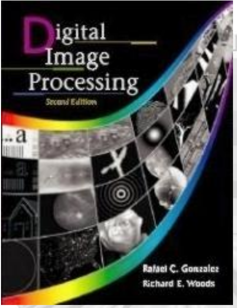
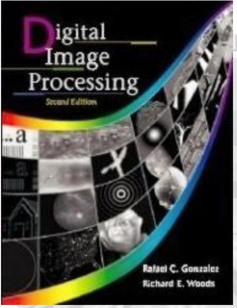


Chapter 3
Intensity Transformations & Spatial Filtering

Md. Hasanul Kabir, Ph.D.

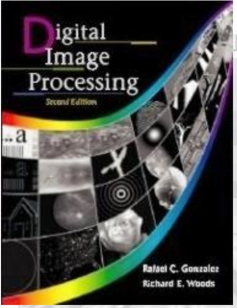


Objectives of Enhancement

- **Process an image so that the result will be more suitable than the original image for a specific application.**
- **The suitableness is up to each application.**
- **A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another images**

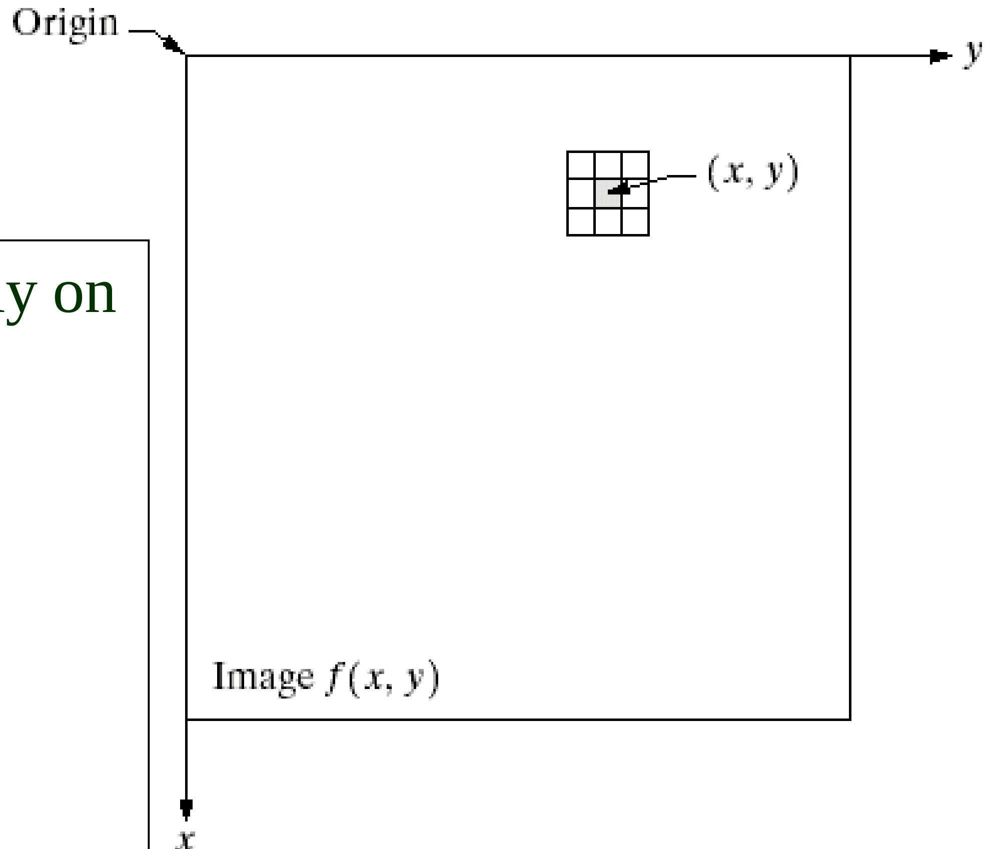
Background Infomation

- Spatial Domain : Techniques are based on direct manipulation of pixels in an image
- Frequency Domain : Techniques are based on modifying the Fourier transform of an image
- Good Images
  - For human visual
    - ✓ The visual evaluation of image quality is a highly subjective process.
    - ✓ It is hard to standardize the definition of a good image.
  - For machine perception
    - ✓ The evaluation task is easier.
    - ✓ A good image is one which gives the best machine recognition results.
- A certain amount of trial and error usually is required before a particular image enhancement approach is selected.



Chapter 3
Image Enhancement in the Spatial Domain

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



➢ Procedures that operate directly on pixels.

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

where

- $f(x, y)$ is the input image
- $g(x, y)$ is the processed image
- T is an operator on f defined over some neighborhood of (x, y)

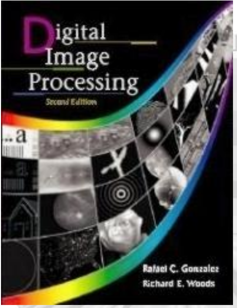
Point Processing

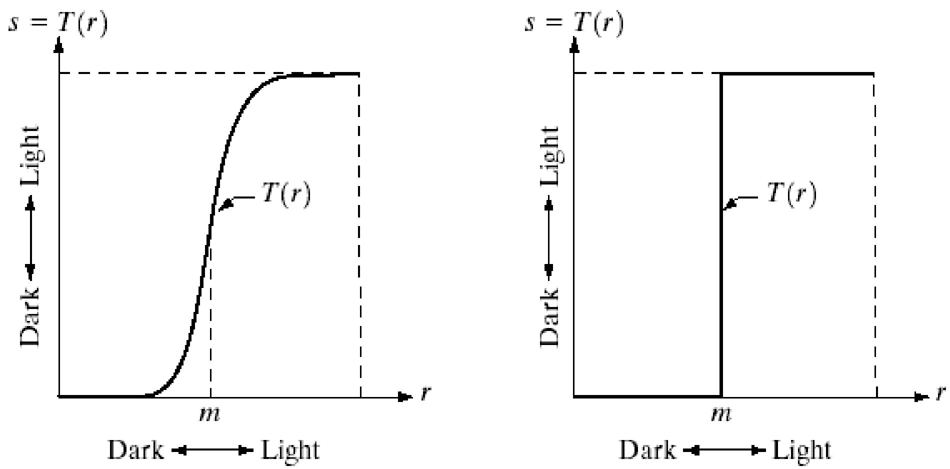
- **Neighborhood = 1x1 pixel**
- g **=depends on only the value of** f **at** (x, y)
- T **= gray level (or intensity or mapping) transformation function**

$$s = T(r)$$

- **Where**
  - r **=gray level of** $f(x, y)$
  - s**=gray level of** $g(x, y)$

Contrast Stretching and Thresholding





a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

- ✓ (a) Produce higher contrast than the original by
  - ✓ darkening the levels below m in the original image
  - ✓ Brightening the levels above m in the orginal image
- ✓ (b) Produce a two-level (binary) image

3 basic gray-level transformation functions

- Linear function
  - Negative and identity transformation
- Logarithm function
  - Log and inverse-log transformation
- Power-law function
  - n^{th} power and n^{th} root transformation

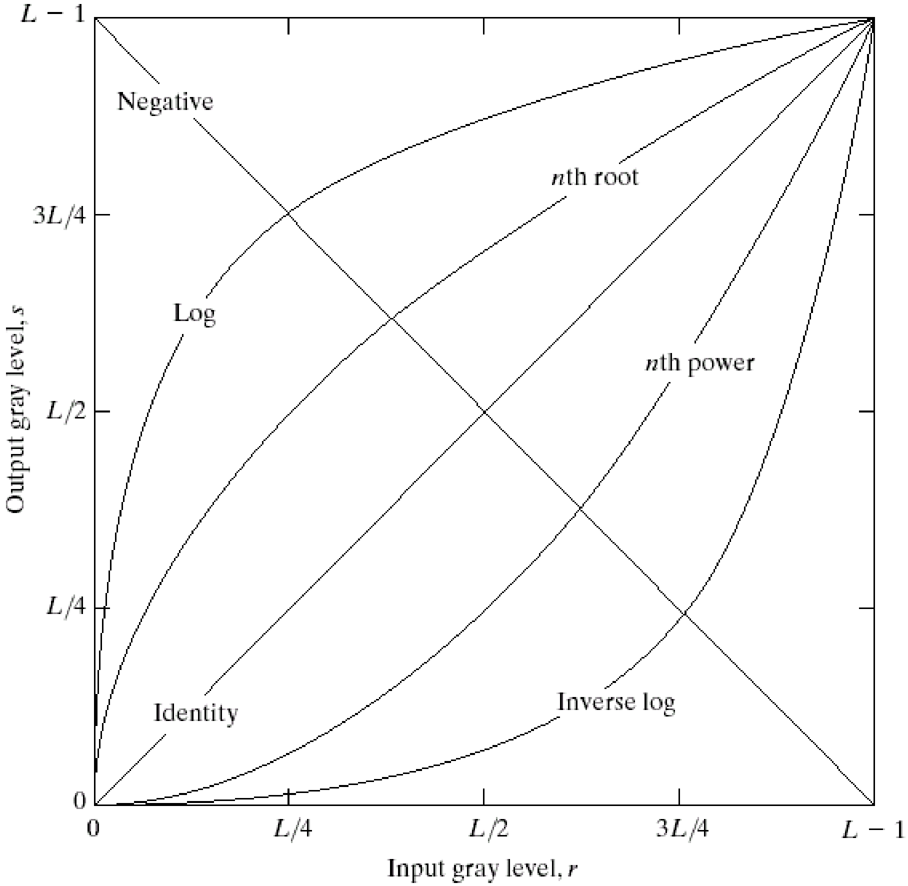
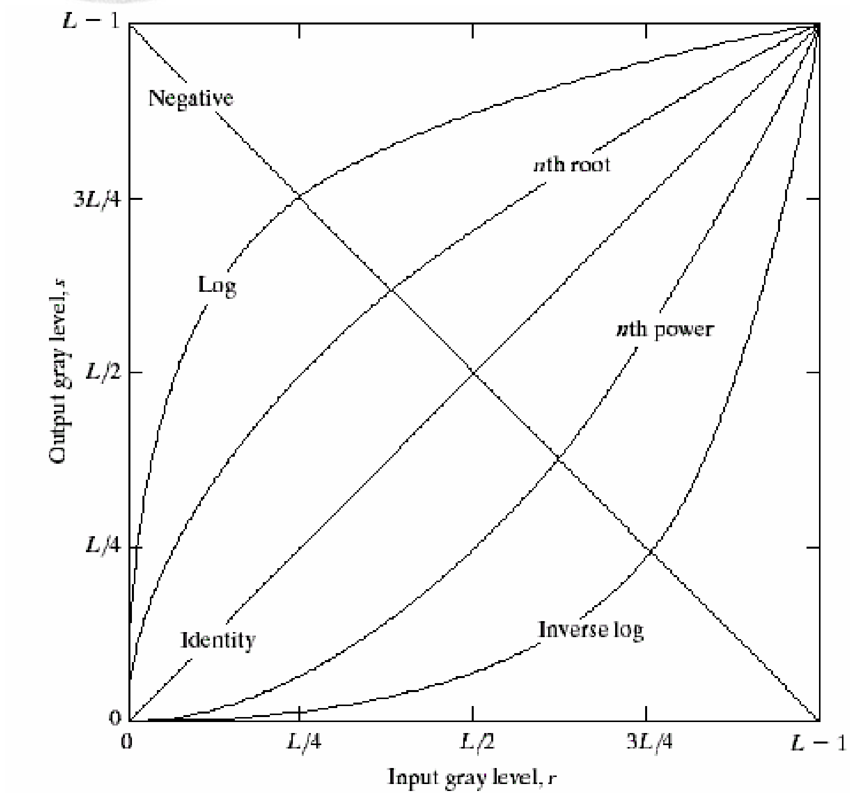


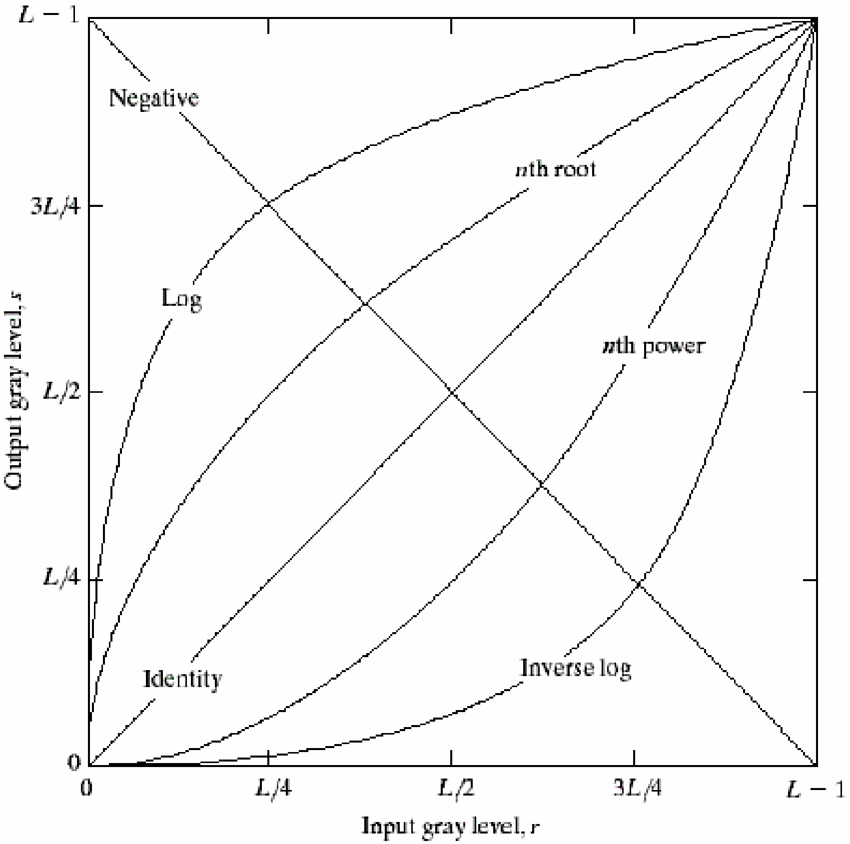
FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

Identity Function



- **Output intensities are identical to input intensities.**
- **Is included in the graph only for completeness.**

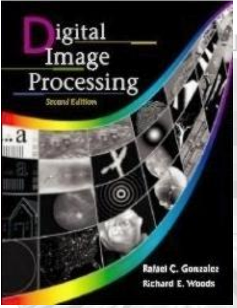
Image Negatives



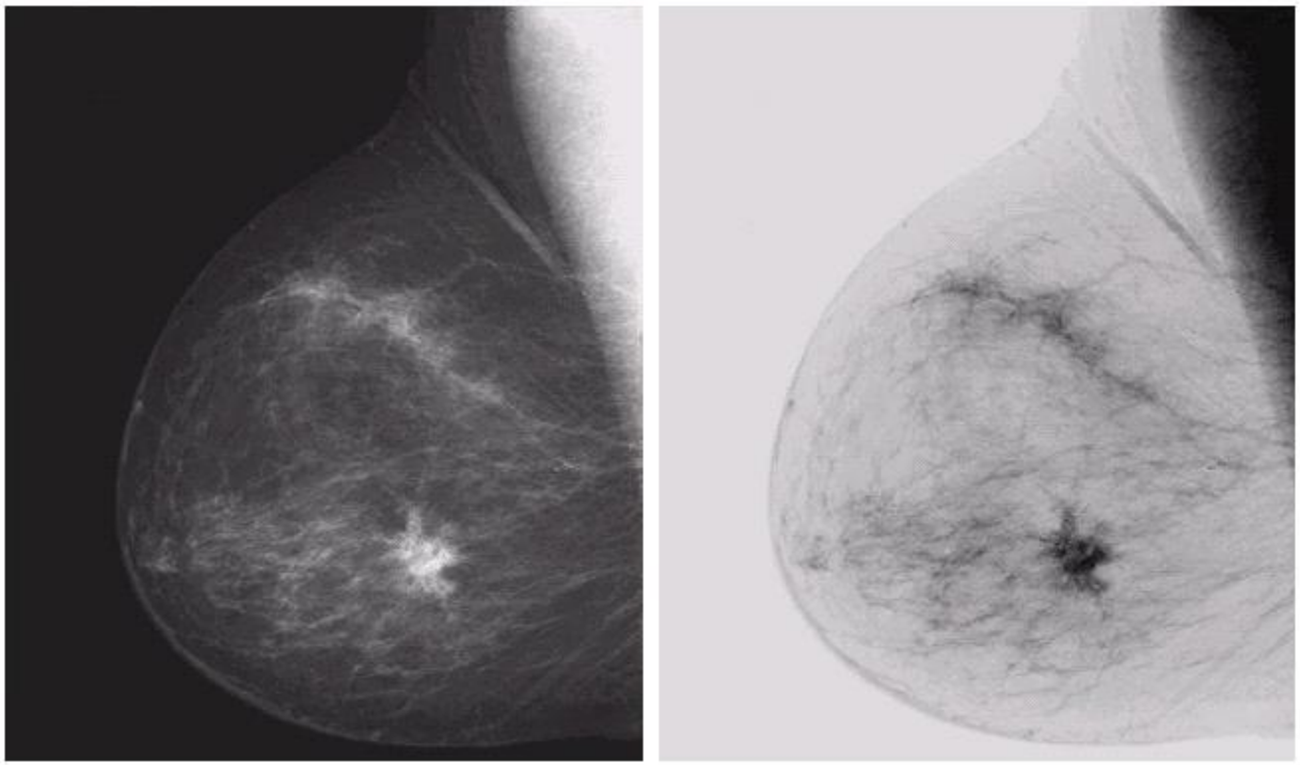
- An image with gray level in the range $[0, L-1]$ where $L = 2^n$; n=1,2,…
- Negative transformation :

$$s = L - 1 - r$$

- Reversing the intensity levels of an image.
- Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area dominant in size.

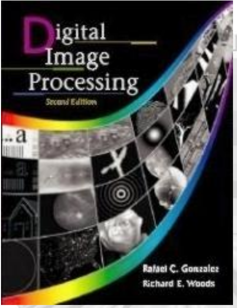


Example of Negative Image

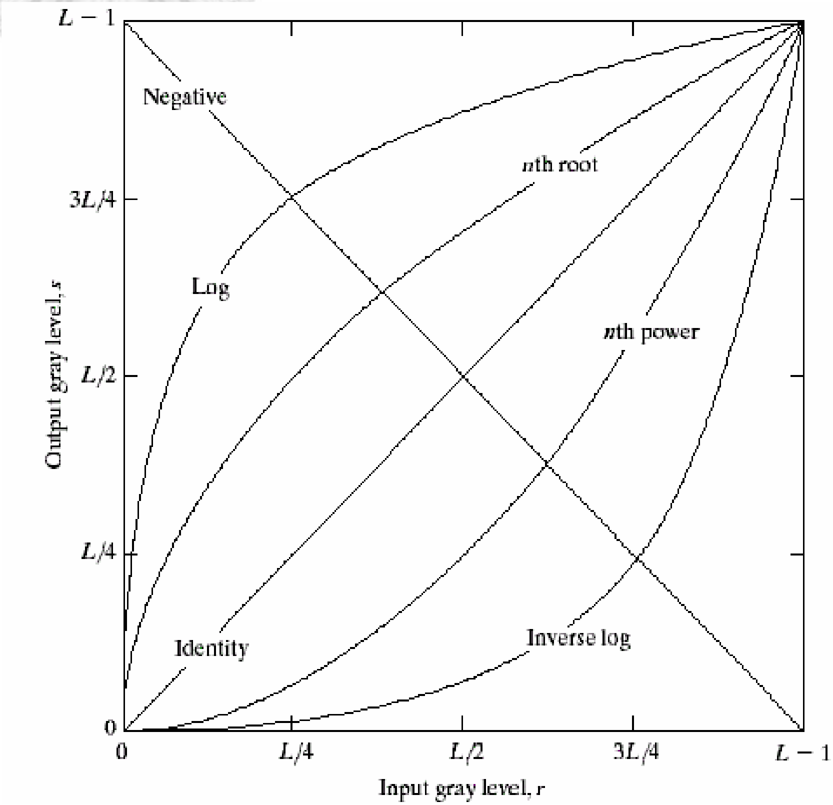


a b

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

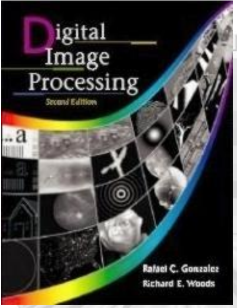


Log Transformations



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

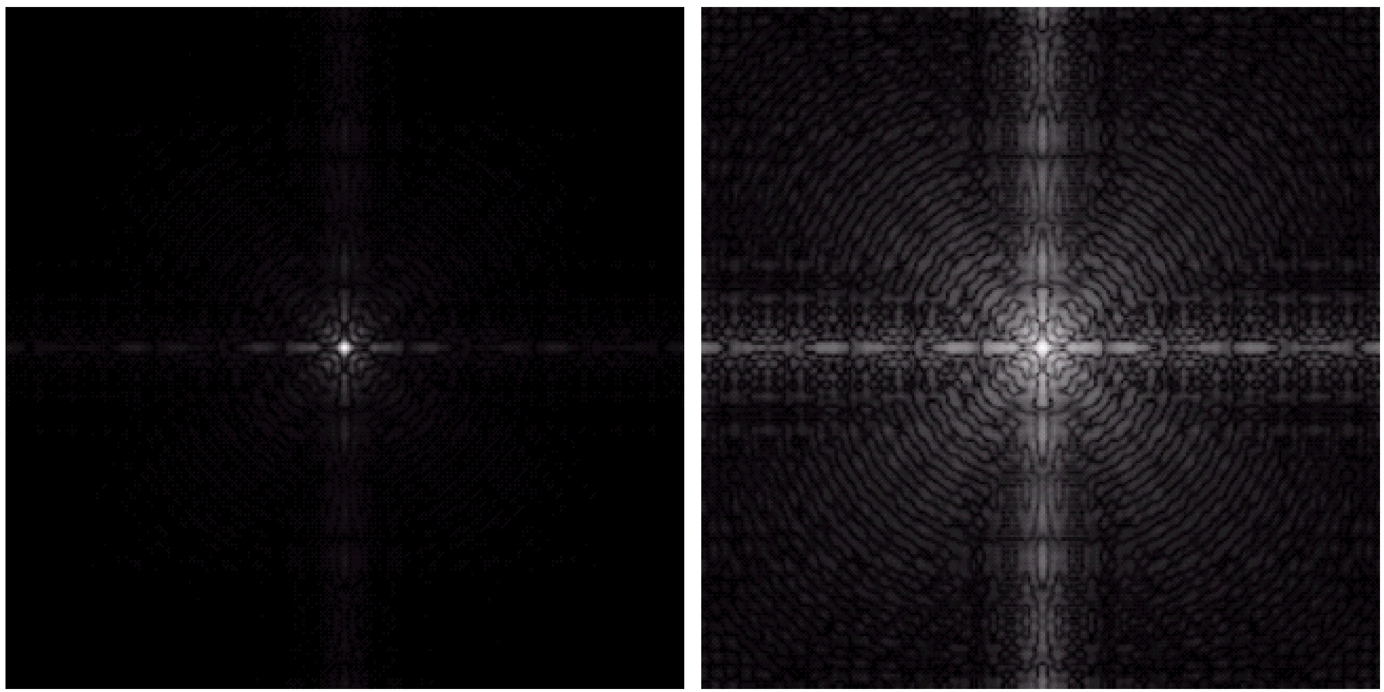
- **c is a contrast and r≥0**
- **Log curve maps a narrow range of low gray-level values in the input image into a wider range of output levels.**
- **Used to expand the values of dark pixels in an image while compressing the higher-level values.**



