PPTs ON
DIGITAL IMAGE PROCESSING

B.Tech VII semester

(Autonomous R16) (2019-20)

Department of Electronics and Communication Engineering

By

Dr.S.China Venkateswarlu,Professor
Ms.M.Saritha, Assistant Professor
Mr. D.Khalandar Basha, Assistant Professor
Mr. B.Santhosh Kumar, Assistant Professor

Presentation
for
UNIT-I

INTRODUCTION

INTRODUCTION

- Image – A two-dimensional signal that can be observed by human visual system

- Digital image – Representation of images by sampling in time and space.

- Digital image processing – perform digital signal processing operations on digital images

DIGITAL IMAGE FUNDAMENTALS & IMAGE TRANSFORMS

- The field of digital image processing refers to processing digital images by means of a digital computer.

- An image may be defined as a two- dimensional function, f(x,y) where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point.

- When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image

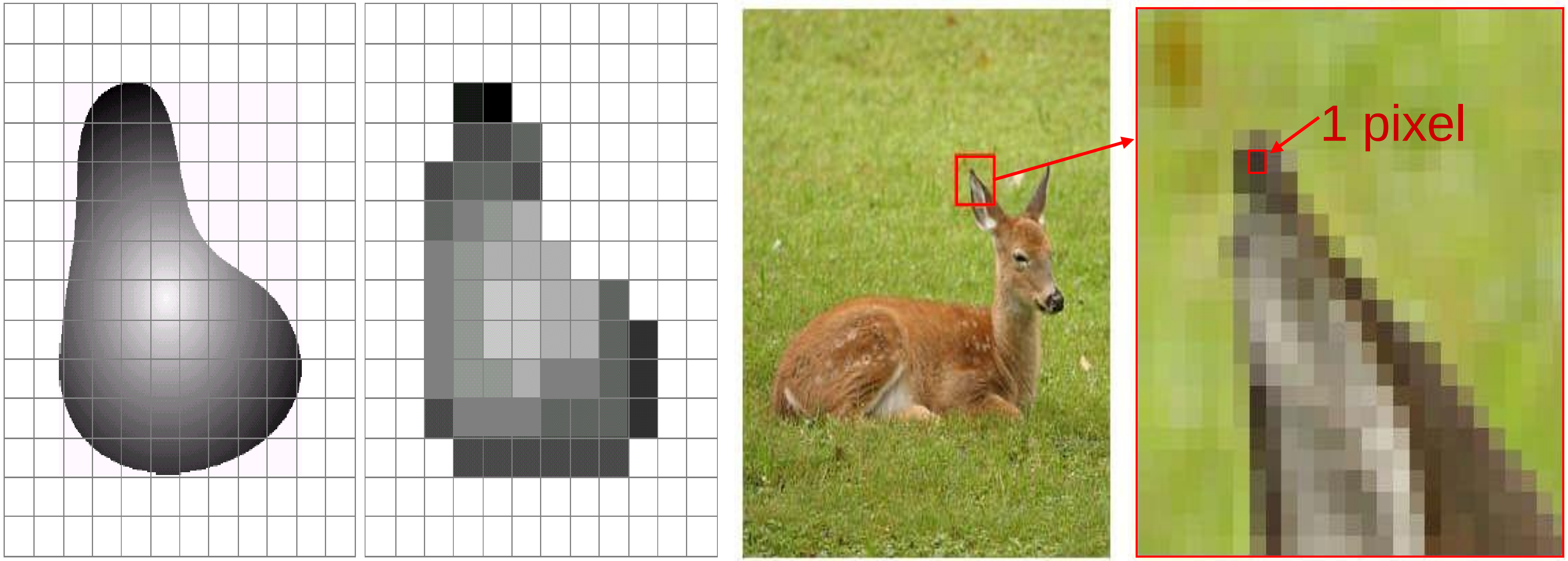
The Origins of Digital Image Processing

- One of the first applications of digital images was in the newspaper industry, when pictures were first sent by submarine cable between London and New York.

- Specialized printing equipment coded pictures for cable transmission and then reconstructed them at the receiving end.

What is a Digital Image?

- Pixel values typically represent gray levels, colours, heights, opacities etc
- Remember digitization implies that a digital image is an approximation of a real scene



What is a Digital Image?

Common image formats include:

- 1 sample per point (B&W or Grayscale)
- 3 samples per point (Red, Green, and Blue)
- 4 samples per point (Red, Green, Blue, and “Alpha”, a.k.a. Opacity)

What is Digital Image Processing

- Digital image processing focuses on two major tasks
  - Improvement of pictorial information for human interpretation
  - Processing of image data for storage, transmission and representation for autonomous machine perception.
- Some argument about where image processing ends and fields such as image analysis and computer vision start

Applications of DIP

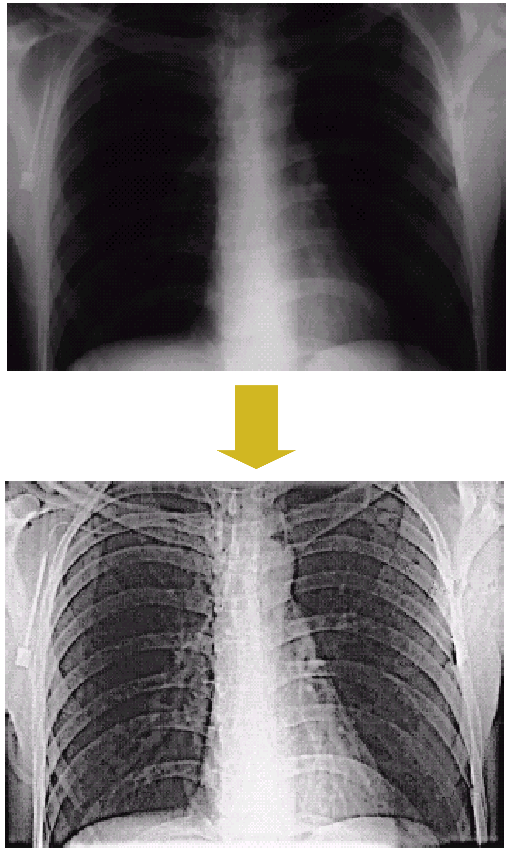
- The field of image processing has applications medicine and the space program.

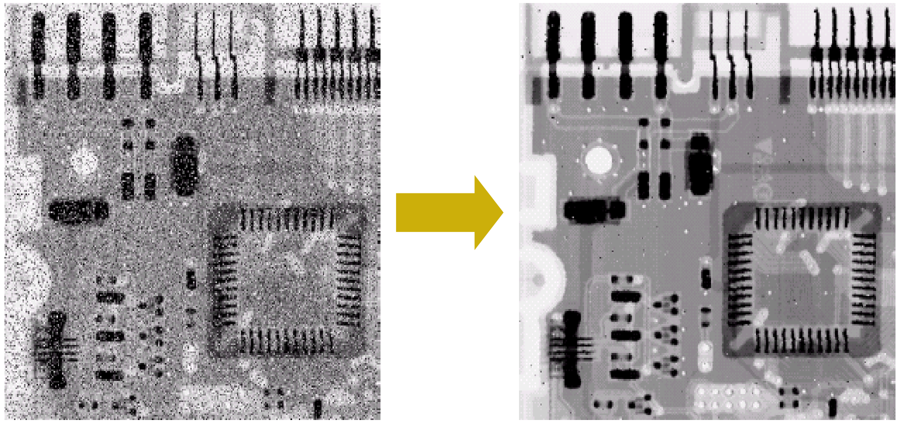
- Computer procedures are used to enhance the contrast or code the intensity levels into color for easier interpretation of X-rays and other images used in industry, medicine, and the biological sciences

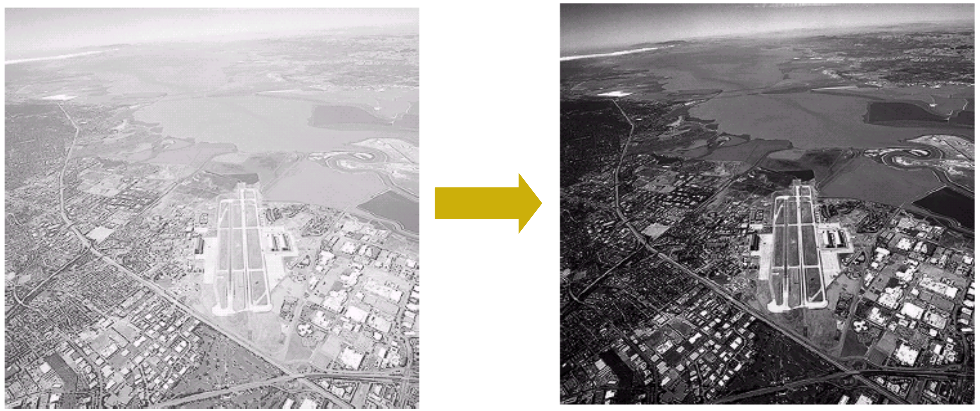
- Geographers use the same or similar techniques to study pollution patterns from aerial and satellite imagery

Applications of DIP

One of the most common uses of DIP techniques: improve quality, remove noise etc

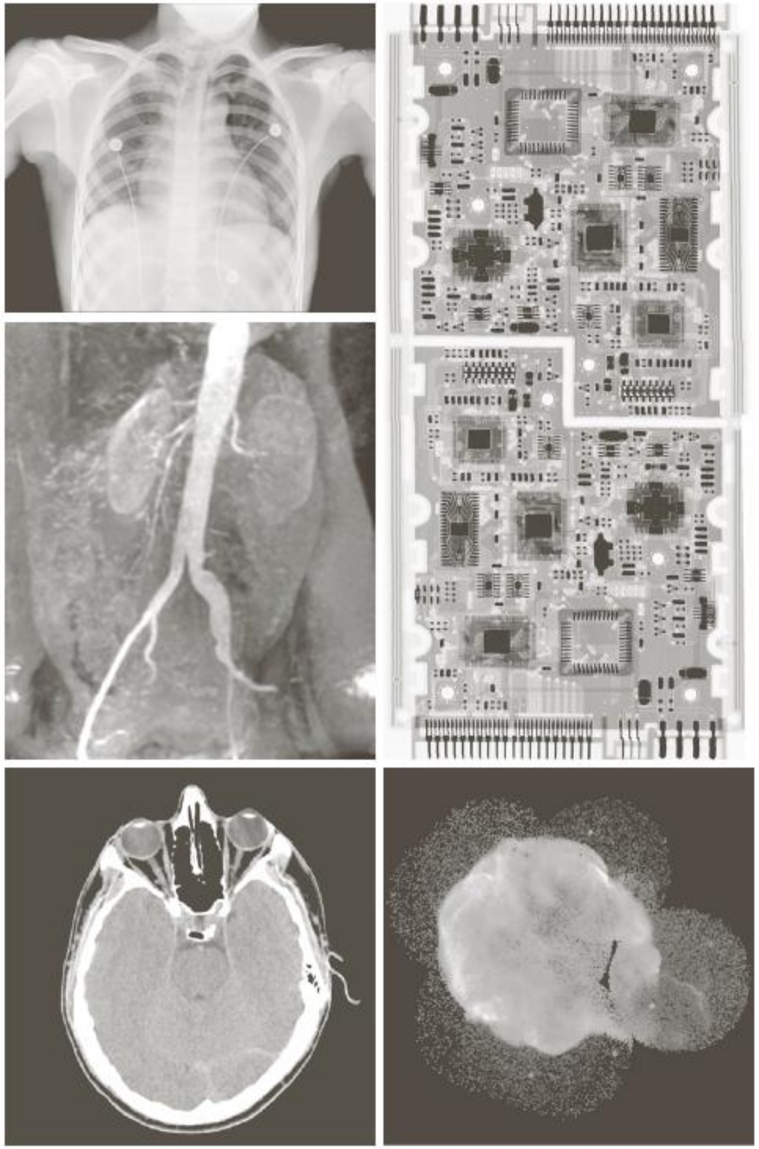






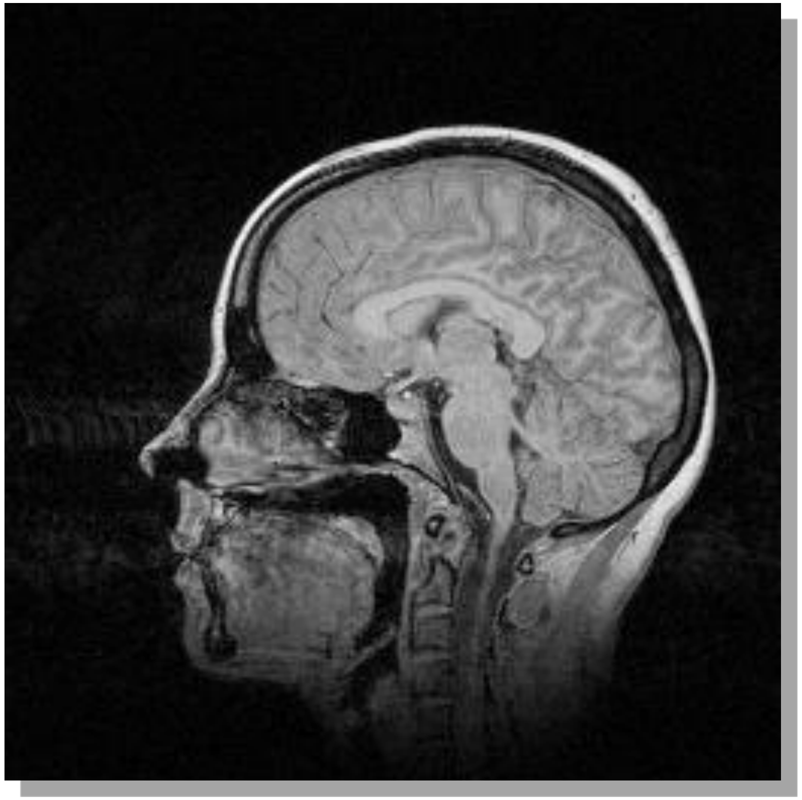
Applications: Medicine

X-ray imaging



Applications: Medicine

- Radio frequencies
- Magnetic Resonance Imaging (MRI)



Key Stages in Digital Image Processing

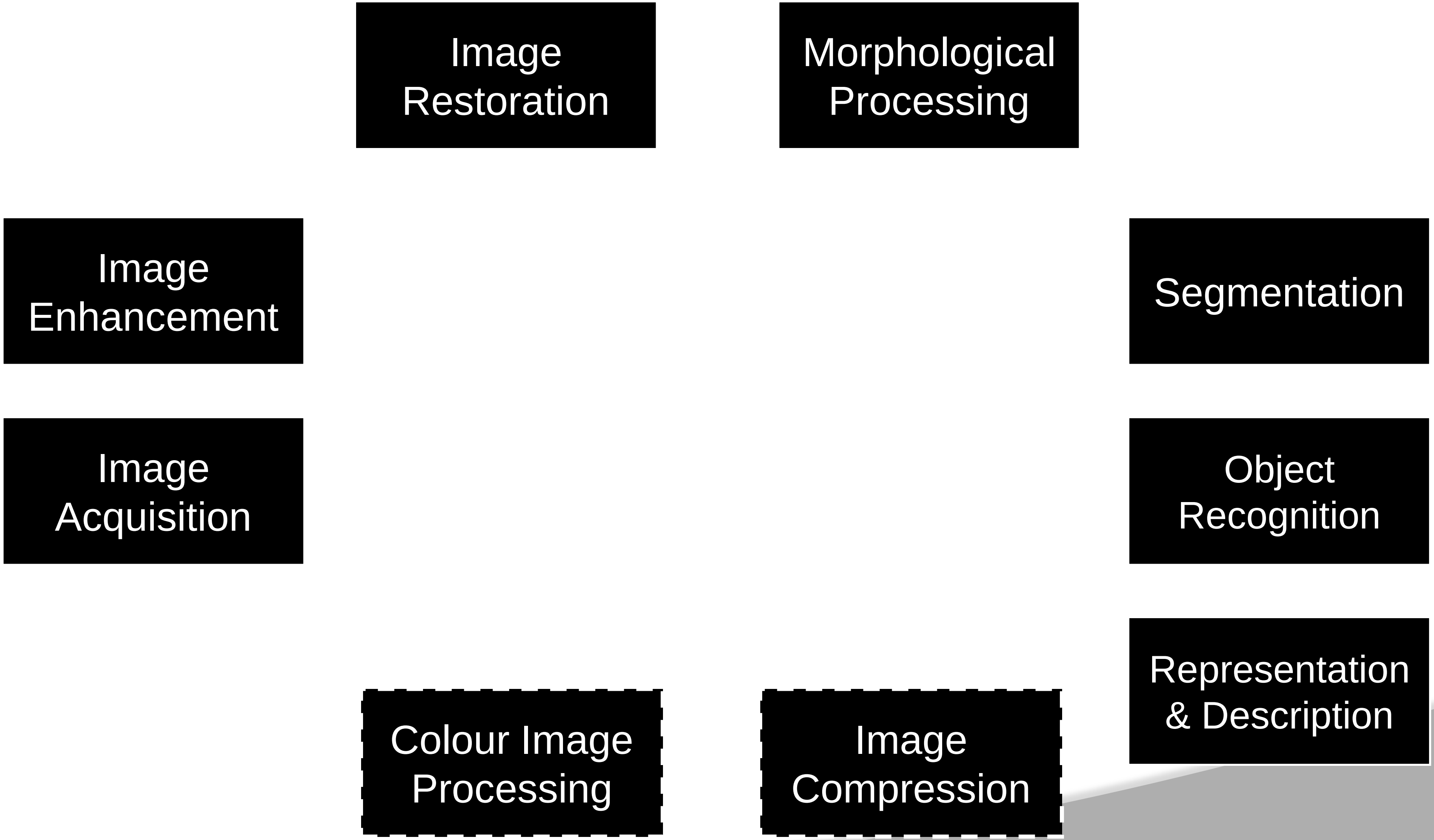


Image Restoration

Morphological Processing

Image Enhancement

Segmentation

Image Acquisition

Object Recognition

Representation & Description

Colour Image Processing

Image Compression

Key Stages in Digital Image Processing:Image Aquisition

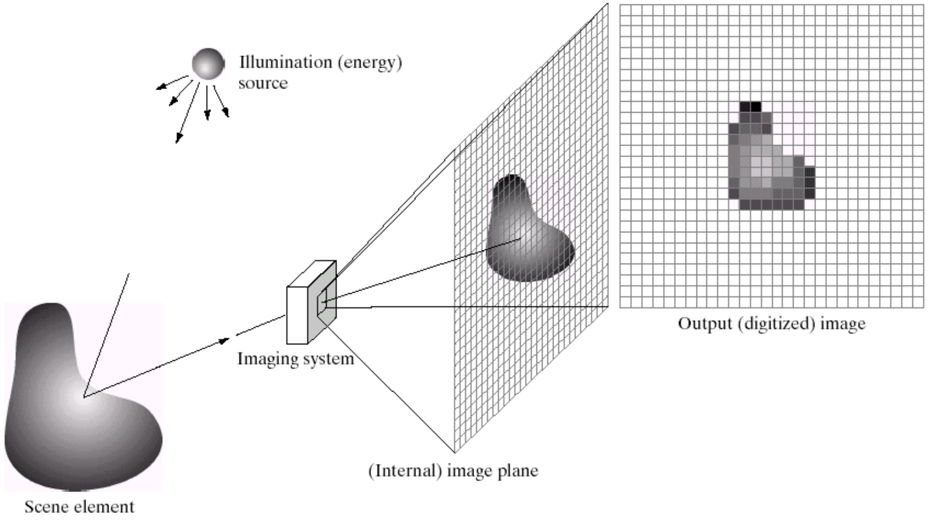
Image Restoration

Morphological Processing

Image Enhancement

Segmentation

Image Acquisition



Object Recognition

Representation & Description

Colour Image Processing

Image Compression

Key Stages in Digital Image Processing: Image Enhancement

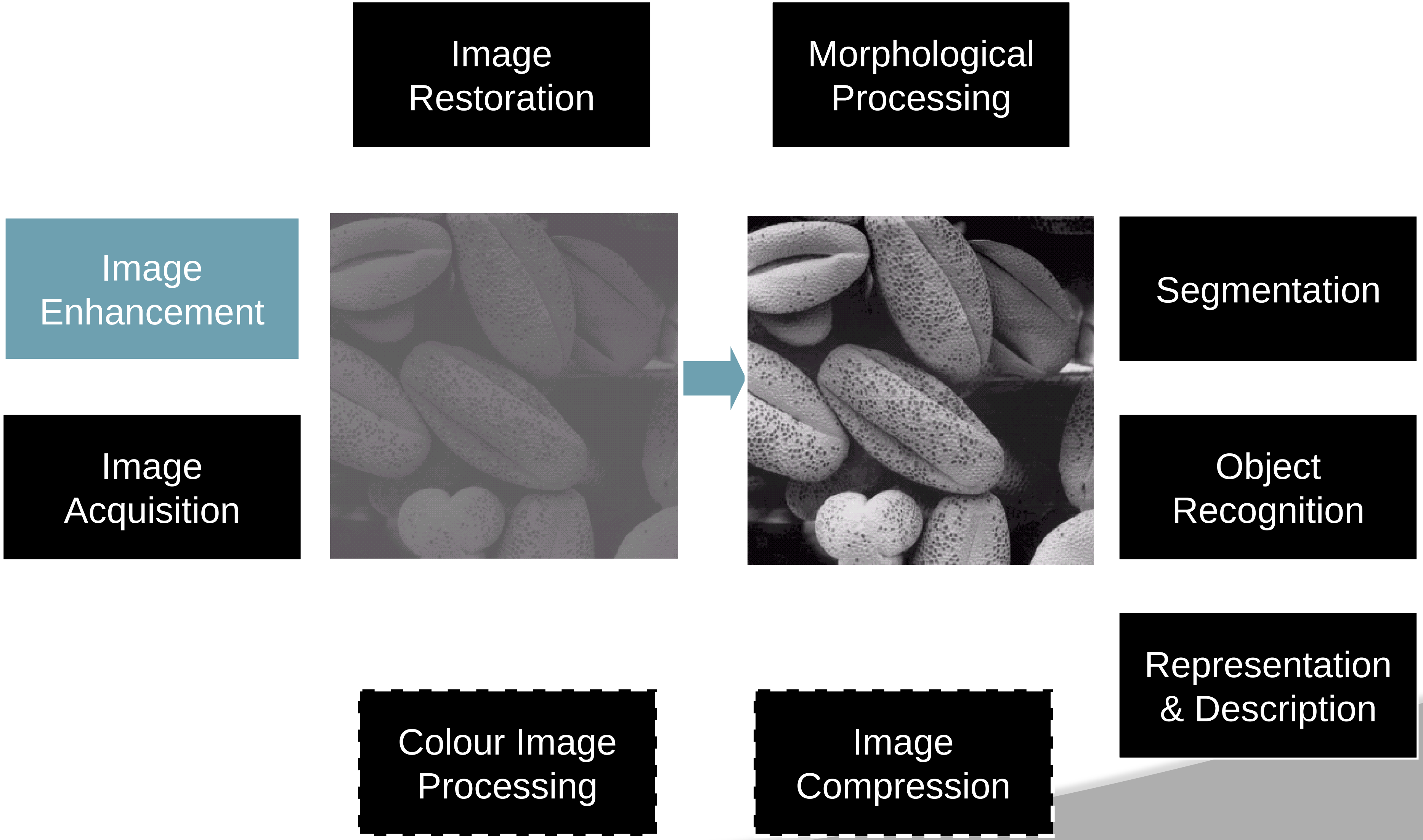


Image Restoration

Morphological Processing

Image Enhancement

Segmentation

Image Acquisition

Object Recognition

Representation & Description

Colour Image Processing

Image Compression

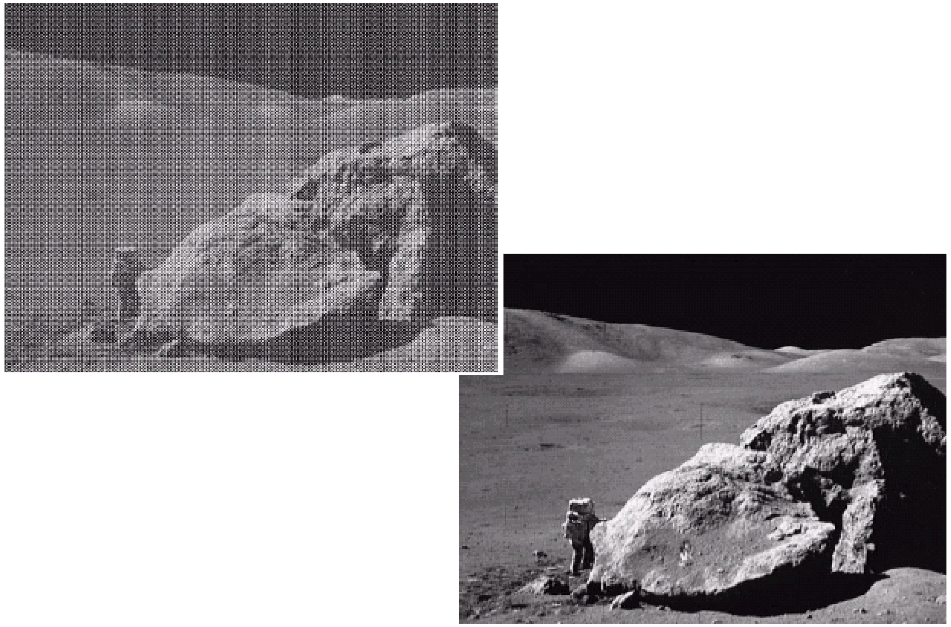
Key Stages in Digital Image Processing: Image Restoration

Image Restoration

Morphological Processing

Image Enhancement

Image Acquisition



Segmentation

Object Recognition

Representation & Description

Colour Image Processing

Image Compression

Key Stages in Digital Image Processing: Morphological Processing

Image Restoration

Morphological Processing

Image Enhancement

Segmentation

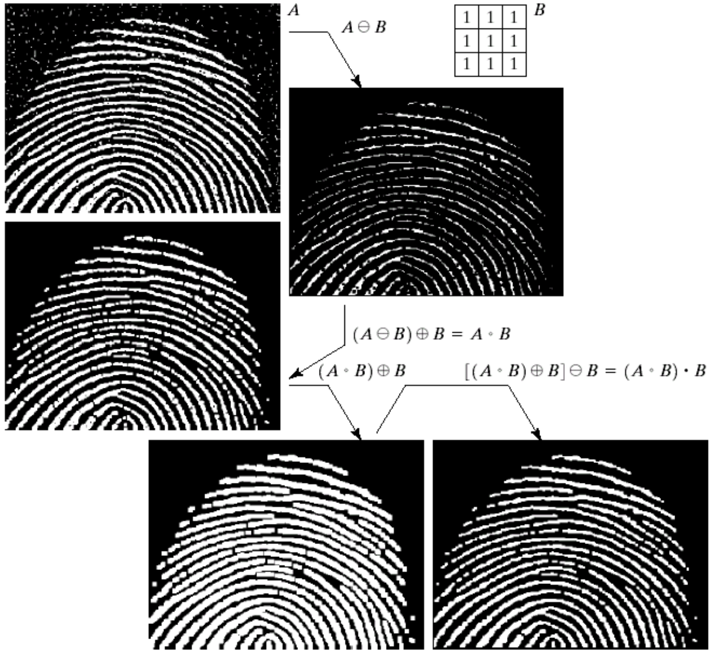

A
$A \ominus B$
1 1 1
1 1 1
1 1 1
B
$(A \ominus B) \oplus B = A \circ B$
$(A \circ B) \oplus B$
$[(A \circ B) \oplus B] \ominus B = (A \circ B) \bullet B$

Image Acquisition

Object Recognition

Representation & Description

Colour Image Processing

Image Compression

Key Stages in Digital Image Processing: Segmentation2

Image Restoration

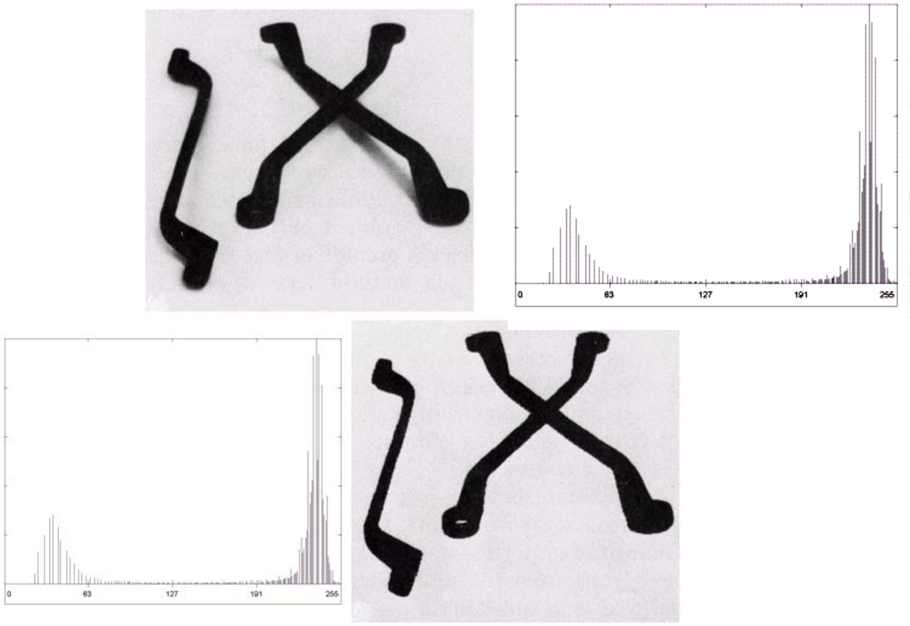
Morphological Processing

Image Enhancement

Segmentation

Image Acquisition

Object Recognition



Representation & Description

Colour Image Processing

Image Compression

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Object Recognition

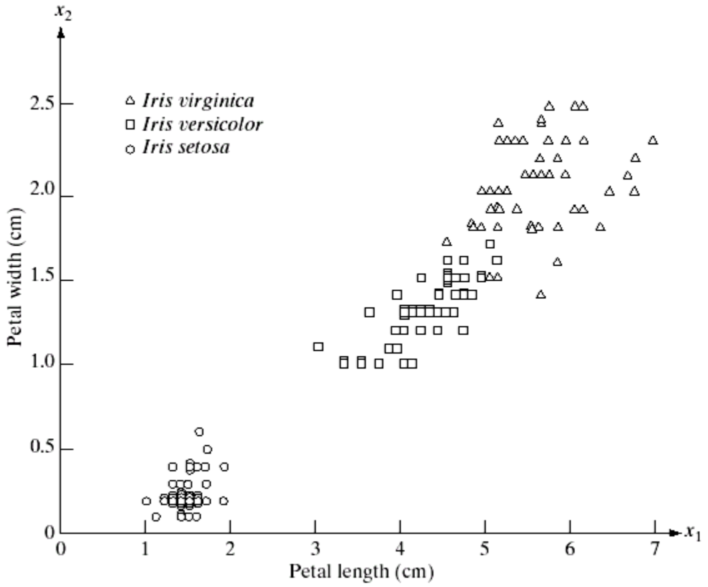
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Restoration

Morphological Processing

Image Enhancement

Image Acquisition



Segmentation

Object Recognition

Representation & Description

Colour Image Processing

Image Compression

Key Stages in Digital Image Processing: Representation & Description

Image Restoration

Morphological Processing

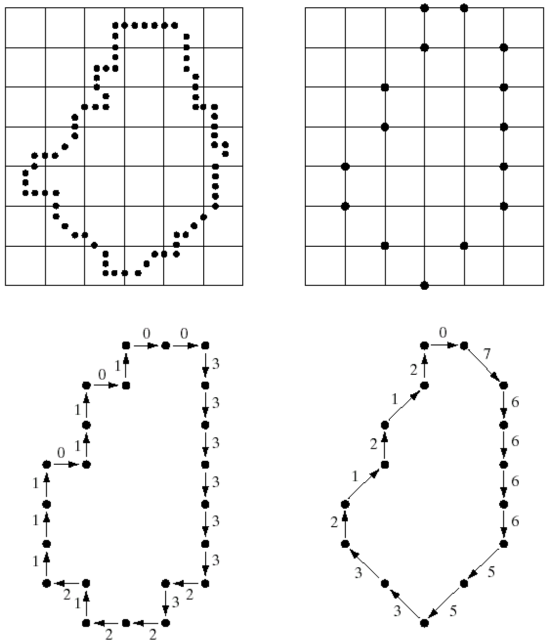
Image Enhancement

Segmentation

Image Acquisition

Object Recognition

Representation & Description



Colour Image Processing

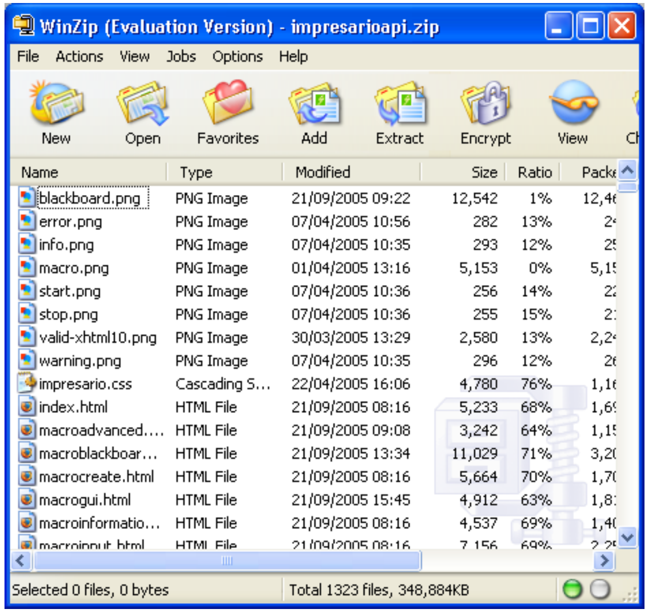
Image Compression

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Image Compression

Image Restoration

Morphological Processing



Name	Type	Modified	Size	Ratio	Pack
blackboard.png	PNG Image	21/09/2005 09:22	12,542	1%	12,4
error.png	PNG Image	07/04/2005 10:56	282	13%	2
info.png	PNG Image	07/04/2005 10:35	293	12%	2
macro.png	PNG Image	01/04/2005 13:16	5,153	0%	5,1
start.png	PNG Image	07/04/2005 10:36	256	14%	2
stop.png	PNG Image	07/04/2005 10:36	255	15%	2
valid-xhtml10.png	PNG Image	30/03/2005 13:29	2,580	13%	2,2
warning.png	PNG Image	07/04/2005 10:35	296	12%	2
impresario.css	Cascading S...	22/04/2005 16:06	4,780	76%	1,1
index.html	HTML File	21/09/2005 08:16	5,233	68%	1,6
macroadvanced....	HTML File	21/09/2005 09:08	3,242	64%	1,1
macroblackboar...	HTML File	21/09/2005 13:34	11,029	71%	3,2
macrocreate.html	HTML File	21/09/2005 08:16	5,664	70%	1,7
macrogui.html	HTML File	21/09/2005 15:45	4,912	63%	1,8
macroinformatio...	HTML File	21/09/2005 08:16	4,537	69%	1,4
macroinput.html	HTML File	21/09/2005 08:16	7,156	69%	2,2

Selected 0 files, 0 bytes

Total 1323 files, 348,884KB

Image Enhancement

Segmentation

Image Acquisition

Object Recognition

Representation & Description

Colour Image Processing

Image Compression

Key Stages in Digital Image Processing: Colour Image Processing

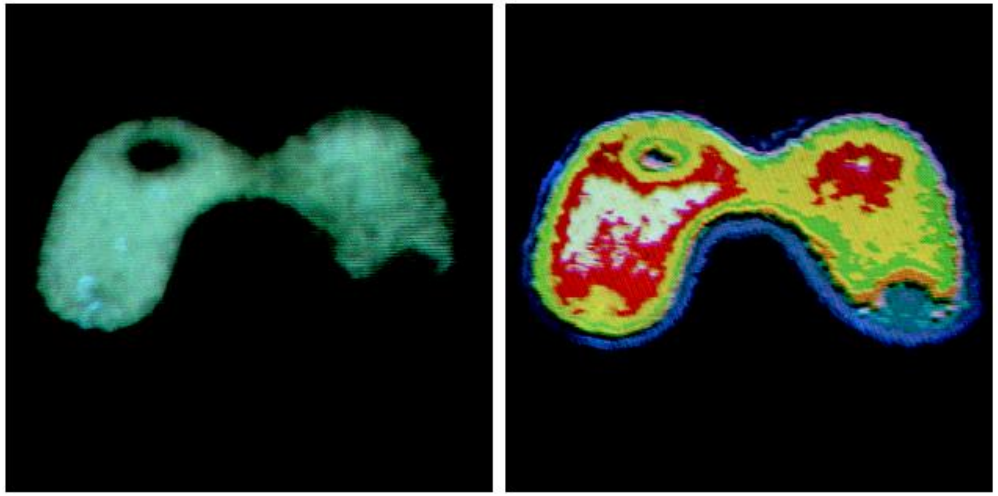


Image Restoration

Morphological Processing

Image Enhancement

Segmentation

Image Acquisition

Object Recognition

Representation & Description

Colour Image Processing

Image Compression

Unipolar Encoding

- Figure was transmitted in this way and reproduced on a telegraph printer fitted with typefaces simulating a halftone pattern

- The initial problems in improving the visual quality of these early digital pictures were related to the selection of printing procedures and the distribution of intensity levels

Unipolar Encoding

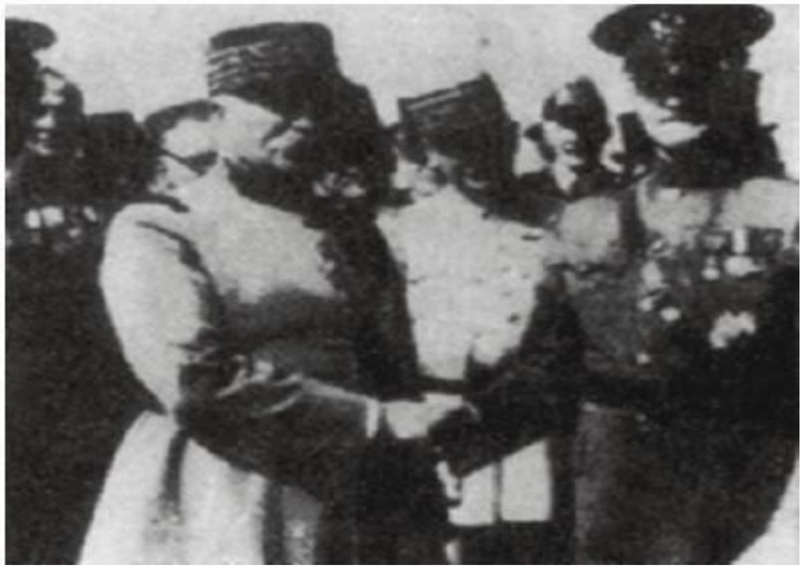
- The printing technique based on phographic reproduction made from tapes perforated at the perforated at the telegraph receiving terminal from 1921

- Figure shows an image obtained using this method.

