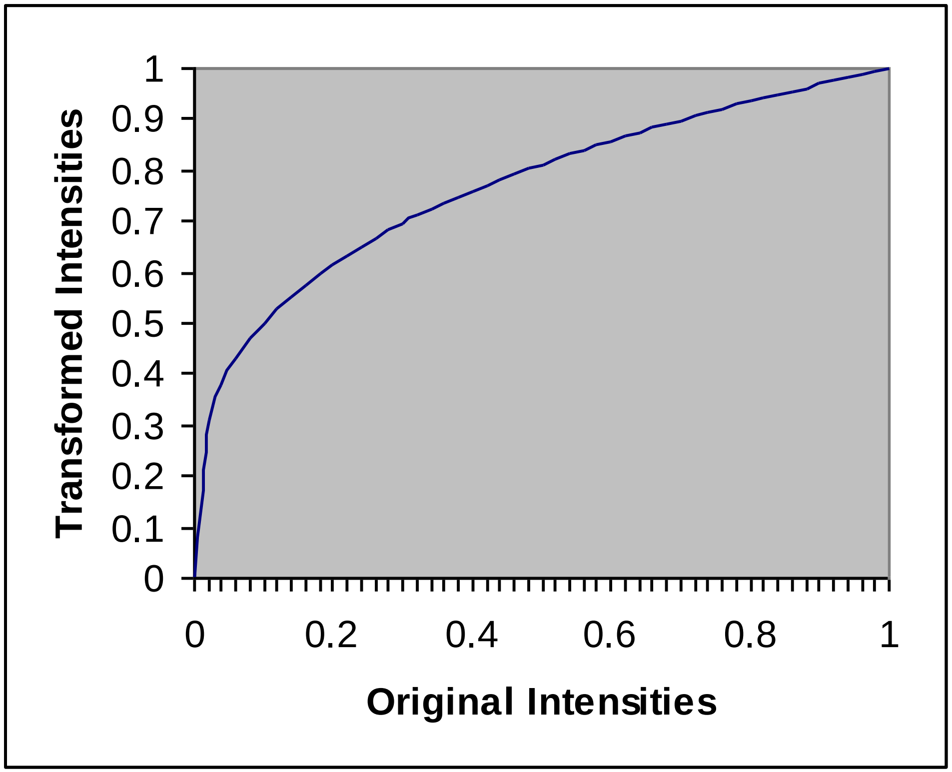
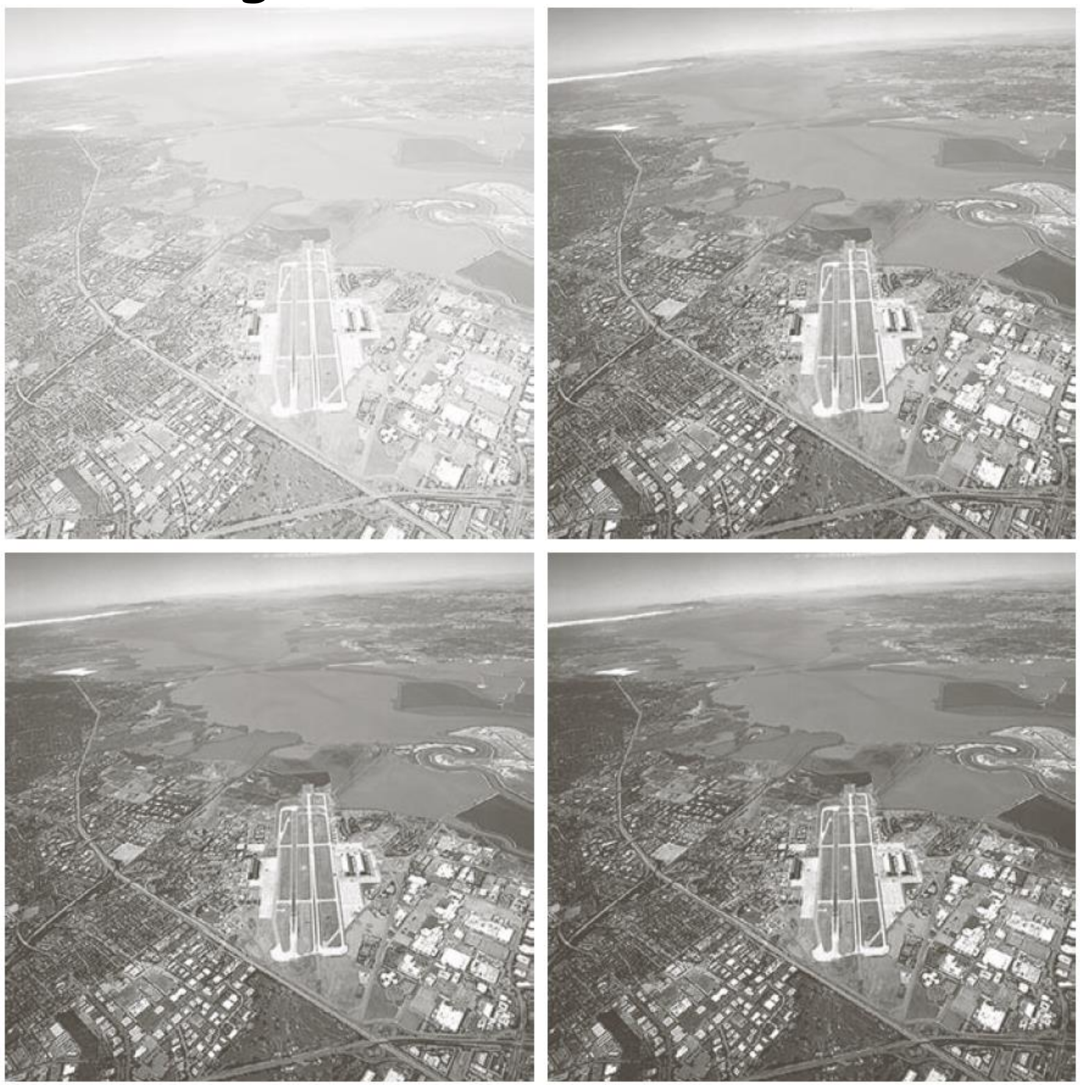


Figure 3.8(d)

Power-Law Transformations : Gamma Correction Application

Original



a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3-5) with $\gamma = 3.0$, 4.0, and 5.0, respectively. ($c = 1$ in all cases.) (Original image courtesy of NASA.)

$\gamma = 3$

Ariel images after Gamma Correction

$\gamma = 4$

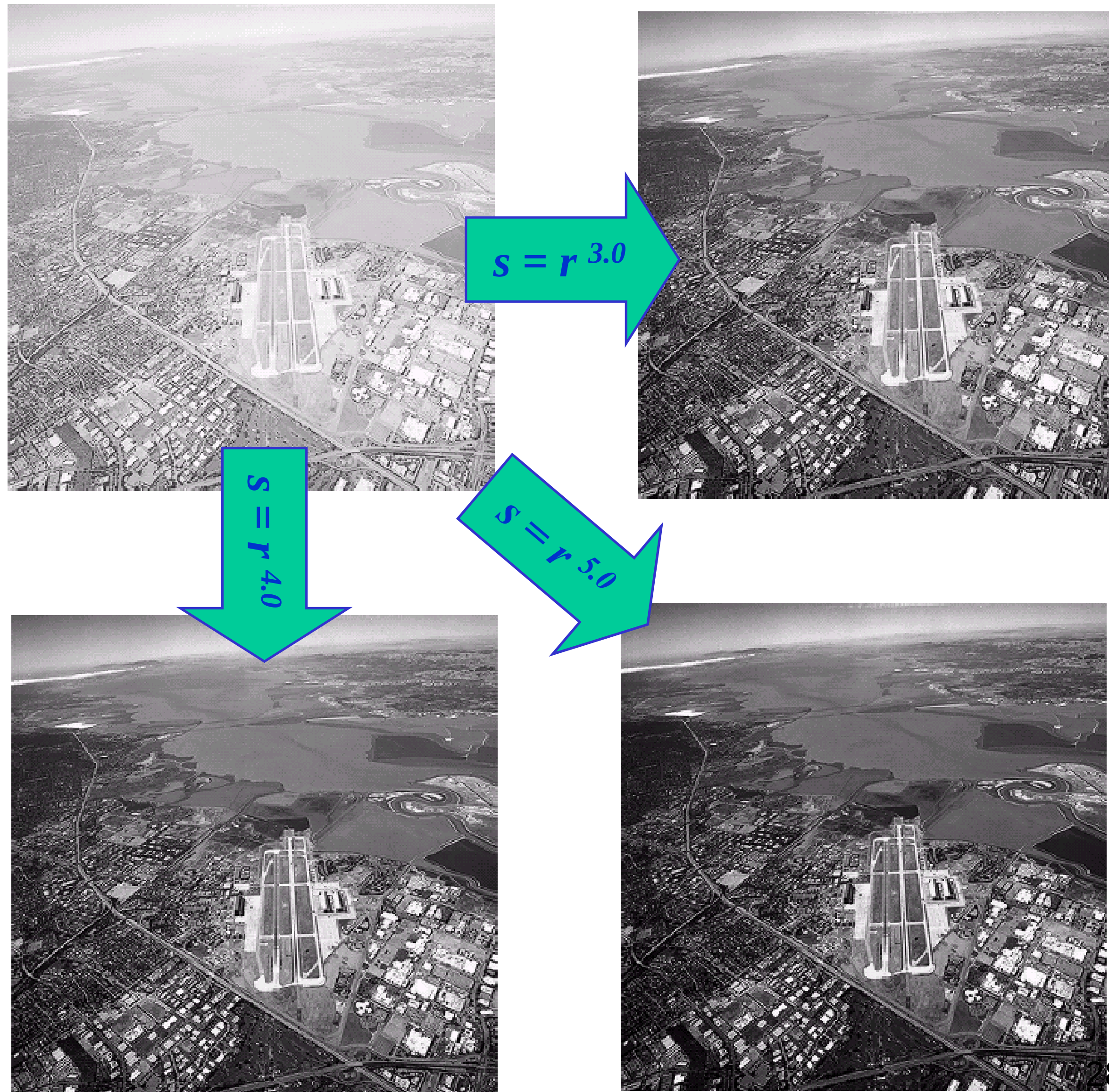
$\gamma = 5$

Power Law Transformations (cont…)

- Power law transforms are used to darken the image with better contrast

- Different curves highlight different detail

Figure 3.9



Power Law Example

Original Image

Figure 3.9(a)

a	b
c	d

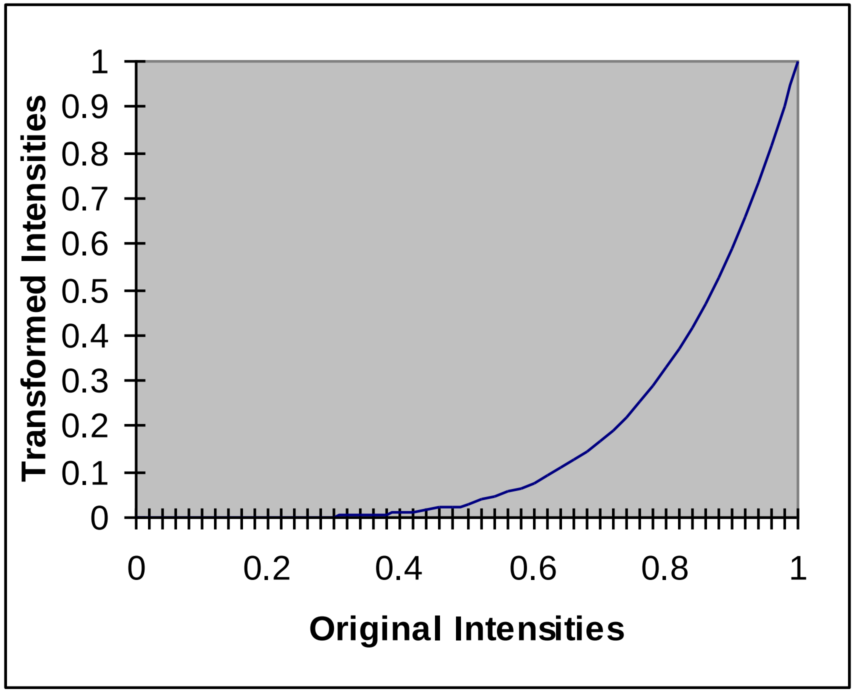
FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3-5) with $\gamma = 3.0, 4.0$, and 5.0, respectively. ($c = 1$ in all cases.) (Original image courtesy of NASA.)

Power Law Example (cont…)

Ariel image after Gamma Correction

Figure 3.9(d)

$$\gamma = 5.0$$



3.2.4 Piecewise Linear Transformation Functions

- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms
- Images show a **contrast stretching** linear transform to add contrast to a poor quality image

a b
c d

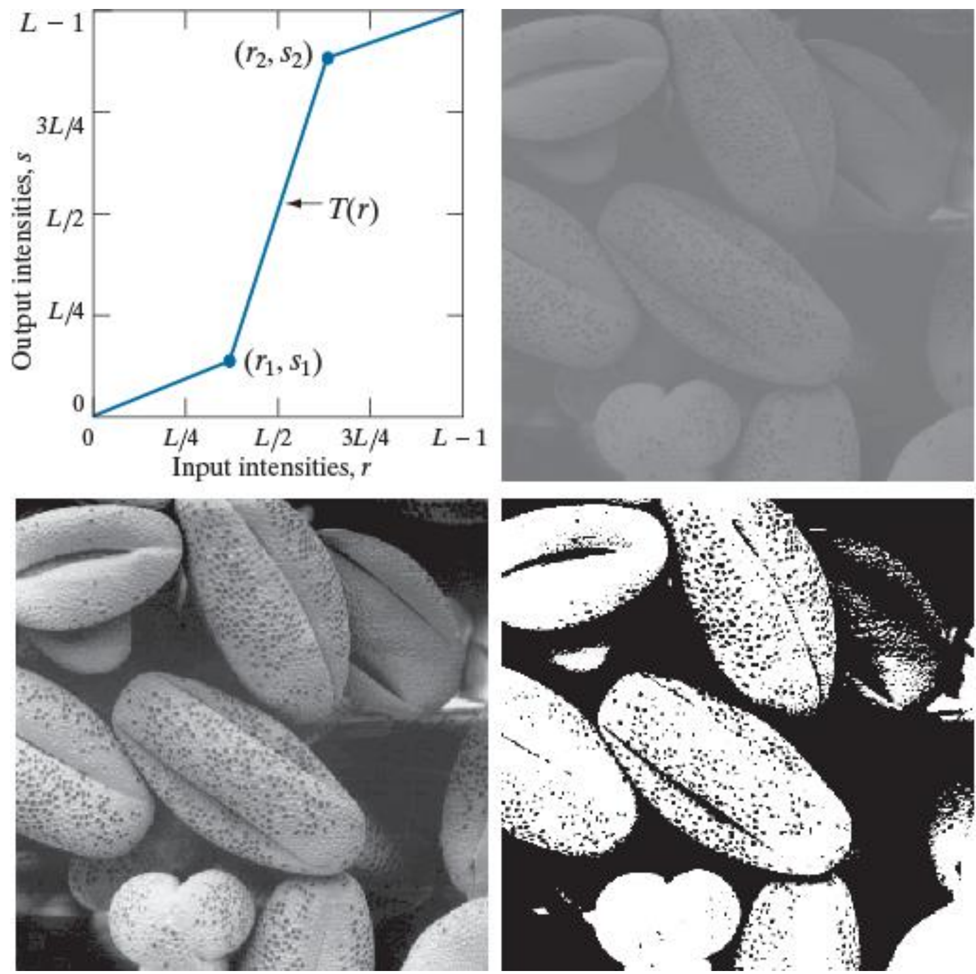
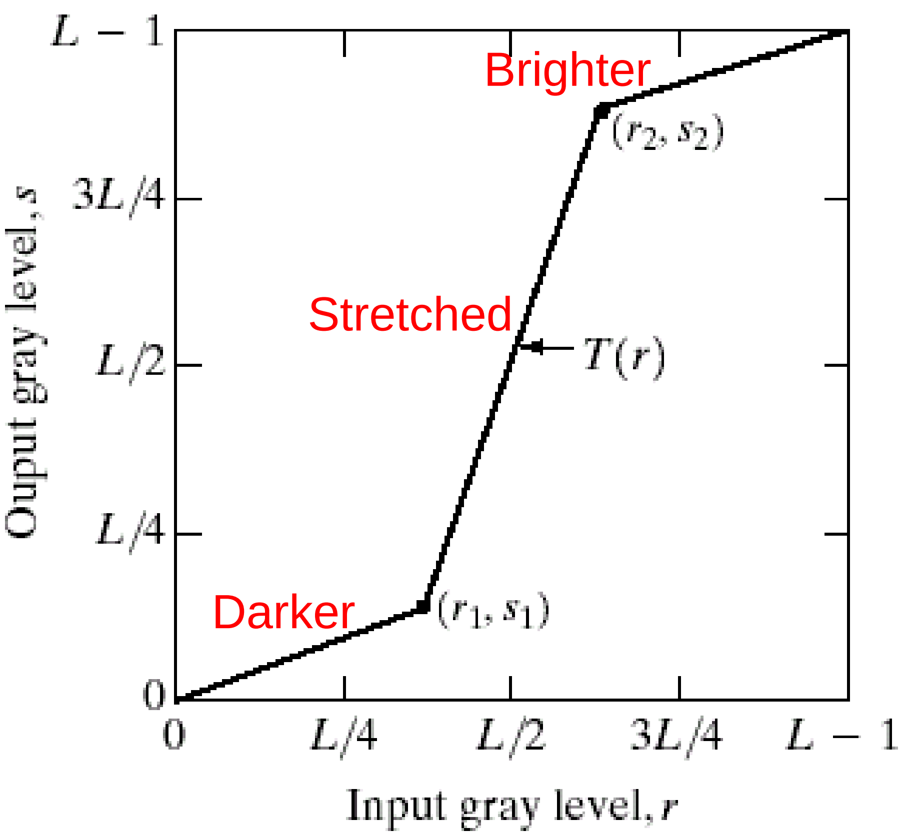


FIGURE 3.10 Contrast stretching. (a) Piecewise linear transformation function. (b) A low-contrast electron microscope image of pollen, magnified 700 times. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

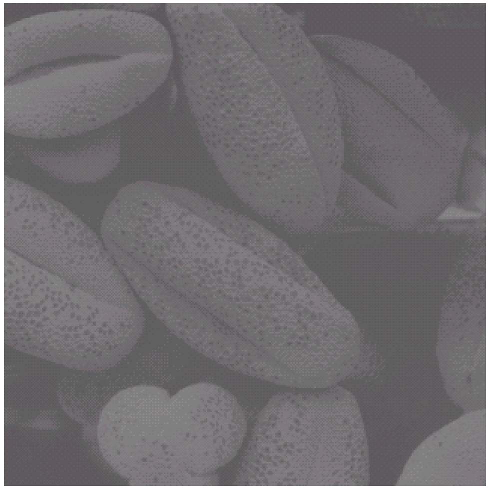
Contrast Stretching

Contrast: The difference between the brightest and darkest intensities

Figure 3.10



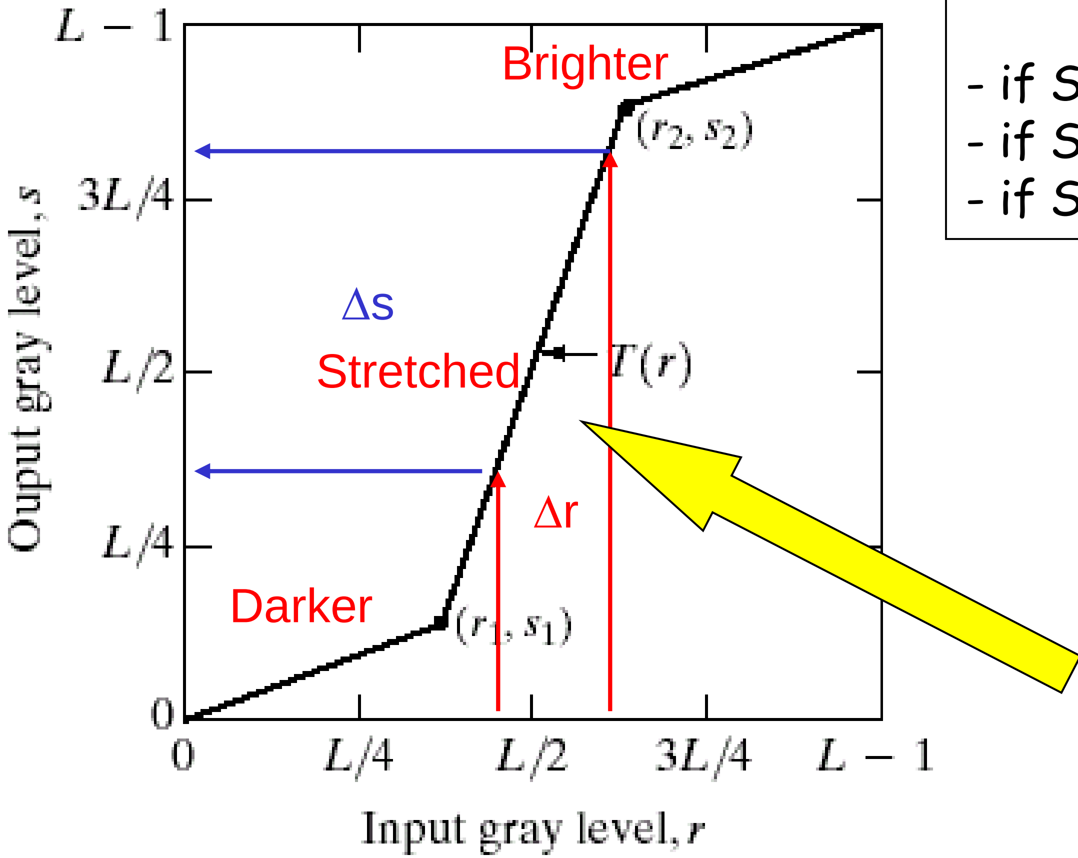
Before contrast enhancement



After

Contrast Enhancement: Slope of $T(r)$

Figure 3.10



Slope of $T(r)$ is the Key:

- if Slope > 1 → Contrast increases
- if Slope < 1 → Contrast decrease
- if Slope = 1 → no change

Smaller Δr yields wider Δs
→ Increased Contrast

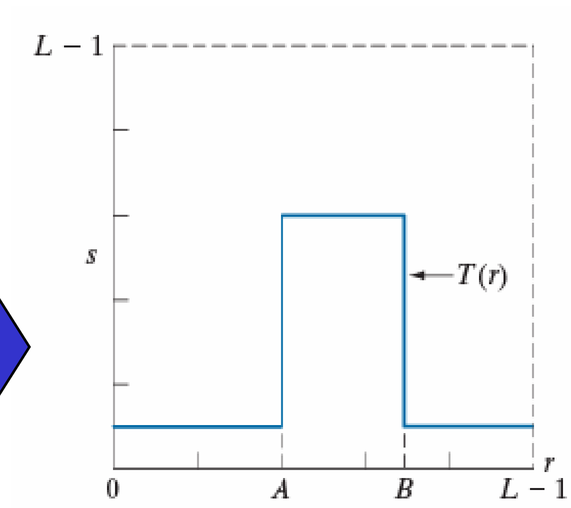
Intensity Level Slicing

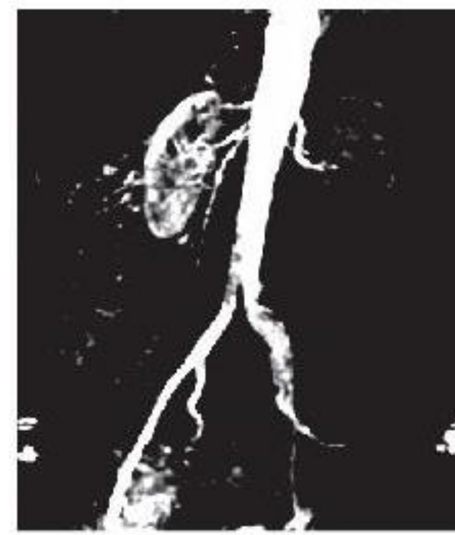
- Highlights a specific range of grey levels
  - Similar to thresholding
  - Other levels can be suppressed or maintained
  - Useful for highlighting features in an image

a b

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

Figure 3.11 (a)





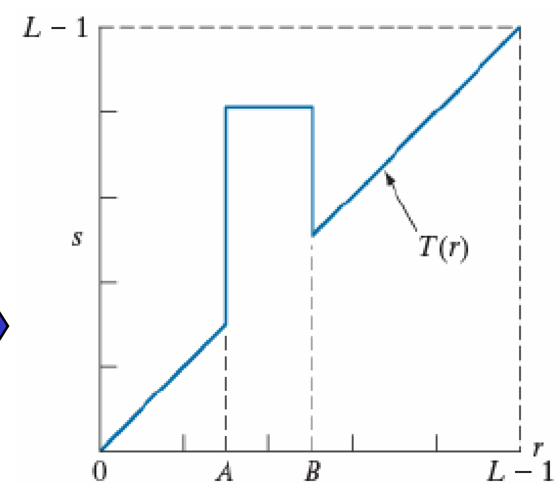
Intensity Level Slicing

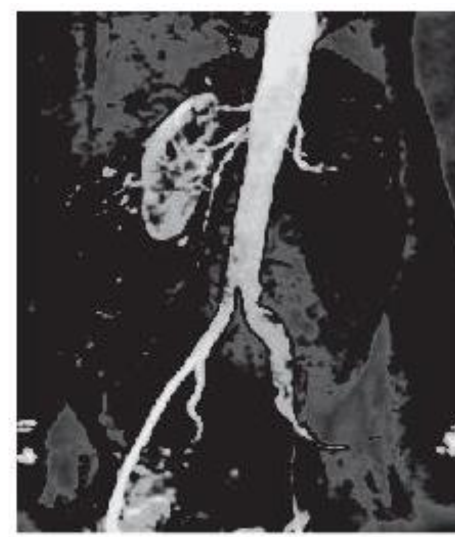
- Highlights a specific range of grey levels
  - Similar to thresholding
  - Other levels can be suppressed or maintained
  - Useful for highlighting particular features in an image

a b

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

Figure 3.11 (b)





Bit Plane Slicing

- Isolate particular bits of pixel values to highlight interesting aspects of an image
  - Higher-order bits usually contain most of the significant visual information
  - Lower-order bits contain subtle details

FIGURE 3.13 Bit-planes of an 8-bit image.

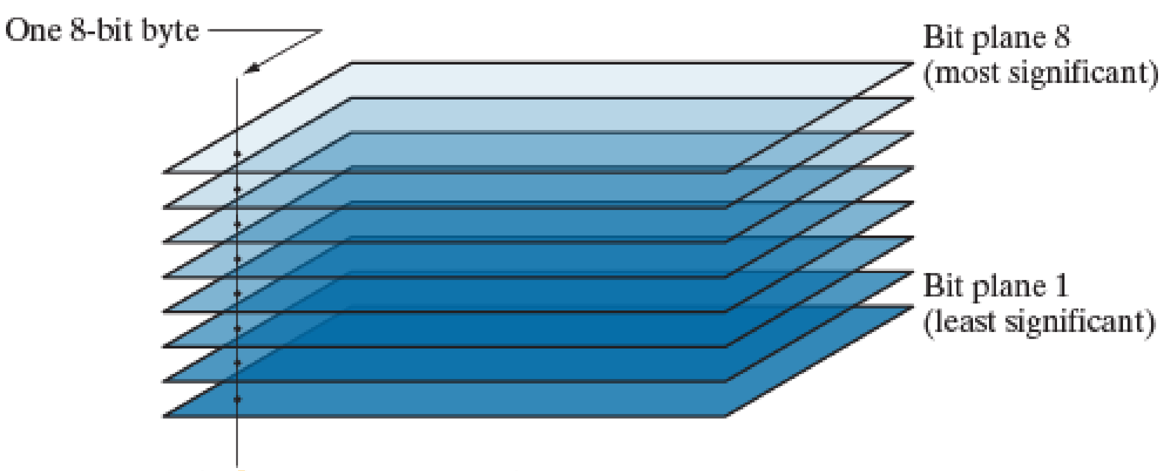
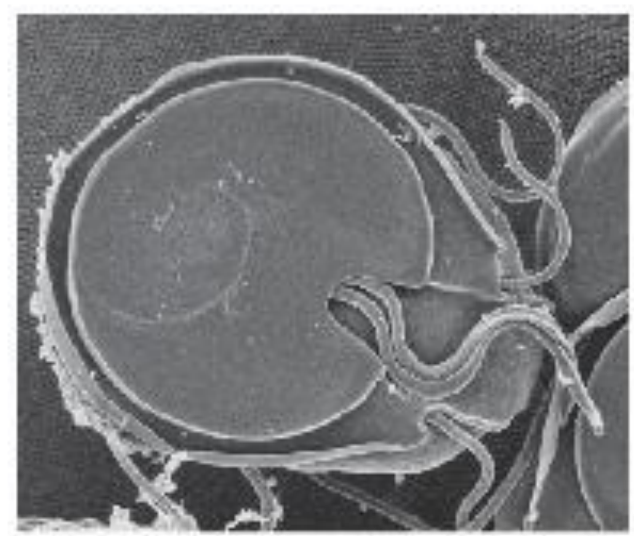


FIGURE 3.14 (a) An 8-bit gray-scale image of size 837×988 pixels.



WRIGHT STATE UNIVERSITY

Bit Plane Slicing (cont…)

Figure 3.14

[10000000]

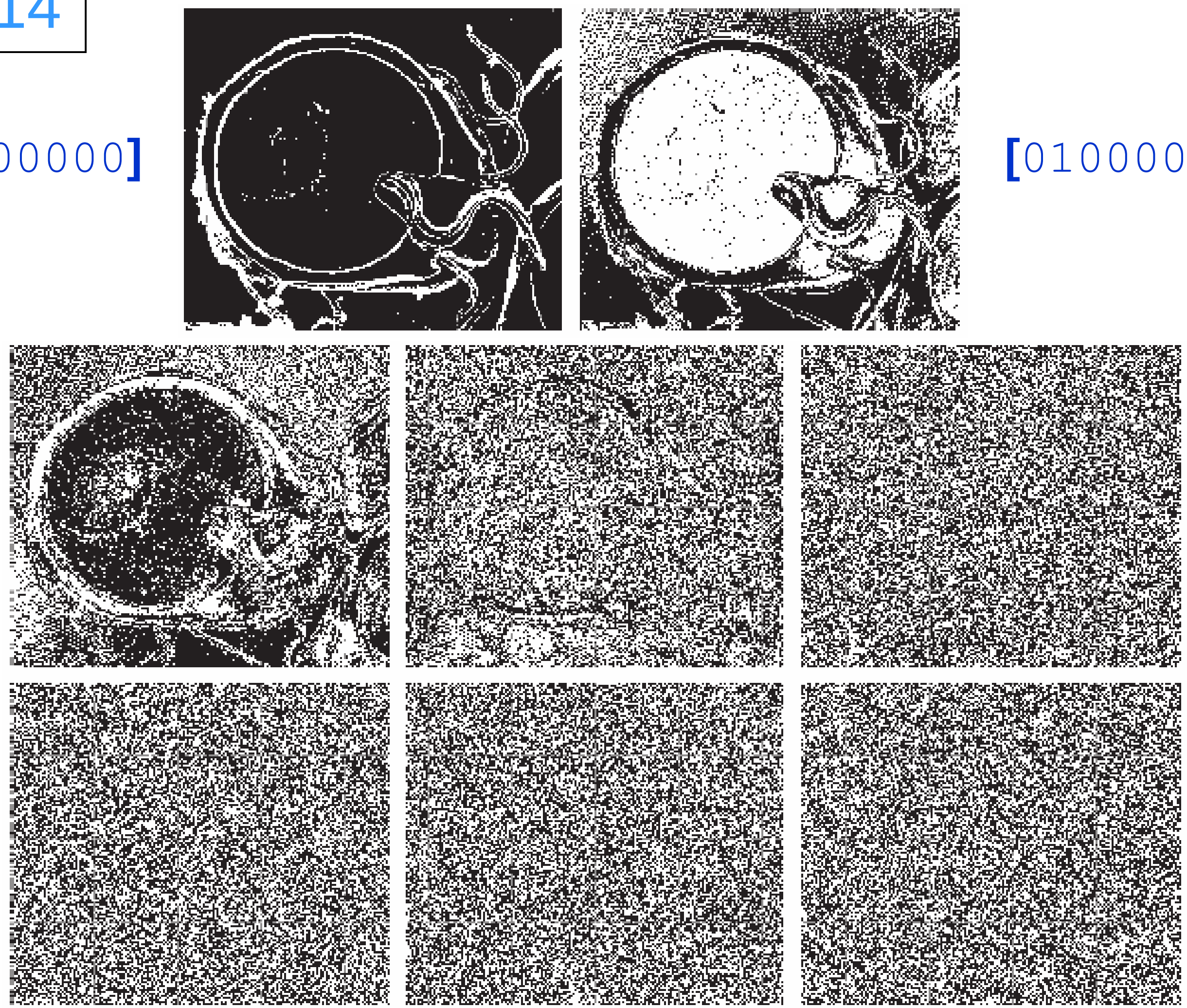
[01000000]

[00100000]

[00001000]

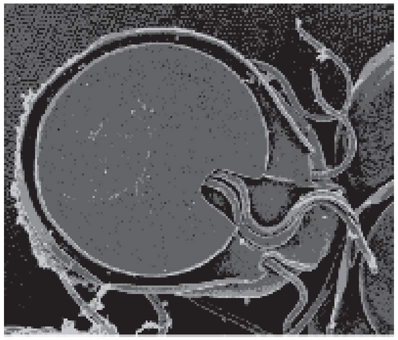
[00000100]

[00000001]



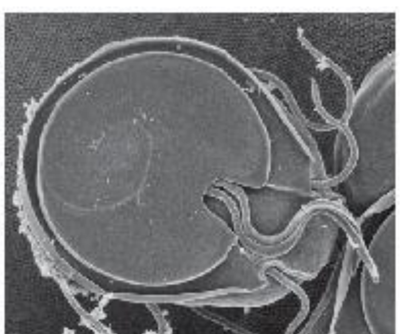
Bit Plane Slicing (cont…)

Figure 3.15



Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6



Reconstructed image using only bit planes 8, 7, 6 and 5

a b c

FIGURE 3.15 Image reconstructed from bit planes: (a) 8 and 7; (b) 8, 7, and 6; (c) 8, 7, 6, and 5.

More Bit Plane Slicing Examples

Figure 3.14

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.