Examples of Logarithm Image

a b

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



Power-Law Transformations

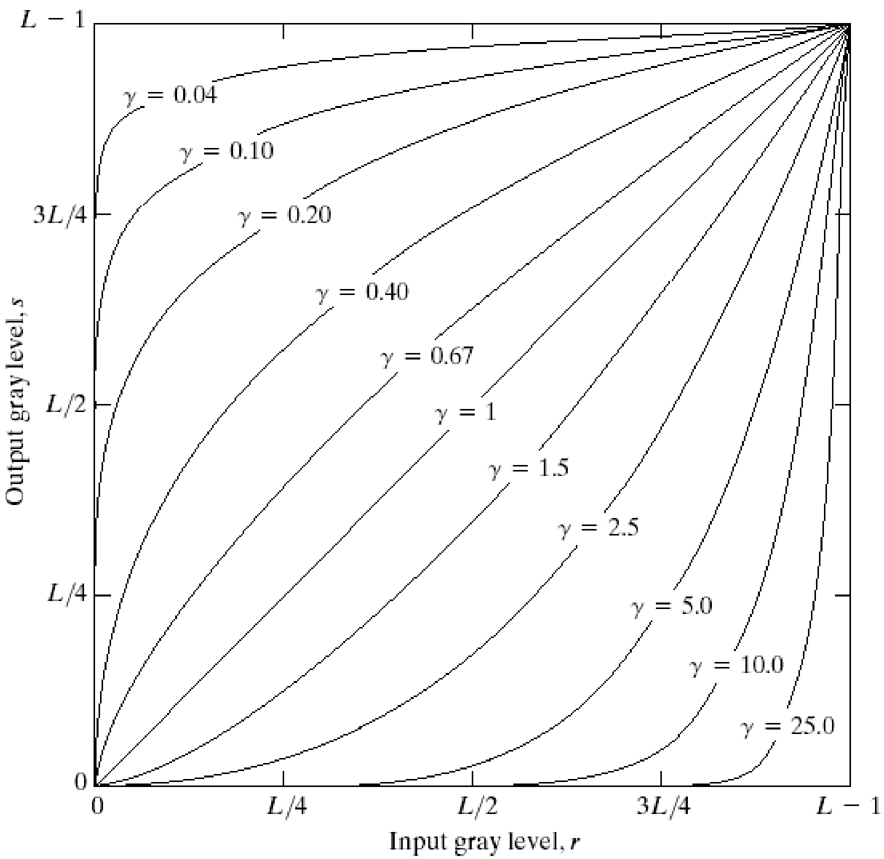
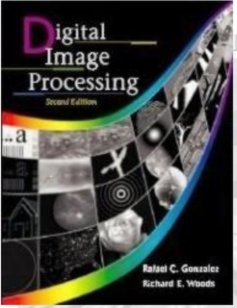


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

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

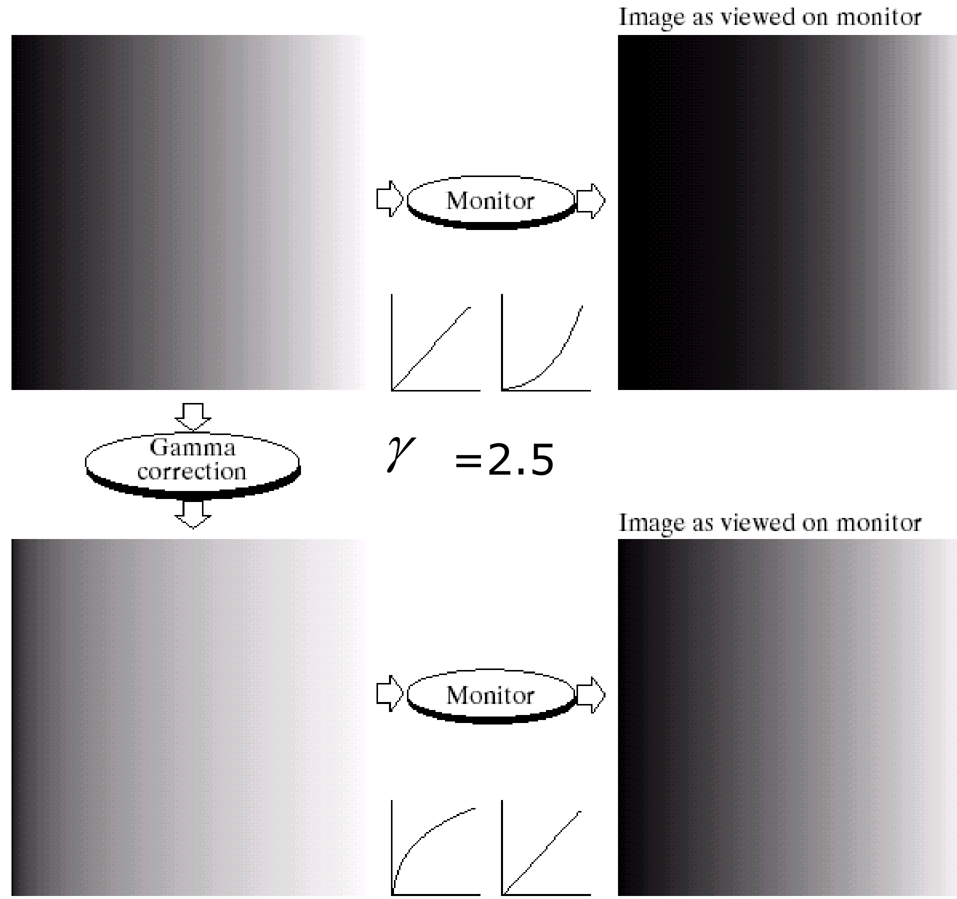
- c and γ are positive constants
- Power-low curves with fractional values of γ map a narrow range of dark input values with the opposite being true for higher values of input levels.
- c = γ =1 → Identity function



Gamma correction

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.

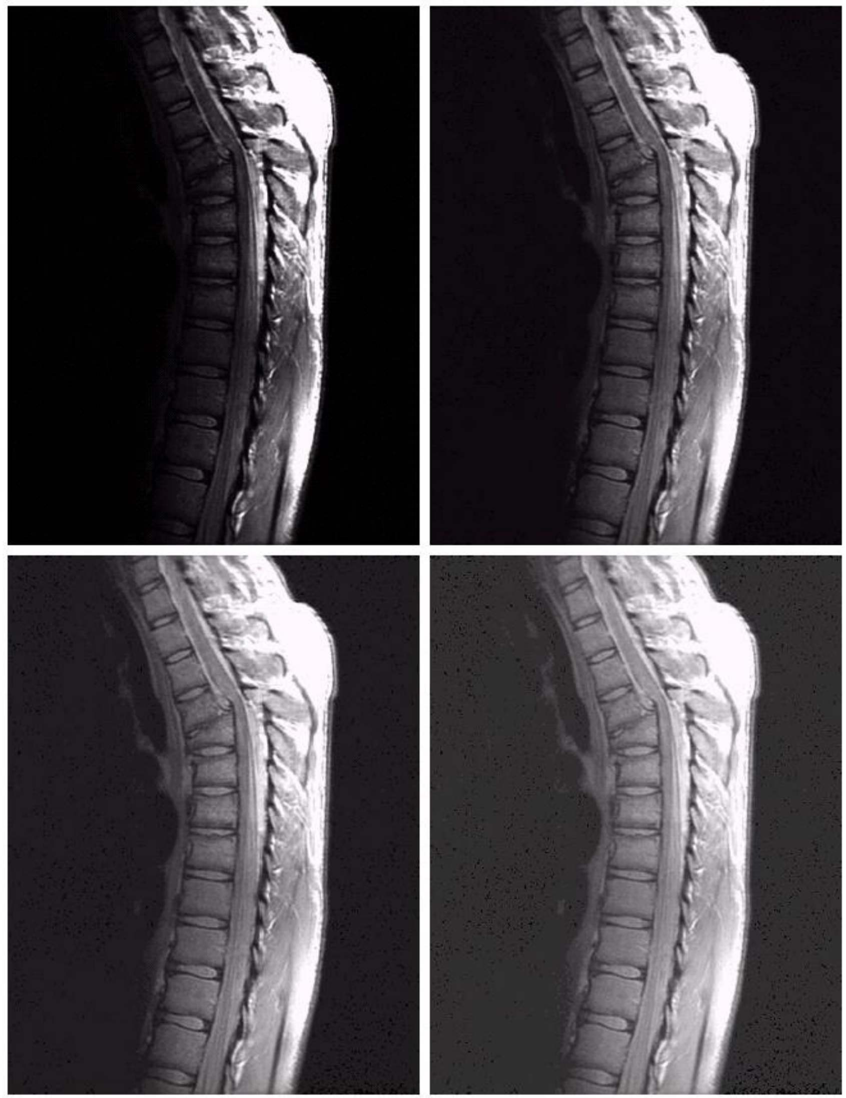


γ =2.5

γ =1/2.5=4

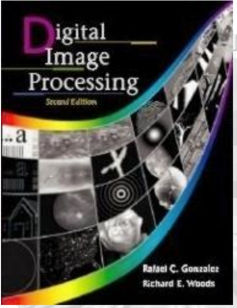
✓ Gamma correction is done by preprocessing the image before inputting it to the monitor with $s = c \times r^{1/\gamma}$

Another example : MRI



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.)



Another example

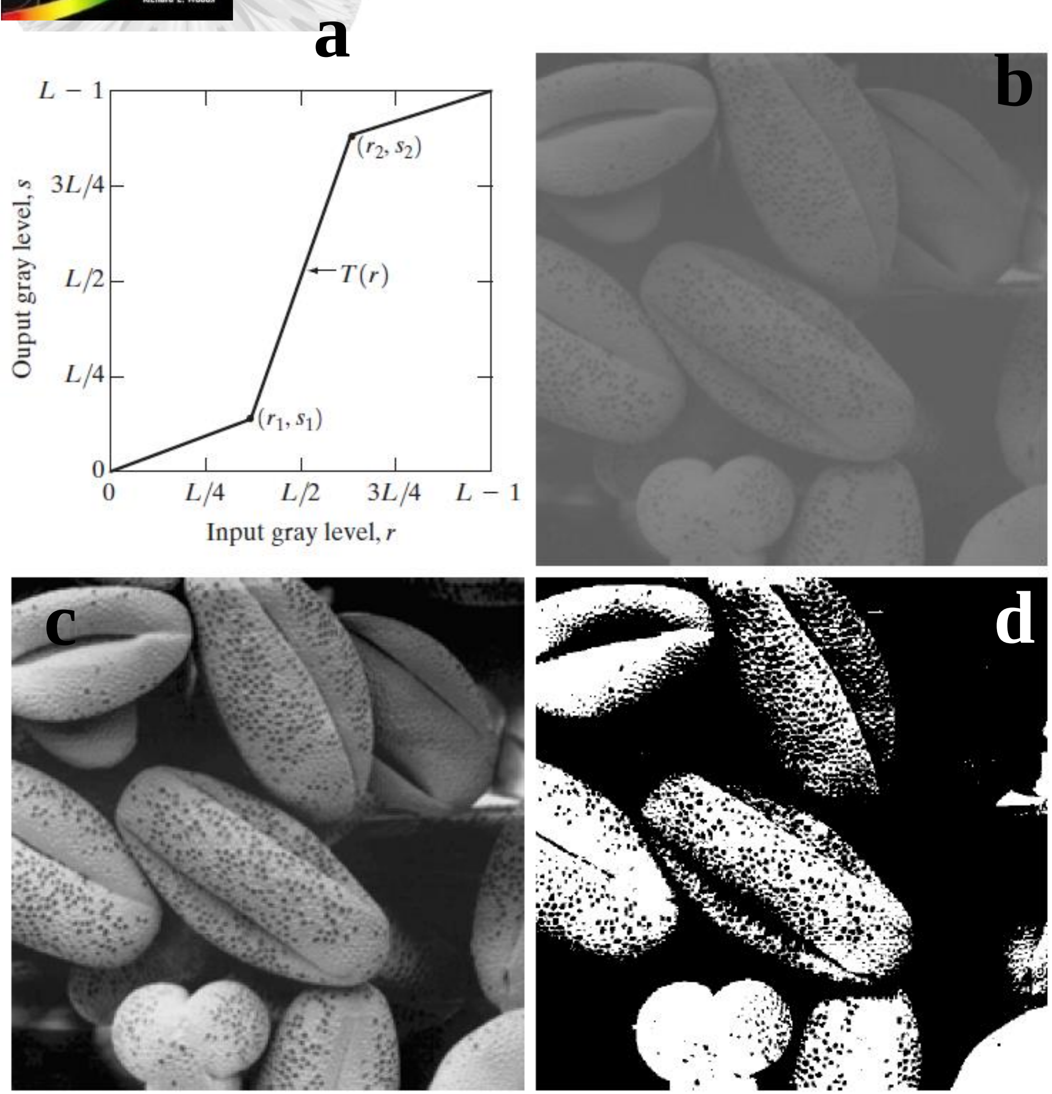
a b
c d

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

Piecewise-Linear Transformation Functions

- **Advantage :**
  - **The form of piecewise function can be arbitrarily complex**
- **Disadvantage:**
  - **Their specification requires considerably more user input**

Contrast Stretching

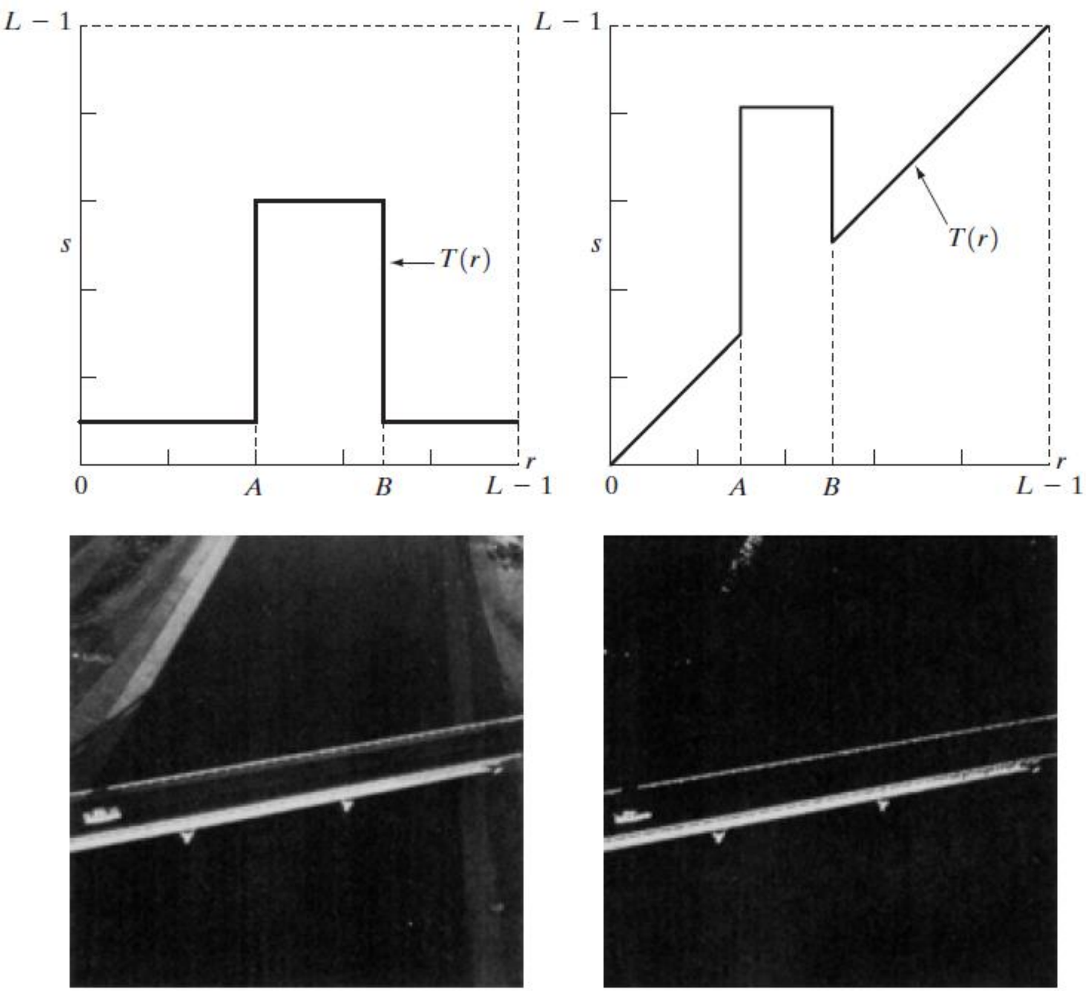


- ✓ Increase the dynamic range of the gray levels in the image
- ✓ (b) A low-contrast image: result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition
- ✓ (c) Result of contrast stretching :

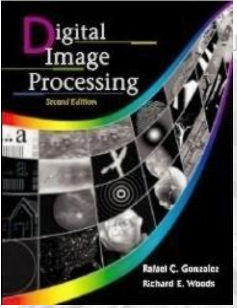
$$(r_1, s_1) = (r_{\min}, 0) \quad \text{and} \quad (r_2, s_2) = (r_{\max}, L-1)$$

