MIT School of Computing
Department of Computer Science & Engineering

MIT-ADT UNIVERSITY
PUNE, INDIA

Artificial Intelligence

Class -L.Y. (SEM-I), AIA

Unit - II Image Enhancement in the Spatial Domain

Dr. Nilima Kulkarni- Course Coordinator
Prof. Jyoti Manoorkar

AY 2025-2026 SEM-I

Chapter 2
Image Enhancement in the Spatial Domain

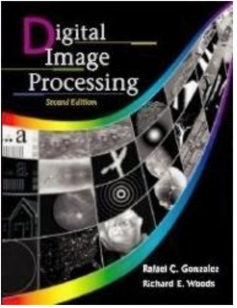
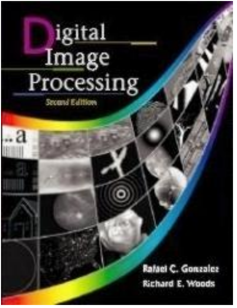
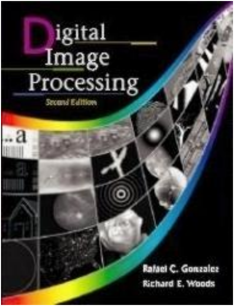


Image Enhancement in the Spatial Domain

- The **spatial domain**:
  - The image plane
  - For a digital image is a Cartesian coordinate system of discrete rows and columns. At the intersection of each row and column is a pixel. Each pixel has a value, which we will call intensity.
- The **frequency domain** :
  - A (2-dimensional) discrete Fourier transform of the spatial domain
  - We will discuss it in unit 3.
- **Enhancement** :
  - To “improve” the usefulness of an image by using some transformation on the image.
  - Often the improvement is to help make the image “better” looking, such as increasing the intensity or contrast.



- Spatial domain processing: direct manipulation of pixels in an image
- Two categories of spatial (domain) processing
    - Intensity transformation:
        - Operate on single pixels
        - Contrast manipulation, image thresholding
    - Spatial filtering
        - Work in a neighborhood of every pixel in an image
        - Image smoothing, image sharpening



Background

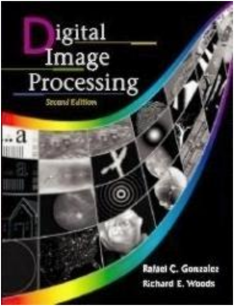
- A mathematical representation of spatial domain enhancement:

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

where $f(x, y)$: the input image

$g(x, y)$: the processed image

T: an operator on f, defined over some neighborhood of (x, y)



Spatial filtering

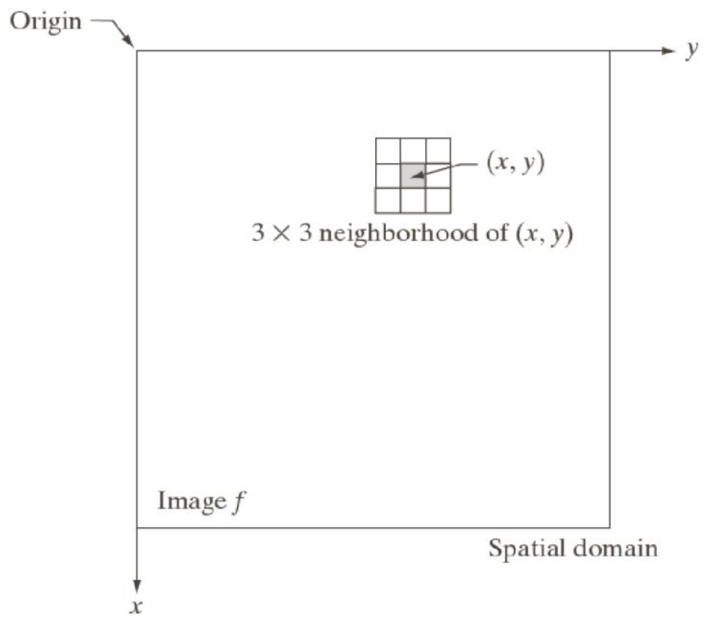
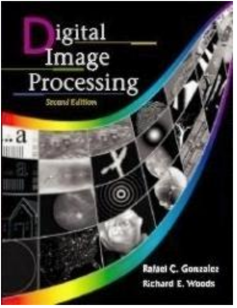


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

- For any location (x,y), output image g(x,y) is equal to the result of applying T to the neighborhood of (x,y) in f
- Filter: mask, kernel, template, window



3.1 Background

- The simplest form of T: g depends only on the value of f at (x, y)

  T becomes intensity (gray-level) transformation function

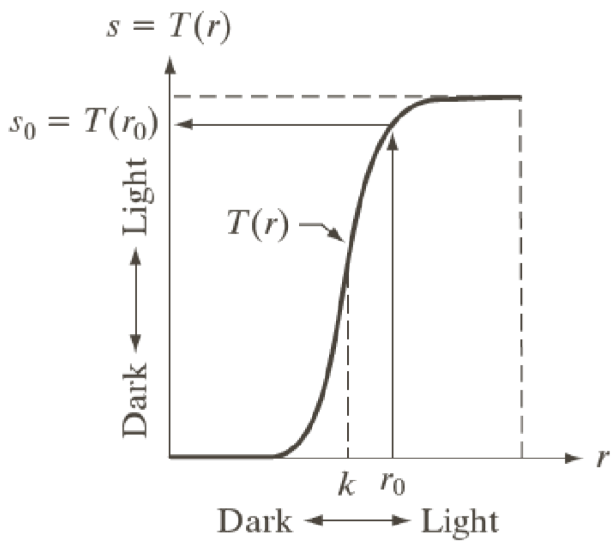
  $s = T(r)$

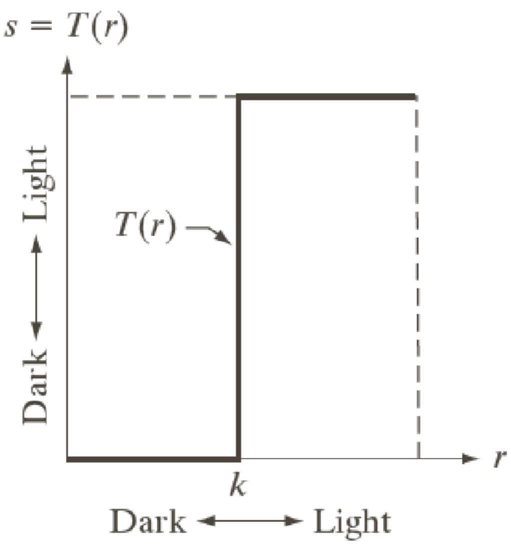
  r: intensity of $f(x,y)$

  s: intensity of $g(x,y)$

- Point processing: enhancement at any point depends only on the gray level at that point

3.1 Background



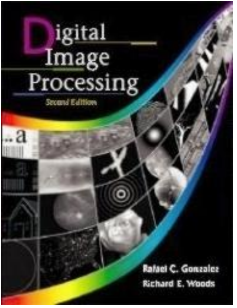


Point processing

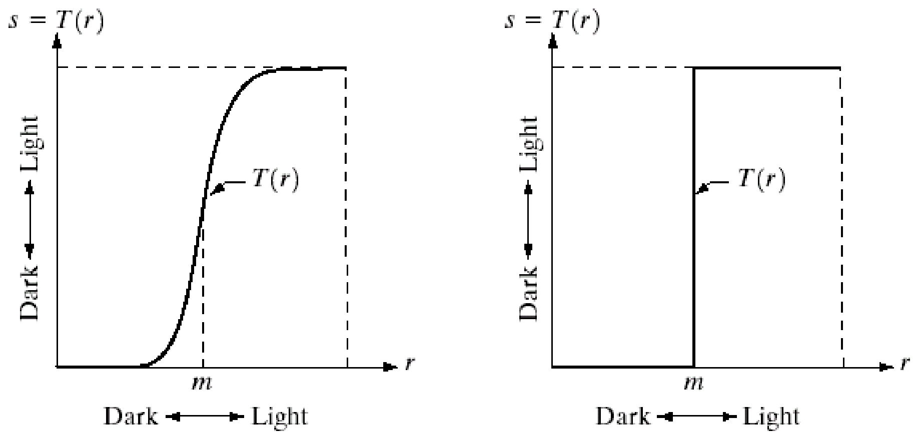
a b

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

- (a) Contrast stretching
  - Values of r below k are compressed into a narrow range of s
- (b) Thresholding



Gray-level Transformation



a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

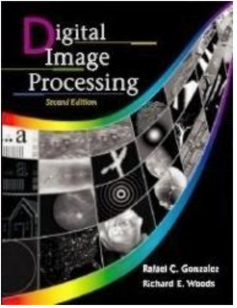
3.2 Some Basic Intensity Transformation Functions

- r: pixel value before processing

  s: pixel value after processing

  T: transformation

  s = T(r)
- 3 types
  - Linear (identity and negative transformations)
  - Logarithmic (log and inverse-log transformations)
  - Power-law (*n*th power and *n*th root transformations)



Some Basic Gray Level Transformations

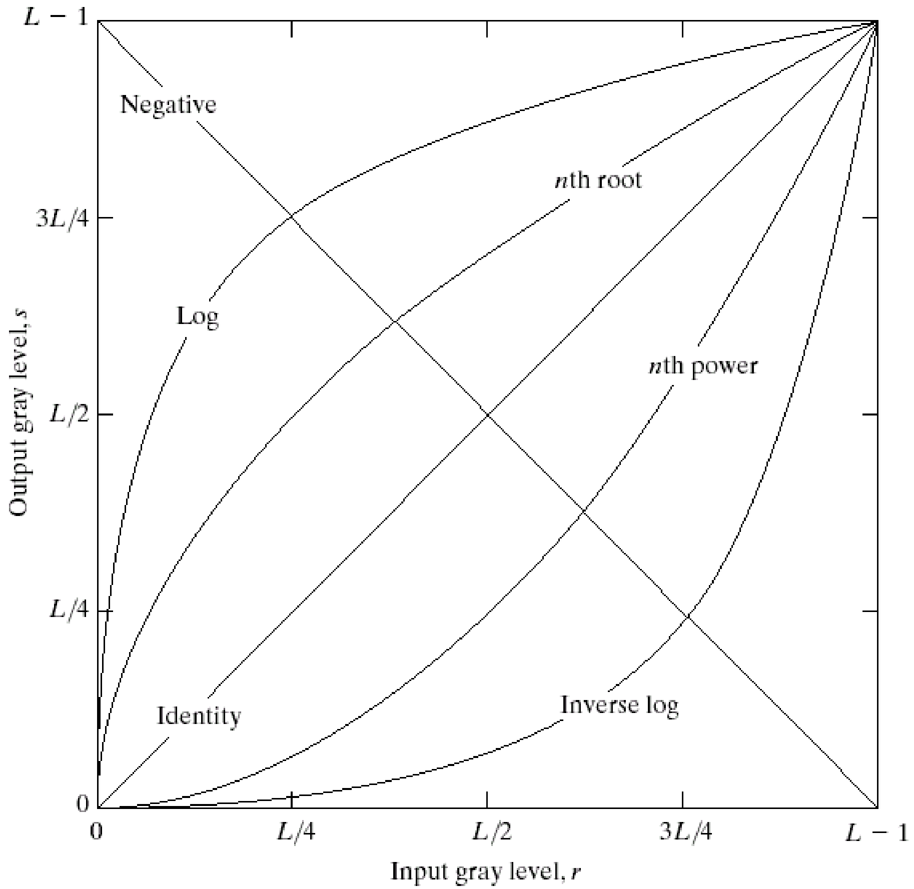


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

- Negative of an image with gray level [0, L-1]