3rd Edition

Bit Plane Slicing (cont…)

Figure 3.14 (a)

3rd Edition

Bit Plane Slicing (cont…)

Figure 3.14 (b): Bit plane 1

3rd Edition

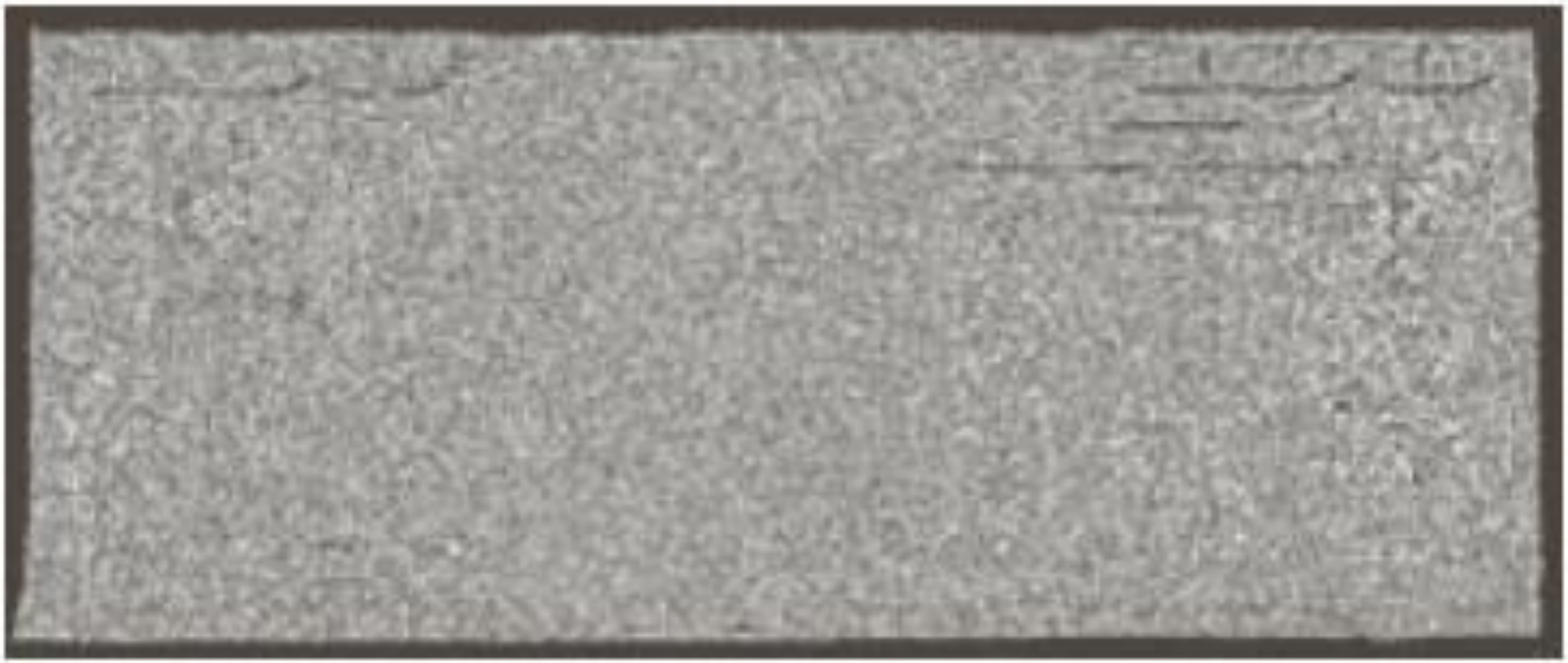
Bit Plane Slicing (cont…)

Figure 3.14 (c): Bit plane 2

3rd Edition

Bit Plane Slicing (cont…)

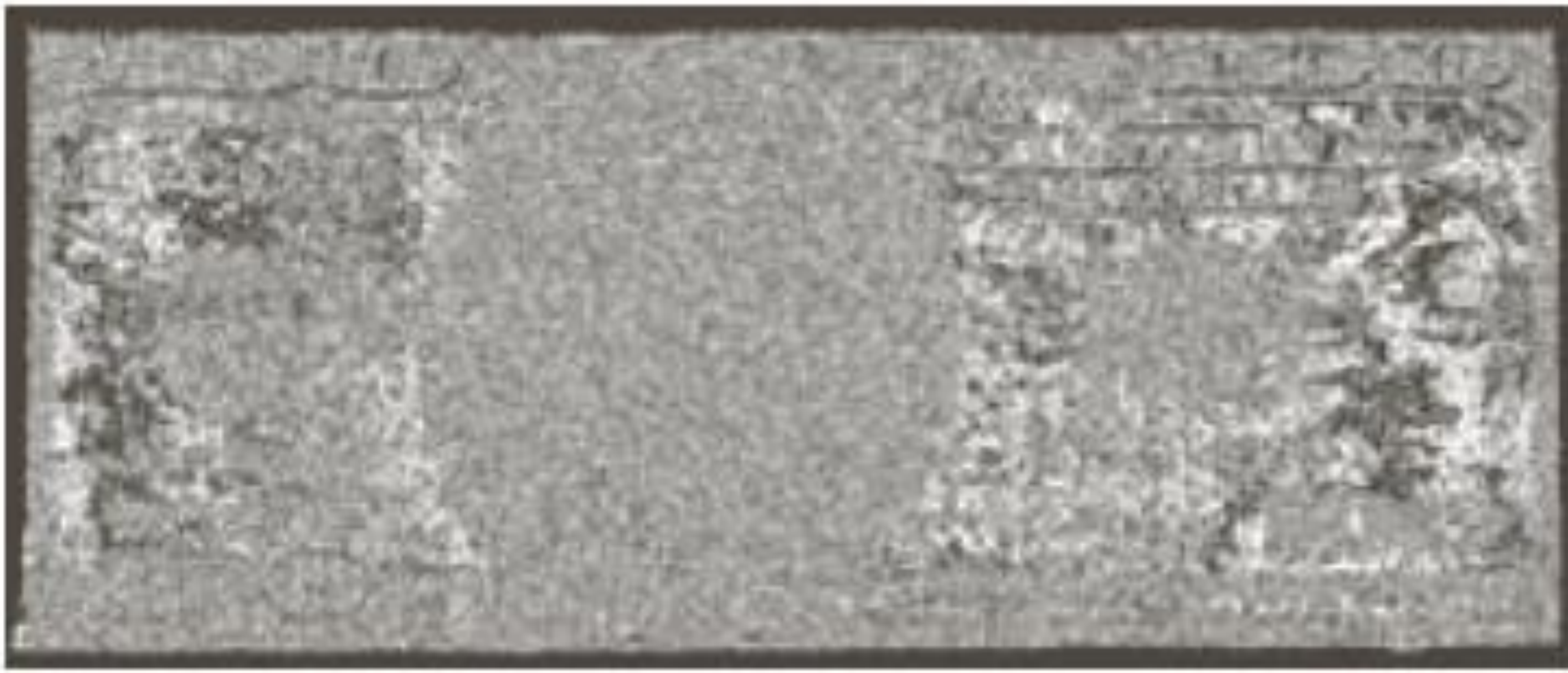
Figure 3.14 (d): Bit plane 3



3rd Edition

Bit Plane Slicing (cont…)

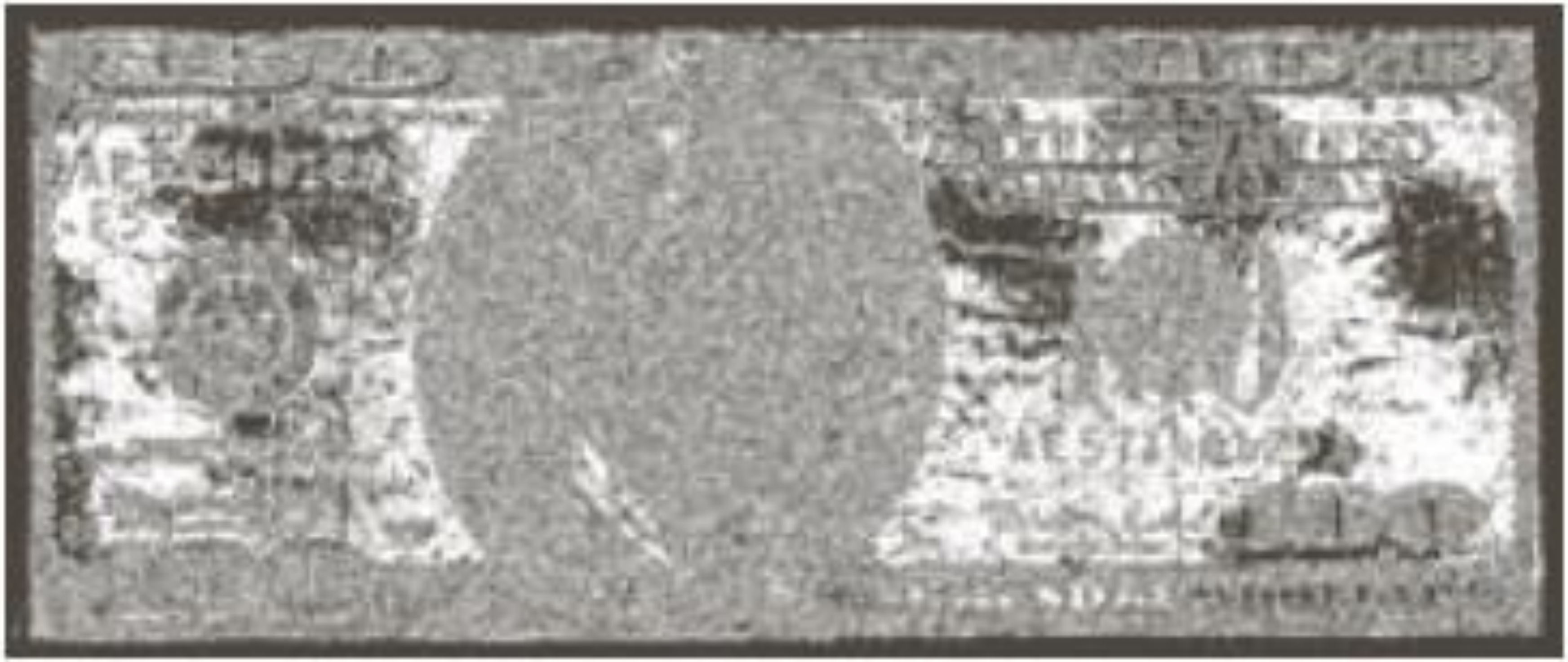
Figure 3.14 (e): Bit plane 4



3rd Edition

Bit Plane Slicing (cont…)

Figure 3.14 (f): Bit plane 5



3rd Edition

Bit Plane Slicing (cont…)

Figure 3.14 (g): Bit plane 6

Bit Plane Slicing (cont…)

Figure 3.14 (h): Bit plane 7

3rd Edition

Bit Plane Slicing (cont…)

Figure 3.14 (i): Bit plane 8

Bit Plane Slicing (cont…)

Figure 3.14

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

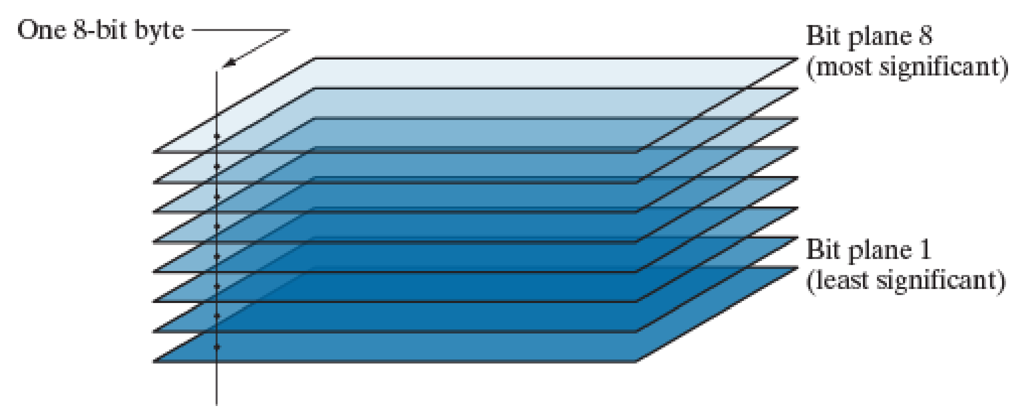
Reconstructed image using only bit planes 8,7, 6 and 5

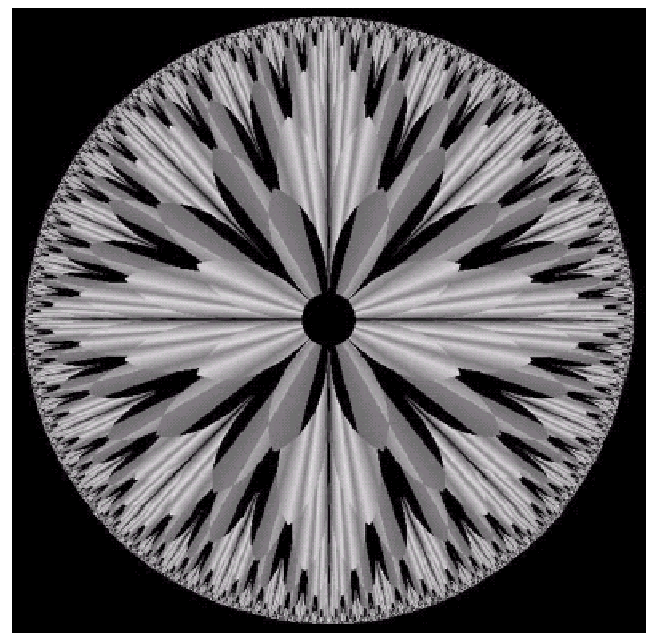
3rd Edition

Additional Bit Plane Slicing Example

- Isolate particular bits of pixel values to highlight interesting aspects of an image
  - Higher-order bits usually contain most of the significant visual information
  - Lower-order bits contain subtle details

FIGURE 3.13
Bit-planes of an 8-bit image.

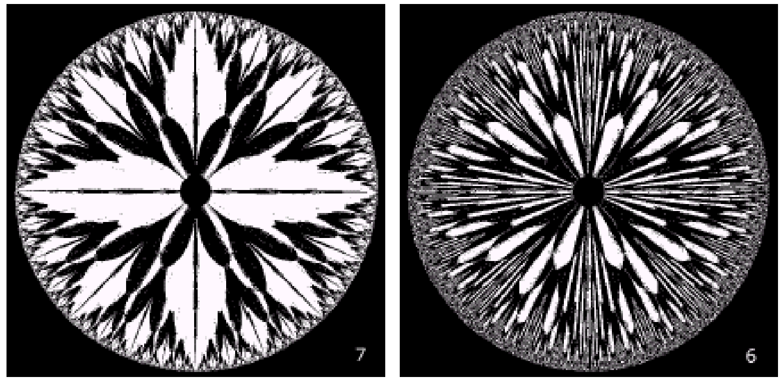




2nd Edition

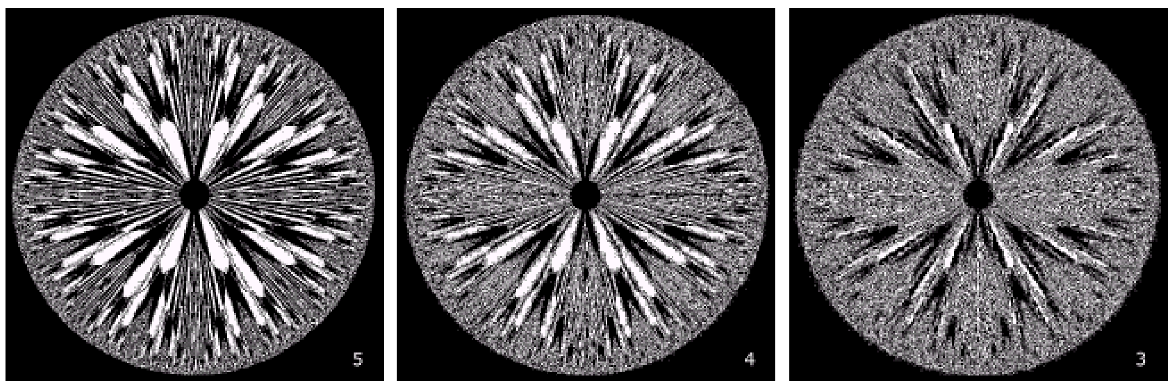
Bit Plane Slicing (cont…)

[10000000]



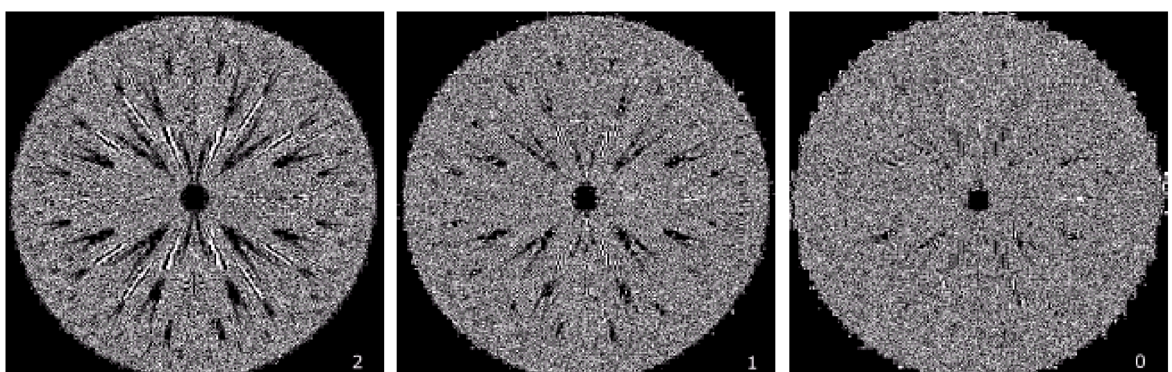
[01000000]

[00100000]



[00001000]

[00000100]



[00000001]

2nd Edition

Summary

- We have looked at different kinds of point processing image enhancement
- Next time we will consider neighbourhood operations – in particular *filtering* and *convolution*

Chapter-3, Part-2

3.3 Histogram Processing

What Is Image Enhancement?

- Image Enhancement: The process of making images more useful

- Objectives of Image Enhancement include:
  - Highlighting interesting detail in images
  - Removing noise from images
  - Making images more visually appealing

Image Enhancement Examples

Image Enhancement Examples (cont…)

Image Enhancement Examples (cont…)

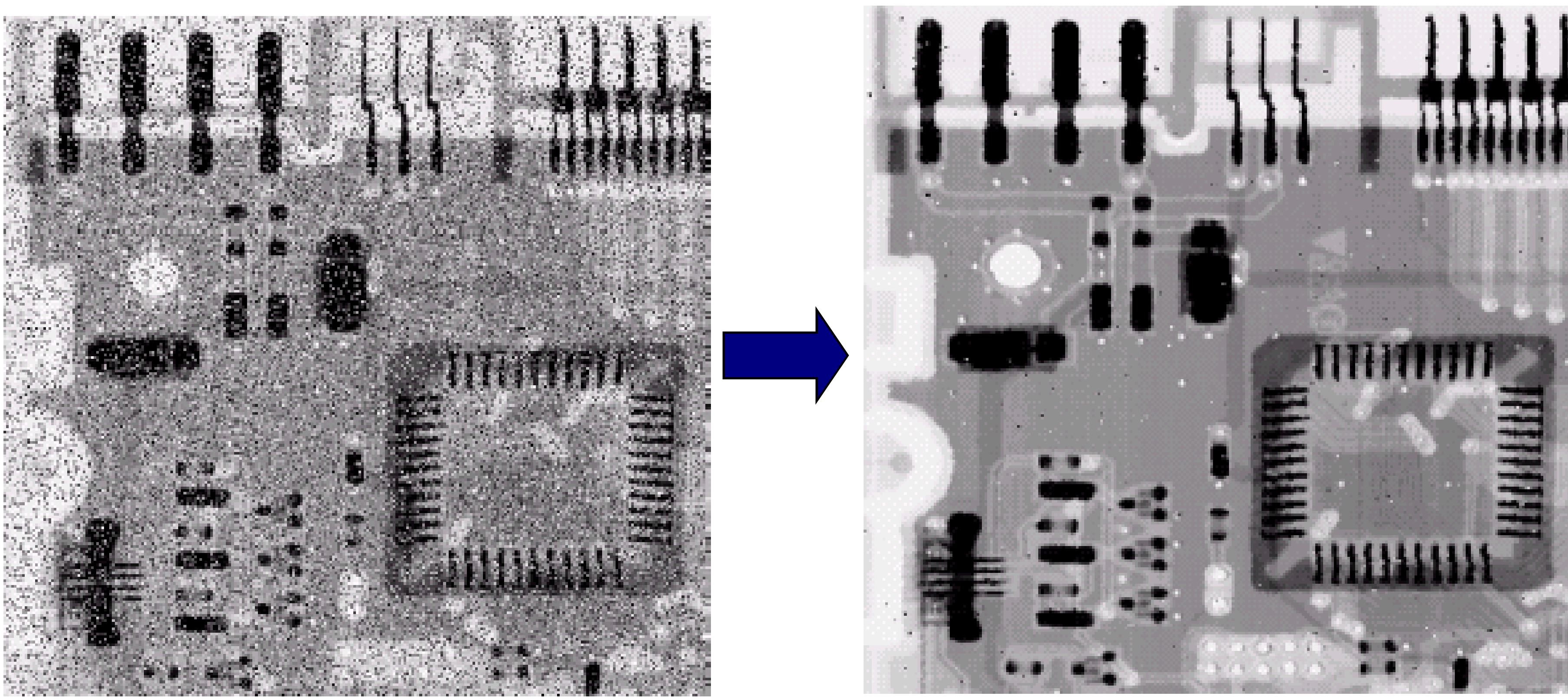


Image Enhancement Examples (cont…)

Image Histograms

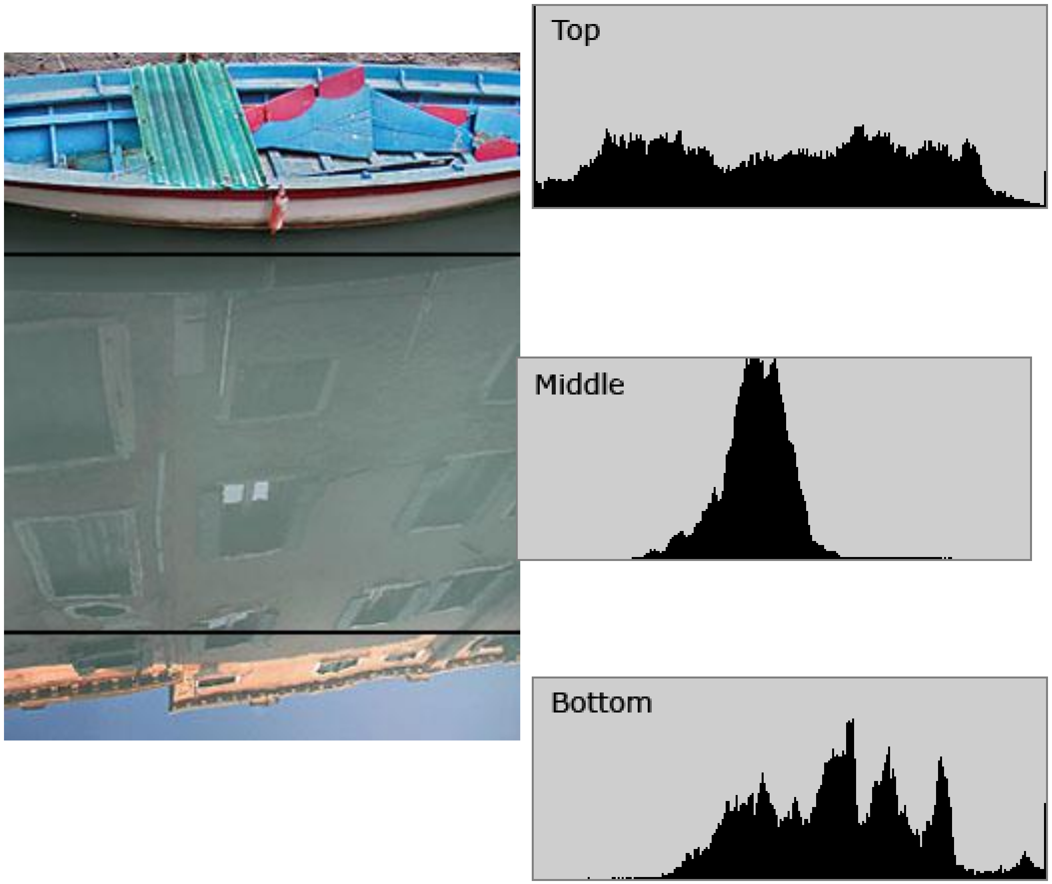
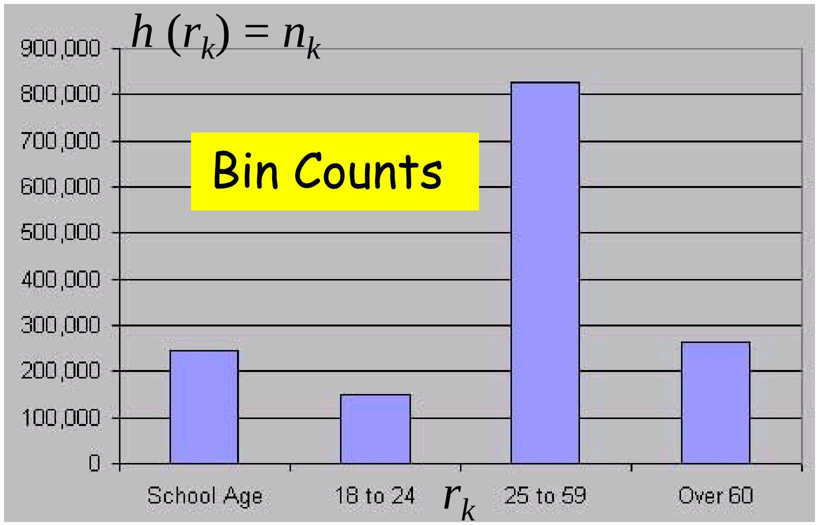
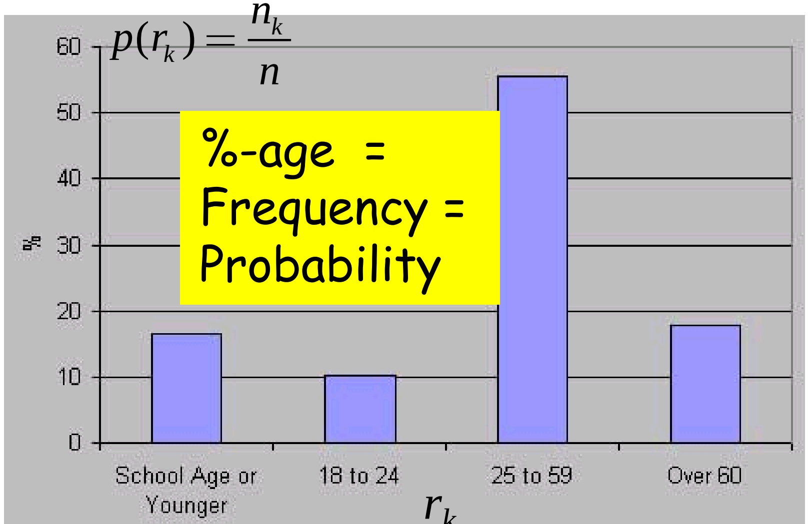


Image histogram:

- Shows the distribution of grey levels in the image
- A graphical representation of the number of image pixels (Y-axis) as a function of intensity (X-axis)

Histogram Concept

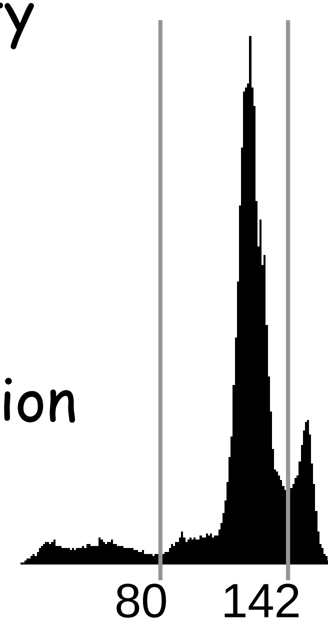




- Histogram
  - Made up of *bins*
  - Each bin: Represents certain range of values
- Height of a bin
  - Number of objects associated with the *bin* (= Bin Count)
  - % Probability of objects associated with the *bin*
    - Probability Mass Function
    - Approximation of PDF

Steps for constructing an Image Histogram

- **Steps for Image Histogram:**
  - Find the Max and Min intensities among all pixels in the image
  - Divide the range of intensities into a number of bins
  - Assign each pixel to a bin depending on the pixel intensity
  - Count the no. of pixels in each bin (i.e., how many pixels have values in that range)
- **Normalized histogram:**
  - Gives the probability distribution of different pixel values



- The seal image shows three image regions:
- pixels below *T1* = 80
- those above *T2* = 142
- those between the two

intensity

Histogram Examples

- Histogram (Bin-Count) of a digital image with intensity levels in the range $[0, L\text{-}1]$:

$$h(r_k) = n_k,$$

where n_k is the number of pixels having intensity r_k

Normalized histogram of an $M \times N$ Image:

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

Histogram Examples

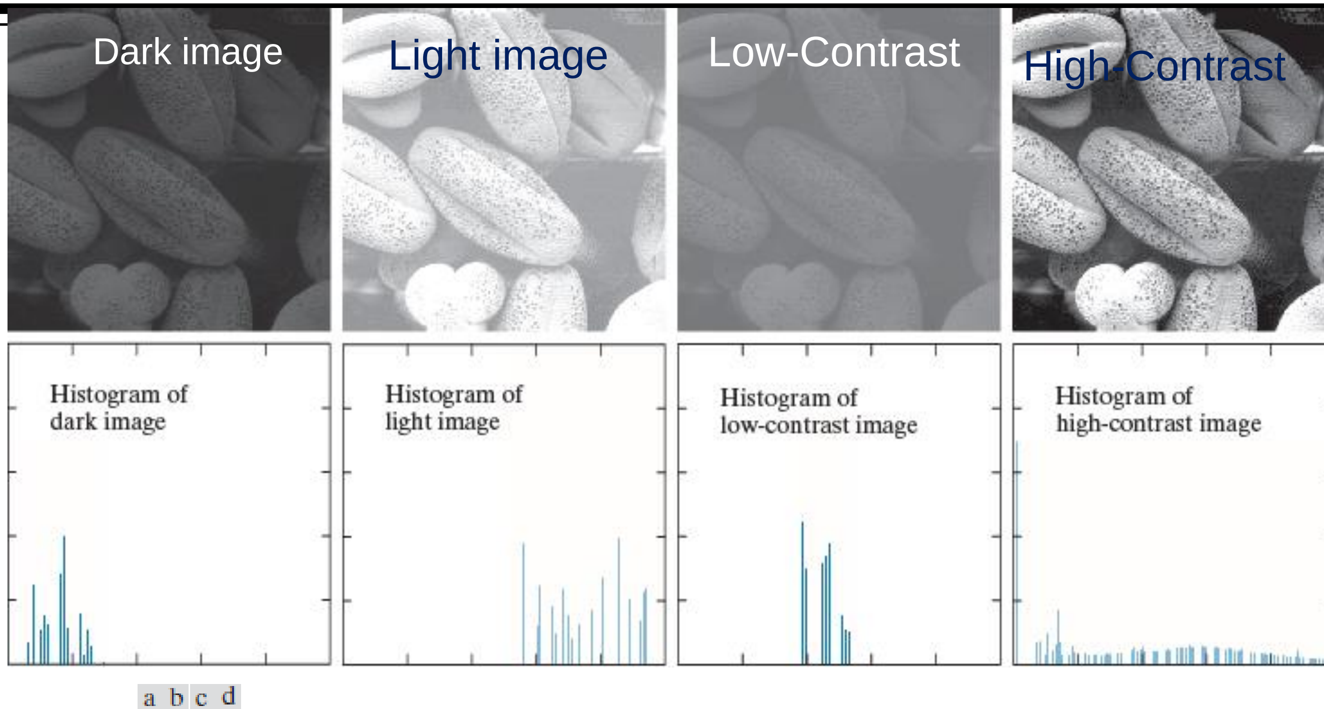


FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

- Notice the relationships between the images and their histograms
- High contrast image has the most evenly spaced histogram

3.3 Normalized Histogram

Normalized histogram of an $M \times N$ Image:

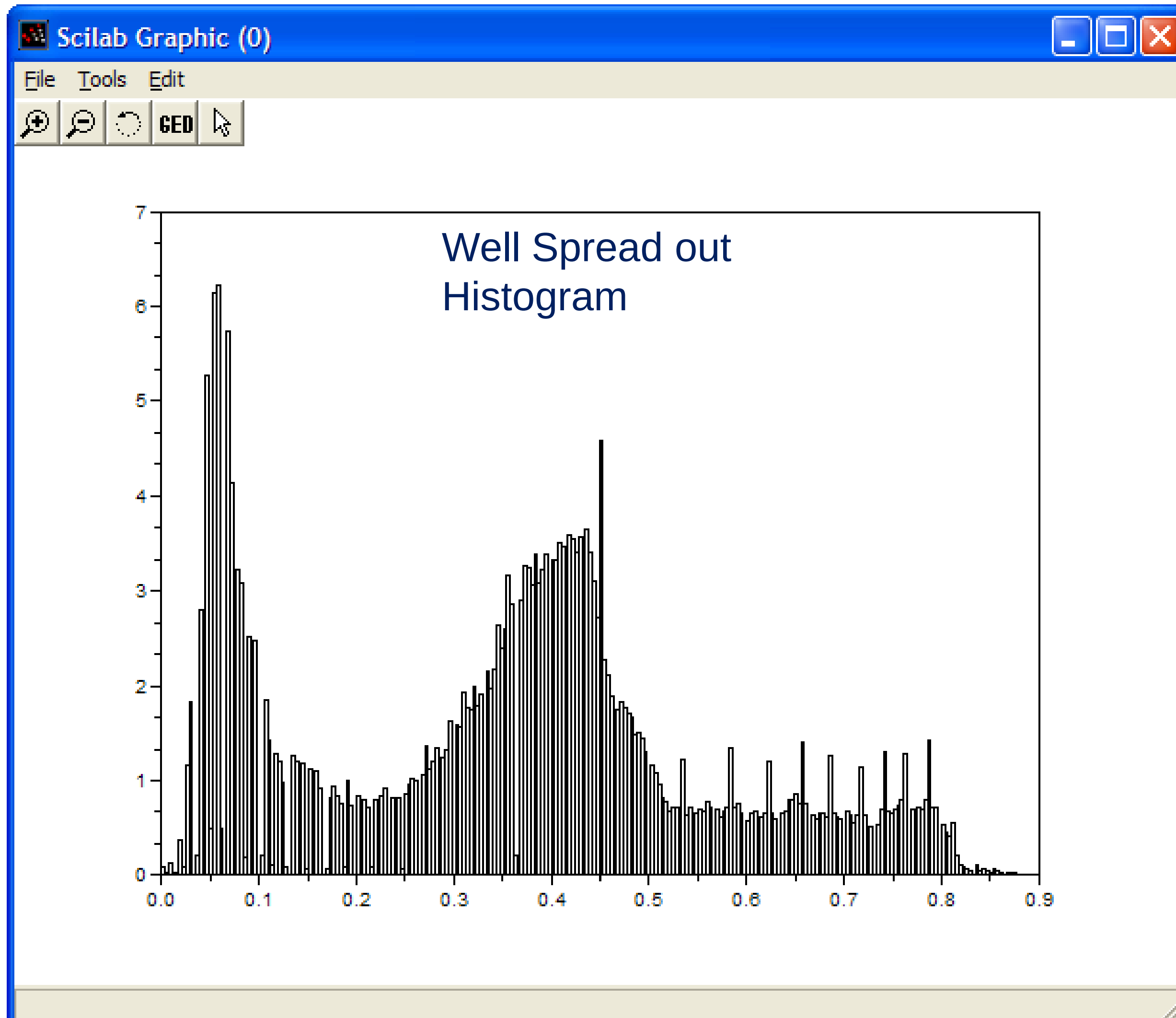
$$p(r_k) = \frac{n_k}{n} = \frac{h(r_k)}{MN} \;; \quad k = 0,1,\ldots, L\text{-}1$$