- ✓ (d) Result of thresholding

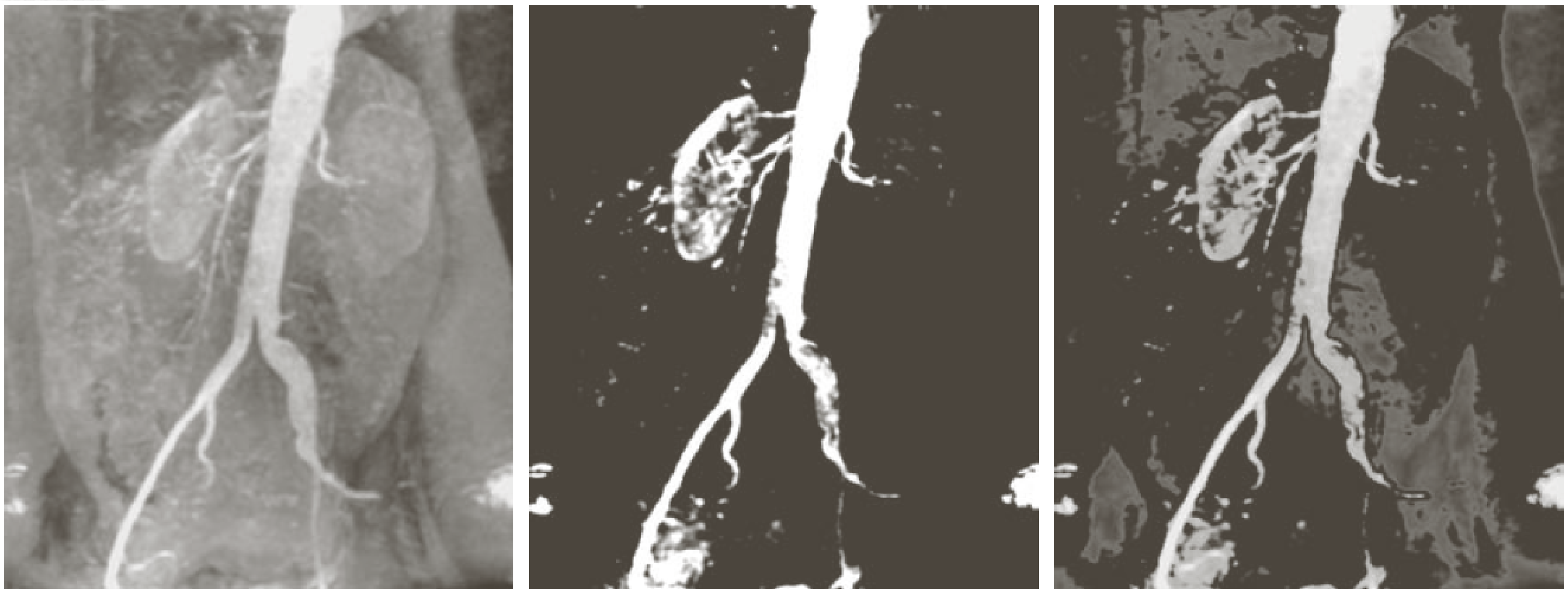
Gray-level slicing



- ✓ Highlighting a specific range of gray levels in an image
  - ✓ Display a high value of all gray levels in the range of interest and a low value of all other gray levels
- ✓ (a) transformation highlights range [A,B] of gray level and reduces all others to a contrast level
- ✓ (b) transformation highlights range [A,B] but preserves all other levels



Gray-level slicing



a b c

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

Bit-plane slicing

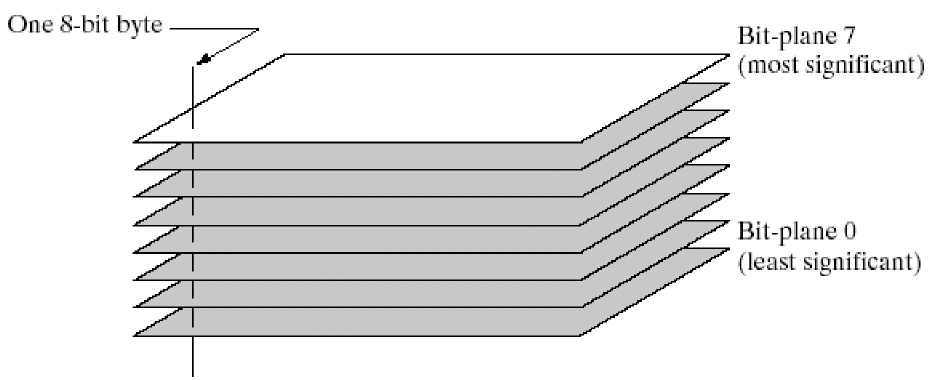


FIGURE 3.12 Bit-plane representation of an 8-bit image.

- Highlighting the contribution made to total image appearance by specific bits
- Suppose each pixel is represented by 8bits
- Higher-order bits contain the majority of the visually significant data
- Useful for analyzing the relative important played by each bit of the image

Example

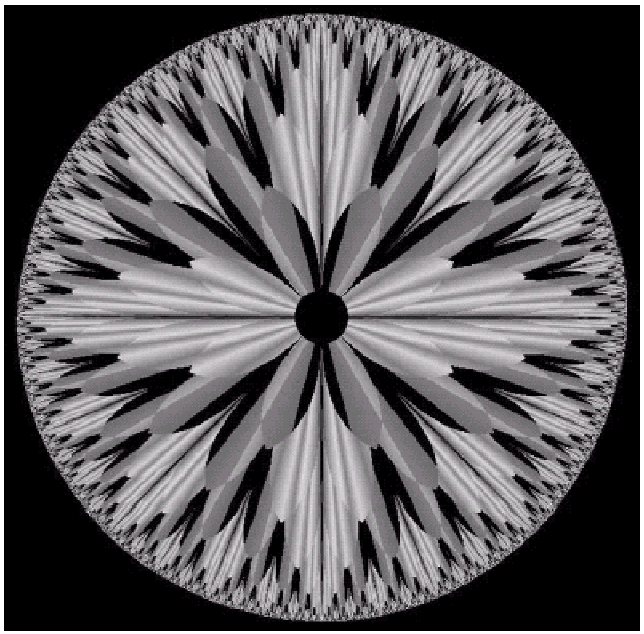
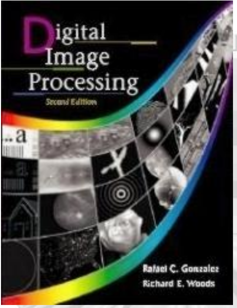


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

✓ The(binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformations.

- ✓ Map all levels between 0 and 127 to 0
- ✓ Map all levels between 129 and 255 to 255



8 bit planes

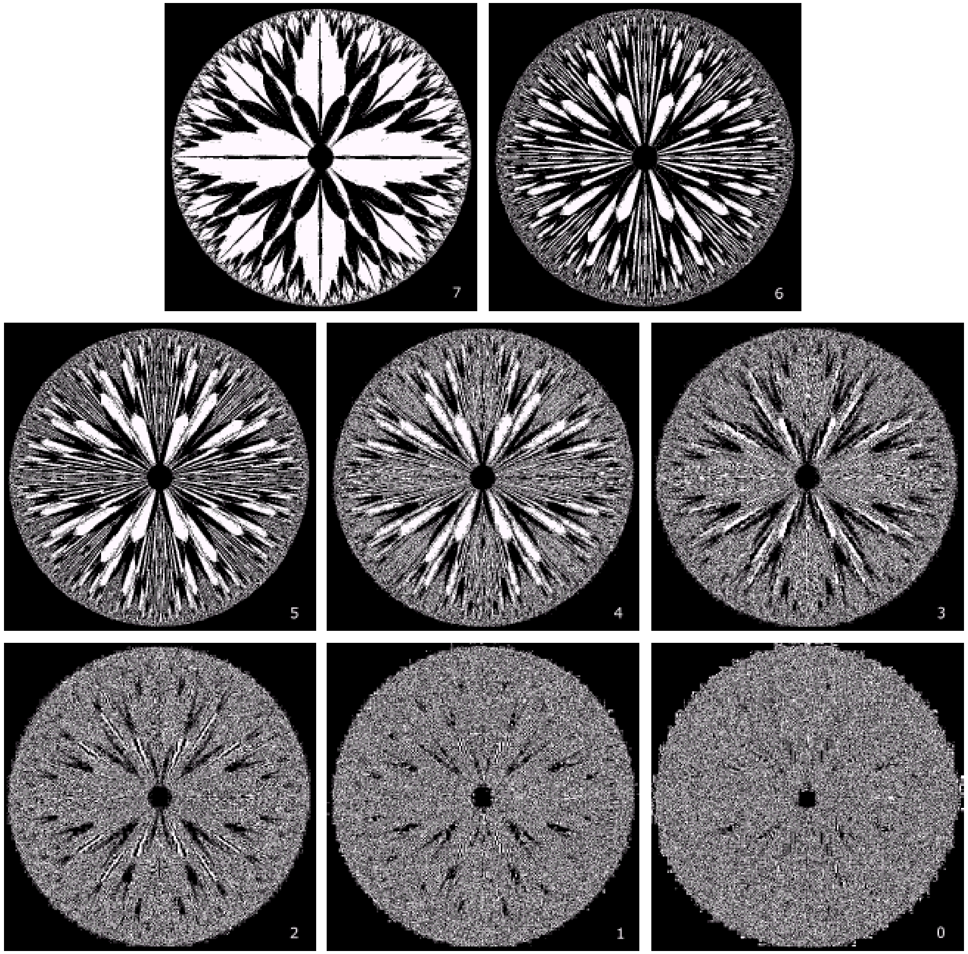


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

<table>
<tr><td colspan="3">Bit-plane 7</td><td colspan="3">Bit-plane 6</td></tr>
<tr><td colspan="2">Bit-plane 5</td><td colspan="2">Bit-plane 4</td><td colspan="2">Bit-plane 3</td></tr>
<tr><td colspan="2">Bit-plane 2</td><td colspan="2">Bit-plane 1</td><td colspan="2">Bit-plane 0</td></tr>
</table>

8 bit planes

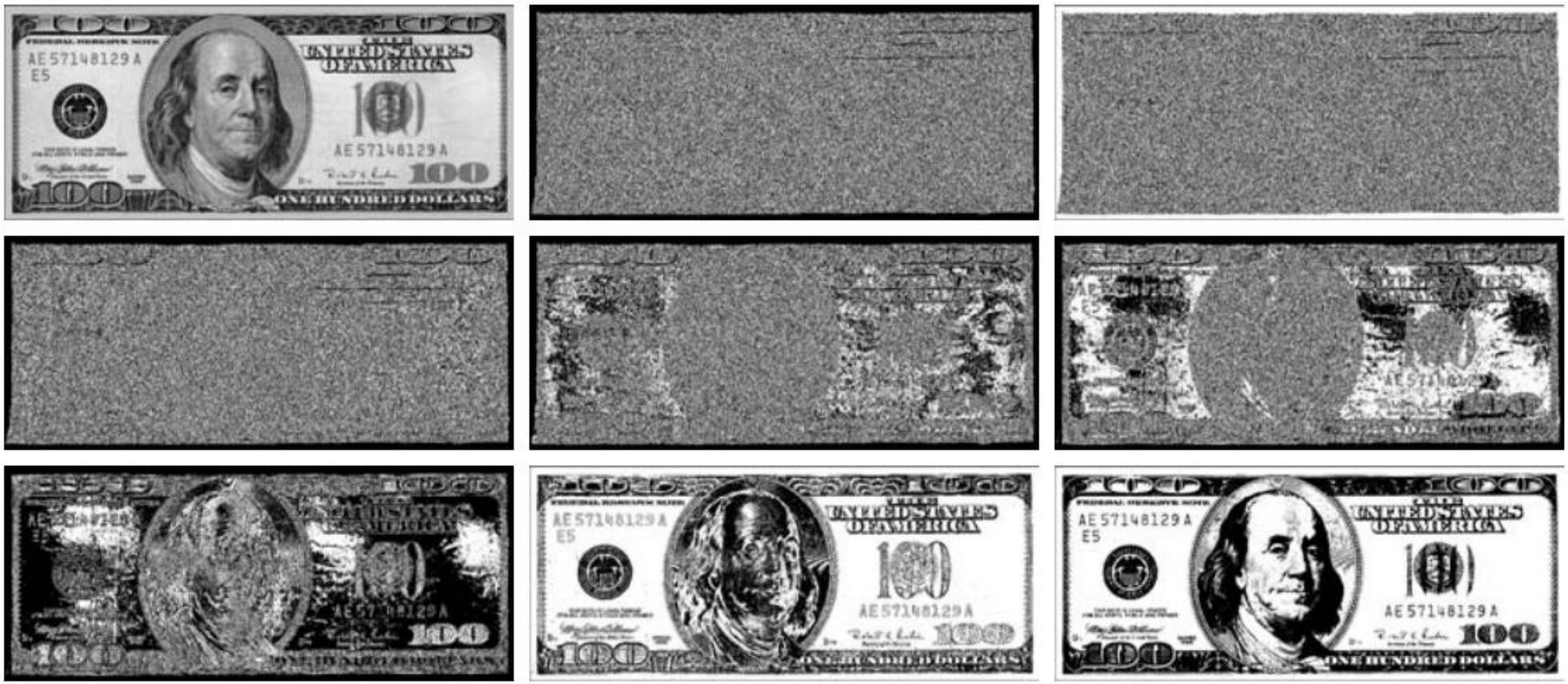
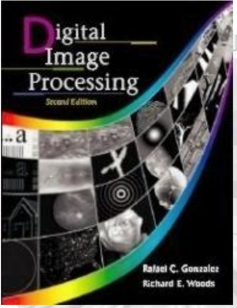


Image Reconstruction from Bit-planes

Original

a b c

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



Histogram Processing

- **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 **: the kth gray level**
  - n_k **: the number of pixels in the image having gray level** r_k
  - $h(r_k)$**: histogram of digital image with gray levels** r_k

Normalized Histogram

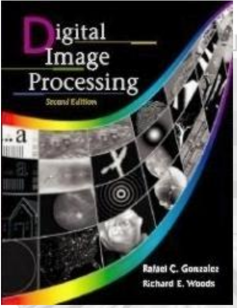
- **dividing each of histogram at gray level r_k by the total number of pixels in the image,** $n = M \times N$

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

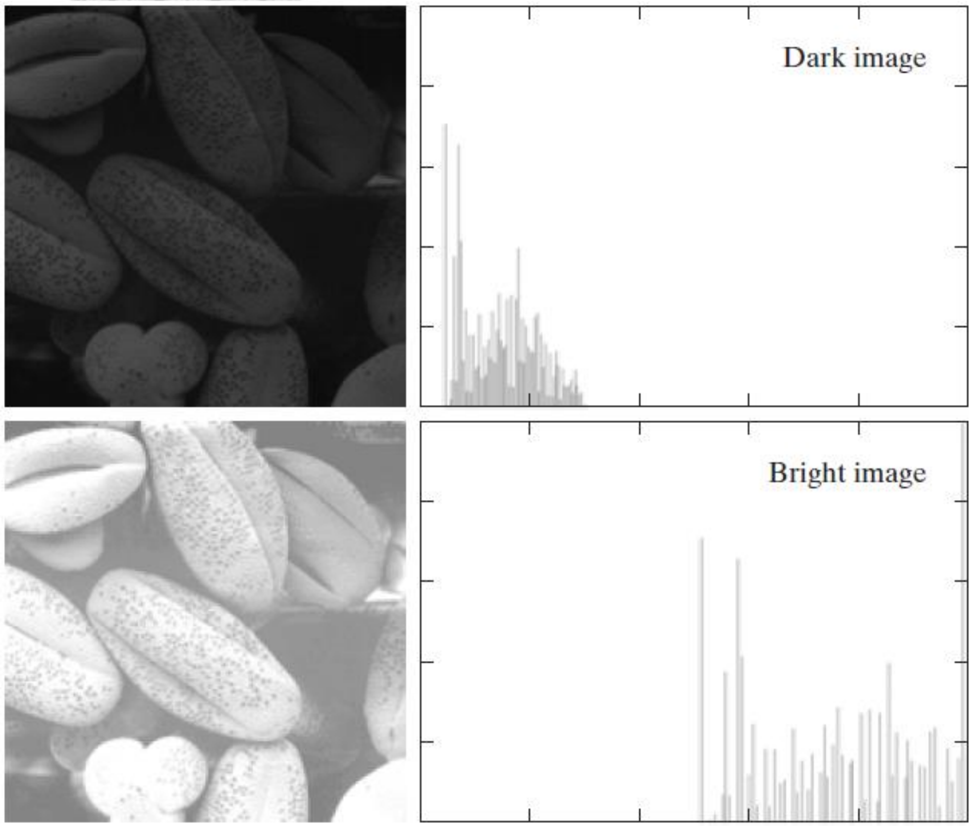
- **For k=0,1,…,L-1**
- $p(r_k)$ **gives an estimate of the probability of occurrence of gray level r_k**
- **The sum of all components of a normalized histogram is equal to 1**

Histogram Processing

- **Basic for numerous spatial domain processing techniques**
- **Used effectively for image enhancement**
- **Information inherent in histograms also is useful in image compression and segmentation**



Example



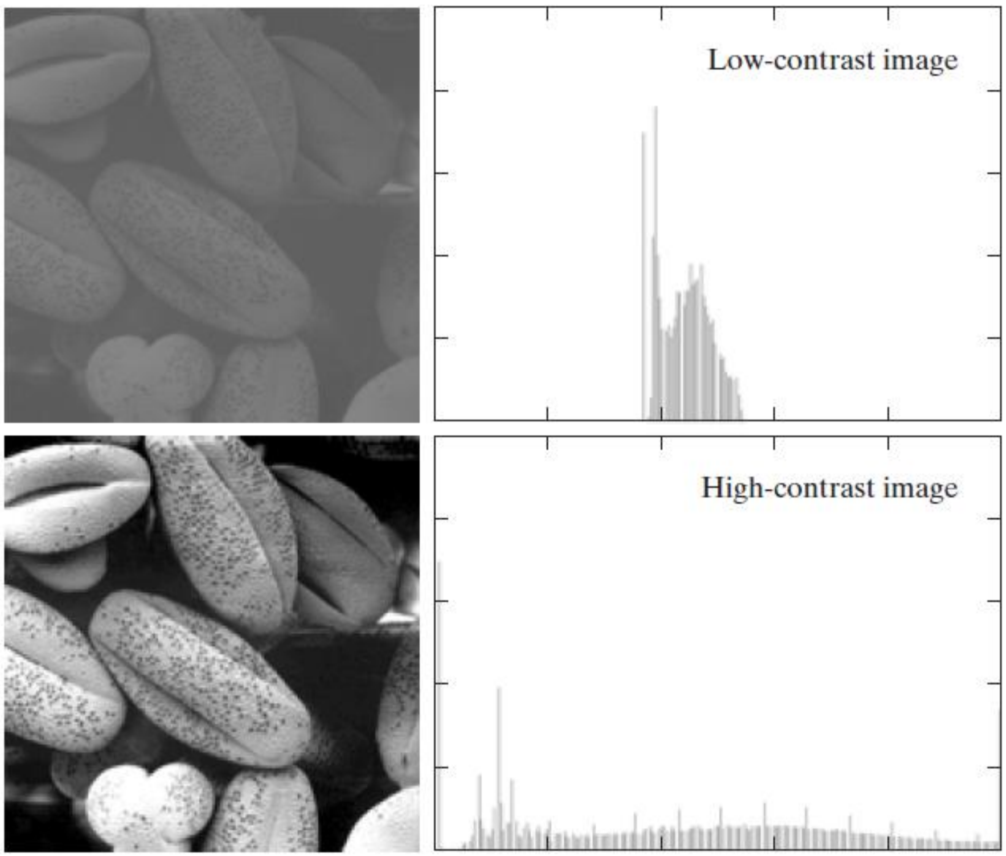
- ✓ Dark image

  components of histogram are concentrated on the low side of the gray scale

- ✓ Bright image

  components of histogram are concentrated on the high side of the gray scale

Example

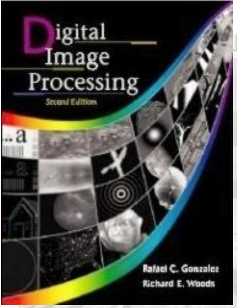


- ✓ Low-contrast image

  histogram is narrow and centered toward the middle of the gray scale

- ✓ High-contrast image

  histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others



Histogram Transformation

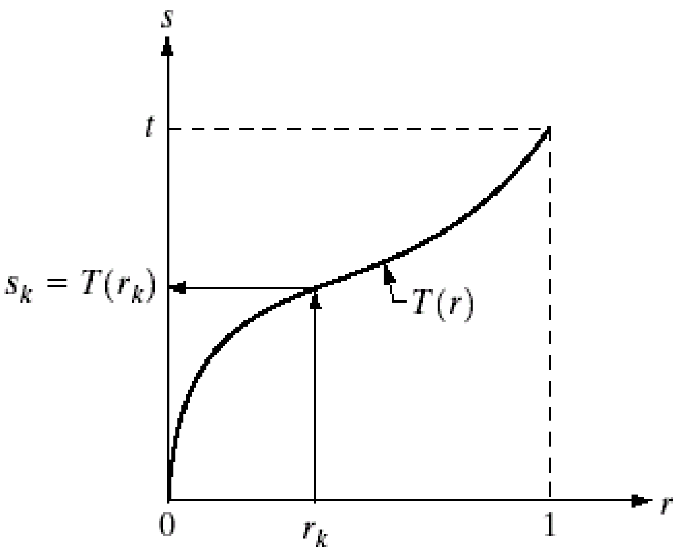


FIGURE 3.16 A gray-level transformation function that is both single valued and monotonically increasing.

$$s = T(r)$$

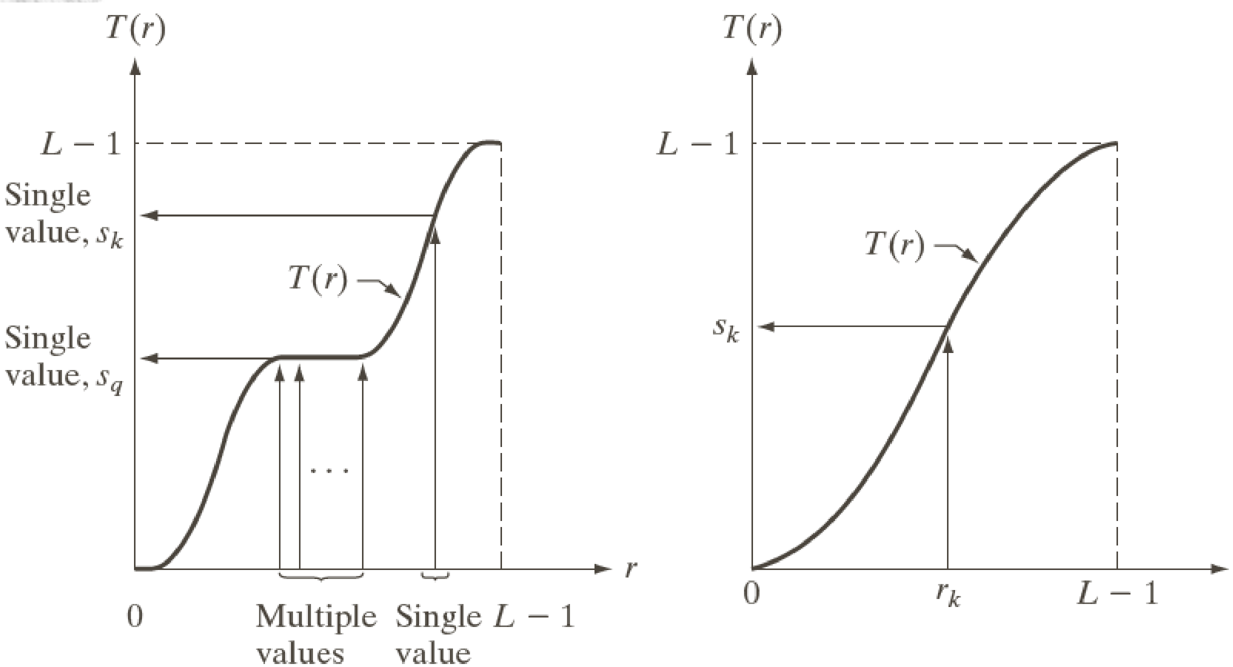
- Where $0 \le r \le 1$
- T(r) satisfies
  - T(r) is single-vlaued and monotonically increasing in the interval $0 \le r \le 1$
  - $0 \le T(r) \le 1$ for $0 \le r \le 1$

Conditions for T(r)

- Single-valued (one-to-one relationship) guarantees that the inverse transformation will exist
- Monotonicity condition preserves that output intensity values will never be less than the corresponding input values, thus preventing artifacts created by reversals.
- $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$ guarantees that the output gray levels will be in the same range as the input levels.
- The inverse transformation from s back to r is

$$r = T^{-1}(s)\ ; \quad 0 \leq s \leq 1$$

Conditions for T(r)



a b

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

†Recall that a function $T(r)$ is *monotonically increasing* if $T(r_2) \geq T(r_1)$ for $r_2 > r_1$. $T(r)$ is a *strictly monotonically increasing* function if $T(r_2) > T(r_1)$ for $r_2 > r_1$. Similar definitions apply to monotonically decreasing functions.