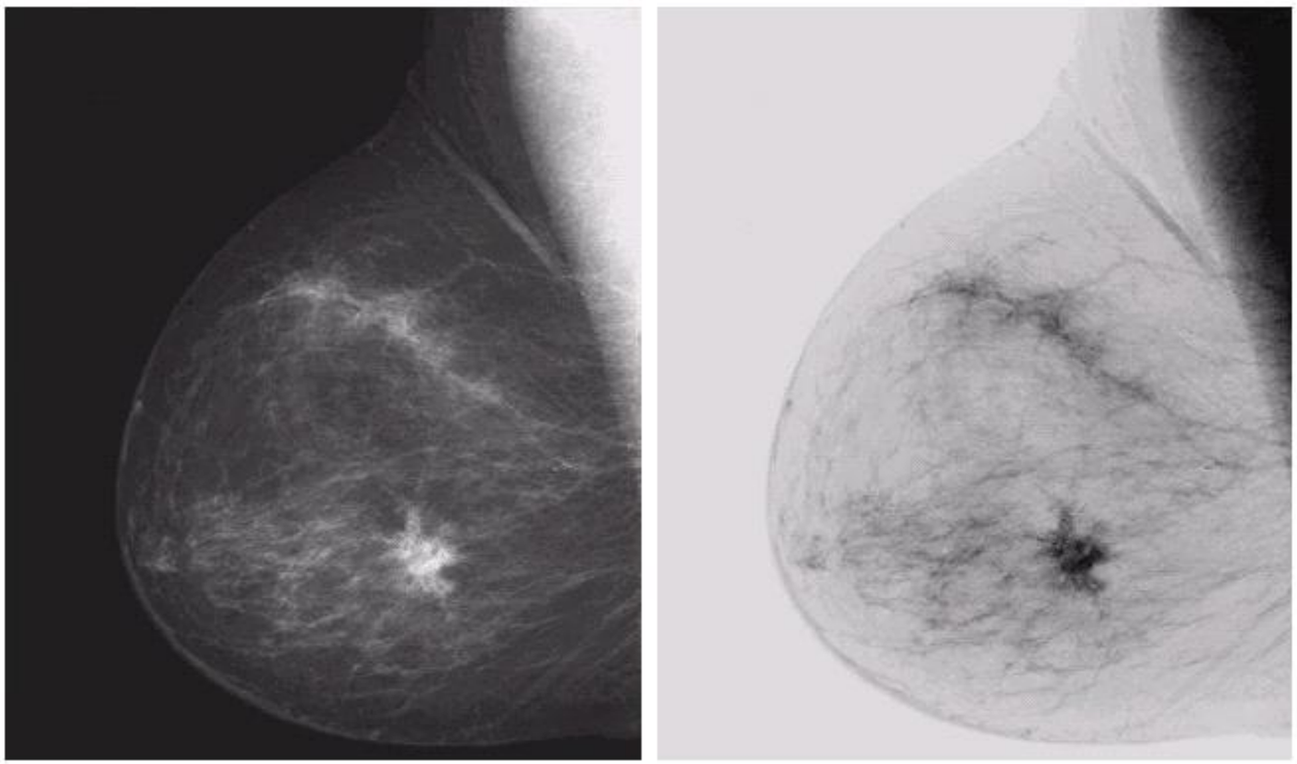
  $s = L - 1 - r$

- Enhancing white or gray detail embedded in dark regions of an image

Image Negatives

- Let the range of gray level be [0, L-1], then

$$s = L - 1 - r$$

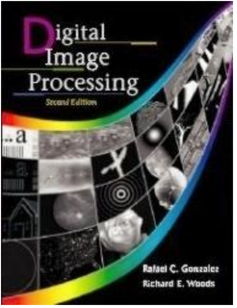


a b

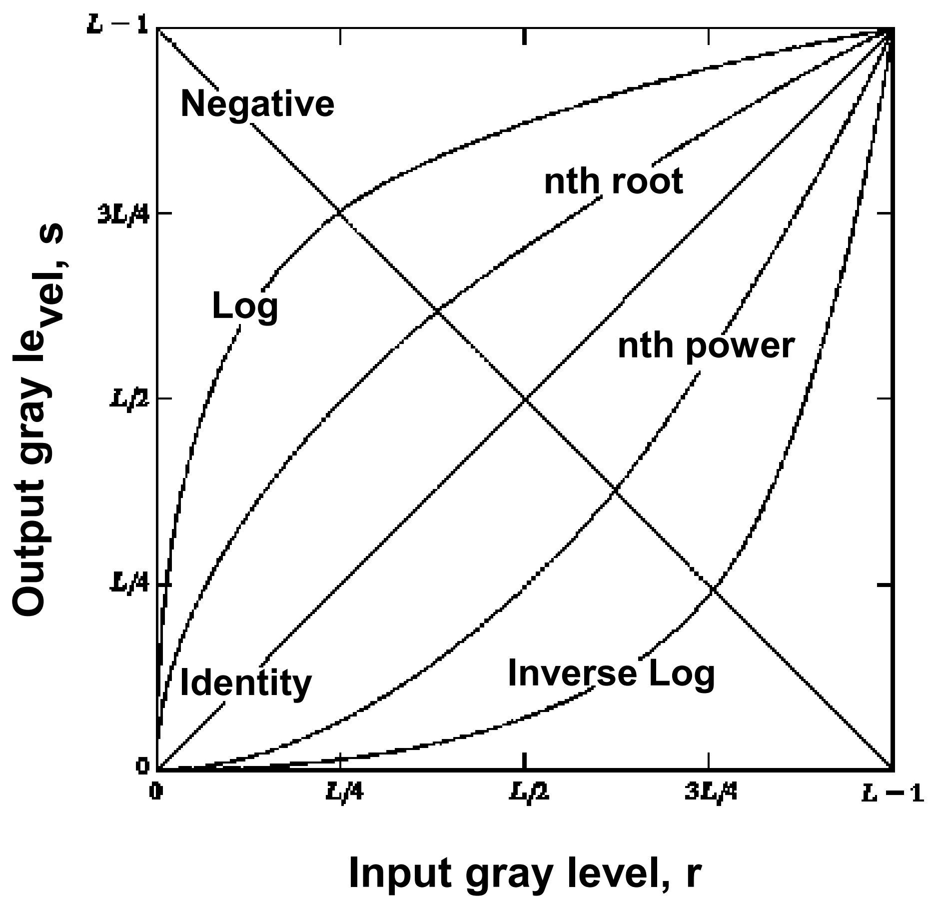
FIGURE 3.4
(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).
(Courtesy of G.E. Medical Systems.)

Log Transformations

- It compresses the dynamic range of images with large variations in pixel values
- Example of image with dynamic range: Fourier spectrum image
- It can have intensity range from 0 to 10^6 or higher.
- We can't see the significant degree of detail as it will be lost in the display.

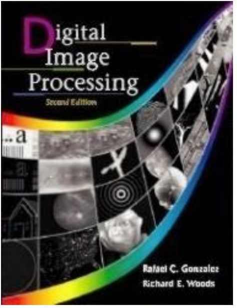


Log Transformations



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

- c is a constant and $r \geq 0$
- 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.



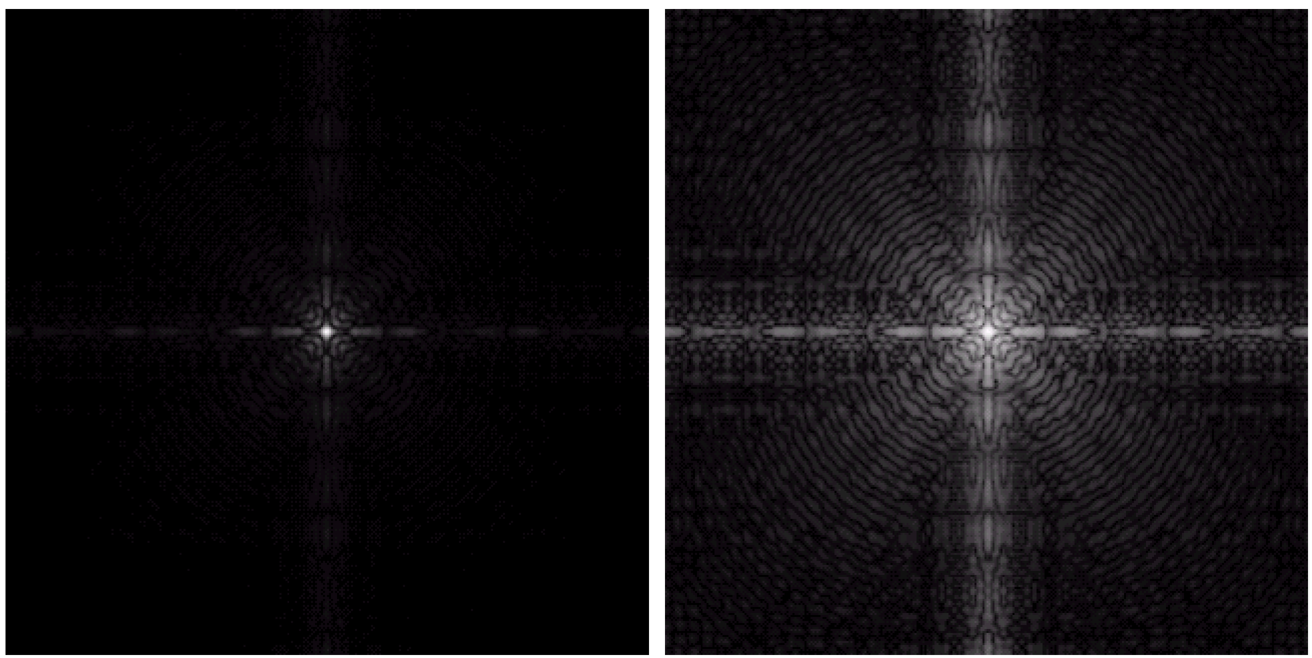
Log Transformations

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

where c : constant

$r \geq 0$

Enhance details in the darker regions of an image by expensing detail in brighter regions.

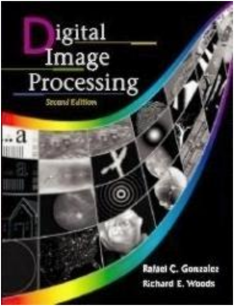


a b

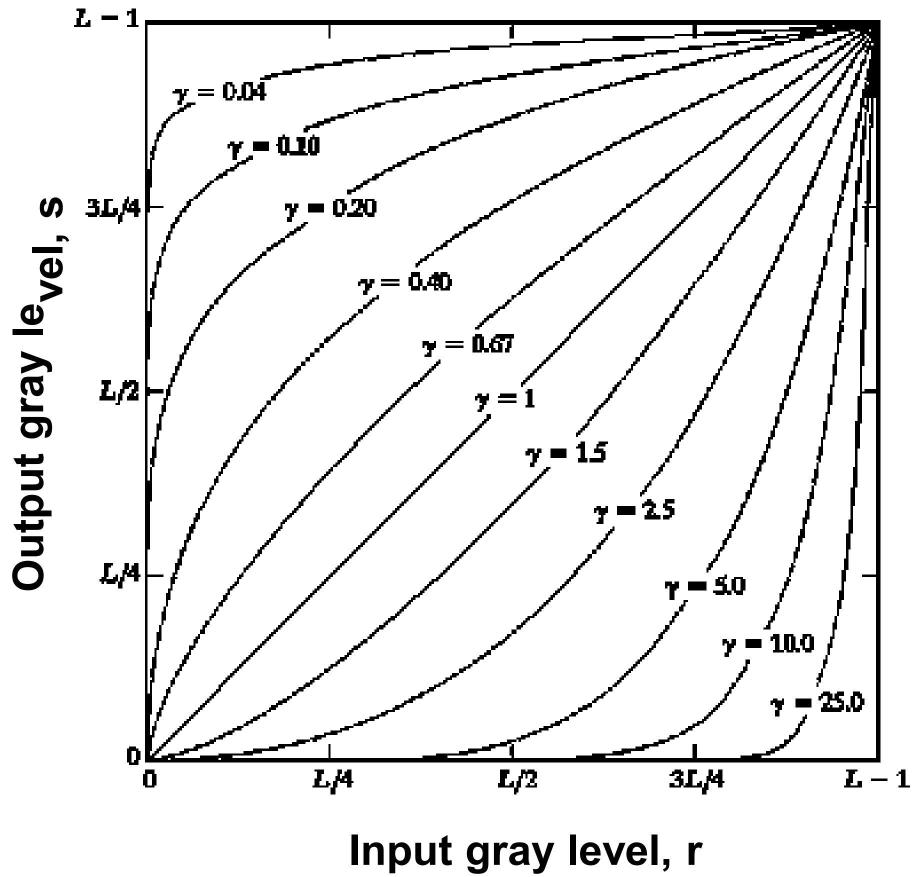
FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.

Inverse Logarithm Transformations

- Do opposite to the Log Transformations
- Used to expand the values of high pixels in an image while compressing the darker-level values.