- r_k : k-th intensity level or gray level
- $n_k = h(r_k)$: Histogram Bin-Count (i.e., number of pixels) in the image with gray level r_k at k-th intensity level
- $n = M \times N$: Total number of pixels in the image
- $p(r_k)$: Estimate of probability of occurrence of gray-level r_k

Histogram Examples

- Contains a variety of intensity levels

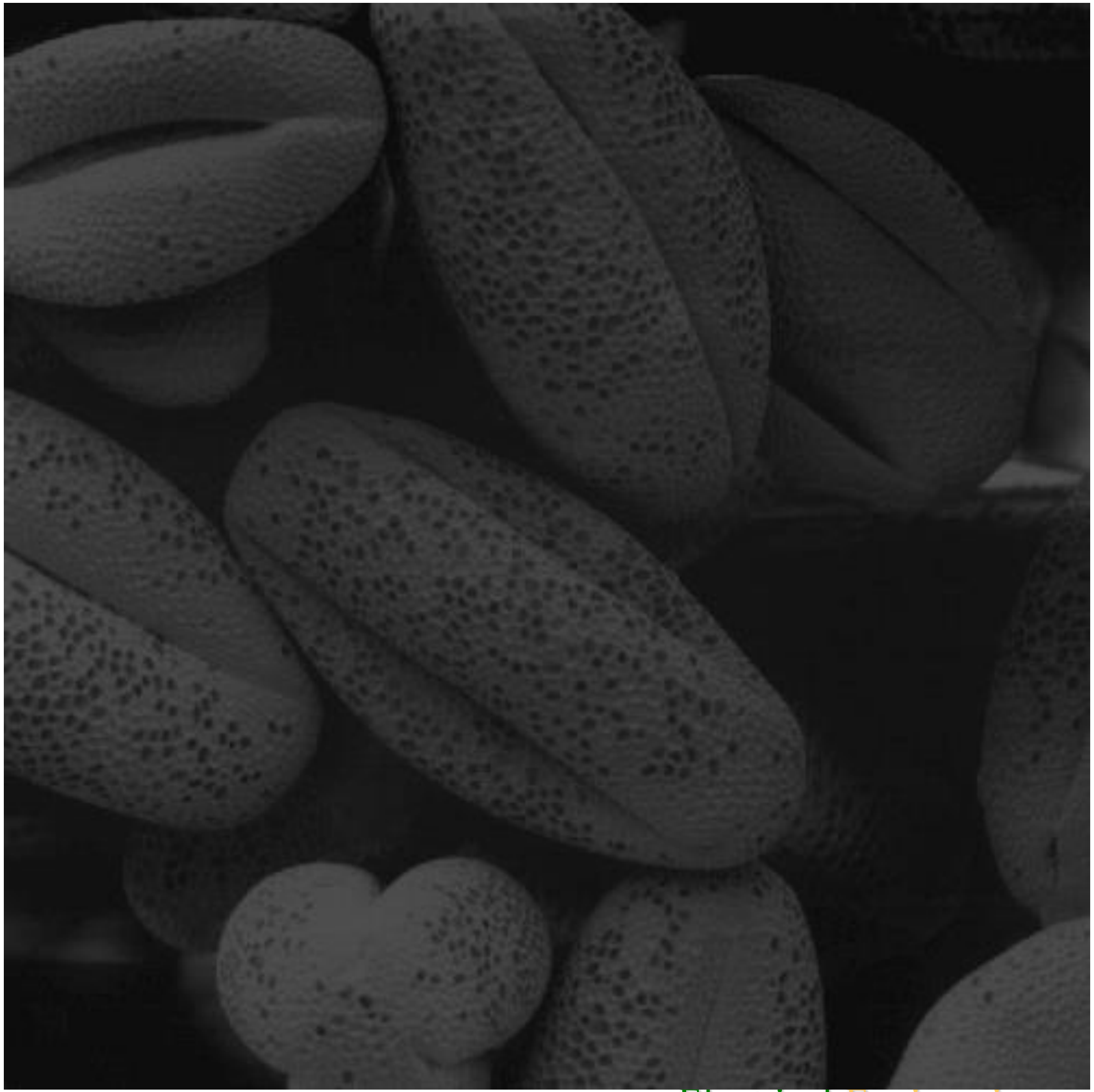
Histogram Examples (cont…)



- Contains a variety of intensity levels

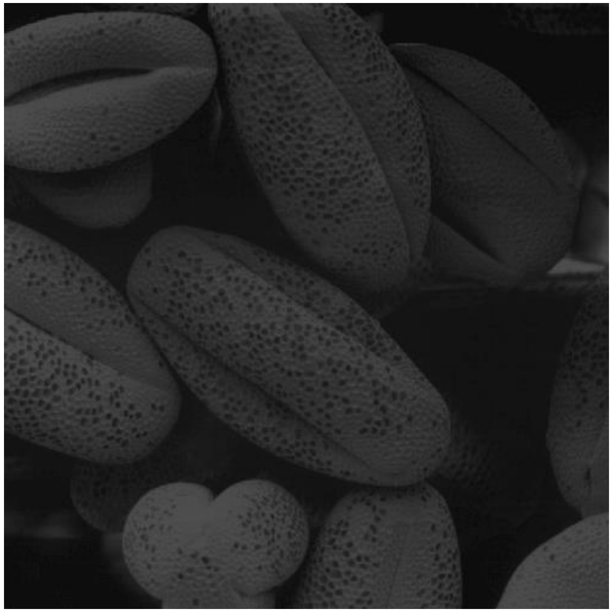
Histogram Examples (cont…)

Figure 3.16 (a)



- Contains mostly Dark (small-valued) intensity levels

Histogram Examples (cont...)



• Contains mostly Dark (small-valued) intensity levels

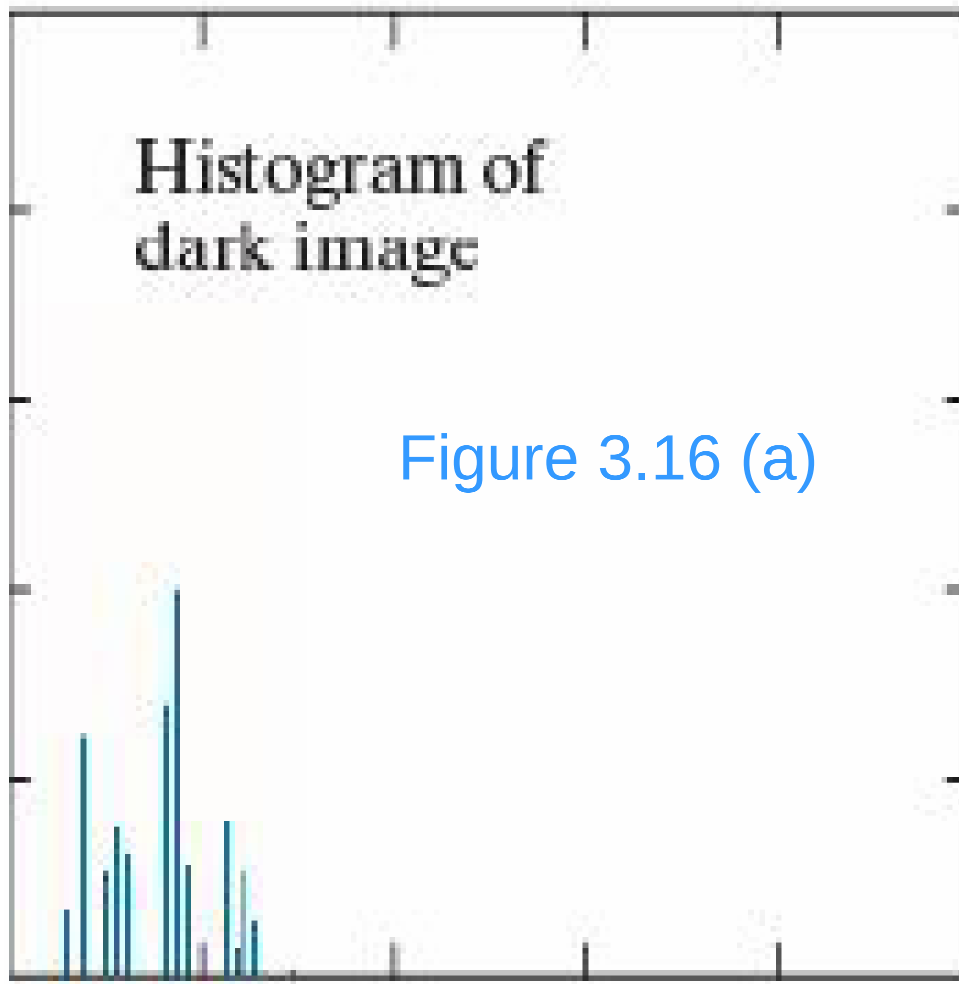


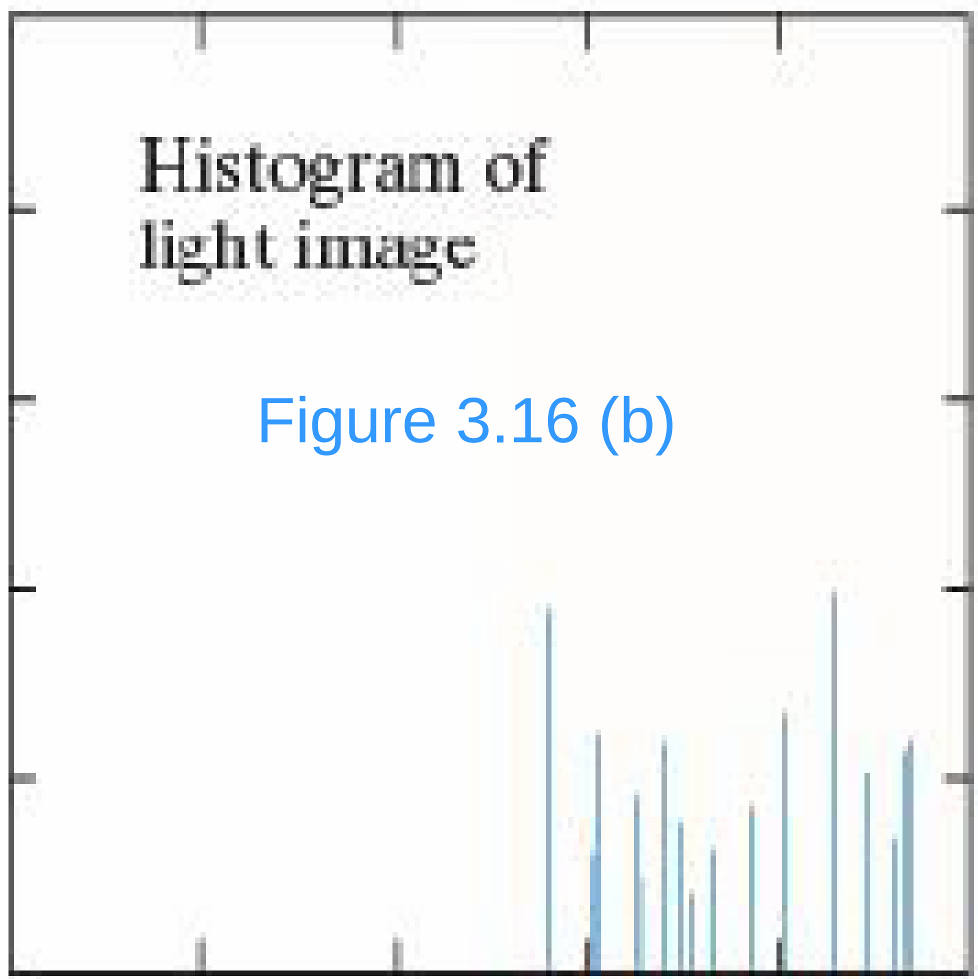
Figure 3.16 (a)

Histogram Examples (cont…)

Figure 3.16 (b)

- Contains mostly bright (higher-valued) intensity levels

Histogram Examples (cont…)

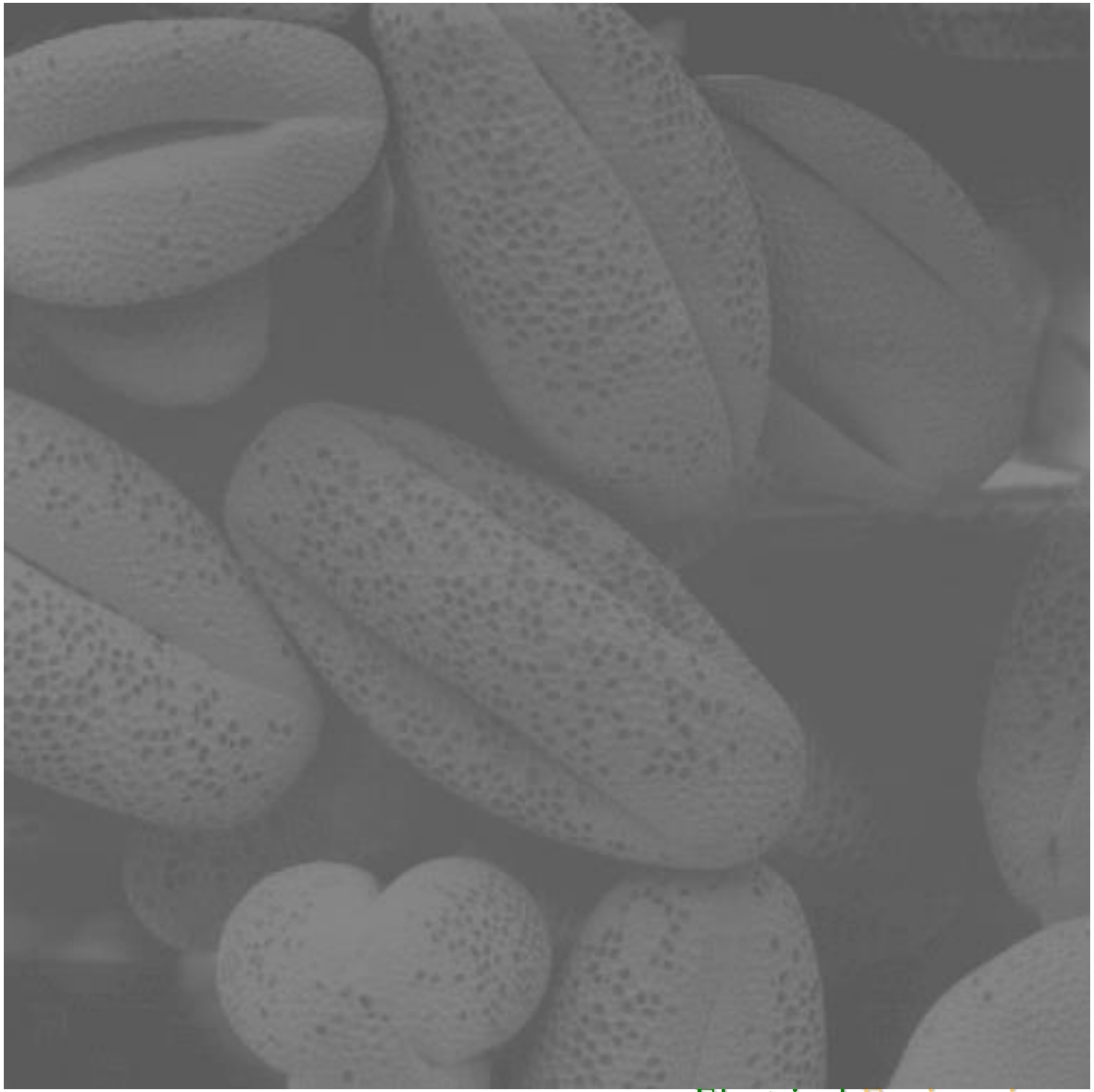


- Contains mostly bright (i.e., higher) intensity levels

Histogram Examples (cont…)

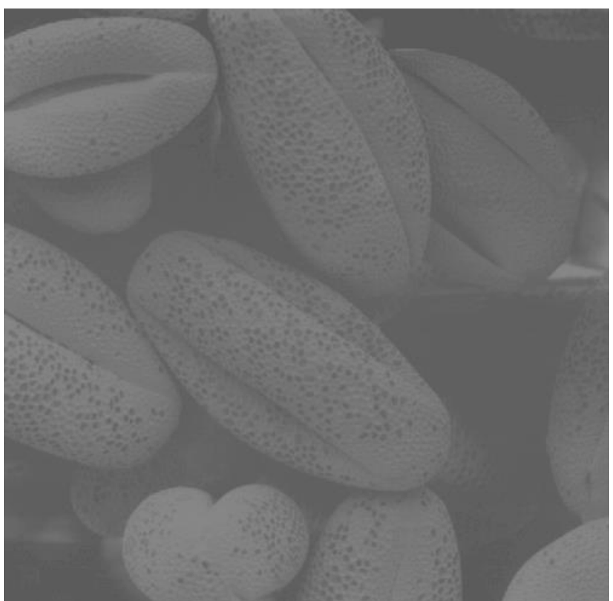
- **Low-Contrast Image:** Contains neither dark nor bright intensity values

Figure 3.16 (c)



Histogram Examples (cont…)

• **Low-Contrast Image:** Contains neither dark nor bright intensity values



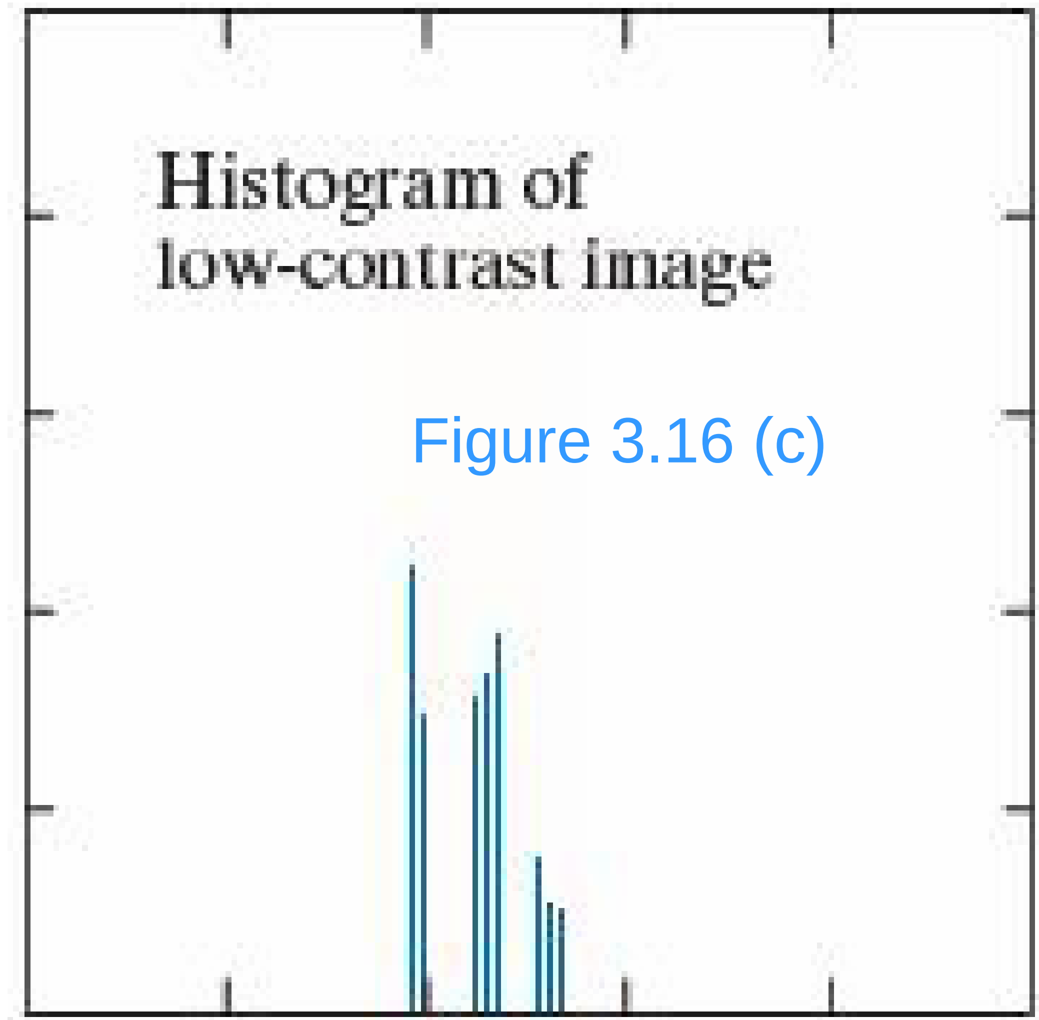


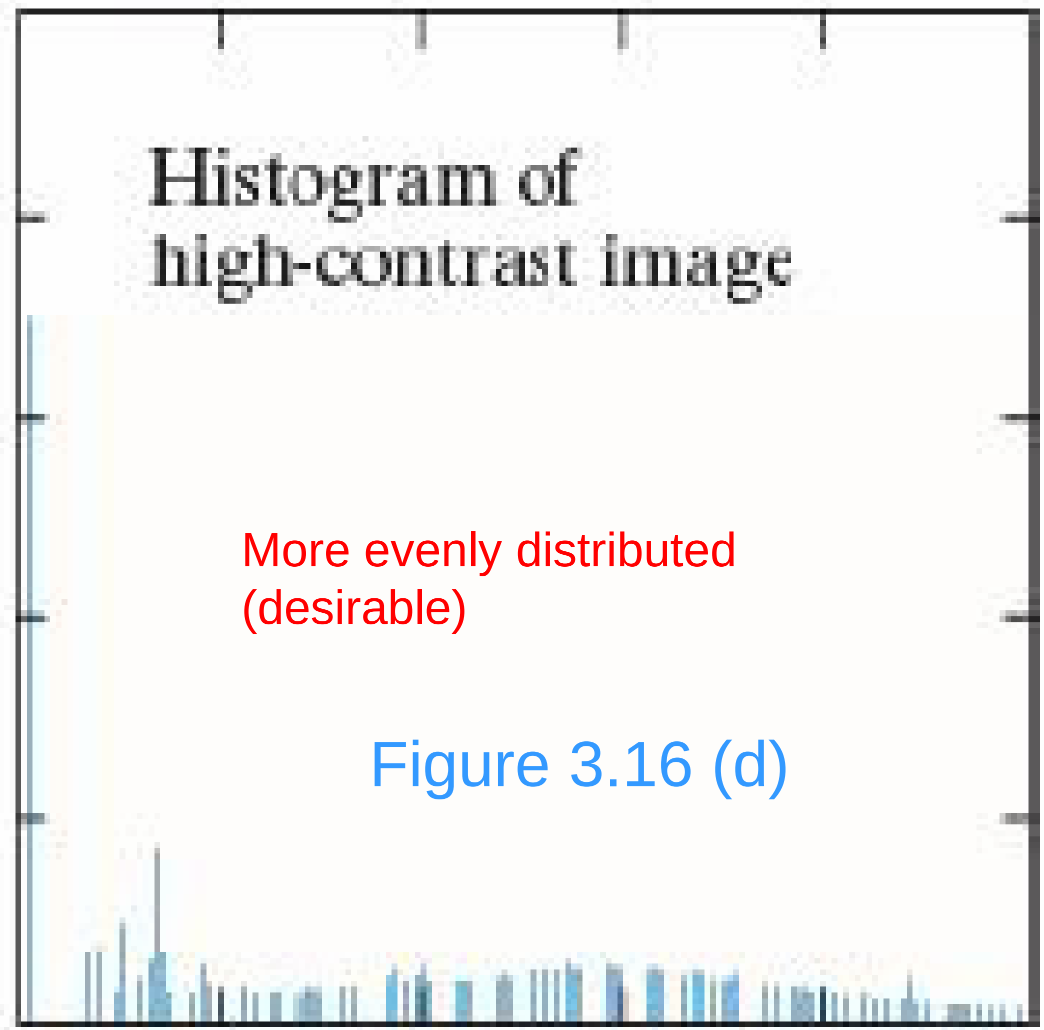
Figure 3.16 (c)

Histogram Examples (cont…)

- **High-Contrast Image:**
Contains dark, bright and mid-level intensity values

Figure 3.16 (d)

Histogram Examples (cont…)



- **High-Contrast Image:** Contains dark, bright and mid-level intensity values

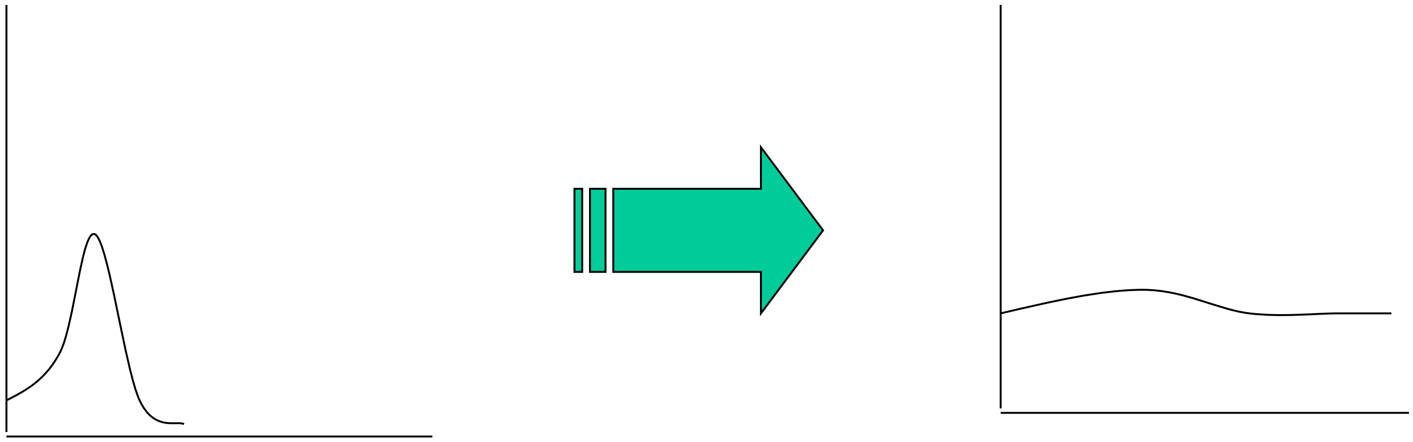
Contrast Stretching

- **Rationale:** Images with poor contrast can be improved by applying a relatively simple contrast modification. i.e.,
  - Stretch the histogram

- **Question:** How do we decide on the transformation function?

3.3 Histogram Processing

- **Intensity Transformation**
  - Modify current histogram to yield desired histogram
- **Histogram Equalization**
  - To make histogram distributed uniformly



- **Histogram Matching**
  - To make histogram to have desired shape

A Note About Grey Levels

- So far image grey level values are in the range [0, 255]
  - 0 → black
  - 255 → white
- There is no reason to use this range in all cases
  - The range [0,255] stems from display technologies
- For many image processing operations in this lecture grey levels are assumed to be in the range [0.0, 1.0]

3.3.1 Histogram Equalization

- Histogram Equalization
  - To improve the contrast of an image
  - To transform an image in such a way that the transformed image has a nearly uniform distribution of pixel values
- Transformation
  - Assume r is in the range $[0, L\text{-}1]$, with $r = 0$ representing black and $r = L\text{-}1$ representing white, and intensity mappings of the form:

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

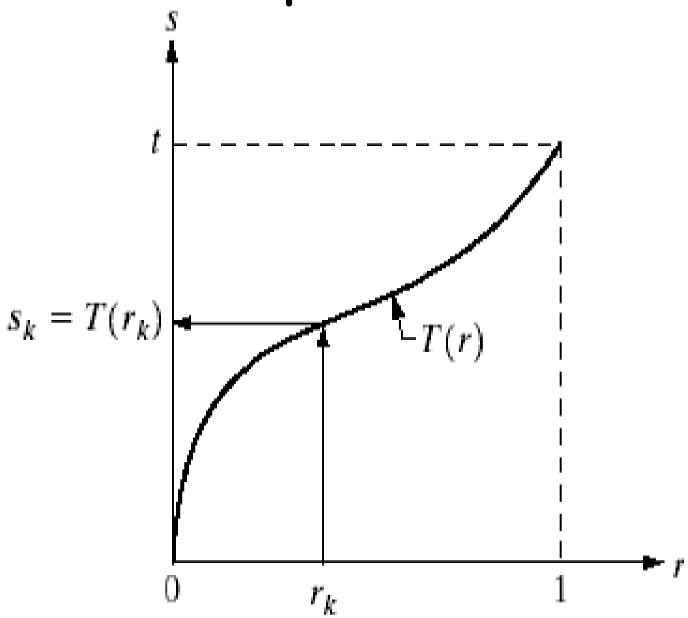
  - The transformation function satisfies two conditions:

1) $T(r)$: Single-valued & **monotonically increasing** in $0 \le r \le L\text{-}1$ and

2) $0 \le T(r) \le 1 \quad \text{for} \quad 0 \le r \le L-1$

Histogram Equalization (contd.)

- Example:



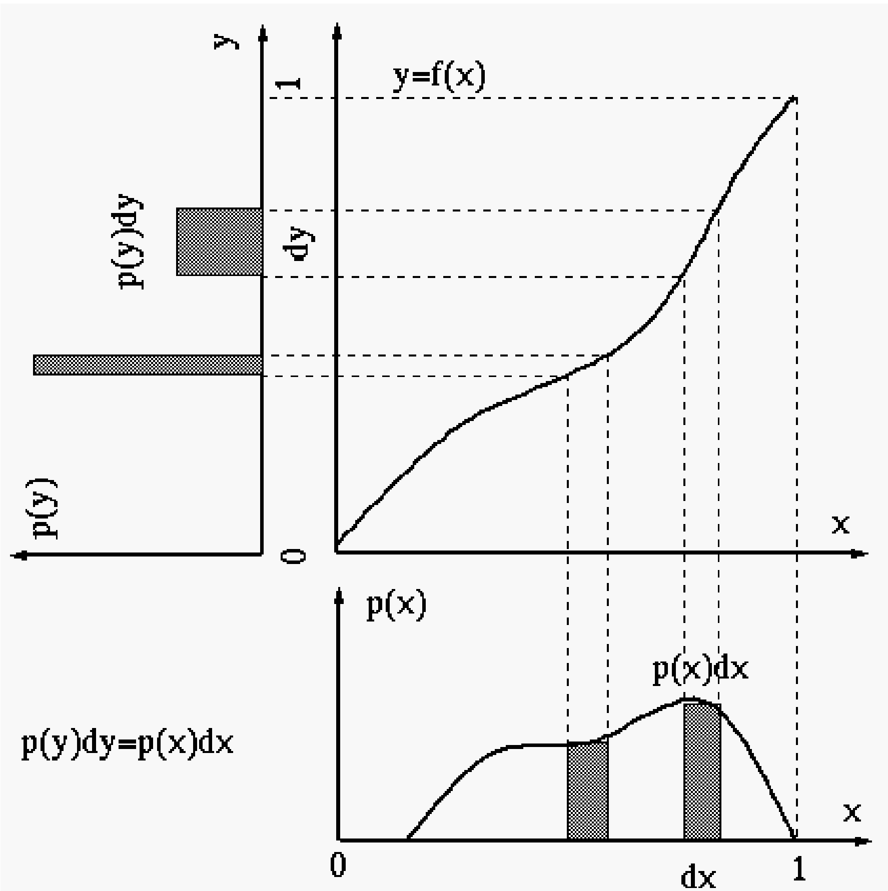
- **Histogram:** Analogous to Probability Density Function (PDF) which represents distribution of samples in a population
- Let $p_r(r)$ and $p_s(s)$ denote the PDFs of before/after random variables r and s, respectively.
- If $p_r(r)$ and $s = T(r)$ are known, then the PDF $p_s(s)$ of the transformed variable s can be obtained as,

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

See next page for explanation

Histogram Equalization (contd.)