- The improvements are tonal quality and in resolution

Unipolar Encoding

- The early Bartlane systems were capable of coding images in five distinct levels of gray.
- This capability was increased to 15 levels in 1929



- Figure is typical of the type of images that could be obtained using the 15-tone equipment

Image Sampling and Quantization

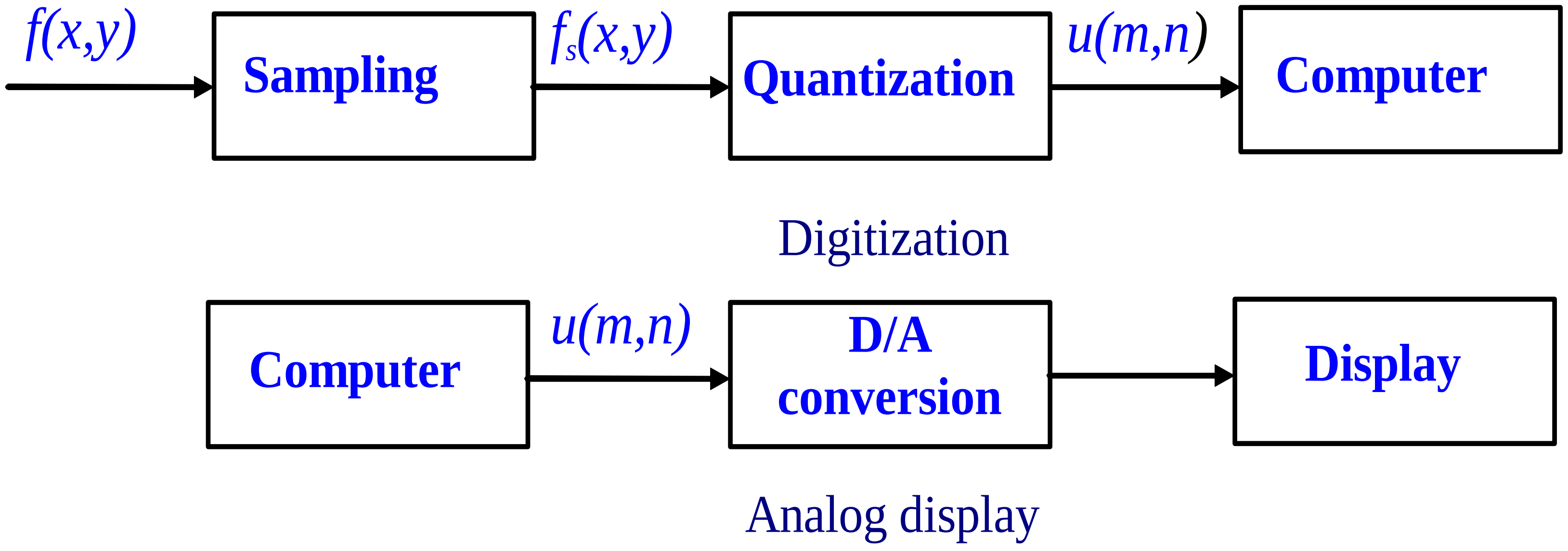


Fig 1 Image sampling and quantization / Analog image display

Image Sampling and Quantization

Sampling in the two-dimensional space Basics on image sampling

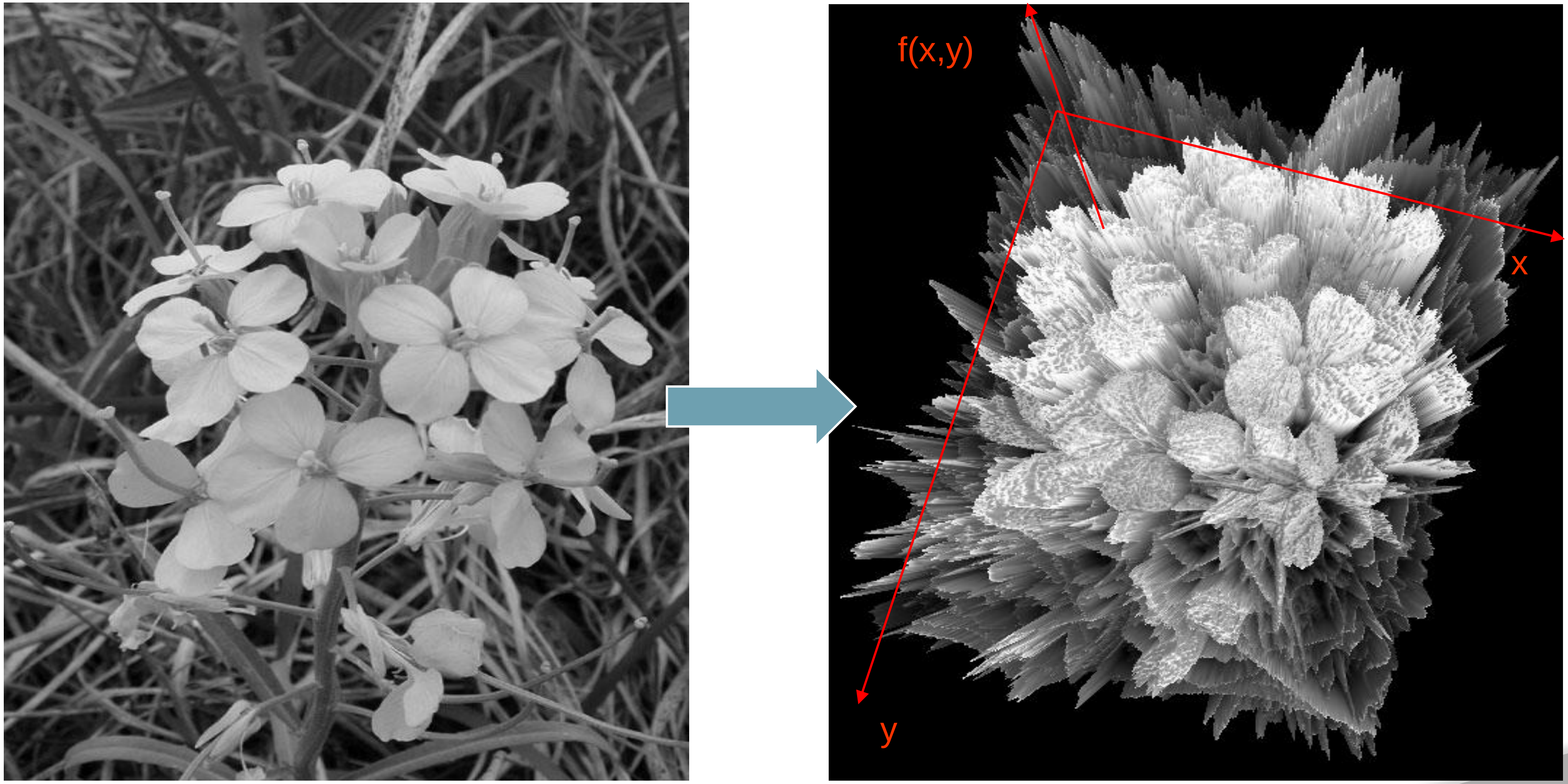
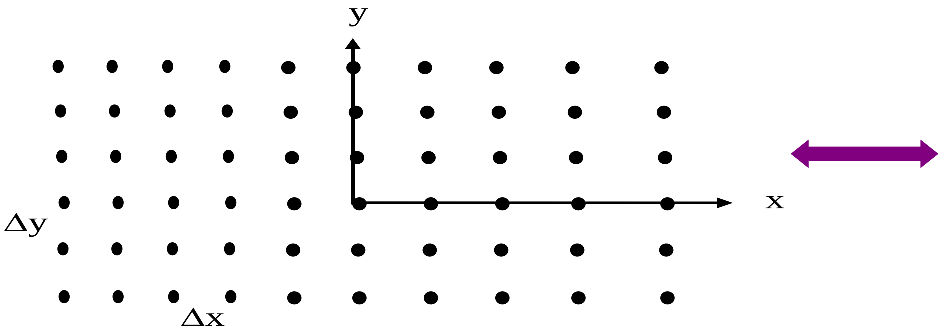
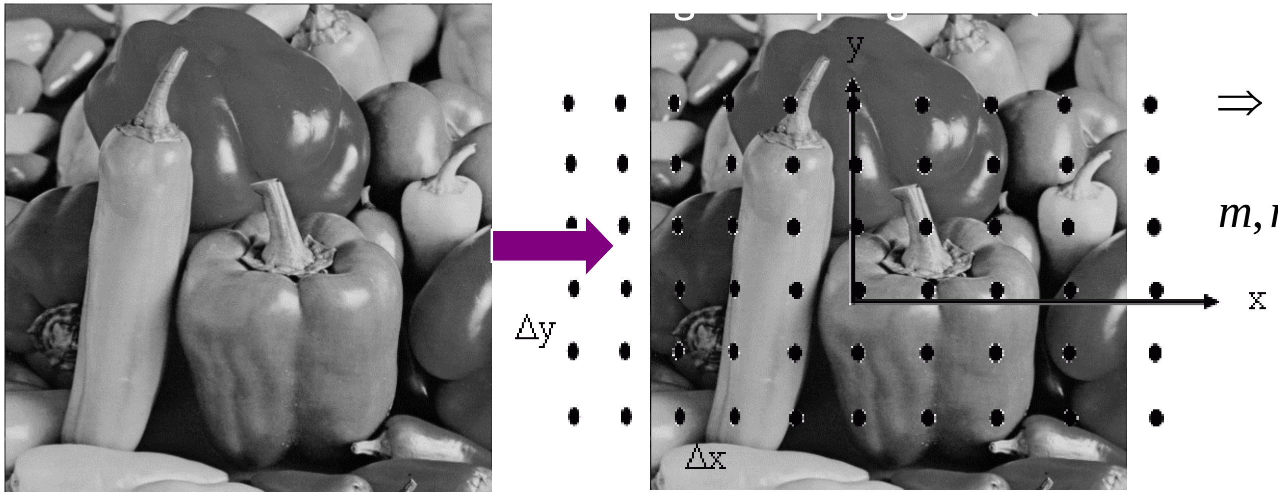


Image Sampling and Quantization



$$g_{(\Delta x,\Delta y)}(x,y)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\delta(x-m\Delta x,\,y-n\Delta y)$$

Image sampling = read from the original, spatially continuous, brightness function f(*x,y*), **only** in the black dots positions (⇔ **only** where the grid allows):



$$\Rightarrow f_s(x,y)=\begin{cases} f(x,y), & x=m\Delta x,\, y=n\Delta y \\ 0, & otherwise \end{cases},$$

$m,n\in\mathbf{Z}.$

$$\Leftrightarrow f_s(x,y)=f(x,y)g_{(\Delta x,\Delta y)}(x,y)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}f(m\Delta x,n\Delta y)\delta(x-m\Delta x,\,y-n\Delta y)$$

Image Sampling and Quantization

The Nyquist rate. The aliasing. The fold-over frequencies

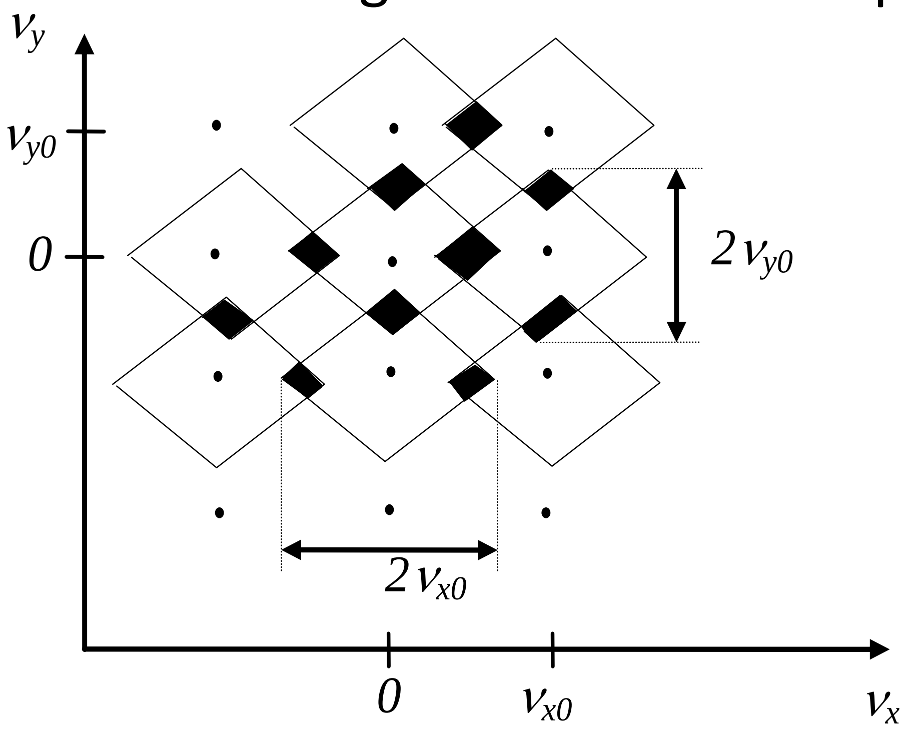
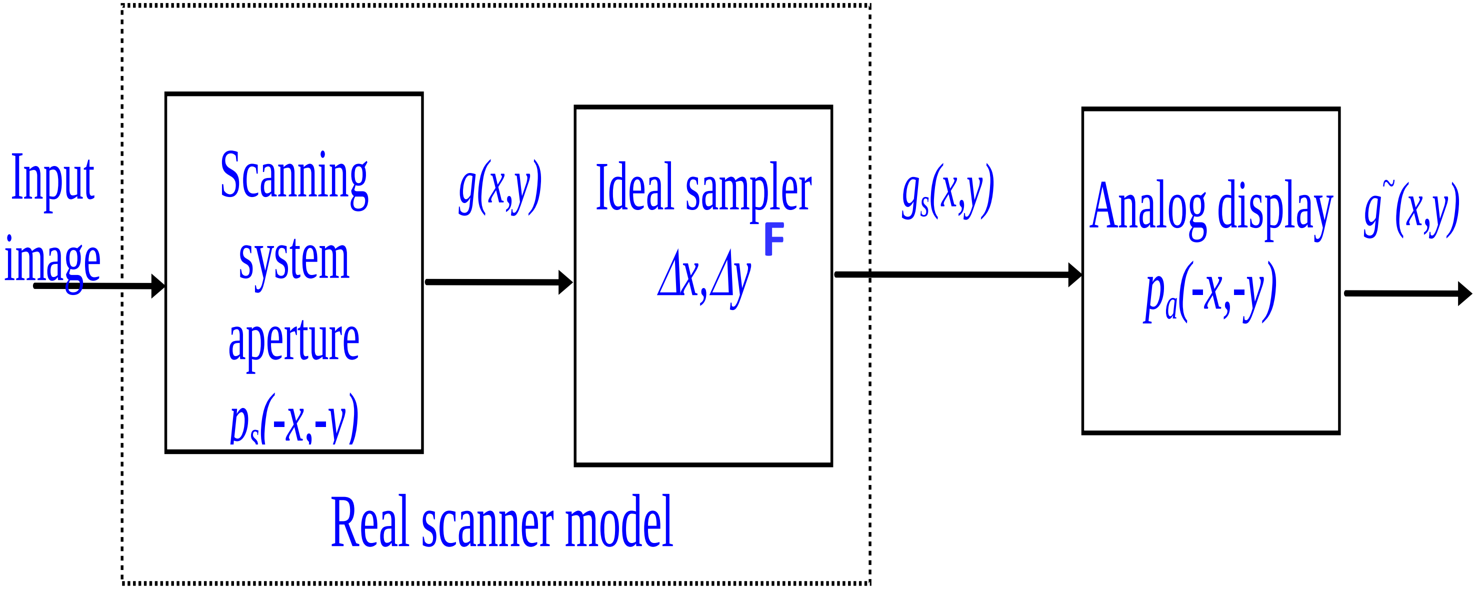


Fig. 5 Aliasing – fold-over frequencies

The sampling theorem in the two-dimensional case

Practical limitations in image sampling and reconstruction



ig. 7 The block diagram of a real sampler & reconstruction (display) system

Practical limitations in image sampling and reconstruction

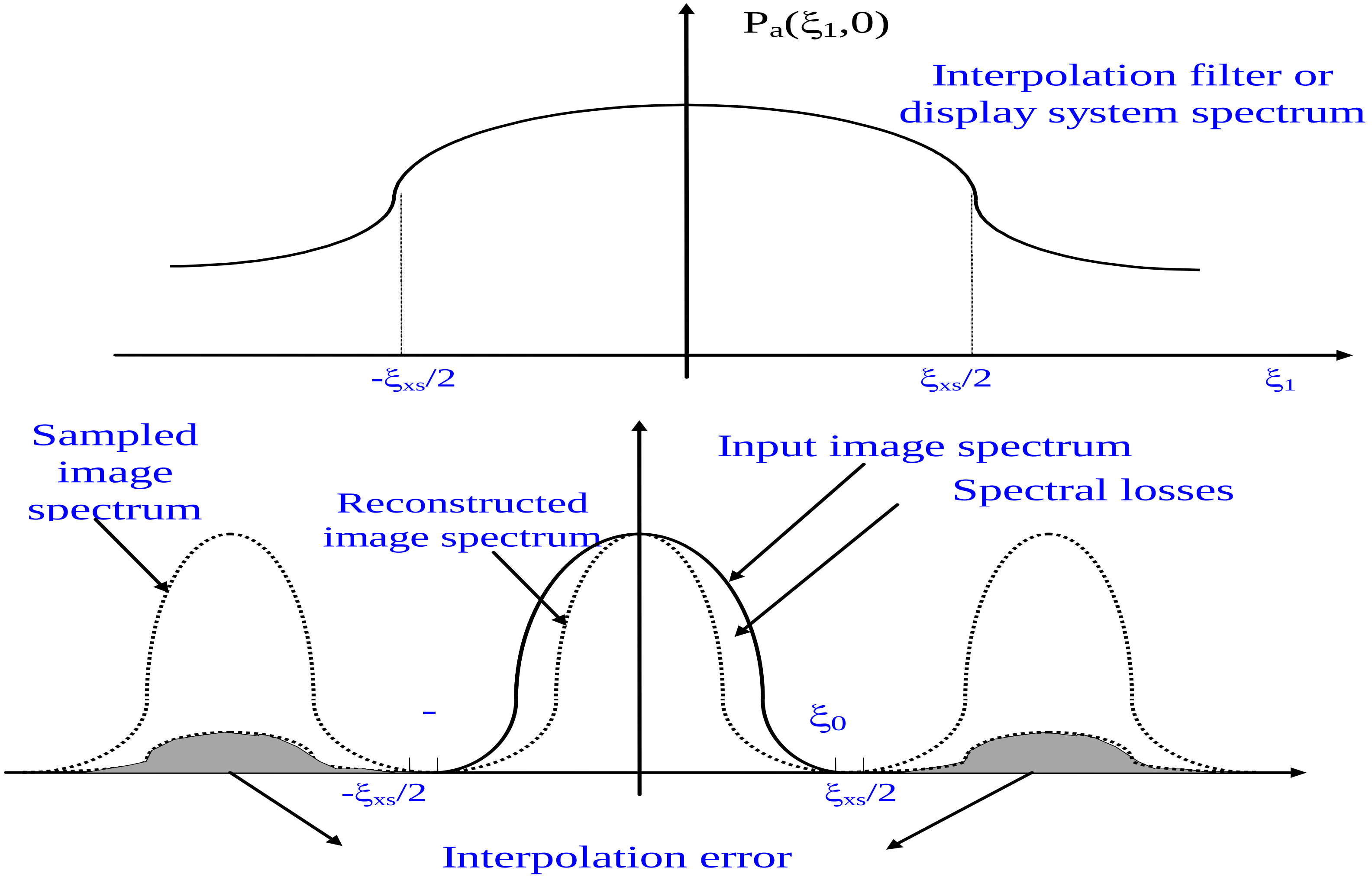


Image Sampling and Quantization

-

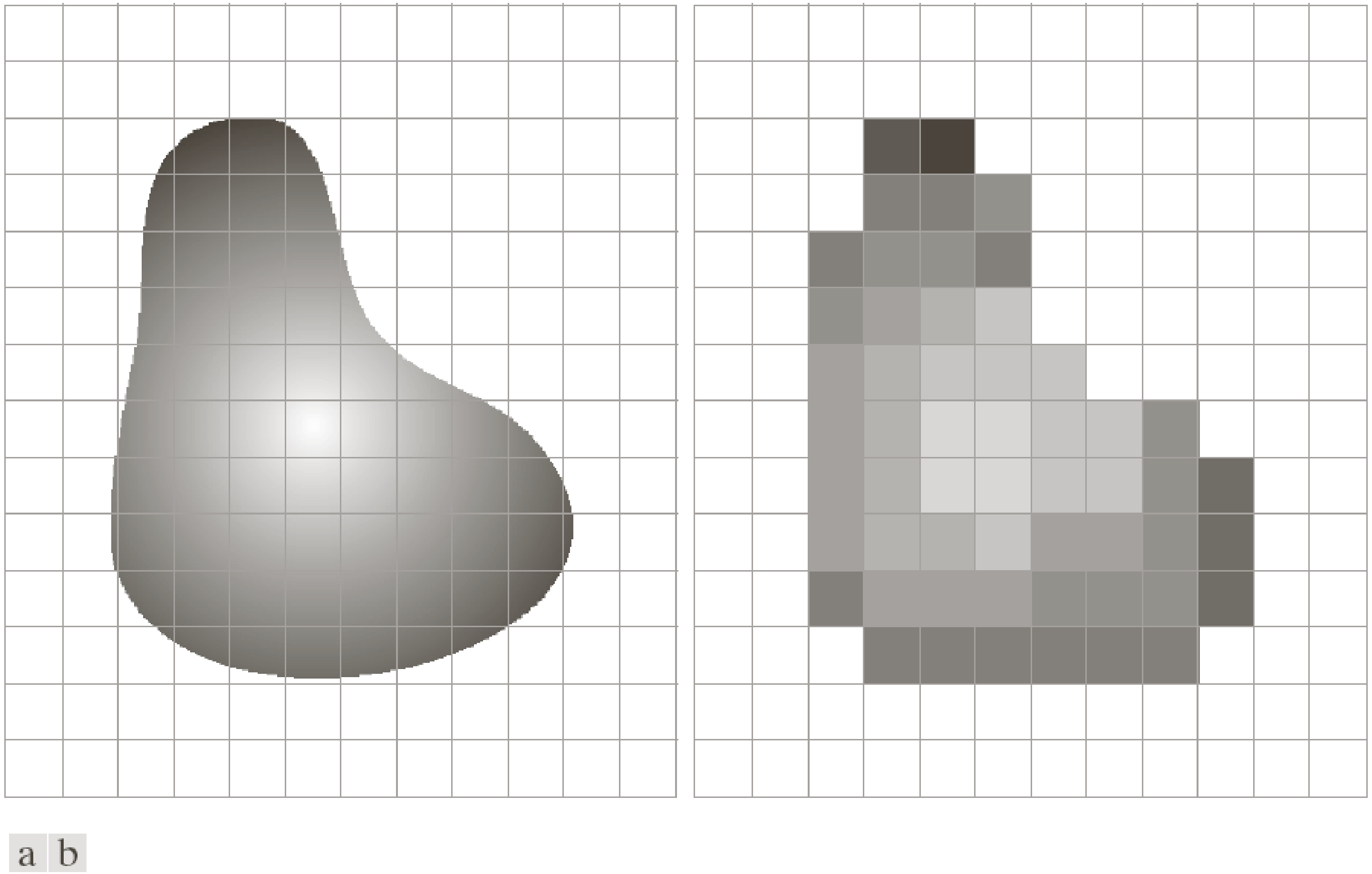
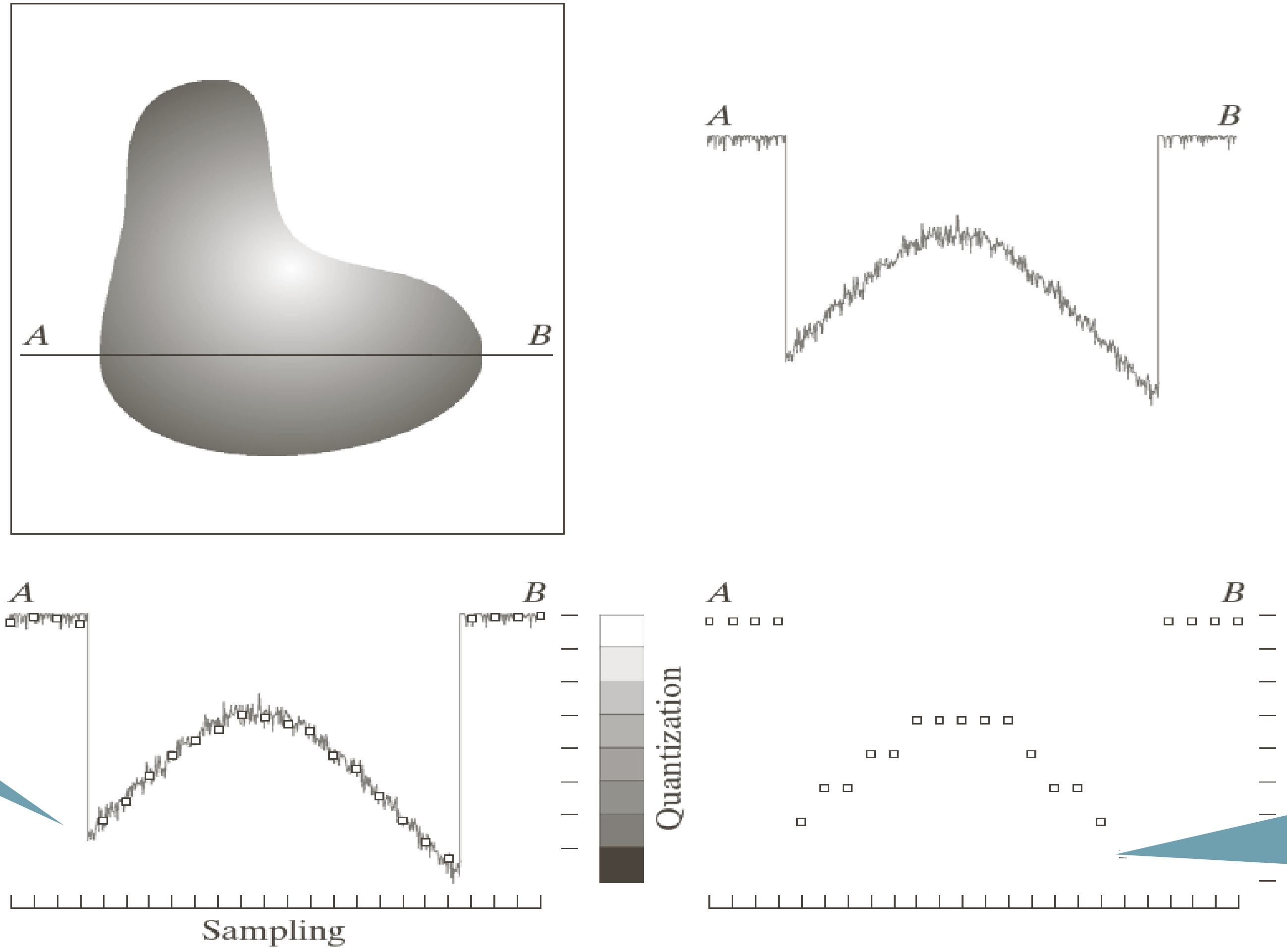


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Image Sampling and Quantization



a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Digitizing the coordinate values

Digitizing the amplitude values

Picture elements, Image elements, pels, and pixels

- A digital image is composed of a finite number of elements, each of which has a particular location and value.

- These elements are referred to as picture elements, image elements, pels, and pixels.

- Pixel is the term most widely used to denote the elements of a digital image.

Basic Relationships Between Pixels

1. Neighbors of a Pixel :-

A pixel p at coordinates (x, y) has four *horizontal and vertical neighbors whose coordinates are given by* (x+1, y), (x-1, y), (x, y+1), (x, y-1)

- This set of pixels, called the 4-*neighbors of p, is denoted by* N_4(p).

- Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.

Basic Relationships Between Pixels

N_D(p) and N_8(p)

- The four diagonal neighbors of p have coordinates (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)

and are denoted by ND(p).

- These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by N_8(p).

- If some of the points in ND(p) and N8(p) fall outside the image if (x, y) is on the border of the image.

Basic Relationships Between Pixels

We consider three types of adjacency:

(a) 4-adjacency.

Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.

(b) 8-adjacency.

Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.

(c) m-adjacency (mixed adjacency).

(d) Two pixels p and q with values from V are m-adjacent if

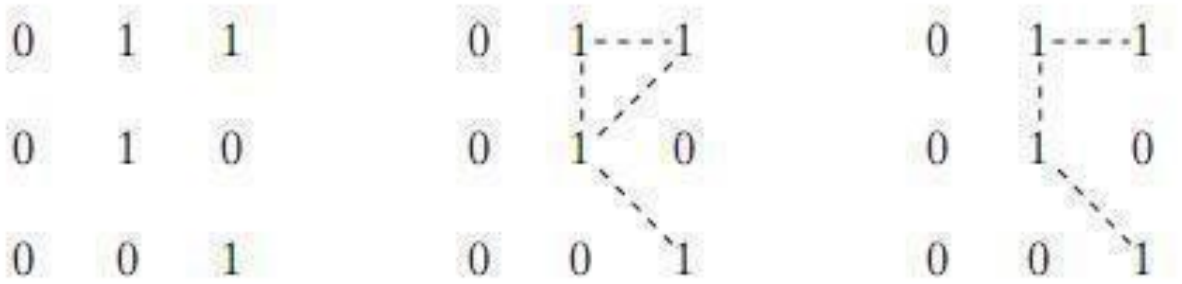
- (i) q is in $N_4(p)$, or

  - (ii) q is in $N_D(p)$ and the set whose values are from V.

Basic Relationships Between Pixels

- Two pixels p and q are said to be connected in S if there exists a path between them consisting entirely of pixels in S.

- For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S.



a b c

FIGURE (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.

Image transforms

<u>why transform?</u>

- Better image processing
  - Take into account long-range correlations in space
  - Conceptual insights in spatial-frequency information. what it means to be “smooth, moderate change, fast change, …”
- Fast computation: convolution vs. multiplication

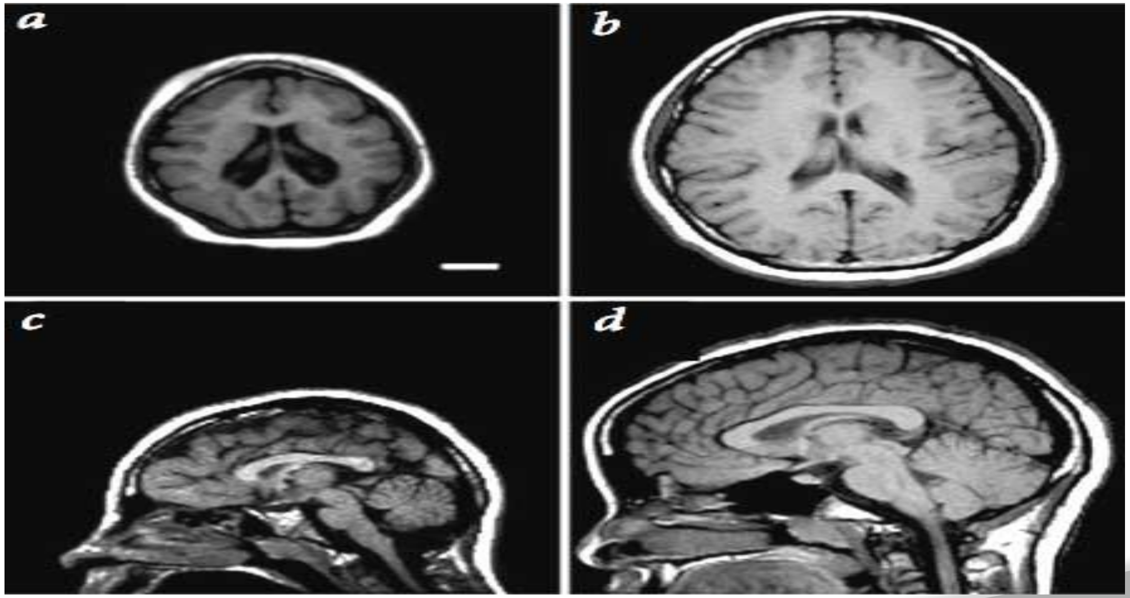
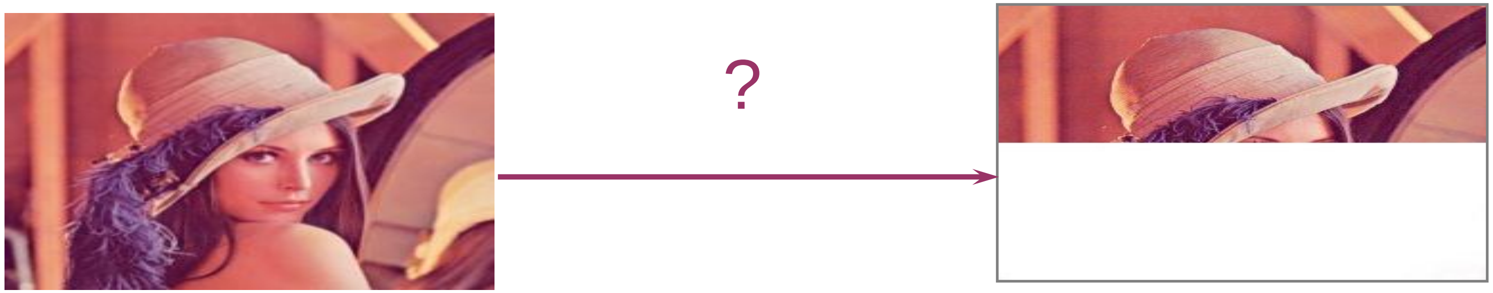


Image transforms

- Alternative representation and sensing
  - Obtain transformed data as measurement in radiology images (medical and astrophysics), inverse transform to recover image

- Efficient storage and transmission
  - Energy compaction
  - Pick a few “representatives” (basis)
  - Just store/send the “contribution” from each basis



Why Fourier Transform

- The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components.

- The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression.

Why 2D Fourier Transform

- Two-Dimensional Fourier Transform can be generalized to higher dimensions. For example, many signals are functions of 2D space defined over an x-y plane. Two-dimensional Fourier transform also has four different forms depending on whether the 2D signal is periodic and discrete.
- A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT).

Implementation of the 2-D Walsh Transform

- The 2-D Walsh transform is separable and symmetric.
- Therefore it can be implemented as a sequence of two 1-D Walsh transforms, in a fashion similar to that of the 2-D DFT.
- The Walsh transform consists of basis functions whose values are only 1 and -1.
- They have the form of square waves.
- Remember that the Fourier transform is based on trigonometric terms

Implementation of the 2-D Walsh Transform

- These functions can be implemented more efficiently in a digital environment than the exponential basis functions of the Fourier transform.
- The concept of frequency exists also in Walsh transform basis functions.
- We can think of frequency as the number of zero crossings or the number of transitions in a basis vector and we call this number sequency.

Computation of the Walsh Transform

- For the fast computation of the Walsh transform there exists an algorithm called Fast Walsh Transform (FWT).

- This is a straightforward modification of the FFT.

Properties of the Hadamard Transform

- Most of the comments made for Walsh transform are valid here.
- The Hadamard transform differs from the Walsh transform only in the order of basis functions. The order of basis functions of the Hadamard transform does not allow the fast computation of it by using a straightforward modification of the FFT.

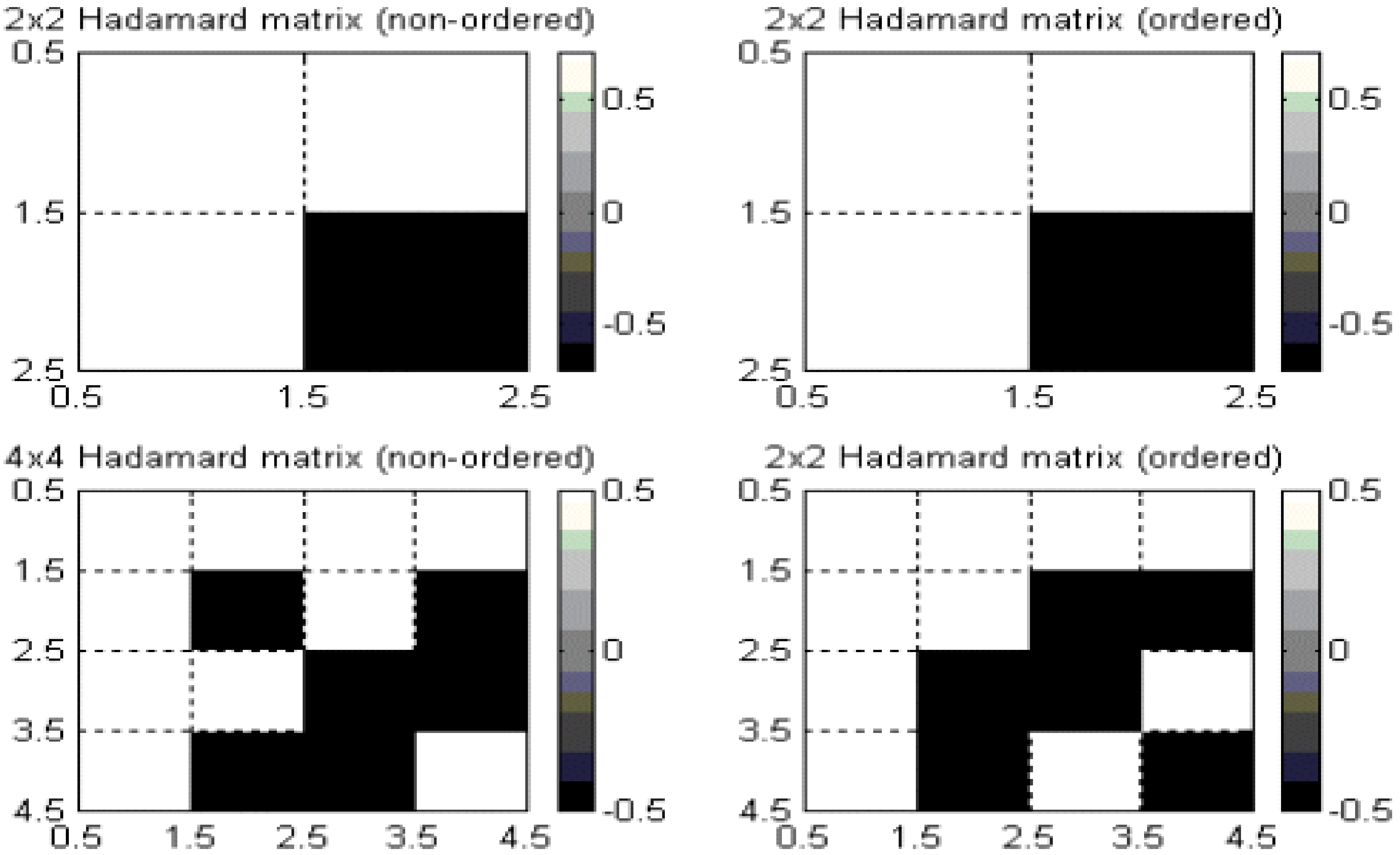
Recursive Relationship of the Hadamard Transform

- An important property of Hadamard transform is that, letting H_N represent the Hadamard matrix of order N the recursive relationship holds :

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

- Therefore, starting from a small Hadamard matrix we can compute a Hadamard matrix of any size.
- This is a good reason to use the Hadamard transform!