Histogram and Probability Density Function)

- The gray levels in an image may be viewed as random variables in the interval [0,1]
- The normalized histogram may viewed as a Probability Density Function (PDF)

Transformation Function

- A cumulative distribution function (CDF) of random variable r :

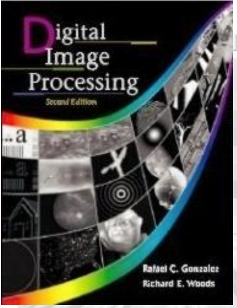
$$s = T(r) = \int_0^r p_r(w)\,dw$$

where w is a dummy variable of integration

- **Note that *T(r)* depends on $p_r(r)$**

Cumulative Distribution Function

- CDF is an integral of a probability function (always positive) is the area under the function
- Thus, CDF is always single valued and monotonically increasing
- Thus, CDF satisfies the condition (a)
- We can use CDF as a transformation function



Using CDF as a Transformation Function

- As $p_s(s)$ is a probability function, it must be zero outside the interval [0,1] in this case because its integral over all values of ***s*** must equal 1.
- $p_s(s)$ called as **a uniform probability a uniform probability density function density function**
- $p_s(s)$ is always a uniform, independent of the form of $p_r(r)$

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

yield

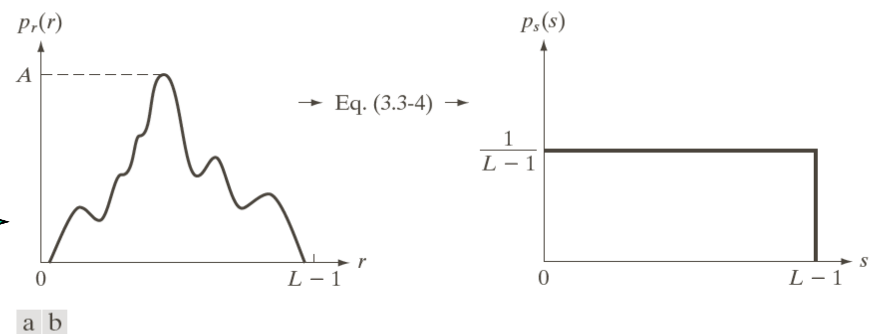


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Discrete Transformation Function

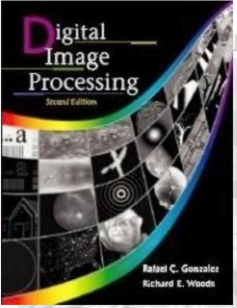
- The probability of occurrence of gray level in an image is approximated by

$$p_r(r_k) = \frac{n_k}{n}$$, where k = 0, 1, …, L-1

- The discrete version of transformation

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

$$= \sum_{j=0}^{k} \frac{n_j}{n}$$, where k = 0, …, L-1



Histogram Equalization

- As the low-contrast image's histogram is narrow and centered toward the middle of the gray scale, if we distribute the histogram to a wider range the quality if the image will be improved.
- We can do it by adjusting the probability density function of the original histogram of the image so that the probability spread equality
- Equalization can be achieved by the following transformation function

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

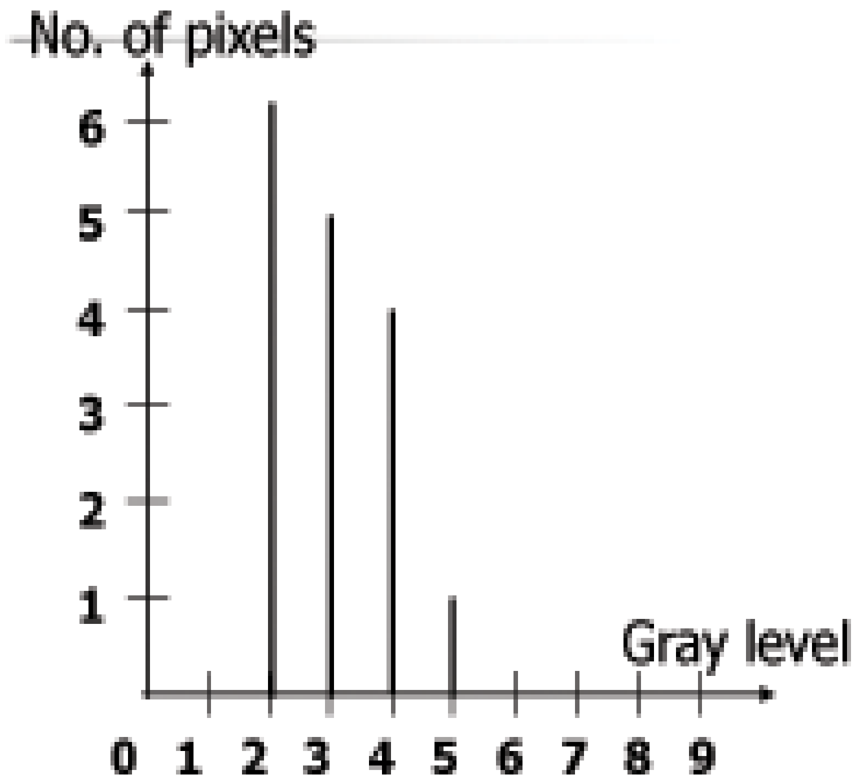
$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, \ldots, L-1$$

Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

Example-Transformation

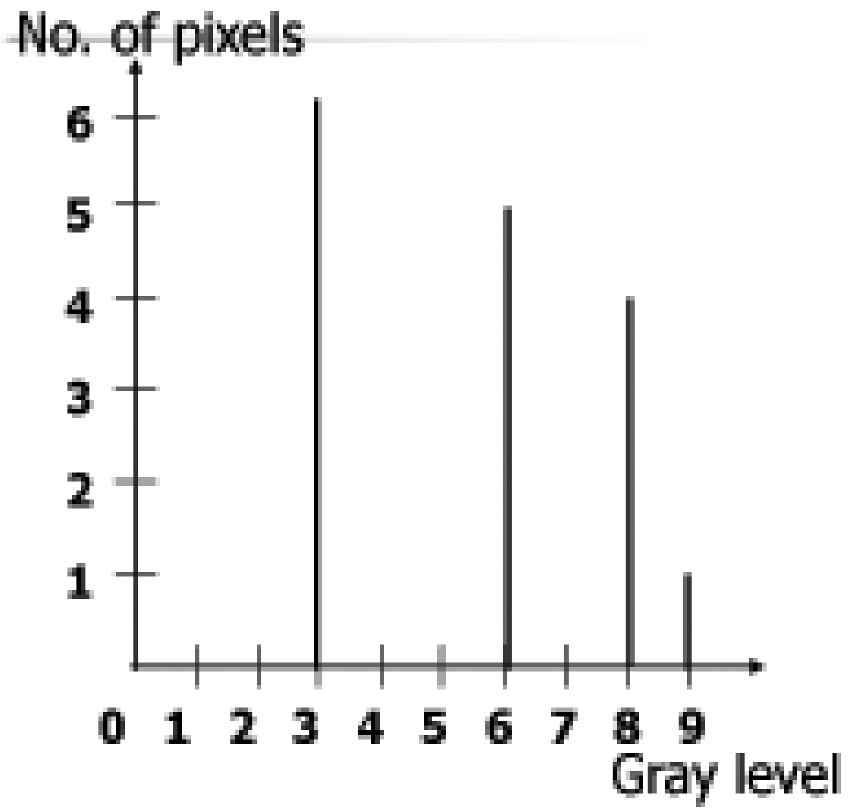
Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
$s \times 9$	0	0	3.3 $\approx$3	6.1 $\approx$6	8.4 $\approx$8	9	9	9	9	9

Example-Result

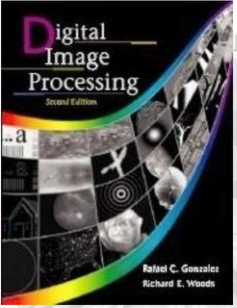
3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



Histogram equalization

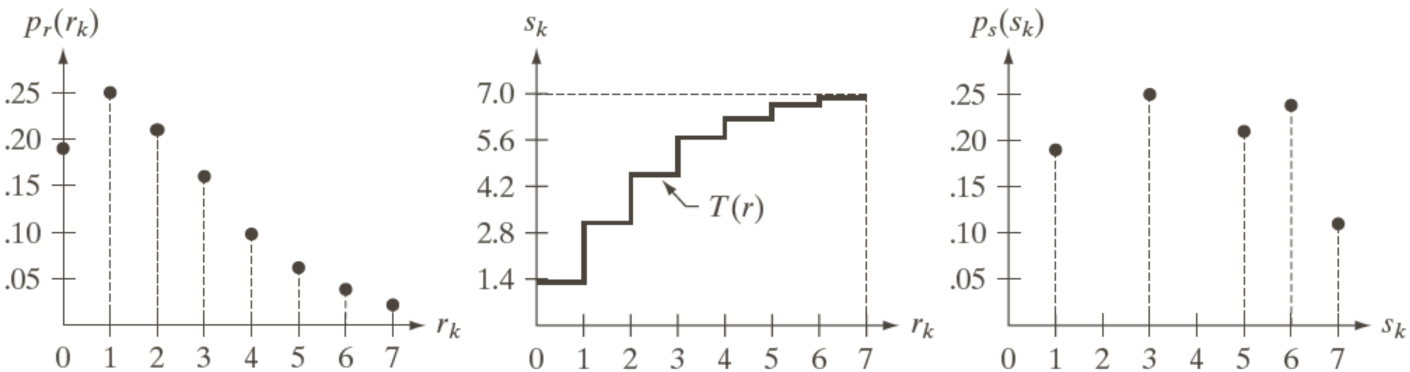


Histogram Equalization

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

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

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



a b c

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

Example

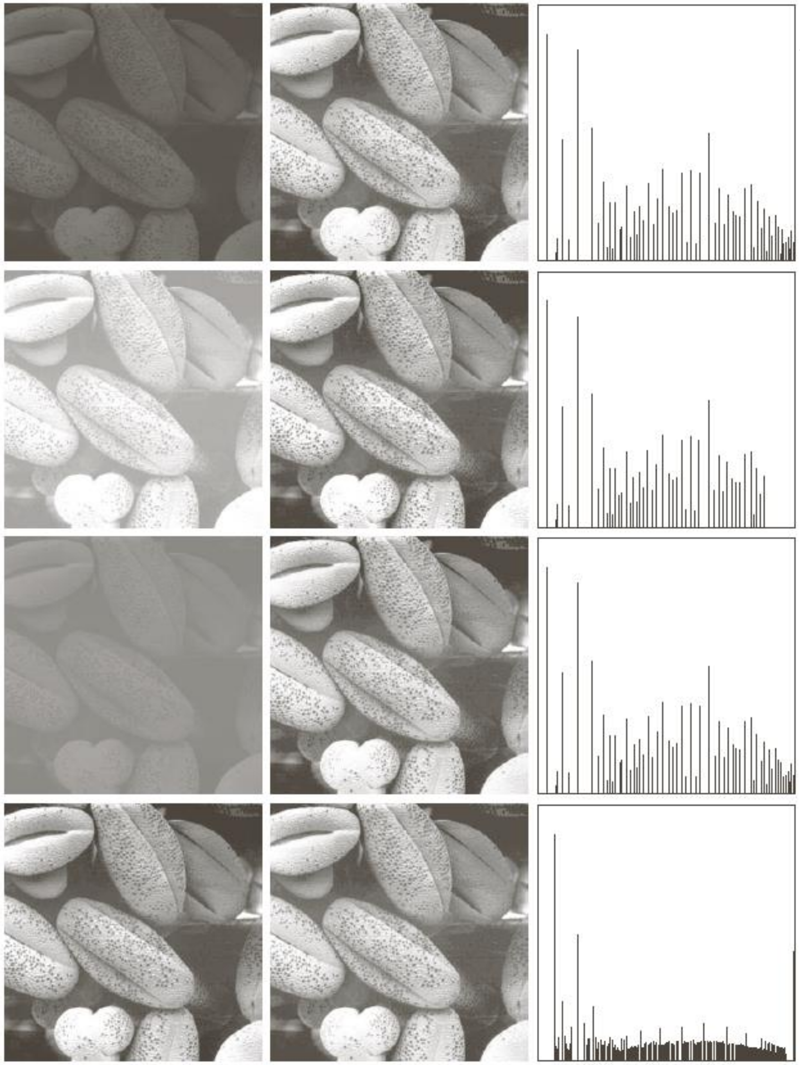
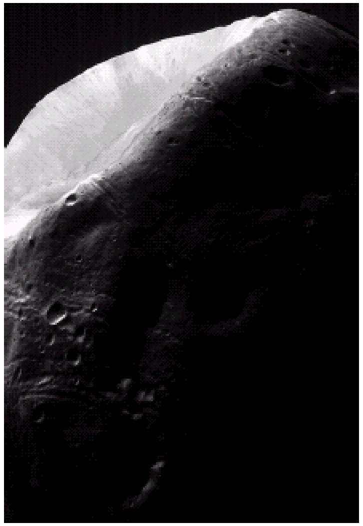
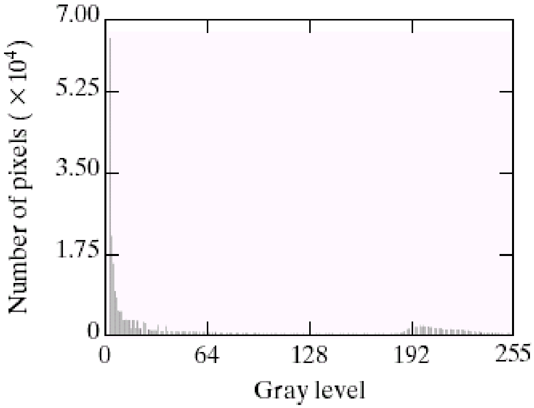


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Example





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.)

✓ Image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in pixels in the dark end of the gray scale

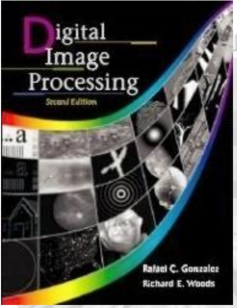


Image Equalization

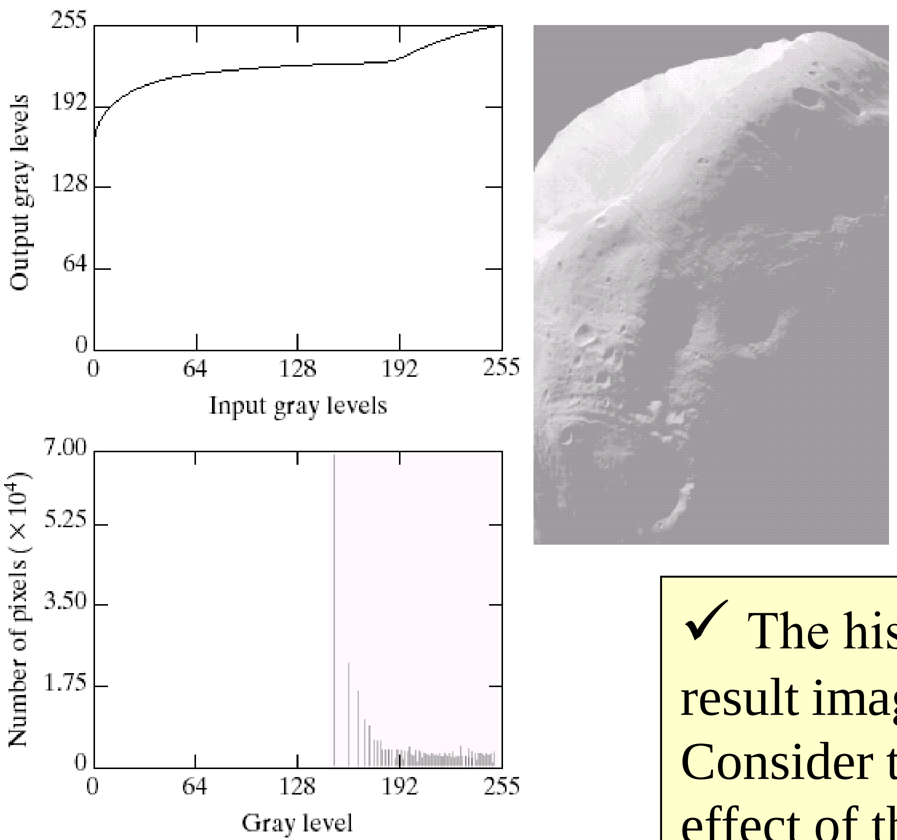
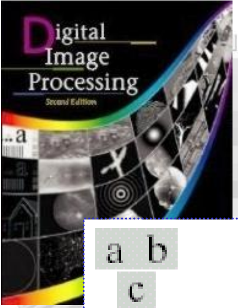


FIGURE 3.21 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).

✓ The histogram equalization doesn't make the result image look better than the original image. Consider the histogram of the result image, the net effect of this method is to map a very narrow interval of dark pixels into the upper end of the gray scale of the input image. As a consequence, the output image is light and has a washed-out appearance.