Power-Law Transformations

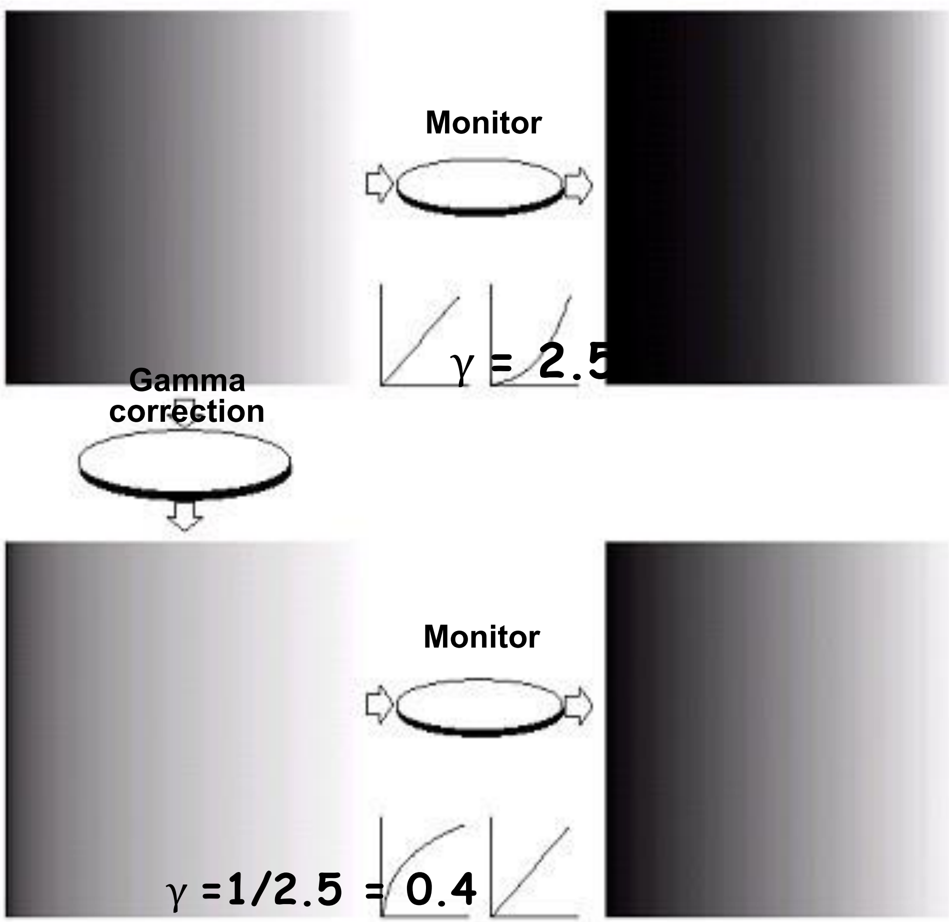


Plots of $s = cr^{\gamma}$ for various values of γ (c = 1 in all cases)

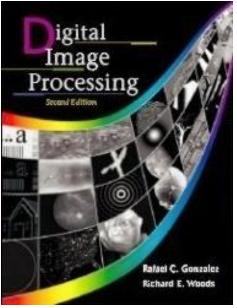
$$s = cr^{\gamma}$$

- c and γ are positive constants
- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being

Gamma correction



- Cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with γ varying from 1.8 to 2.5
- The picture will become darker.
- Gamma correction is done by preprocessing the image before inputting it to the monitor with $s = cr^{1/\gamma}$



Effect of decreasing gamma

- When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight "wash-out" look, especially in the background

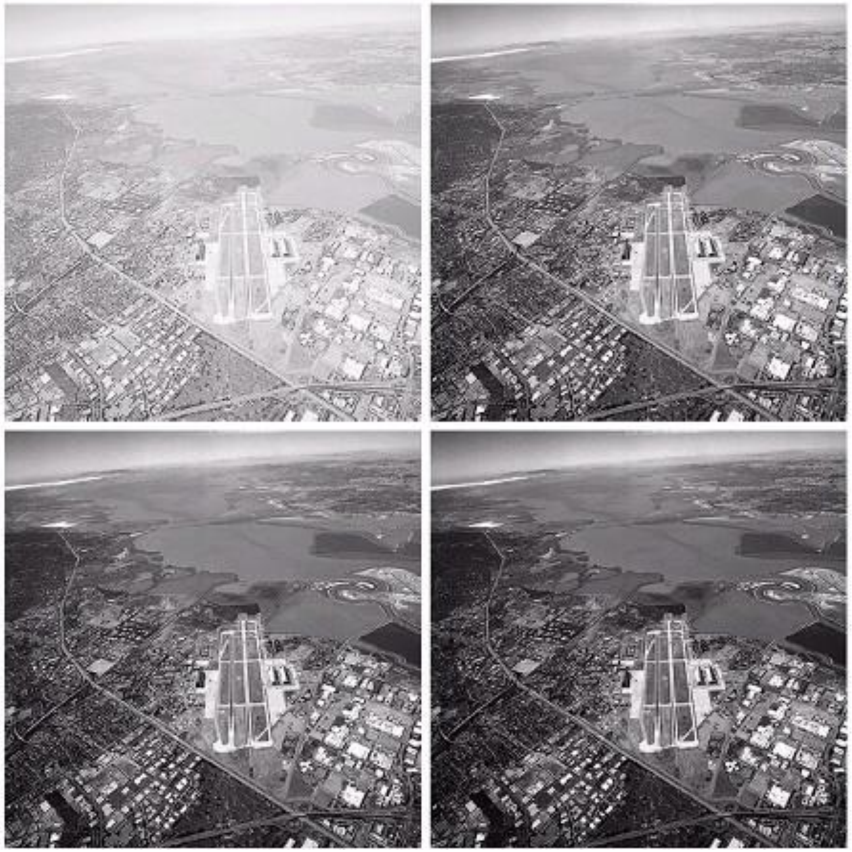
Another example

a

b

c

d



(a)image has a washed-out appearance, it needs a compression of gray levels
needs $\gamma > 1$

(b)result after power-law transformation with $\gamma = 3.0$ (suitable)

(c)transformation with $\gamma = 4.0$ (suitable)

(d)transformation with $\gamma = 5.0$ (high contrast, the image has areas that are too dark, some detail is lost)

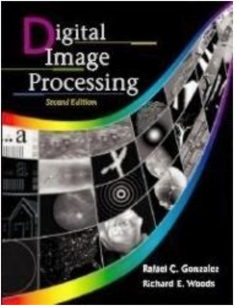
Piecewise-Linear Transformation Functions

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

Contrast Stretching

Contrast stretching (also known as **normalization**) is a simple image enhancement technique used to improve the contrast of an image by expanding the range of intensity values it contains.

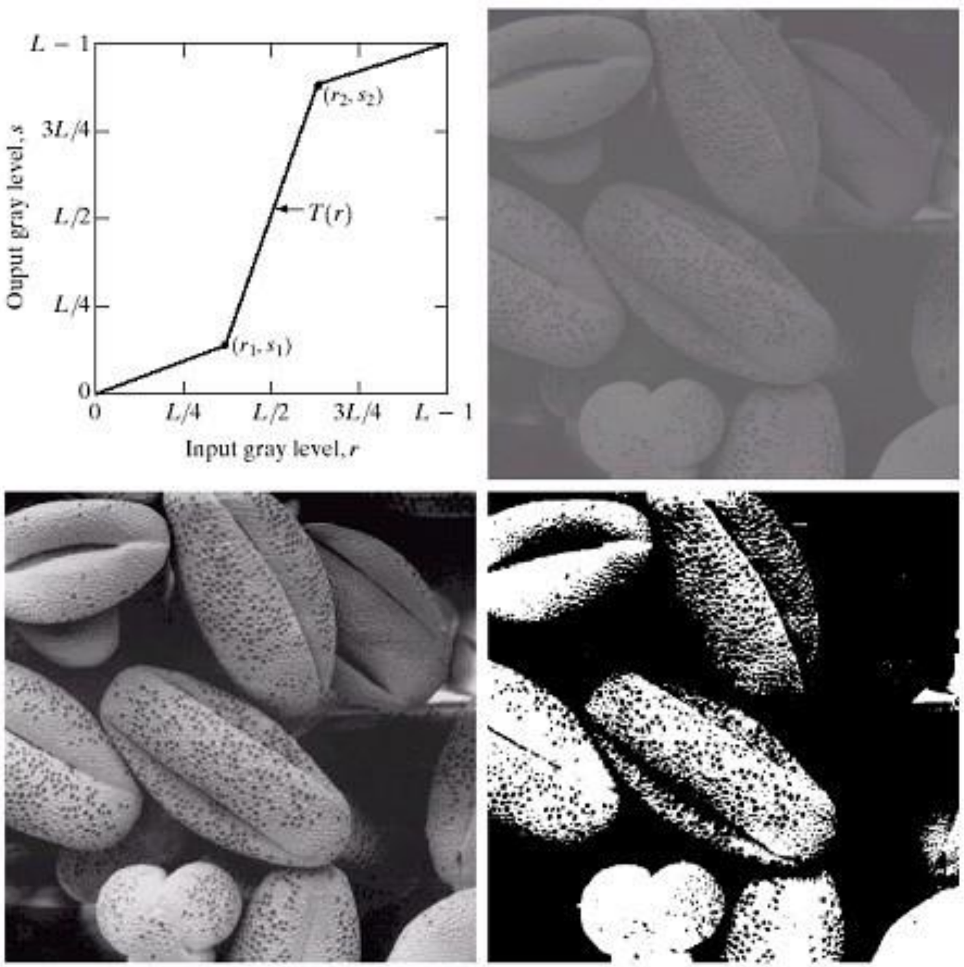
It is particularly useful for images with poor contrast due to insufficient illumination or narrow dynamic range.



When to Use Contrast Stretching?

- When the image has **low contrast but a well-defined min/max range**.
- For **quick enhancement** without complex computations.
- As a **preprocessing step** before further analysis.

Contrast Stretching



- (a)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: $(r_1,s_1) = (r_{min},0)$ and $(r_2,s_2) = (r_{max},L-1)$
- (d)result of thresholding

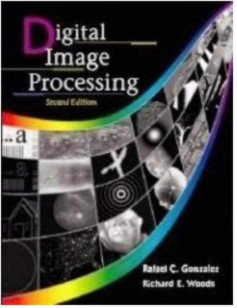
Contrast Stretching

If r1=s1 and r2=s2, the transformation is a linear function that produces no changes in gray levels.

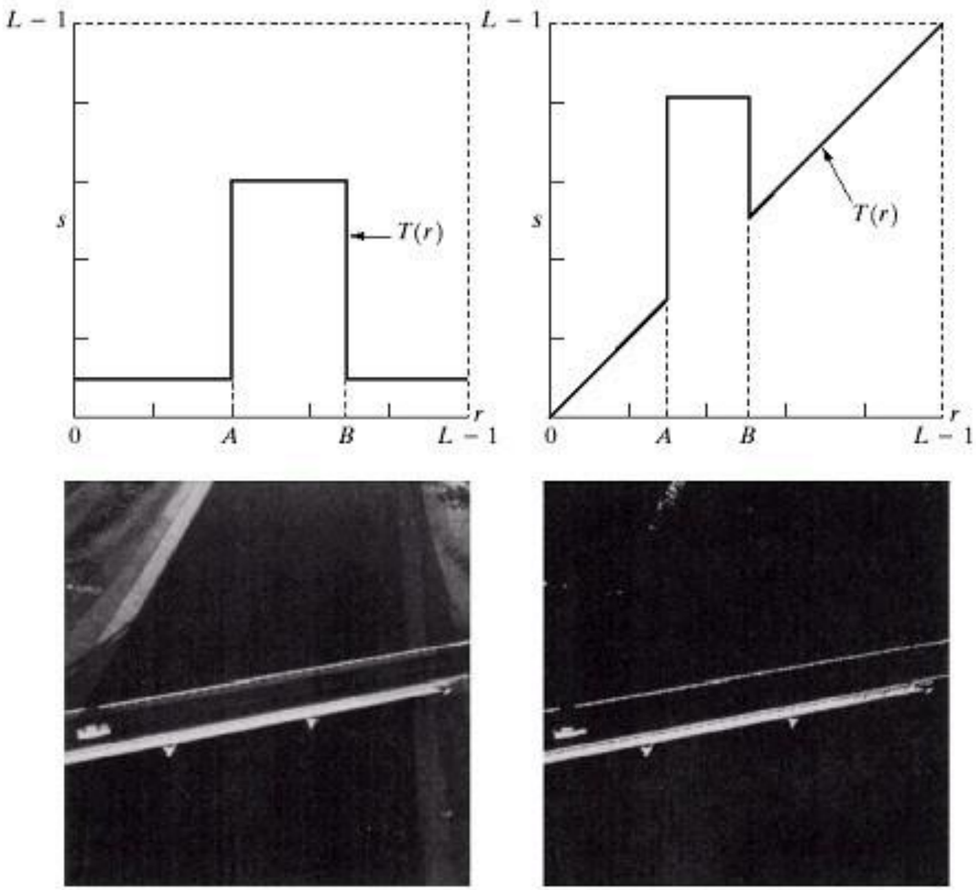
If r1=r2, s1=0 and s2=L-1, the transformation becomes a *thresholding function* that creates a binary image, as illustrated

in Fig. 3.2(b). Intermediate values of r1, s1 and r2, s2 produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.