Images of 1-D Hadamard matrices



Slant transform

- The Slant transform matrix of order N x N is the recursive expression S_n is given by

$$
= \frac{1}{\sqrt{2}}
\left[\begin{array}{cc|c|cc|c}
1 & 0 & 0 & 1 & 0 & 0 \\
a_N & b_N & & -a_N & b_N & \\
\hline
\multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & I_{(N/2)-2} \\
\hline
0 & 1 & 0 & 0 & -1 & 0 \\
-b_N & a_N & & b_N & a_N & \\
\hline
\multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & -I_{(N/2)-2}
\end{array}\right]
\left[\begin{array}{c|c}
S_{N/2} & 0 \\
\hline
0 & S_{N/2}
\end{array}\right]
$$

Where I_m is the identity matrix of order M x M, and

$$
S_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}
$$

Properties of Slant transform

- The slant transform is real and orthogonal.

  $S = S^{*}$ S-1 = ST

- The slant transform is fast, it can be implemented in (N $\log_2$N) operations on an N x 1 vector.

- The energy deal for images in this transform is rated in very good to excellent range.

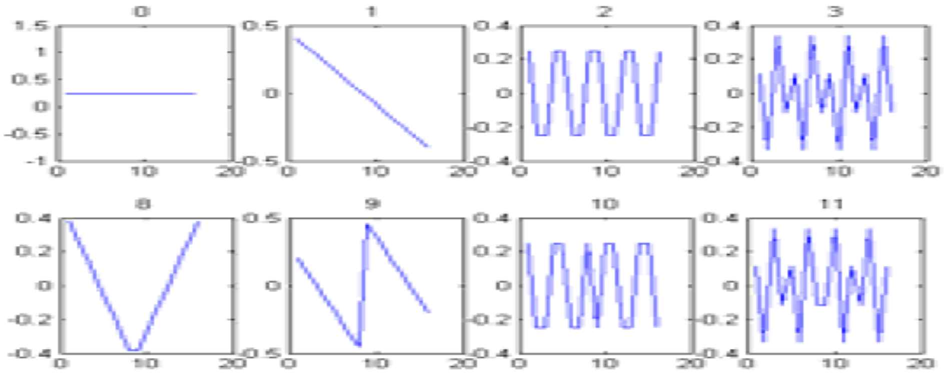
- The mean vectors for slant transform matrix S are not sequentially ordered for n ≥ 3.

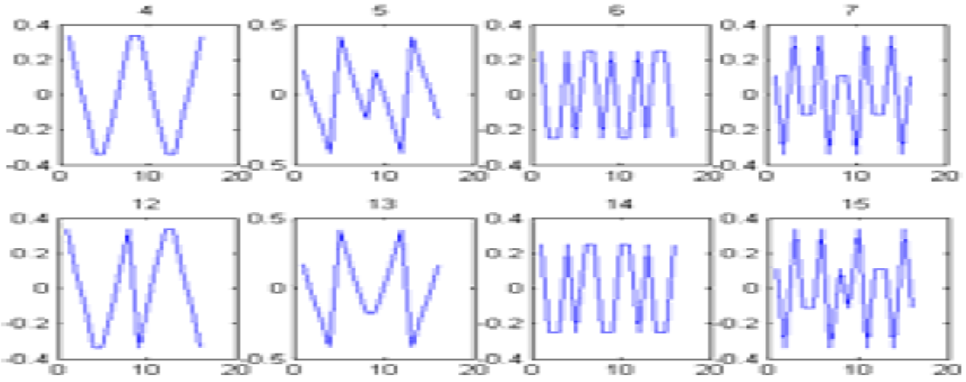
-

Slant transform

- similar to STFT (short-time Fourier transform)
  - partition a NxN image into mxn sub-images
  - save computation: O(N) instead of O(NlogN)
  - loose long-range correlation

$$\begin{bmatrix}
0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\
0.5401 & 0.3858 & 0.2315 & 0.0772 & -0.0772 & -0.2315 & -0.3858 & -0.5401 \\
0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\
0.1581 & -0.4743 & 0.4743 & -0.1581 & 0.1581 & -0.4743 & 0.4743 & -0.1581 \\
0.4743 & 0.1581 & -0.1581 & -0.4743 & -0.4743 & -0.1581 & 0.1581 & 0.4743 \\
0.2415 & -0.0345 & -0.3105 & -0.5866 & 0.5866 & 0.3105 & 0.0345 & -0.2415 \\
0.3536 & -0.3536 & -0.3536 & 0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 \\
0.1581 & -0.4743 & 0.4743 & -0.1581 & -0.1581 & 0.4743 & -0.4743 & 0.1581
\end{bmatrix}$$





Energy compaction comparison

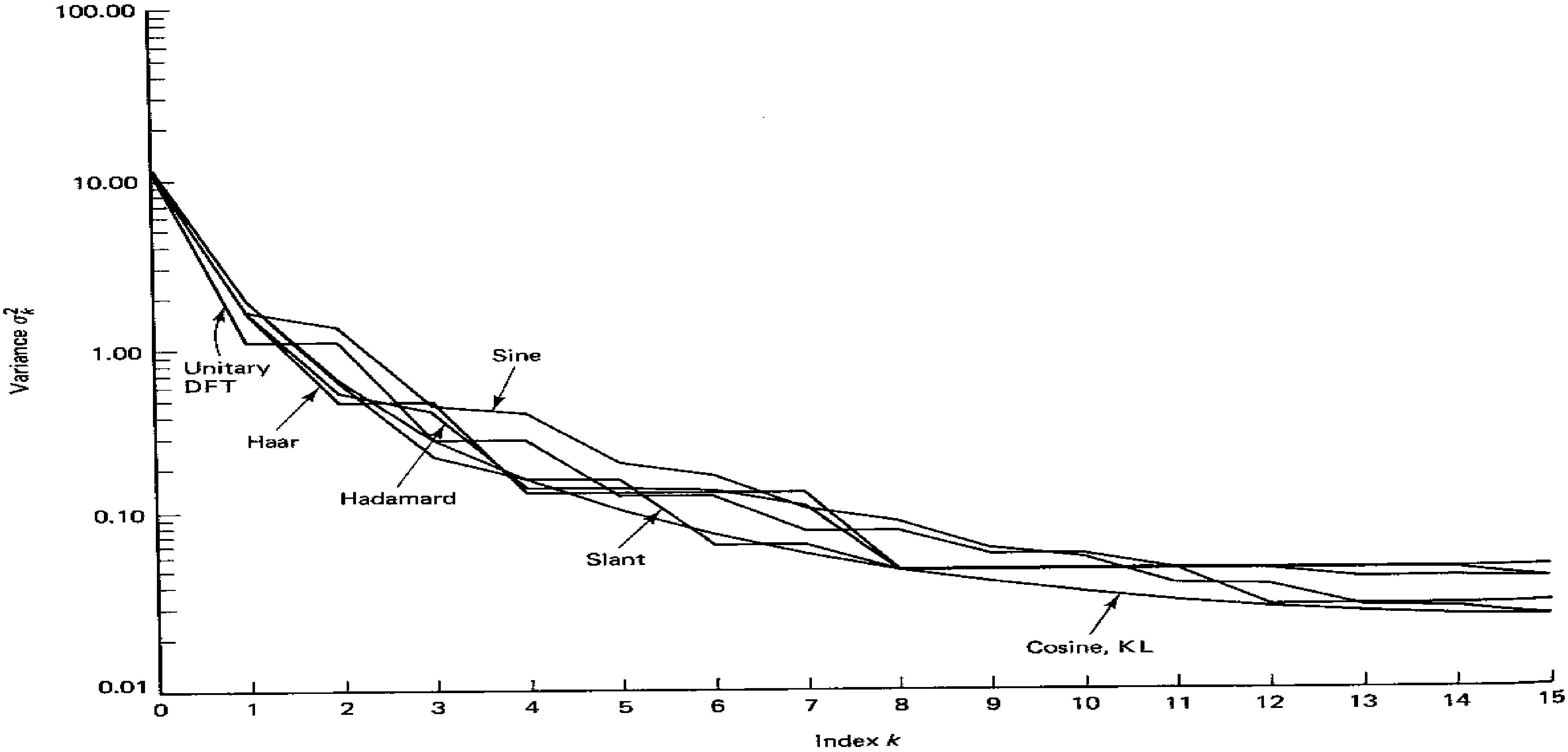


Figure 5.18 Distribution of variances of the transform coefficients (in decreasing order) of a stationary Markov sequence with $N = 16$, $\rho = 0.95$ (see Example 5.9).

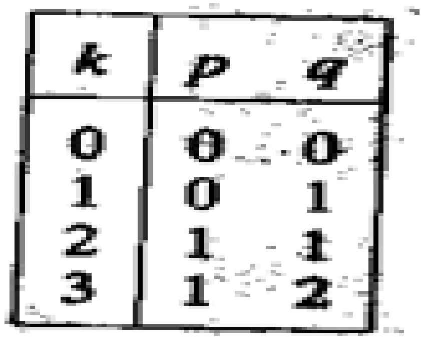
Haar Transform

- The Haar transform is based on the Haar functions, $h_k(z)$, which are defined over the continuous, closed interval z ε [0, 1],
- and for $k = 0, 1, 2 \ldots , N-1$, where $N = 2^n$. The first step in generating the Haar transform is to note that the integer k can be decomposed uniquely as

$$k = 2^p + q - 1$$

- where $0 \leq p \leq n-1$, $q = 0$ or 1 for $p = 0$, and $1 \leq q \leq 2^p$ for $p \neq 0$. For example, if N = 4, k, q, p have following values

Haar Transform



k	p	q
0	0	0
1	0	1
2	1	1
3	1	2

The Haar functions are defined as

$$h_0(z) \triangleq h_{00}(z) = \frac{1}{\sqrt{N}}$$ for z ε [0, 1] ……. (1)

The Discrete Haar Transform

- A complete orthogonal system of functions in

  [0, 1], p ∈ [0, ∞] which take values from the set {0, 2j : j ∈ N} was

defined by Haar [1].

- This system of functions has property that each function continuous on interval [0, 1] may be represented by a uniformly and convergent series in terms of elements of this system. There are some other definitions of the Haar functions .

The Discrete Haar Transform

- Those definitions are mutually differing with respect to the values of Haar functions at the points of discontinuity.

- For example the original Haar definition is as follows [4]:

- $haar(0, t) = 1$, for $t \in [0, 1)$; $haar(1, t) = 1$, for $t \in [0$, and $haar(k, 0) = limt \to 0+ \; haar(k, t)$, $haar(k, 1) = limt \to 1- \; haar(k, t)$ and at the points of discontinuity within the interior $(0, 1)$ $haar(k, t) = 1$

Properties of Haar transform

- The Haar transform is real and orthogonal.

- The Haar transform is very fast. It can implement O(n) operations on an N x 1 vector.

- The mean vectors of the Haar matrix are sequentially ordered.
- It has a poor energy deal for images.

Hotelling transform

- The basic principle of hotelling transform is the statistical properties of vector representation. Consider a population of random vectors of the form,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Hotelling transform

- And the mean vector of the population is defined as the expected value of x i.e.,
- $m_{x} = E\{x\}$
- The suffix m represents that the mean is associated with the population of x vectors. The expected value of a vector or matrix is obtained by taking the expected value of each elememt.
- The covariance matrix C_x in terms of x and m_x is given as

$$C_x = E\{(x-m_x)\ (x-m_x)^T\}$$

Hotelling transform

- T denotes the transpose operation. Since, x is n dimensional, $\{(x-m_x)(x-m_x)^T\}$ will be of n x n dimension. The covariance matrix is real and symmetric. If elements x_i and x_j are uncorrelated, their covariance is zero and, therefore,

$$c_{ij} = c_{ji} = 0.$$

- For M vector samples from a random population, the mean vector and covariance matrix can be approximated from the samples by

$$\mathbf{m}_x = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k$$

$$\mathbf{C}_x = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_x\mathbf{m}_x^T.$$

DIGITAL IMAGE PROCESSING

UNIT-II

IMAGE ENHANCEMENT

IMAGE ENHANCEMENT

Image enhancement:

1. Improving the interpretability or perception of information in images for human viewers
2. Providing `better' input for other automated image processing techniques

Spatial domain methods:

operate directly on pixels

Frequency domain methods:

operate on the Fourier transform of an image

IMAGE ENHANCEMENT

- To process an image so that the result is more suitable than the original image for a specific application.

- Spatial domain methods and frequency domain methods.

Spatial Domain Methods

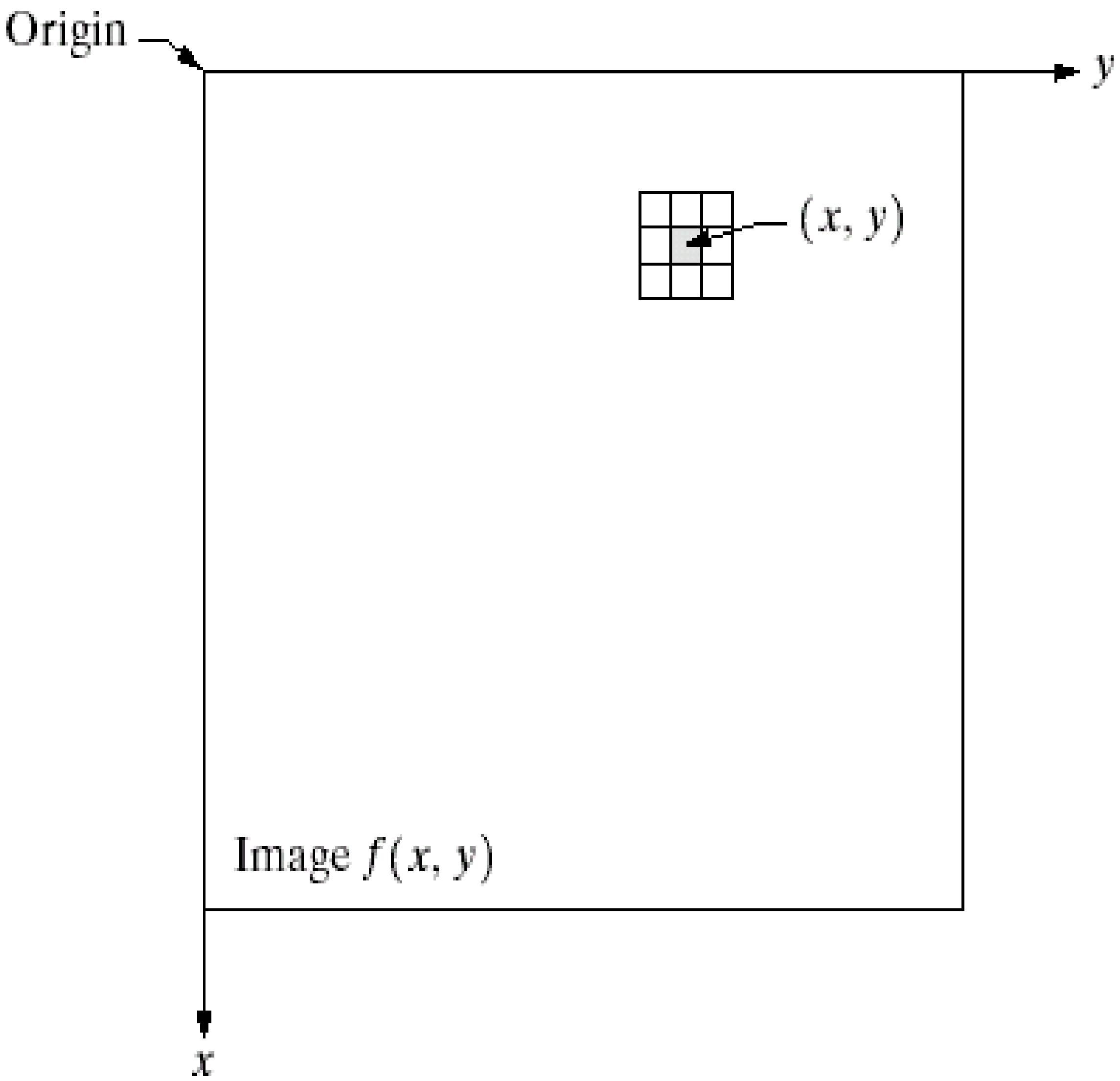


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.

Spatial Domain Methods

- Procedures that operate directly on the aggregate of pixels composing an image

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

- A neighborhood about (x,y) is defined by using a square (or rectangular) subimage area centered at (x,y).

Spatial domain

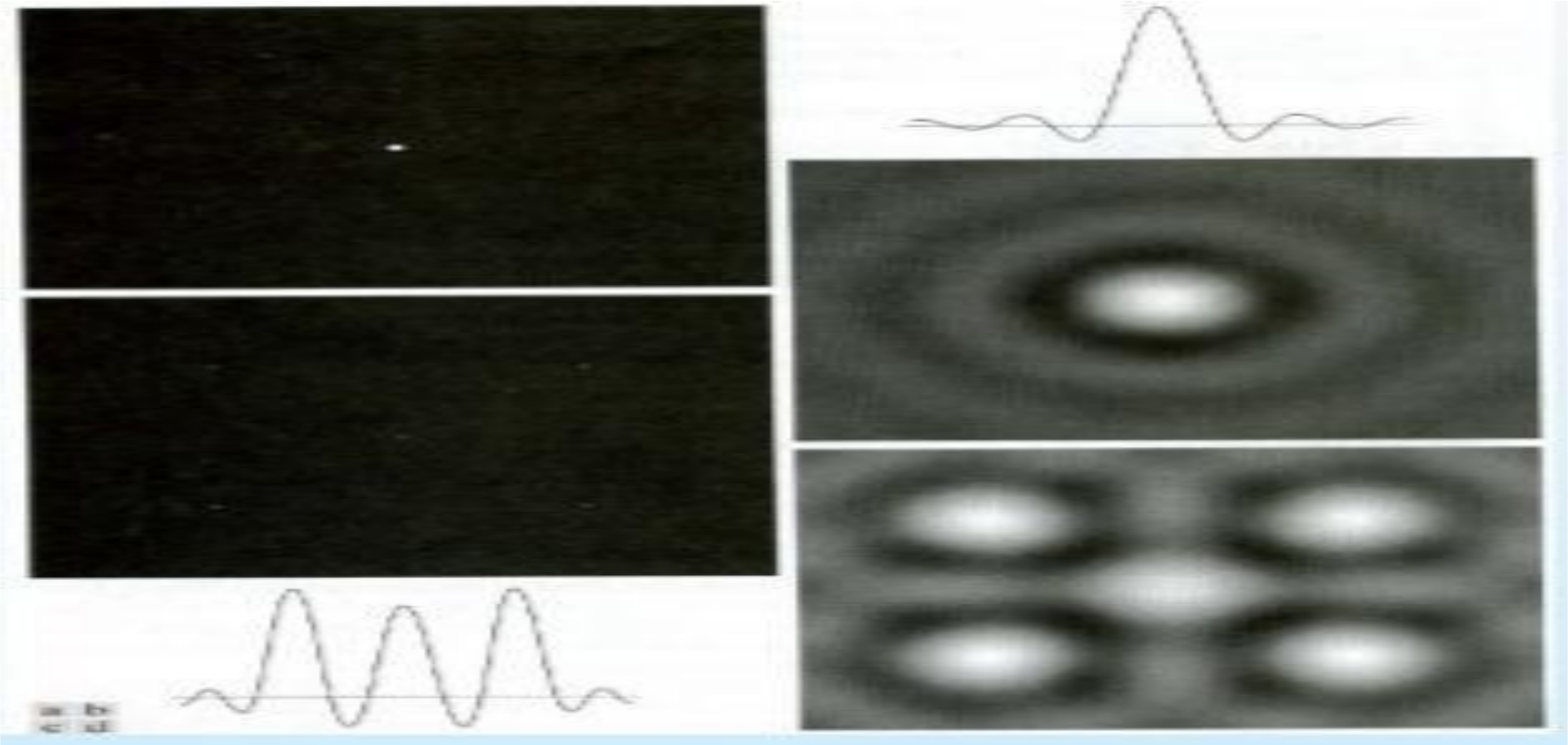


Fig. Spatial representation of ILPFs of order 1 and 20 and corresponding intensity

Spatial domain

- Spatial domain: Image Enhancement Three basic type of functions are used for image enhancement.
- Image enhancement point processing techniques: Linear (Negative image and Identity transformations) Logarithmic transformation Power law transforms (nth power and nth root transformations) Grey level slicing Bit plane slicing We are dealing now with image processing methods that are based only on the intensity of single pixels.
- Intensity transformations Linear function Negative and identity Transformations

Point Processing

- The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself •In this case T is referred to as a grey level transformation function or a point processing operation.

- Point processing operations take the form •s = T (r) •where s refers to the processed image pixel value and r refers to the original image pixel value
- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image – Note how much clearer the tissue is in the negative image of the mammogram below s = 1.0 - r Original Image Negative Image

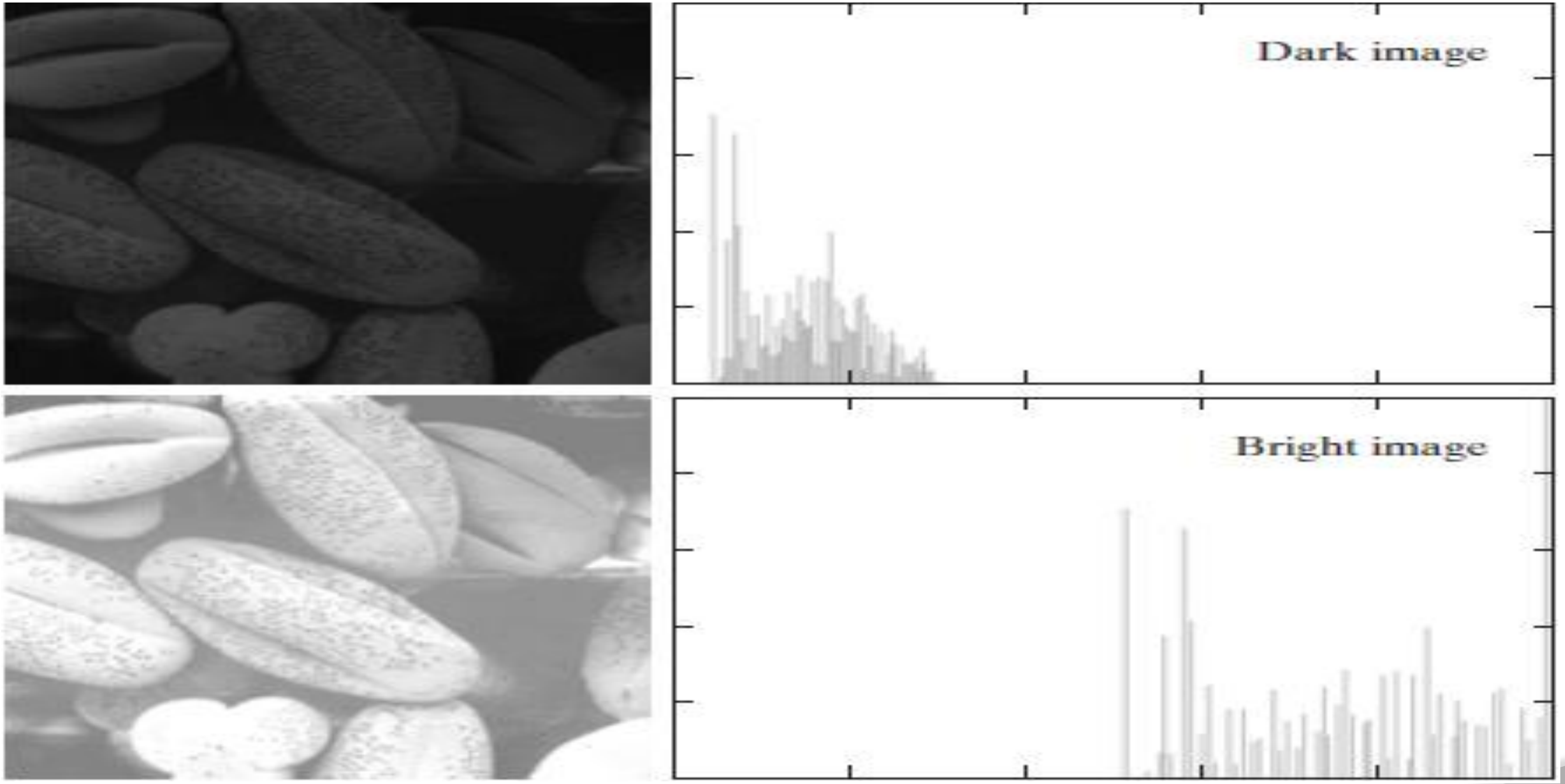
What is a Histogram?

- In Statistics, Histogram is a graphical representation showing a visual impression of the distribution of data.
- An Image Histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image. It plots the number of pixels for each value.
- The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $h(r_k) = n_k$, where r_k is the kth gray level and n_k is the number of pixels in the image having gray level r_k

Why Histogram?

- Histograms are the basis for numerous spatial domain processing techniques.
- Histogram manipulation can be used effectively for image enhancement
- Histograms can be used to provide useful image statistics

An Example of Histogram



Spatial domain

- Spatial domain: Image Enhancement Three basic type of functions are used for image enhancement.
- Image enhancement point processing techniques: Linear (Negative image and Identity transformations) Logarithmic transformation Power law transforms (nth power and nth root transformations) Grey level slicing Bit plane slicing We are dealing now with image processing methods that are based only on the intensity of single pixels.
- Intensity transformations Linear function Negative and identity Transformations

Spatial domain

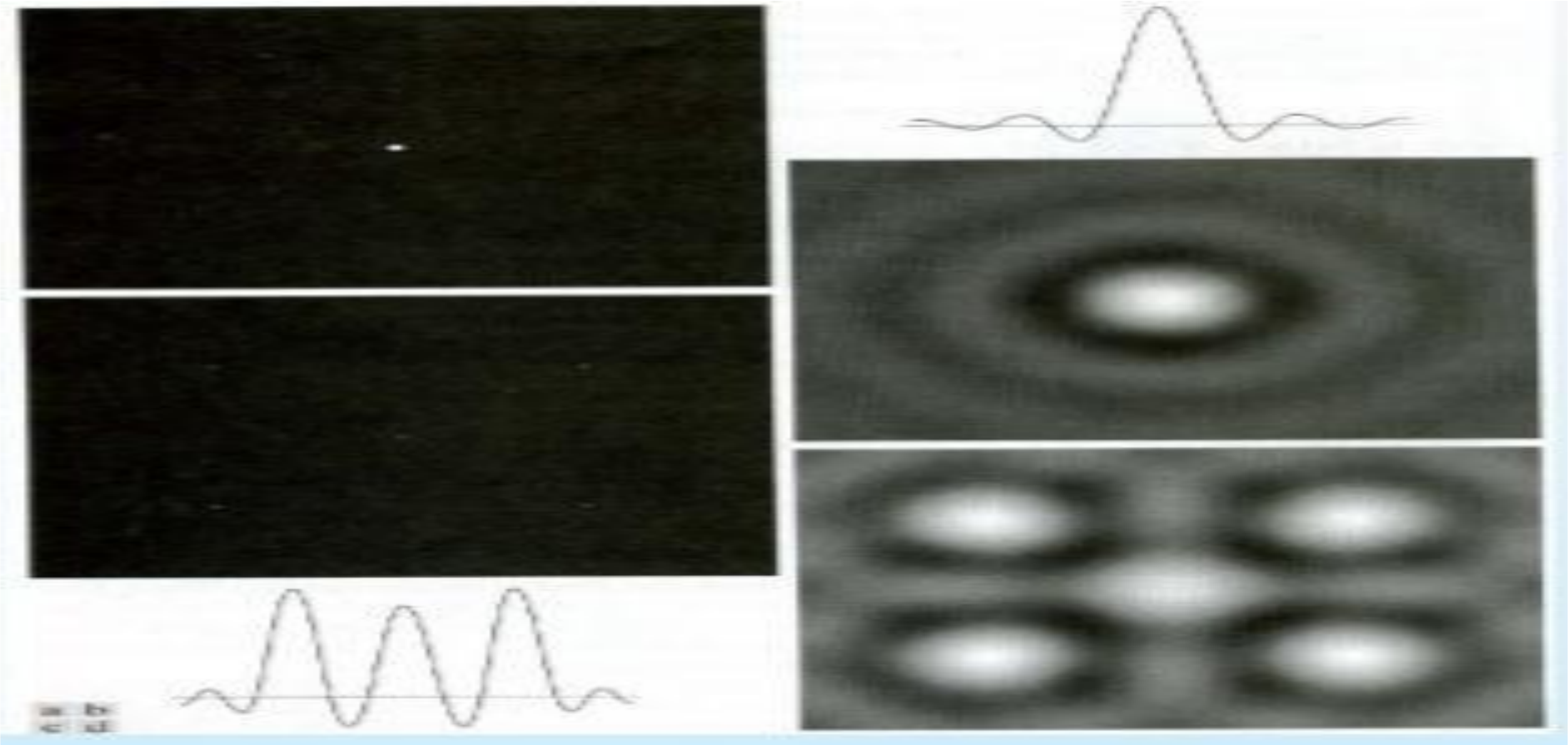


Fig. Spatial representation of ILPFs of order 1 and 20 and corresponding intensity

Spatial domain

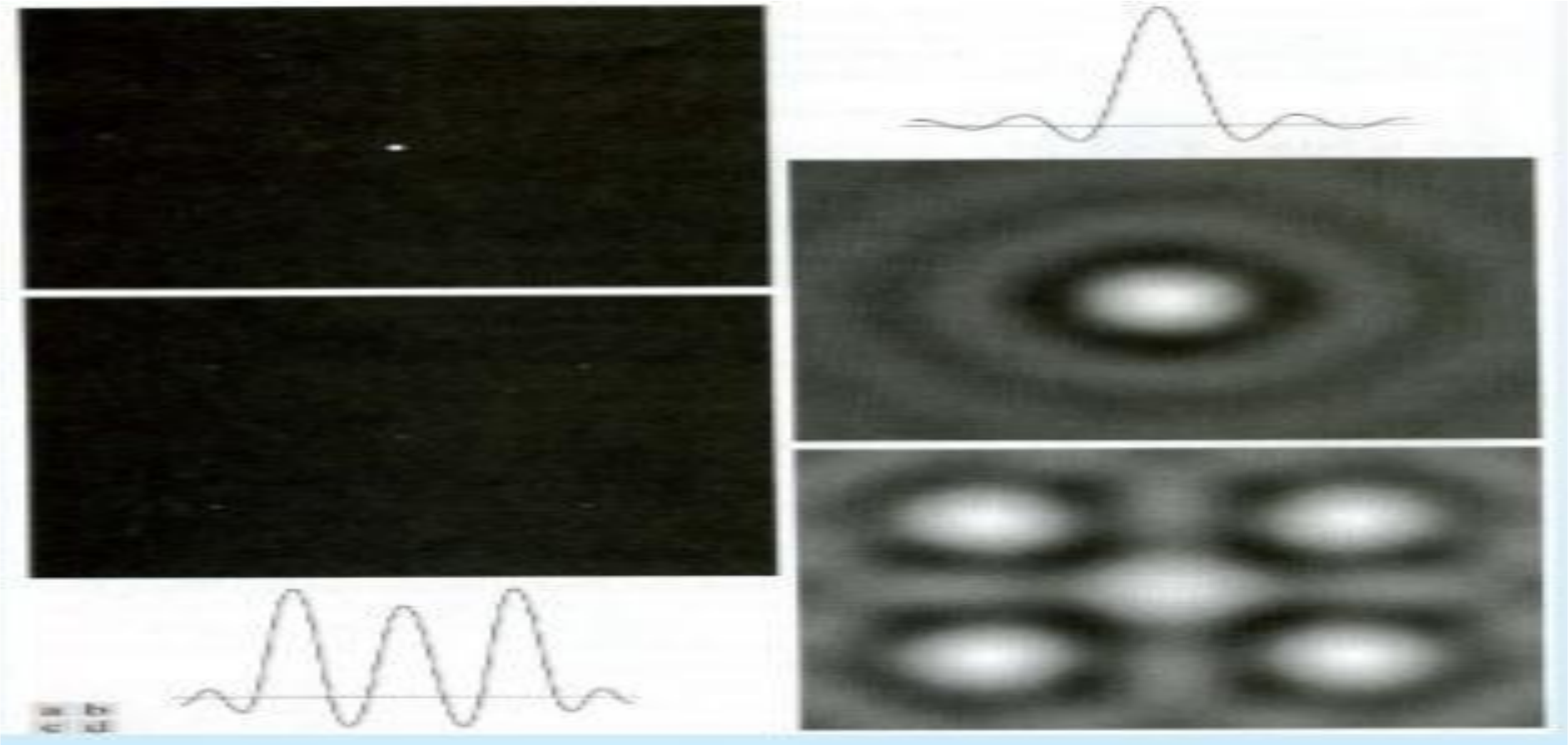


Fig. Spatial representation of ILPFs of order 1 and 20 and corresponding intensity

Spatial Domain Methods

- Procedures that operate directly on the aggregate of pixels composing an image
-
- A neighborhood about (x,y) is defined by using a square (or rectangular) subimage area centered at (x,y).

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

Spatial Domain Methods

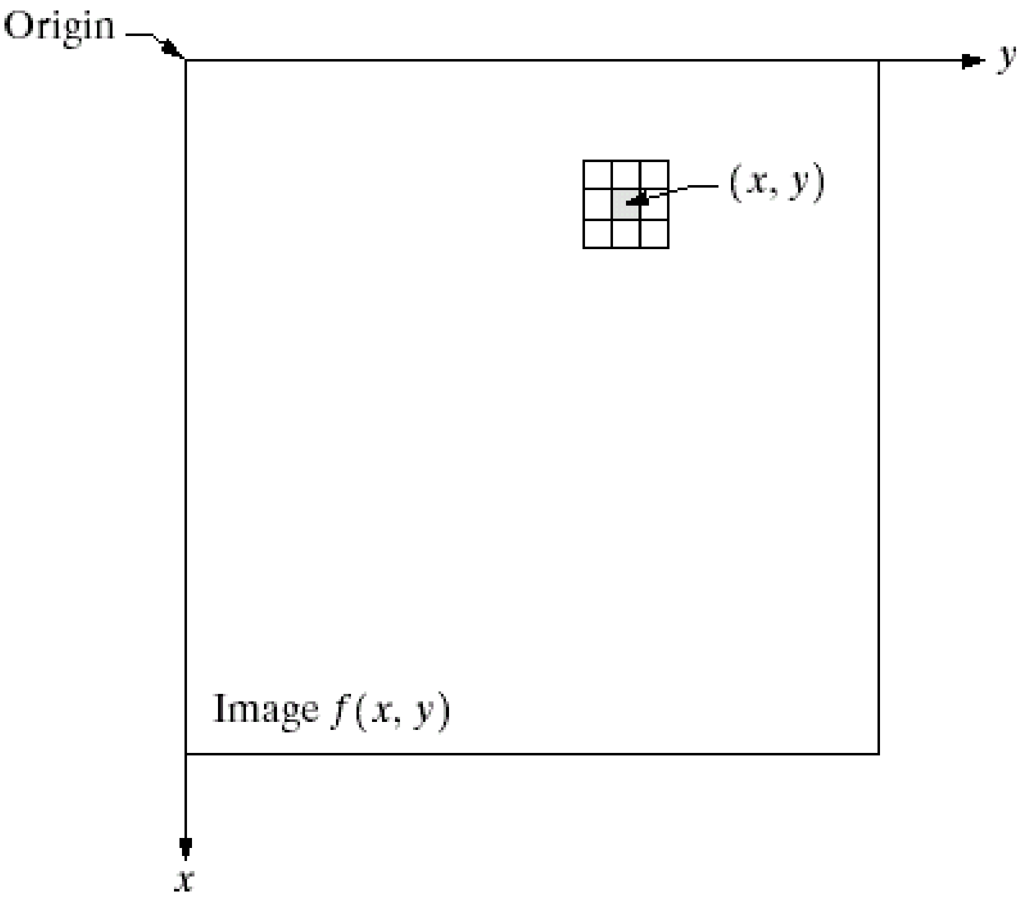


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.