Constant Areas



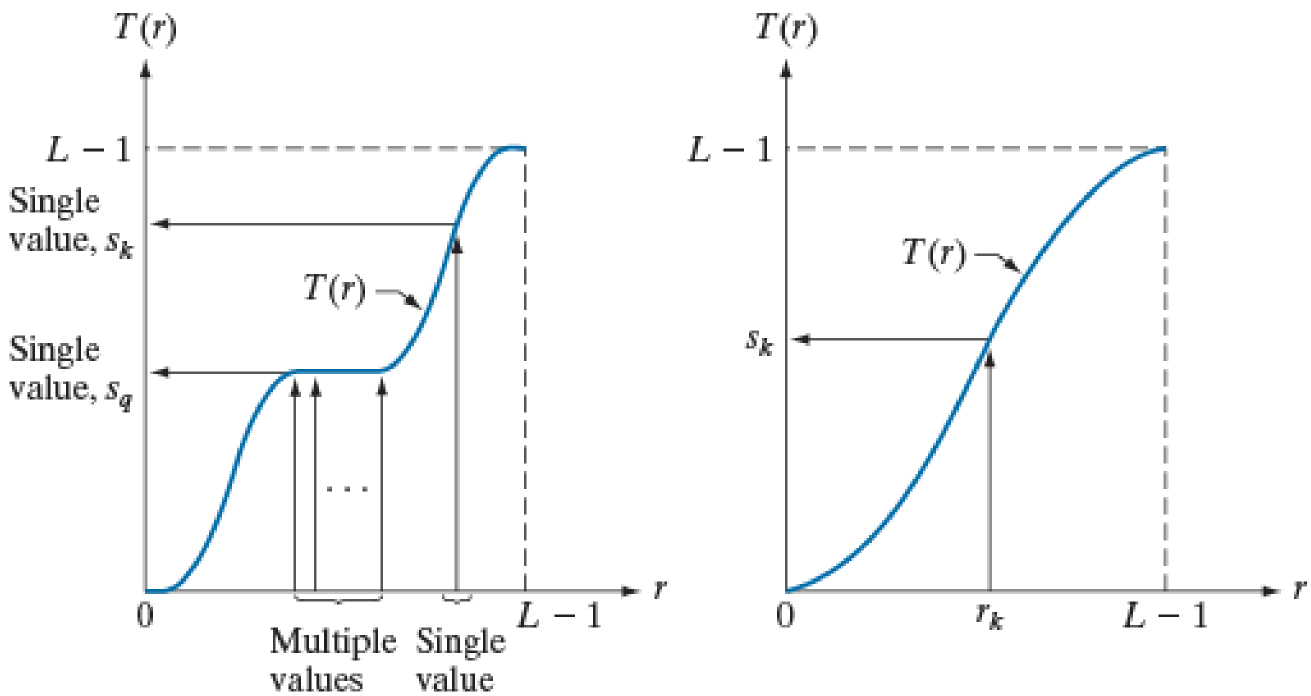
In our case, y→ s & x→ r

- Equalization involves transferring the grey levels so that the histogram of the output image is equalized
- Figure shows that for any given mapping function between the input and output images, the following holds (the areas covered by the histograms are equal):

$$p(y)\, dy = p(x)\, dx$$

Histogram Equalization (contd.)

$$s = T(r)$$



a b

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

Histogram Equalization (contd.)

- Consider a particular transformation function :

Try this T(r): $$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\,F_R(r) \qquad (3.11)$$

Justification

- $F_R(r) = P(R \le r) = \int_0^r p_r(w)dw$: Cumulative Distribution Function (CDF) of r
- When $r = 0$, then integral $= 0$ and $s = 0$
- When $r = L$-1, then integral $= 1$ and $s = L$-1 is the Maximum

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\cdot\left|\frac{1}{\frac{ds}{dr}}\right| = p_r(r)\cdot\frac{1}{(L-1)\left|\frac{d}{dr}\left(\int_0^r p_r(w)dw\right)\right|}$$

$$= p_r(r)\cdot\frac{1}{(L-1)\left|p_r(r)\right|} \quad \text{(Using Leibniz Rule)}$$

$$= \frac{1}{(L-1)} \qquad 0 \le s \le L\text{-}1 \qquad (3.13)$$

$p_r(r) \rightarrow p_s(s)$

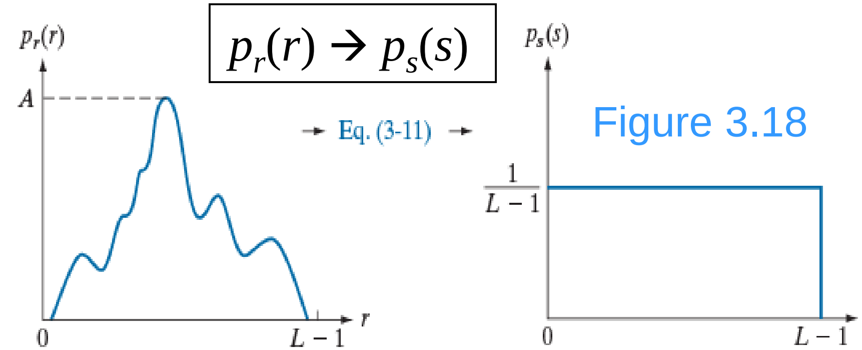


Figure 3.18

- Which *is* the uniform Probability Density Function
- Therefore, $s = T(r)$, as in (3.11), always transforms to uniform PDF !

Histogram Equalization (contd.)

- **Discrete Version (More Practical)**
  - The probability of occurrence of grey level r_k in an image:

$$p(r_k) = \frac{n_k}{n} = \frac{n_k}{MN};\ k = 0,1,\ldots,L\text{-}1 \quad \text{(Probability Mass Function)}$$

- n : Total number of pixels in the image, $n = MN$
- n_k : Number of pixels that have grey level r_k
- L : Total number of possible grey levels in the image

- **The transformation function (Discrete Version of 3.11):**

Accumulative Sum of $p_r(r)$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \quad k = 0, 1, 2, \ldots L-1 \qquad (3.15)$$

- Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k.

Histogram Equalization Example

Intensity	# pixels
0	20
1	5
2	25
3	10
4	15
5	5
6	10
7 (=L-1)	10
Total	100

Accumulative Sum of P_r
20/100 = 0.2
(20+5)/100 = 0.25
(20+5+25)/100 = 0.5
(20+5+25+10)/100 = 0.6
(20+5+25+10+15)/100 = 0.75
(20+5+25+10+15+5)/100 = 0.8
(20+5+25+10+15+5+10)/100 = 0.9
(20+5+25+10+15+5+10+10)/100 = 1.0
1.0

WRIGHT STATE UNIVERSITY

Histogram Equalization Example

Intensity (r)	No. of Pixels (n_j)	Accumul Sum of P_r	Output value (L=8)	Quantized Output (s) Nearest integer
0	20	0.2	0.2*7 = 1.4	1
1	5	0.25	0.25*7 = 1.75	2
2	25	0.5	0.5*7 = 3.5	4
3	10	0.6	0.6*7 = 4.2	4
4	15	0.75	0.75*7 = 5.25	5
5	5	0.8	0.8*7 = 5.6	6
6	10	0.9	0.9*7 = 6.3	6
7	10	1.0	1.0*7 = 7	7
Total	100			

Histogram Equalization - Example

- Let f be an image with size 64x64 pixels and L=8 and let f has the intensity distribution as shown in the table. After Transformation:

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

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 7(0.19) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7(p_r(r_0) + p_r(r_1)) = 7(0.19 + 0.25) = 3.08$$

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

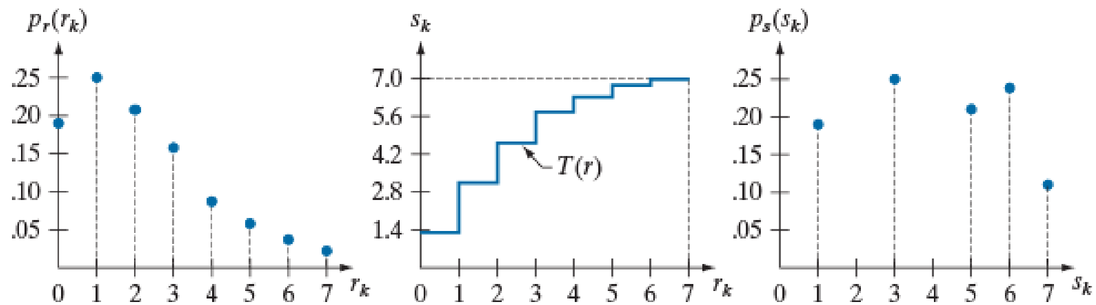
Round the values to the nearest integer

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

Equalized – More Uniform

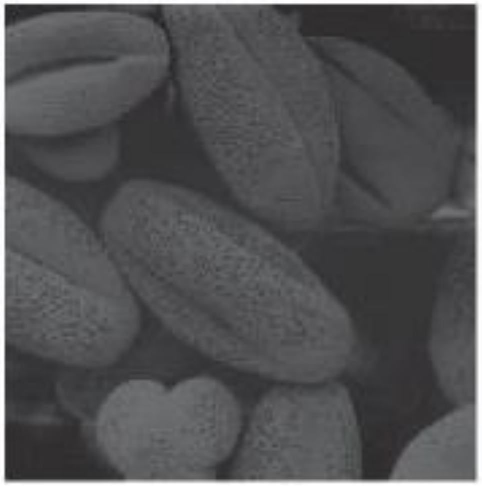
a b c

FIGURE 3.19 Histogram equalization. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.



Histogram Equalization – Dark Image

(1) Before Equalization



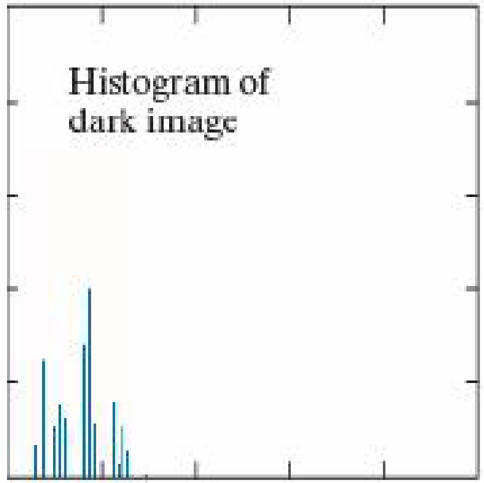
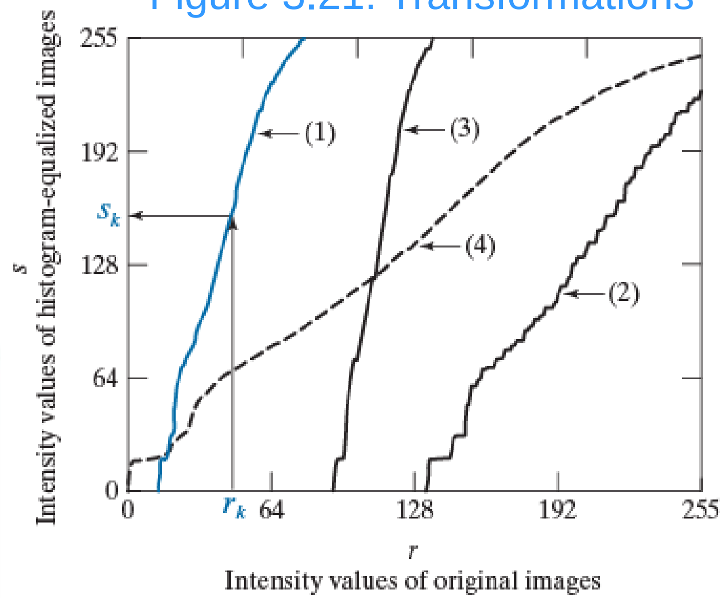
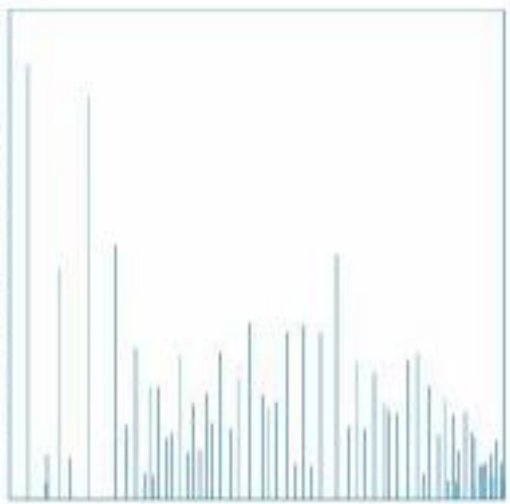


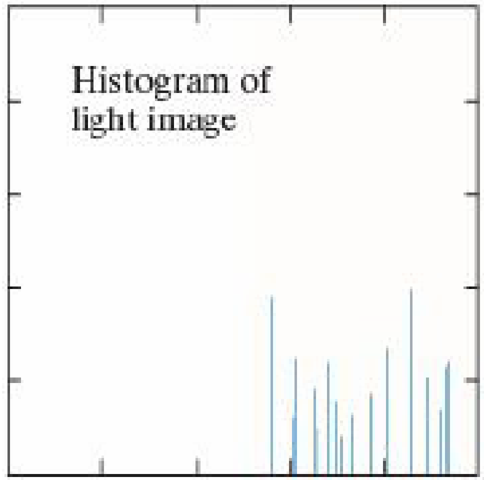
Figure 3.21: Transformations





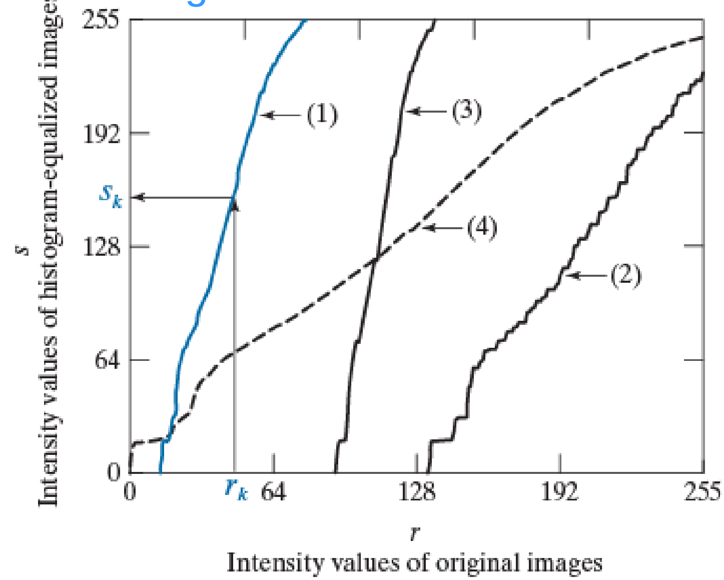
After Equalization (1)
Using (3.15)

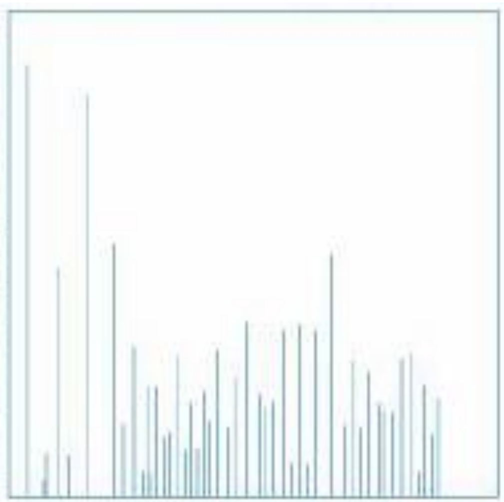
Histogram Equalization – Light Image



(2) Before Equalization

Figure 3.21: Transformations





After Equalization (2)
Using (3.15)

WRIGHT STATE UNIVERSITY

Histogram Equalization – Low Contrast Image

(3) Before Equalization

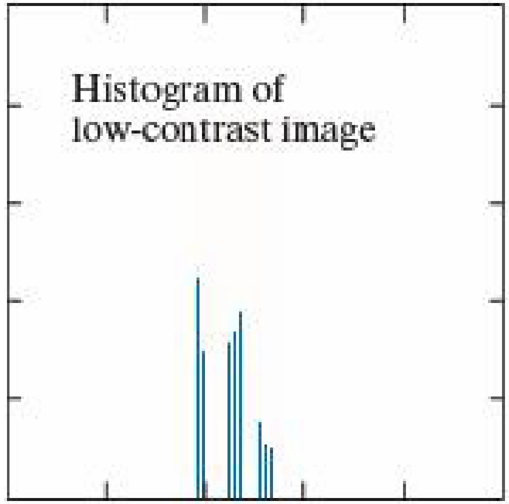
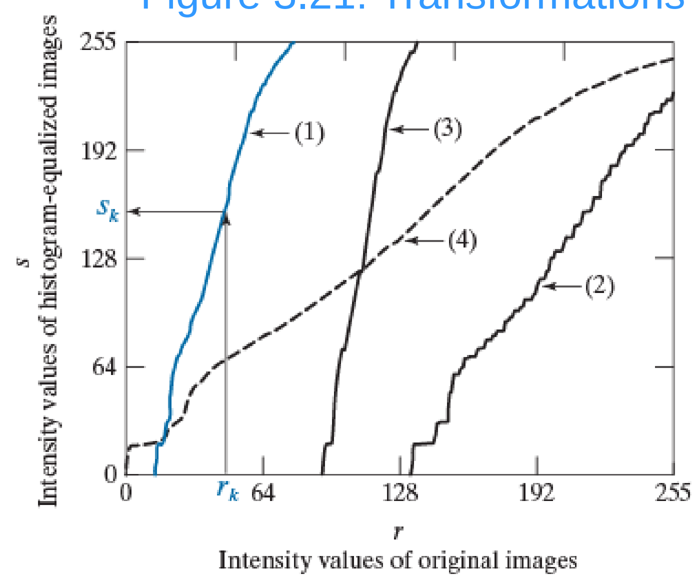
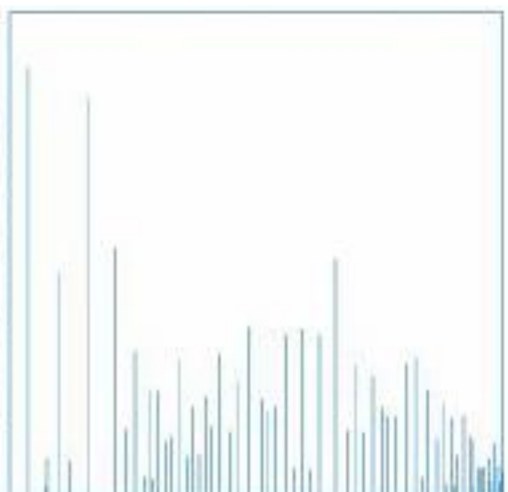


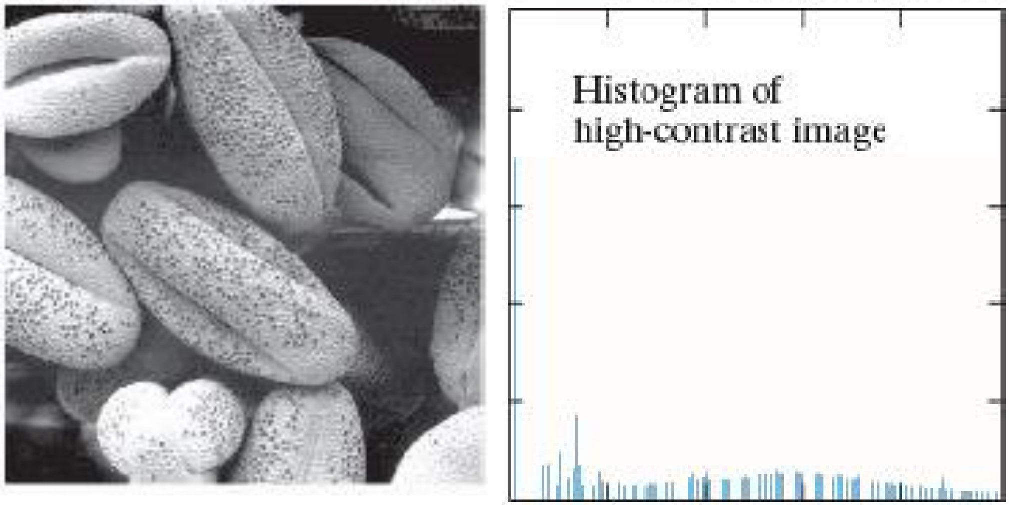
Figure 3.21: Transformations





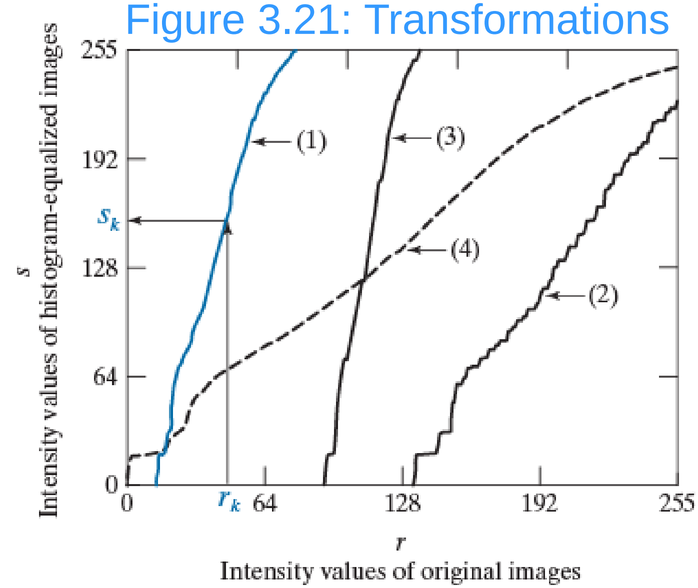
After Equalization (3) Using (3.15)

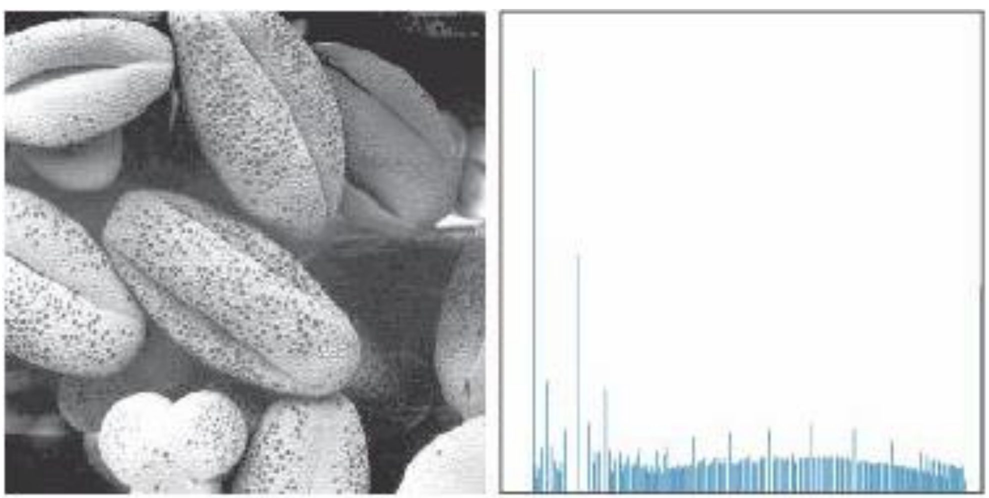
Histogram Equalization – High Contrast Image



(4) Before Equalization

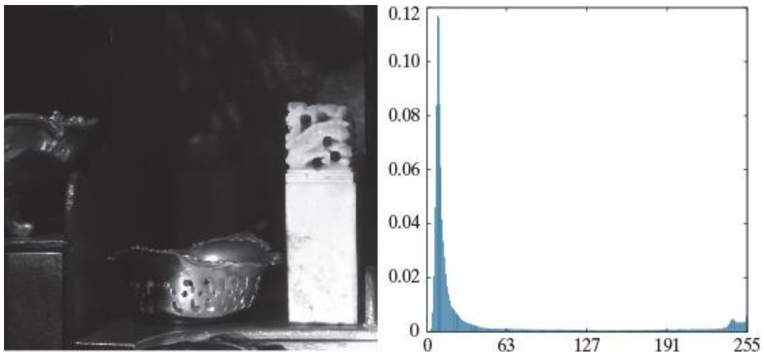
Figure 3.21: Transformations





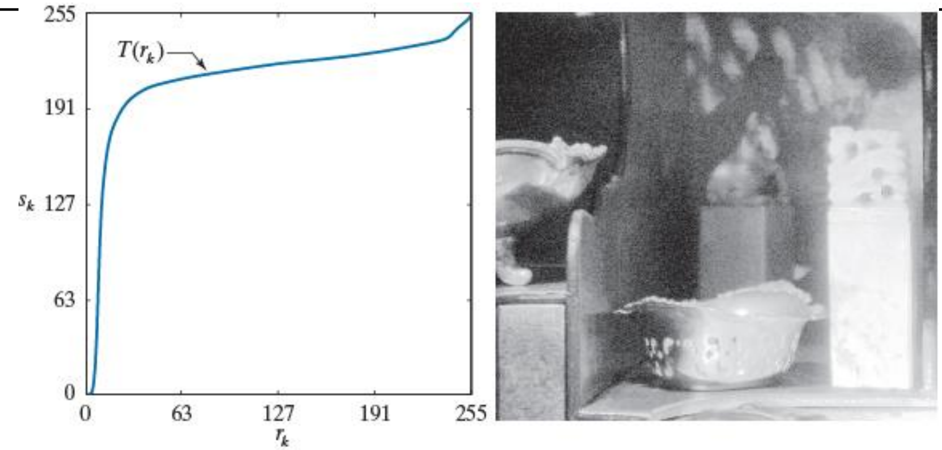
- After Equalization (4) Using (3.15)
- Not much difference!

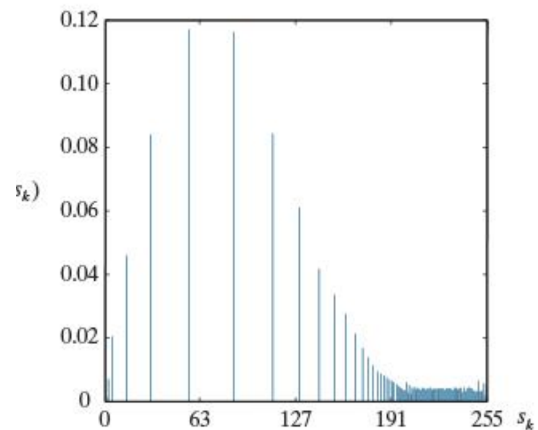
Problem with Histogram Equalization



a b

FIGURE 3.24
(a) An image, and
(b) its histogram.



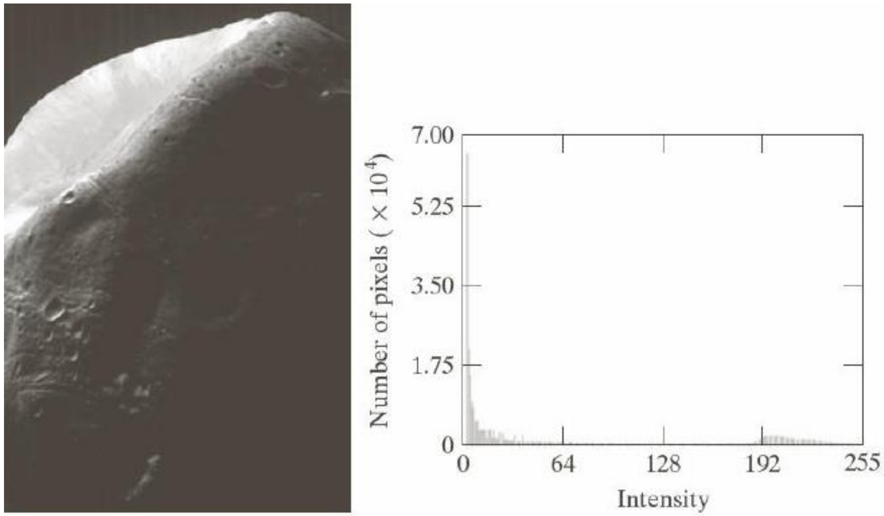


a b
c

FIGURE 3.25
(a) Histogram equalization transformation obtained using the histogram in Fig. 3.24(b).
(b) Histogram equalized image.
(c) Histogram of equalized image.

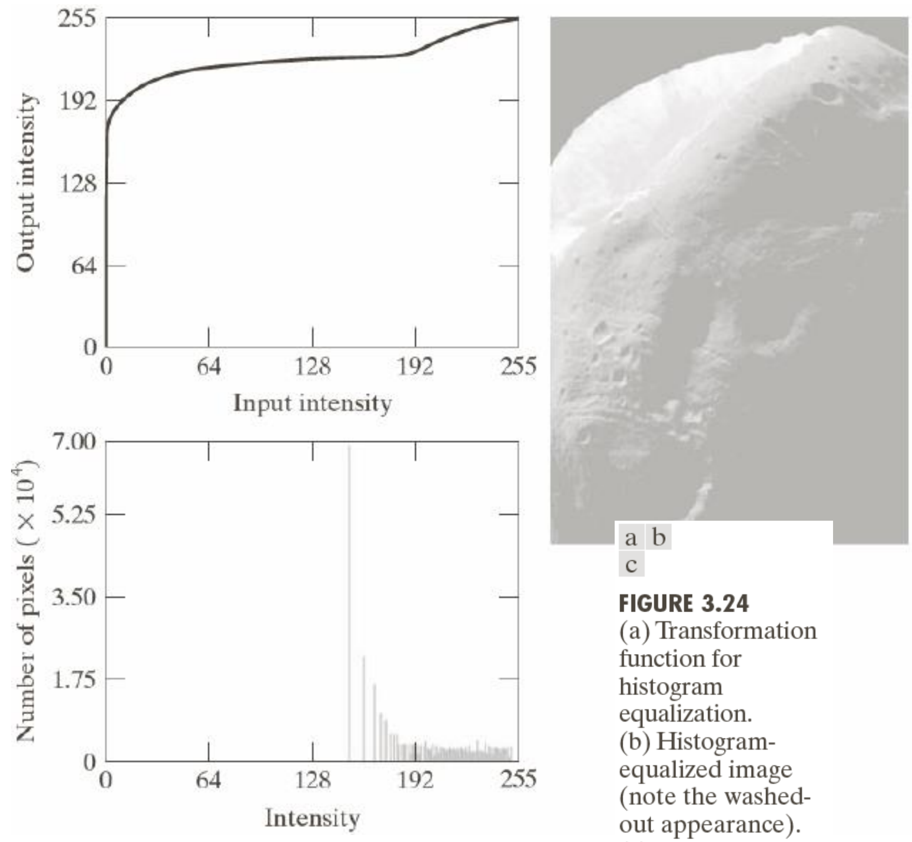
- In this case, net effect of applying histogram equalization:
  - Transform a very narrow interval of dark pixels into the upper end of the grey scale → Washed-out image!

Problem with Histogram Equalization



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.)



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).

- In this case, net effect of applying histogram equalization:
  - ➢ Transform a very narrow interval of dark pixels into the upper end of the grey scale → Washed-out image!