Histogram Specification (Matching)

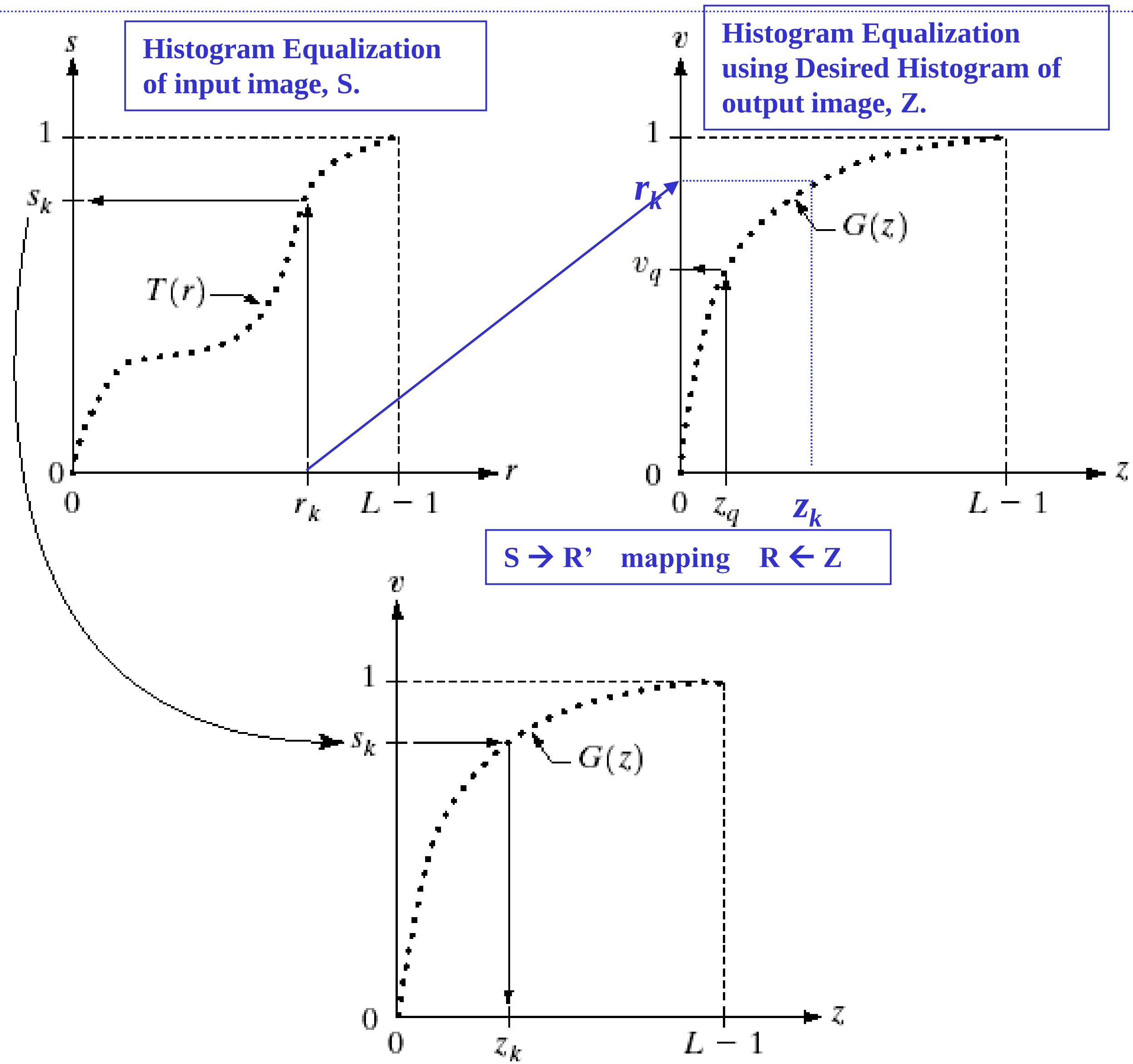
- Histogram equalization has a disadvantage which is that it can generate only one type of output image.
- With Histogram Specification, we can specify the shape of the histogram that we wish the output image to have.
- It doesn’t have to be a uniform histogram

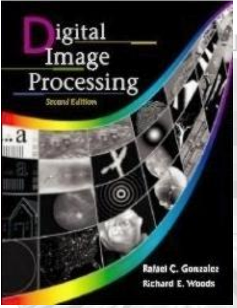


Histogram Specification

a b
c

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





Histogram Specification Example

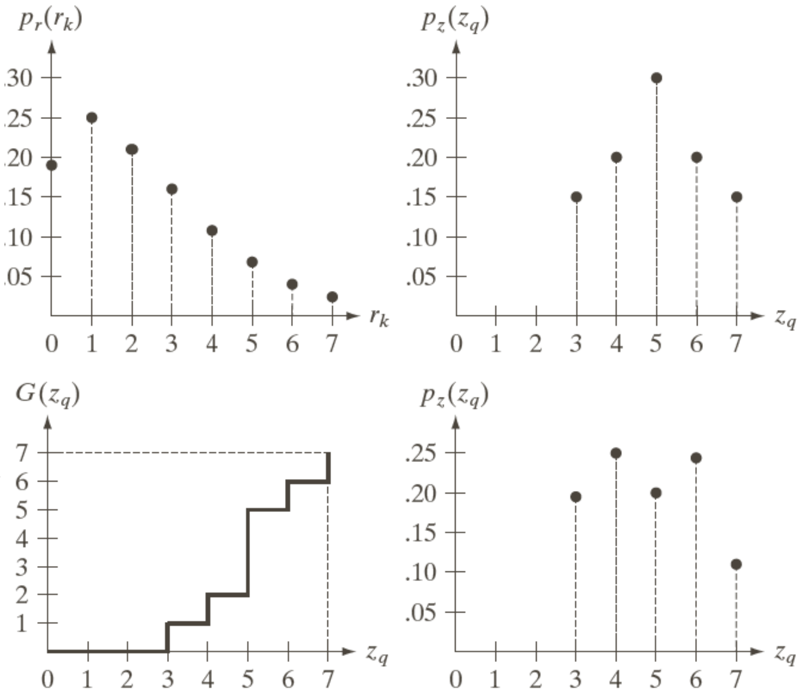
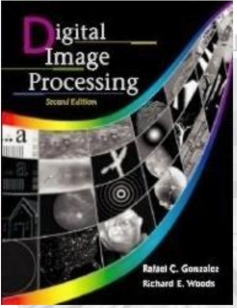


FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

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

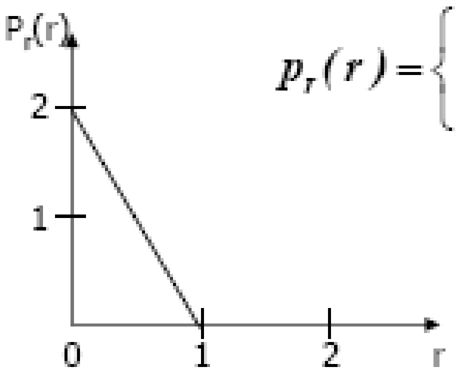
TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).

$$G(z_0) = 0.00 \rightarrow 0$$
$$G(z_1) = 0.00 \rightarrow 0$$
$$G(z_2) = 0.00 \rightarrow 0$$
$$G(z_3) = 1.05 \rightarrow 1$$
$$G(z_4) = 2.45 \rightarrow 2$$
$$G(z_5) = 4.55 \rightarrow 5$$
$$G(z_6) = 5.95 \rightarrow 6$$
$$G(z_7) = 7.00 \rightarrow 7$$



Example (continuous space)

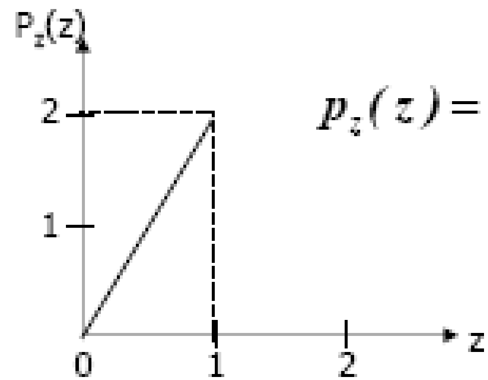
- Transform an input image with the PDF a) to the output image having the PDF b)



$$p_r(r) = \begin{cases} -2r+2 & ;0 \le r \le 1 \\ 0 & ;\text{elsewhere} \end{cases}$$

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

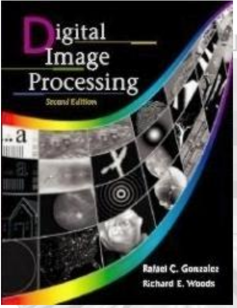
a)



$$p_z(z) = \begin{cases} 2z & ;0 \le z \le 1 \\ 0 & ;\text{elsewhere} \end{cases}$$

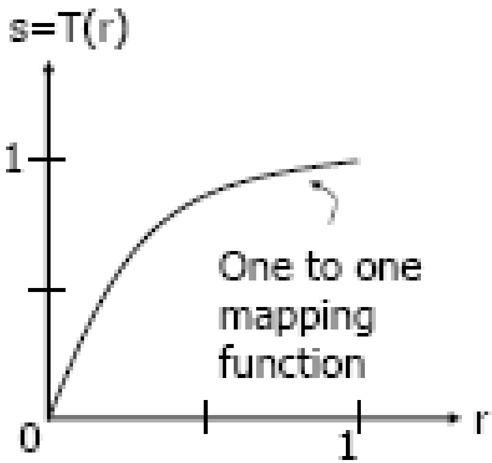
$$\int_0^z p_z(w)dw = 1$$

b)



Example (continuous space)

- Obtain the Transformation function T(r)



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

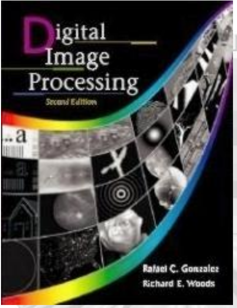
$$= \int_0^r (-2w + 2)\,dw$$

$$= -w^2 + 2w \Big|_0^r$$

$$= -r^2 + 2r$$

- Obtain the transformation function G(z)

$$G(z) = \int_0^z (2w)\,dw \quad = z^2 \Big|_0^z \quad = z^2$$



Example (continuous space)

- Obtain the inverse transform function G^{-1}

$$G(z) = T(r)$$

$$z^2 = -r^2 + 2r$$

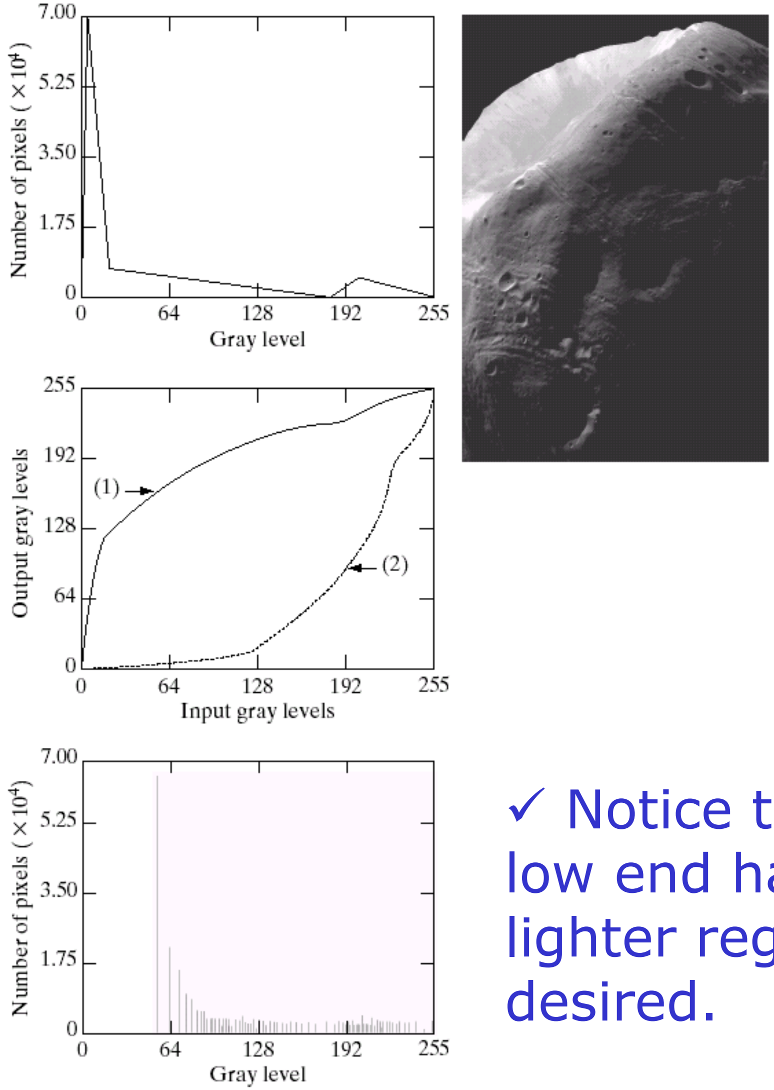
$$z = \sqrt{2r - r^2}$$

- We can guarantee that $0 \le z \le 1$ when $0 \le r \le 0$

Histogram Specification

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) the transformation function G(z) obtained from

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

$$k = 0, 1, 2, \ldots L-1$$

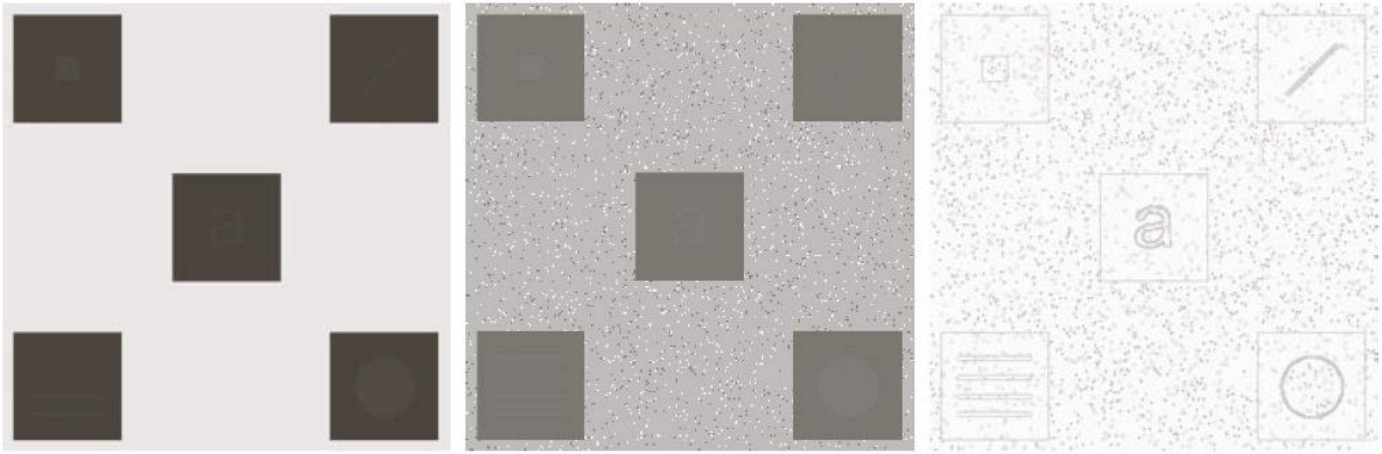
(2) the inverse transformation $G^{-1}(s)$

✓ Notice that the output histogram's low end has shifted right toward the lighter region of the gray scale as desired.

Note

- Histogram specification is a trial-and-error process
- There are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.
- Histograms processing methods are global processing, in the sense that pixels are modified by a transformation function based in the gray-level content of an entire image.
- Sometimes, we may need to enhance details over small areas in an image, which in called a local enhancement.

Local Enhancement



a b c

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

- Define a square or rectangular neighborhood and move the center of this area from pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed and either histogram equalization or histogram specification transformation function is obtained.
- Another approach used to reduce computation is to utilize nonoverlapping regions, but it usually produces an undesirable “blocky” effect.

Chapter 3
Histogram Statistics for Image Enhancement

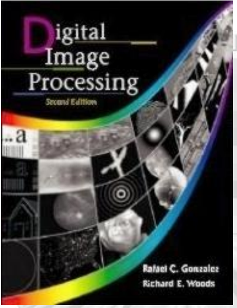
- **Image statistics (*n*-th moment) for enhancement.**

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

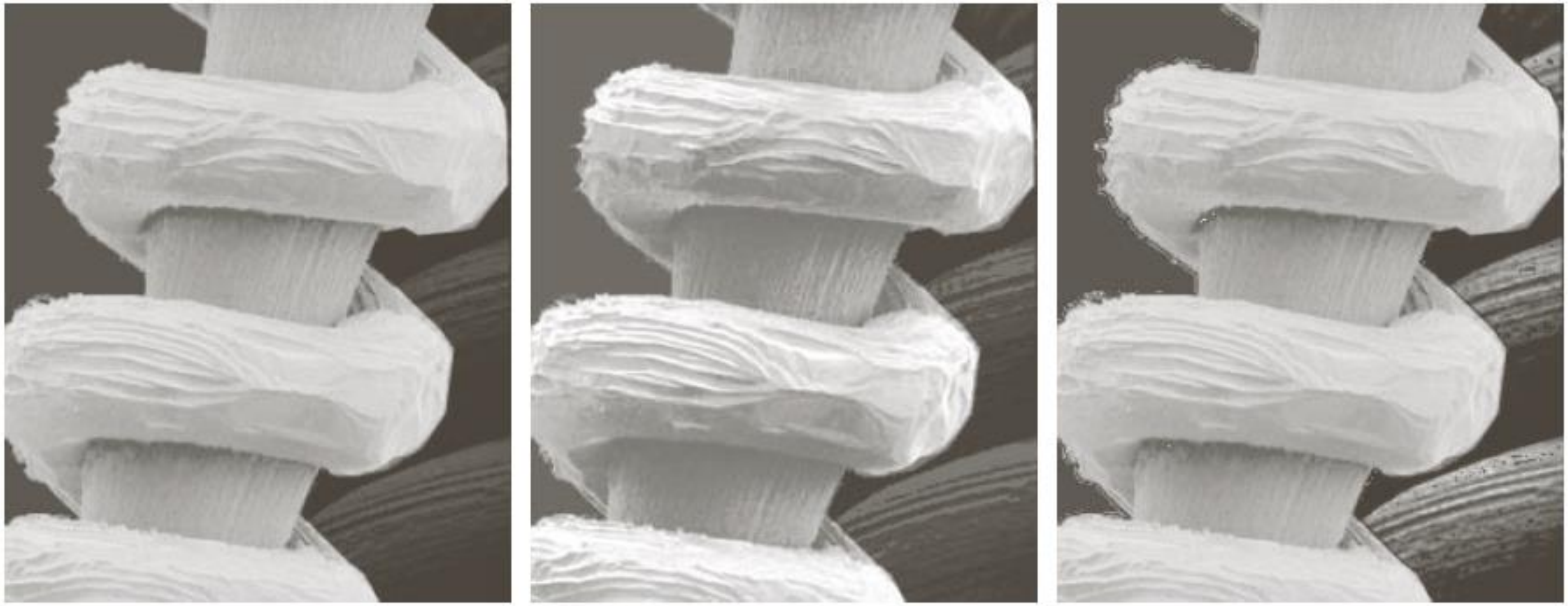
mean **variance**

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

- **Global mean and variance are computed over an entire image and are useful for gross adjustments in overall intensity and contrast.**
- **Variance are used as the basis for making changes that depend on image characteristics in a neighborhood about each pixel in an image**



Chapter 3
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.)

Chapter 3
Histogram Statistics for Image Enhancement

- **The mean and variance values of the pixels in this neighborhood S_{xy} is given by the expression**

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

- **Enhancement performed by**

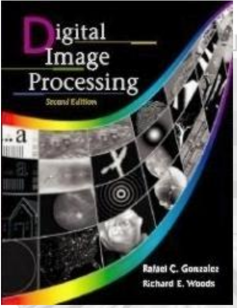
$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{if } m_{S_{xy}} \le k_0 m_G \text{ AND } k_1\sigma_G \le \sigma_{S_{xy}} \le k_2\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

  - **k_0 is positive and less than 1.0**
  - **k_2 is positive (>1.0 for bright areas OR <1.0 for dark areas)**
  - **$k_1 < k_2$ to set a lower limit on the local standard deviation**

Enhancement using Arithmetic/Logic Operations

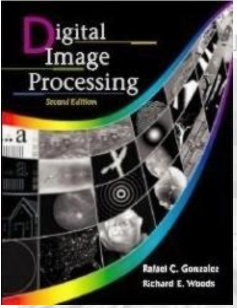
- **Arithmetic/Logic operations perform on pixel by pixel basis between two or more images**

- **except NOT operation which perform only on a single image**

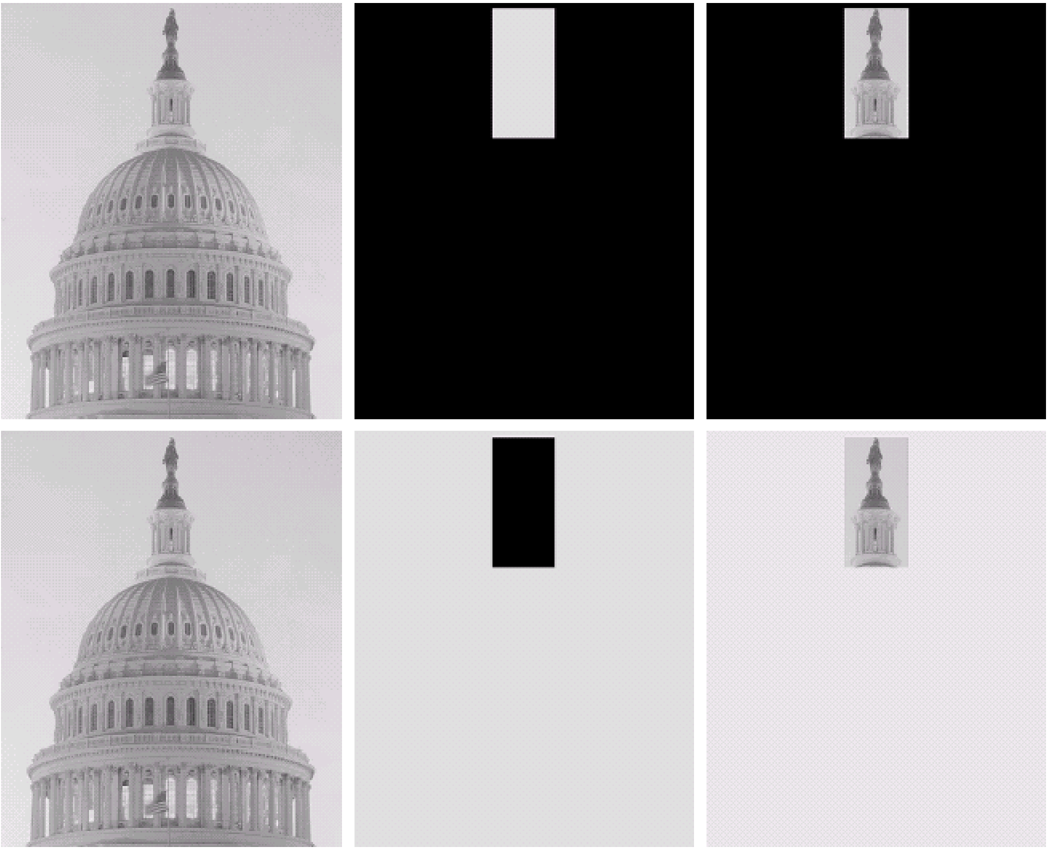


Logic Operations

- **Logic operation performs on gray-level images, the pixel values are processed as binary numbers**
- **light represents a binary 1, and dark represents a binary 0**
- **NOT operation=negative transformation**



Example of AND/OR 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).