Spatial Domain Methods

- When the neighborhood is 1 x 1 then g depends only on the value of f at (x,y) an becomes a gray-level transformation (or mapping) function:

$$s=T(r)$$

r,s: gray levels of f(x,y) and g(x,y) at (x,y)

- Point processing techniques (e.g. contrast stretching, thresholding)

Spatial Domain Methods

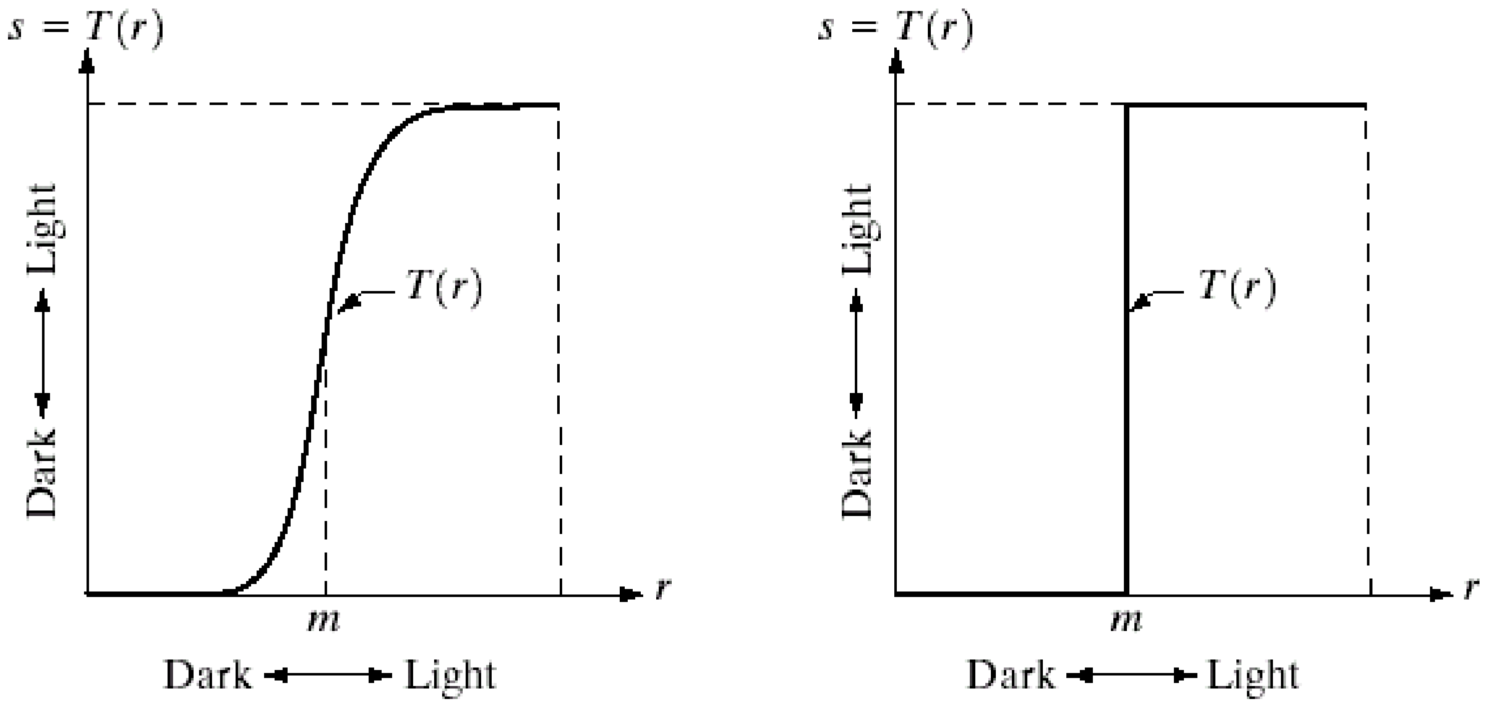


FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

Contrast Stretching

Thresholding

Spatial Domain Methods

- Mask processing or filtering: when the values of f in a predefined neighborhood of (x,y) determine the value of g at (x,y).
  - Through the use of masks (or kernels, templates, or windows, or filters).
- These are methods based only on the intensity of single pixels.
  - r denotes the pixel intensity before processing.
  - s denotes the pixel intensity after processing.

Spatial Domain Methods

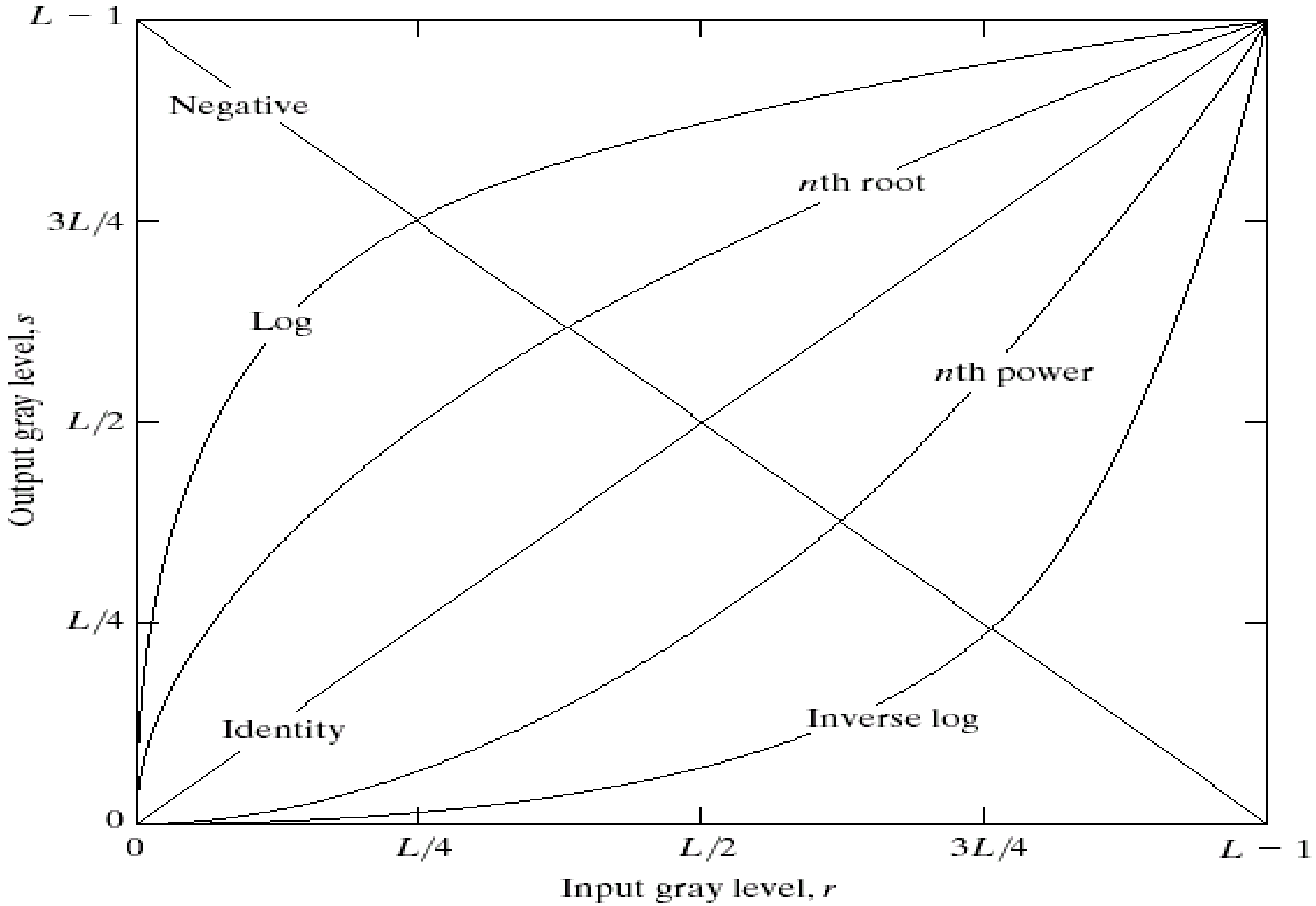
Intensity Transformations

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

- Implemented via Look-Up Tables (LUT) where values of T are stored in a 1-D array (for 8-bit, LUT will have 256 values)

Spatial Domain Methods

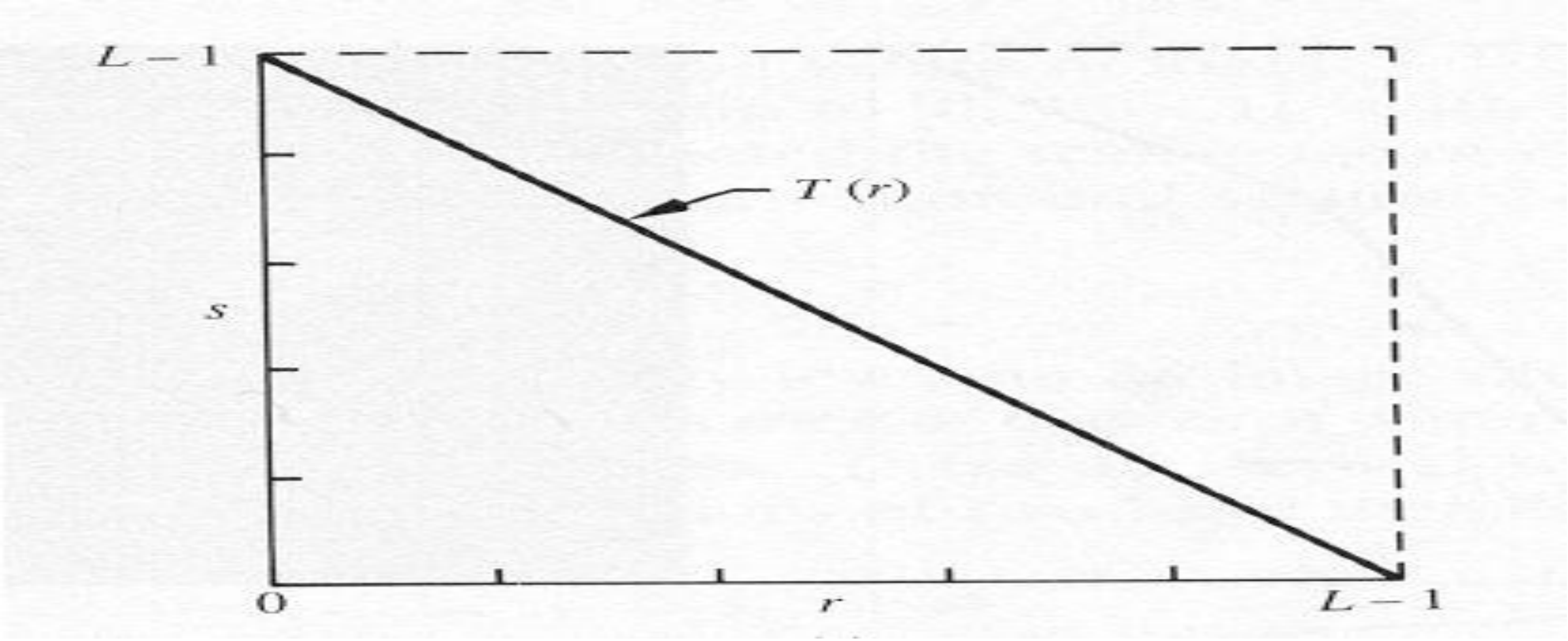
FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Spatial Domain Methods

Image Negatives

- Are obtained by using the transformation function s=T(r).



Spatial Domain Methods

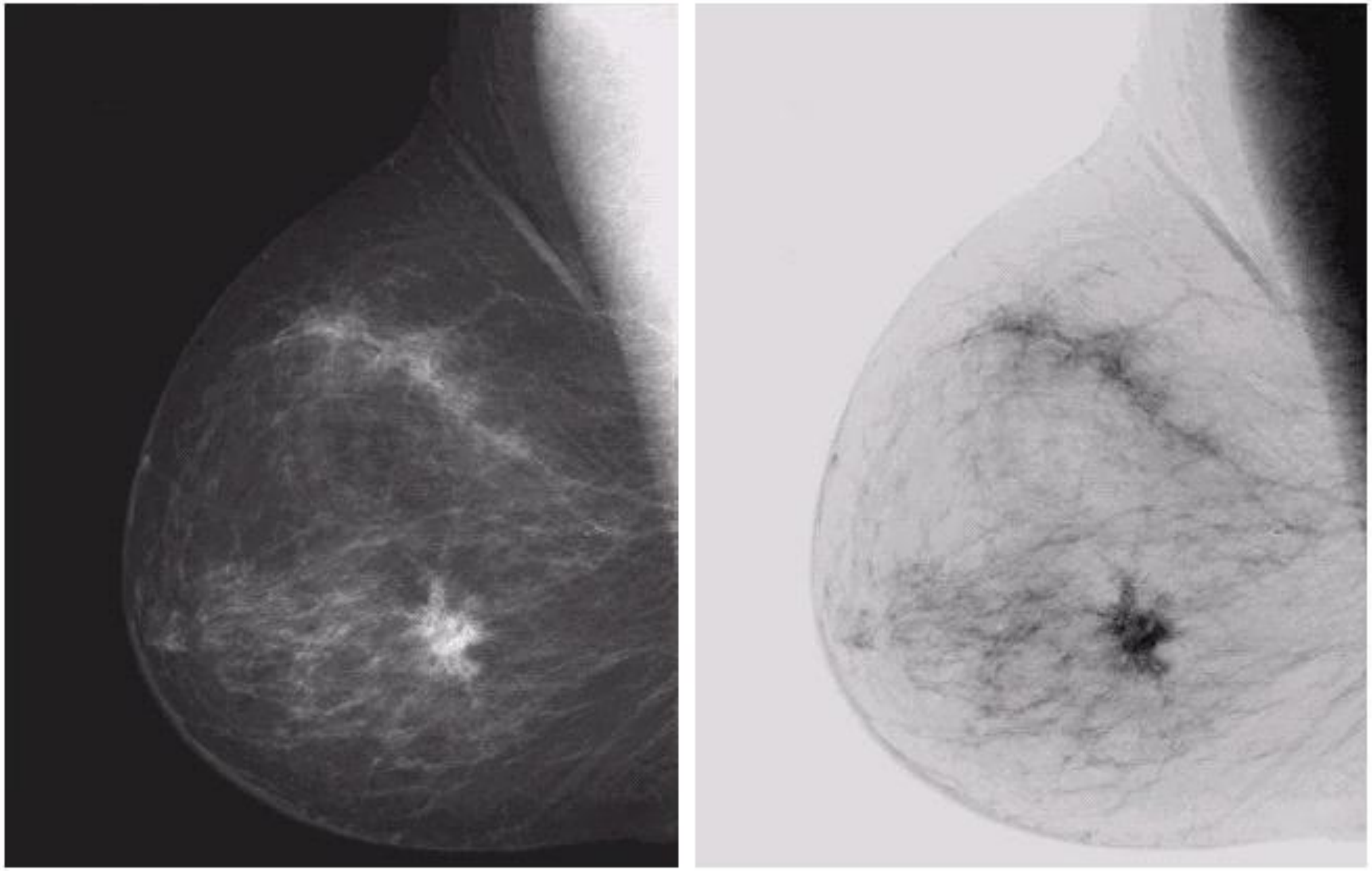


FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Spatial domain

- Function reverses the order from black to white so that the intensity of the output image decreases as the intensity of the input increases.

- Used mainly in medical images and to produce slides of the screen.

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

c: constant

- Compresses the dynamic range of images with large variations in pixel values

Spatial Domain Methods

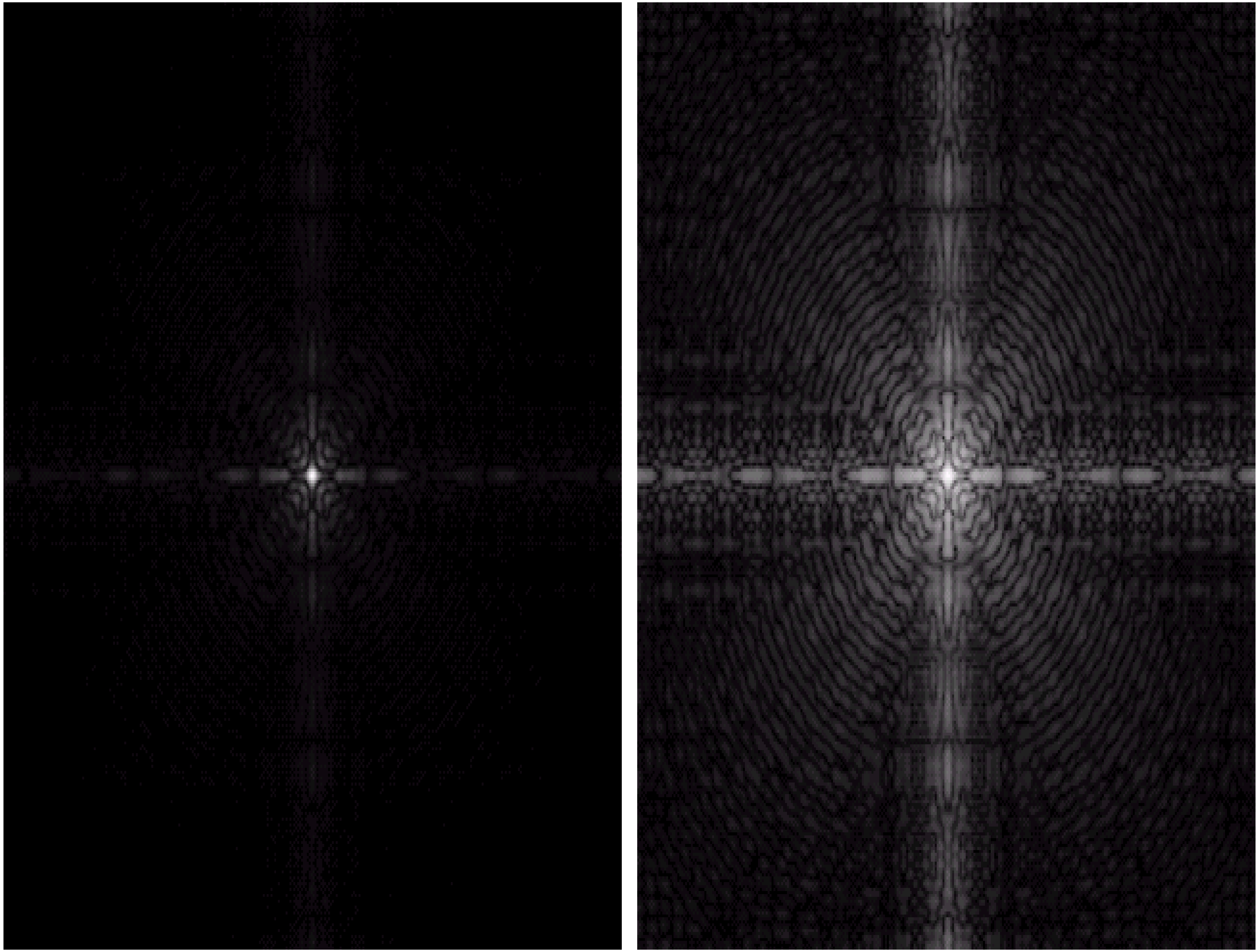


FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.

Spatial Domain Methods

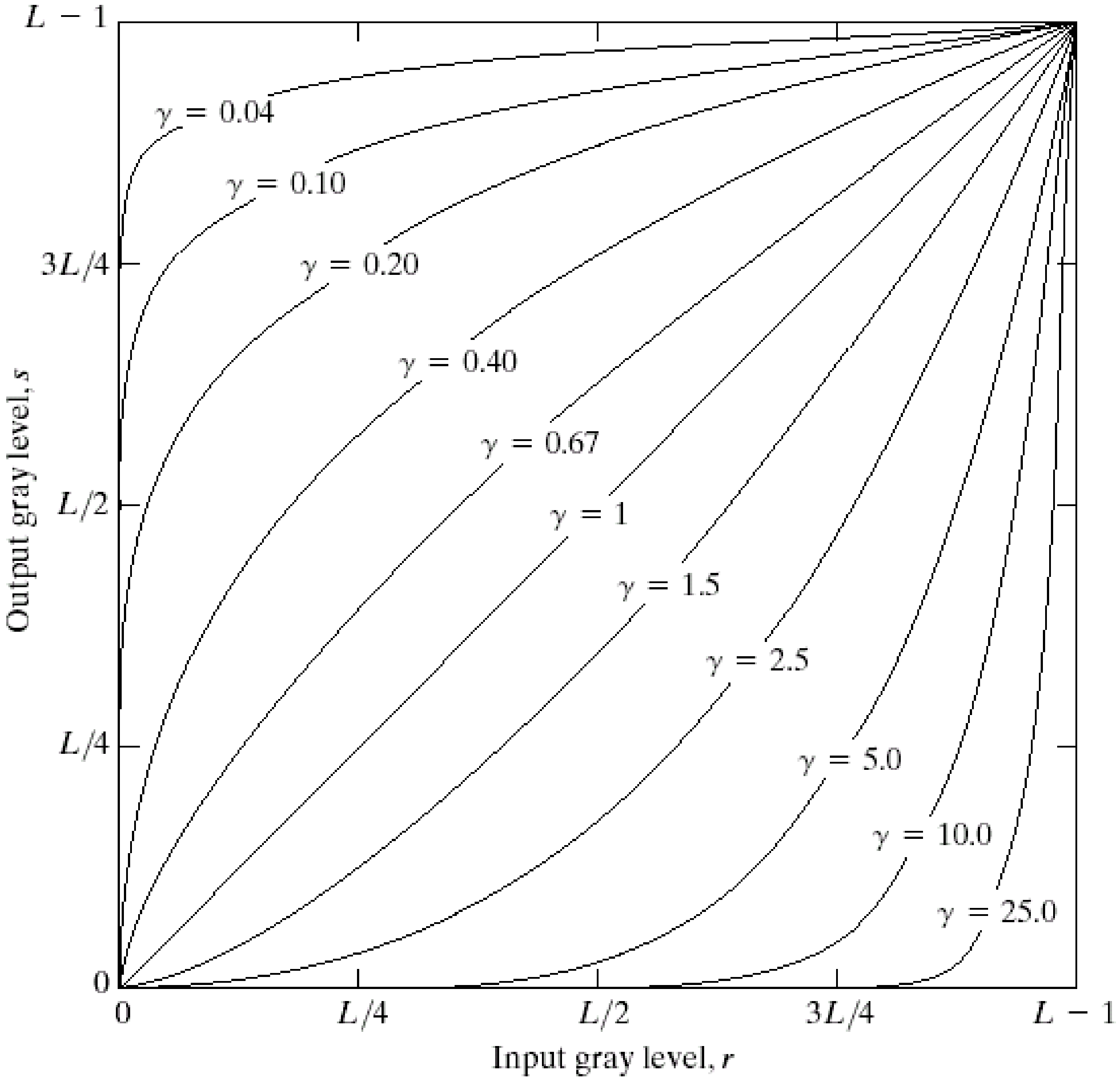
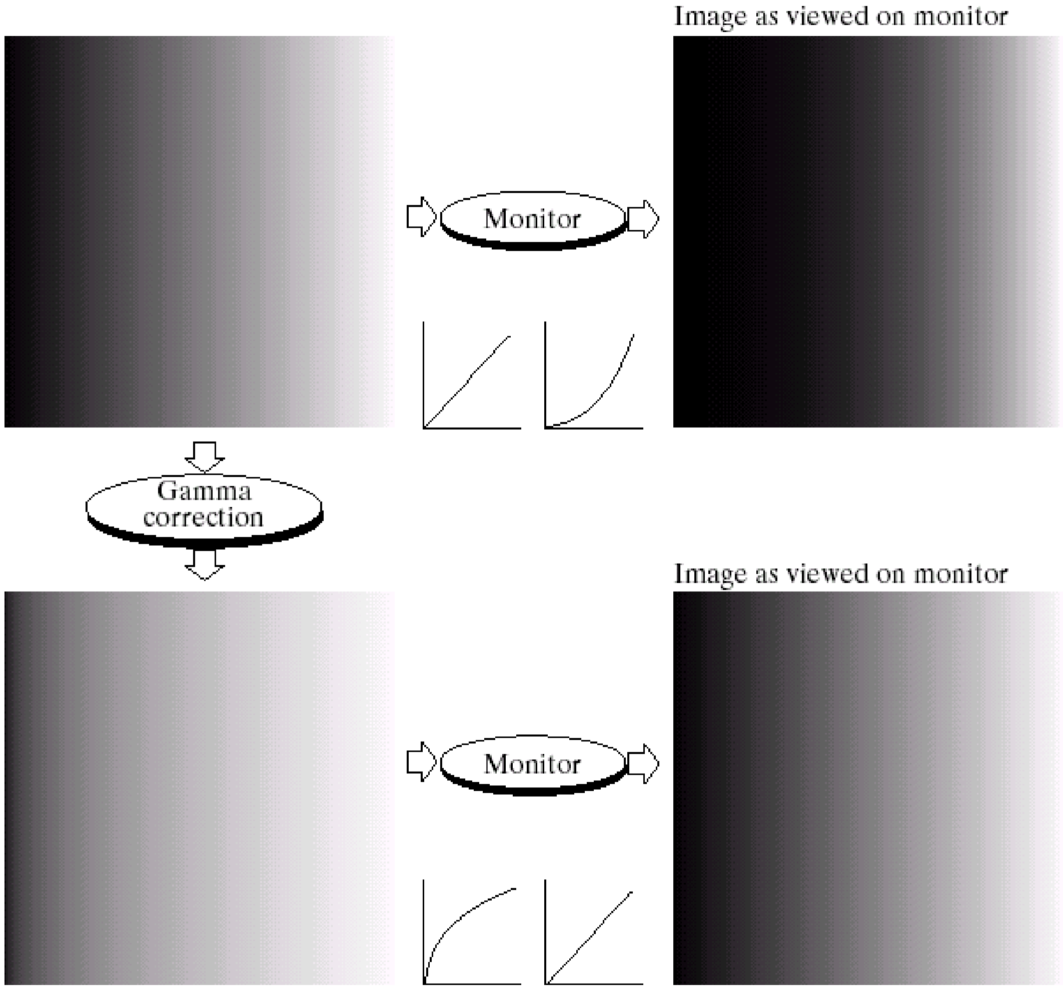


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

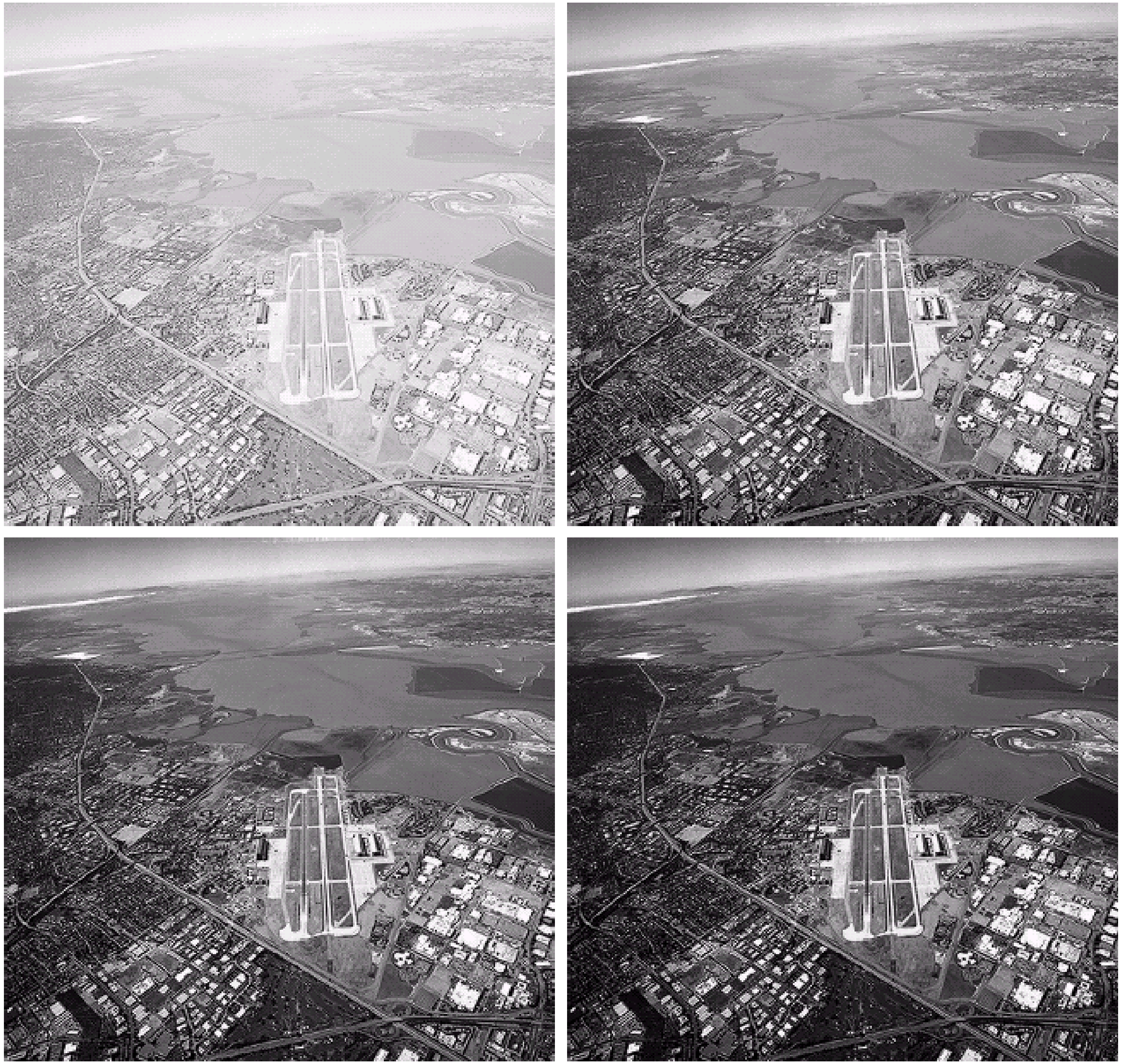
Spatial Domain Methods



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.

Spatial Domain Methods

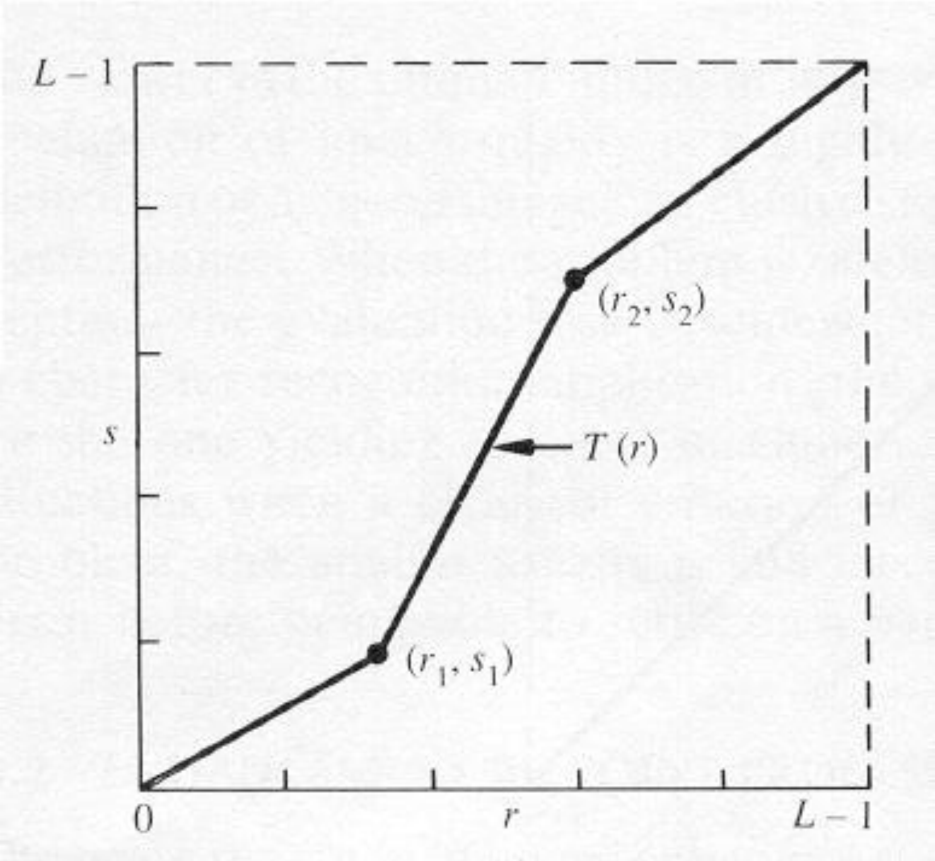


a b
c d

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
Contrast Stretching

- To increase the dynamic range of the gray levels in the image being **processed.**



Contrast Stretching

- The locations of (r_1,s_1) and (r_2,s_2) control the shape of the transformation function.
  - If $r_1= s_1$ and $r_2= s_2$ the transformation is a linear function and produces no changes.
  - If $r_1=r_2$, $s_1=0$ and $s_2=L-1$, the transformation becomes a thresholding function that creates a binary image.

Contrast Stretching

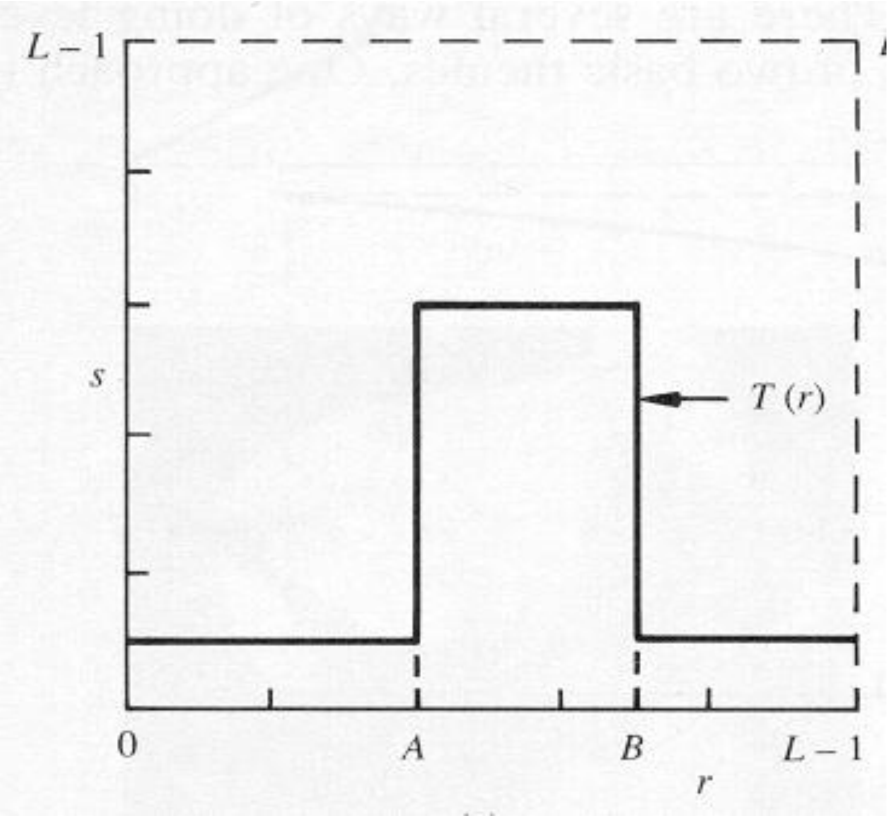
- More on function shapes:

  - Intermediate values of (r_1,s_1) and (r_2,s_2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.

  - Generally, $r_1 \leq r_2$ and $s_1 \leq s_2$ is assumed.

Gray Level Slicing

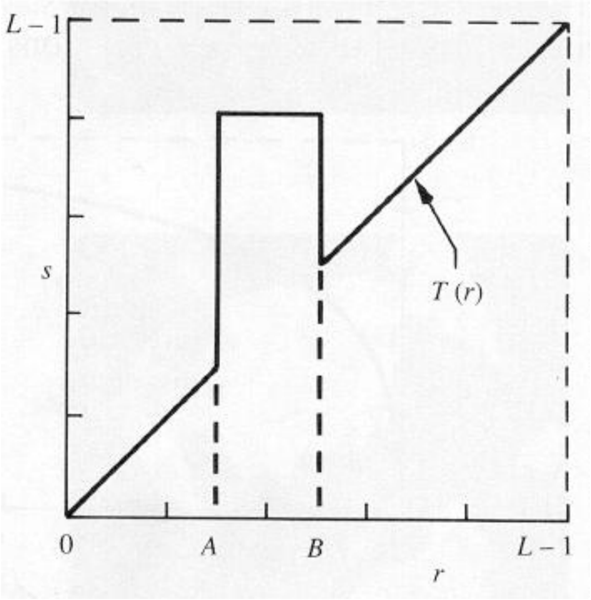
- To highlight a specific range of gray levels in an image (e.g. to enhance certain features).



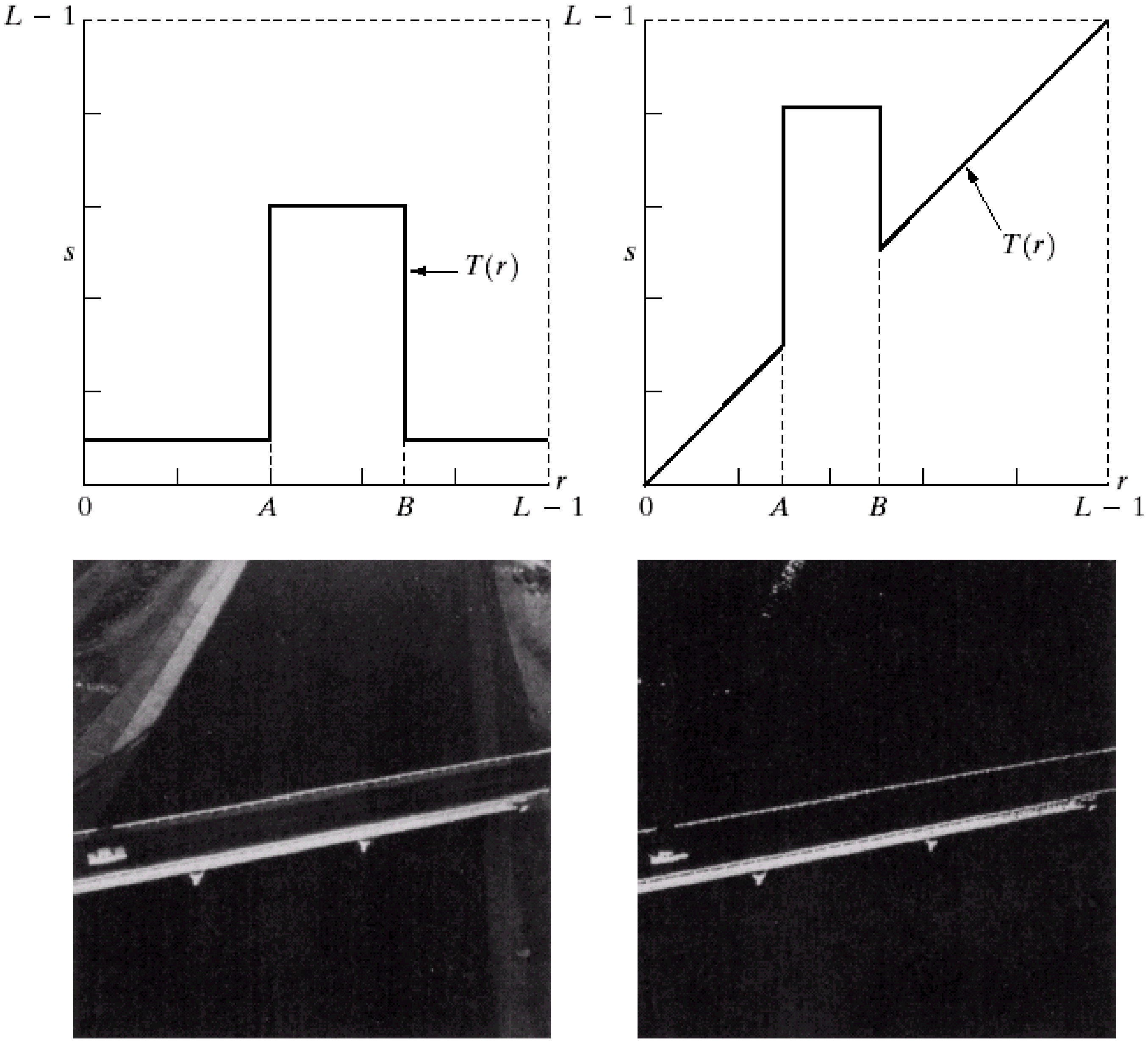
One way is to display a high value for all gray levels in the range of interest and a low value for all other gray levels (binary image).