3rd Edition

Solution → Histogram Matching

- **Objective:** Transform image histogram to a desired shape
- **Concept :** From Histogram equalization: ($p_r(r)$ known)

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw\,; \;\rightarrow\; \text{Leads to } p_s(s) = 1/(L\text{-}1)$$

- Want an output image to have desired PDF: $p_z(z)$ (known)
- Apply histogram equalization to $p_z(z)$ to get intermediate $v{=}s$

$$v = G(z) = (L-1)\int_0^z p_z(u)\,du = s\,; \;\rightarrow\; \text{Gives } p_v(v) = 1/(L\text{-}1) = p_s(s)$$

- Since $p_s(s) = p_v(v) = 1/(L\text{-}1)$, s and v are equivalent
- Therefore, transform r to z by

$$\boxed{v = G(z); \quad z = G^{-1}(v)}$$

$$r \rightarrow \boxed{T(\,)} \xrightarrow[\;s\,=\,v\;]{\text{uniform}} \boxed{G^{-1}(\,)} \rightarrow z$$

$$\boxed{p_S(s) = p_V(v) = \frac{1}{L-1}}$$

3.3.2 Histogram Matching

$p_r(r)$ Histogram of Input Image

Intensity (r)	# pixels
0	20
1	5
2	25
3	10
4	15
5	5
6	10
7	10
Total	**100**

$p_z(z)$: Specified "Desired" Histogram of Output Image (NOT Uniform)

Intensity (z)	# pixels
0	5
1	10
2	15
3	20
4	20
5	15
6	10
7	5
Total	**100**

Histogram Matching Example

Step-1. Histogram Equalization

After Multiplying by 7

r	(n_j)	Σp_r	s uniform
0	20	0.2	$1.4 \rightarrow 1$
1	5	0.25	$1.75 \rightarrow 2$
2	25	0.5	$3.5 \rightarrow 4$
3	10	0.6	$4.2 \rightarrow 4$
4	15	0.75	$5.25 \rightarrow 5$
5	5	0.8	$5.6 \rightarrow 6$
6	10	0.9	$6.3 \rightarrow 6$
7	10	1.0	$7 \rightarrow 7$

z	(n_j)	Σp_z	v uniform
0	5	0.05	$0.35 \rightarrow 0$
1	10	0.15	$1.05 \rightarrow 1$
2	15	0.3	$2.1 \rightarrow 2$
3	20	0.5	$3.5 \rightarrow 4$
4	20	0.7	$4.9 \rightarrow 5$
5	15	0.85	$5.95 \rightarrow 6$
6	10	0.95	$6.65 \rightarrow 7$
7	5	1.0	$7 \rightarrow 7$

Input $s_k = T(r_k)$

Desired $v_k = G(z_k)$

Histogram Matching Example

Step-2. Mapping done using v closest to s

r → s

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

$s_k = T(r_k)$

s → v

uniform

v → z

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

$z_k = G^{-1}(v_k)$

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

Actual Output Histogram

z	# Pixels
0	0
1	20
2	5
3	35=25+10
4	15
5	15=5+10
6	10
7	0

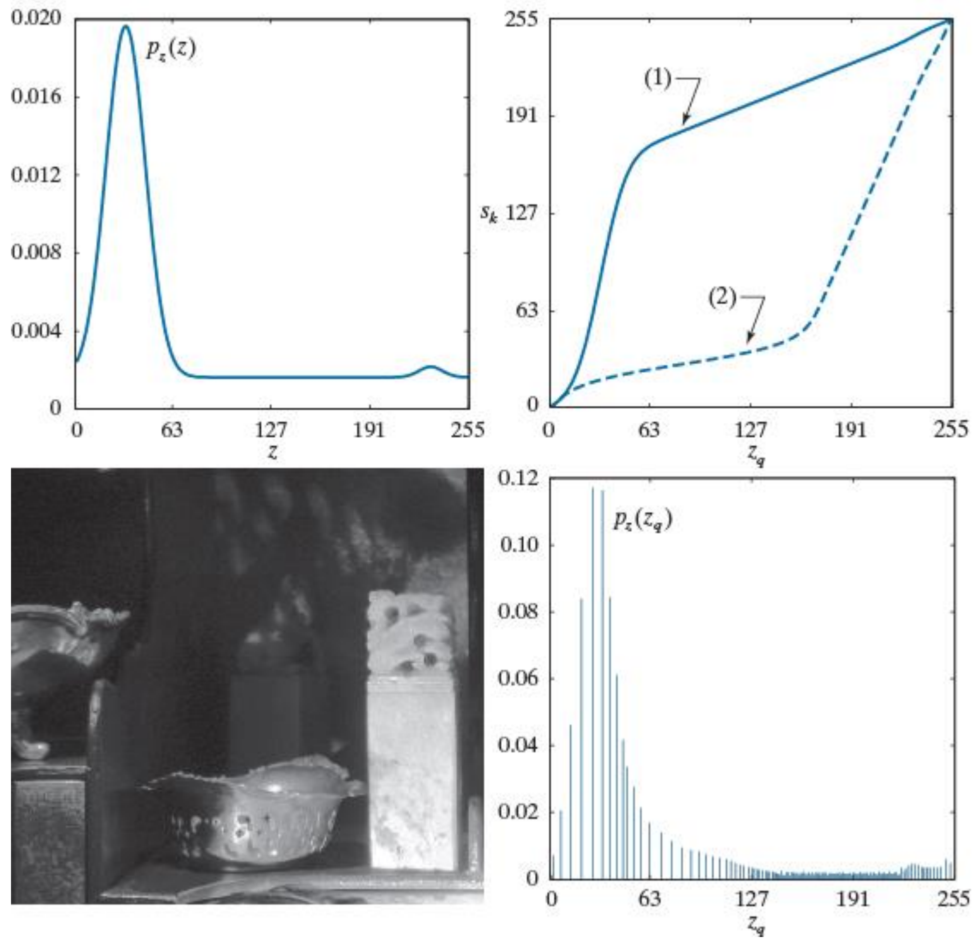
Histogram Matching Example

(1) $G(z)$ from (3.21) (2) $G^{-1}(s)$ from (3.23)

Desired Histogram

Transfer Function

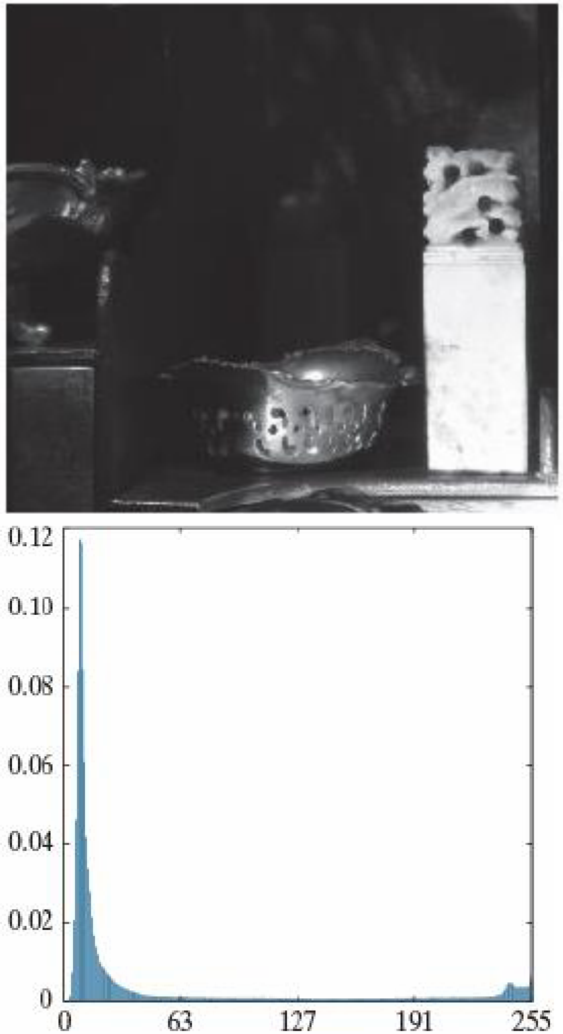
Actual Histogram



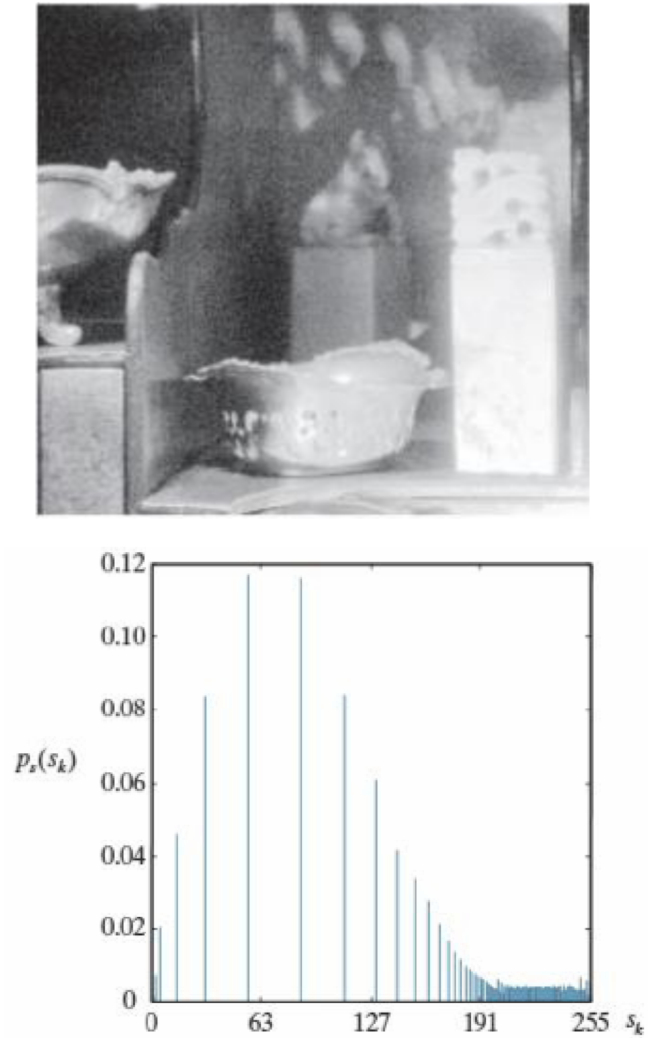
a b
c d

FIGURE 3.26 Histogram specification. (a) Specified histogram. (b) Transformation $G(z_q)$, labeled (1), and $G^{-1}(s_k)$, labeled (2). (c) Result of histogram specification. (d) Histogram of image (c).

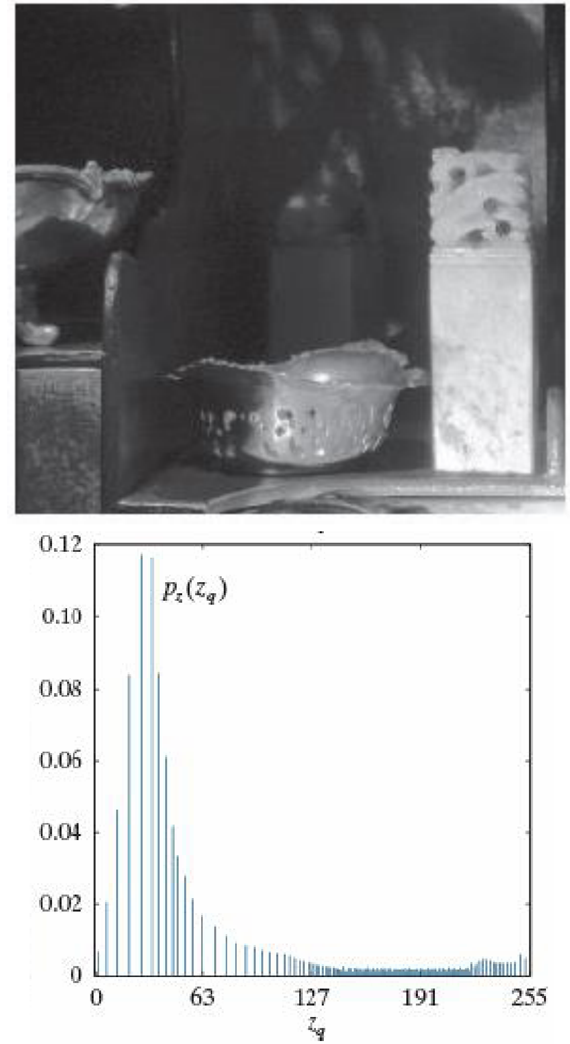
Histogram Matching Example



Original Image



Histogram Equalization



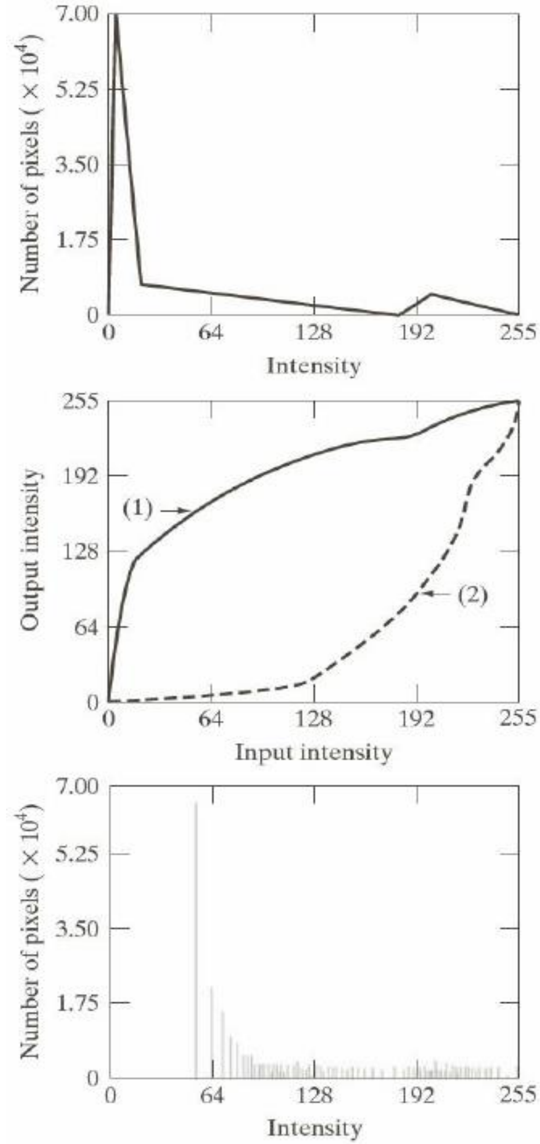
Histogram Matching

Histogram Matching Example

Desired Histogram

Transfer Function

Actual Histogram



(1) $G(z)$ from (3.21)

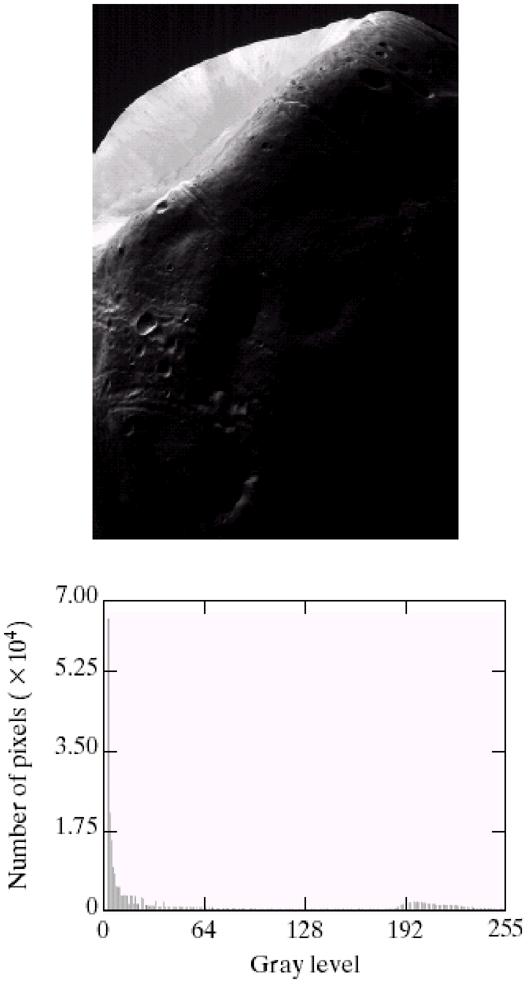
(2) $G^{-1}(s)$ from (3.23)

a	c
b	
d	

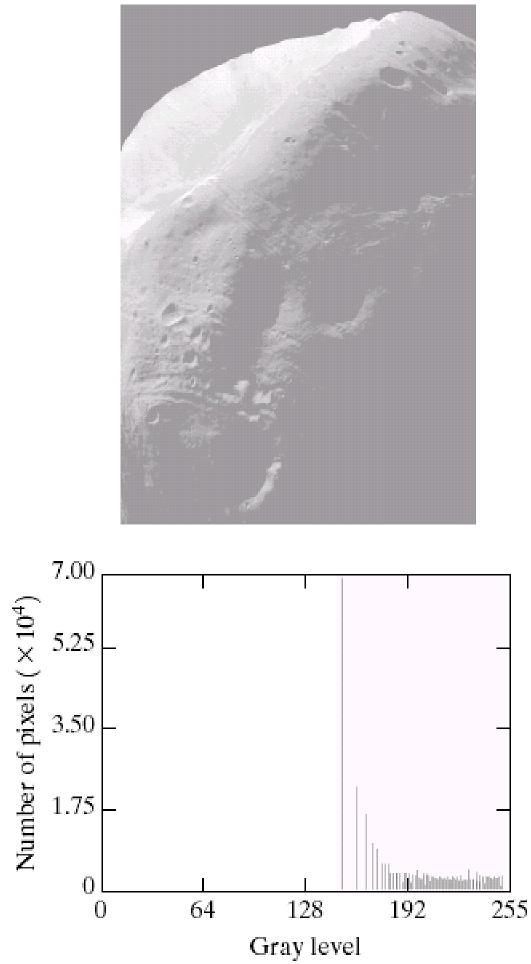
FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

3rd Edition

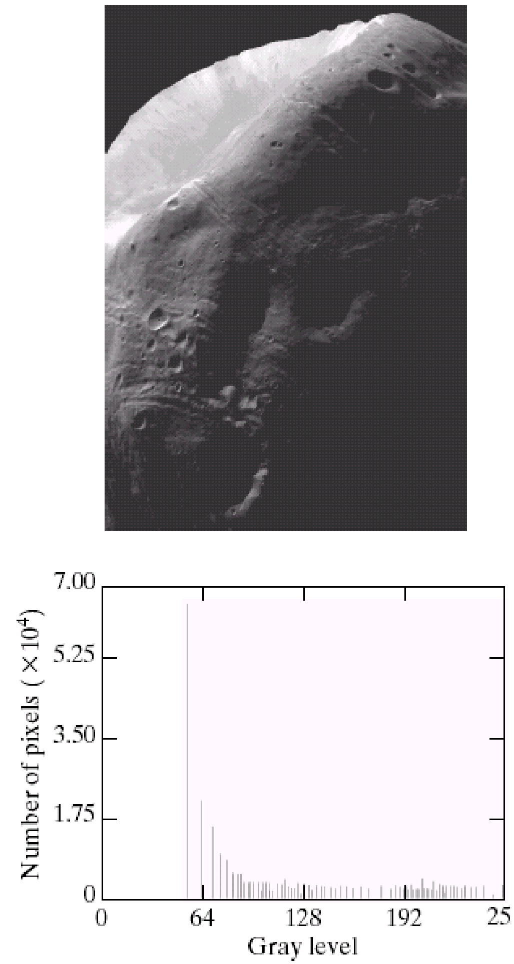
Histogram Matching Example



Original Image



Histogram Equalization



Histogram Matching

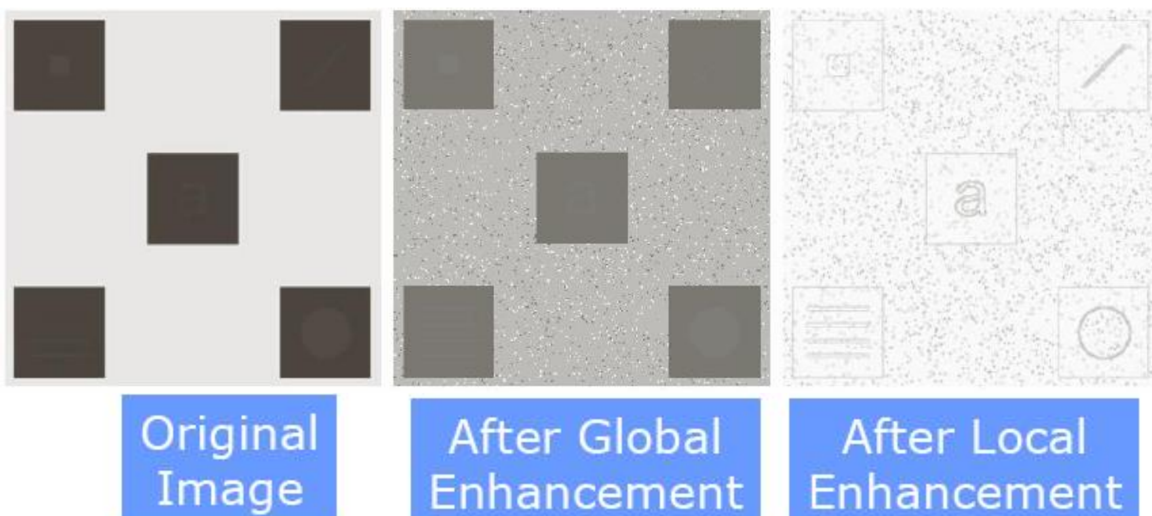
3.3.3 Enhancement with Local Histogram Processing

Motivation

- Enhance details over small areas in an image
- Computation of global transformation does not guarantee desired local enhancement

Solution

- Define a small square or rectangular window or neighborhood
- Move the center of this window from pixel to pixel
- Apply Histogram equalization in each window region



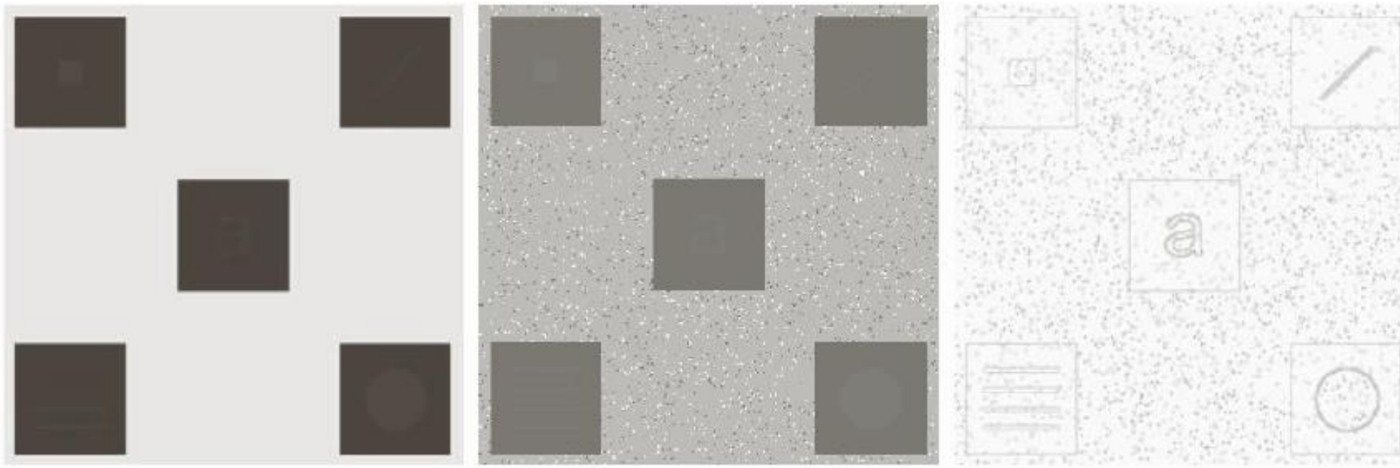
a b c

FIGURE 3.32 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization.

3.3.3 Local Histogram Processing

a b c

FIGURE 3.32 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization.



- Image in (a) is slightly noisy but the noise is imperceptible.
- HE enhances the noise in smooth regions (b).
- Local HE reveals structures having values close to the values of the squares and small sizes to influence HE (c).

Histogram Statistics for Image Enhancement

r_i : Intensity value in the range $[0, L\text{-}1]$

$p(i)$: Histogram component corresponding to value r_i

m is the mean value of r (its average gray level):

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

The intensity variance:

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

Histogram Statistics for Image Enhancement

- Let (x, y) be the coordinates of a pixel in an image, and let S_{xy} denote a neighborhood (subimage) of specified size, centered at (x, y).

- The ***mean value*** of the pixels in this neighborhood is given by

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

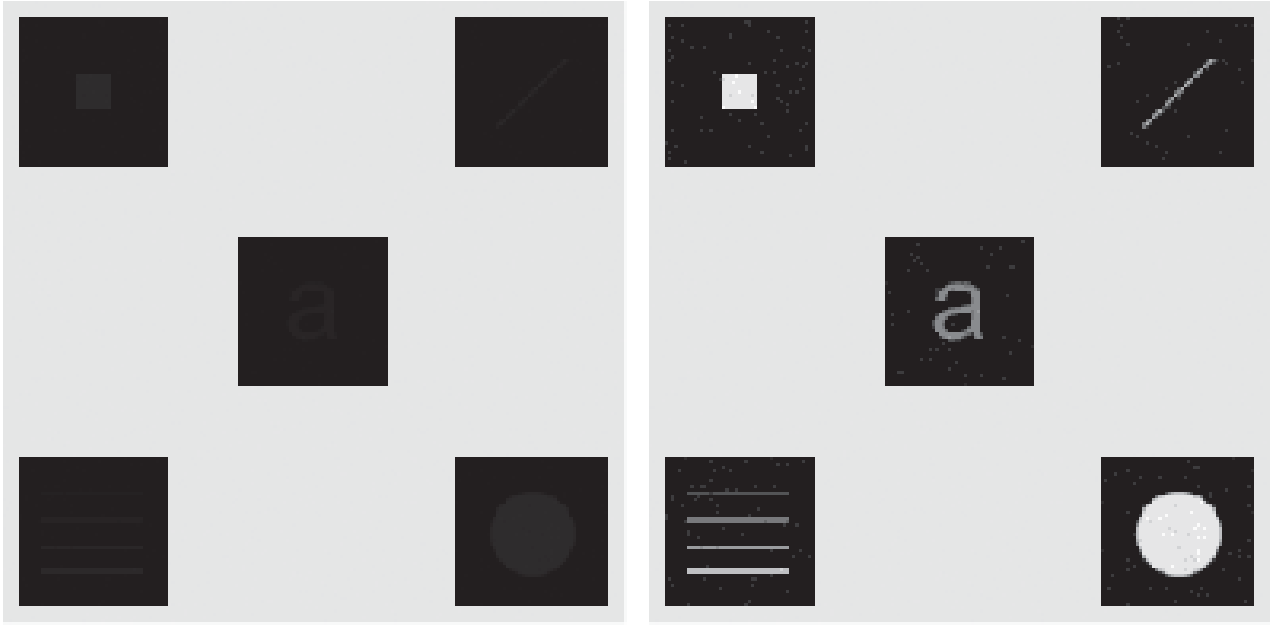
where, $p_{S_{xy}}$ is the histogram of the pixels in region S_{xy}.

- The ***variance*** of the pixels *in the neighborhood* is given by

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

Using Histogram Statistics for Image Enhancement

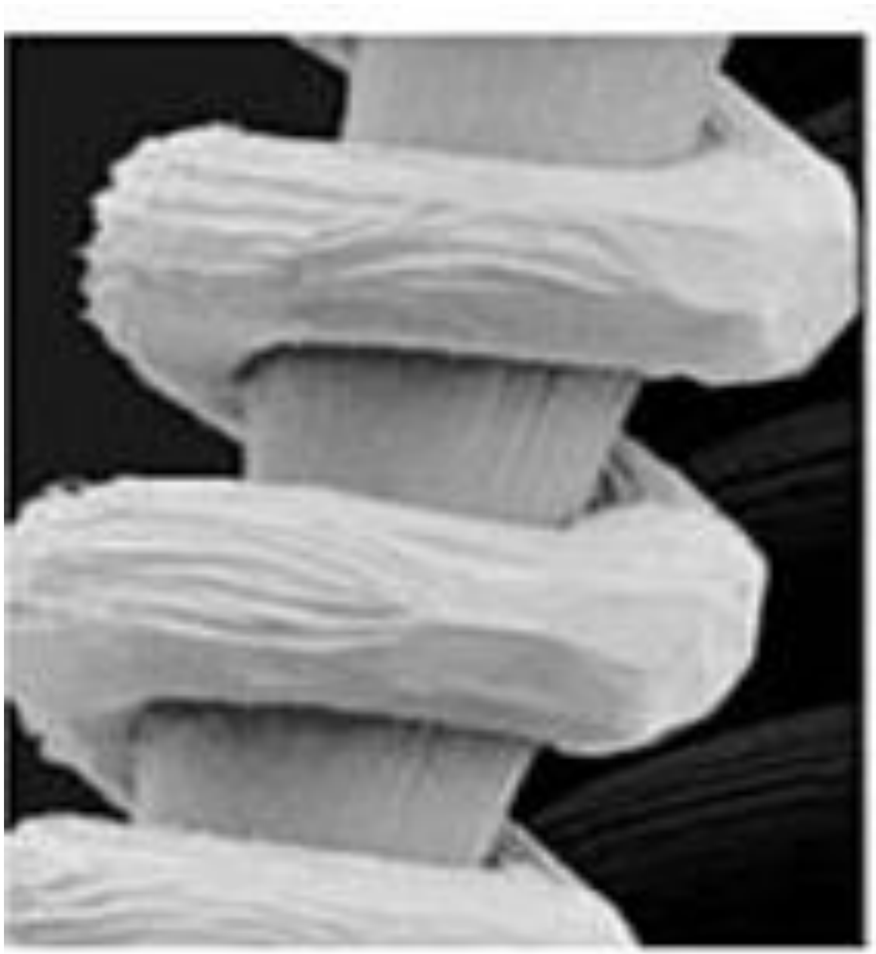
See Example 3.12 (details omitted)



a b

FIGURE 3.33 (a) Original image. (b) Result of local enhancement based on local histogram statistics. Compare (b) with Fig. 3.32(c).

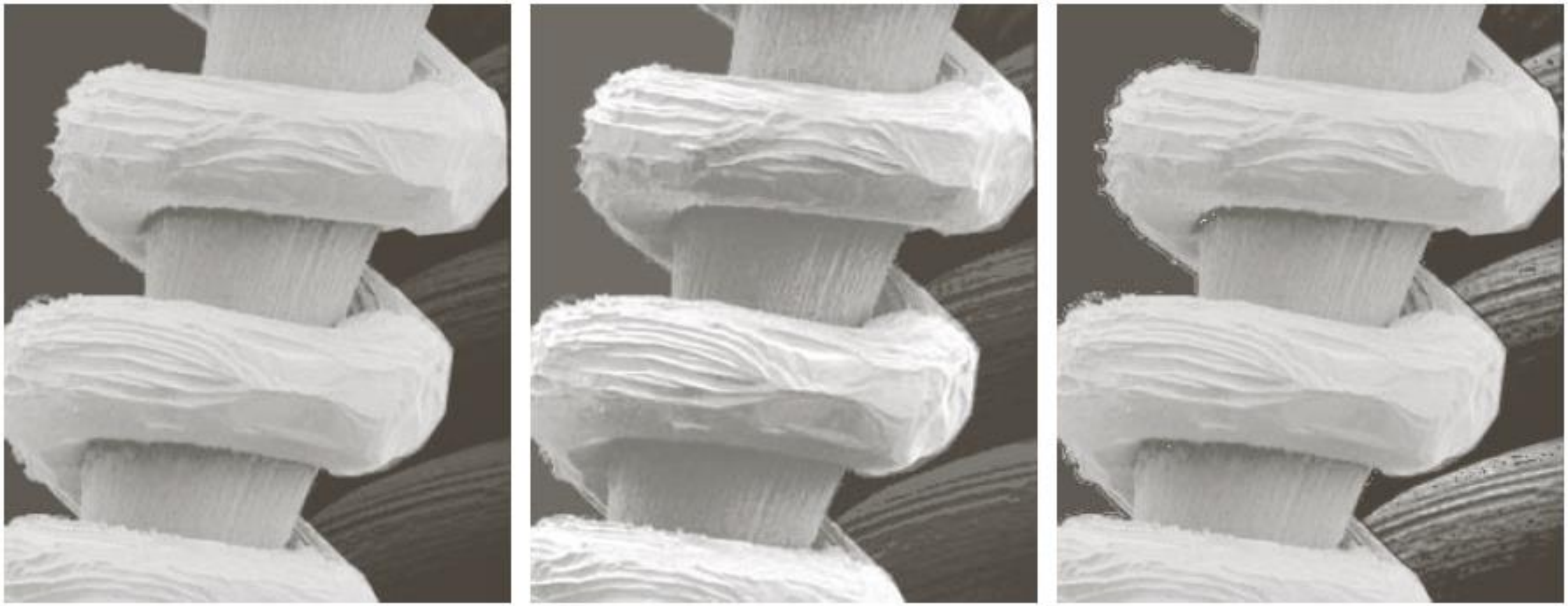
Histogram for Image Enhancement



Tungsten filament

3rd Edition

Using Histogram Statistics for Image Enhancement



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.)

3rd Edition

Final Thoughts on Histogram Processing

- To a large extent, histogram specification is a trial-and-error process.
- There are no particular rules for specifying histograms
- One must resort to analysis on a case-by-case basis for any given enhancement tasks

Chapter-3

Part-3: Spatial Filtering

3.4 Basics of Spatial Filtering

MASK: Also known as:

- Filter
- Kernel
- Template
- Window

- The first three are more prevalent in DIP

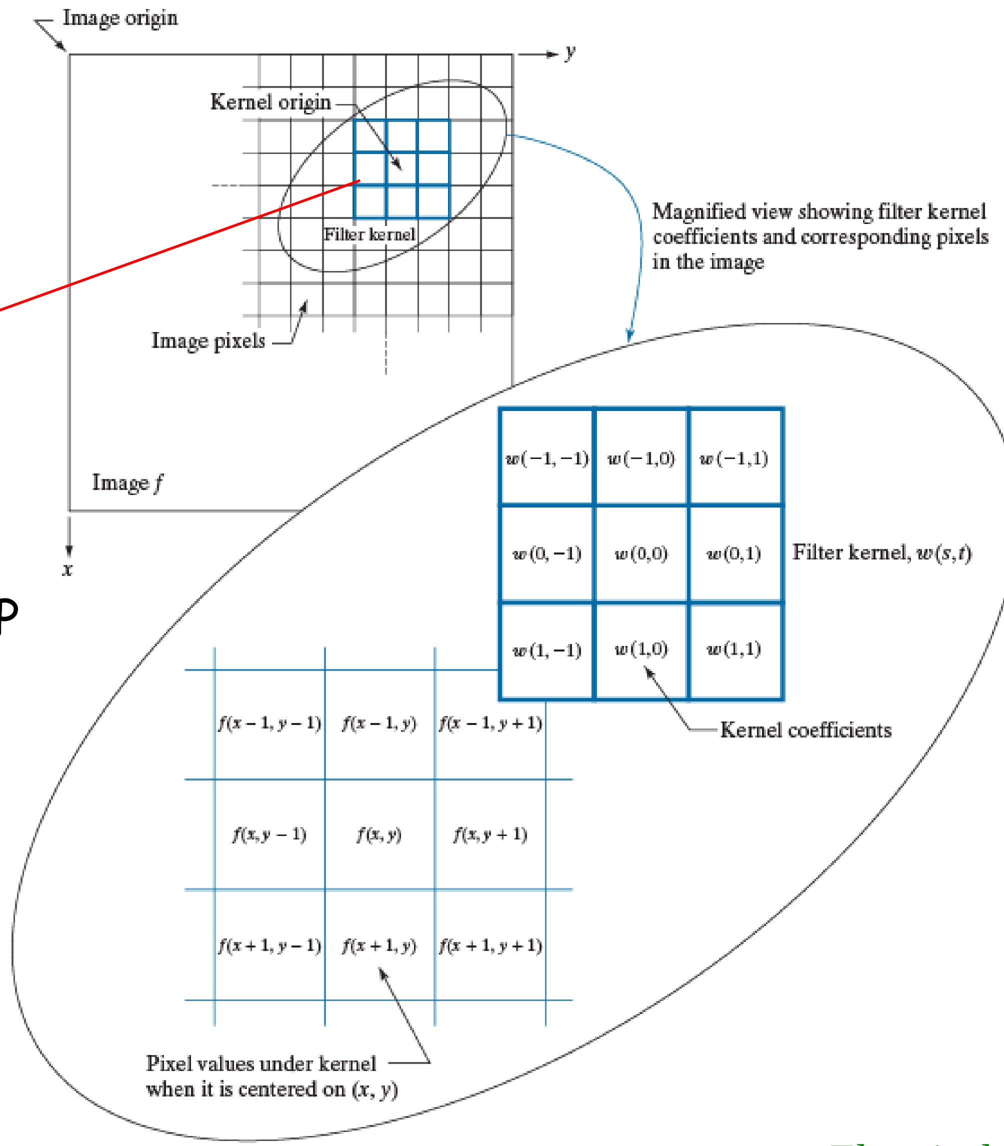


FIGURE 3.34 The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

Basics of Spatial Filtering

- Linear filtering of $M \times N$ image with a $m \times n$ filter mask:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t); \quad x = 0,1,\ldots,M-1;\, y = 0,1,\ldots,N-1$$

  - $a = (m\text{-}1)/2$
  - $b = (n\text{-}1)/2$ (m and n are assumed to be odd)

- The process is similar to convolution.
- Hence, referred to as convolving a mask with an image.
- Output of mask at any pair of coordinates is called the "response of filter mask"
- For a 3×3 general mask:

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

3.4.2 2D Convolution

- The filtered image is the convolution of the original image with the filter impulse response (or "mask").
- So, if $f(x, y)$ denotes the original image and $w(x, y)$ the filter impulse response, then their convolution is

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t); \quad x = 0,1,\ldots,M-1; y = 0,1,\ldots,N-1$$

- Note the negative signs in $(x\text{-}s, y\text{-}t)$.
- *Steps:*
  - Either the impulse response or the original image is mirrored (flipped) vertically and horizontally
  - Pixel-wise product and sum

3.4.2 2D Correlation

- Correlation is quite similar to convolution
- Mathematically

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t); \quad x=0,1,\ldots,M-1; y=0,1,\ldots,N-1$$

- No negative sign → No flipping or mirroring involved!
- Differences to usage in statistics
  - This operation is called cross-correlation in statistics
  - In image correlation, normalization can be neglected
- Important for Matched Filtering

WRIGHT STATE UNIVERSITY

1D Convolution & Correlation

Correlation

	Origin f w	
(a)	0 0 0 **1** 0 0 0 0 **1 2 4 2 8**	
(b)	0 0 0 **1** 0 0 0 0	
	1 2 4 2 8	Starting position alignment
	Zero padding	
(c)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	1 2 4 2 8	Starting position
(d)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	1 2 4 2 8	Position after 1 shift
(e)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	1 2 4 2 8	Position after 3 shifts
(f)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	1 2 4 2 8	Final position
	Correlation result	
(g)	0 8 2 4 2 1 0 0	
	Extended (full) correlation result	
(h)	0 0 0 8 2 4 2 1 0 0 0 0	

Convolution

	Origin f w rotated 180°	
(i)	0 0 0 **1** 0 0 0 0 **8 2 4 2 1**	
(j)	0 0 0 **1** 0 0 0 0	
	8 2 4 2 1	Starting position alignment
	Zero padding	
(k)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	8 2 4 2 1	Starting position
(l)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	8 2 4 2 1	Position after 1 shift
(m)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	8 2 4 2 1	Position after 3 shifts
(n)	0 0 0 0 0 **1** 0 0 0 0 0 0	
	8 2 4 2 1	Final position
	Convolution result	
(o)	0 1 2 4 2 8 0 0	
	Extended (full) convolution result	
(p)	0 0 0 1 2 4 2 8 0 0 0 0	

FIGURE 3.35 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.