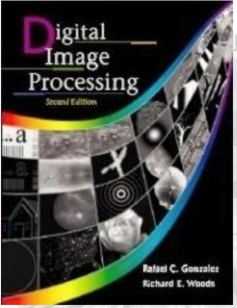
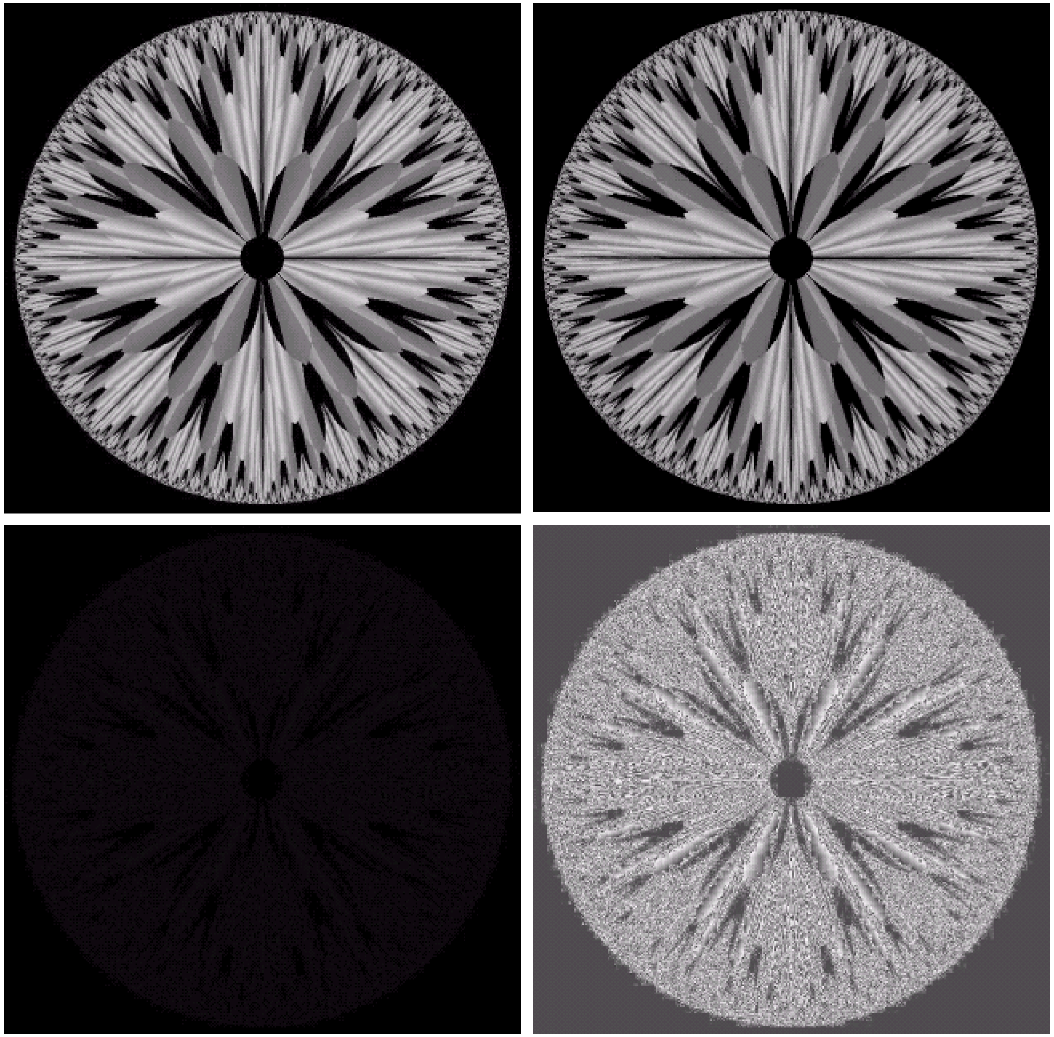


Image Subtraction



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).

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

- Enhancement of the differences between images

Note

- **We may have to adjust the gray-scale of the subtracted image to be [0,255] (if 8-bit is used)**
  - **first, find the minimum gray value of the subtracted image**
  - **second, find the maximum gray value of the subtracted image**
  - **Set the minimum value to be zero and the maximum to be 255**
  - **While the rest are adjusted according to the interval [0,255], by timing each value with 255/max**

$$s = (L-1) \times \frac{r - f_{\min}}{f_{\max} - f_{\min}}$$

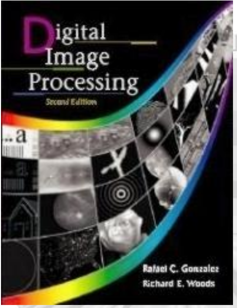


Image Averaging

- **consider a noisy image g(x,y) formed by the addition of noise η(x,y) to an original image f(x,y)**

$$g(x, y) = f(x, y) + \eta(x, y)$$

- **if noise has zero mean and be uncorrelated then it can be shown that if**

$\overline{g}(x, y) =$ image formed by averaging K different noisy images

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

Image Averaging

- **then**

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

$$\sigma^2_{\bar{g}(x,y)}, \sigma^2_{\eta(x,y)} = \text{variances of } \bar{g} \text{ and } \eta$$

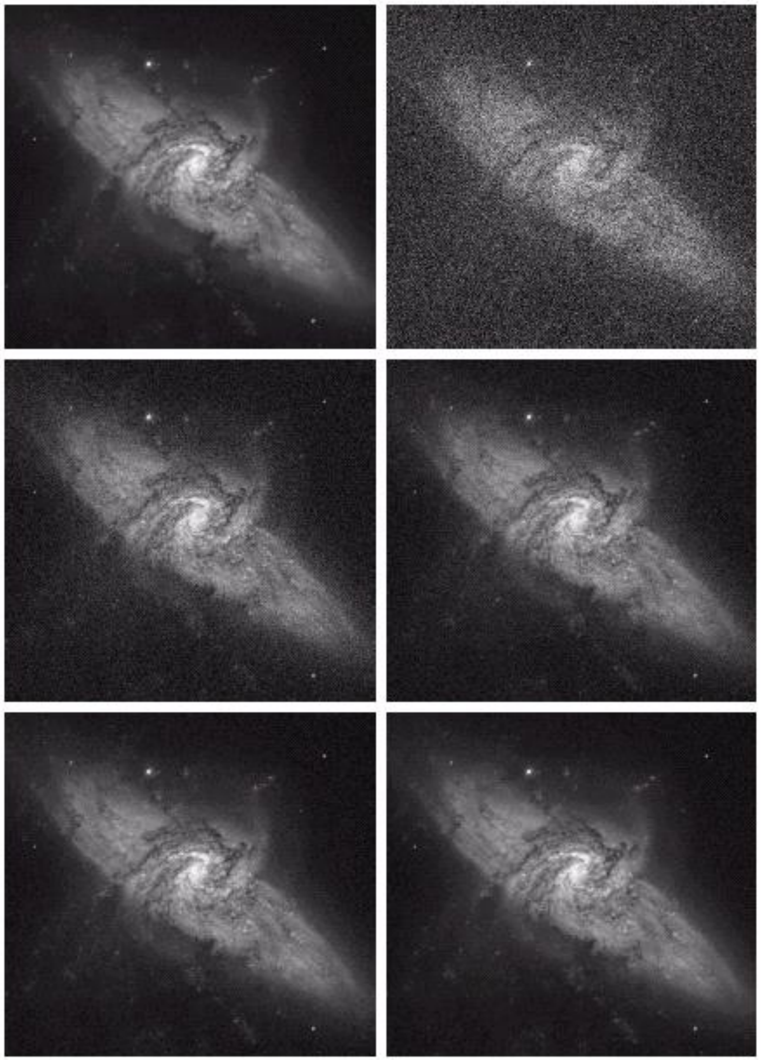
if K increase, it indicates that the variability (noise) of the pixel at each location (x,y) decreases.

- **thus**

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

$$E\{\bar{g}(x,y)\} = \text{expected value of } \bar{g}$$
$$\text{(output after averaging)}$$
$$= \text{original image } f(x,y)$$

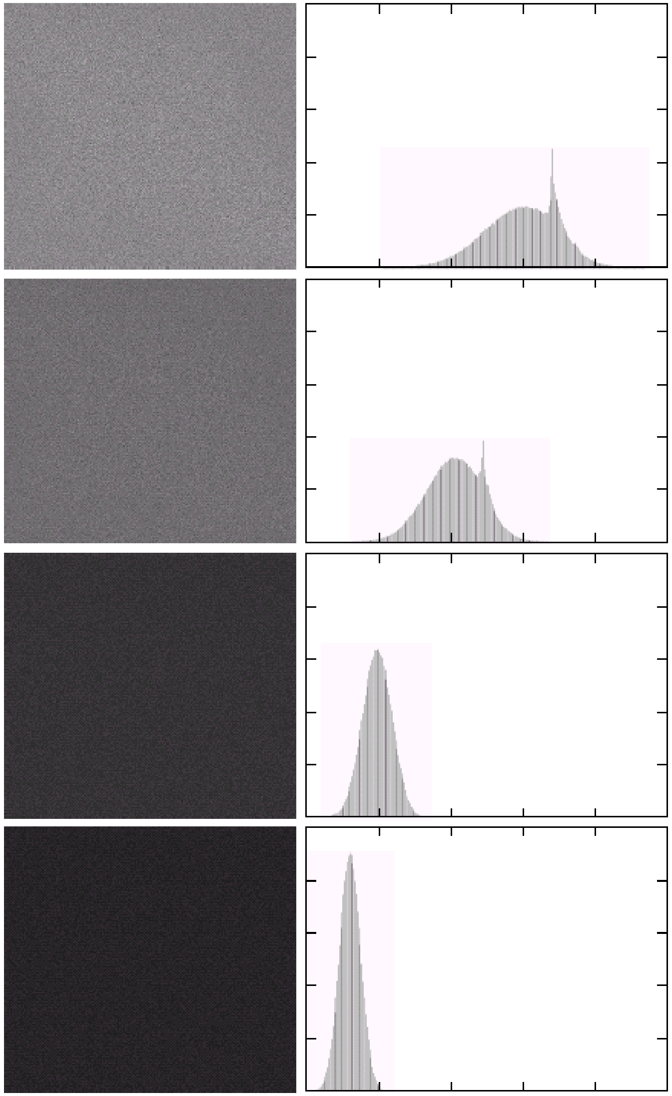
Example



a b
c d
e f

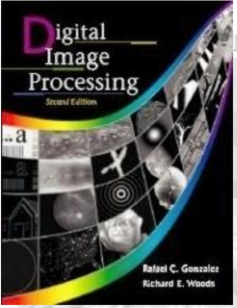
FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

Chapter 3
Image Enhancement in the Spatial Domain

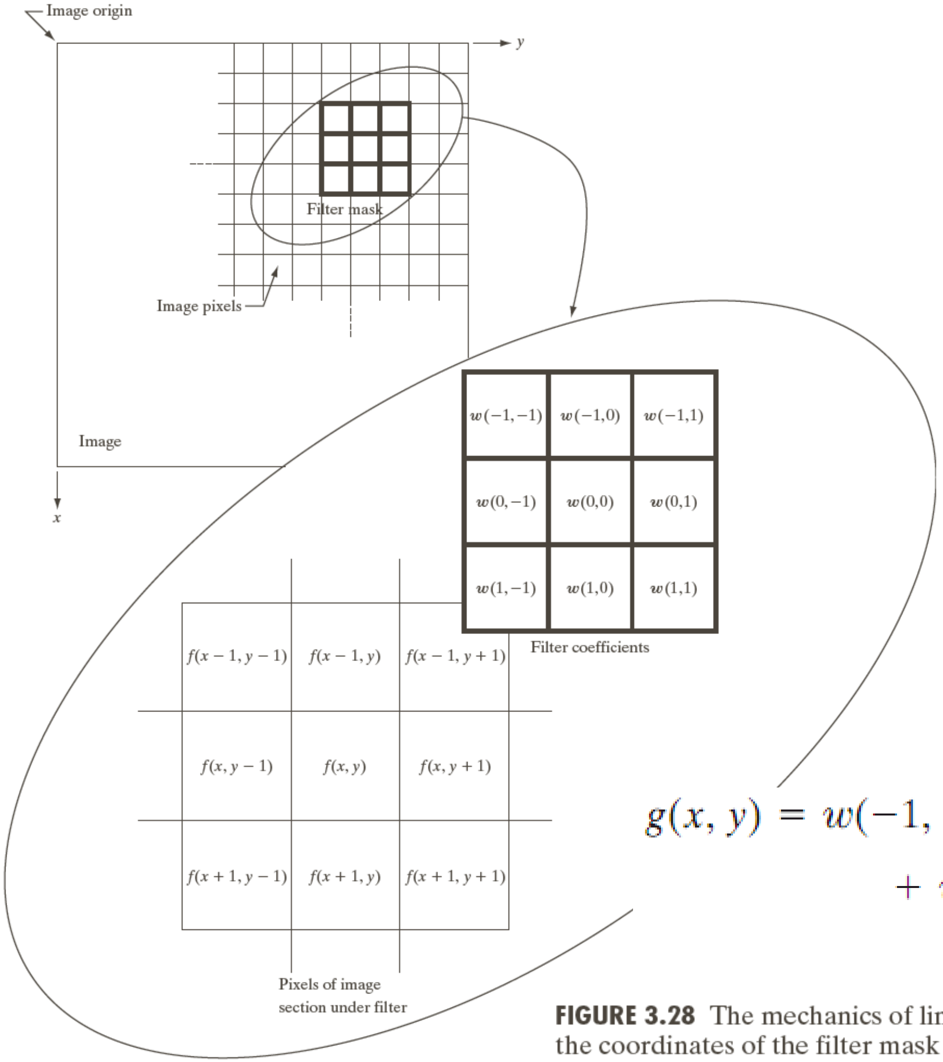


a b

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.



Spatial Filtering



- use filter (can also be called as mask/kernel/template or window)
- the values in a filter subimage are referred to as coefficients, rather than pixel
- our focus will be on masks of odd sizes, e.g. 3x3, 5x5,…

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

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

Correlation Operation (Step1)

Input

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

$n_H \, x \, n_W = 6 \, x \, 6$

$*$

Filter

1	0	-1
1	0	-1
1	0	-1

Parameters:

Size: $f = 3$

Stride: $s = 1$

Padding: $p = 0$

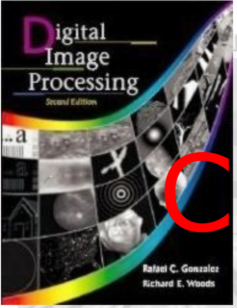
$=$

Result

2			

$\boxed{2}$ = 4*1 + 9*0 + 2*(-1) +
5*1 + 6*0 + 2*(-1) +
2*1 + 4*0 + 5*(-1)

https://indoml.com



Correlation Operation (Step2)

Input

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

$n_H \, x \, n_W = 6 \, x \, 6$

$*$

Filter

1	0	-1
1	0	-1
1	0	-1

$=$

Result

2	6		

Parameters:

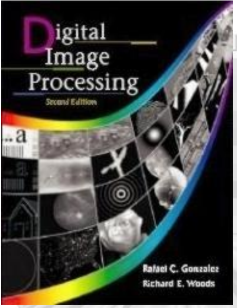
Size: $f = 3$

Stride: $s = 1$

Padding: $p = 0$

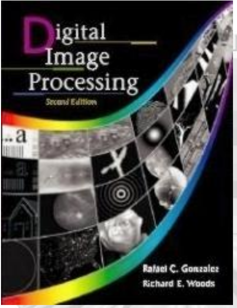
$\boxed{6} = 9*1 + 2*0 + 5*(-1) + 6*1 + 2*0 + 4*(-1) + 4*1 + 5*0 + 4*(-1)$

https://indoml.com



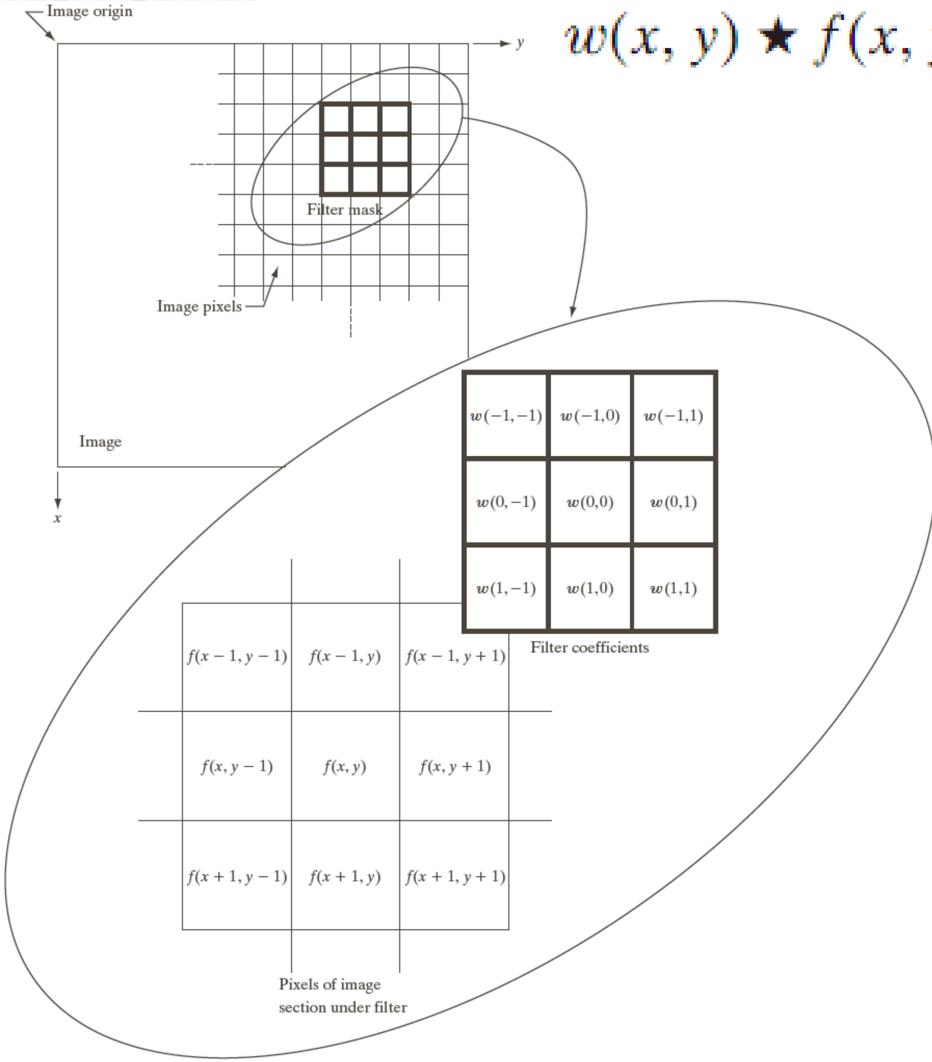
Spatial Correlation and Convolution

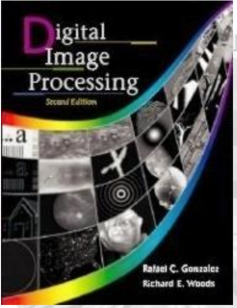
- **Correlation:** $w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x+s, y+t)$
  - **Process of moving a filter mask over the image and computing the sum of products at each location.**
- **Convolution:** $w(x, y) \bigstar f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x-s, y-t)$
  - **The mechanics of convolution are the same, except that the filter is first rotated by 180°.**



Spatial Convolution

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

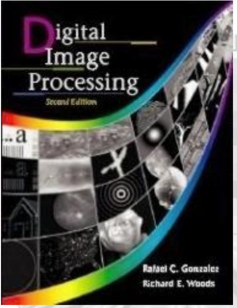




1D Correlation & Convolution

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

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



2D Correlation & Convolution

Origin $f(x, y)$

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

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

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

(b)

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

Initial position for w

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

(c)

Full correlation result

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

(d)

Cropped correlation result

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

(e)

Rotated w

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

(f)

Full convolution result

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

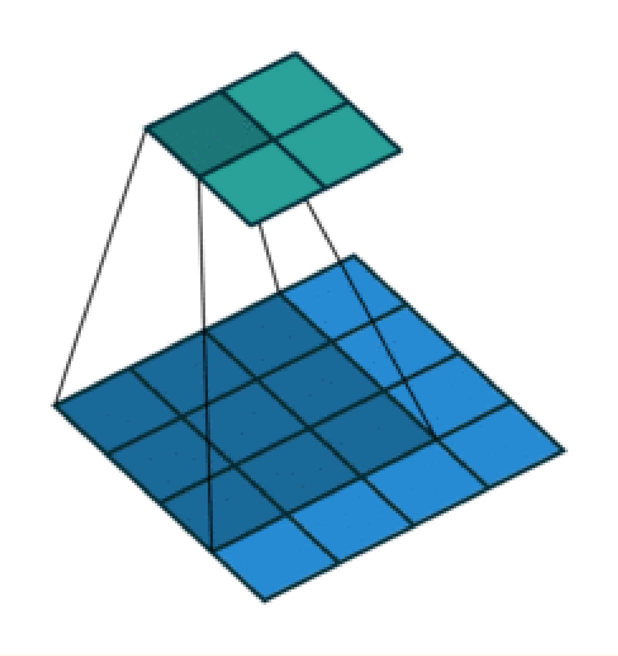
(g)

Cropped convolution result

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

(h)

➢ Correlation is a function
of r.
ne
n.