Gray Level Slicing

- The second approach is to brighten the desired range of gray levels but preserve the background and gray-level tonalities in the image:



Spatial Domain Methods



a b
c d

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

Bit Slicing

- To highlight the contribution made to the total image appearance by bits.
  - i.e. Assuming that each pixel is represented by 8 bits, the image is composed of 8 1-bit planes.
  - Plane 0 contains the least significant bit and plane 7 contains the most significant bit.

Bit Plane Slicing

- More on bit planes:
  - Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details.
  - Plane 7 corresponds exactly with an image thresholded at gray level 128.

Spatial Domain

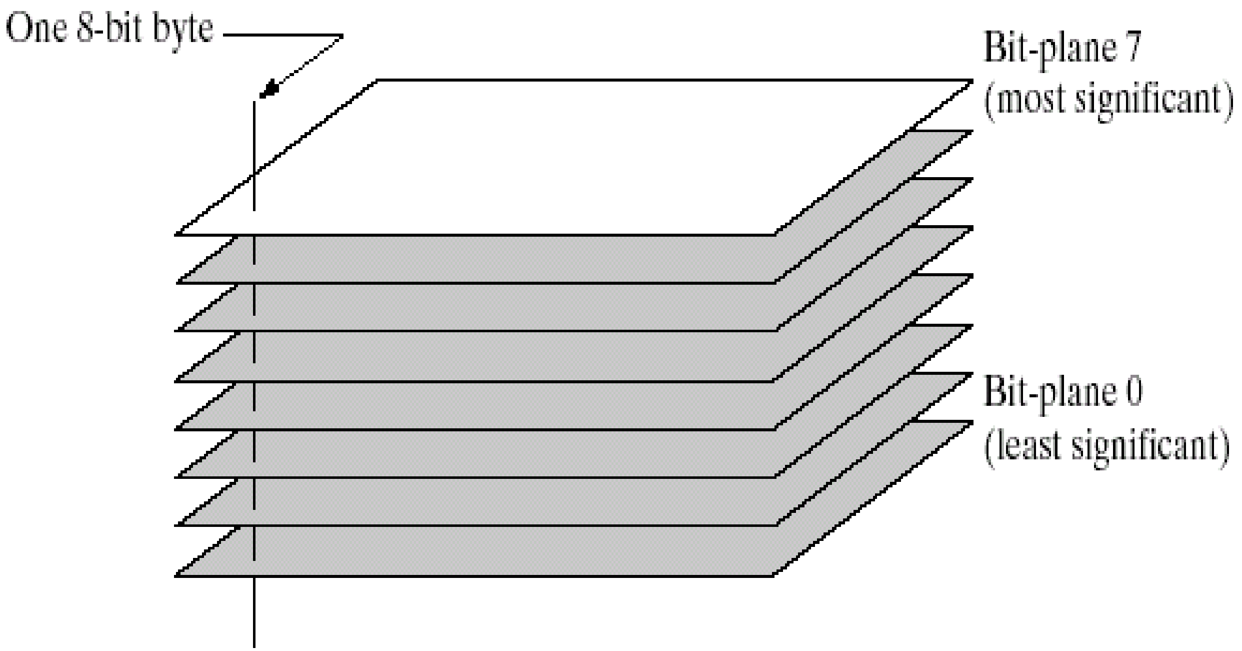


FIGURE 3.12 Bit-plane representation of an 8-bit image.

Spatial Domain Methods

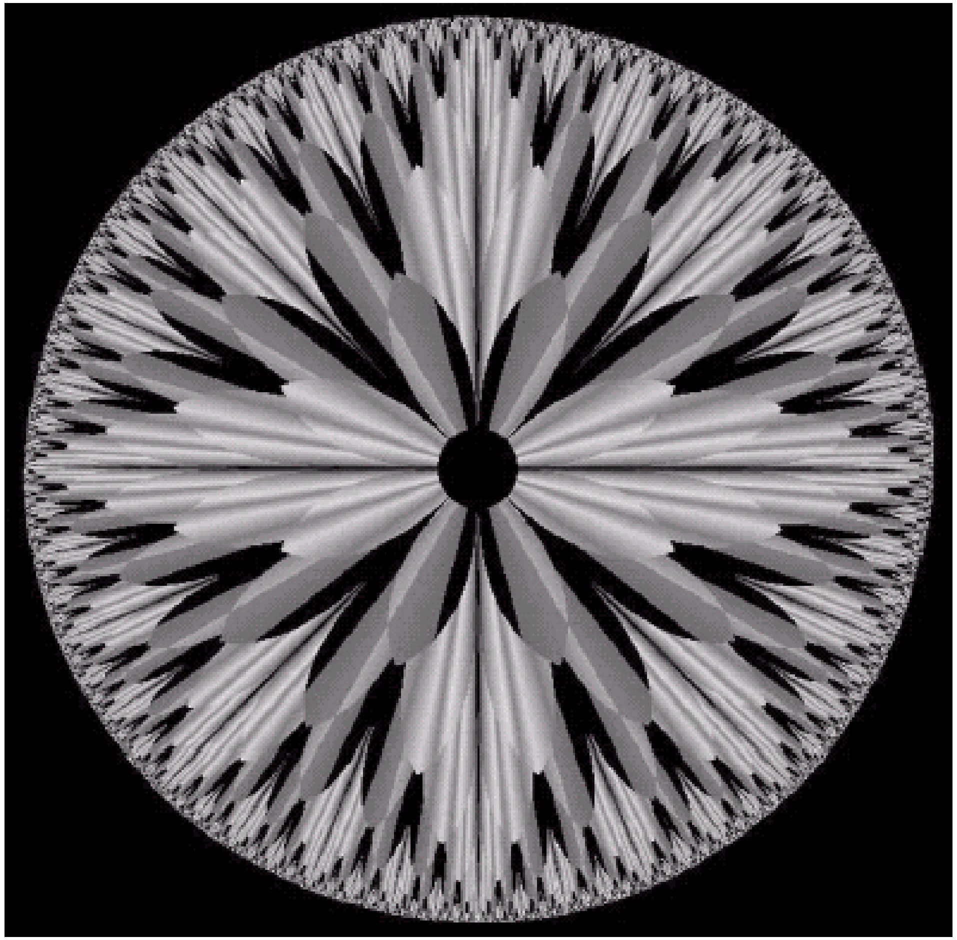


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

Spatial Domain Methods

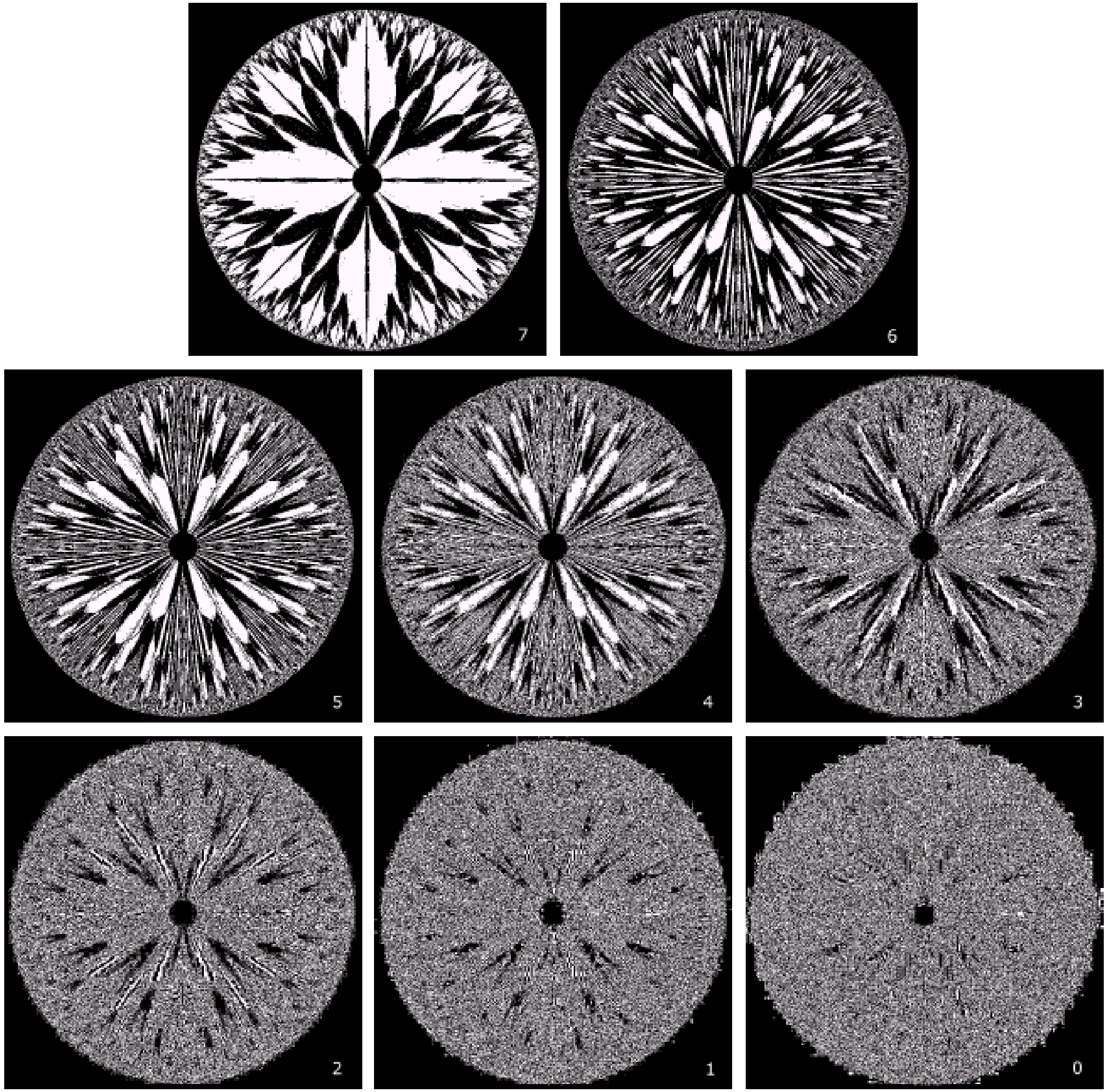


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Spatial Domain Filtering

averaging filters

lowpass filters

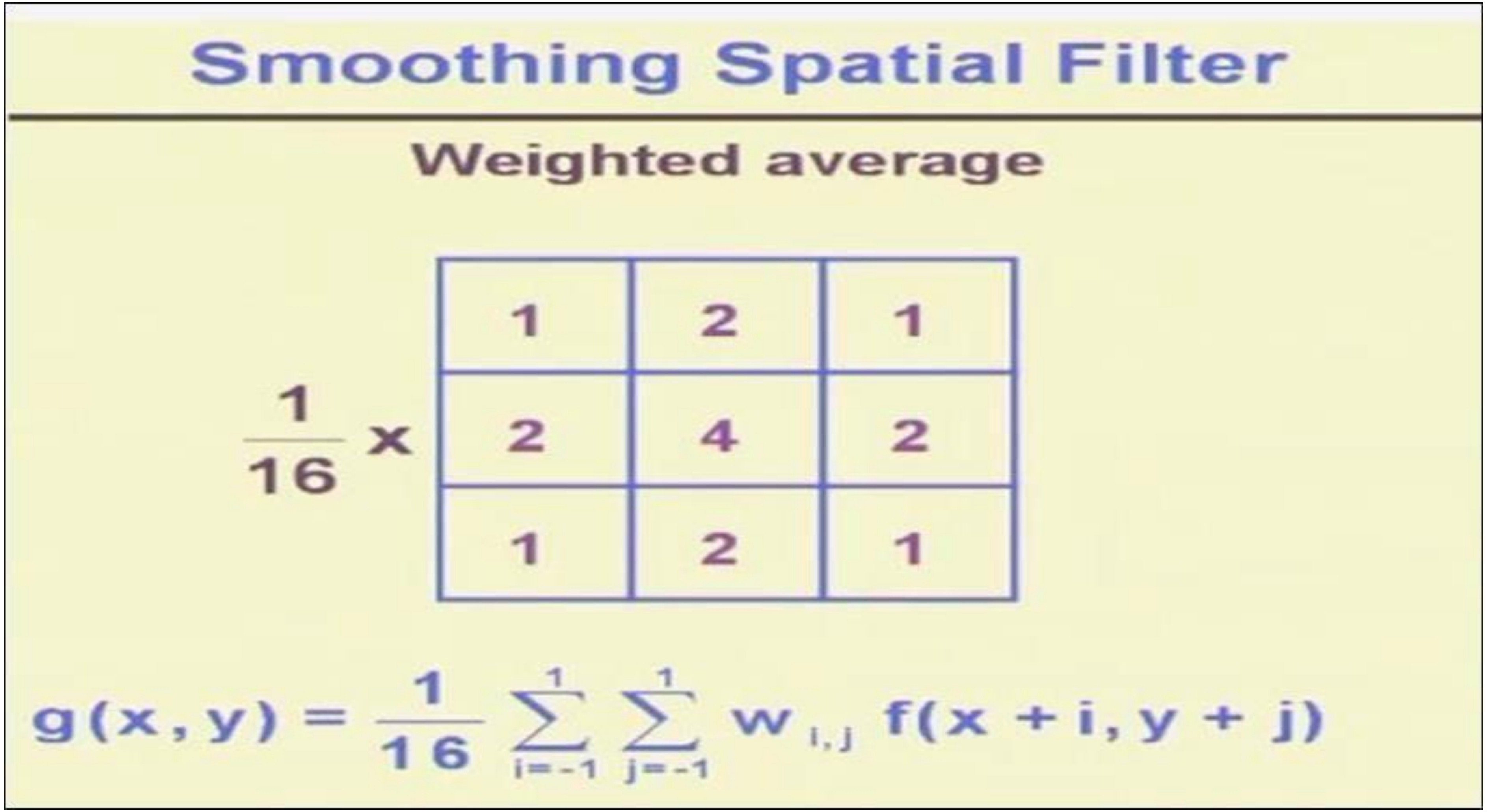
$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

a b

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

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

Spatial Domain Filtering



Smoothing Spatial Filter

Weighted average

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

$$g(x,y) = \frac{1}{16} \sum_{i=-1}^{1} \sum_{j=-1}^{1} w_{i,j}\, f(x+i, y+j)$$

Spatial Domain Filtering

Examples of Low Pass Masks (Local Averaging

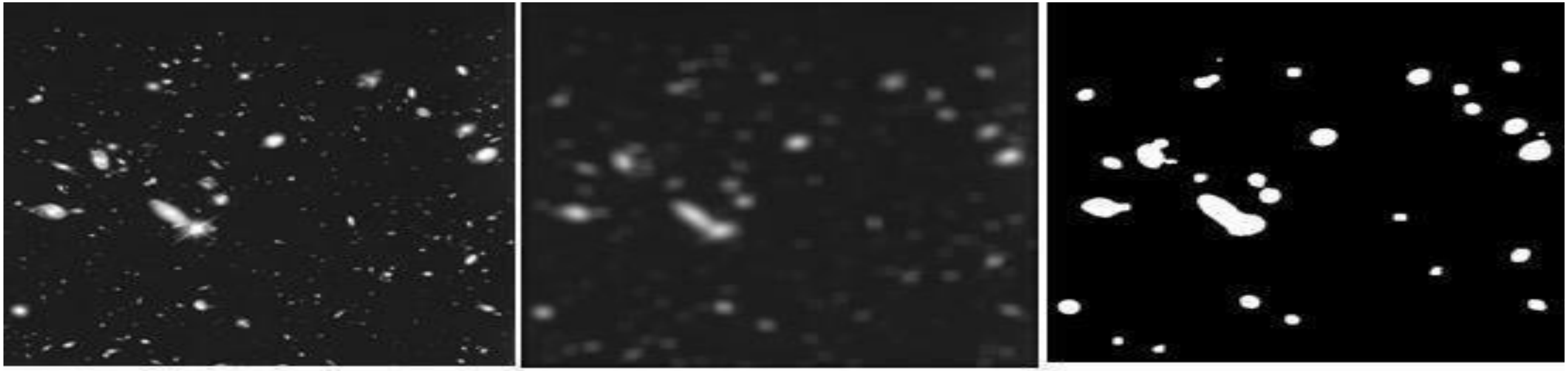


Image from Hubble telescope, image processed by 5x5 averaging window, and image after thresholding (nasa)

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1&1&1\\\hline 1&1&1\\\hline 1&1&1\\\hline\end{array}$$

$$\frac{1}{25}\times\begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline\end{array}$$

$$\frac{1}{49}\times\begin{array}{|c|c|c|c|c|c|c|}\hline 1&1&1&1&1&1&1\\\hline 1&1&1&1&1&1&1\\\hline 1&1&1&1&1&1&1\\\hline 1&1&1&1&1&1&1\\\hline 1&1&1&1&1&1&1\\\hline 1&1&1&1&1&1&1\\\hline 1&1&1&1&1&1&1\\\hline\end{array}$$

Spatial Domain Filtering

- The most popular masks for low pass filtering are masks with all their coefficients positive and equal to each other as for example the mask shown below. Moreover, they sum up to 1 in order to maintain the mean of the image.

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Gaussian filtering

- The two dimensional Gaussian mask has values that attempts to approximate the continuous function. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. The following shows a suitable integer-valued convolution kernel that approximates a Gaussian with a of 1.0.

Non-Linear Filters

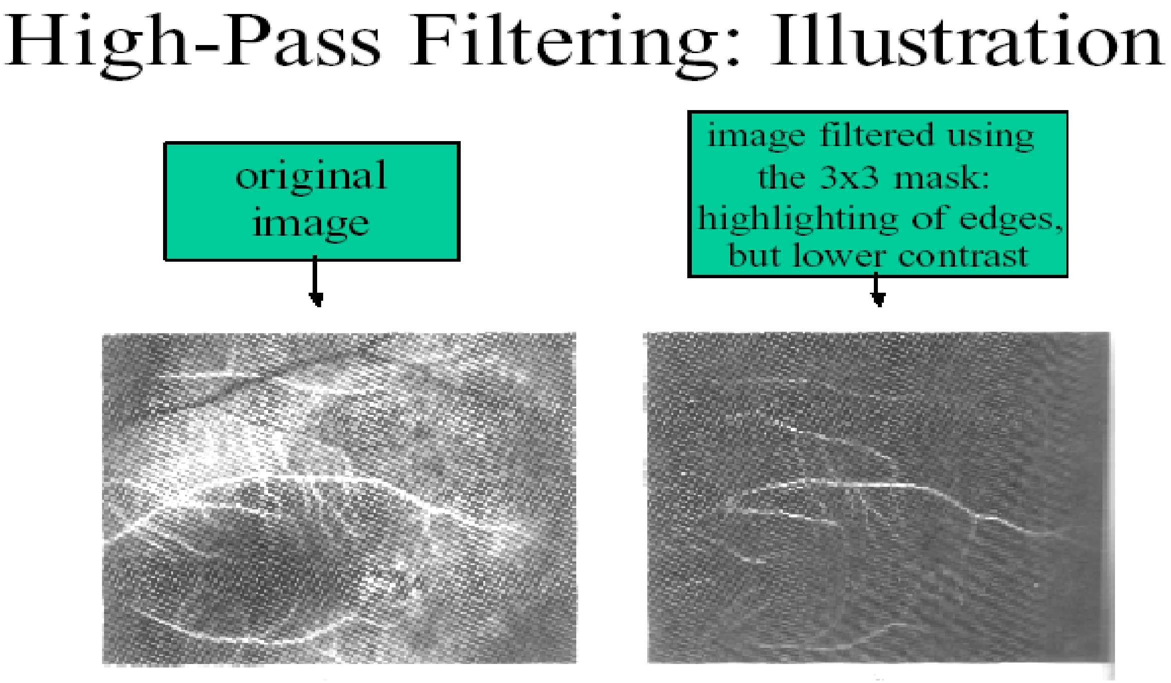
➢The best-known example in this category is the *Median filter*, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median).

➢Order static filter / ;ŶoŶ-liŶeaƌ filter) / median filter Objective:Replace the valve of the pixel by the median of the intensity values in the neighbourhood of that pixel

➢Although the median filter is by far the most useful order-statistics filter in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but the reader will recall from basic statistics that ranking lends itself to many other possibilities.

Non-Linear Filters

High pass = Original – Low pass



High-Pass Filtering: Illustration

original image

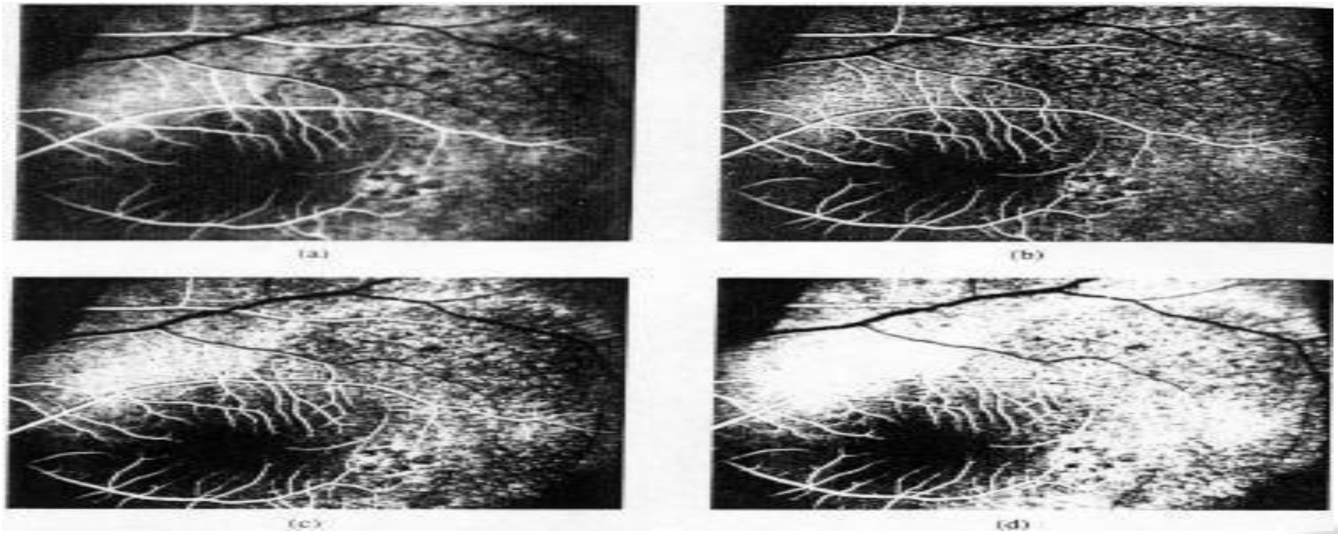
image filtered using the 3x3 mask: highlighting of edges, but lower contrast

Non-Linear Filters

High pass filtered image may be computed as the difference between the original image and a lowpass filtered version of that image as follows

Highpass = Original – Lowpass

- $$\begin{aligned} Highboost &= A(Original) - Lowpass \\ &= (A-1)(Original) + Original - Lowpass \\ &= (A-1)(Original) + Highpass \end{aligned}$$



Non-Linear Filters

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & A & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \frac{1}{9} \times \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & -1 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \qquad \frac{1}{9} \times \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 9A-1 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

High-Pass Filtering

- Shape of impulse response: +ve coefficients near its centre, -ve coefficients in periphery.
- E.g. 3x3 mask with +ve value in the middle, surrounded by 8 neighbours of –ve values.

$$1/9 \text{ x } \quad \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Sharpening Spatial Filters

- To integration, it is logical to conclude that sharpening could be accomplished by spatial differentiation.

- This section deals with various ways of defining and implementing operators for Image sharpening by digital differentiation.

- Fundamentally, the strength 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 (such as noise) and deemphasizes areas with slowly varying gray- level values.

Sharpening Non linear Filter

Dimensional high pass spatial filters:

An edge is the boundary between two regions with relatively distinct grey level properties. The idea underlying most edge detection techniques is the computation of a local derivative operator.

The magnitude of the first derivative calculated within a neighborhood around the pixel of interest, can be used to detect the presence of an edge in an image.

First derivatives in image processing are implemented using the magnitude of the gradient.

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

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \text{mag}(\nabla \mathbf{f}) \\ &= [G_x^2 + G_y^2]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}. \end{aligned}$$

Spatial Filters

Sharpening Spatial Filters:

derivative

(1)must be zero in flat segments (areas of constant gray-level values);

(2)must be nonzero at the onset of a gray-level step or ramp; and
(3)must be nonzero along ramps.

Similarly, any definition of a second derivative

(1)must be zero in flat areas;

(2)must be nonzero at the onset and end of a gray-level step or ramp;
(3)must be zero along ramps of constant slope.

Spatial Filters

$$\frac{\partial^2 f}{\partial^2 x^2} + \frac{\partial^2 f}{\partial^2 y^2} = \nabla^2 f = \left[f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)\right] - 4f(x,y).$$

Laplacian operator (for enhancing fine details)

The Laplacian of a 2-D function $f(x,y)$ is a second order derivative defined as

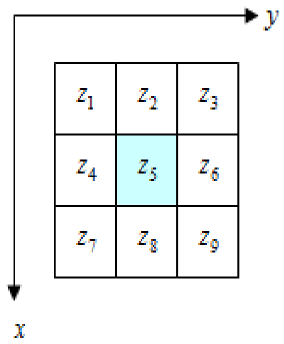
$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

In practice it can be also implemented using a 3x3 mask

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

Consider a pixel of interest $f(x,y) = z_5$ and a rectangular neighborhood of size $3 \times 3 = 9$ pixels (including the pixel of interest) as shown below.

The main disadvantage of the Laplacian operator is that it produces double edges



z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Spatial Filters

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

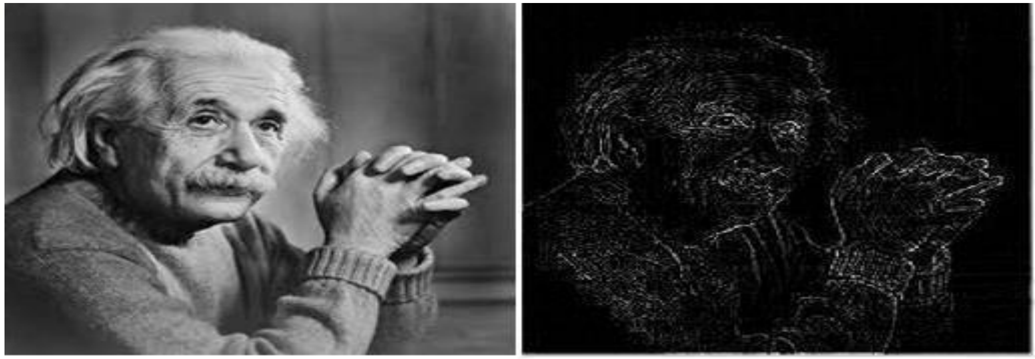
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

(a) Filter mask used to implement the digital Laplacian

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

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

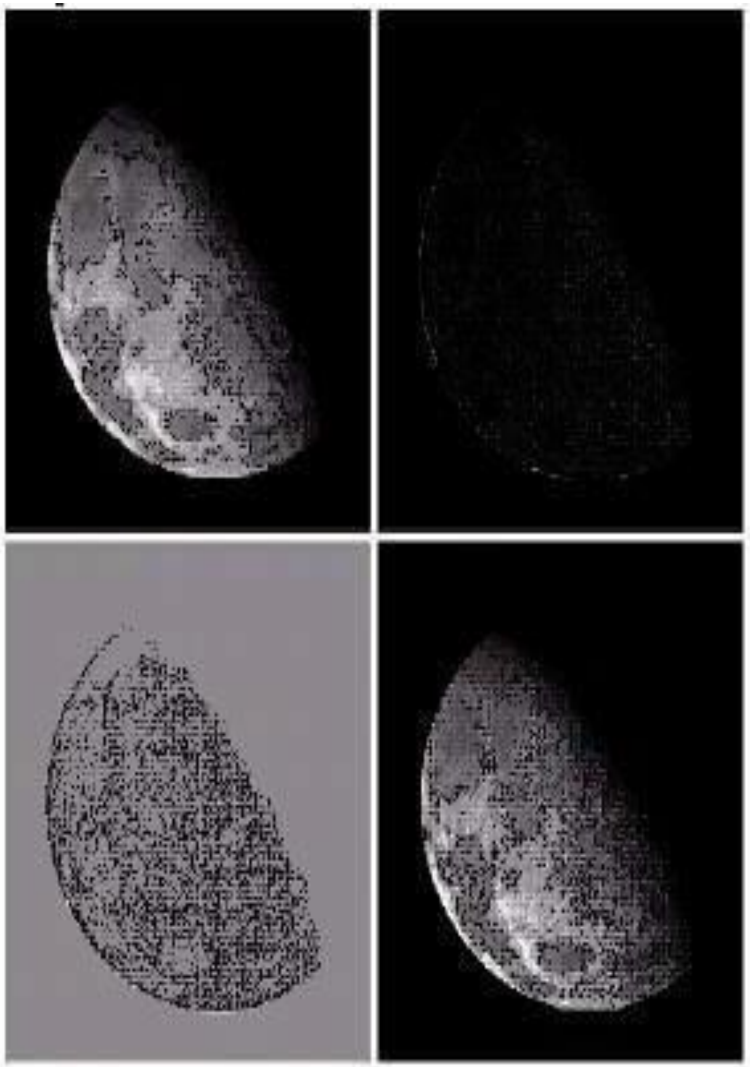


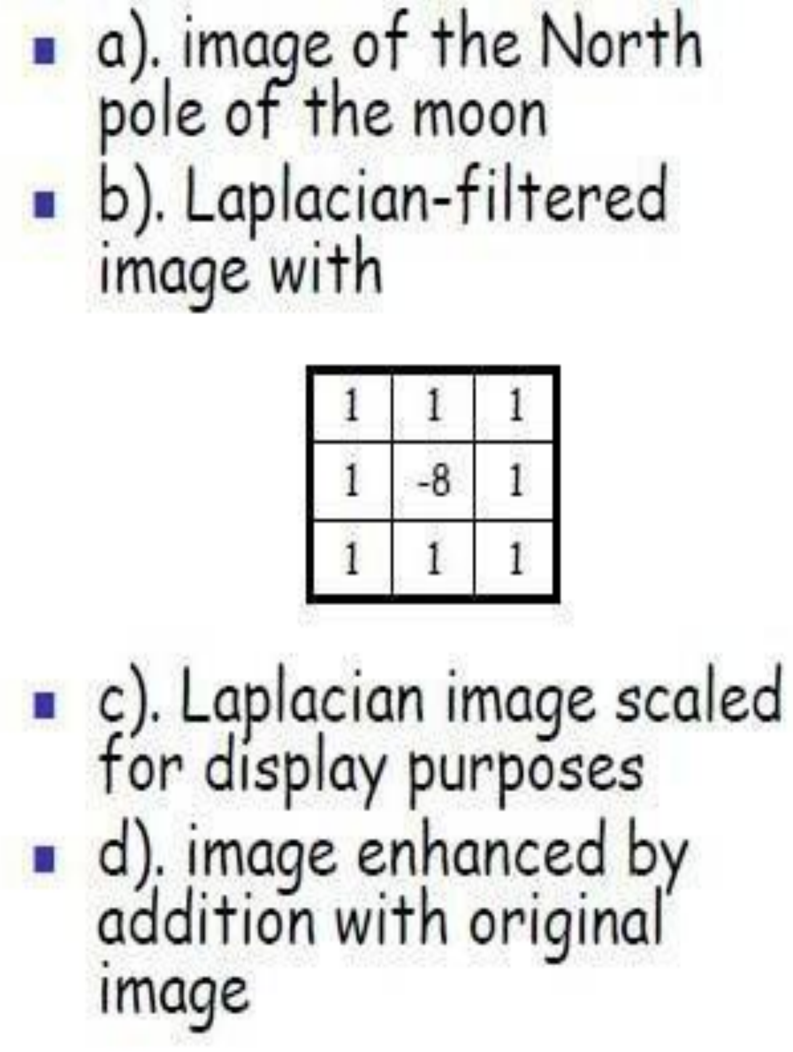
Spatial Filters

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Spatial Filters





- a). image of the North pole of the moon
- b). Laplacian-filtered image with

1	1	1
1	-8	1
1	1	1

- c). Laplacian image scaled for display purposes
- d). image enhanced by addition with original image

Use of first derivatives for Image Sharpening (Non linear)

About two dimensional high pass spatial filters

An edge is the boundary between two regions with relatively distinct grey level properties. The idea underlying most edge detection techniques is the computation of a local derivative operator.

The magnitude of the first derivative calculated within a neighborhood around the pixel of interest, can be used to detect the presence of an edge in an image.

First derivatives in image processing are implemented using the magnitude of the gradient.

For a function f(x, y), the gradient of *f* at coordinates (x, y) is defined as the two- dimensio

Spatial Filters

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\[2ex] \dfrac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \operatorname{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}. \end{aligned}$$

Spatial Filters

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Gradient Mask

- simplest approximation, 2x2

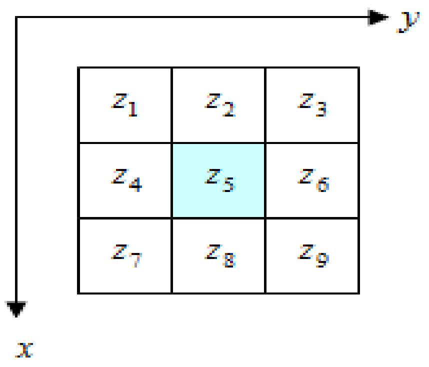
$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

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

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

Spatial Filters

Consider a pixel of interest $f(x,y)=z_5$ and a rectangular neighborhood of size $3\times3=9$ pixels (including the pixel of interest) as shown below.



z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Roberts operator

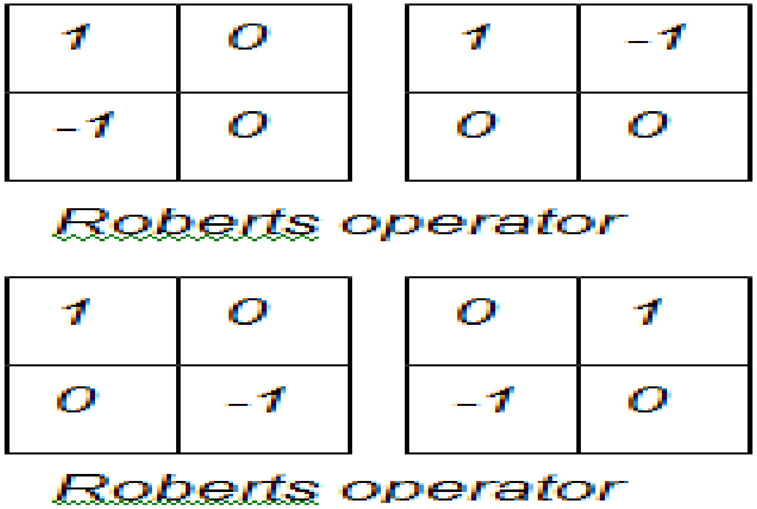
Above Equation can be approximated at point Z_5 in a number of ways. The simplest is to use the difference $(Z_5 - Z_8)$ in the x direction and $(Z_5 - Z_6)$ in the y direction. This approximation is known as the **Roberts** operator, and is expressed mathematically as follows

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

Another approach for approximating the equation is to use cross differences

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

Spatial Filters



1	0
-1	0

1	-1
0	0

Roberts operator

1	0
0	-1

0	1
-1	0

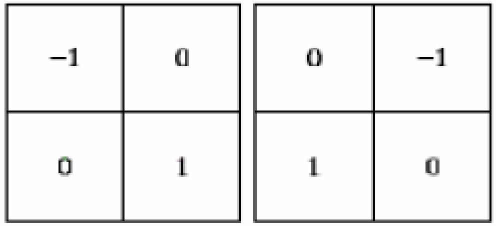
Roberts operator

- 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

Spatial Filters

$\nabla f\,(x, y)$ r approximation to the abtrix is:

$$\nabla f \approx \left|(z_7+z_8+z_9)-(z_1+z_2+z_3)\right| + \left|(z_3+z_6+z_9)-(z_1+z_4+z_7)\right|$$

The difference between the first and third rows approximates the derivative in the *x*
direction

•The difference between the first and third columns approximates the derivative in the *y* *direction*

•The *Prewitt operator masks may be used to implement the* above approximation

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

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

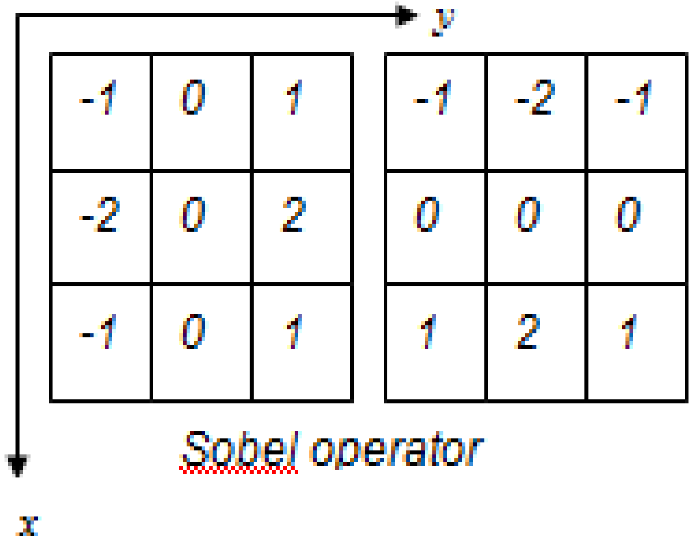
Spatial Filters

Sobel operator.

Definition and comparison with the Prewitt operator (gives weightage to centre pixel)
The most popular approximation of equation (1) but using a 3×3 mask is the following.

$$\nabla f \cong \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

This approximation is known as the Sobel operator.



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

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

Sobel operator

If we consider the left mask of the Sobel operator, this causes differentiation along the y direction.