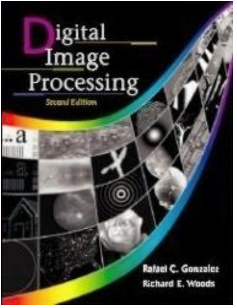
In general, r1 <= r2 and s1 <=s2 is assumed



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
- for all other gray levels
  - (a)transformation highlights range [A,B] of gray level and reduces all others to a
- constant level
  - (b)transformation highlights range [A,B] but preserves all other levels



Bit-plane slicing

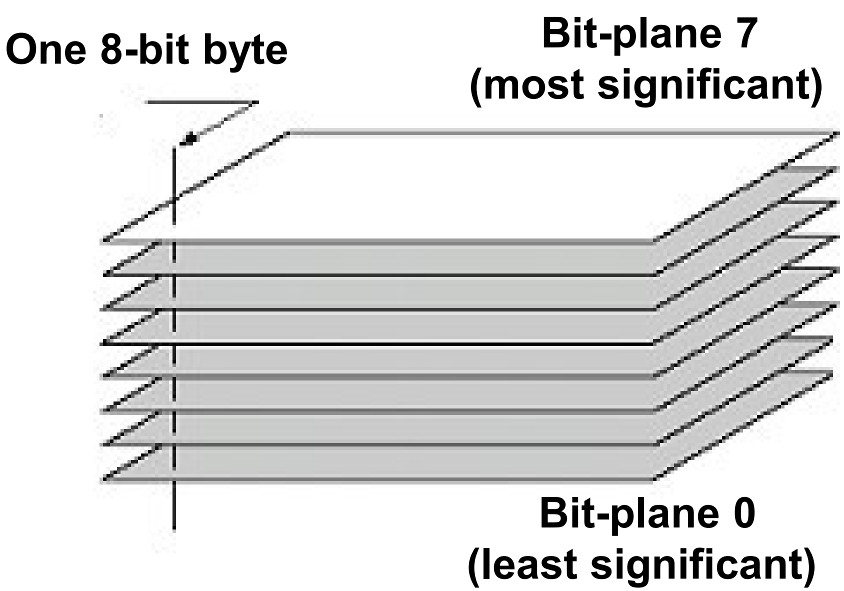
- Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits might be desired.
- Suppose that each pixel in an image is represented by 8 bits. Imagine that the image is composed of eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bit plane 7 for the most significant bit.
- In terms of 8-bit bytes, plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits

  Note that the higher-order bits (especially the top four) contain the majority of the visually significant data.The other bit planes contribute to more subtle details in the image
- Separating a digital image into its bit planes is useful for analyzing the relative

  importance played by each bit of the image, a process that aids in determining

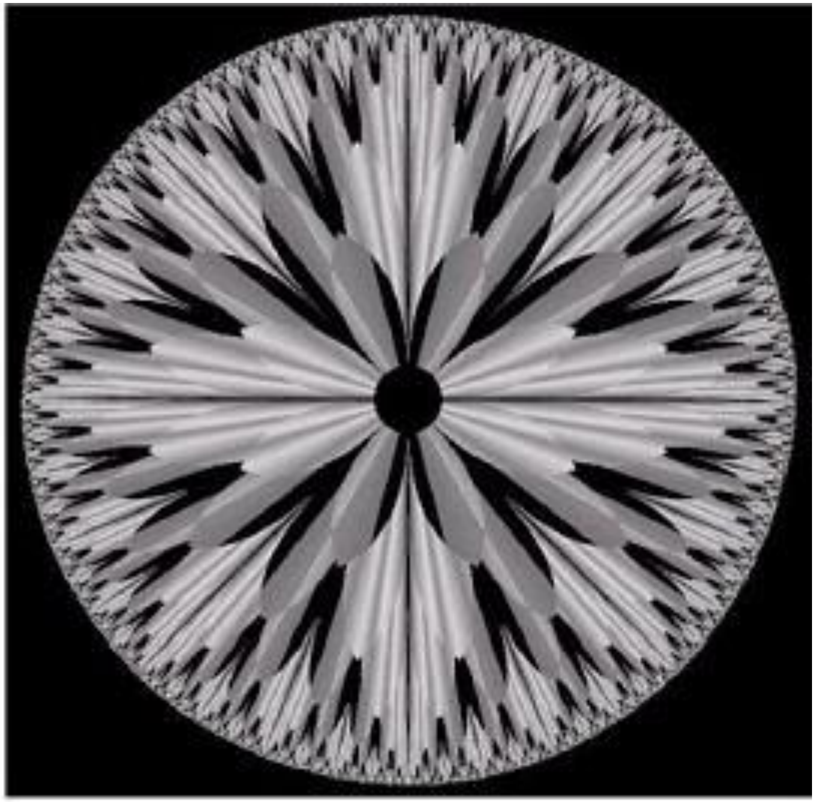
  the adequacy of the number of bits used to quantize each pixel

Bit-plane slicing



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

Example



An 8- bit fractal image

- The (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation.
  - Map all levels between 0 and 127 to 0
  - Map all levels between 129 and 255 to 255

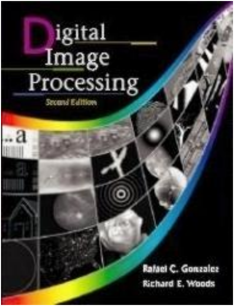
In **digital image processing**, especially for **grayscale images**, each pixel is typically represented using **8 bits** (values from 0 to 255). These 8 bits can be thought of as layers, or **"bit-planes"**, stacked on top of each other.

Each bit in a pixel contributes differently to its intensity:

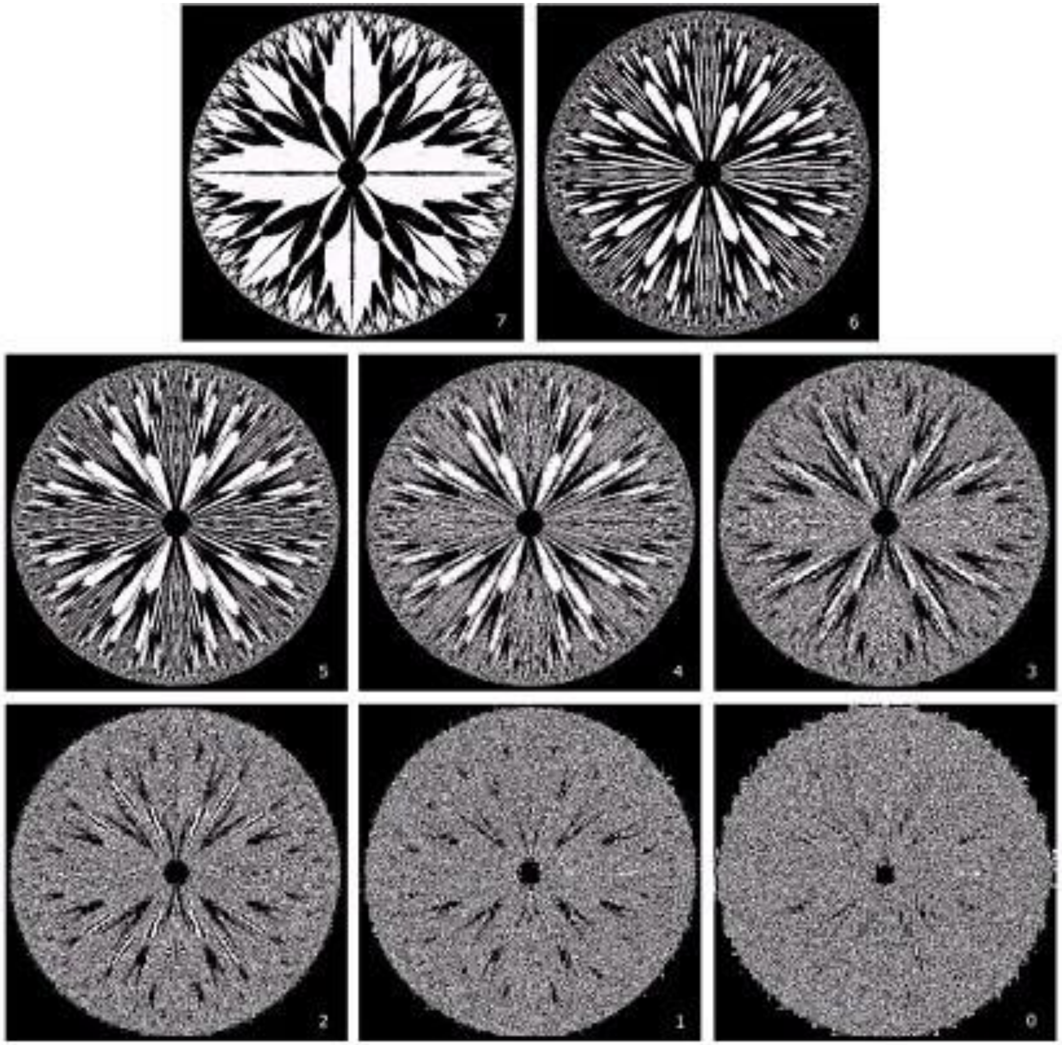
- The **Most Significant Bit (MSB)** — bit 7 — has the **highest weight**.
- The **Least Significant Bit (LSB)** — bit 0 — has the **lowest weight**.

Bit-plane slicing separates these individual bit layers to:

- Analyze **how each bit contributes** to the image.
- Enhance features in certain planes.
- Remove noise or hide secret data (e.g., in **steganography**).



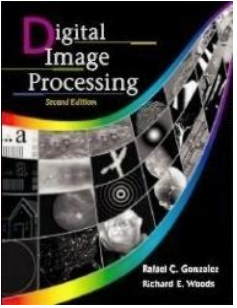
8 bit planes



Bit-plane 7 Bit-plane 6

Bit-plane 5 Bit-plane 4 Bit-plane 3

Bit-plane 2 Bit-plane 1 Bit-plane 0

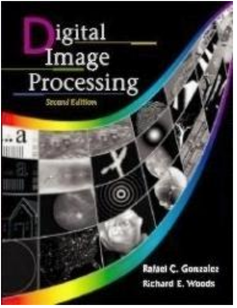


What each bit-plane shows:

Bit-Plane	Weight	Meaning
Bit 7	128	Most significant – rough structure or edges
Bit 6	64	Major features
Bit 5	32	Important textures
Bit 4	16	Less important features
Bit 3	8	Some fine detail
Bit 2	4	Tiny detail
Bit 1	2	Often random
Bit 0	1	Least significant – often looks like noise or used for hiding data

If you use only:

- **Bit 7 to Bit 4 → you'll get a lossy version, but it looks close to the original.**
- **Bit 3 to Bit 0 → you'll get a noisy, less useful image.**



What You Can Do After Extraction

1. Reconstruct the Image

2. Visualize and Study

You can display each plane as an image:

- Bit 7 image = high contrast, like edges or outlines
- Bit 0 image = often looks like noise

This helps in **pattern recognition, feature extraction**, etc.

3. Apply Compression

Since lower bits (bit 0–2) contribute less:

- You can **discard them** to reduce file size.
- This is similar to **bit-depth reduction** (e.g., 8-bit → 4-bit).