2D Convolution & Correlation

Origin f

$$\begin{array}{ccccc} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{array} \qquad w \quad \begin{array}{ccc} 1&2&3\\ 4&5&6\\ 7&8&9 \end{array}$$

(a)

Padded f

$$\begin{array}{ccccccc} 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{array}$$

(b)

Initial position for w

$$\begin{array}{ccccccc} 1&2&3&0&0&0&0\\ 4&5&6&0&0&0&0\\ 7&8&9&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{array}$$

(c)

Correlation result

$$\begin{array}{ccccc} 0&0&0&0&0\\ 0&9&8&7&0\\ 0&6&5&4&0\\ 0&3&2&1&0\\ 0&0&0&0&0 \end{array}$$

(d)

Full correlation result

$$\begin{array}{ccccccc} 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&9&8&7&0&0\\ 0&0&6&5&4&0&0\\ 0&0&3&2&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{array}$$

(e)

Rotated w

$$\begin{array}{ccccccc} 9&8&7&0&0&0&0\\ 6&5&4&0&0&0&0\\ 3&2&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{array}$$

(f)

Convolution result

$$\begin{array}{ccccc} 0&0&0&0&0\\ 0&1&2&3&0\\ 0&4&5&6&0\\ 0&7&8&9&0\\ 0&0&0&0&0 \end{array}$$

(g)

Full convolution result

$$\begin{array}{ccccccc} 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&1&2&3&0&0\\ 0&0&4&5&6&0&0\\ 0&0&7&8&9&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{array}$$

(h)

FIGURE 3.36 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. (3-45) and (3-46) regarding full correlation and convolution.

Separable Filter Kernels

- A 2-D filter is Separable if $G(x,y) = G_1(x)G_2(y)$
- Separable filters can be implemented by successive application of two simpler 1-D filters.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

- Separability results in significant time and computation saving.
- For a mask of size $n \times n$, for each image pixel
  - Originally, n^2 multiplications and n^2-1 additions
  - After separation, $2n$ multiplications and $2n$-2 additions

Separable Discrete Filters

- For a Separable 2-D filter: $w = w_1 \star w_2$
- An image f can be processed using associative and commutative properties of convolution as

$$w \star f = \; w_1 \star w_2 \; \star f = \; w_2 \star w_1 \; \star f$$

$$= w_1 \star \; w_2 \star f \; = w_2 \star \; w_1 \star f$$

- Gaussian filter:
  - ➢ Separable and Isotropic (Circular Symmetric)

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

- Many 2-D filters, including wavelets, are separable

3.5 Smoothing (Lowpass) Spatial filters

- **Smoothing filters:** Used for blurring and noise reduction
- **Blurring**: Used in preprocessing steps,
  - Remove small details prior to large object extraction
  - Bridge small gaps in lines or curves
  - Smoothing used to reduce false contouring
- Replace every pixel by average intensity in neighborhood
  → Results in reduced sharp transitions in intensities
- Effect: Average of pixels in the neighborhood
- Sometimes called averaging filter or lowpass filter
- Disadvantage: While smoothing can reduce "sharp" transitions due to noise, it has the undesirable side effect of thickening at edges.

3x3 Smoothing Filter Examples

- Two types of smoothing filters are generally used
  - Box filter → Equal-valued weights
  - Weighted Average Filter
    - Give some pixels more importance (weight) at the expense of others
    - Objective: Attempt to reduce blurring in the smoothing process

Box Kernel

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

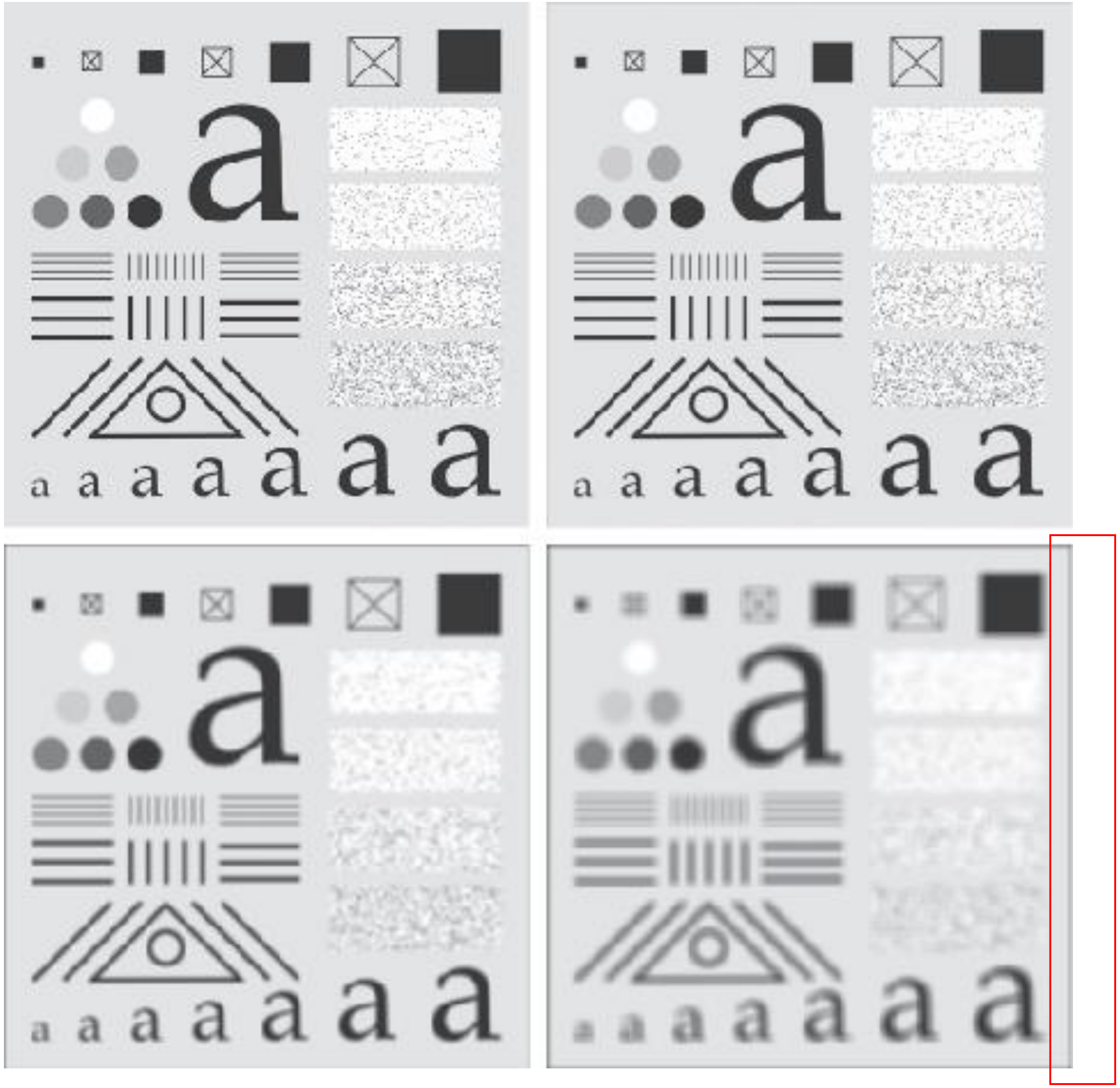
Weighted Average Filter

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

Gaussian Kernel

$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

Effect of Smoothing Filter



a	b
c	d

FIGURE 3.39
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.

Effect of Smoothing Filter

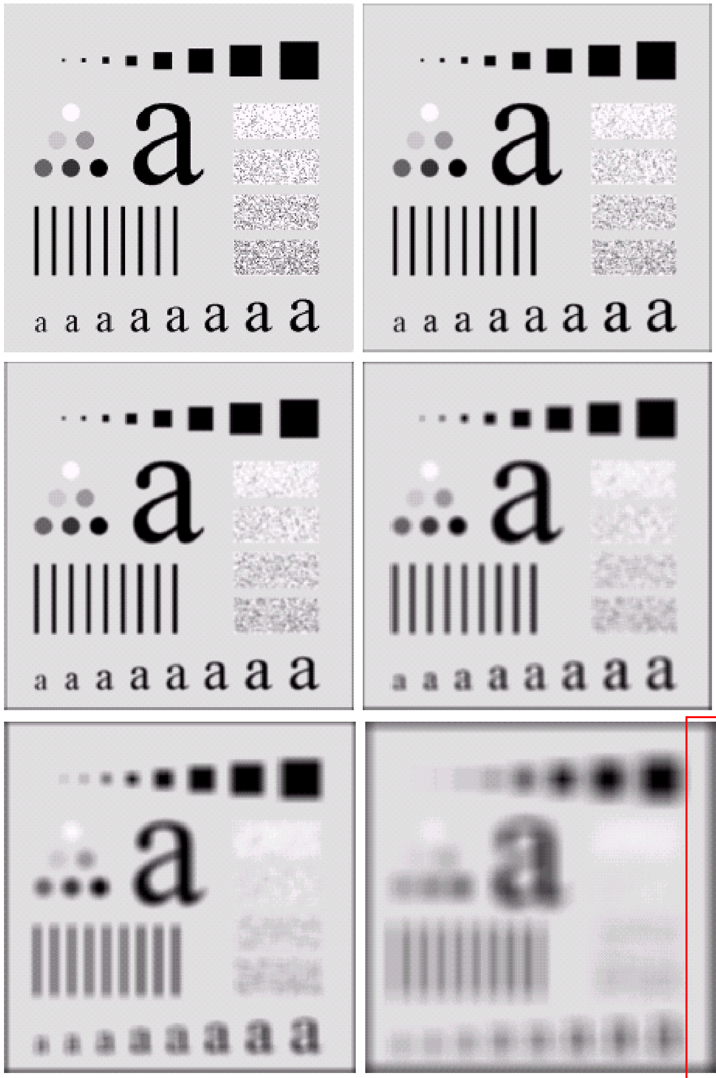


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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.

3rd Edition

Lowpass Gaussian Kernel

- **Box Filters**: Simple and visually acceptable result
  - Do not work well for images with high level of detail or with strong geometrical components
  - Directionality of box filters often produce undesirable results

- **Gaussian filter:**
  - Separable
  - Isotropic (Circular Symmetric)
  - Only Circular Symmetric kernel that is also separable

Lowpass Gaussian Kernel

- Gaussian filter:
  - Separable and Isotropic (Circular Symmetric)

$$w(s,t) = G(s,t)$$

$$= K\, e^{-\frac{s^2+t^2}{2\sigma^2}}$$

$$G(r) = K\, e^{-\frac{r^2}{2\sigma^2}}$$

where, $r^2 = s^2 + t^2$

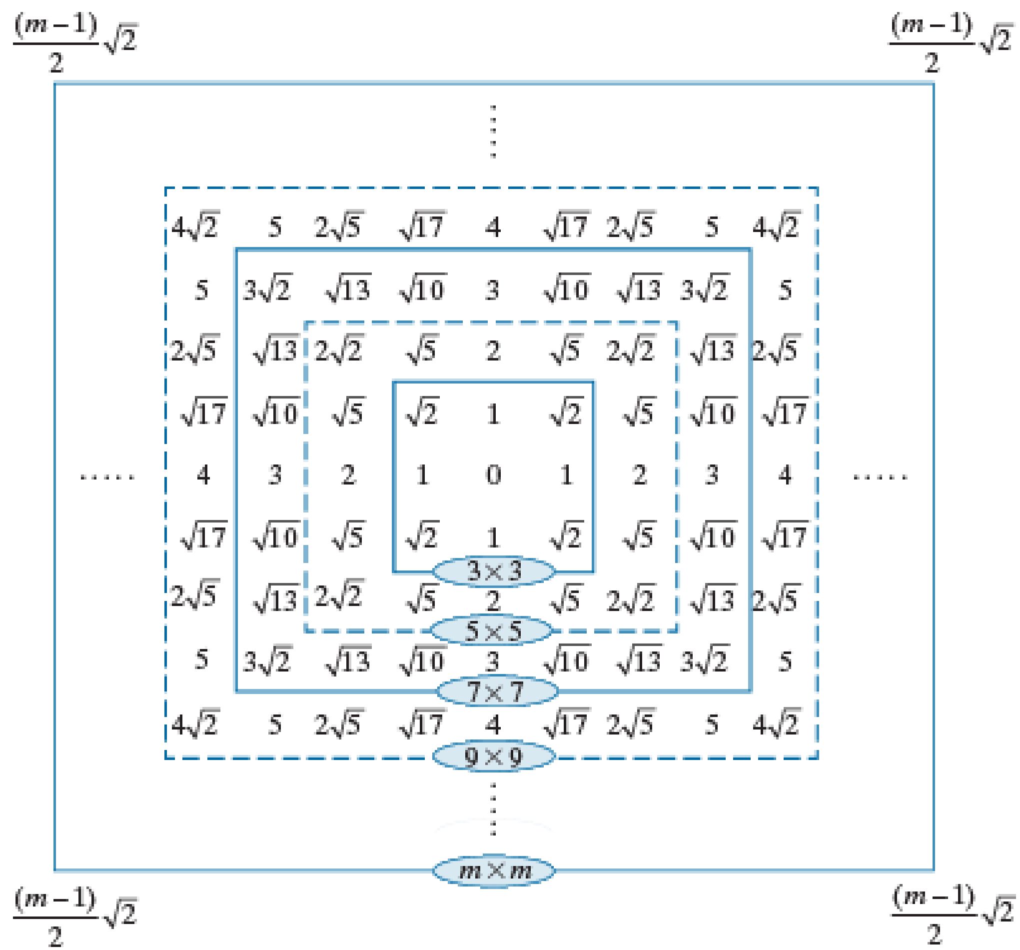
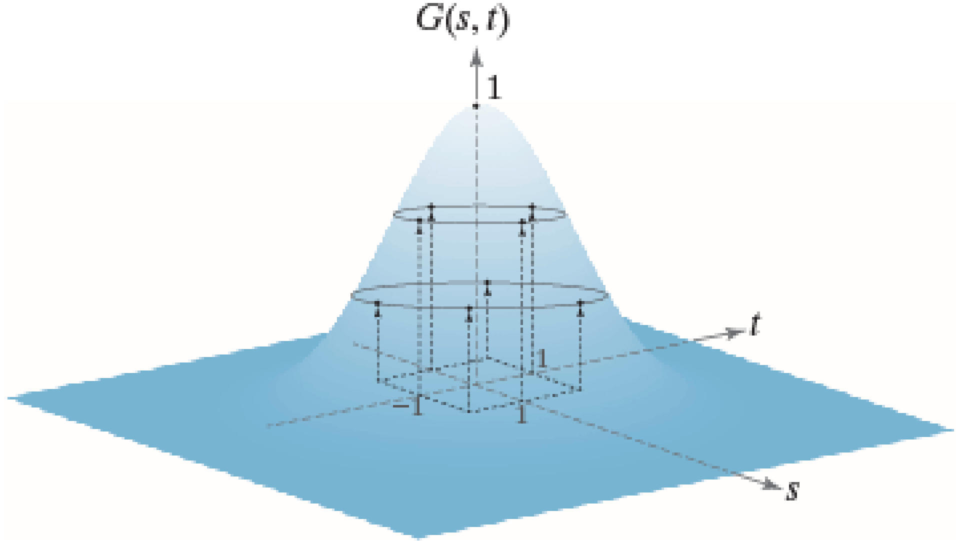


FIGURE 3.40 Distances from the center for various sizes of square kernels.

Discrete Gaussian Kernel

a b

FIGURE 3.41 (a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting 3×3 kernel [this is the same as Fig. 3.37(b)].

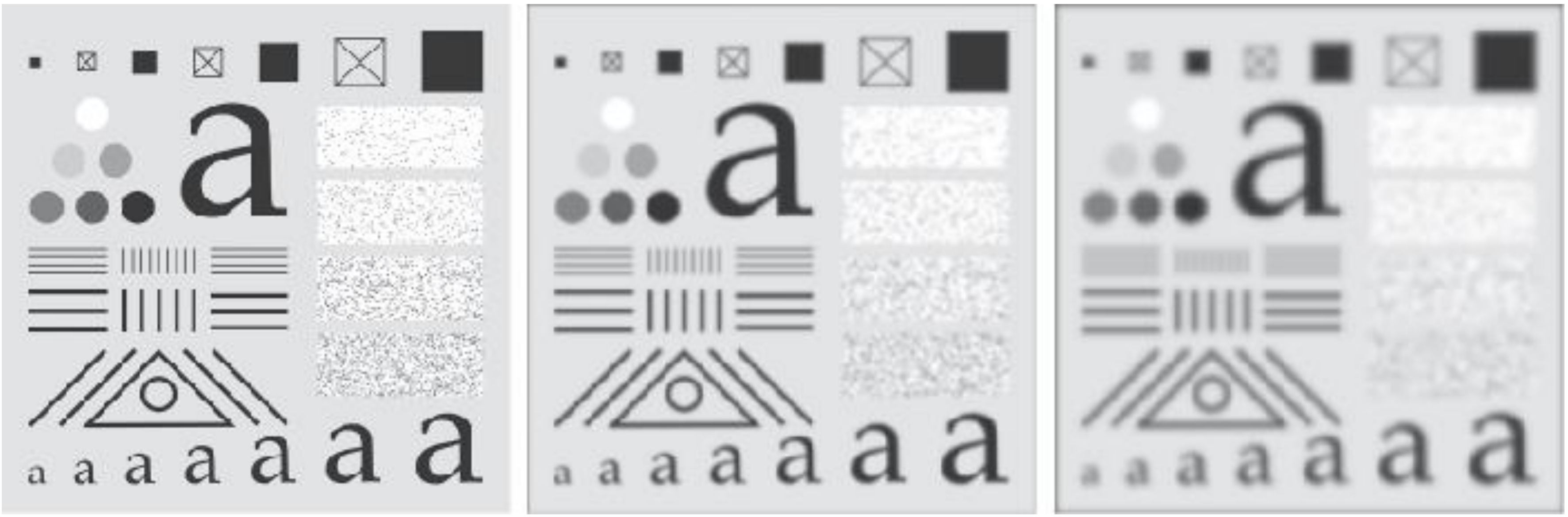


$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

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 3.33).

	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	$\sigma_{f\times g} = \sqrt{\dfrac{\sigma_f^2\sigma_g^2}{\sigma_f^2 + \sigma_g^2}}$	$\sigma_{f\star g} = \sqrt{\sigma_f^2 + \sigma_g^2}$

Lowpass Filtering with Gaussian Kernel



a b c

FIGURE 3.42 (a) A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.39(d). We used $K = 1$ in all cases.

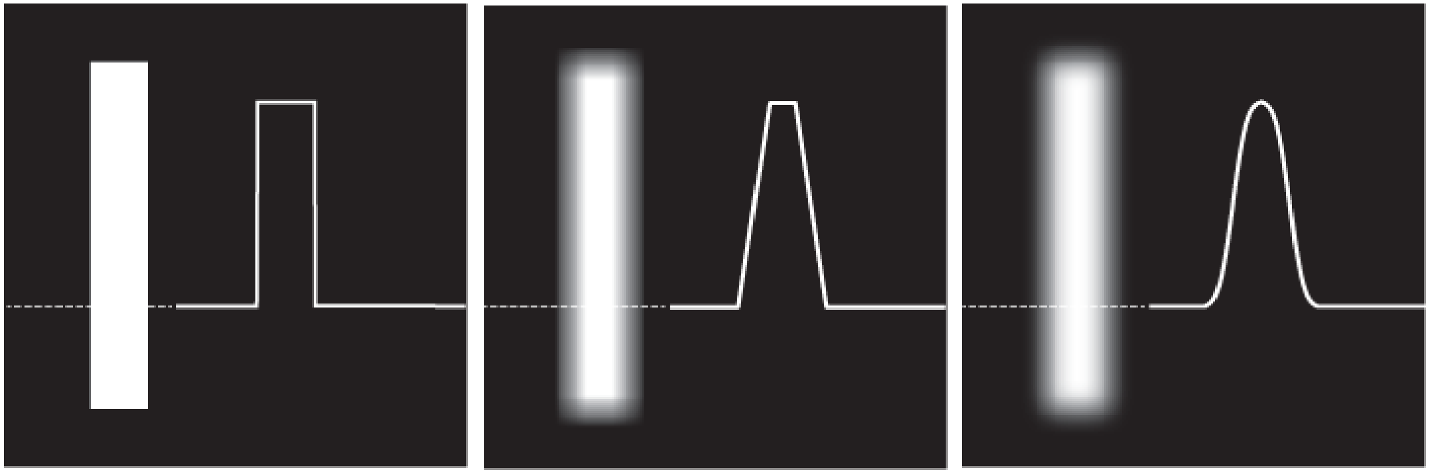
Comparison of Gaussian & Box Kernels

a b c

FIGURE 3.43 (a) Result of filtering Fig. 3.42(a) using a Gaussian kernels of size 43 × 43, with $\sigma = 7$. (b) Result of using a kernel of 85 × 85, with the same value of σ. (c) Difference image.

- **Box Filter**: Linear Smoothing (ramp shaped) at the edges
  - ➢ Preferred when less smoothing desired at edges
- **Gassian Kernel**: Significant smoothing at edge transitions
  - ➢ Preferred when more smoothing desired at edges

Comparison of Gaussian & Box Kernels



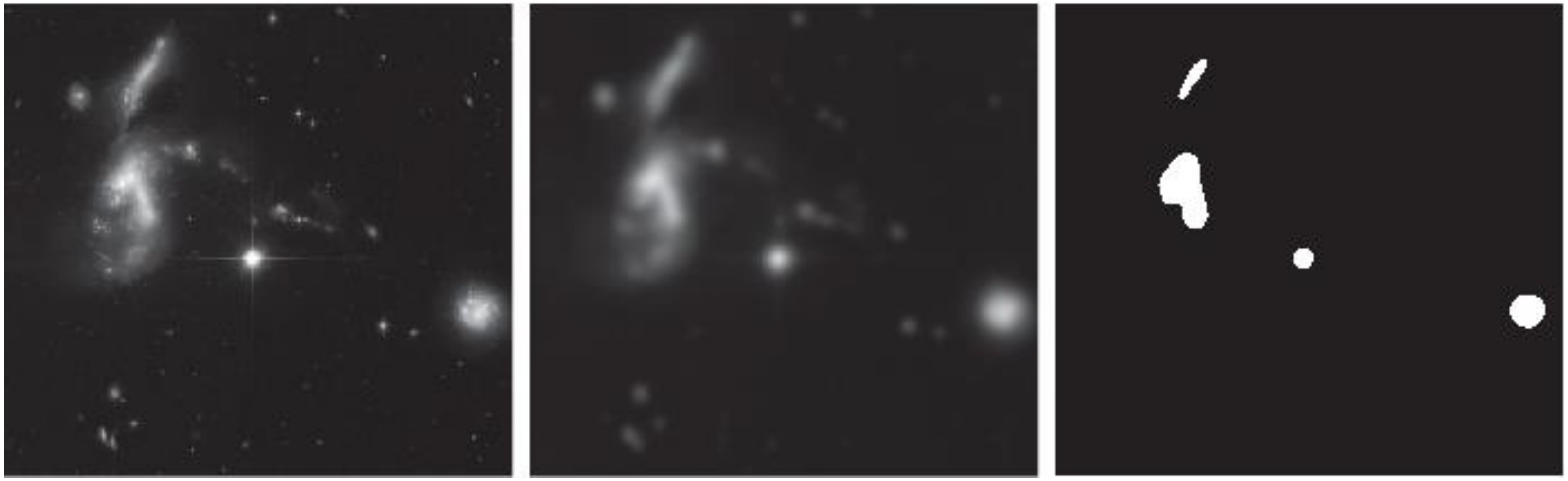
a b c

Box | Gaussian

FIGURE 3.44 (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.

- **Box Filter**: Linear Smoothing (ramp shaped) at the edges
  - ➢ Preferred when less smoothing desired at edges
- **Gassian Kernel:** Significant smoothing at edge transitions
  - ➢ Preferred when more smoothing desired at edges

Smoothing + Thresholding



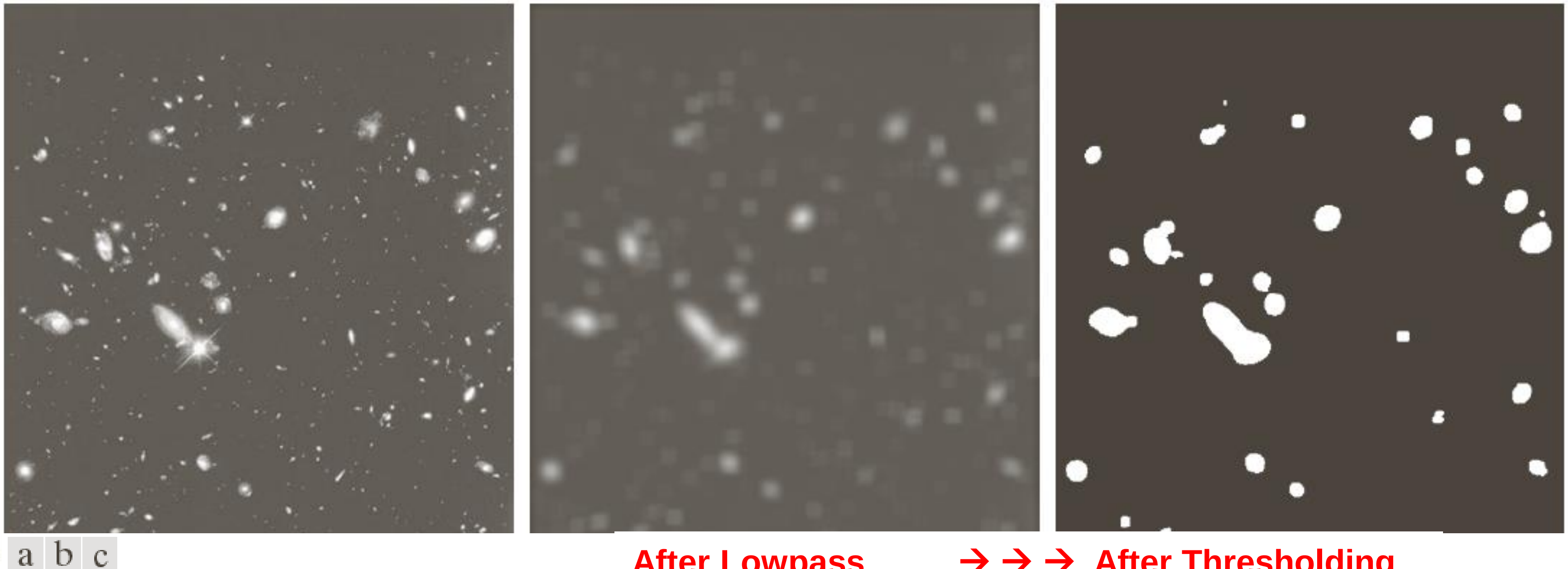
a b c

After Lowpass → → → After Thresholding

FIGURE 3.47 (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.)

- Effect: Gross representation of objects of interest
  - Smaller objects blend with the background
  - Larger objects become "bloblike"

Effect of Smoothing Filter



After Lowpass → → → After Thresholding

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.)

- Effect: Gross representation of objects of interest
  - Smaller objects blend with the background
  - Larger objects become "bloblike"

3rd Edition

Shading Correction using Lowpass Fitlering

a b c

After Lowpass → → → After Division of (a) by (b)

FIGURE 3.48 (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).

- Also known as *Flat-Field Correction*
- Useful when Shading pattern is unknown and to be estimated
- The Gaussian kernel smoothed the checker-board pattern

3.5.2 Order-Statistical Filters (Nonlinear Smoothing)

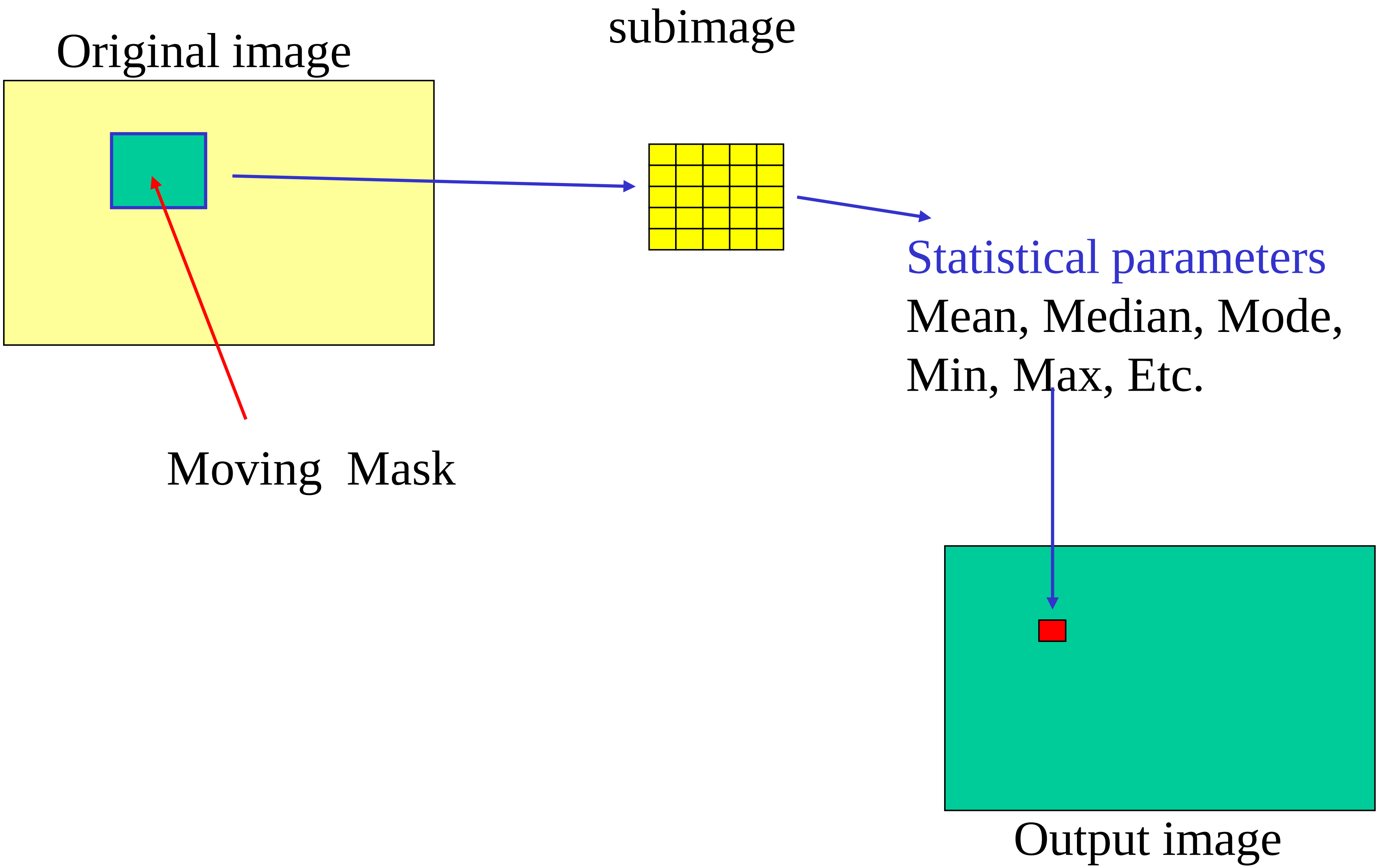
- **Objective:** Reduced Blurring
- **Steps:**
  - Order or Rank the pixels encompassed by filter mask of median filter, max filter, min filter and others (such as 50th percentile of a ranked set of numbers)
  - Replace the center pixel with the ranking result
- **Median Filter → Special class of order-statistical filter**
  - Provide excellent noise reduction with less blurring
- **Median:** Half of the intensity values less than or equal to median and other half are greater than or equal to median
  - Sort pixels in the neighborhood
  - Determine the median
  - Assign median to the corresponding pixel

Order-Statistics Filters

- **Median**
  - In 3×3 neighborhood is the 5^{th} largest value
  - In 5×5 neighborhood is the 13^{th} largest value
- Principal function of median filtering is to force pixels with distinct intensities to be more like their neighbors

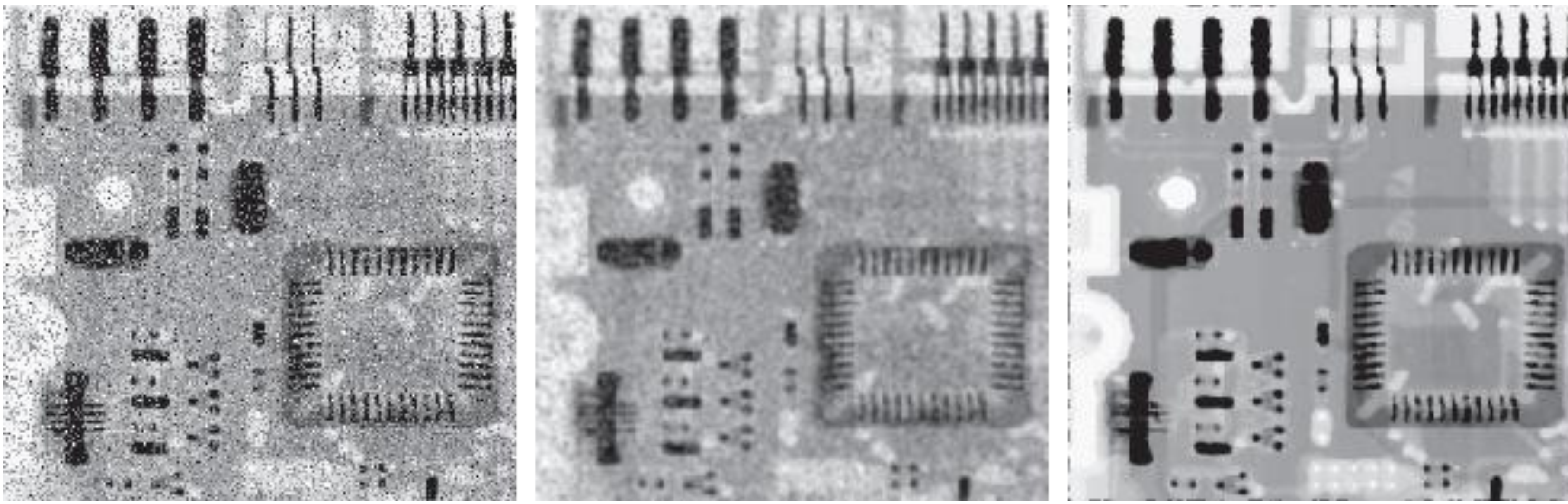
- **Other order-statistics filters**
  - **Max filter:** Find the brightest intensity in the image and assign max-value to the corresponding pixel
  - **Min filter:** Find the darkest intensity in the image and assign min-value to the corresponding pixel