Spatial Linear Filters

- Sobel operators, 3x3

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

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

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

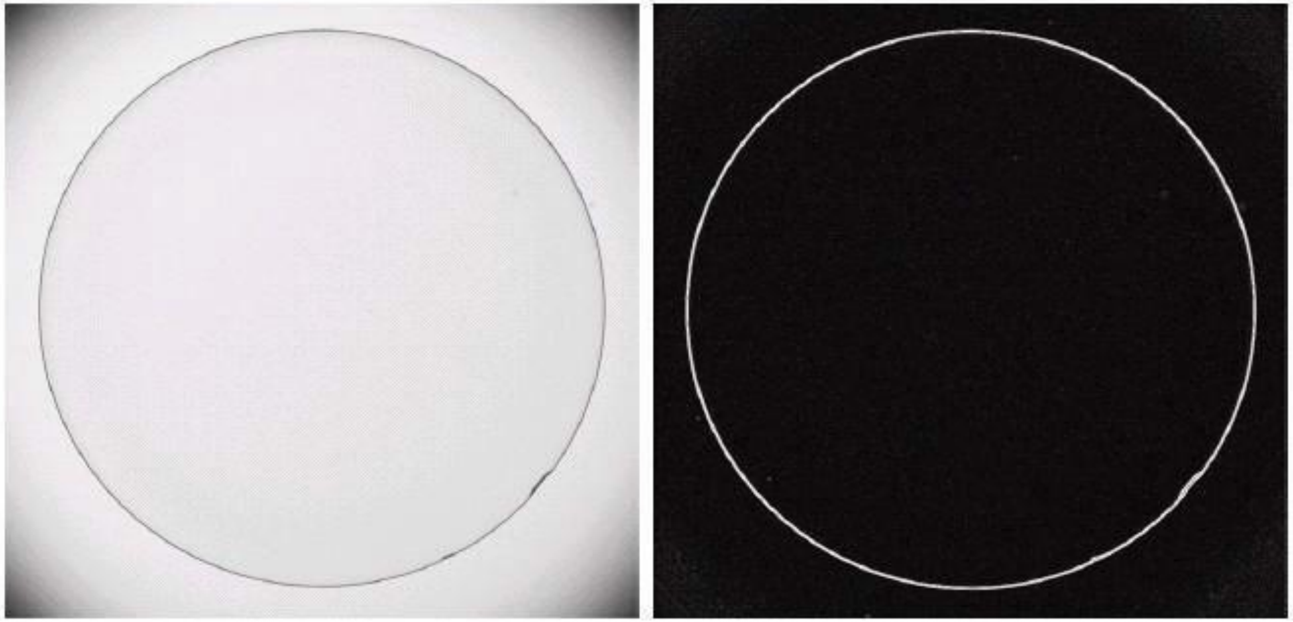
the weight value 2 is to achieve smoothing by giving more important to the center point

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

150

Spatial Filters



a b

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

Filtering in the Frequency Domain

Filters in the frequency domain can be divided in four groups: Low pass filters

……….IMAGE BLUR

Remove frequencies away from the origin
Commonly, frequency response of these filters is symmetric around the origin;
The largest amount of energy is concentrated on low frequencies, but it represents just
image luminance and visually not so important part of image.

High pass filters …………EDGES DETECTION
Remove signal components around and further away from origin
Small energy on high frequency corresponds to visually very important image features such as edges and details. Sharpening = boosting high frequency pixels

Filtering in the Frequency Domain

Low pass filters (smoothing
filteIrdse)al Low Pass filters Butterworth low pass filters Gaussian low pass filters

ILPF
BLPF
GLPF

High Pass Filters (Sharpening
filters)
Ideal High pass filters Butterworth High pass filters Gaussian High pass filters Laplacian in frequency domain
High boost , high frequency emphasis filters

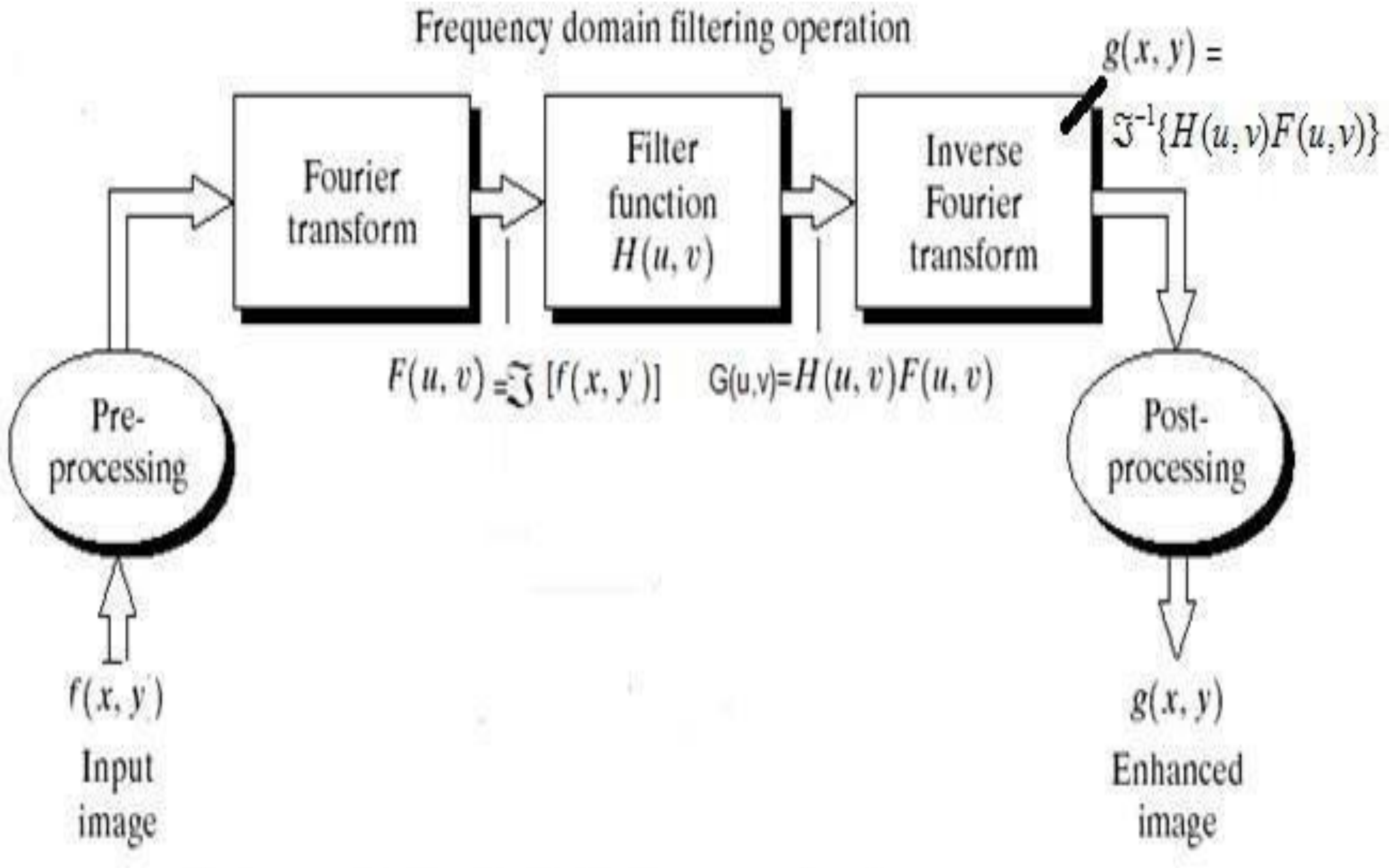
IHPF
BHPF
GHPF

Homomorthic filters log n or ln

f(x,y)= i(x,y) r(x,y)

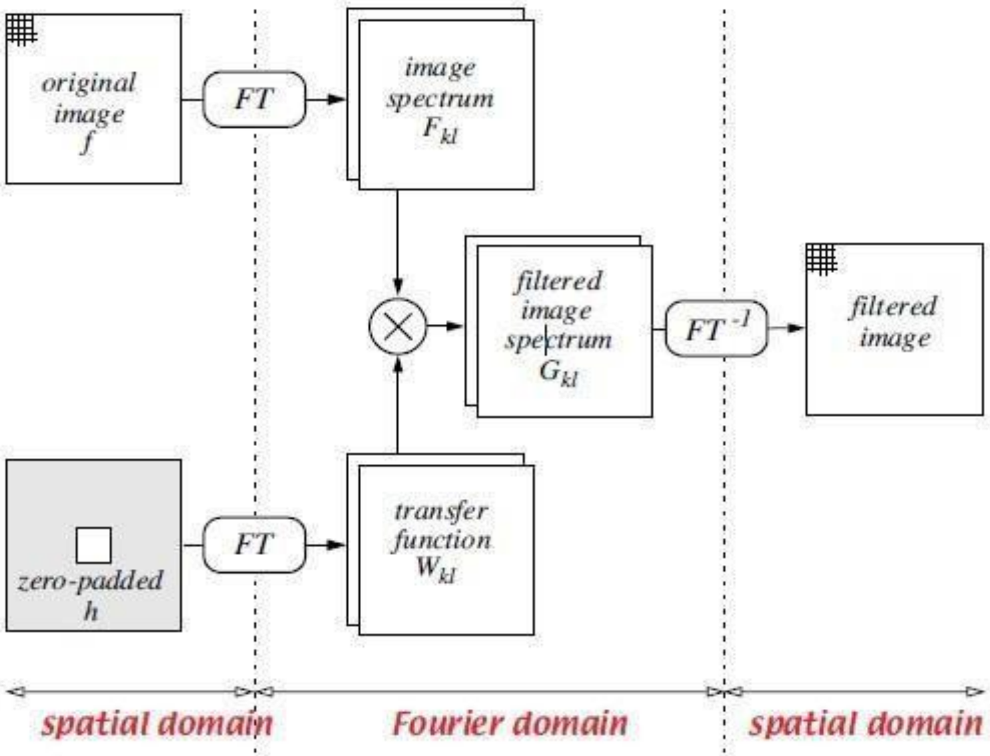
F[f(u,v)]= F [log n [i(x,y) r(x,y)] = F [log n [i(x,y)] + F [log n [r(x,y)]

Filtering in the Frequency Domain


Frequency domain filtering operation
$g(x, y) =$
$\Im^{-1}\{H(u,v)F(u,v)\}$
Fourier transform
Filter function $H(u, v)$
Inverse Fourier transform
$F(u, v) = \Im\,[f(x, y)]$
$G(u,v) = H(u, v)F(u, v)$
Pre-processing
Post-processing
$f(x, y)$
Input image
$g(x, y)$
Enhanced image

Basic steps for filtering in the frequency domain.

Filtering in the Frequency Domain



Note, multiplication in Fourier domain is full complex operation

Filtering in the Frequency Domain

Filtering in the Frequency Domain

- Basic Steps for zero padding

Zero Pad the input image f(x,y) to p =2M-1, and q=2N-1, if arrays are of same size.

If functions f(x,y) and h(x,y) are of size MXN and KXL, respectively, P шM + N - 1

Q шK + L- 1

Zero-pad h and f

- Pad both to at least
- Radix-2 FFT requires power of 2

For example, if M = N = 512 and K = L = 16, then P = Q = 1024

- Results in linear convolution
- Extract center MxN

Filtering in the Frequency Domain

1.Multiply the input padded image by (-1) $^{x+y}$ to center the transform.

2.Compute F(u,v), the DFT of the image from (1).

3.Multiply F(u,v) by a filter function H(u,v).

4.Compute the inverse DFT of the result in (3).

5.Obtain the real part of the result in (4).

6.Multiply the result in (5) by (-1)$^{x+y}$.

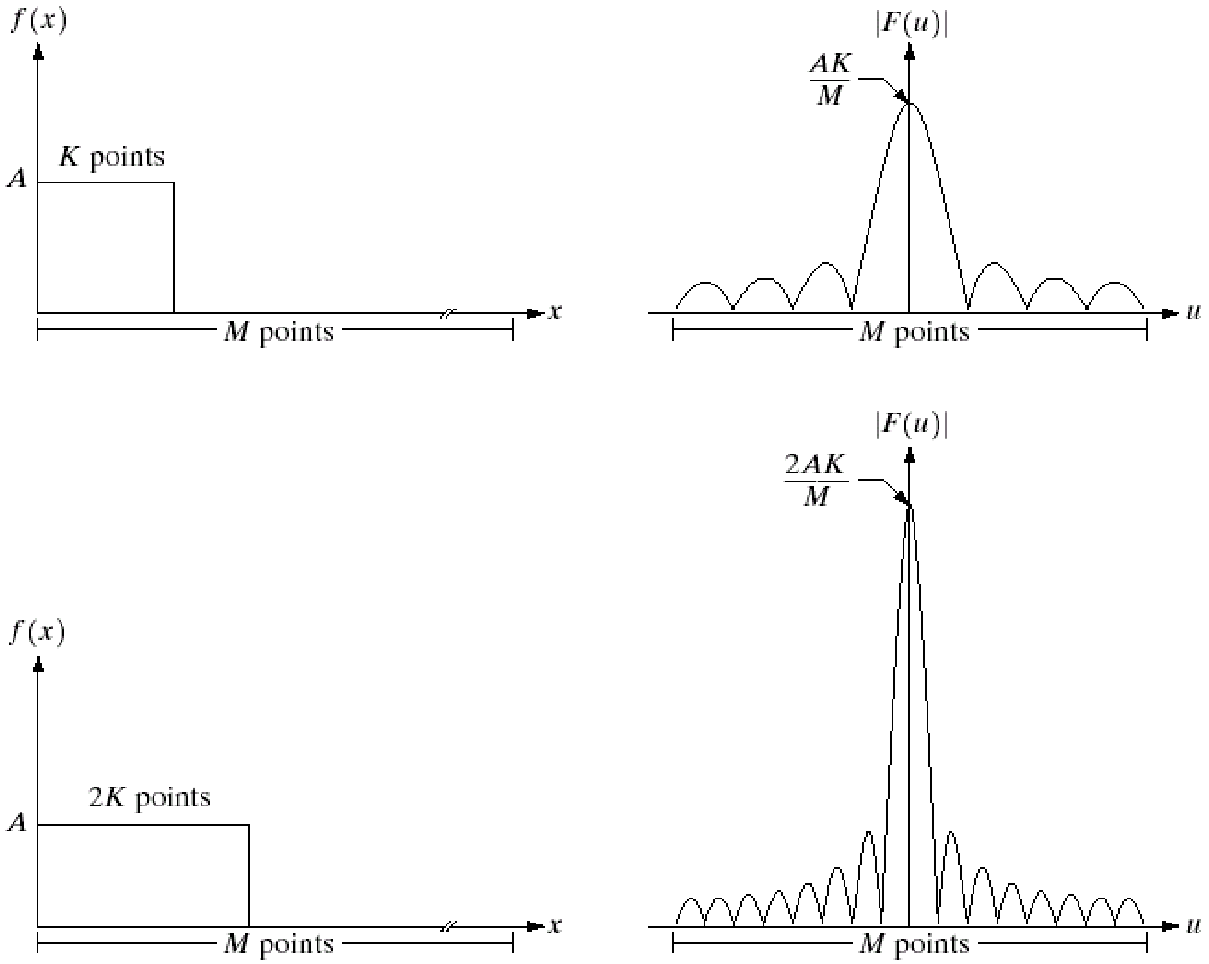
Given the filter H(u,v) (filter transfer function OR filter or filter function) in the frequency domain, the Fourier transform of the output image (filtered image) is given by:

G (u,v)= H (u,v) F (u,v) Step (3) is array multiplication

The filtered image g(x,y) is simply the inverse Fourier transform of G(u,v).

$$g_p(x, y) = \{\text{real}[\Im^{-1}[G(u, v)]]\}(-1)^{x+y}$$

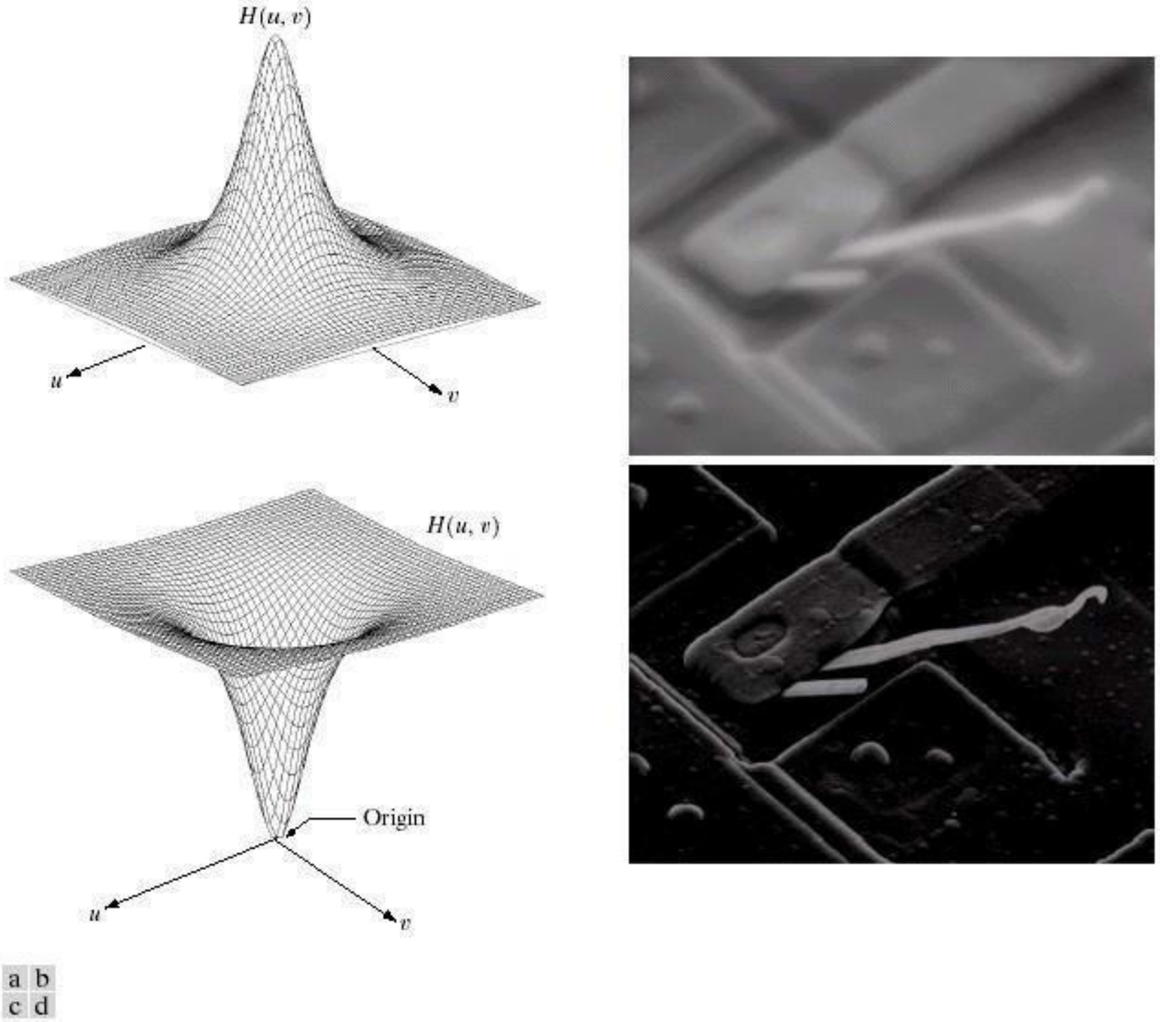
Filtering in the Frequency Domain



a b
c d

FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

Filtering in the Frequency Domain



Low Pass Filter

High Pass Filter

(a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image

Filtering in the Frequency Domain

- Since h(x,y) can be obtained from the response of a frequency domain filter to an impulse, h(x,y) spatial filter is some times referred as r finite impulse response filter (FIR) of H(u,v)

⟺

- f(x,y) * h (x,y) F(u,v) H(u,v)

- Spatial domain processing, in general, is faster than frequency domain processing.
- In some cases, it is desirable to generate spatial domain mask that approximates a given frequency domain filter.
- The following procedure is one way to create these masks in a
- *least square error sense.*
- Recall that filter processing in frequency domain, which is product of filter and function, becomes convolution of function and filter in spatial domain.

Filtering in the Frequency Domain

- ➢ This restriction in effect creates an n x n convolution mask $\hat{h}$, with Fourier transform of

$$\hat{H}(u,v) = \frac{1}{N}\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}\hat{h}(x,y)\,e^{-j2\pi\frac{ux+vy}{N}}$$

where u, v = 0, 1, 2, ..., N-1.

- ➢ The objective is to find the coefficients of $\hat{h}(x,y)$ such that mean square error is minimized.

$$e^2 = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\left|\hat{H}(u,v) - H(u,v)\right|^2$$

Filtering in the Frequency Domain

Consider the following filter transfer function:

$$H(u,v)=\begin{cases}0 & if\ (u,v)=(M/2,N/2)\\ 1 & otherwise\end{cases}$$

- This filter will set F(0,0) to zero and leave all the other frequency components. Such a filter is called the notch filter, since it is constant function with a hole (notch) at the origin.

Filtering in the Frequency Domain

HOMOMORPHIC FILTERING

an image can be modeled mathematically in terms of illumination and reflectance as follow:

$f(x,y) = I(x,y)\ r(x,y)$

Note that:

$F\{ f(x, y)\} \neq \quad F\{i(x, y)\}\ \ F\{r(x, y)\}$

To accomplish separability, first map the model to natural log domain and then take the Fourier transform of it. $z(x, y) = ln\{ f(x, y)\} = ln\{i(x, y)\}+ ln\{r(x, y)\}$ Then,

$F\{z(x, y)\} = F\{ln\ i(x, y)\}+ F\{ln\ r(x, y)\}$

or

$Z(u, v) = I(u, v) + R(u, v)$

Now, if we process Z(u,v) by means of a filter function H(u,v) then,

Filtering in the Frequency Domain

- Now, if we process Z(u,v) by means of a filter function H(u,v) then,

$$S(u,v) = H(u,v)Z(u,v) =$$
$$= H(u,v)I(u,v) + H(u,v)R(u,v)$$

- Taking inverse Fourier transform of S(u,v) brings the result back into natural log domain,

$$s(x,y) = F^{-1}\{S(u,v)\}$$
$$= F^{-1}\{H(u,v)I(u,v)\} + F^{-1}\{H(u,v)R(u,v)\}$$

By letting

$$i'(x,y) = F^{-1}\{H(u,v)I(u,v)\}$$
$$r'(x,y) = F^{-1}\{H(u,v)R(u,v)\}$$

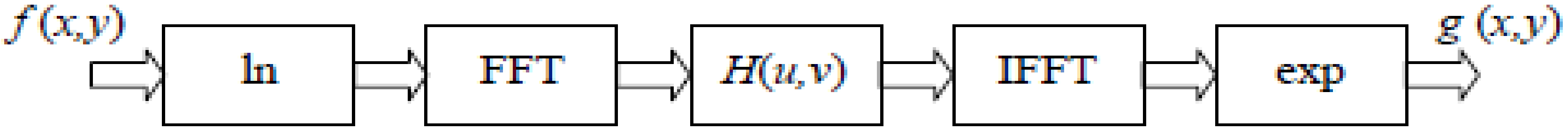
Filtering in the Frequency Domain

- Now, to get back to spatial domain, we need to get inverse transform of natural log, which is exponential,

$$
\begin{aligned}
s(x, y) &= i'(x, y) + r'(x, y) \\
g(x, y) &= \exp[s(x, y)] \\
&= \exp[i'(x, y)] \cdot \exp[r'(x, y)] \\
&= i_o(x, y) r_o(x, y)
\end{aligned}
$$

Where $i_o(x,y)$ is illumination and $r_o(x,y)$ is reflectance components of the output image.

- This method is based on a special case of a class of systems known as *homomorphic systems*.
- The overall model in block diagram will look as follow:



$f(x,y)$ → ln → FFT → $H(u,v)$ → IFFT → exp → $g(x,y)$

DIGITAL IMAGE PROCESSING

UNIT-III

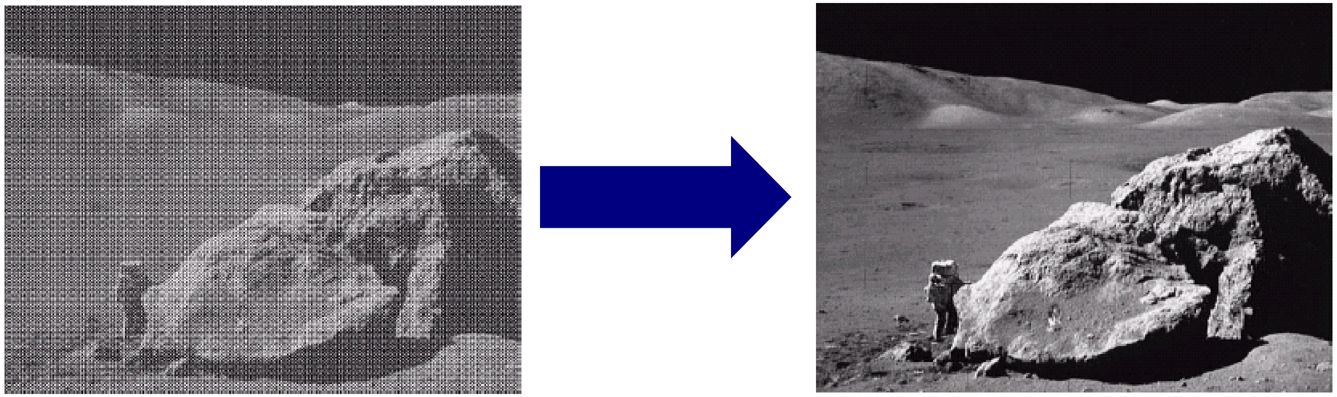
IMAGE RESTORATION

- Image restoration degradation model
- algebraic approach to restoration
- inverse filtering
- Least mean square filters
- constrained least square restoration
- interactive restoration

What is Image Restoration?

Image restoration attempts to restore images that have been degraded

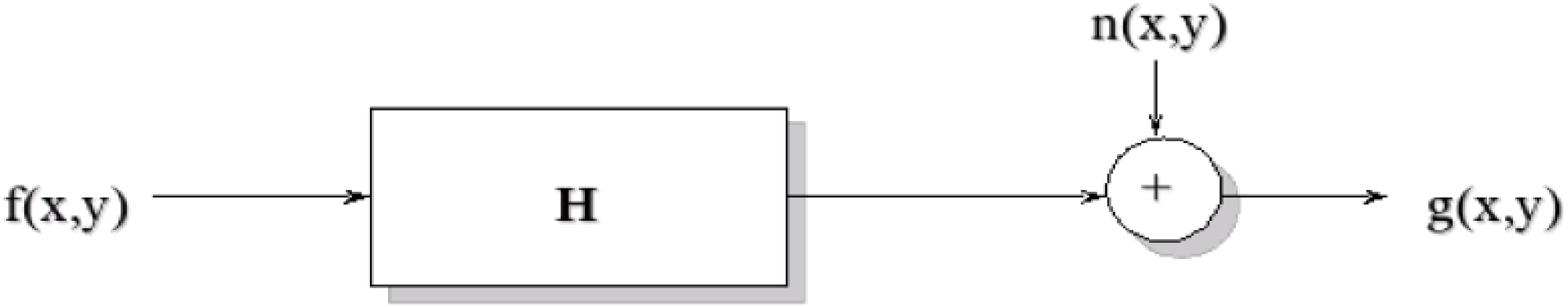
- Identify the degradation process and attempt to reverse it
- Similar to image enhancement, but more objective



INTRODUCTION

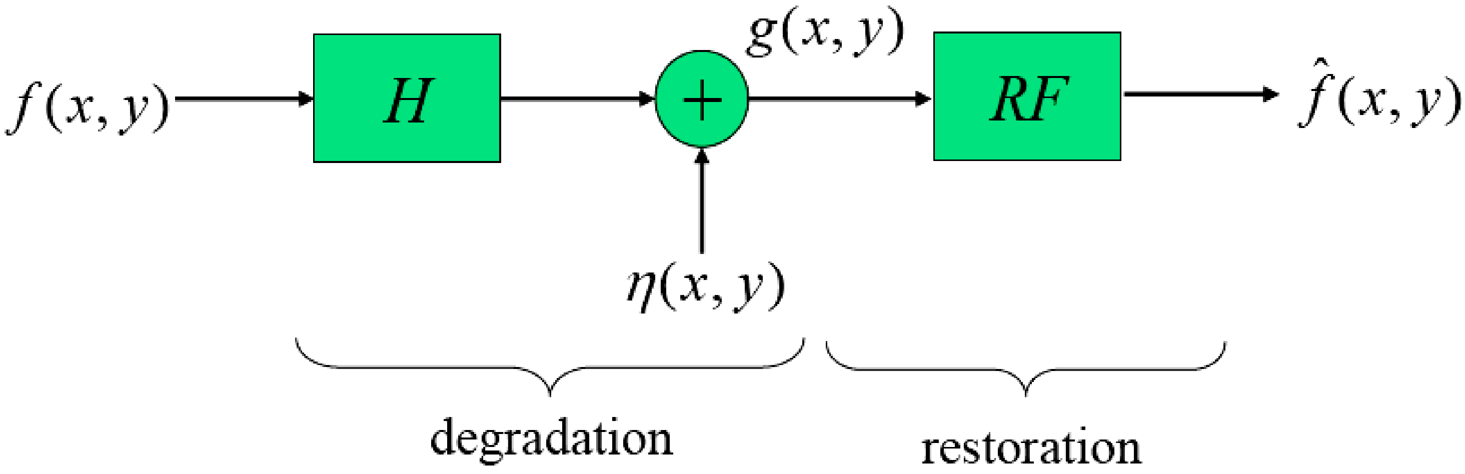
- **Objective of image restoration**
  - **to recover a distorted image to the original form based on idealized models.**
- **The distortion is due to**
  - **Image degradation in sensing environment e.g. random atmospheric turbulence**
  - **Noisy degradation from sensor noise.**
  - **Blurring degradation due to sensors**
    - **e.g. camera motion or out-of-focus**
  - **Geometric distortion**
    - **e.g. earth photos taken by a camera in a satellite**

Model



- $f(x, y)$ is the original image.
- H represents the system that affects our image.
- $n(x, y)$ is disturbance, e.g., noise or external contribution.
- Obtained degraded image $g(x, y) = H(f(x, y)) + n(x, y)$.
- Possible defects in the imaging system causing degradation:
  - Bad focusing.
  - Motion.
  - Non-linearity of the sensor.
  - Noise.
  - etc...

Image degradation / restoration model



- **When *H* is a LSI system**

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

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

An Image Degradation Model

- Two types of degradation
  - Additive noise
    - Spatial domain restoration (denoising) techniques are preferred
  - Image blur
    - Frequency domain methods are preferred
- We model the degradation process by a degradation function h(x,y), an additive noise term, η(x,y), as $g(x,y)=h(x,y)*f(x,y)+\eta(x,y)$
  - f(x,y) is the (input) image free from any degradation
  - g(x,y) is the degraded image
  - $*$ is the convolution operator
  - The goal is to obtain an estimate of f(x,y) according to the knowledge about the degradation function h and the additive noise η
  - In frequency domain: $G(u,v)=H(u,v)F(u,v)+N(u,v)$
- Three cases are considered in this Chapter
  - $g(x,y)=f(x,y)+\eta(x,y)$ (5-2~5-4)
  - $g(x,y)=h(x,y)*f(x,y)$ (5-5~5-6)
  - $g(x,y)=h(x,y)*f(x,y)+\eta(x,y)$ (5-7~5-9)

A Model of the Image Degradation/Restoration Process

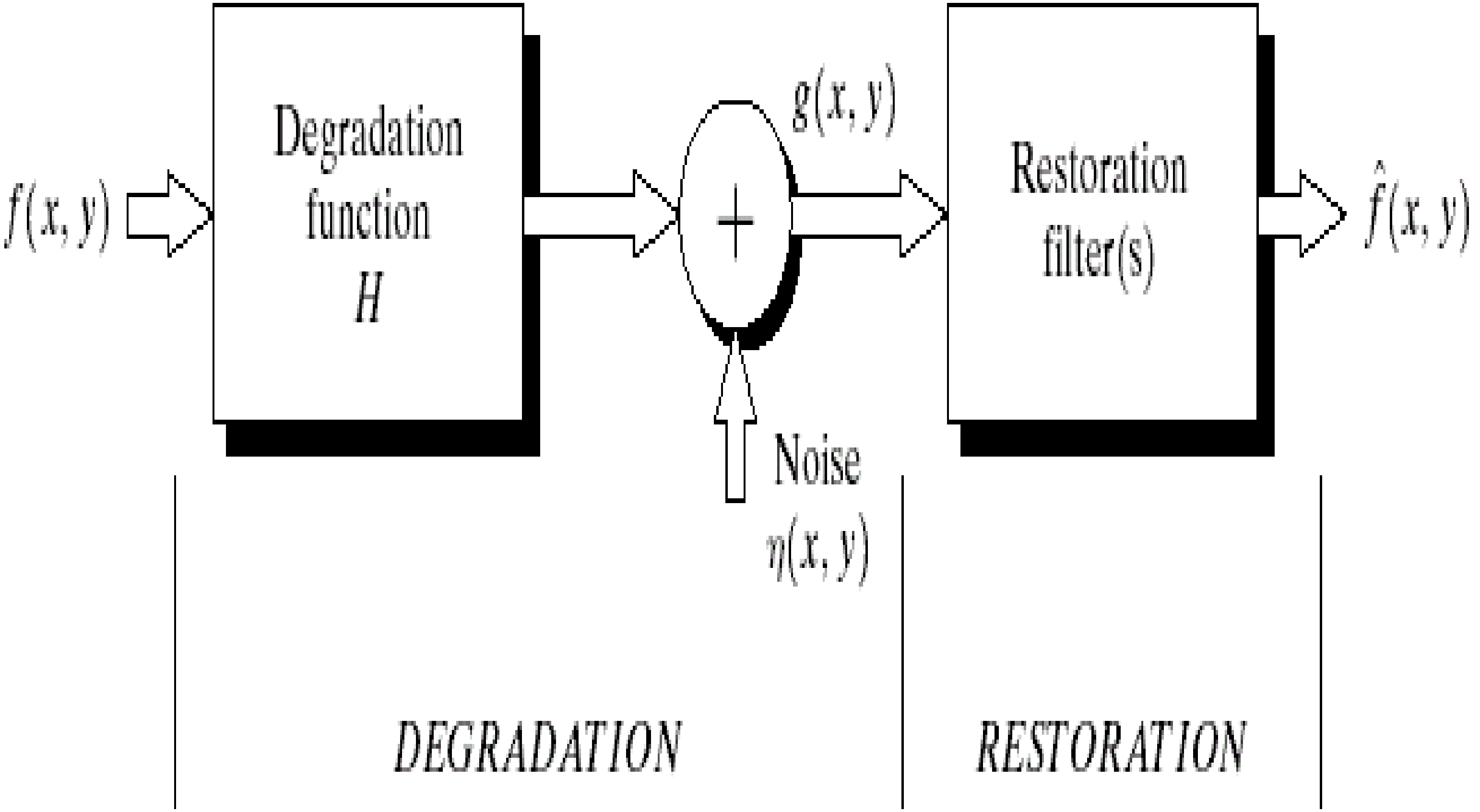
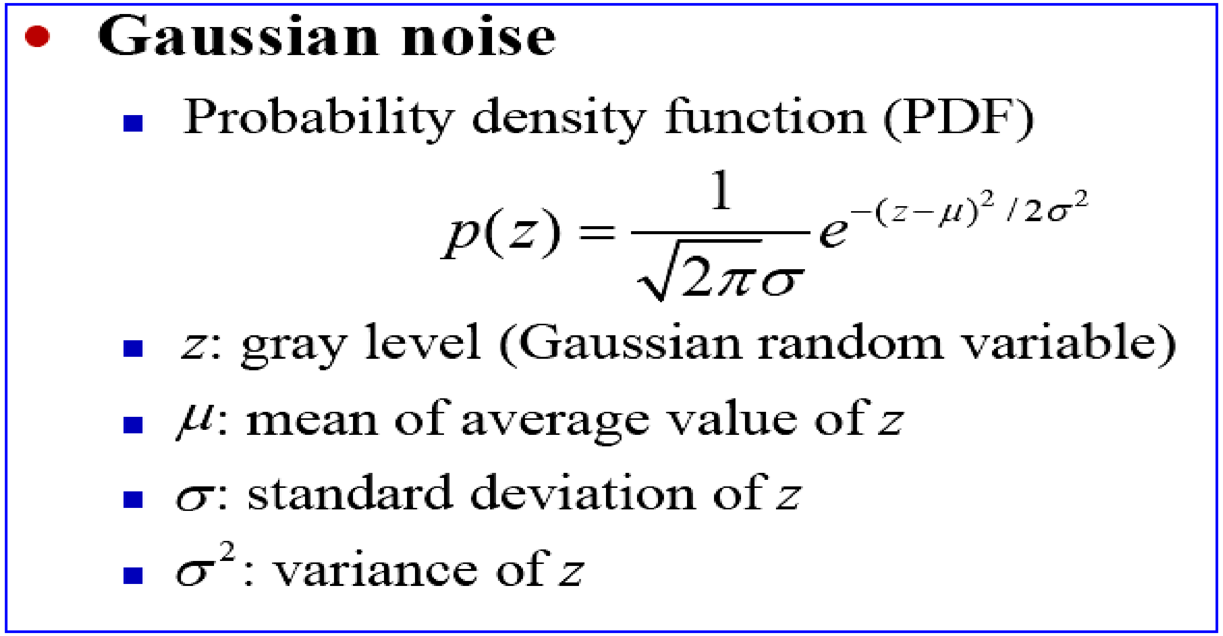


FIGURE 5.1 A model of the image degradation/restoration process

- We first consider the degradation due to noise only
  - h is an impulse for now (H is a constant)
- White noise
  - Autocorrelation function is an impulse function multiplied by a constant
    - $$a(x,y)=\sum_{t=0}^{N-1}\sum_{s=0}^{M-1}\eta(s,t)\cdot\eta(s-x,t-y)=N_0\delta(x,y)$$
    - It means there is no correlation between any two pixels in the noise image
    - There is no way to predict the next noise value
  - The spectrum of the autocorrelation function is a constant (white) (the statement in page 222 about white noise is wrong)

Noise models

- Assuming that noise is
  - independent of spatial coordinates, and
  - uncorrelated with respect to the image content



- **Gaussian noise**
  - Probability density function (PDF)