4. Hide Data (Steganography)

Low-order planes like **bit 0 or 1** are:

- **Not visually important**
- Good for **hiding secret messages or images** inside the image

Example: Replace bit 0 of the image with a message's bits.

Histogram Processing

- Used effectively for image enhancement
- Information inherent in histograms also is useful in image compression and segmentation

Histogram Processing

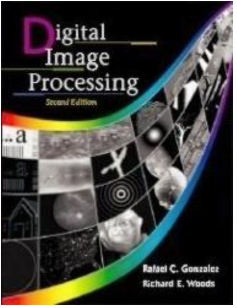
In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values.

This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image.

For an 8-bit grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values.

Histogram Processing

Histograms can also be taken of color images --- either individual histograms of red, green and blue channels can be taken, or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count.

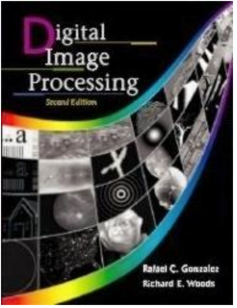


Histogram Processing

- 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 a digital image with gray levels r_k



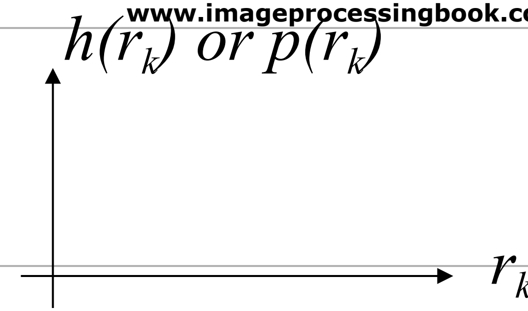
Normalized Histogram

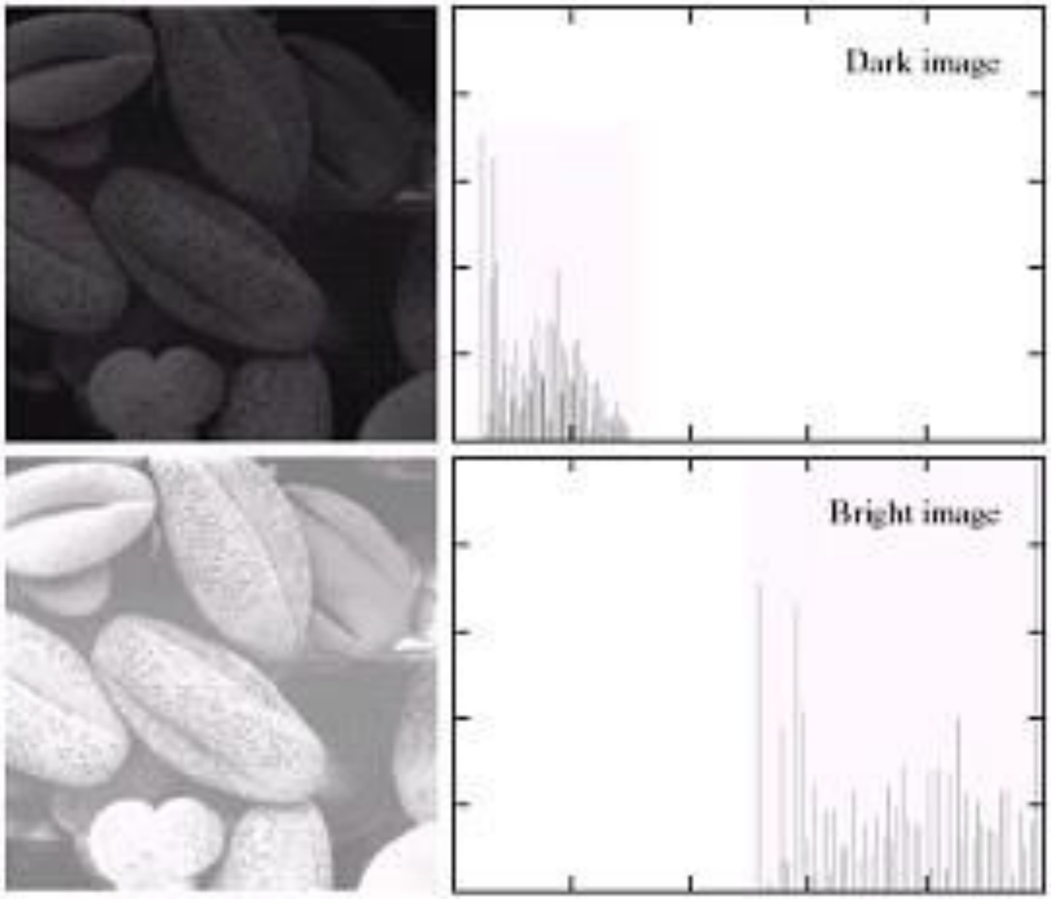
- dividing each of histogram at gray level r_k by the total number of pixels in the image, n

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

- 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

Example





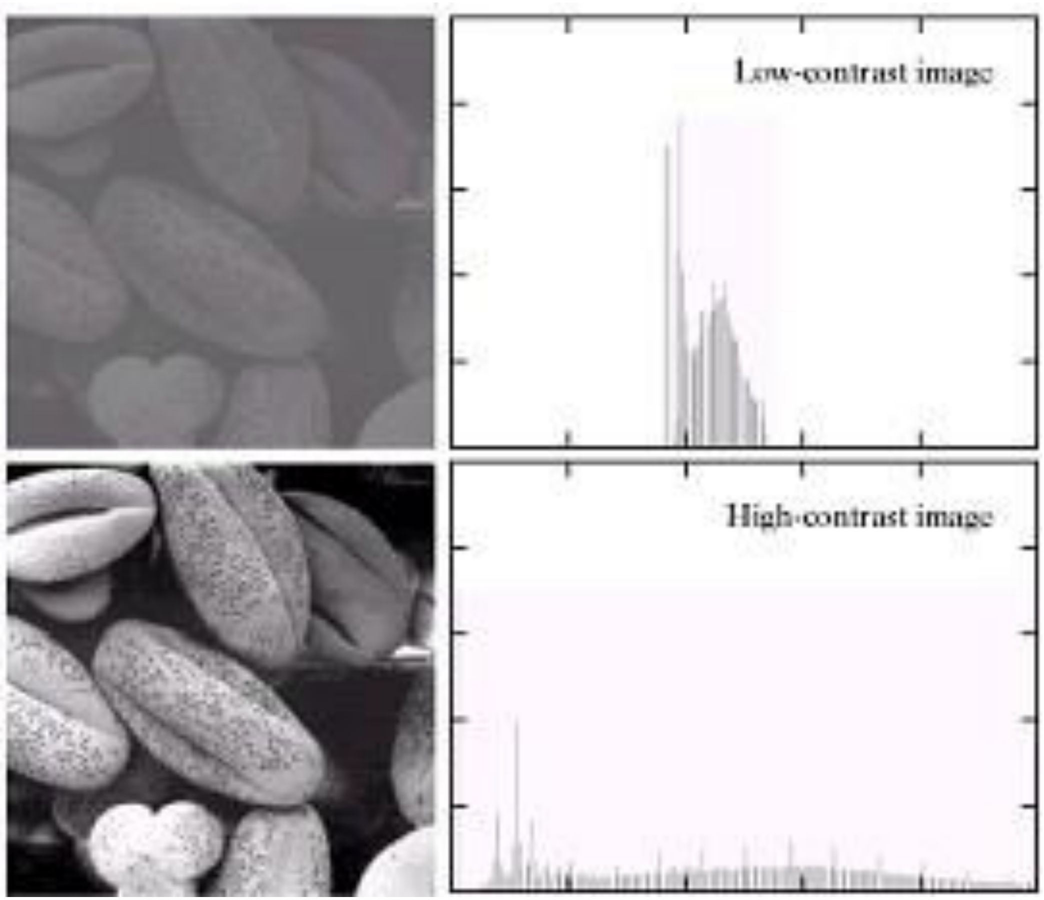
Dark image

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

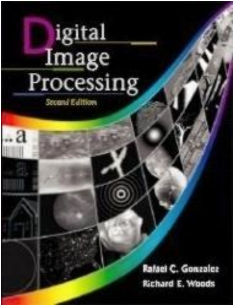


Low-contrast image

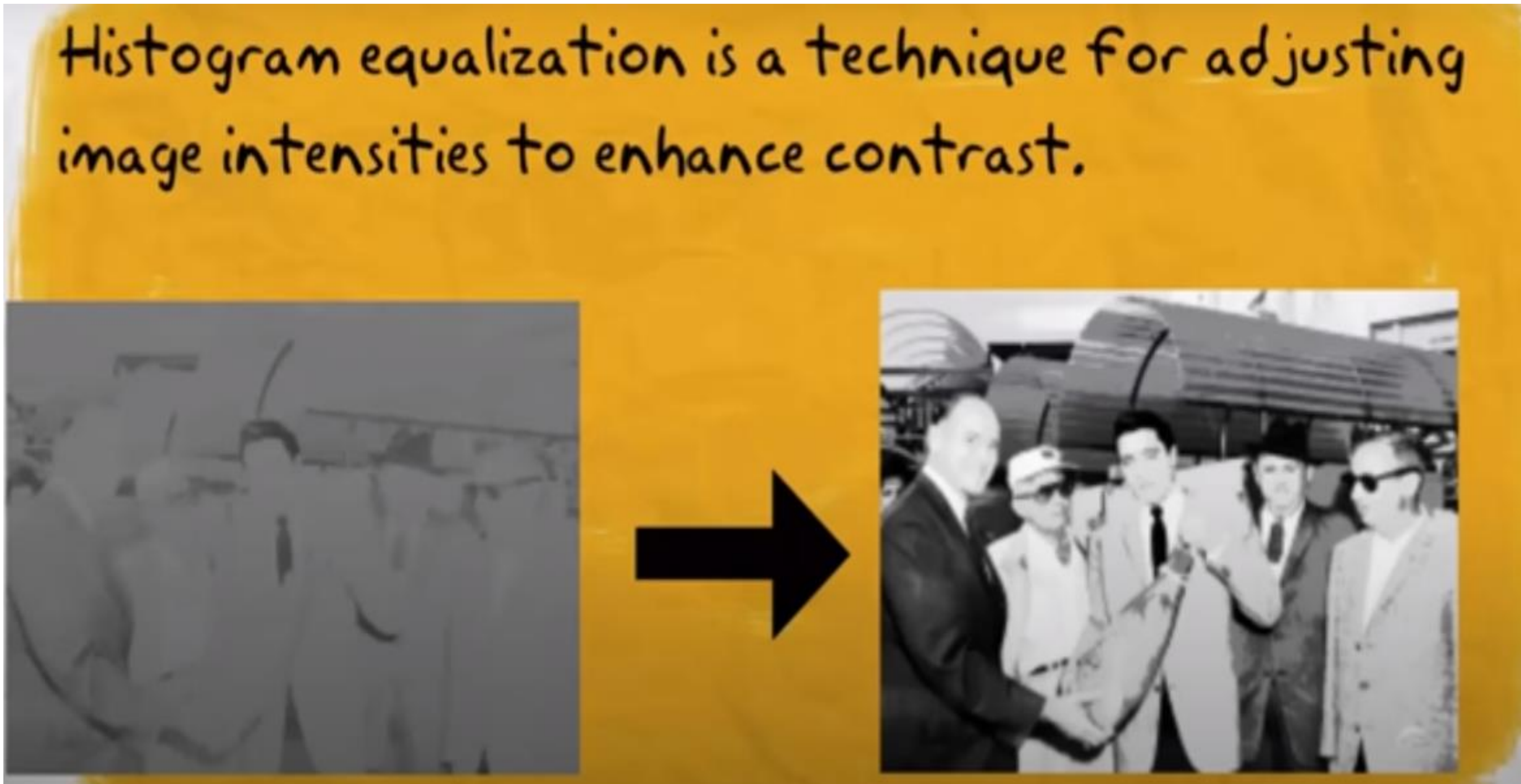
histogram is narrow and centered toward the middle of the gray scale