Order-Statistical Filters



Salt-Pepper Noise Removal with Order-statistics Filters

- **Median Filters:** Particularly effective in the presence of **impulse noise**, also called **salt-and-pepper noise**, with considerably less blurring.



a b c

After Lowpass

After Median Filtering

FIGURE 3.49 (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

3.6 Sharpening (Highpass) Spatial Filters

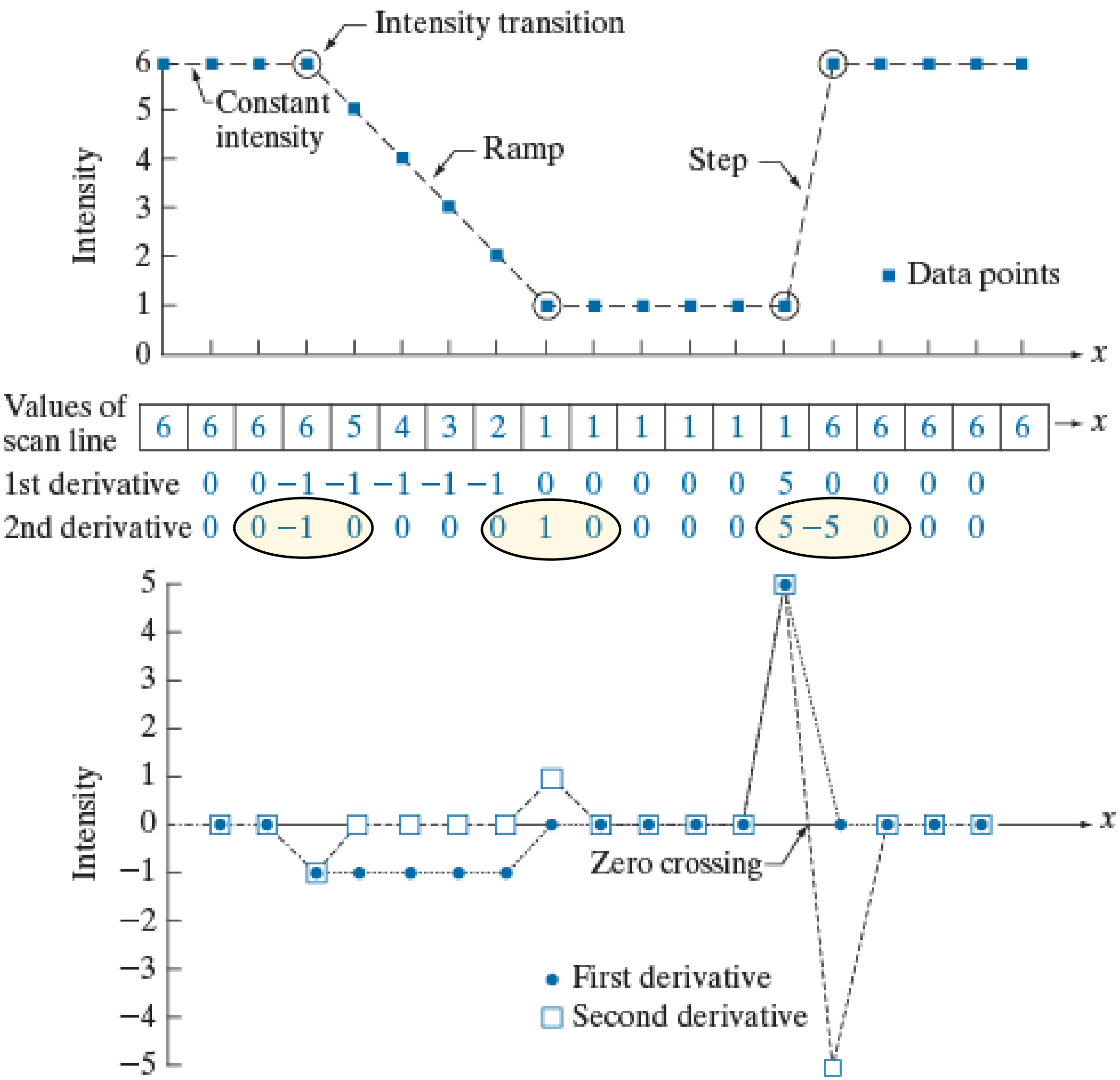
Principal Objective

- Highlight fine details → Sharpening
- Accomplished by Spatial differentiation or derivative operator
- Strength of the response of a derivative operator is proportional to the degree of discontinuity
- Image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray-level values.

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

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

First and Second Derivatives

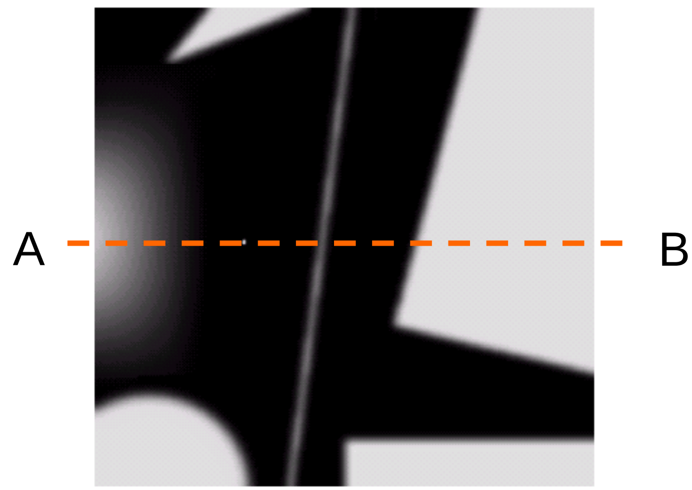


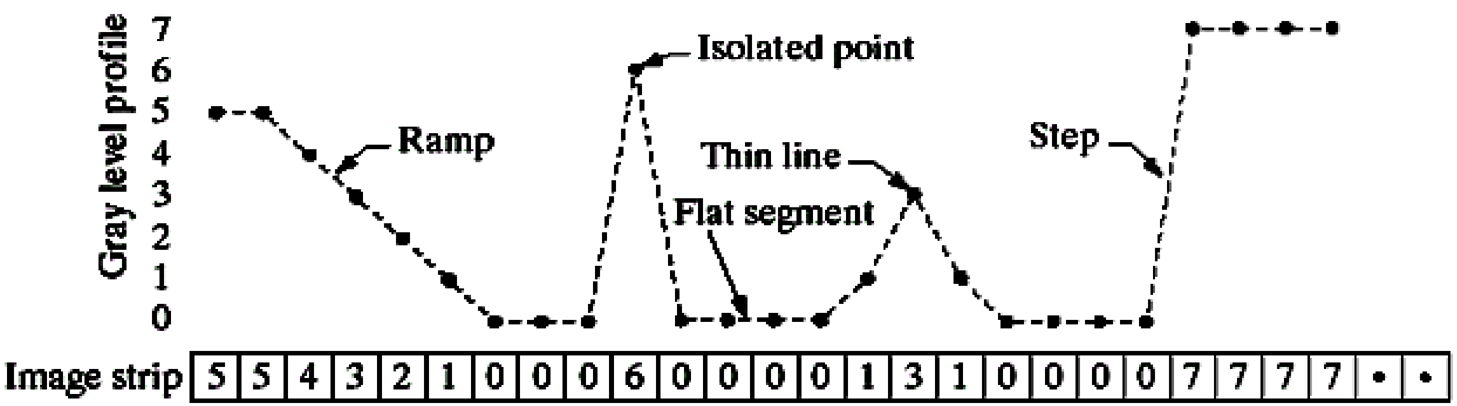
a
b
c

FIGURE 3.50
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.
(b) Values of the scan line and its derivatives.
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.

Spatial Differentiation Example

- Differentiation measures the *rate of change* of a function
- Consider another 1 dimensional example





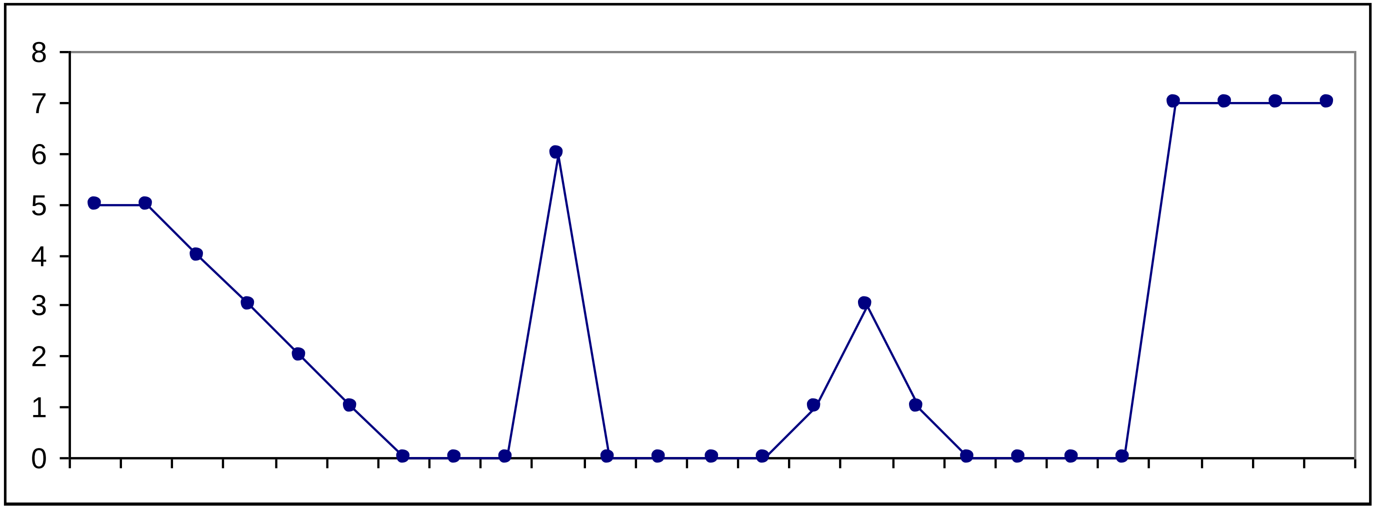
1st Derivative

- Formula for 1st derivative of a function :

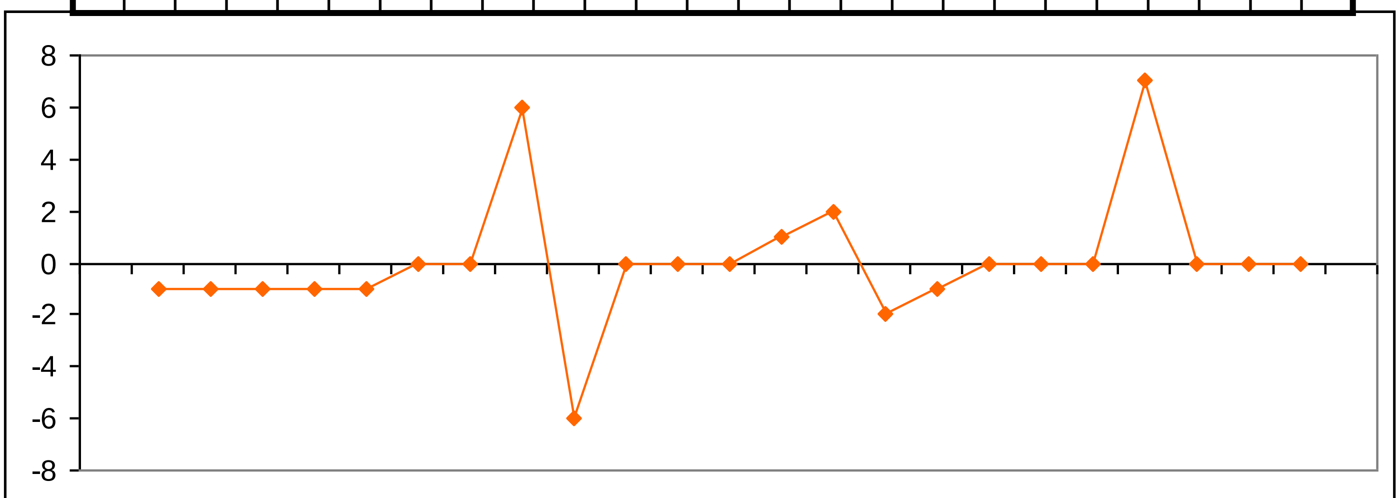
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- It's just the difference between successive values
- Measures rate of change of the function

1st Derivative (cont…)



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
	0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	



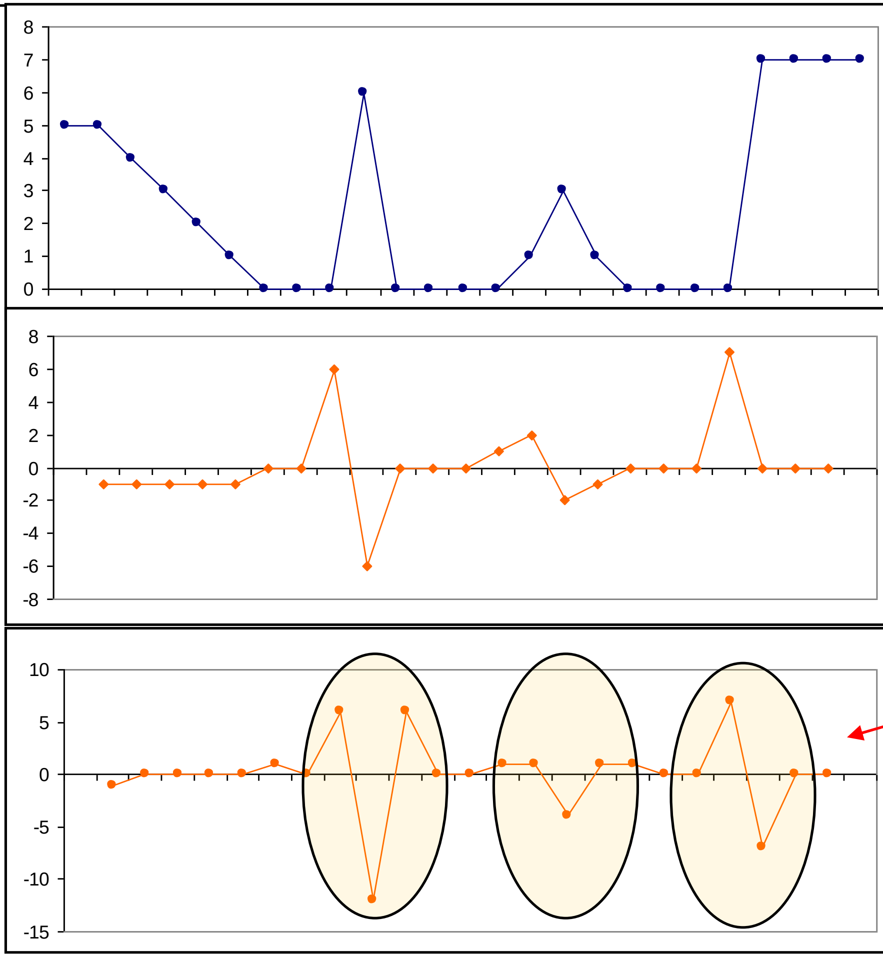
2^{nd} Derivative

- Formula for the 2^{nd} derivative of a function :

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

- Simply takes into account the values both before and after the current value

2nd Derivative (cont…)



Show Zero-Crossings at Strong Edges and isolated points

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

0	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	
	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	

Comparison of the response between first- and second-order derivatives:

1) First-order derivatives generally produce thicker edges in an image
2) Second-order derivatives have a stronger response to fine detail, such as thin lines and isolated points
3) First-order derivatives generally have a stronger response to a gray-level step
4) Second-order derivatives produce a double response at step changes in gray level (double-edge effect)
5) For similar changes in gray-level values, the response of second-order derivatives is stronger to a line than to step, and to a point than to a line

3.6.2 Using Second Derivatives for Image Sharpening → The Laplacian

- 2^{nd} derivative is more useful for image enhancement than 1^{st} derivative
  - Stronger response to fine detail
  - Simpler implementation
  - Will revisit 1^{st} order derivative later

- The first sharpening filter → *Laplacian*
  - Isotropic
  - One of the simplest sharpening filters
  - Need discrete/digital implementation

3.6.2 The Laplacian Operator

- Laplacian is a type of linear, isotropic filter
- Rotation invariant

→ Independent of direction of discontinuities in the image

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

- Partial 2nd order derivative in the x-direction

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

- y-direction

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

- Combined, the digital implementation of Laplacian: (3.62)

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

Filter Mask of Digital Laplacian

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a b c d

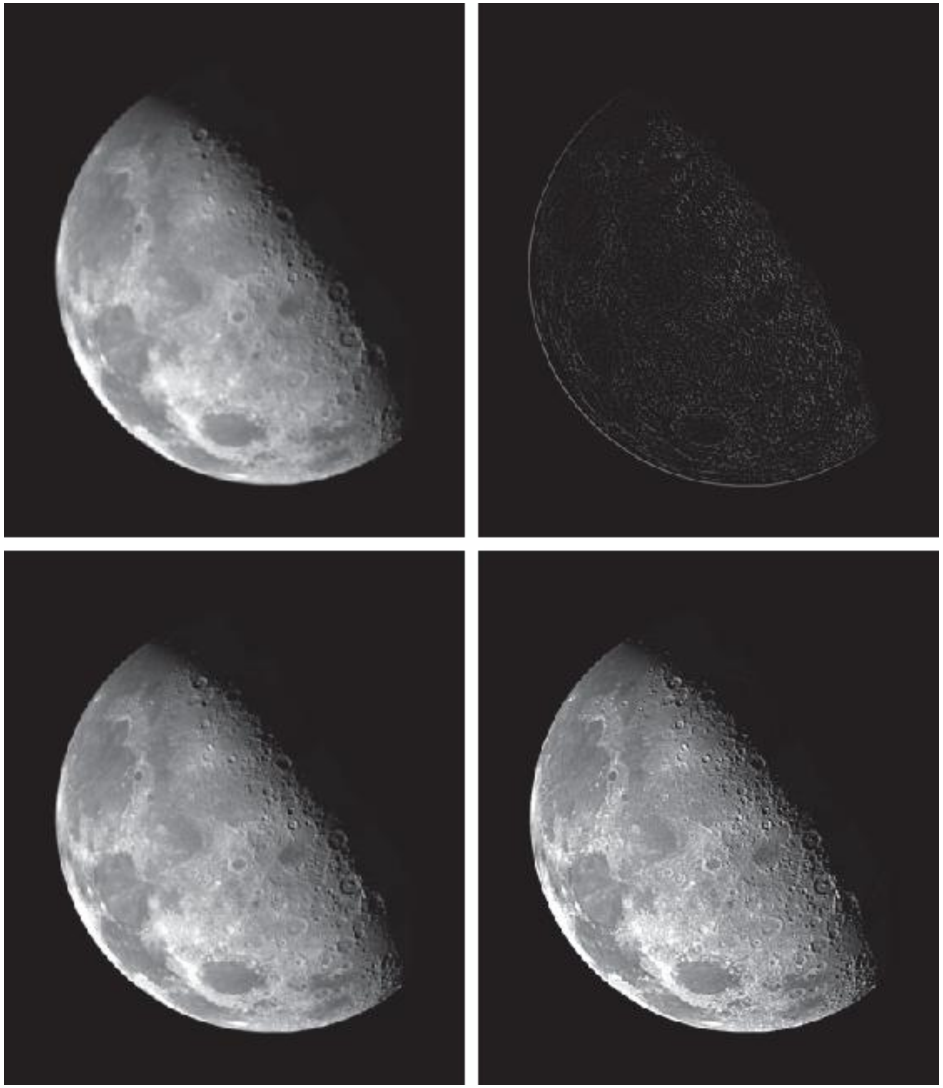
FIGURE 3.51 (a) Laplacian kernel used to implement Eq. (3-62). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

- Basic Image Sharpening equation

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right] \qquad (3.63)$$

  - Use c = -1 for Laplacian Kernels in Fig. 3.51 (a) or (b)
  - Use c = +1 for Laplacian Kernels in Fig. 3.51 (c) or (d)

Effect of Laplacian



Why is the appearance grayish?
Negative values are clipped to 0

a b
c d

FIGURE 3.52
(a) Blurred image of the North Pole of the moon.
(b) Laplacian image obtained using the kernel in Fig. 3.51(a).
(c) Image sharpened using Eq. (3-63) with $c = -1$.
(d) Image sharpened using the same procedure, but with the kernel in Fig. 3.51(b).
(Original image courtesy of NASA.)

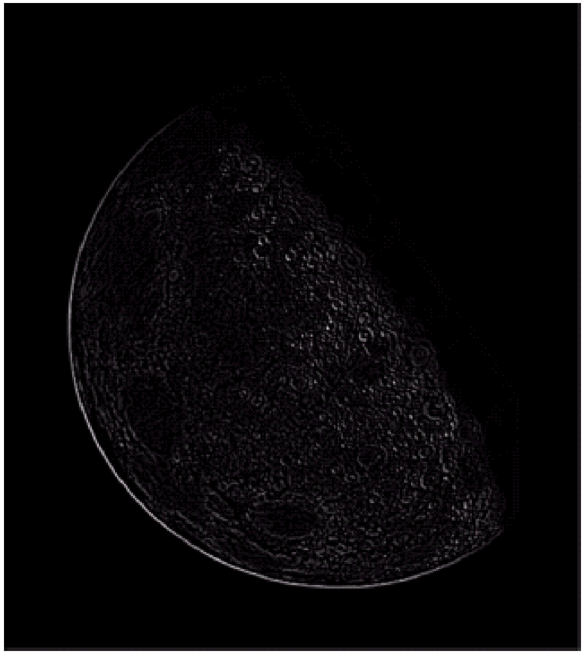
Effect of Laplacian

FIGURE 3.53 The Laplacian image from Fig. 3.52(b), scaled to the full [0, 255] range of intensity values. Black pixels correspond to the most negative value in the unscaled Laplacian image, grays are intermediate values, and white pixels corresponds to the highest positive value.

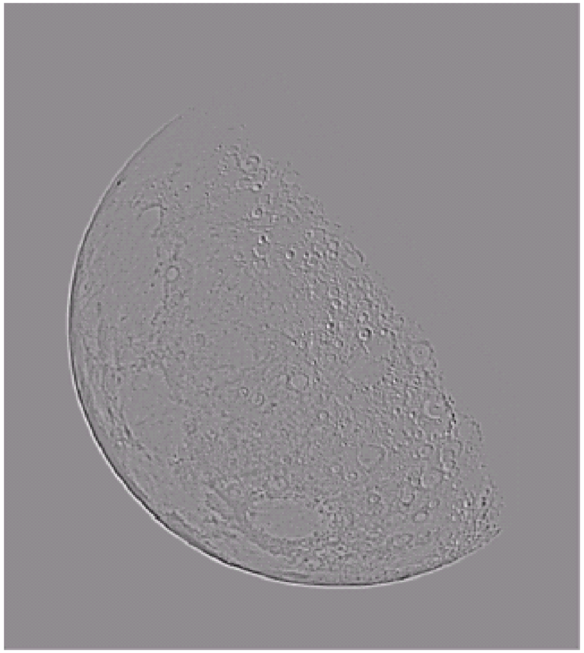
The Laplacian (cont…)

Applying the Laplacian to an image produces a new image that highlights edges and other discontinuities

Original Image



Laplacian Filtered Image

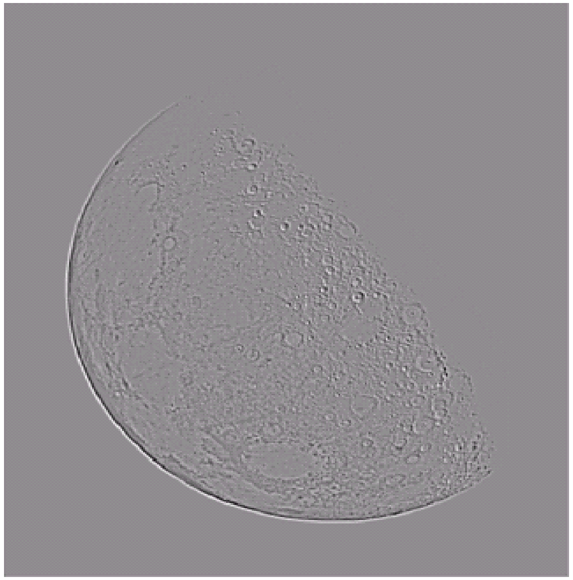


Laplacian Filtered Image Scaled for Display

Result: Not Very Enhanced

- The result of a Laplacian filtering is not an enhanced image
- Need additional steps to get final image
- Subtract the Laplacian result from the original image to generate the final sharpened enhanced image

$$g(x,y) = f(x,y) - \nabla^2 f(x,y)$$

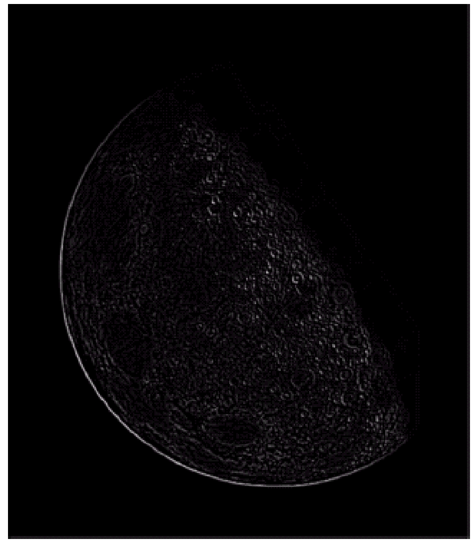


Laplacian Filtered Image Scaled for Display

Laplacian Image Enhancement

Original Image

\-



Laplacian Filtered Image

=

Sharpened Image

In the final sharpened image edges and fine detail are much more obvious

Laplacian Image Enhancement

Original Image	Sharpened Image
	

3.6.3 Unsharp Masking and High-Boost Filtering

- Main Idea of Unsharp masking:
  - Sharpen Image using Lowpass Filtering → How?
    - Blur original image: $\bar{f}(x,y)$
    - Subtract the blurred version of image from the original

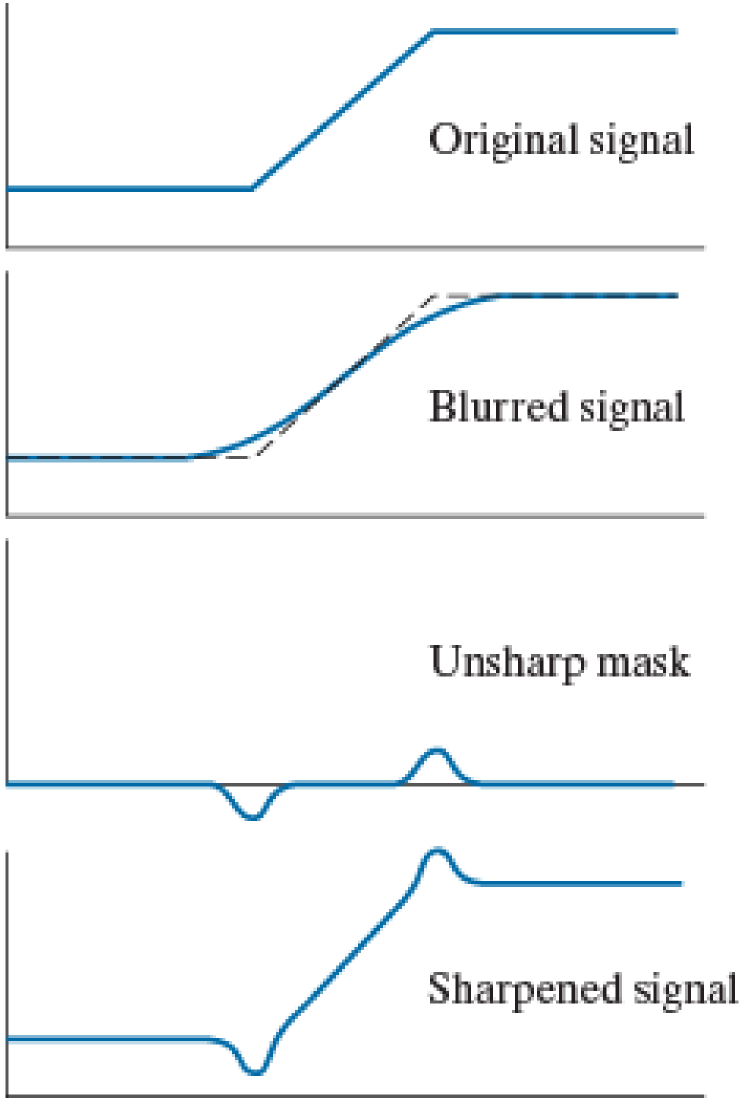
$$g_{\text{mask}}(x,y) = f(x,y) - \bar{f}(x,y)$$ → Unsharp Mask

    - Add the previous result to the original

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

    - Three Cases:
      - $k = 1$ → Unsharp Masking
      - $k > 1$ → Highboost Filtering
      - $k < 1$ → Deemphasizes contribution of the unsharp mask

Illustration of Unsharp Masking



a
b
c
d

FIGURE 3.54 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).

Illustration of Unsharp Masking

a b c
d e f

FIGURE 3.55 (a) Unretouched "soft-tone" digital image of size 469×600 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$. (c) Mask. (d) Result of unsharp masking using Eq. (3-65) with $k = 1$. (e) and (f) Results of highboost filtering with $k = 2$ and $k = 3$, respectively.

3.6.4 First-Order Derivatives for Sharpening – The Gradient

- First derivatives in image processing are implemented using the magnitude of the gradient defined as,

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

  - Magnitude of gradient vector:

$$M(x, y) = \|\nabla f\| = \text{mag}(\nabla f) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} = \sqrt{g_x^2 + g_y^2}$$

- Components of gradient: Linear operators, but not rotation invariant (i.e., non-isotropic)
- Magnitude of gradient: Not linear, but rotation invariant
- Usually in image processing, the magnitude of the gradient is referred as the gradient

First-Order Derivatives for Sharpening – Gradient

The masking forms of the above operators

Let, $z_5 = f(x,y)$; Then, $z_1 = f(x-1, y-1)$;
$z_6 = f(x, y+1)$ and so on

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$(z_9 - z_5)$

$(z_8 - z_6)$: cross-difference

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Note: Sum of all coefficients is 0

a
b c
d e

FIGURE 3.56 (a) A 3 × 3 region of an image, where the zs are intensity values. (b)–(c) Roberts cross-gradient operators. (d)–(e) Sobel operators. All the kernel coefficients sum to zero, as expected of a derivative operator.

Approximation of First-Order Derivatives

- To reduce computational burden, following approximation is also used as gradient. But the isotropic feature is lost.

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

- Digital approximations of gradient:
  - Roberts cross-gradient operators

$$g(x) = (z_9 - z_5),$$

$$g(y) = (z_8 - z_6)$$

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

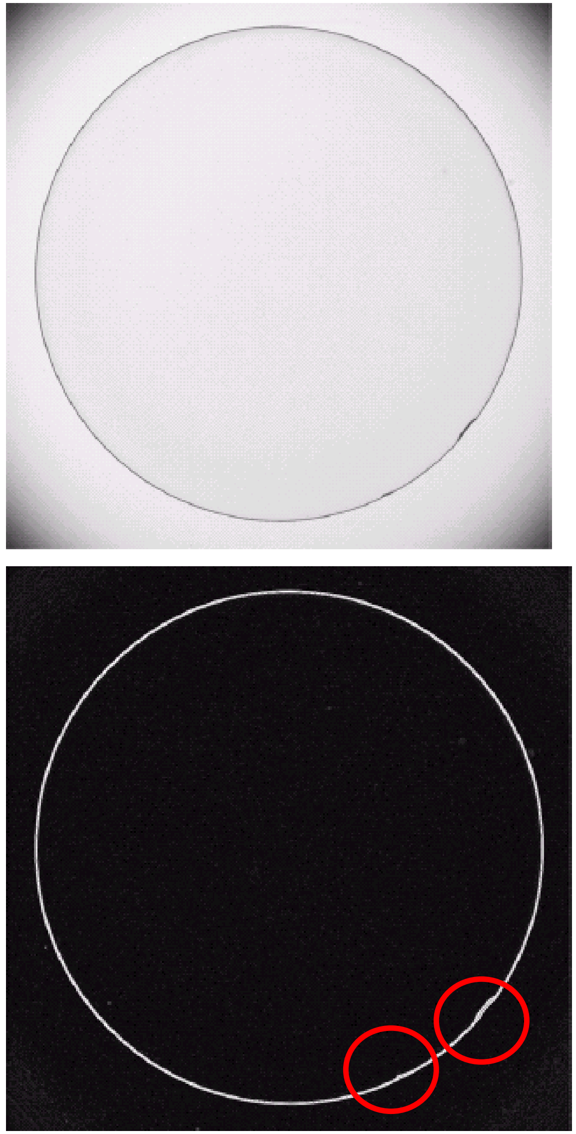
  - Sobel operators

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

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

Effect of Sobel Gradient

Example: Enhance the defect and eliminate slowly changing background in industrial inspection

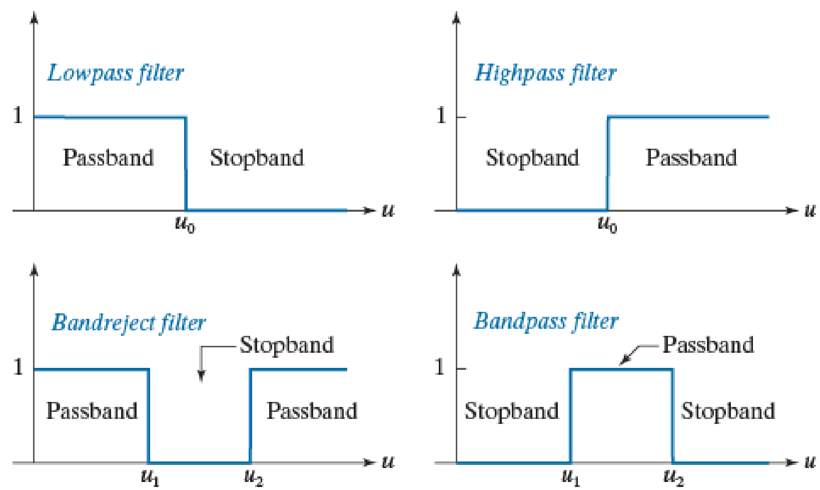


a b

FIGURE 3.57 (a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Perceptics Corporation.)