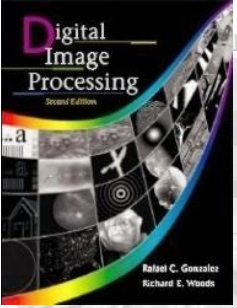
➢ Correlation of a function with a discrete unit impulse yields a rotated version of the function at the location of the impulse.

Vector Representation of Linear Filtering

- **simply move the filter mask from point to point in an image.**
- **at each point (x,y), the response of the filter at that point is calculated using a predefined characteristics.**

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn}$$
$$= \sum_{i=1}^{mn} w_i z_i$$

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



Linear Filtering

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

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

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

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

Smoothing Spatial Filters

- **Used for blurring and for noise reduction**
- **Blurring is used in preprocessing steps, such as**
  - **removal of small details from an image prior to object extraction**
  - **bridging of small gaps in lines or curves**
- **Noise reduction can be accomplished by blurring with a linear filter and also by a nonlinear filter**

Smoothing Linear Filters

- **Output is simply the average of the pixels contained in the neighborhood of the filter mask.**
- **Called averaging filters or lowpass filters.**

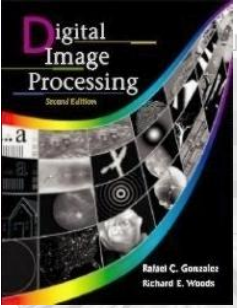
$$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$

Average Filter Response

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

Gaussian function of two variables

- **2-D Gaussian function has a bell shape, and that the standard deviation controls the "tightness" of the bell**



3x3 Smoothing Linear Filters

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

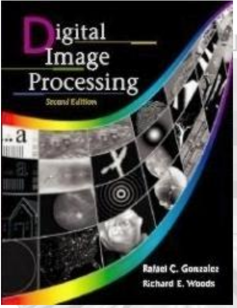
Box filter   Weighted Average Filter

a b

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

Weighted Average:

the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask



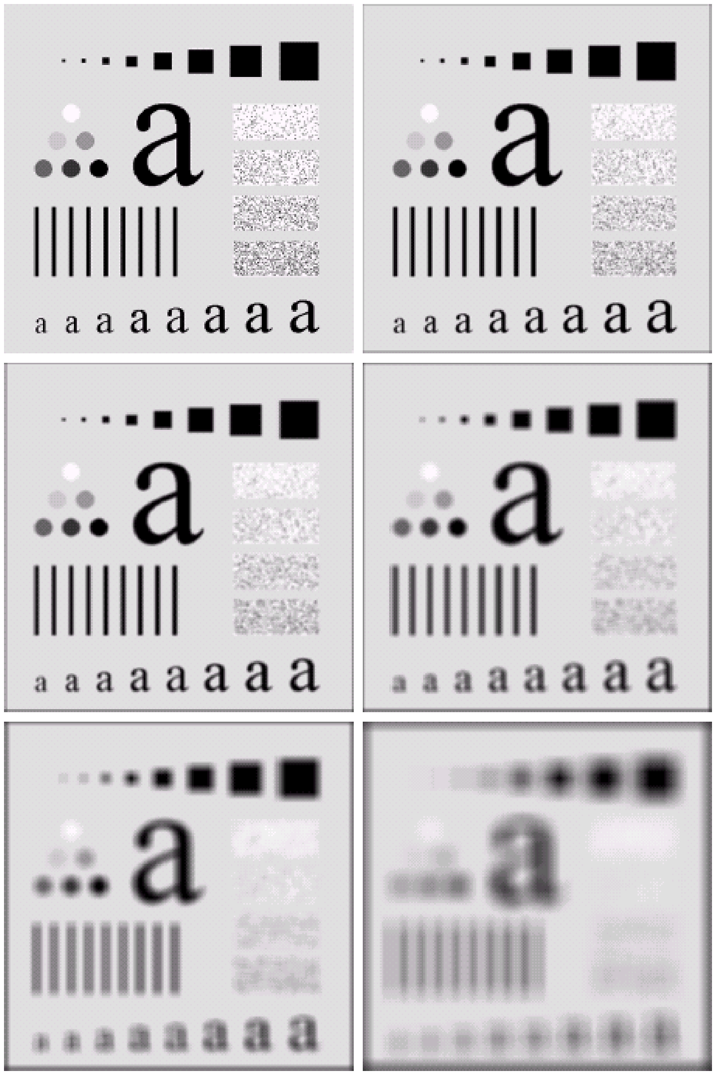
General form : Smoothing mask

- filter of size mxn (m and n odd)

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

summation of all coefficient of the mask

Example: Image Smoothing

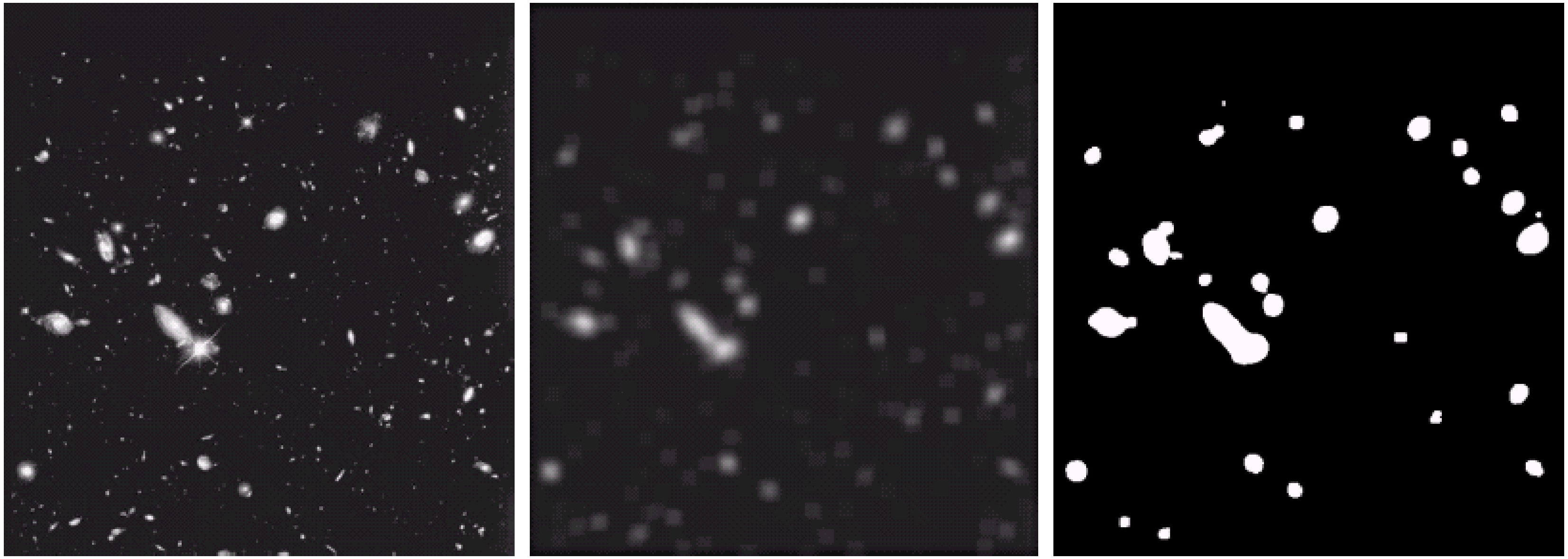


- Big mask is used to eliminate small objects from an image
- The size of the mask establishes that relative size of the objects that will be blended with the background.

a b
c d
e f

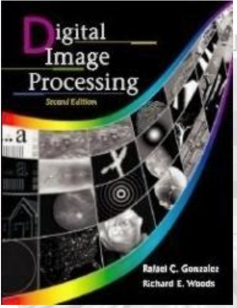
FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart.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

Real Life Example



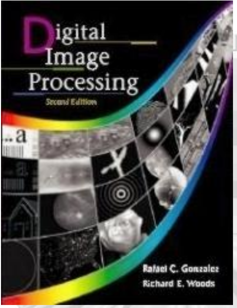
a b c

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.)



Order-Statistics Filters (Nonlinear Filters)

- **The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter**
- **Example**
  - **median filter : $R = \text{median}\{Z_k \mid k = 1, 2, \ldots, n\text{x}n\}$**
  - **max filter : $R = \max\{Z_k \mid k = 1, 2, \ldots, n\text{x}n\}$**
  - **min filter : $R = \min\{Z_k \mid k = 1, 2, \ldots, n\text{x}n\}$**
- **Note : nxn is the size of the mask**



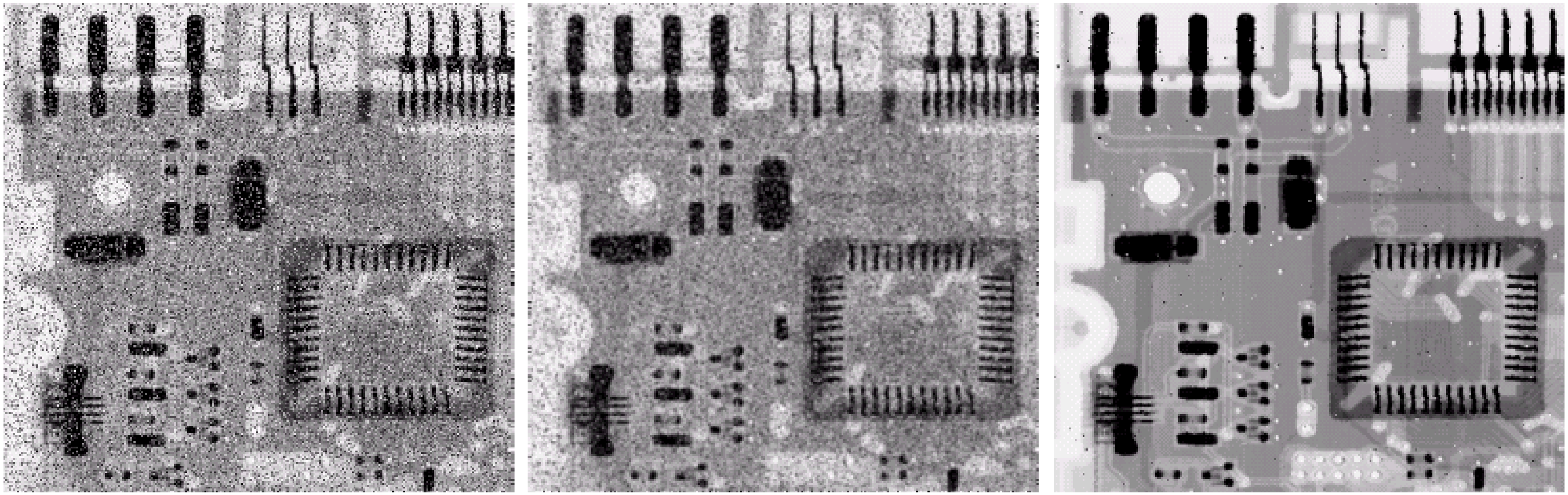
Median Filters

- **Replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median)**
- **Popular because for certain types of random noise (impulse noise → salt and pepper noise), they provide excellent noise-reduction capabilities, with considering less blurring than linear smoothing filters of similar size.**
- **The median represents the 50th percentile of a ranked set of numbers**

Median Filters

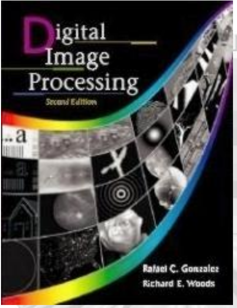
- Forces the points with distinct gray levels to be more like their neighbors.
- Isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n x n median filter.
- Forced to have the value equal the median intensity of the neighbors.
- Larger clusters are affected considerably less

Example : Median Filter



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.)



Sharpening Spatial Filters

Objectives:

- **to highlight fine detail in an image**
- **to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition**

Blurring vs. Sharpening

- **as we know that blurring can be done in spatial domain by pixel averaging in a neighbors**
- **since averaging is analogous to integration**
- **thus, we can guess that the sharpening must be accomplished by spatial differentiation.**

Derivative Operator

- **the strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.**
- **thus, image differentiation**
  - **enhances edges and other discontinuities (noise)**
  - **deemphasizes area with slowly varying gray-level values.**

First-order derivative

- a basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

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

Second-order derivative

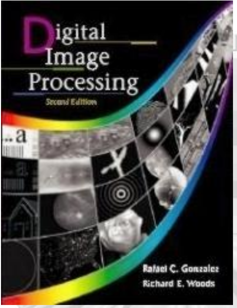
- similarly, we define the second-order derivative of a one-dimensional function f(x) is the difference

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

Conditions of Derivative Output

- **For a first derivative**
  - **(1) must be zero in areas of constant intensity;**
  - **(2) must be nonzero at the onset of an intensity step or ramp; and**
  - **(3) must be nonzero along ramps.**
- **For a second derivative**
  - **(1) must be zero in constant areas;**
  - **(2) must be nonzero at the onset and end of an intensity step or ramp; and**
  - **(3) must be zero along ramps of constant slope.**

- **Because we are dealing with digital quantities whose values are finite, the maximum possible intensity change also is finite, and the shortest distance over which that change can occur is between adjacent pixels.**



First and Second-order derivative of f(x,y)

- when we consider an image function of two variables, f(x,y), at which time we will dealing with partial derivatives along the two spatial axes.

Gradient operator

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

Laplacian operator

(linear operator)

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

Discrete Form of Laplacian

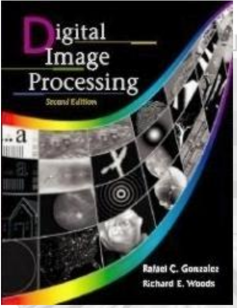
from

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

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

yield,

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



Example

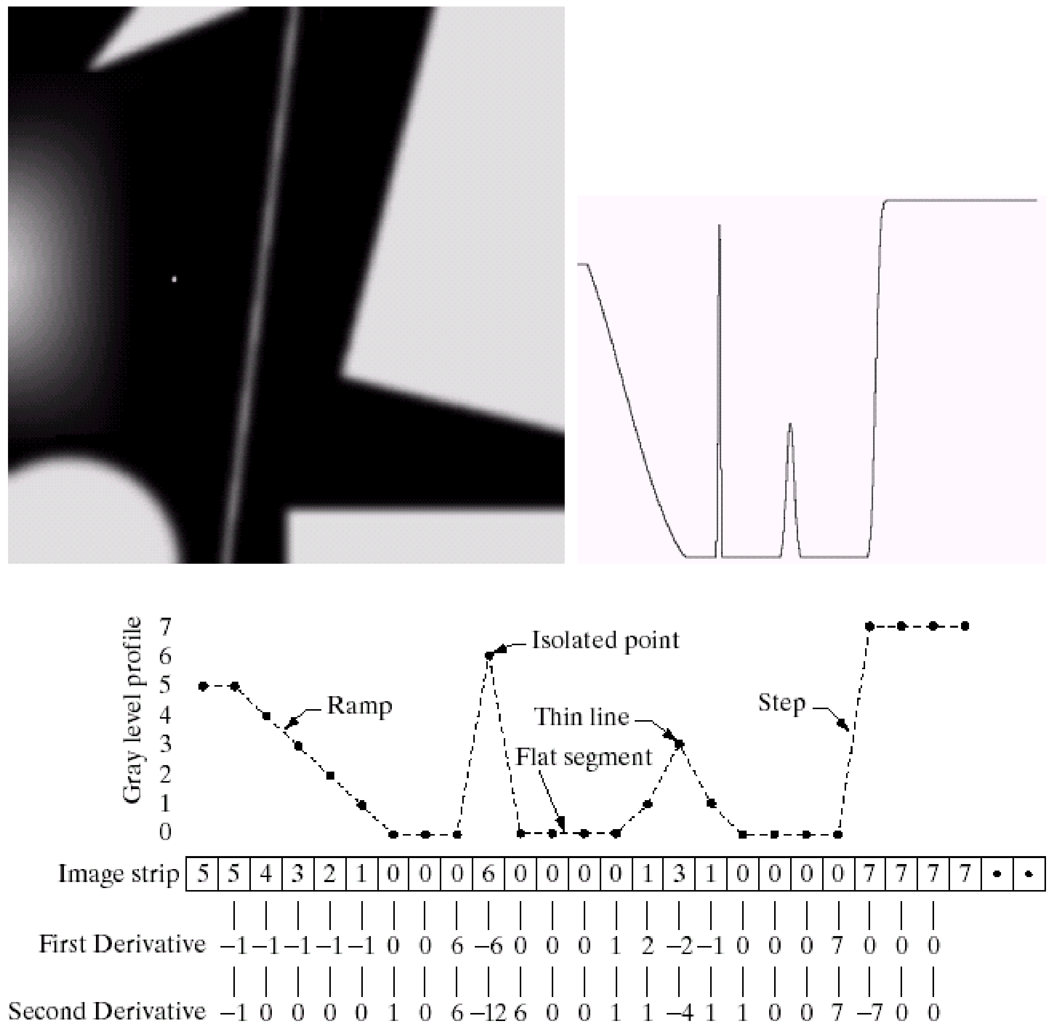
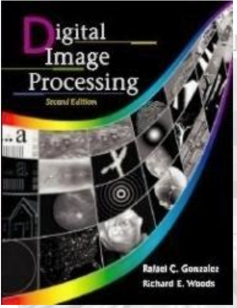


FIGURE 3.38 (a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



Laplacian masks

- We are interested in isotropic filters, whose response is independent of the direction of the discontinuities in the image to which the filter is applied.
  - In other words, isotropic filters are rotation invariant

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

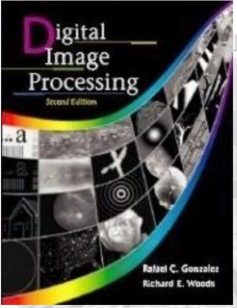
0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a	b
c	d

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



Effect of Laplacian Operator

- **as it a derivative operator,**
  - **it highlights gray-level discontinuities in an image**
  - **it deemphasizes regions with slowly varying gray levels**
- **tends to produce images that have**
  - **grayish edge lines and other discontinuities,**

  all superimposed on a dark, featureless background.