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

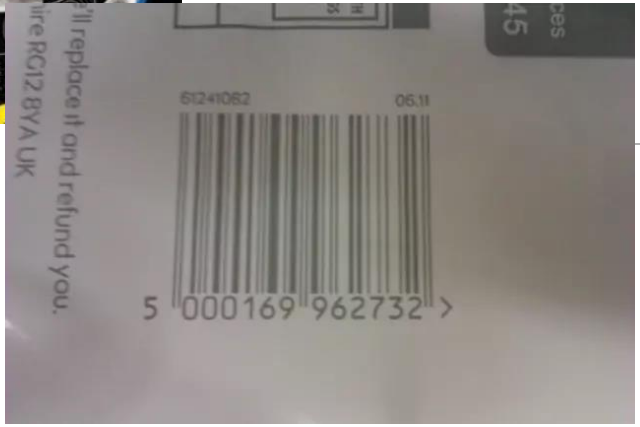


Histogram Equalization

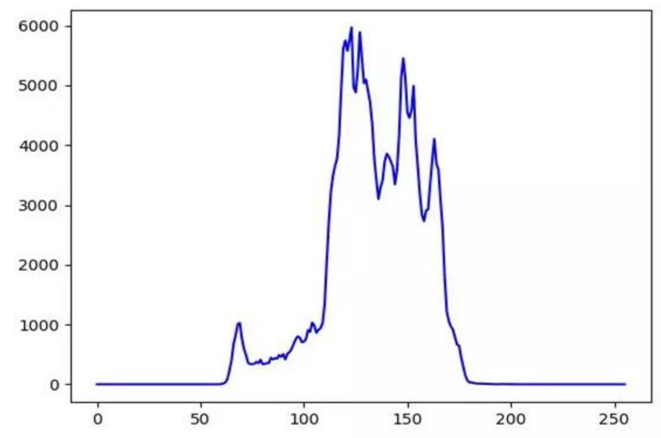


Histogram Equalization

- 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 of 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 equally

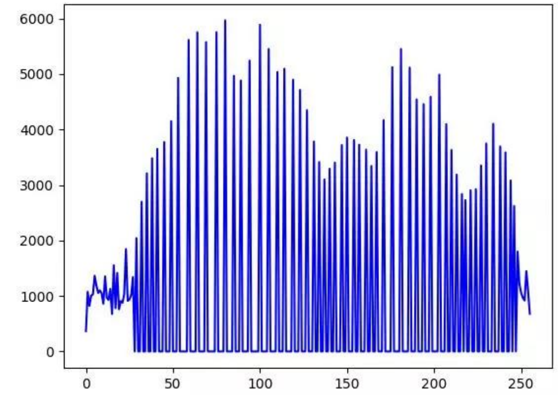


Before Histogram Equalization

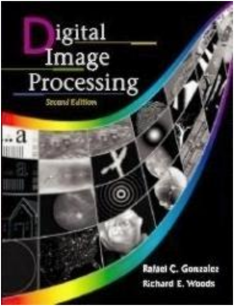


Histogram Chart Before Equalization

After Histogram Equalization



Histogram Chart After Equalization

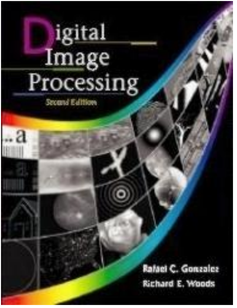


Histogram Matching (Specification)

- **Histogram Matching (also called Histogram Specification)** is a technique in **digital image processing** used to adjust the **histogram of one image** so that it **resembles the histogram of another ima**
- **What Is Histogram Matching?**
  - Histogram Matching is a process where the **input image** is transformed so that its histogram **matches** a specified **target histogram**.
  - Unlike histogram equalization (which stretches the histogram to cover the entire range), histogram matching **aims to shape** the histogram of the input image to **look like another image's histogram**

Histogram Matching (Specification)

- Equalize the levels of the original image.
- Specify the desired density function (histogram) and obtain the transformation function $G(z)$.
- Apply the inverse transformation function $z = G^{-1}(s)$ to the levels obtained in first step.

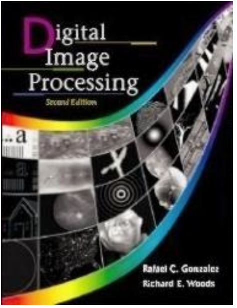


Spatial Filtering Process

- 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 relationship.

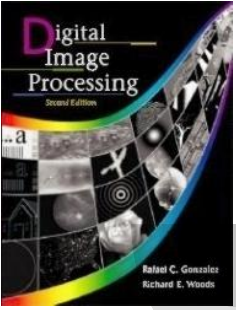
$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn}$$

$$= \sum_{i=i}^{mn} w_i z_i$$

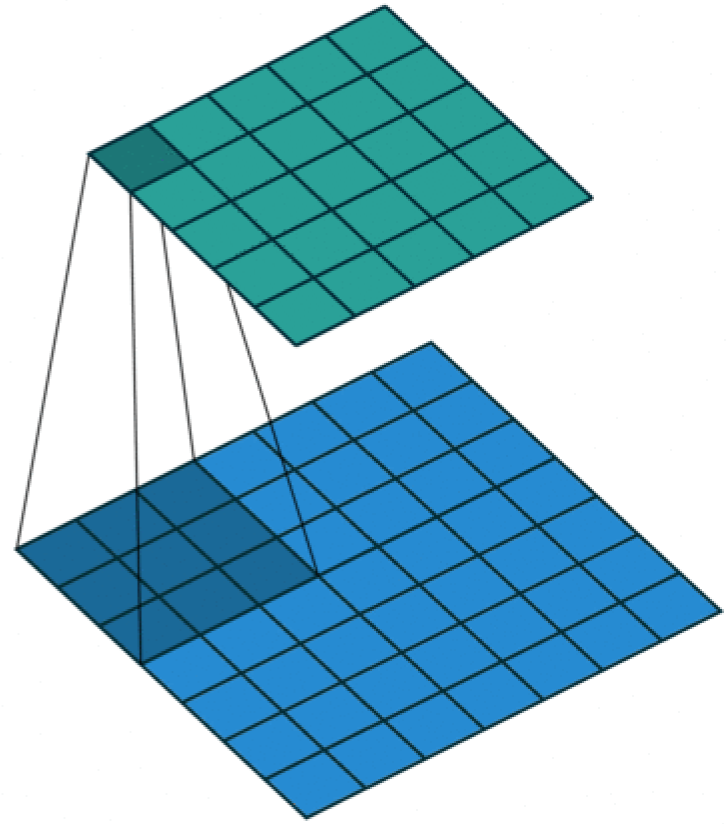


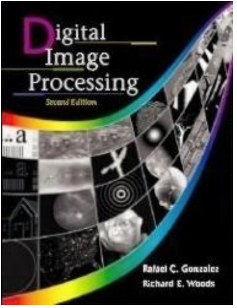
Smoothing Spatial Filters

- 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



- In the this image, the blue matrix is the input and the green matrix is the output. Whereas we have a kernel moving through the input matrix to get/extract the features.





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.

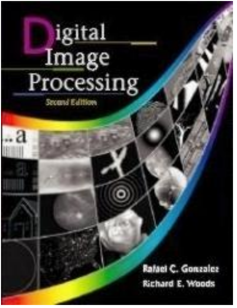
Smoothing Linear Filters

- Replacing the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the "sharp" transitions in gray levels.
- sharp transitions
  - random noise in the image
  - edges of objects in the image
- thus, smoothing can reduce noises (desirable) and blur edges (undesirable)

General Spatial Filter

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9



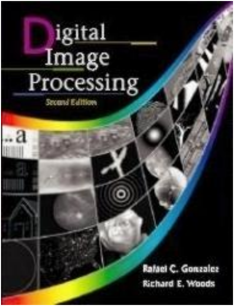
3x3 Smoothing Linear Filters

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

box filter

weighted average

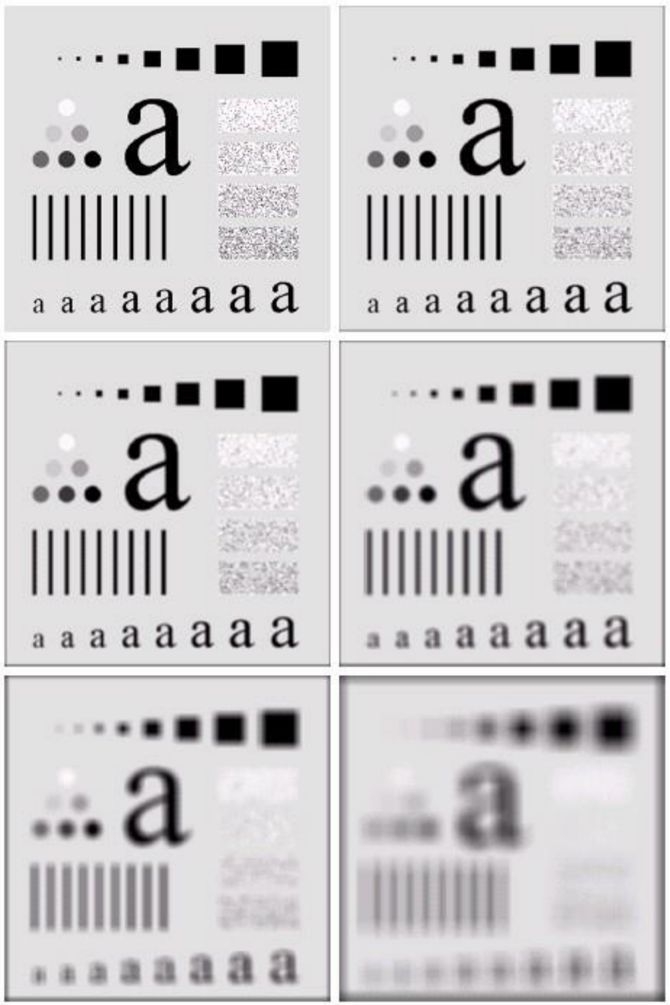
the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask



Weighted average filter

- the basic strategy behind weighting the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process.

Example

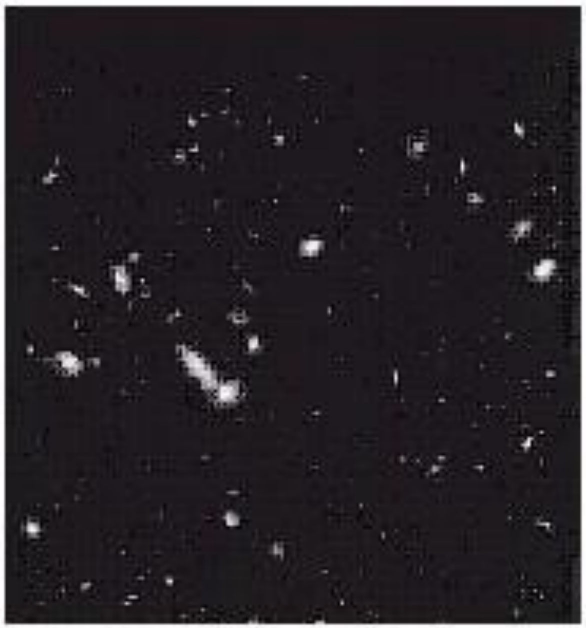


- a) original image 500x500 pixel
- b) - f). results of smoothing with square averaging filter masks of size n = 3, 5, 9, 15 and
- 35, respectively.

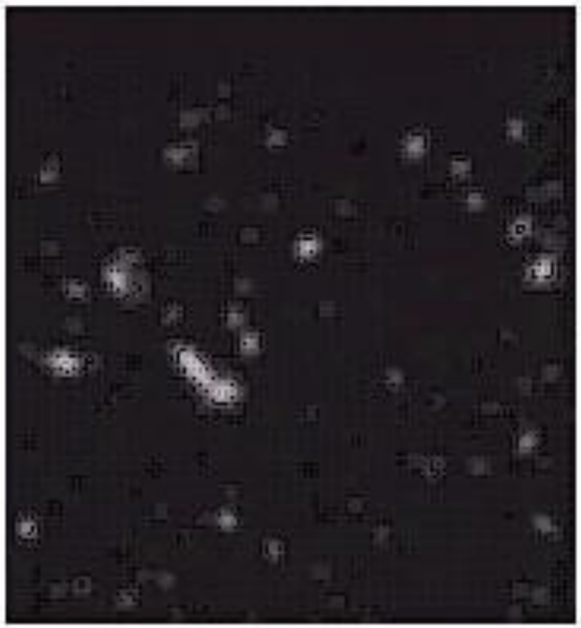
Note:

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

Example



original image



result after smoothing with 15x15 averaging mask

result of thresholding

we can see that the result after smoothing and thresholding, the remains are the largest and brightest objects in the image.