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2 / 2\sigma^2}$$

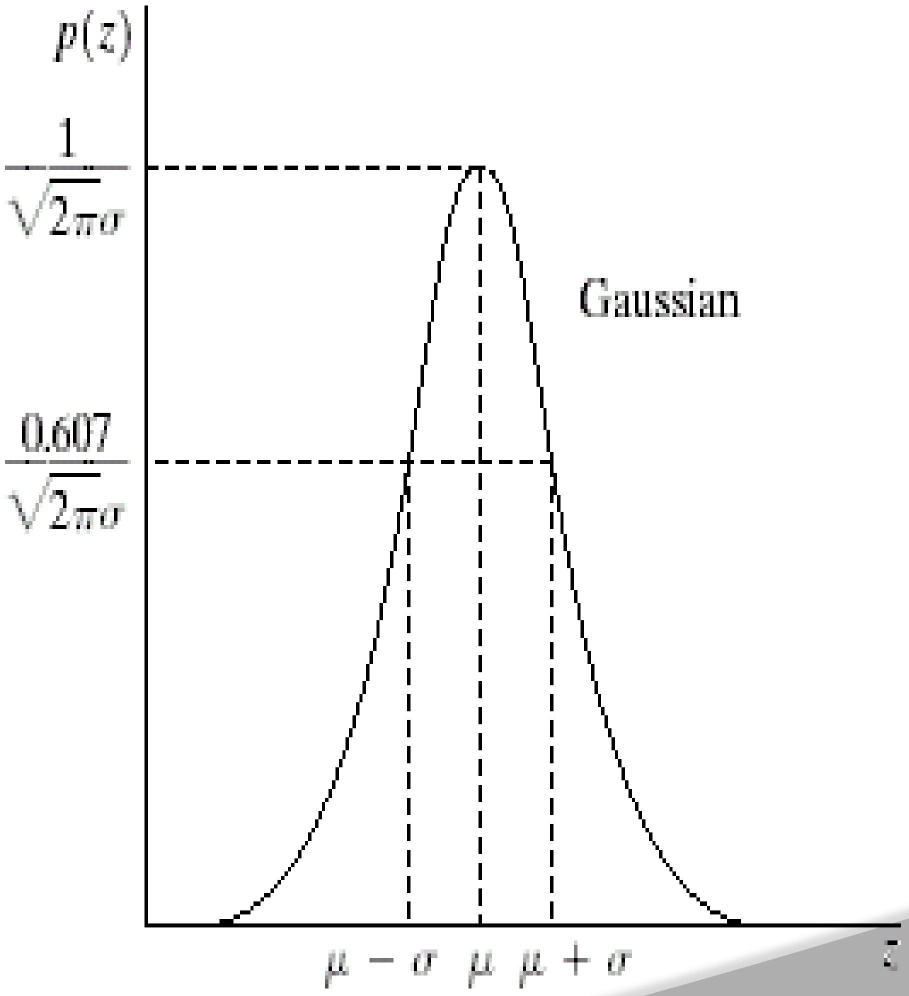
  - z: gray level (Gaussian random variable)
  - μ: mean of average value of z
  - σ: standard deviation of z
  - σ^2: variance of z

Noise (image) can be classified according the distribution of the values of pixels (of the noise image) or its (normalized) histogram

- Gaussian noise is characterized by two parameters, μ (mean) and σ² (variance), by

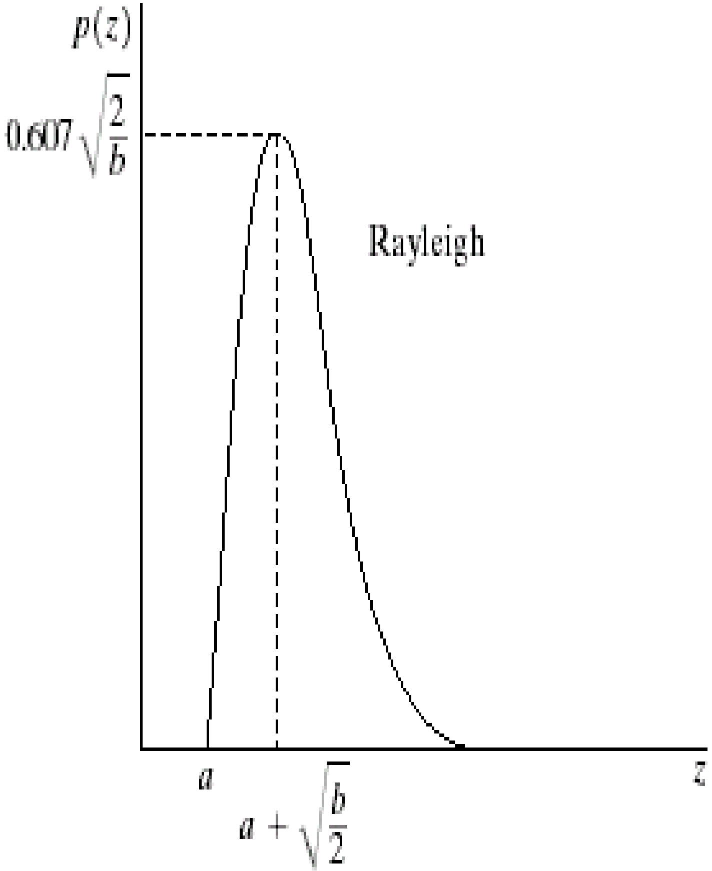
$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$

- 70% values of z fall in the range [(μ-σ),(μ+σ)]
- 95% values of z fall in the range [(μ-2σ),(μ+2σ)]



- **Rayleigh noise**

$$p(z)=\begin{cases}\frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z\geq a\\ 0 & \text{for } z<a\end{cases}$$



- **The mean and variance of this density are given by**

$$\mu=a+\sqrt{\pi b/4}\ \text{ and }\sigma^2=\frac{b(4-\pi)}{4}$$

- **a and b can be obtained through mean and variance**

We can consider a noisy image to be modelled as follows:

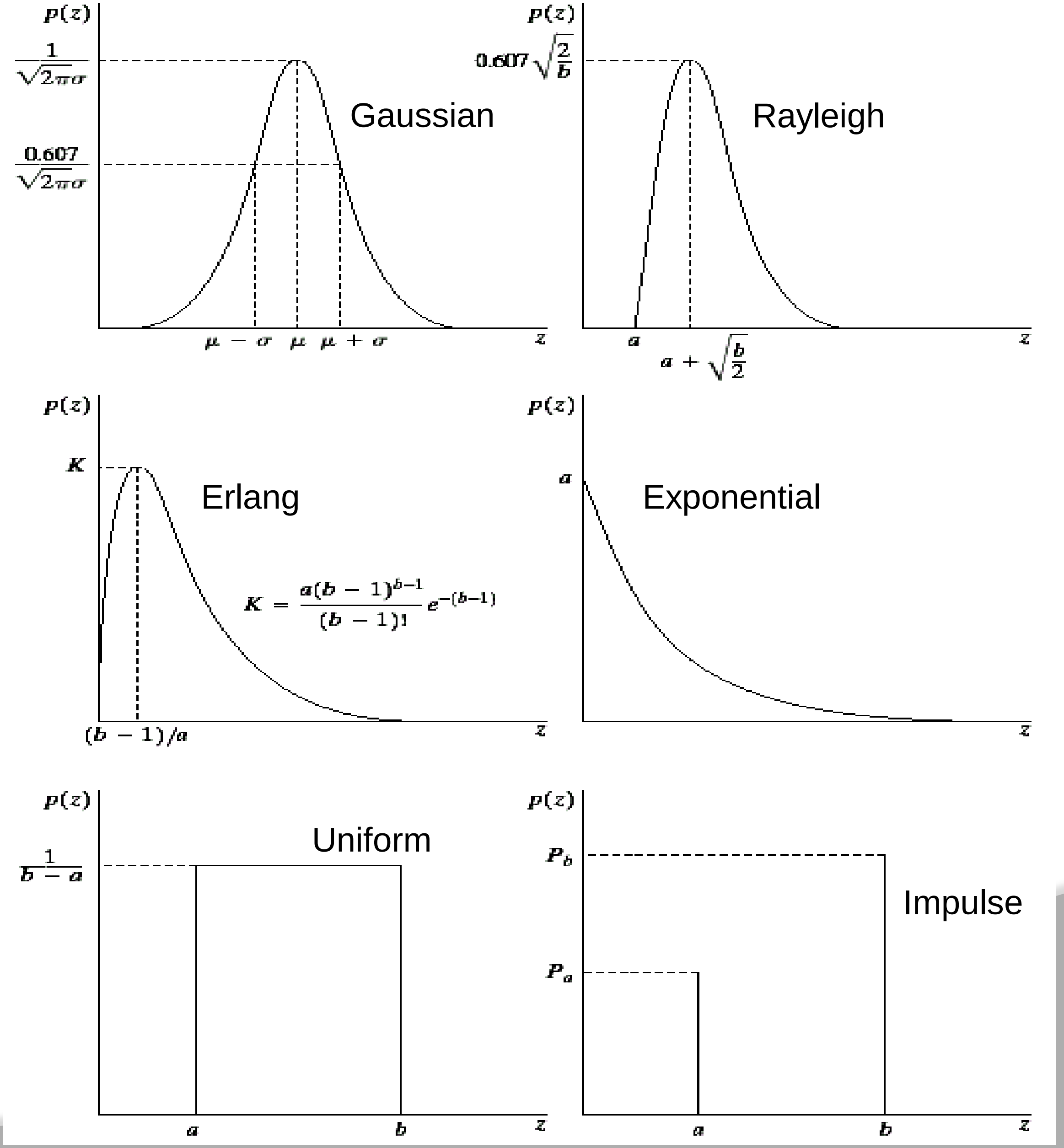
$$g(x, y) = f(x, y) + \eta(x, y)$$

where *f*(*x, y*) is the original image pixel, *η*(*x, y*) is the noise term and *g*(*x, y*) is the resulting noisy pixel

If we can estimate the noise model we can figure out how to restore the image

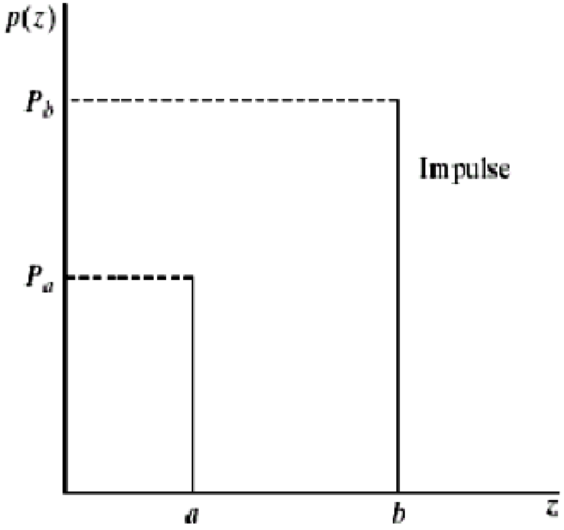
There are many different models for the image noise term *η*(*x*, *y*):

- Gaussian
  - Most common model
- Rayleigh
- Erlang (Gamma)
- Exponential
- Uniform
- Impulse
  - *Salt and pepper* noise



- **Impulse (salt-and-pepper) noise**

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$



- bipolar if $P_a \neq 0,\ P_b \neq 0$
- unipolar if one of P_a and P_b is 0
- noise looks like salt-and-pepper granules if $P_a \approx P_b$
- negative or positive; scaling is often necessary to form digital images
- extreme values occur (e.g. $a = 0$, $b = 255$)

Different restoration approaches

- Frequency domain
  - Inverse filter
  - Wiener (minimum mean square error) filter
- Algebraic approaches
  - Unconstrained optimization
  - Constrained optimization
  - The regularization theory

Algebraic Approach to Restoration

- The concept of algebraic approach is to estimate the original image which minimizes a predefined
- criterion of performances.
- Unconstraint restoration approach 2. Constraint restoration approach

Inverse filtering

simplest approach to restoration is direct inverse filtering:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

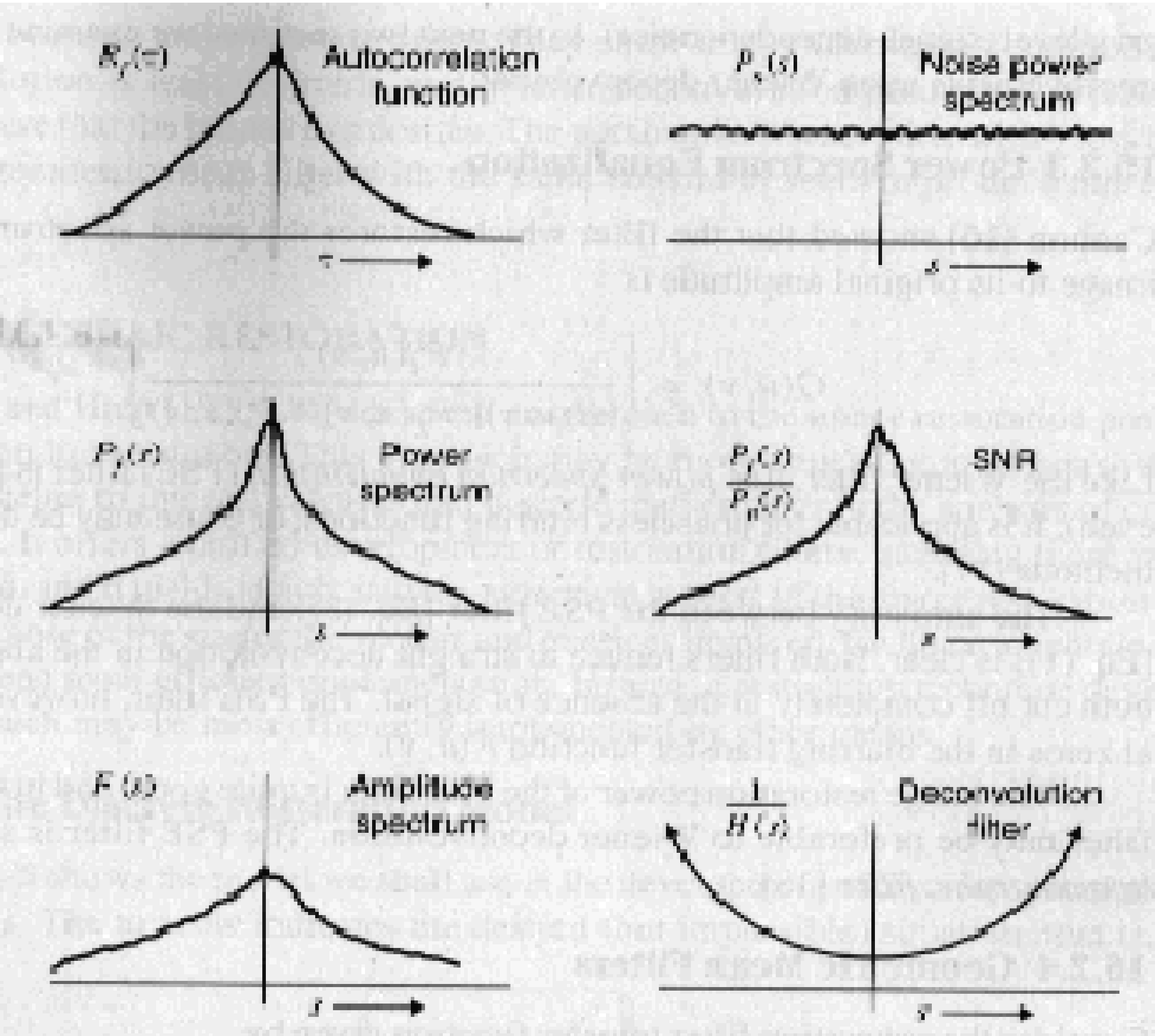
$$\hat{F}(u,v) = F(u,v) + \boxed{\frac{N(u,v)}{H(u,v)}}$$

Even if we know the degradation function, we cannot recover the un-degraded image

If the degradation has zero or very small values, then the ratio *N/H could easily dominate our estimation of F* .

One approach to get around the zero or small-value problem is to limit the filter frequencies to value near the origin.

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$



- In most images, adjacent pixels are highly correlated, while the gray levels of widely separated pixels are only loosely correlated.
- Therefore, the autocorrelation function of typical images generally decreases away from the origin.
- Power spectrum of an image is the Fourier transform of its autocorrelation function, therefore, we can argue that the power spectrum of an image generally decreases with frequency
- Typical noise sources have either a flat power spectrum or one that decreases with frequency more slowly than typical image power spectra.
- Therefore, the expected situation is for the signal to dominate the spectrum at low frequencies while the noise dominates at high frequencies.

Wiener filter (1942)

- Objective function: find an estimate   of *f* such that the mean square error between them is minimized

$$e^2 = E\{(f - \hat{f})^2\}$$

$$\hat{F}(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)} G(u,v)$$

- Potential problems:
  - Weights all errors equally regardless of their location in the image, while the eye is considerably more tolerant of errors in dark areas and high-gradient areas in the image.
  - In minimizing the mean square error, Wiener filter also smooth the image more than the eye would prefer

Algebraic approach – Unconstrained restoration vs. Inverse filter

$$\mathbf{g} = \mathbf{Hf} + \mathbf{n}$$

$$\mathbf{n} = \mathbf{g} - \mathbf{Hf} \xrightarrow[\text{in a least squares sense}]{\text{seek } \hat{\mathbf{f}} \text{ such that } \mathbf{H}\hat{\mathbf{f}} \text{ approximates } \mathbf{g}}$$

$$J\left(\hat{\mathbf{f}}\right) = \left\|\mathbf{n}\right\|^2 = \left\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\right\|^2 \xrightarrow[\text{with respect to } \hat{\mathbf{f}}]{\text{differentiate right hand side}}$$

$$\frac{\partial J\left(\hat{\mathbf{f}}\right)}{\partial \hat{\mathbf{f}}} = 0 = -2\mathbf{H}^T\left(\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\right) \Rightarrow \mathbf{H}^T\mathbf{H}\hat{\mathbf{f}} = \mathbf{H}^T\mathbf{g}$$

$$\hat{\mathbf{f}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{g} = \mathbf{H}^{-1}\left(\mathbf{H}^T\right)^{-1}\mathbf{H}^T\mathbf{g} = \mathbf{H}^{-1}\mathbf{g}$$

Compared to the inverse filter: $\hat{F}(u,v) = \dfrac{G(u,v)}{H(u,v)}$

Algebraic approach – Constrained restoration vs. Wiener filter

Minimizing $\left\|\mathbf{Q}\hat{\mathbf{f}}\right\|^2$, where $\mathbf{Q}$ is a linear operator on $\mathbf{f}$,

subject to the constraint $\left\|\mathbf{g}-\mathbf{H}\hat{\mathbf{f}}\right\|^2=\left\|\mathbf{n}\right\|^2$.

Model this problem using Lagrange optimization method

We seek $\hat{\mathbf{f}}$ that minimizes the criterion (or objective) function

$$J\left(\hat{\mathbf{f}}\right)=\left\|\mathbf{Q}\hat{\mathbf{f}}\right\|^2+\alpha\left(\left\|\mathbf{g}-\mathbf{H}\hat{\mathbf{f}}\right\|^2-\left\|\mathbf{n}\right\|^2\right)$$

α is a constant, called the Lagrange multiplier.

$$\frac{\partial J\left(\hat{\mathbf{f}}\right)}{\partial\hat{\mathbf{f}}}=0=2\mathbf{Q}^T\mathbf{Q}\hat{\mathbf{f}}-2\alpha\mathbf{H}^T\left(\mathbf{g}-\mathbf{H}\hat{\mathbf{f}}\right)$$

$$\hat{\mathbf{f}}=\left(\mathbf{H}^T\mathbf{H}+\gamma\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{H}^T\mathbf{g}$$

Compared to:

$$\hat{F}(u,v)=\frac{1}{H(u,v)}\frac{\left|H(u,v)\right|^2}{\left|H(u,v)\right|^2+S_n(u,v)/S_f(u,v)}G(u,v)$$

Degradation Model

- In noise-free cases, a blurred image can be modeled as

$$y = x * h$$

h : linear space-invariant blur function

x : original image

In the DFT domain, $Y(u,v) = X(u,v)\ H(u,v)$

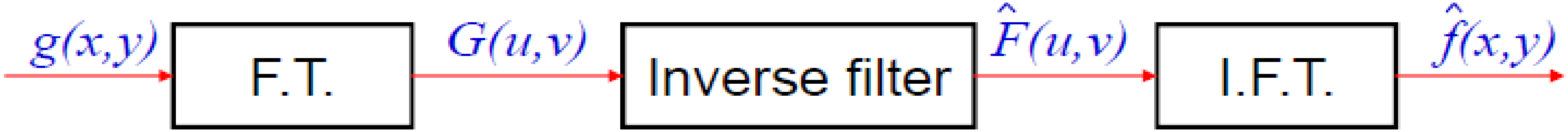
Start from the generative model

$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \Leftrightarrow \quad G(u,v) = H(u,v)\, F(u,v) + N(u,v)$$

and for the moment ignore $n(x,y)$, then an estimate of $f(x,y)$ is obtained from

$$\hat{F}(u,v) = G(u,v) \,/\, H(u,v)$$

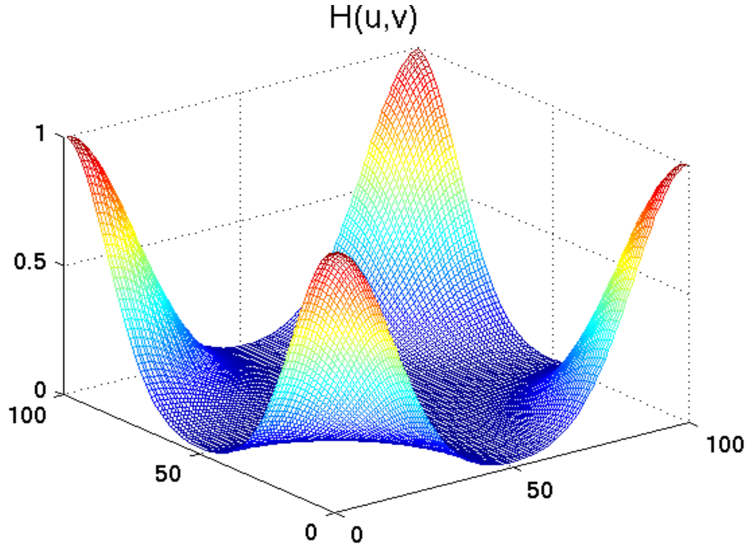
Restoration with an inverse filter

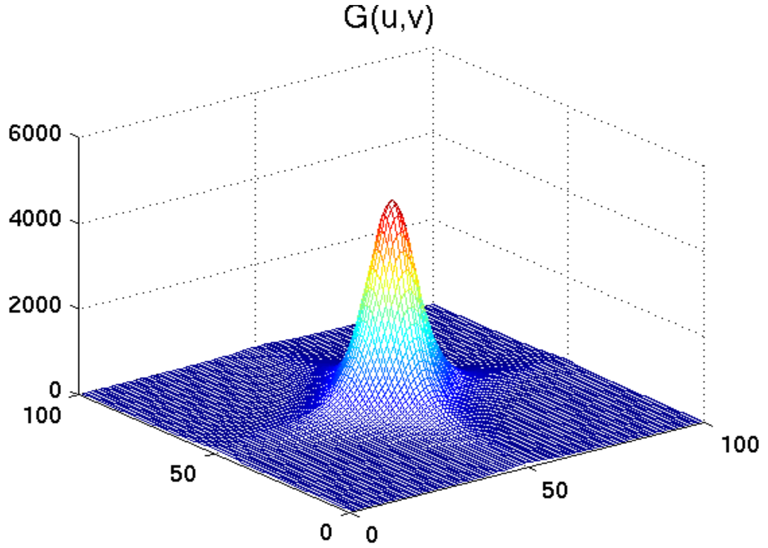


Inverse Filtering

- **Assume h is known (low-pass filter)**
  - Inverse filter G(u,v) = 1 / H(u,v)
  -

$$\tilde{X}(u,v) = Y(u,v)\,G(u,v)$$





Wiener Filter Formulation

- Least Mean Square Filter

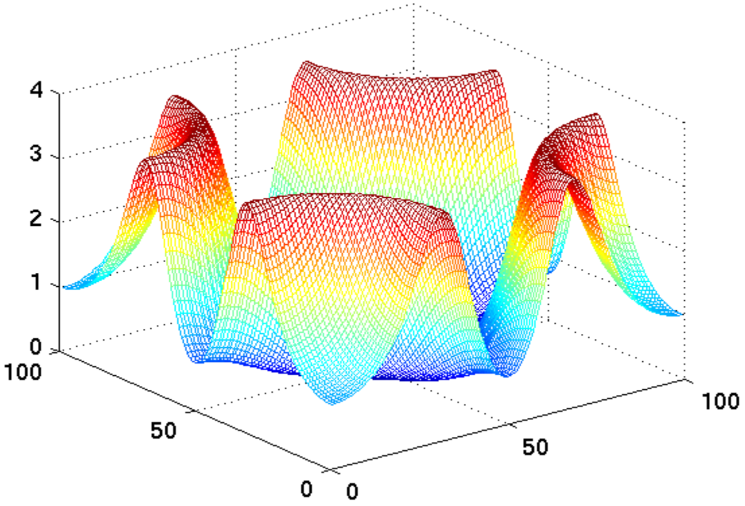
$$G(u,v) = \frac{H^{*}(u,v)}{\left|H(u,v)\right|^{2} + \left[S_{n}(u,v)/S_{x}(u,v)\right]}$$

- In practice

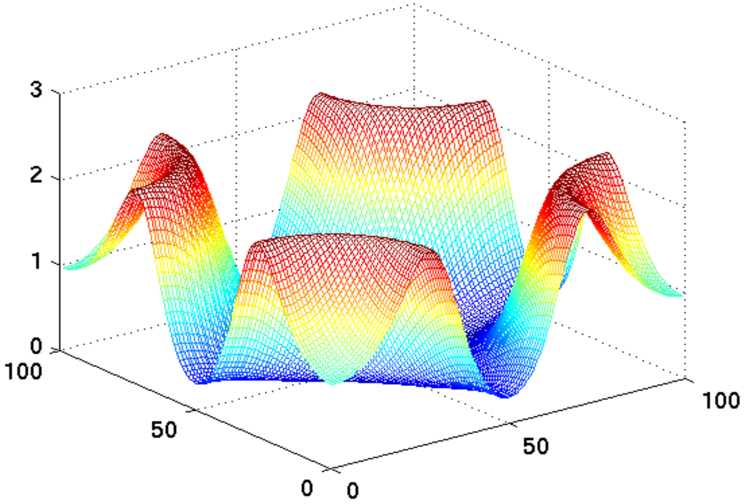
$$G(u,v) = \frac{H^{*}(u,v)}{\left|H(u,v)\right|^{2} + K}$$

Wiener Filter Results

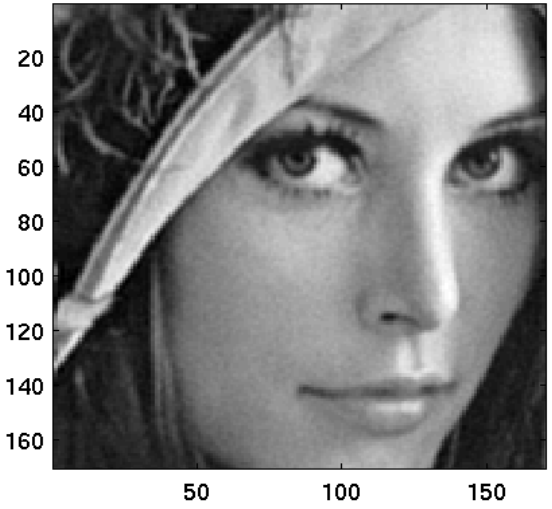
G(u,v), K=0.02



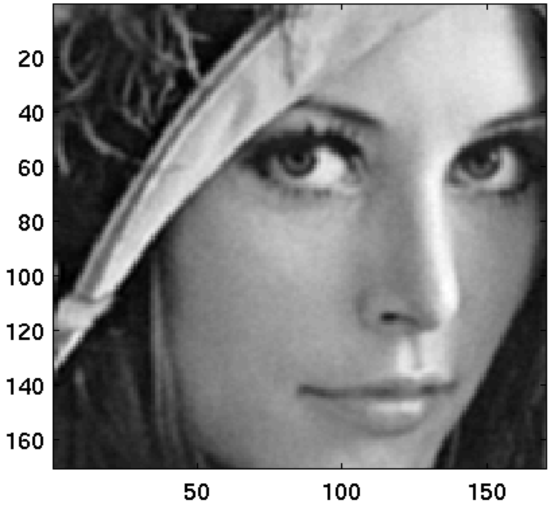
G(u,v), K=0.05



K=0.02



K=0.05



Maximum-Likelihood (ML) Estimation

- h is unknown

- Assume parametric models for the blur function, original image, and/or noise
- Parametric set θ is estimated by

$$\theta_{\mathrm{ml}} = \arg\{\max_{\theta} p(y \mid \theta)\}$$

- Solution is difficult

Expectation-Maximization (EM) Algorithm

Find *complete set* Z: for z $\in Z$, *f*(z)=y

Choose an initial guess of θ

- **Expectation-step**

$$g(\theta \mid \theta^{k}) = E[\mathrm{p}(\mathrm{z} \mid \theta) \mid \mathrm{y}, \theta^{k}]$$

Maximization-step

$$\theta^{\mathbf{k}+1} = \arg\max_{\theta} g(\theta \mid \theta^{k})$$

Minimum Mean Square Error Filtering (Wiener Filtering)

This approach incorporate both the degradation function and statistical characteristic of noise into the restoration process.

Image and noise are random process

$$e^2 = E[(f - \hat{f})^2]$$

The objective is to find an estimation for *f* such that minimized e^2

$$\hat{F}(u,v)=\left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

Constant

Unknown

$$\Downarrow$$

$$\hat{F}(u,v)=\left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+K}\right]G(u,v)$$

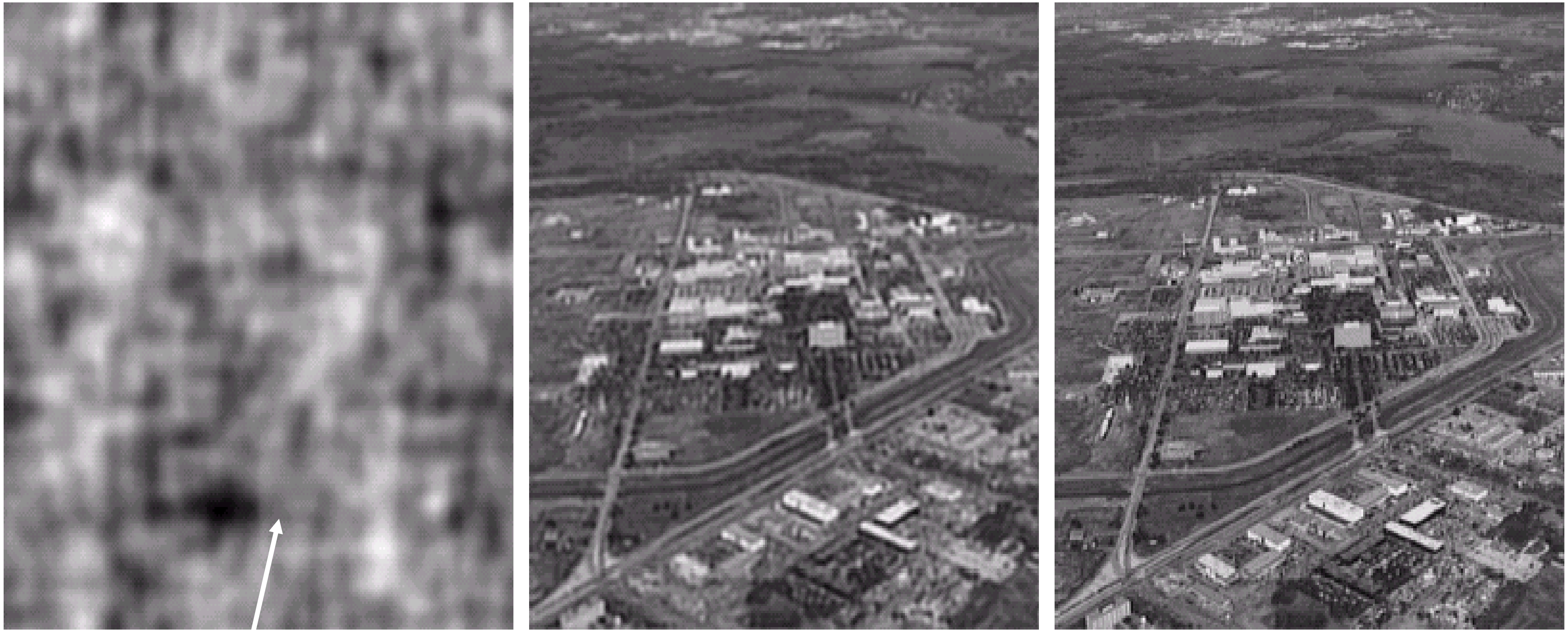


FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

Full inverse filtering

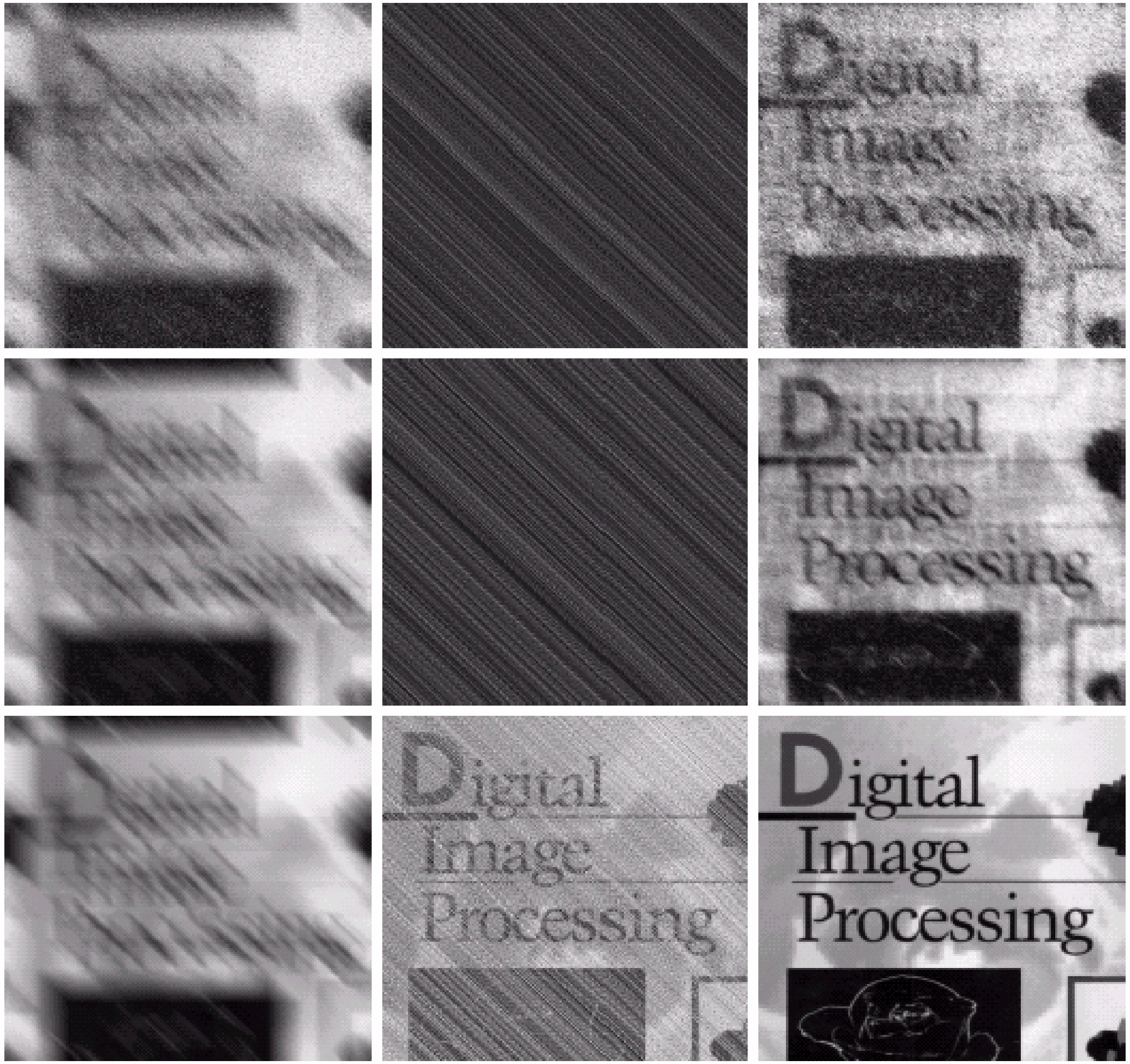
Radially limited inverse filtering

(K was chosen interactively)

Inverse filtering

Wiener filtering

Reduced noise variance

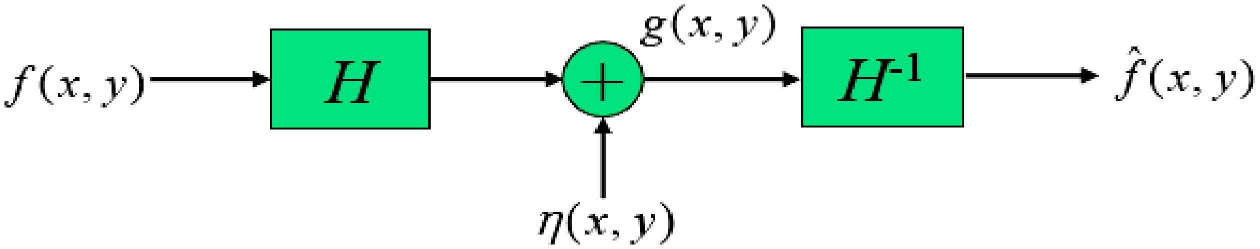


Inverse filtering

- **When *H*(*u*,*v*) is known, the simplest approach to restoration is direct inverse filtering**

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$



- However, even if *H* is known completely, the undegraded image cannot be recovered exactly due to noise *N*

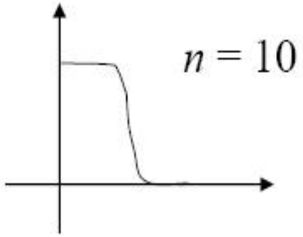
$$\hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}=F(u,v)+\frac{N(u,v)}{H(u,v)}$$

- Even worse when *H* has zero or very small values, *N*/*H* would dominate the estimated image
- One way to get around this problem is to limit the filter frequencies to values near the origin where *H* is large in general

Degradation function $H(u,v) = e^{-k\left[(u-M/2)^2+(v-N/2)^2\right]^{5/6}}$

- **An example**

Butterworth filter H_b



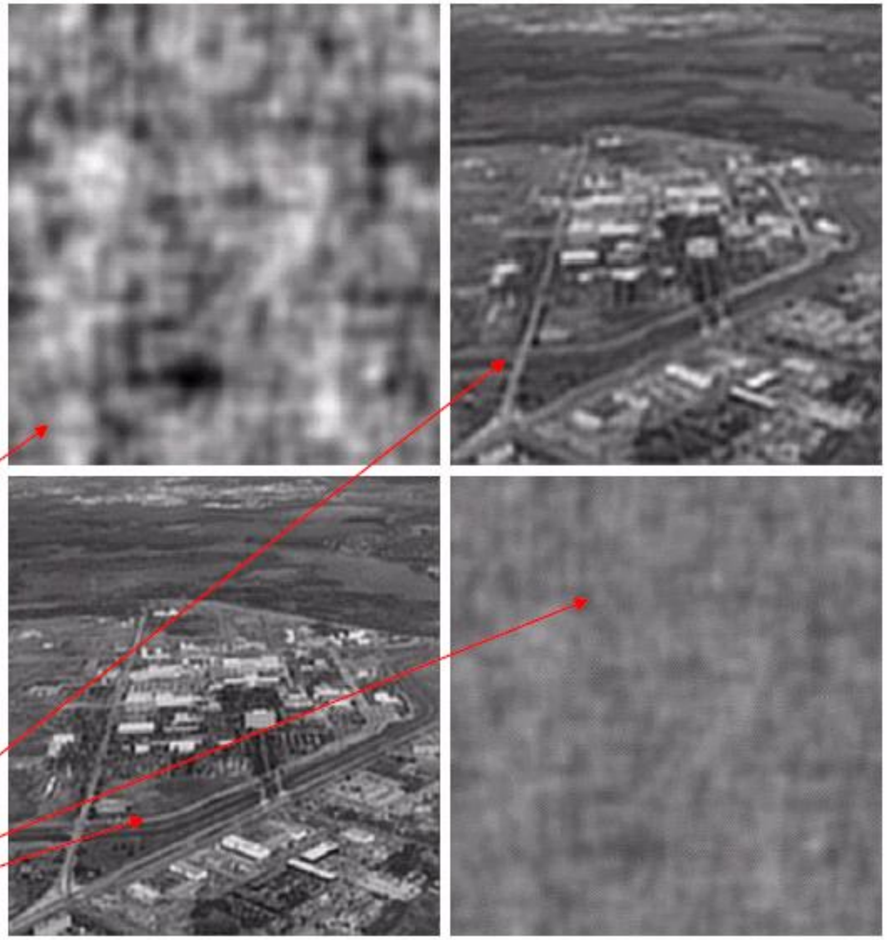
$$\frac{G(u,v)}{H(u,v)} = \hat{F}(u,v) \Leftrightarrow \hat{f}(x,y)$$

$$\frac{G(u,v)}{H(u,v)} H_b(u,v) = \hat{F}(u,v)$$

$$\Updownarrow$$

$$\hat{f}(x,y)$$

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

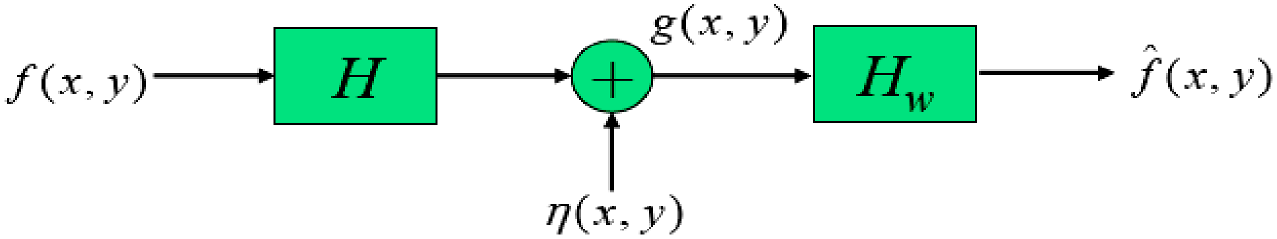


Cutting off values of the ratio outside a radius of 40, 70,85.