3.7 Highpass, Bandpass, and Banreject filters from Lowpass Filters

- Transfer functions of 1-D ideal Filters



- Ideal Filters in terms of Lowpass Filter

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	$bp(x,y) = \delta(x,y) - br(x,y)$ $= \delta(x,y) - [lp_1(x,y) + [\delta(x,y) - lp_2(x,y)]]$

- Only prototype LPF design is needed

3.7 Highpass, Bandpass, and Banreject filters from Lowpass Filters (cont.)

- Zone plate image used for experiments

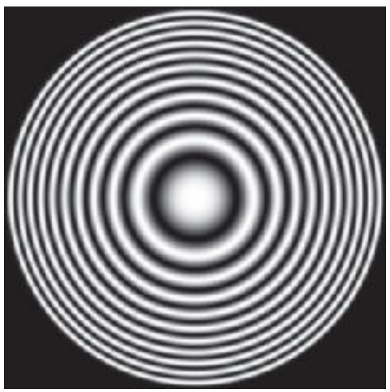
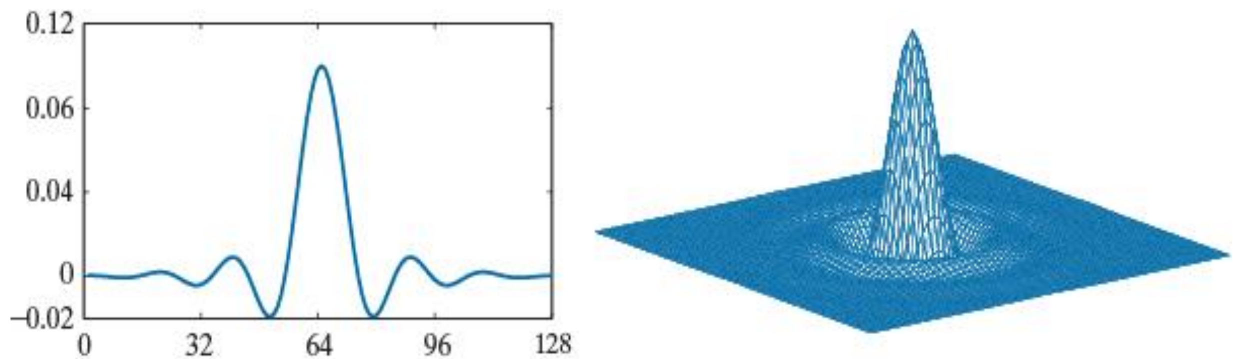


FIGURE 3.59
A zone plate image of size 597×597 pixels.

- Isotropic Gaussian kernel

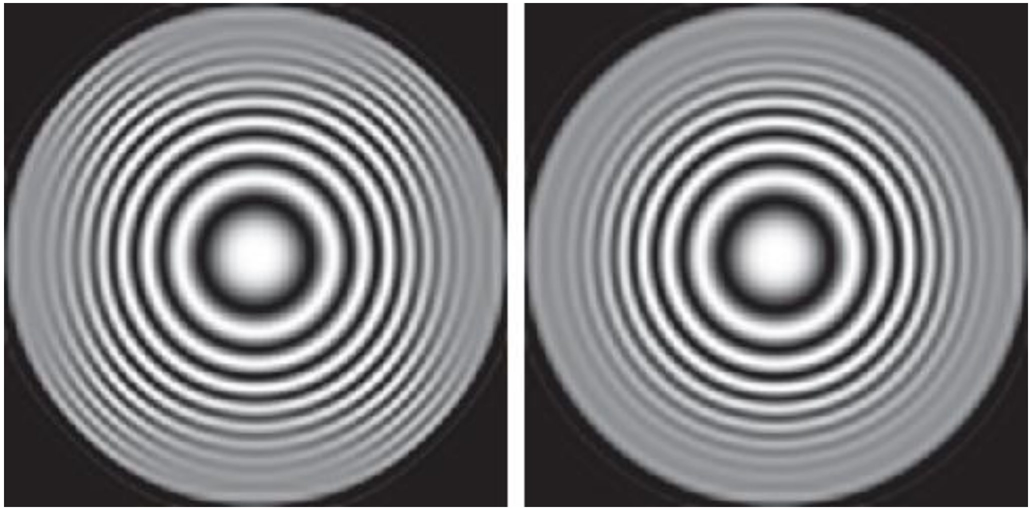


a b

FIGURE 3.60
(a) A 1-D spatial lowpass filter function. (b) 2-D kernel obtained by rotating the 1-D profile about its center.

3.7 Difference between Filtering with Isotropic and Separable Filters

- Separable filters: Square-symmetric
- Isotropic filters: Radially symmetric



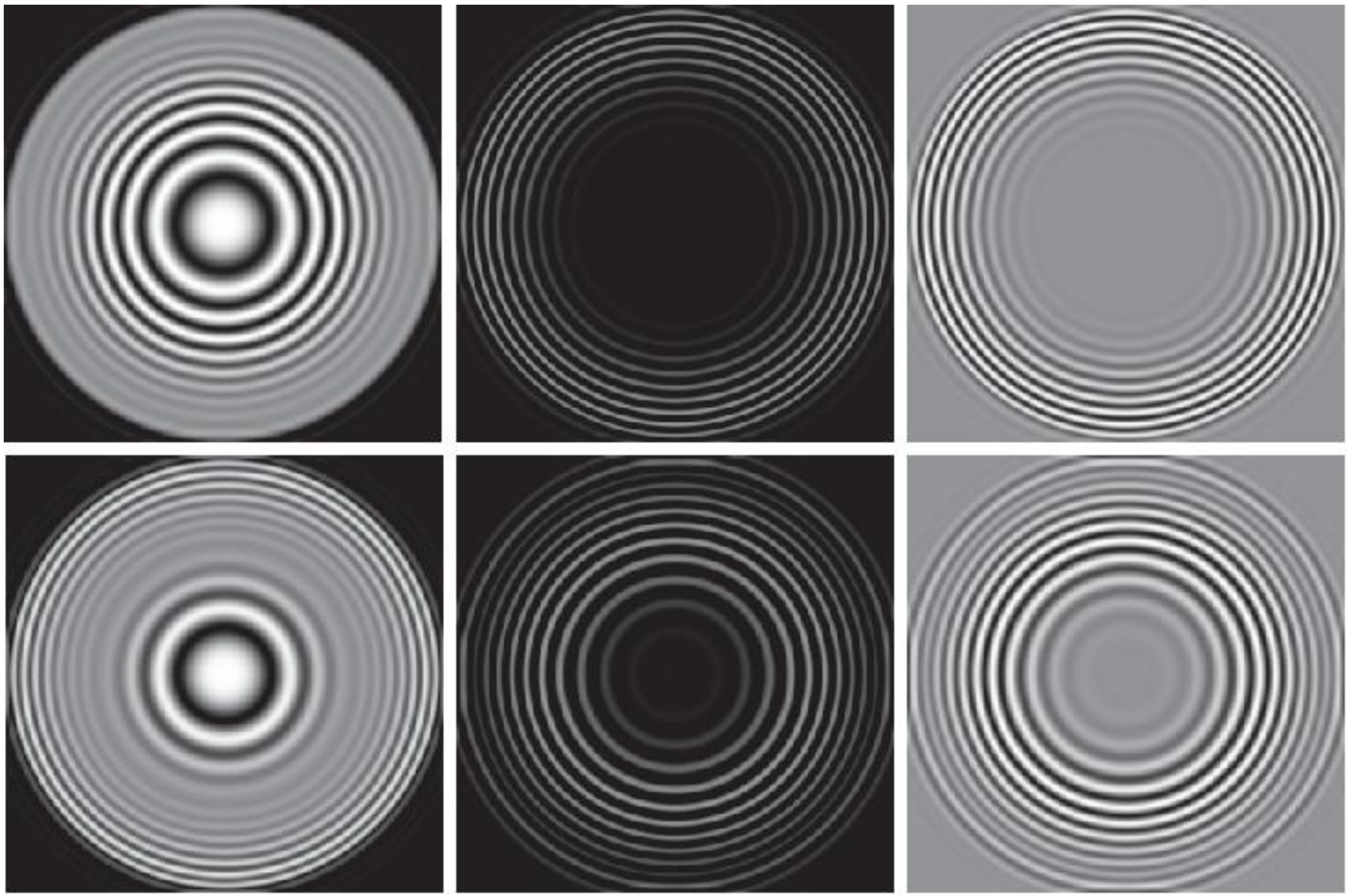
a b

FIGURE 3.61 (a) Zone plate image filtered with a separable lowpass kernel. (b) Image filtered with the isotropic lowpass kernel in Fig. 3.60(b).

- Separbale filter output (left) is "Squarish"
- Isotropic kernel yielded more uniform output (right) in all radial directions

Effect of Highpass, Bandpass, and Banreject filtering

- Isotropic LPF Gaussian kernel in Fig 3.60(b) used



a b c
d e f

FIGURE 3.62
Spatial filtering of the zone plate image. (a) Lowpass result; this is the same as Fig. 3.61(b). (b) Highpass result. (c) Image (b) with intensities scaled. (d) Bandreject result. (e) Bandpass result. (f) Image (e) with intensities scaled.

3.8 Combining Spatial Enhancement Methods

- Successful image enhancement is typically not achieved using a single operation
- Rather, a number of techniques are combined in order to achieve a final result
- This example will focus on enhancing the bone scan to the right

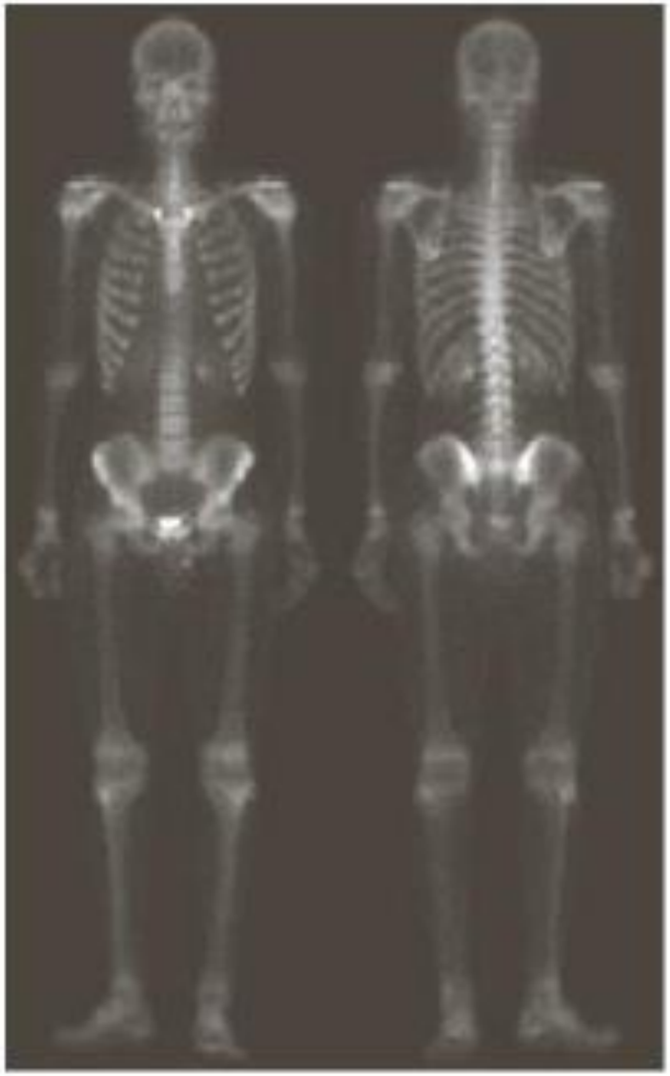


FIGURE 3.63
(a) Image of whole body bone scan.

3.7 Combination of Spatial Enhancement Methods

Some enhancement tasks require application of several complementary enhancement techniques in order to achieve an acceptable result.

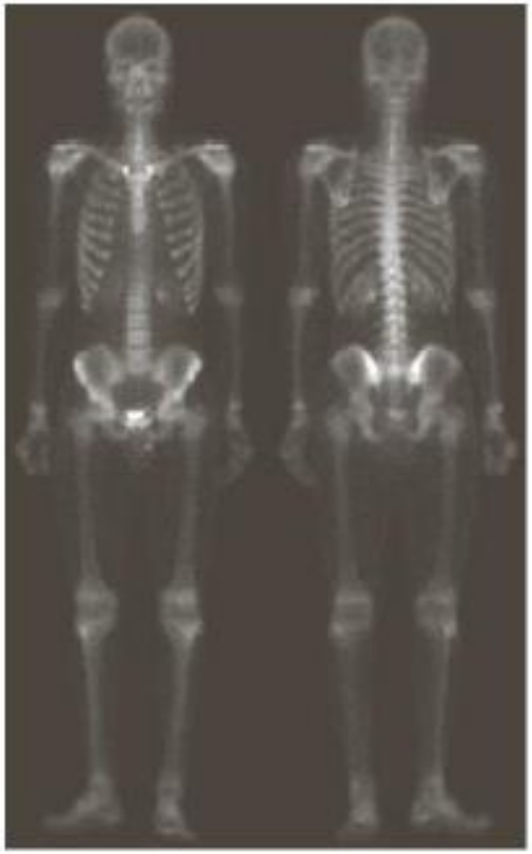


FIGURE 3.63
(a) Image of whole body bone scan.

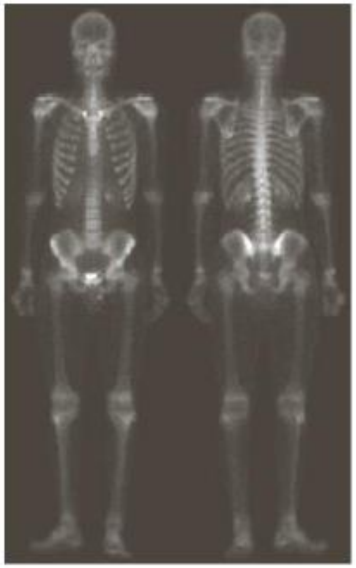
- The image shown left is a **Nuclear Whole Body Bone Scan**, used to detect diseases such as bone infection and tumors.
- **Objective:** Enhance image by sharpening and bringing out more of the skeletal detail.
- **Issues:** The **narrow dynamic range** of the **low gray levels** and **high noise content** make this image difficult to enhance.

WRIGHT STATE UNIVERSITY

The Strategy

- Utilize the *Laplacian* to highlight fine detail
- Utilize the *gradient* to enhance prominent edges
- Combine *Laplacian* and *gradient* to get the detail-enhanced and noise-compressed image
- Increase the contrast of low gray levels by using a gray-level transformation.

Combining Spatial Enhancement Methods (cont…)

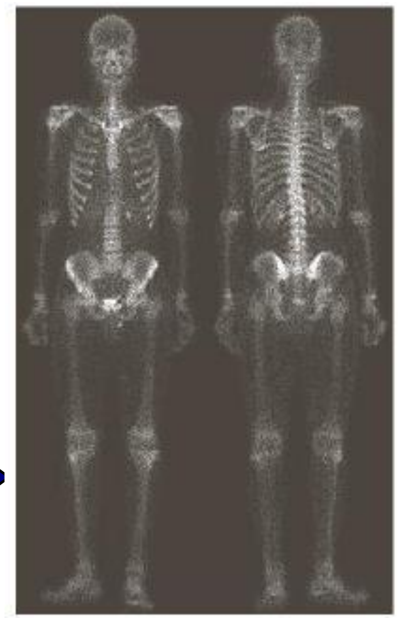


(a)

Laplacian filter of bone scan (a)

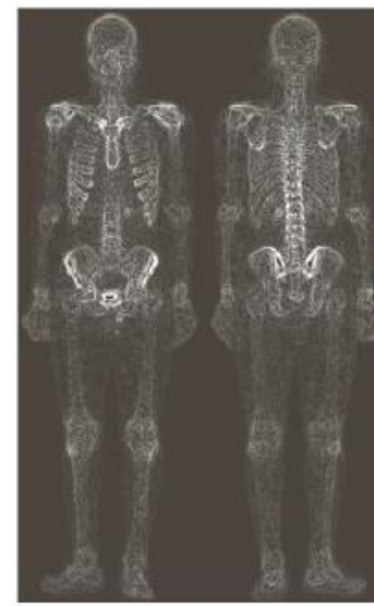
(b)

Sharpened version of bone scan achieved by adding (a) and (b)



(c)

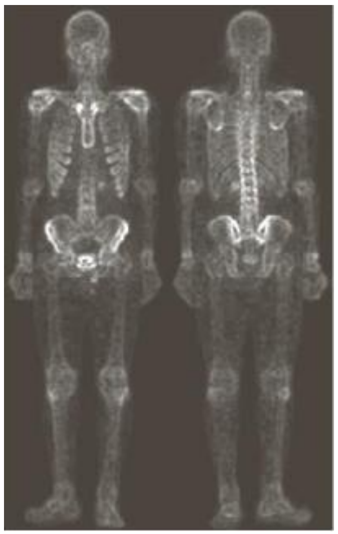
Sobel gradient of bone scan (a)



(d)

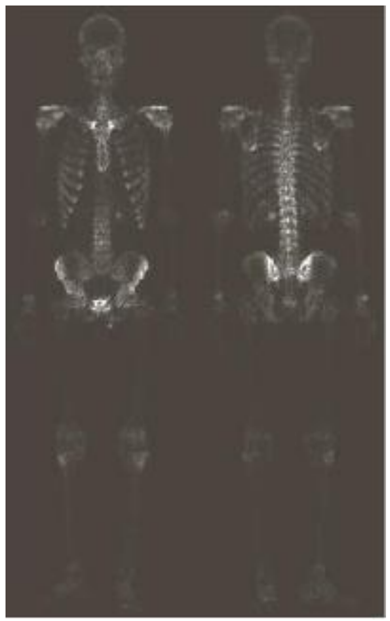
Combining Spatial Enhancement Methods (cont…)

Image (d) smoothed with a 5*5 averaging filter



(e)

The product of (c) and (e) which will be used as a mask



(f)

Sharpened image which is sum of (a) and (f)

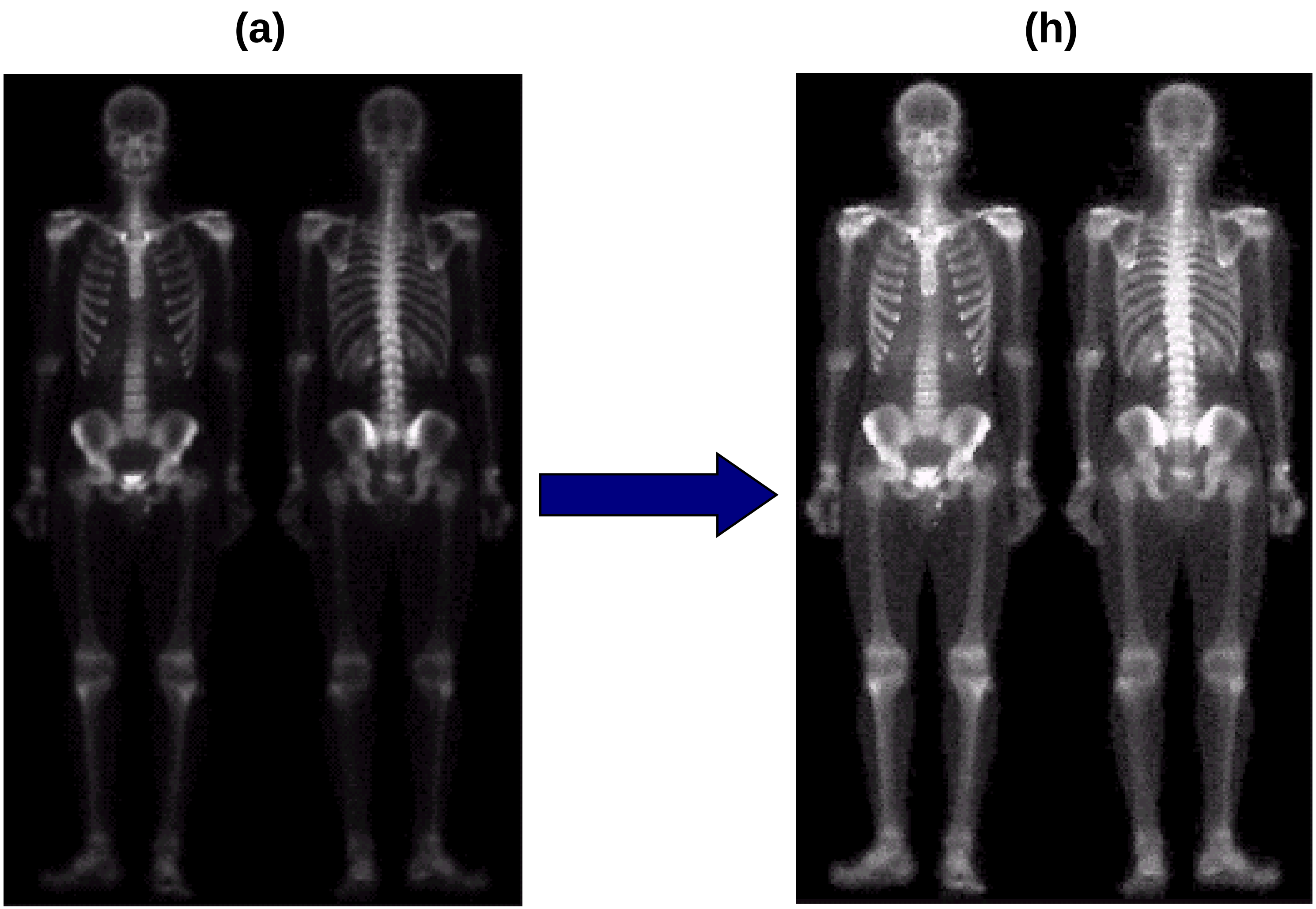
(g)

Result of applying a power-law transform to (g)

(h)

Combining Spatial Enhancement Methods (cont…)

Compare the original (a) and final (h) images

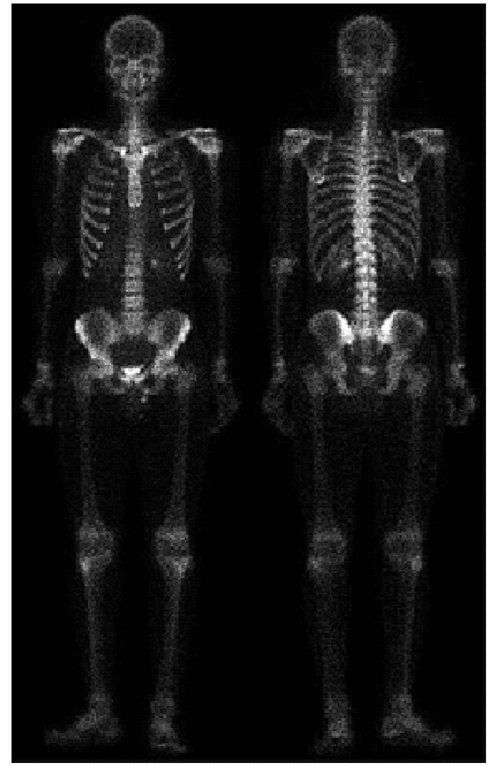


Laplacian Enhancement

(b)

−1	−1	−1
−1	8	−1
−1	−1	−1

$$f(x,y) + \nabla^2 f(x,y)$$



(c)

A rather noisy sharpened image is expected. Median filter is incapable of removing noise in such medical images

Smoothed Gradient as a Mask

- The response of the *gradient* to noise and fine detail is lower than the *Laplacian*'s and can be lowered further by smoothing the *gradient* with an averaging filter.
- We can smooth the *gradient* and multiply it by the *Laplacian* enhanced image. In this case the smoothed *gradient* may be viewed as a mask image.

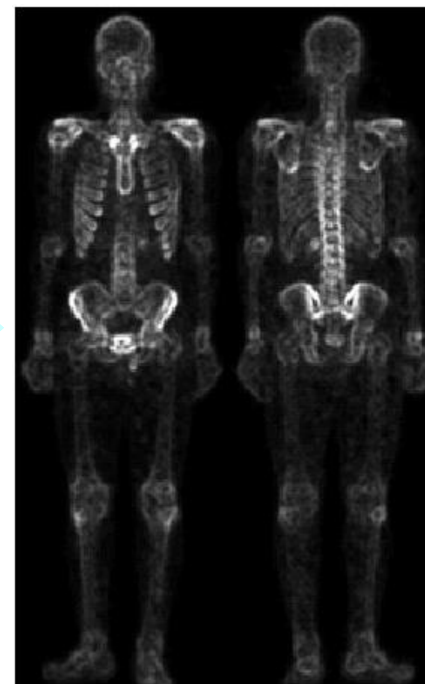
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

(d)

5 x 5 box smooth

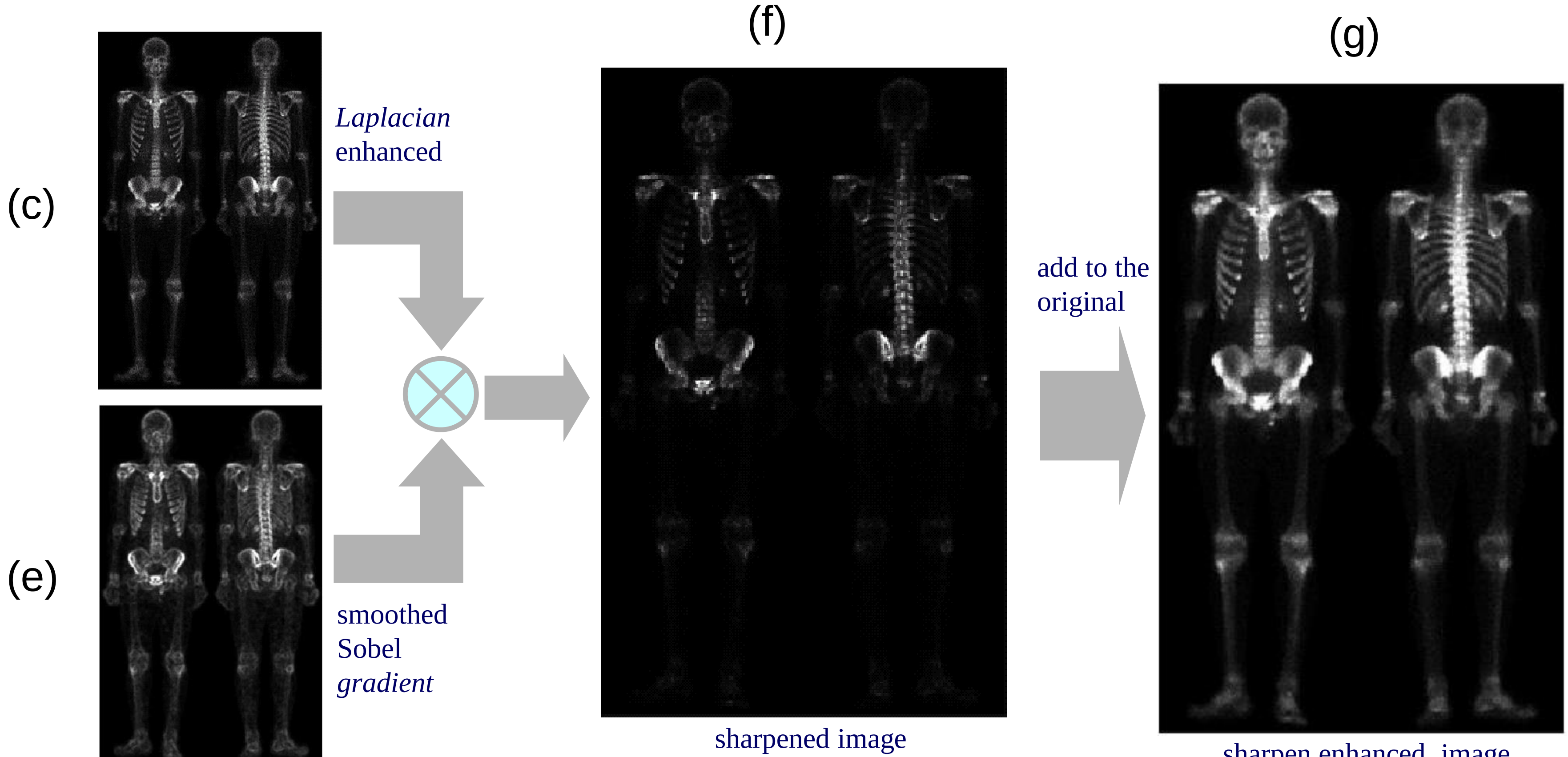


(e)

mask image

The Sharpened Enhanced Image

The final sharpen enhanced image is obtained from the sum of original image and the sharpened image which comes from the product of *Laplacian* enhanced image and the smoothed Sobel *gradient*.



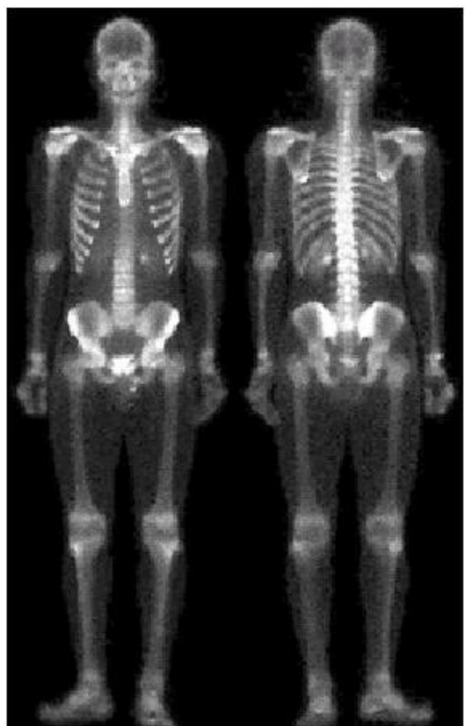
sharpened image
strong edges and the relative lack of visible noise

sharpen enhanced image

Contrast Stretch and the final Result

- The final step of enhancement task is to increase the contrast of the sharpened image. There are a number of gray level transformation functions that can accomplish this objective. The dark characteristics of the images lend themselves to a power-law transformation.

(g)



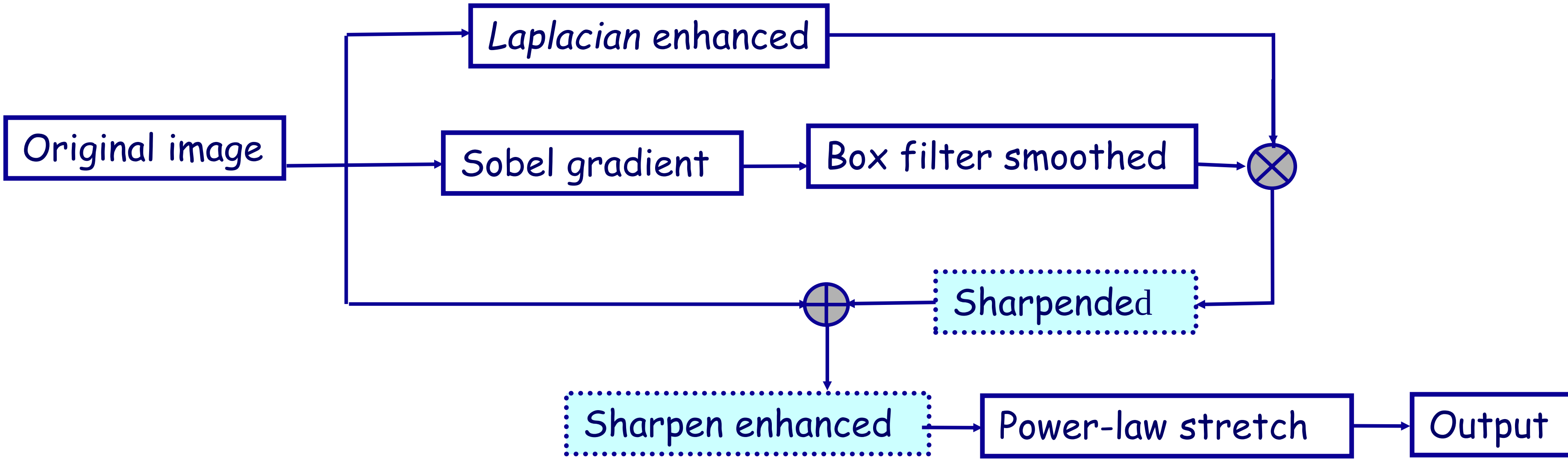
(h)

$c = 1$

$\gamma = 0.5$

- Significant new detail is visible in the result, including the faint definition of the outline of the body, and of body tissue.

Overview of the Processing Flow in the Previous Example



- The way in which the results are used depends on the application and the user.
- Enhanced images are quite useful in highlighting details that can serve as clues for further analysis in the original image or sequence of images.
- There are many areas in which the enhanced result may indeed be the final product, and the principal objective of enhancement is to obtain an image with a higher content of visual detail.

Summary

- The material is representative of spatial domain techniques commonly used in practice for image enhancement.
- This area of image processing is a dynamic field, and new techniques and applications are reported routinely in professional literature and in new product announcements.
- For this reason, the topics included in this Chapter were selected for their value as fundamental material that would serve as a foundation for understanding the state of the art in enhancement techniques, as well as for further study in this field.
- In addition to enhancement, this Chapter served the purpose of introducing a number of concepts, such as filtering with spatial masks, that will be used in numerous occasions throughout image processing

Acknowlegements

Slides are primarily based on the figures and images in the Digital Image Processing textbook by Gonzalez and Woods:

- http://www.imageprocessingplace.com/DIP-3E/dip3e_book_images_downloads.htm

In addition, slides have been adopted and modified from the following excellent sources:

- http://www.comp.dit.ie/bmacnamee/gaip.htm
- http://baggins.nottingham.edu.my/~hsooihock/G52IIP/
- http://gear.kku.ac.th/~nawapak/178353.html