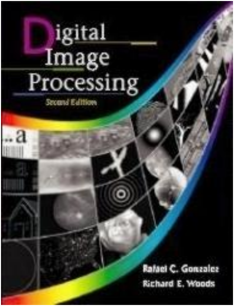
- For image denoising (and other applications), the blurring associated with linear filters is undesired.
- Moreover, linear filters are ineffective to remove some types of noise; e.g., impulsive noise.

- Solution: use nonlinear filters.

Order Statistics Filter:

- **Order Statistics Filter:**
  It is based on the ordering the pixels contained in the image area encompassed by the filter.

- It replaces the value of the center pixel with the value determined by the ranking result. Edges are better preserved in this filtering

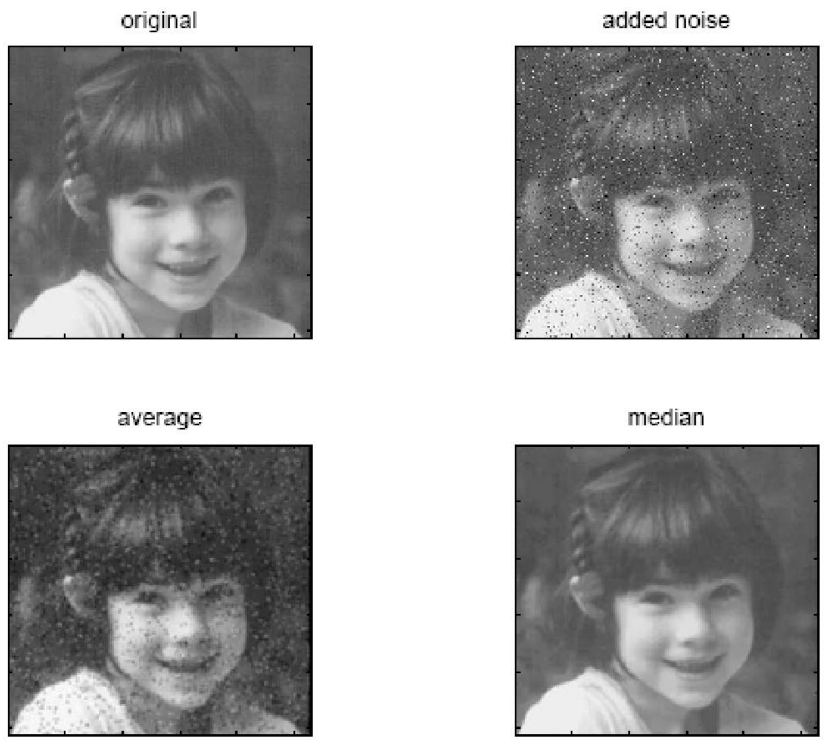


Order Statistics Filter:

- **Types of Order statistics filter:**
- **(i) Minimum filter:** 0th percentile filter is the minimum filter. The value of the center is replaced by the smallest value in the window.
- **(ii) Maximum filter:** 100th percentile filter is the maximum filter. The value of the center is replaced by the largest value in the window.
- **(iii) Median filter:** Each pixel in the image is considered. First neighboring pixels are sorted and original values of the pixel is replaced by the median of the list.

Non-linear Filter

- Median filtering (nonlinear)
  - Used primarily for noise reduction (eliminates isolated spikes)
  - The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).


original
added noise
average
median

Median Filters

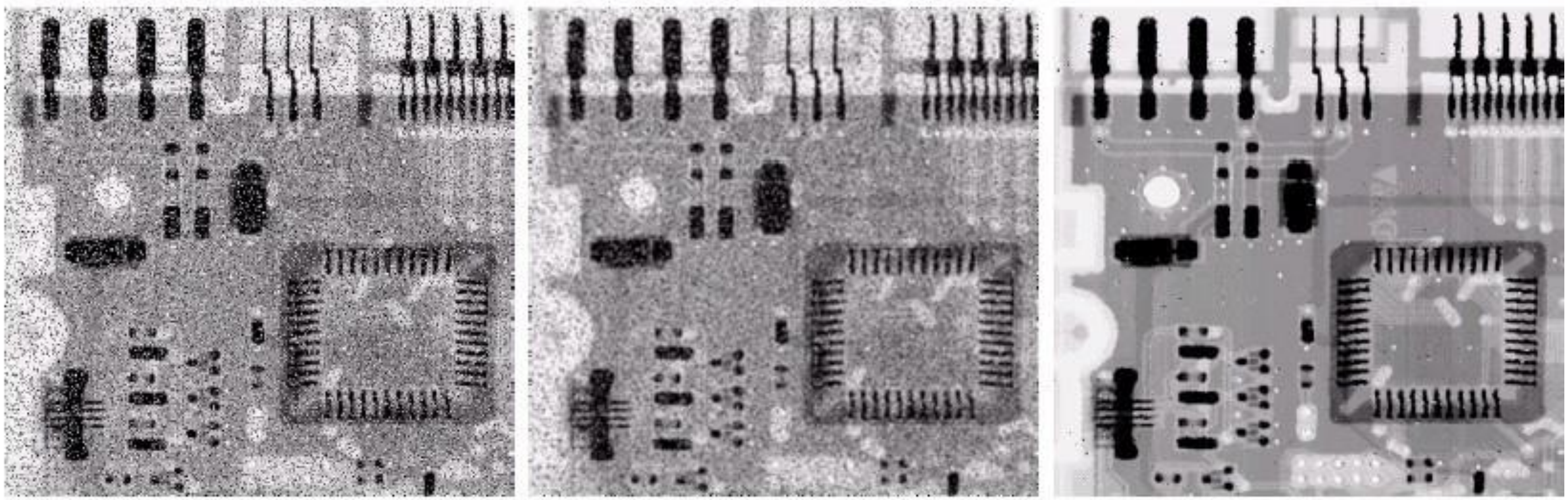
- replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median)
- quite popular because for certain types of random noise (impulse noise □ salt and pepper noise) , they provide excellent noise-reduction capabilities, with considering less blurring than linear smoothing filters of similar size.

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.
- eliminated = forced to have the value equal the median intensity of the neighbors.

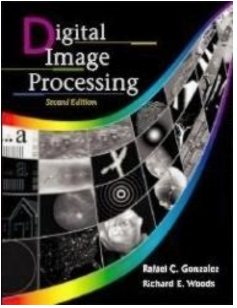
larger clusters are affected considerably less

Example : Median Filters



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

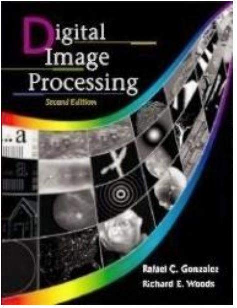


Sharpening Spatial Filters

- to highlight fine detail in an image
- or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.
- **Sharpening Spatial Filter:** It is also known as derivative filter. The purpose of the sharpening spatial filter is just the opposite of the smoothing spatial filter. Its main focus in on the removal of blurring and highlight the edges. It is based on the first and second order derivative.

Sharpening Spatial Filters

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

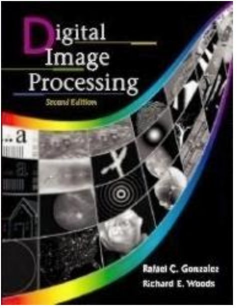


Use of Second Derivatives for Enhancement The Laplacian

- To sharpen an image, the Laplacian of the image is subtracted from the original image.

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

- Example: Figure 3.40



Use of Second Derivatives for Enhancement
The Laplacian

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

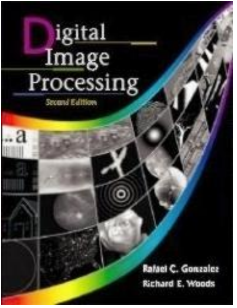
0	−1	0
−1	4	−1
0	−1	0

(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.

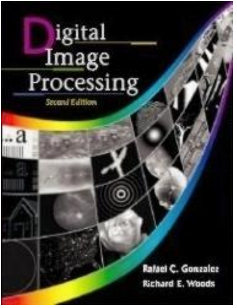


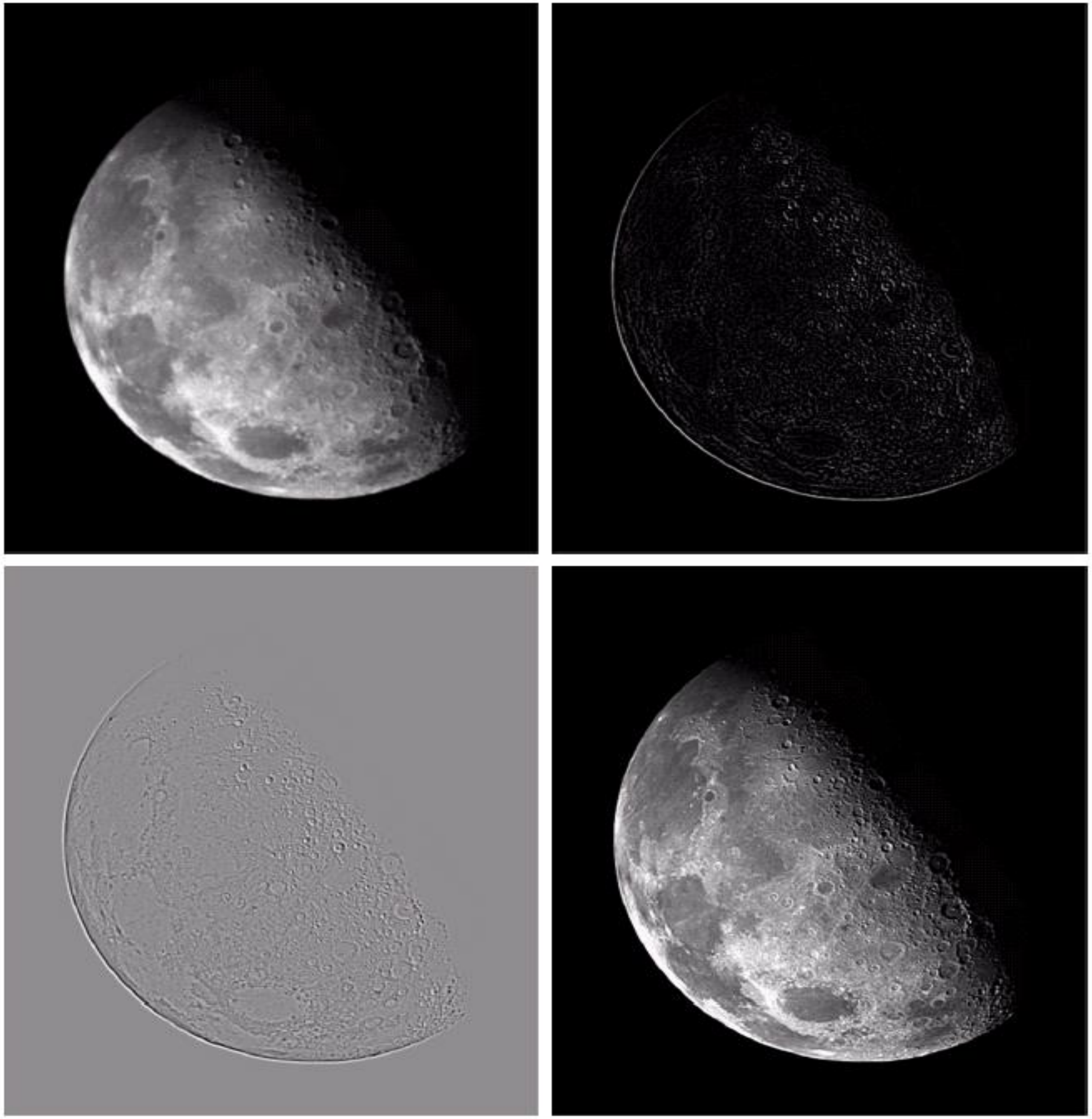
Use of Second Derivatives for Enhancement
The Laplacian

- To sharpen an image, the Laplacian of the image is subtracted from the original image.