- **Correct the Effect of featureless background**

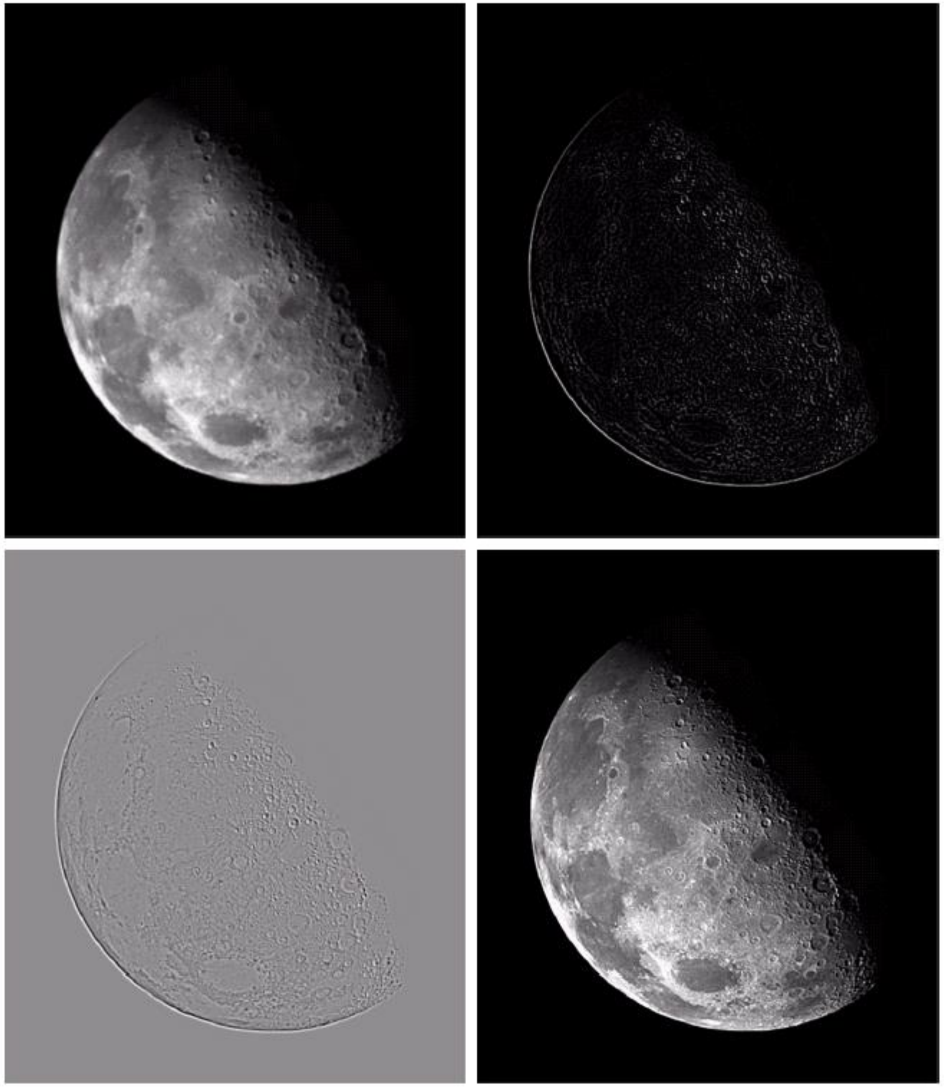
$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

if the center coefficient of the Laplacian mask is negative

if the center coefficient of the Laplacian mask is positive

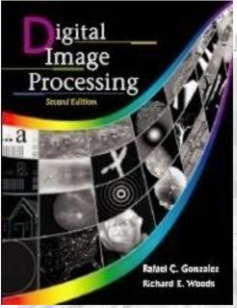
$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

Example



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.)

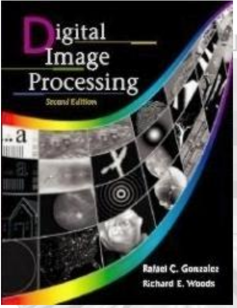


Mask of Laplacian + Addition

- **to simply the computation, we can create a mask which do both operations, Laplacian Filter and Addition the original image.**

$$
\begin{aligned}
g(x,y) &= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\quad + f(x,y+1) + f(x,y-1) + 4f(x,y)] \\
&= 5f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\quad + f(x,y+1) + f(x,y-1)]
\end{aligned}
$$

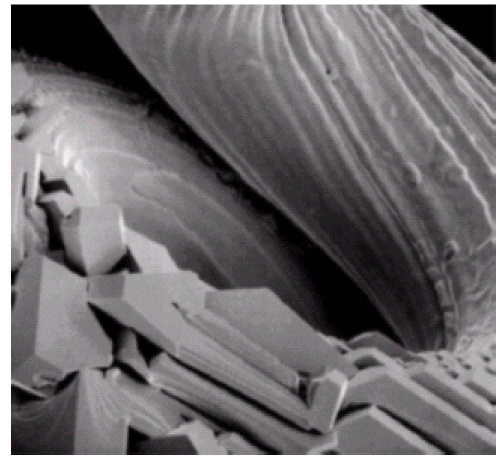
$$
\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}
=
\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}
+
\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}
$$

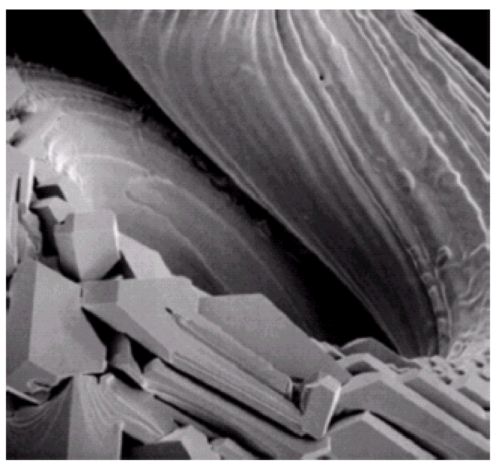


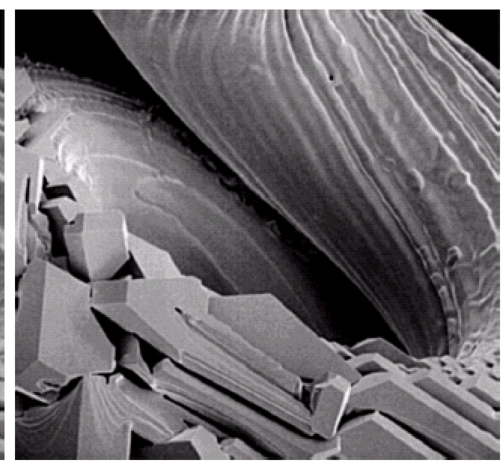
Example

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.)

Unsharp Masking

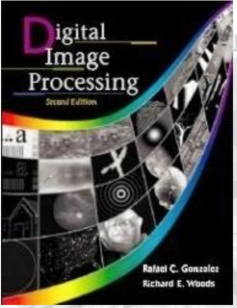
- Popular in printing and publishing industry for image sharpening.
- Objective is to subtract a blurred version of an image from the original image.
    - produces unsharp mask.

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

mask = original image – blurred image

- Add the mask to the original image

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

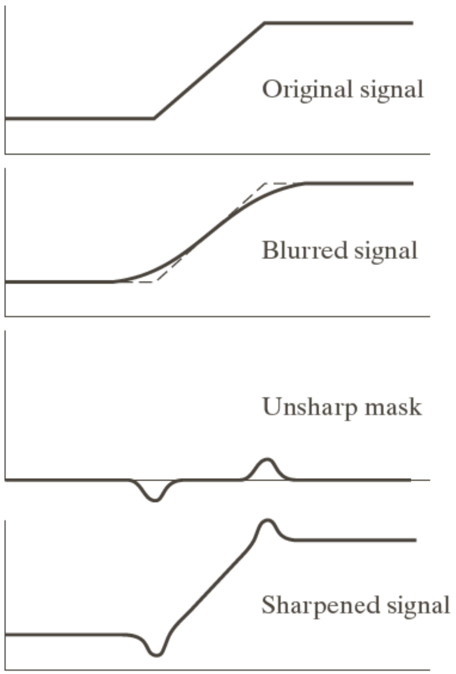


High-boost filtering

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

- Includes a weight factor *k*
  - k=1 for unsharp masking
  - k>1 known as highboost filtering
  - k<1 for deemphasizing the contribution of unsharp masking. Becomes more similar to original image.

Example

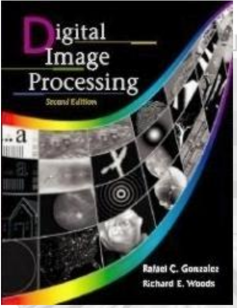


a
b
c
d

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

a
b
c
d
e

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



Gradient Operator

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

➢ First derivatives are implemented using the magnitude of the gradient.

$$\nabla f = mag(\nabla \mathbf{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

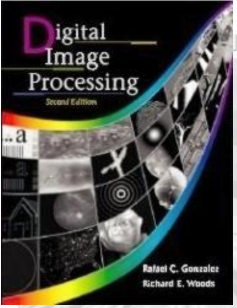
commonly approx.

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

the magnitude becomes nonlinear

➢ Components of the gradient vector are derivatives, they are linear operators. However, magnitude is not.

➢ Partial derivatives are not rotation invariant (isotropic), but the magnitude of the gradient vector is.



Gradient Mask

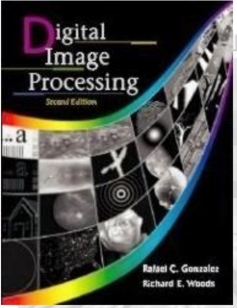
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- simplest approximation, 2x2

$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

$$\nabla f = [G_x^2 + G_y^2]^{\frac{1}{2}} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{\frac{1}{2}}$$

$$\nabla f \approx |z_8 - z_5| + |z_6 - z_5|$$



Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Roberts cross-gradient operators, 2x2

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

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

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

-1	0
0	1

0	-1
1	0

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Sobel operators, 3x3

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

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

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

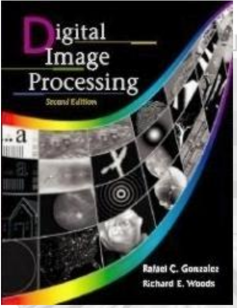
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

the weight value 2 is to achieve smoothing by giving more important to the center point

Note

- **Summation of coefficients in all masks equals 0, indicating that they would give a response of 0 in an area of contrast gray level.**



Masks

a
b c
d e

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.

(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

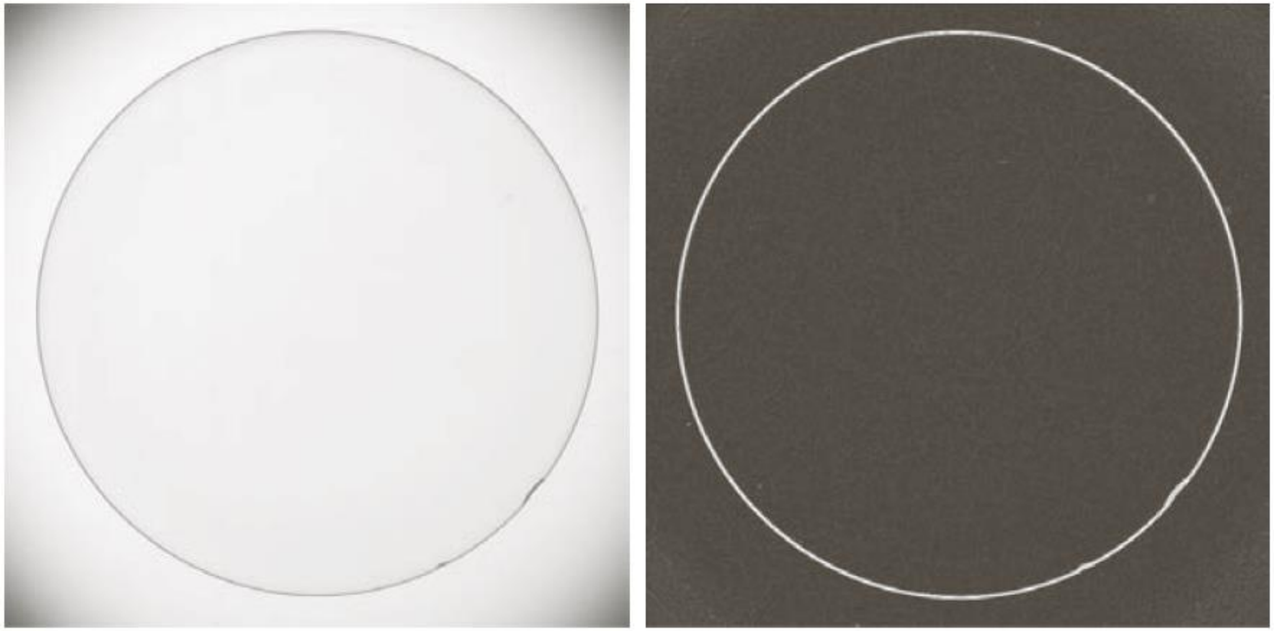
(d)

−1	−2	−1
0	0	0
1	2	1

(e)

−1	0	1
−2	0	2
−1	0	1

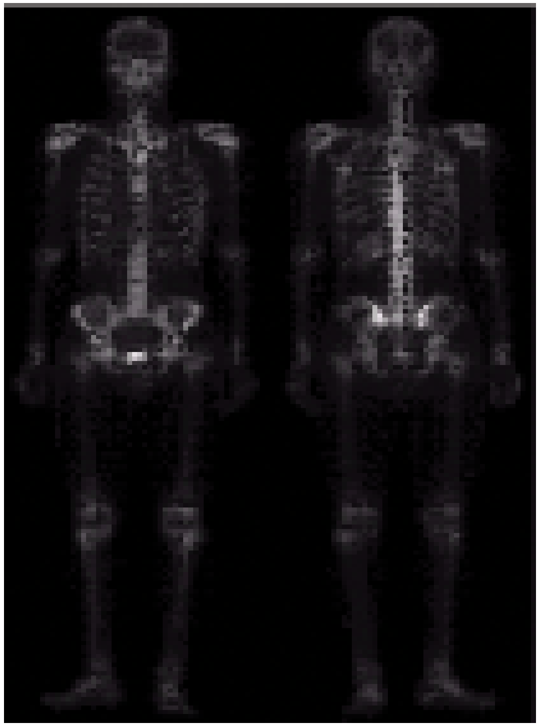
Example



a b

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

Example of Combining Spatial Enhancement Methods

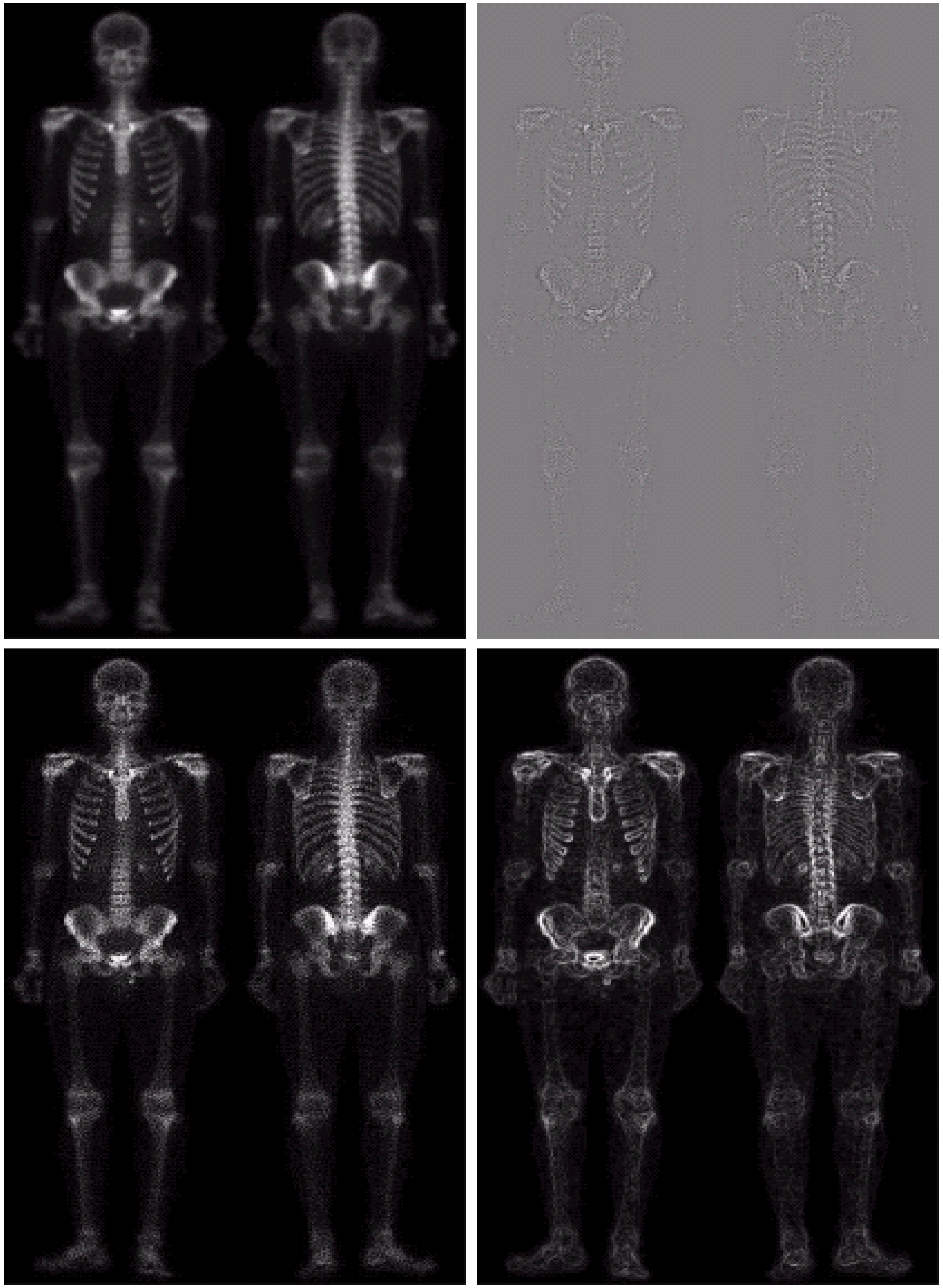


- want to sharpen the original image and bring out more skeletal detail.
- problems : narrow dynamic range of gray level and high noise content makes the image difficult to enhance

Example of Combining Spatial Enhancement Methods

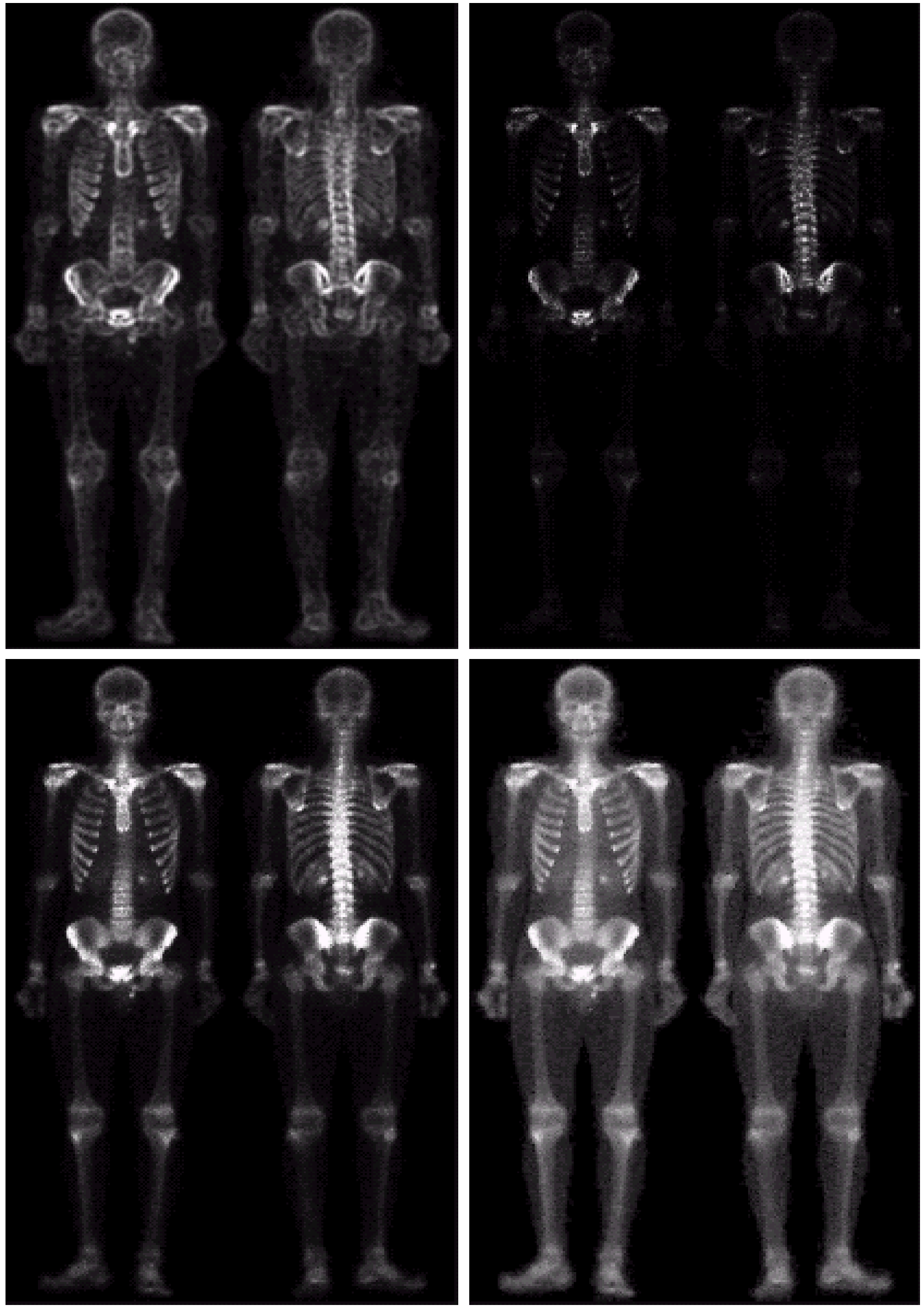
- **solve :**

  1. **Laplacian to highlight fine detail**
  2. **gradient to enhance prominent edges**
  3. **gray-level transformation to increase the dynamic range of gray levels**



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).



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.)

End of Coverage