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

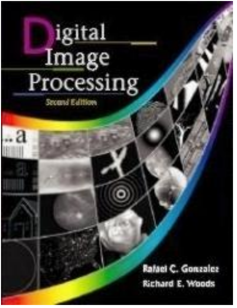
- Example: Figure 3.40





a	b
c	d

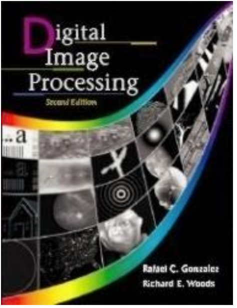
FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)



Use of First Derivatives for Enhancement
The Gradient

- Development of the Gradient method
  - The gradient of function *f* at coordinates (*x*,*y*) is defined as the two-dimensional column vector:

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



Use of First Derivatives for Enhancement
The Gradient

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts cross-gradient operators

−1	−2	−1
0	0	0
1	2	1

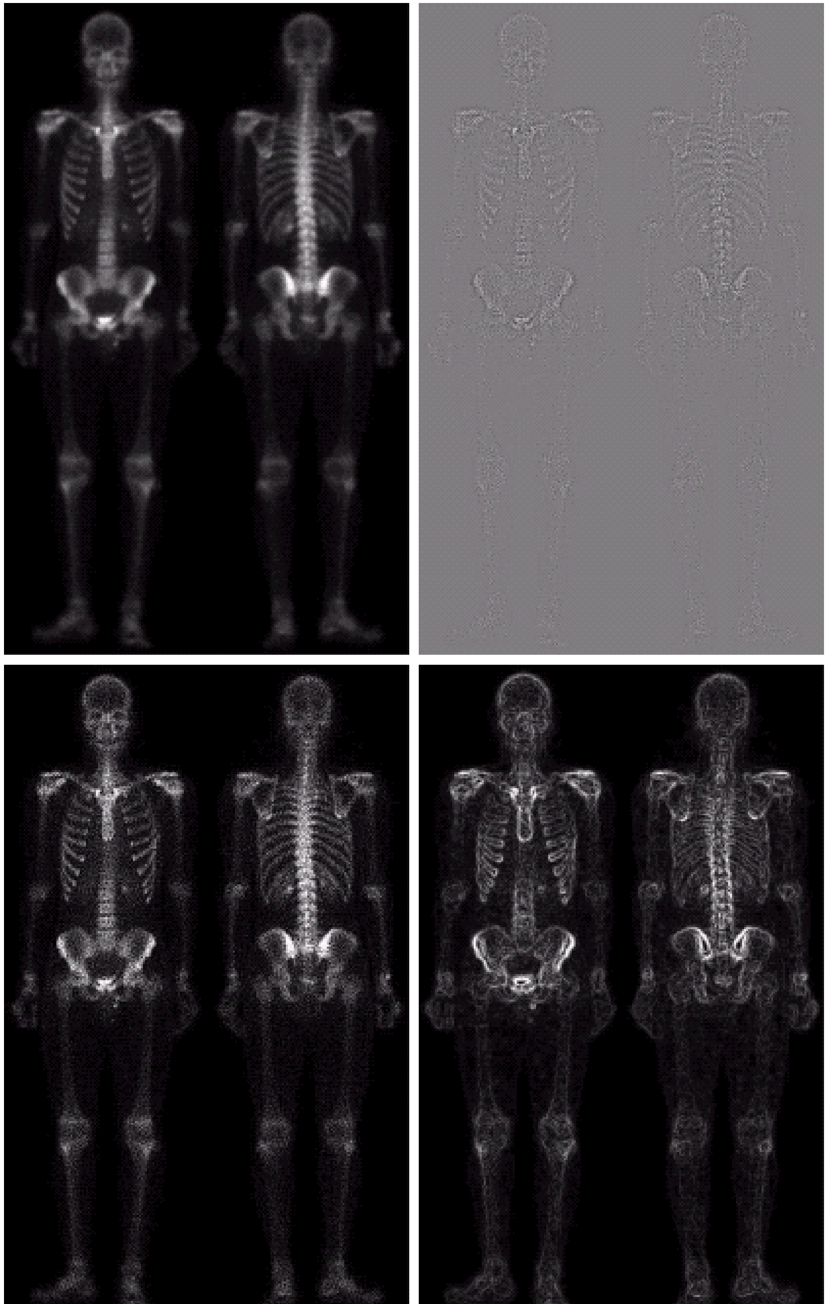
−1	0	1
−2	0	2
−1	0	1

Sobel operators

a
b c
d e

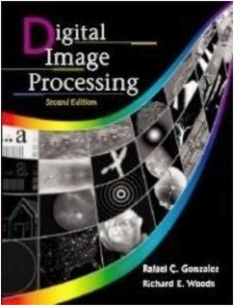
FIGURE 3.44 A 3 × 3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

Combining Spatial Enhancement Methods

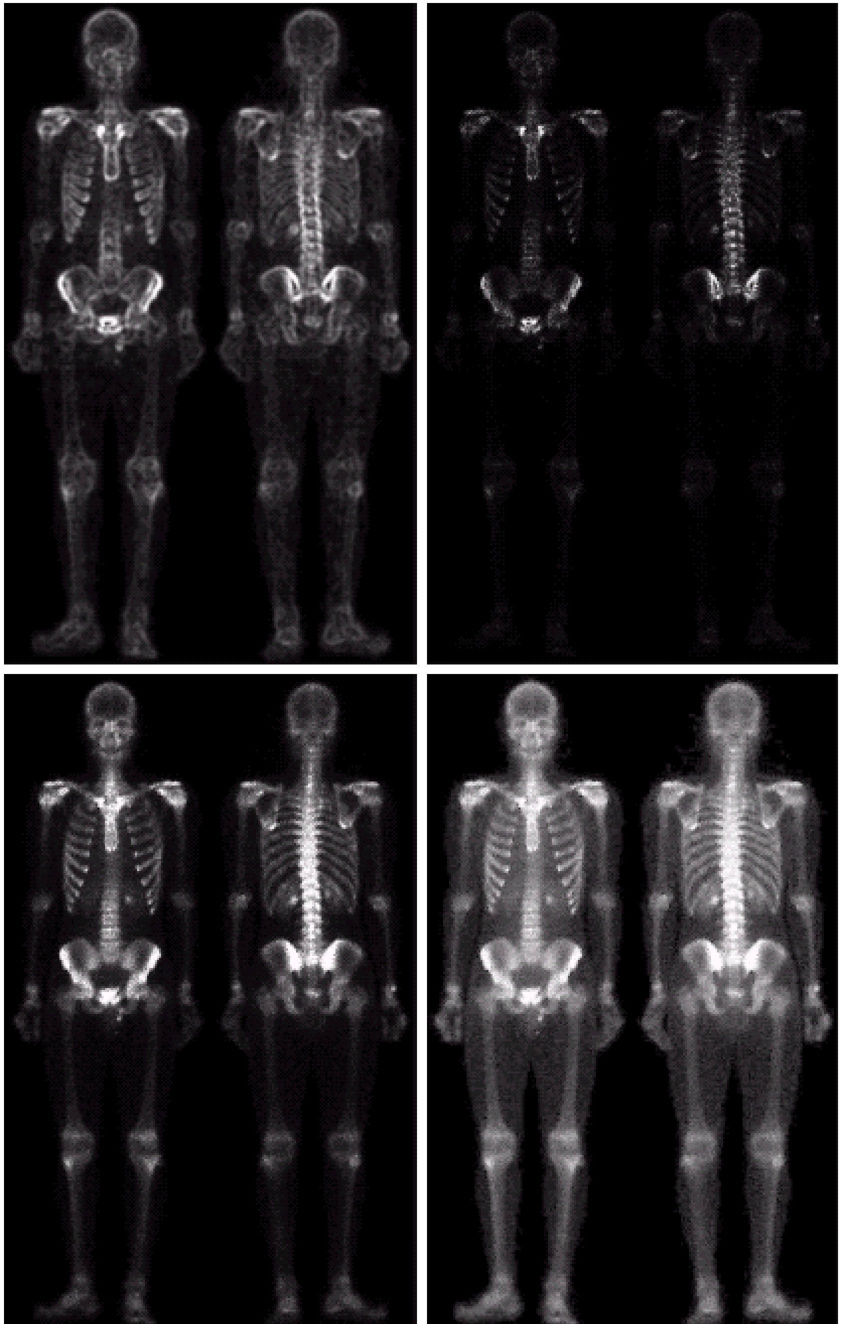


a b
c d

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

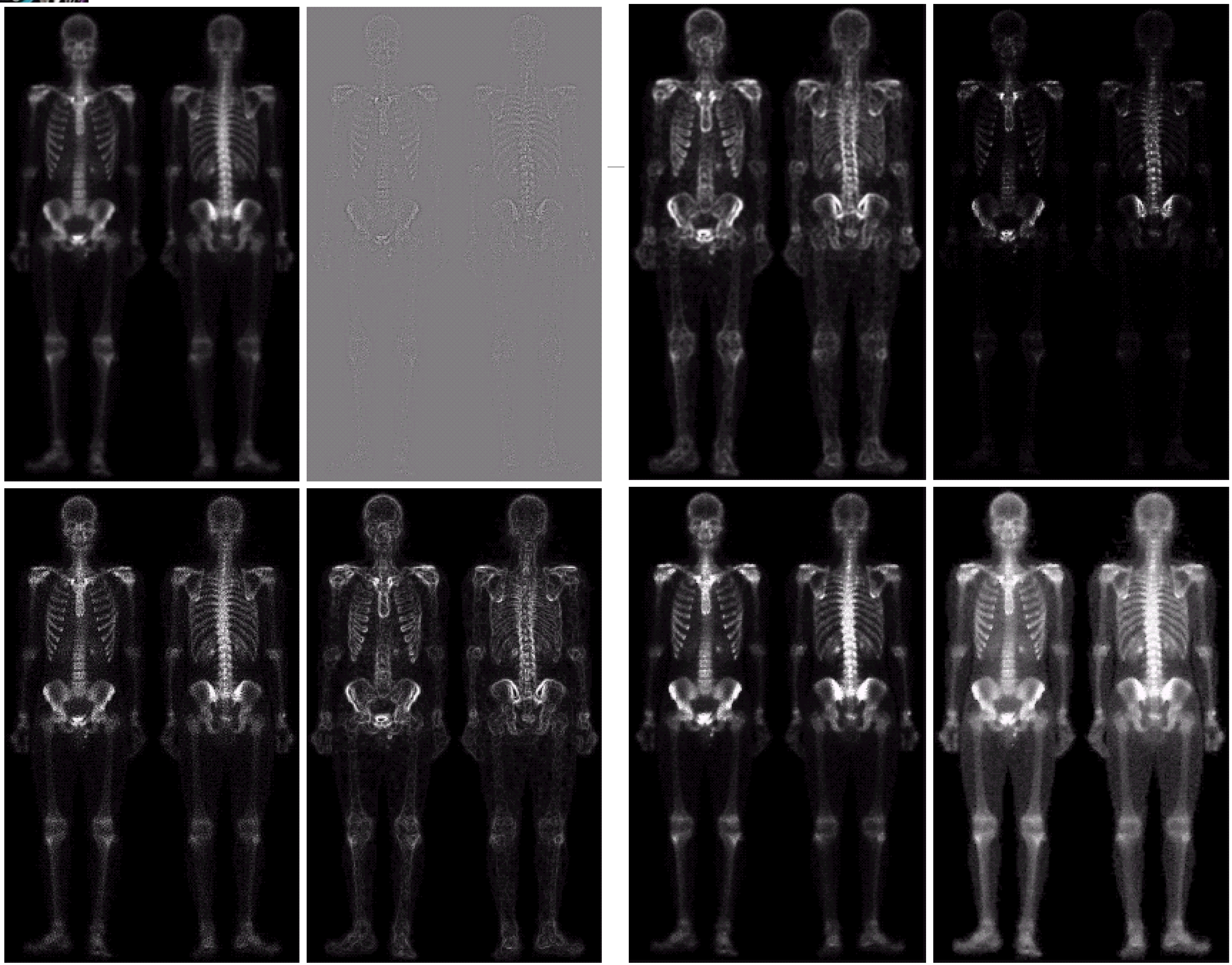


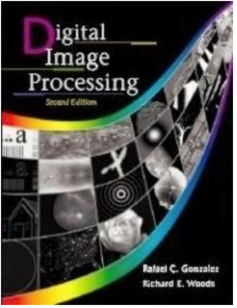
Combining Spatial Enhancement Methods



e	f
g	h

FIGURE 3.46 *(Continued)* (e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)





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.

Thank You