Wiener filtering

- **Main limitation of inverse filtering**
  - Very sensitive to noise
- **Wiener filtering (minimum mean square error filtering)**
  - Use **statistic information** about signal and noise to improve the restoration
  - Consider images and noise as **random processes**
  - Objective:

$$\boxed{e^2 = E\{(f - \hat{f})^2\} \Rightarrow \min}$$



- **Assumptions**
  - The noise and the image are uncorrelated
  - One or the other has zero mean
- **Estimated image**

$$\min\{e^2\} \Rightarrow \hat{F}(u,v)$$

$$\boxed{\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)}$$

$$\left|H(u,v)\right|^2 = H^*(u,v)H(u,v)$$

$$S_\eta(u,v) = \left|N(u,v)\right|^2 : \text{power spectrum of the noise}$$

$$S_f(u,v) = \left|F(u,v)\right|^2 : \text{power spectrum of the original image}$$

$$(\eta(x,y) \Leftrightarrow N(u,v),\ f(x,y) \Leftrightarrow F(u,v))$$

- **Wiener filter**

$$\boxed{H_w(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}}$$

- **When noise is zero** $(N(u,v) = 0)$

$$H_w(u,v) = \frac{1}{H(u,v)} \quad \Rightarrow \text{ the inverse filter}$$

- **When** $S_\eta(u,v)$ **and/or** $S_f(u,v)$ **are unknown**

$$H_w(u,v) \approx \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}$$

some constant

Drawback with Wiener Filtering

Drawback:

- Must know (or approximate) the power spectra of undegraded image and noise.

Better choice:

- Constrained Least Squares Filter (CLSF).
  - Need only knowledge of mean and variance of noise (apart from degradation function).
  - Only one parameter, which can be iteratively computed for optimality.

- **An example**

$$\frac{G(u,v)}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+K}=\hat{F}(u,v)\Leftrightarrow\hat{f}(x,y)$$

$$\frac{G(u,v)}{H(u,v)}H_b(u,v)=\hat{F}(u,v)\Leftrightarrow\hat{f}(x,y)$$

$$H(u,v)=e^{-k(u^2+v^2)^{5/6}}$$

$$\hat{f}(x,y)$$

$$\Updownarrow$$

$$\frac{G(u,v)}{H(u,v)}=\hat{F}(u,v)$$

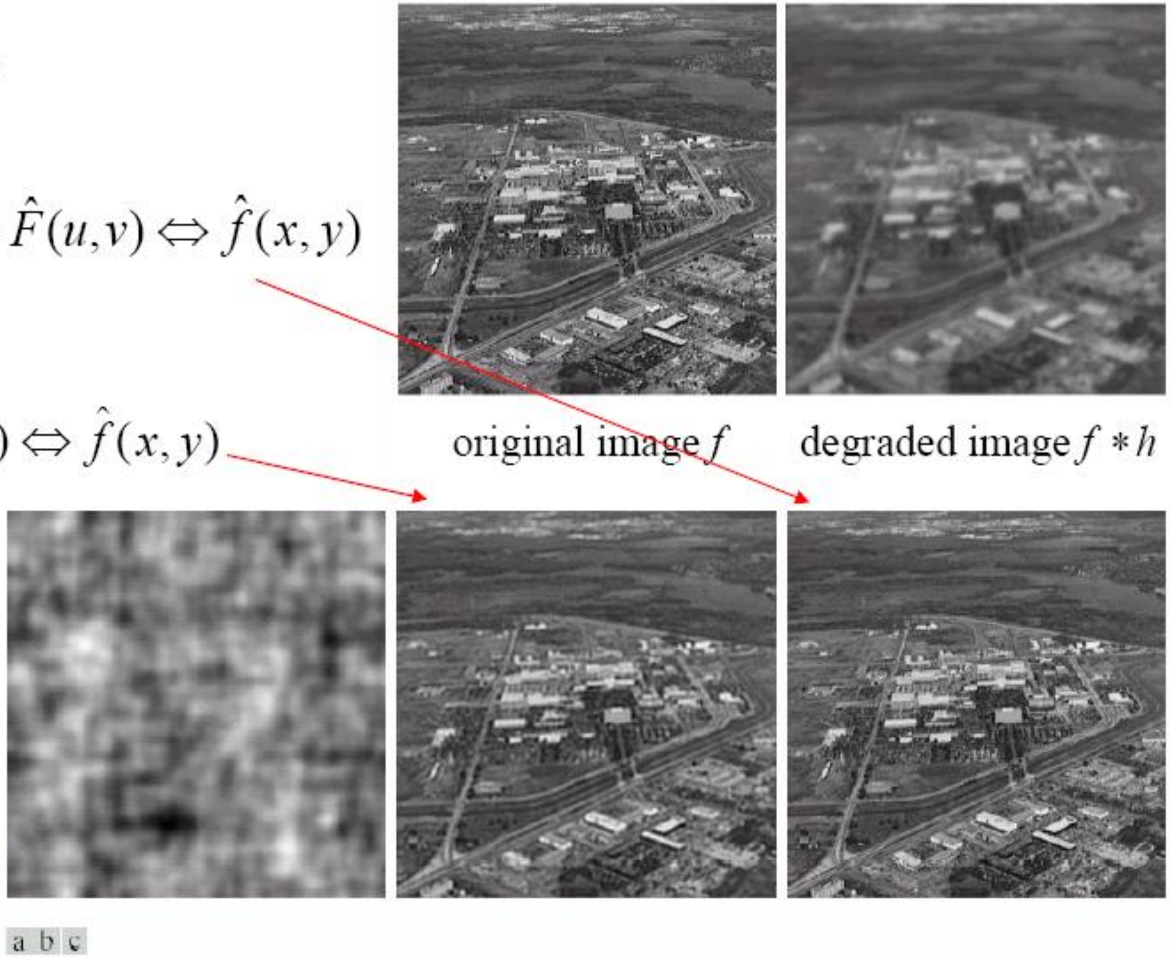
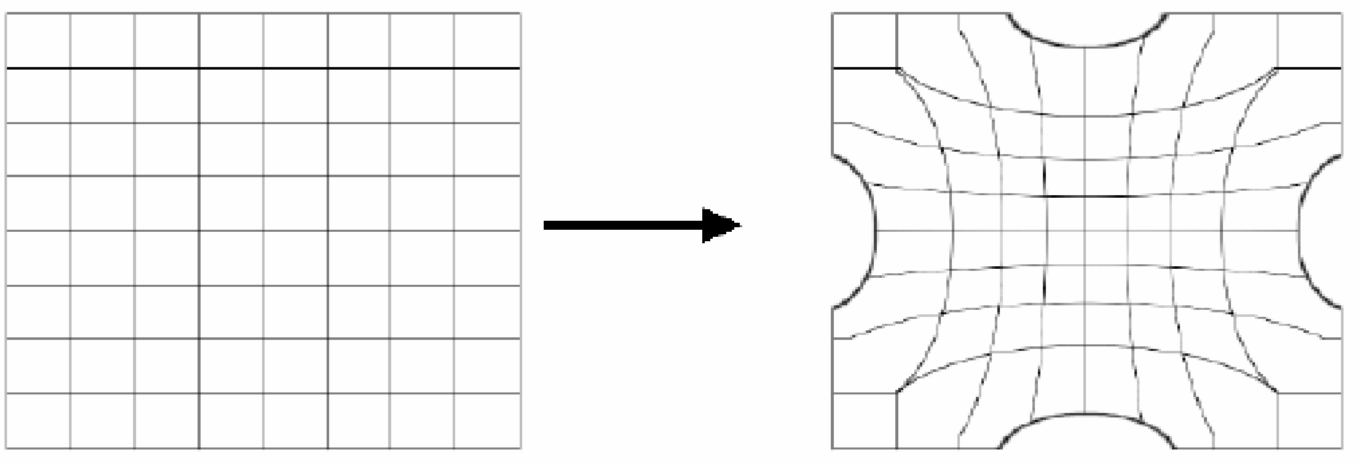


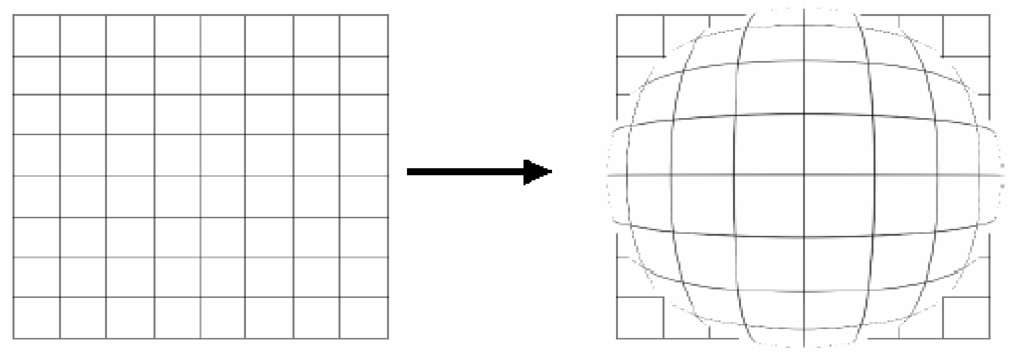
FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). . (b) Radially limited inverse filter result. (c) Wiener filter result.

Geometric transformations

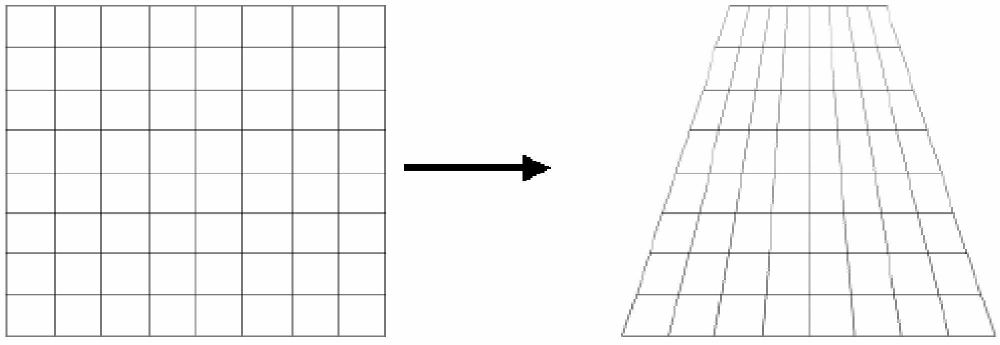
- Objective: to eliminate geometric distortion that occurs when an image is captured
- Examples of geometric distortion
  - Pincushion distortion (associated with zoom lenses)



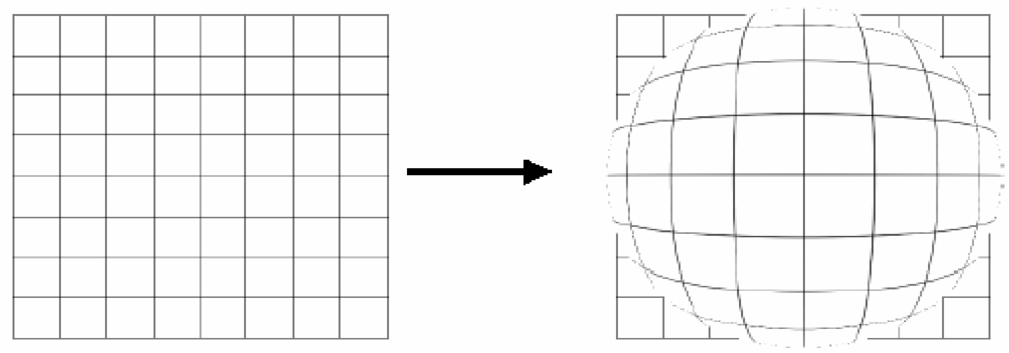
- **Examples of geometric distortion**
  - Barrel distortion (associated with wide angle lenses)



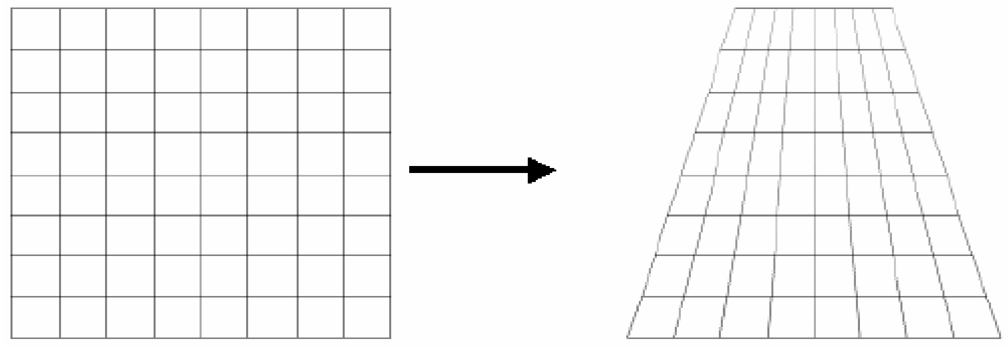
  - Perspective distortion



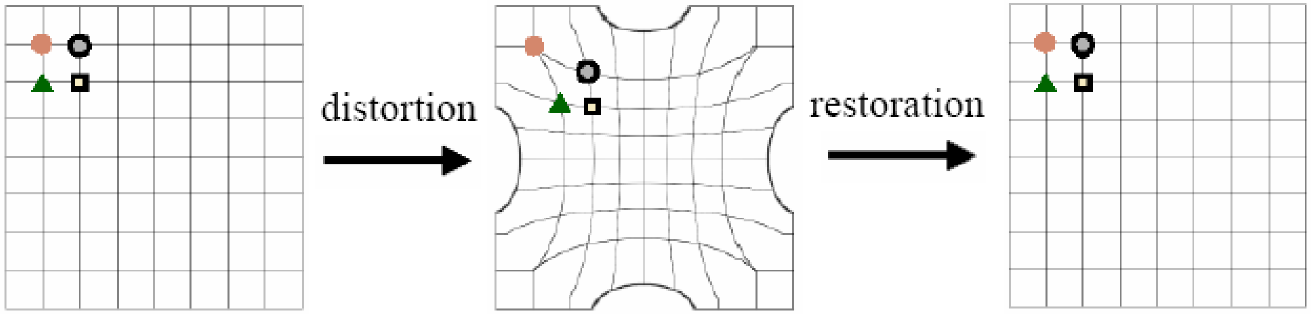
- **Examples of geometric distortion**
  - Barrel distortion (associated with wide angle lenses)



  - Perspective distortion



- **Spatial transformation**



distortion

restoration

$f(x,y)$ $g(x',y')$ $\hat{f}(x,y)$

- Restoration:

$$\begin{cases} x' = r(x,y) \\ y' = s(x,y) \end{cases} \xrightarrow{r,\ s \text{ known}} \begin{cases} x = r'(x',y') \\ y = s'(x',y') \end{cases}$$

- Unfortunately, functions $r(x,y)$ and $s(x,y)$ that describe the geometric distortion over the entire image is generally **unknown**

- **In practice, the geometric distortion is often approximated by the bilinear transformation**

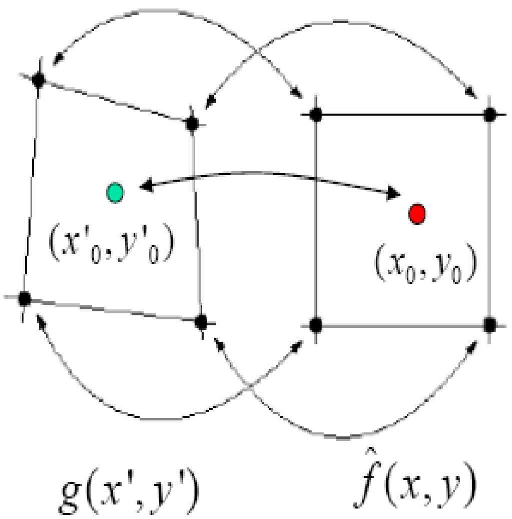
$$x' = r(x, y) = c_1 x + c_2 y + c_3 xy + c_4$$

$$y' = s(x, y) = c_5 x + c_6 y + c_7 xy + c_8$$

- **4 pairs of tiepoints are needed to derive the 8 coefficients**
  - one pair of tiepoints results in two linear equations

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_8 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & x_1 y_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & x_1 y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & x_4 & y_4 & x_4 y_4 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ y'_1 \\ \vdots \\ y'_8 \end{bmatrix}$$

- Restoration of the image within the quadrilateral region
  - $(x_0, y_0) \Longrightarrow x'_0 = r(x_0, y_0),\ y'_0 = s(x_0, y_0)$
  - $\hat{f}(x_0, y_0) = g(x'_0, y'_0)$



- In general, enough tiepoints are needed to generate a set of quadrilaterals that cover the entire image, with each quadrilateral having its own 8 coefficients

- **A problem when calculating**

$$(x',y') = (r(x,y), s(x,y)) \Longleftarrow (x,y)$$

  - x' and y' may not be integers
  - gray-level interpolation is necessary

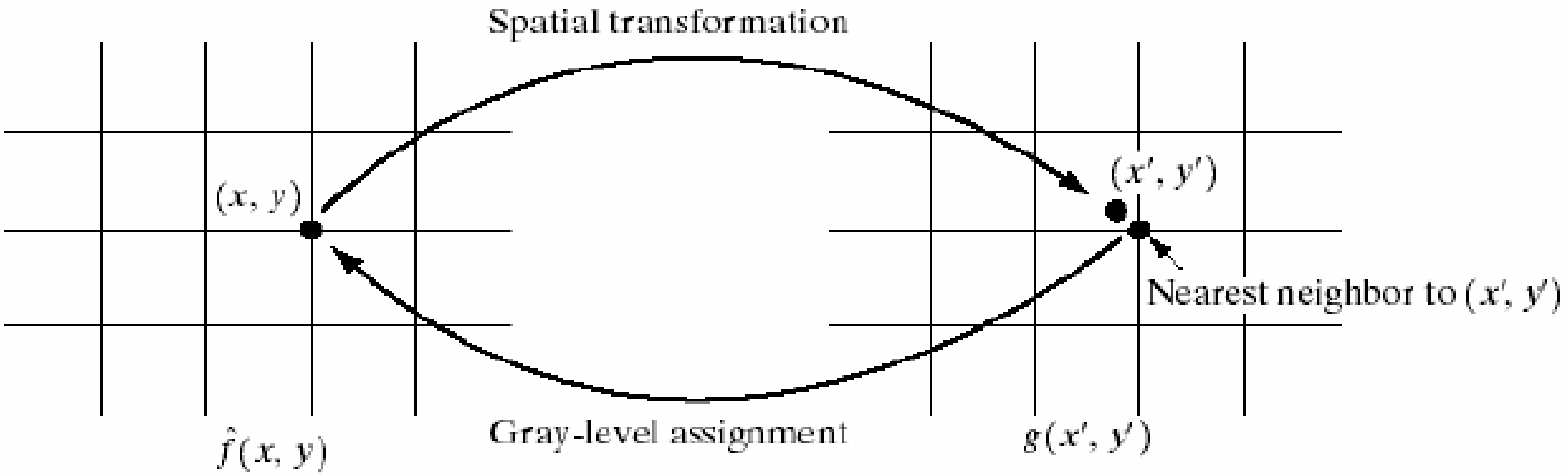


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.

- **Gray-level interpolation**
  - **Nearest neighbor interpolation**

$$\begin{cases} x' = \text{Round}\{r(x,y)\} \\ y' = \text{Round}\{s(x,y)\} \end{cases}$$

    - simple but may result in undesirable artifacts, such as step-like edges that should be straight/smooth before transformation

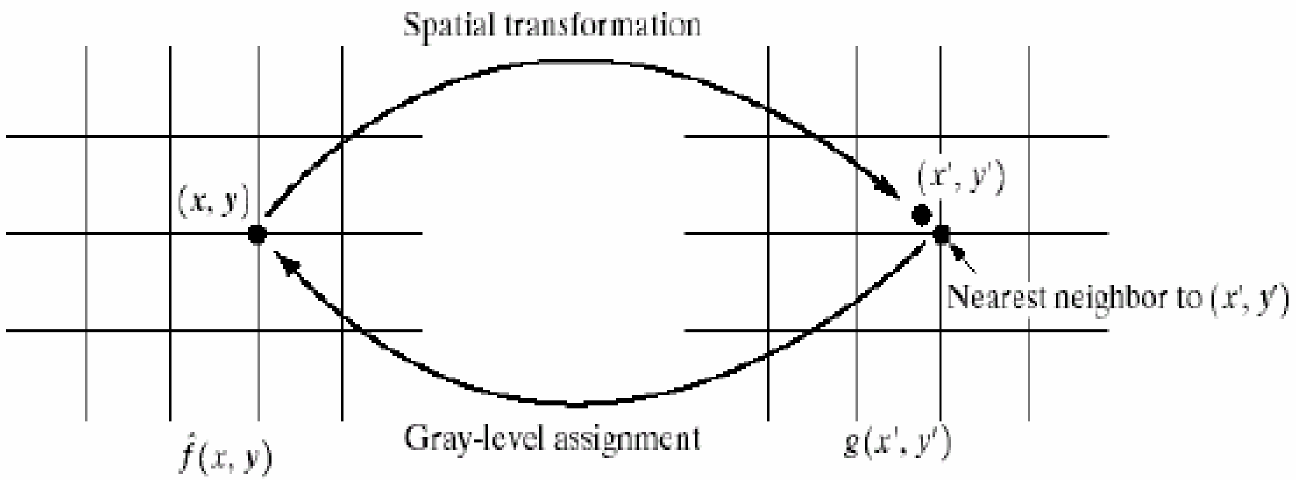
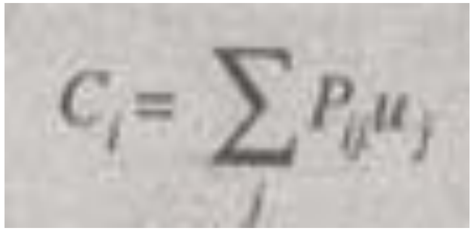


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.

Iterative nonlinear restoration using the Lucy–Richardson algorithm.

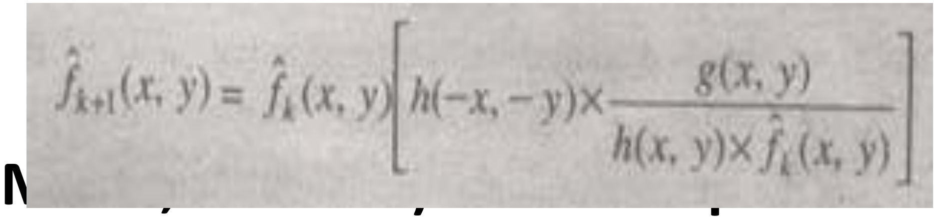
- Lucy-Richardson algorithm is a nonlinear restoration method used to recover a latent image which is blurred by a Point Spread Function (psf). It is also known as Richardson-Lucy de-convolution. With as the point spread function, the pixels in observed image are expressed as,

- u_j = Pixel value at location j in the image
- c_i = Observed value at i^{th} pixel locacation



$$C_i = \sum_j P_{ij} u_j$$

- Here,
- *f* = Estimation of undegraded image.
- The factor *f* which is present in the right side denominator leads to non-linearity. Since, the algorithm is a type of nonlinear restorations; hence it is stopped when satisfactory result is obtained. The basic syntax of function deconvlucy with the L-R algorithm is implemented is given below.



$$\hat{f}_{k+1}(x, y) = \hat{f}_k(x, y)\left[h(-x, -y)\times\frac{g(x, y)}{h(x, y)\times\hat{f}_k(x, y)}\right]$$

- *fr* = Deconvlucy (g, psf, NUMIT, DAN ers are,
- g = Degraded image, f_r = Restored image, psf = Point spread function
- NUMIT = Total number of iterations. The remaining two parameters are,

Iterative deterministic approaches to restoration

They refer to a large class of methods that have been investigated extensively over the last decades. They possess the following advantages.

- There is no need to explicitly implement the inverse of an operator. The restoration process is monitored as it progresses. Termination of the algorithm may take place before convergence.
- The effects of noise can be controlled in each iteration.
- The algorithms used can be spatially adaptive.
- The problem specifications are very flexible with respect to the type of degradation. Iterative techniques can be applied in cases of spatially varying or nonlinear degradations or in cases where the type of degradation is completely unknown (blind restoration).

In general, iterative restoration refers to any technique that attempts to minimize a function of the form

$$M(\mathbf{f})$$

using an updating rule for the partially restored image.

Constrained Least Squares Iteration

In this method we attempt to solve the problem of constrained restoration iteratively. As already mentioned the following functional is minimized

$$M(\mathbf{f},\alpha) = \|\mathbf{y} - \mathbf{Hf}\|^2 + \alpha\|\mathbf{Cf}\|^2$$

The necessary condition for a minimum is that the gradient of $M(\mathbf{f},\alpha)$ is equal to zero. That gradient is

$$\Phi(\mathbf{f}) = \nabla_{\mathbf{f}} M(\mathbf{f},\alpha) = 2[(\mathbf{H}^{\mathbf{T}}\mathbf{H} + \alpha\mathbf{C}^{\mathbf{T}}\mathbf{C})\mathbf{f} - \mathbf{H}^{\mathbf{T}}\mathbf{y}]$$

The initial estimate and the updating rule for obtaining the restored image are now given by

$$\mathbf{f}_0 = \beta\mathbf{H}^{\mathbf{T}}\mathbf{y}$$

$$\mathbf{f}_{k+1} = \mathbf{f}_k + \beta[\mathbf{H}^{\mathbf{T}}\mathbf{y} - (\mathbf{H}^{\mathbf{T}}\mathbf{H} + \alpha\mathbf{C}^{\mathbf{T}}\mathbf{C})\mathbf{f}_k]$$

It can be proved that the above iteration (known as Iterative CLS or Tikhonov-Miller Method) converges if

$$0 < \beta < \frac{2}{\left|\begin{matrix}\lambda \\ -\end{matrix}\right|}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the matrix

DIGITAL IMAGE PROCESSING

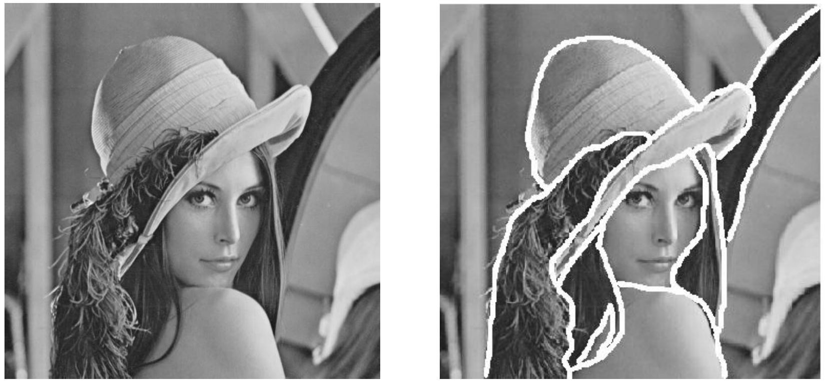
UNIT-IV

IMAGE SEGMENTATION

- Image segmentation detection of discontinuities
- edge linking and boundary detection
- threshold
- region oriented segmentation morphological image processing dilation and erosion
- structuring element decomposition
- the Strel function
- erosion
- Combining dilation and erosion: Opening and closing the hit and miss transformation.

Introduction

- What is segmentation?



- Three major ways to do.

✓ Thresholding

✓ Edge-based segmentation

✓ Region-based segmentation

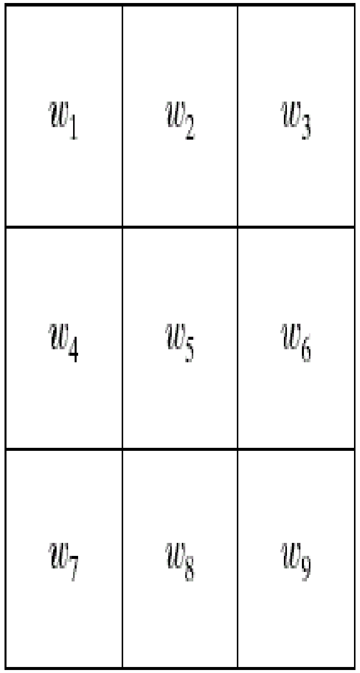
- Image segmentation divides an image into regions that are connected and have some similarity within the region and some difference between adjacent regions.
- The goal is usually to find individual objects in an image.
- For the most part there are fundamentally two kinds of approaches to segmentation: discontinuity and similarity.
  - Similarity may be due to pixel intensity, color or texture.
  - Differences are sudden changes (discontinuities) in any of these, but especially sudden changes in intensity along a boundary line, which is called an edge.

Detection of Discontinuities

- There are three kinds of discontinuities of intensity: points, lines and edges.
- The most common way to look for discontinuities is to scan a small mask over the image. The mask determines which kind of discontinuity to look for.

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

FIGURE 10.1 A general 3 × 3 mask.

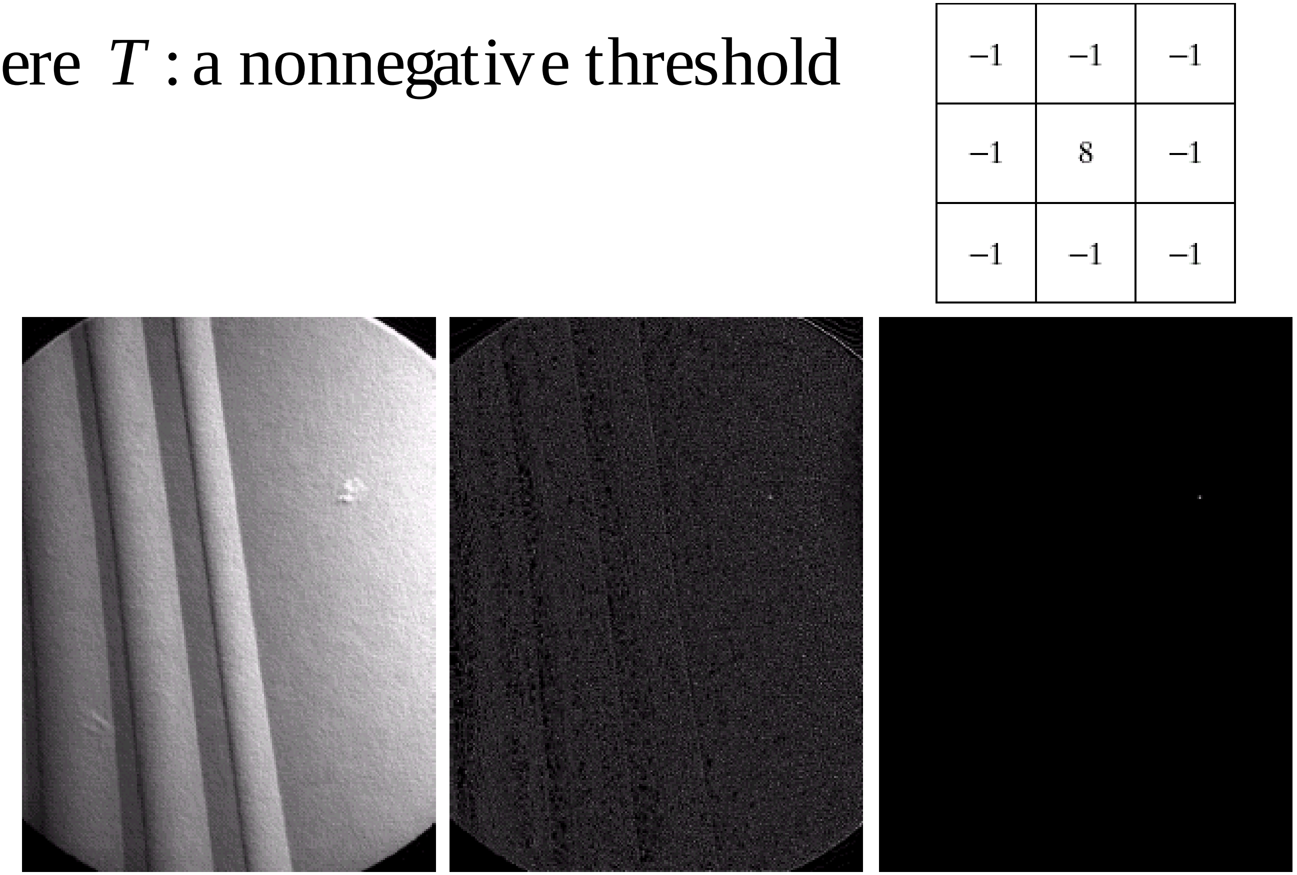


w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Detection of Discontinuities
Point Detection

- ▪ $$|R| \geq T$$

where T : a nonnegative threshold



−1	−1	−1
−1	8	−1
−1	−1	−1

FIGURE 10.2 (a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

Detection of Discontinuities
Point Detection

$$|R| \geq T$$

where T : a nonnegative threshold

−1	−1	−1
−1	8	−1
−1	−1	−1

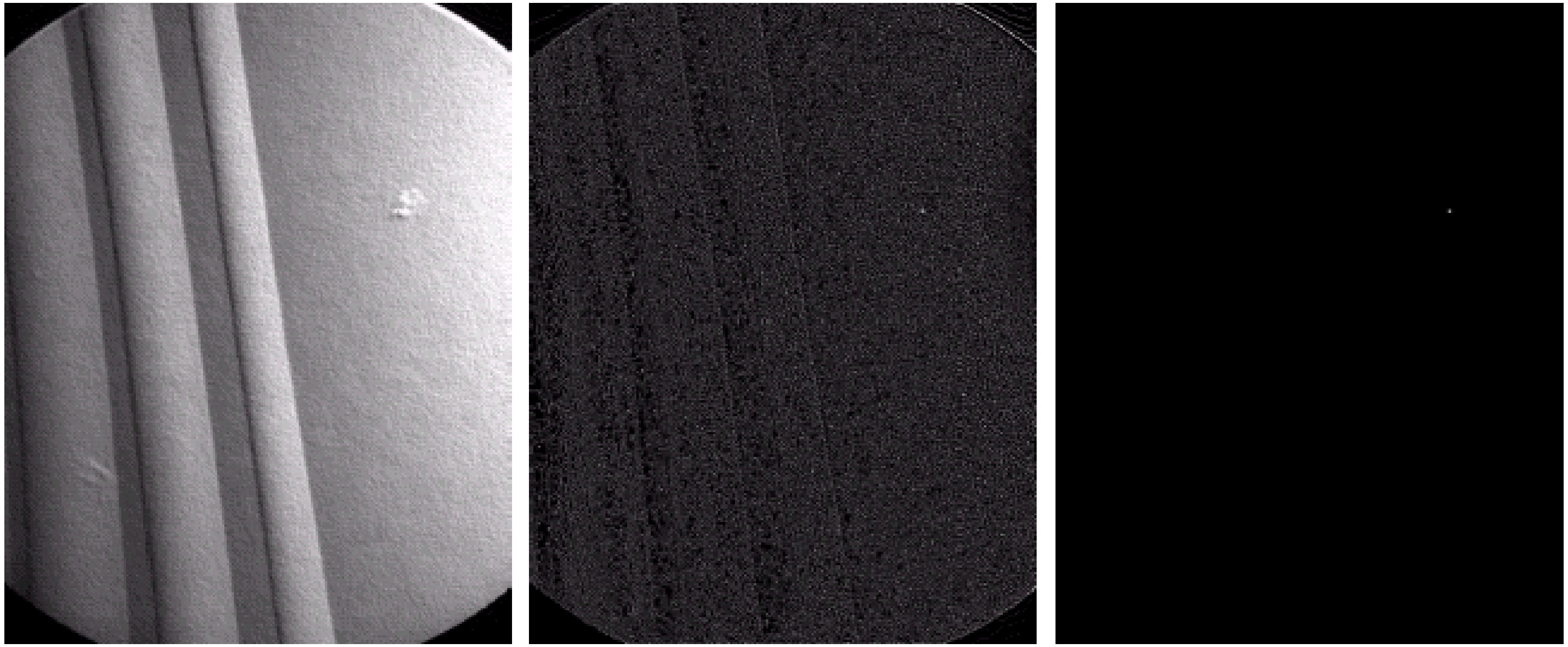


FIGURE 10.2 (a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

Detection of Discontinuities
Point Detection

$$|R| \geq T$$

where T : a nonnegative threshold

–1	–1	–1
–1	8	–1
–1	–1	–1

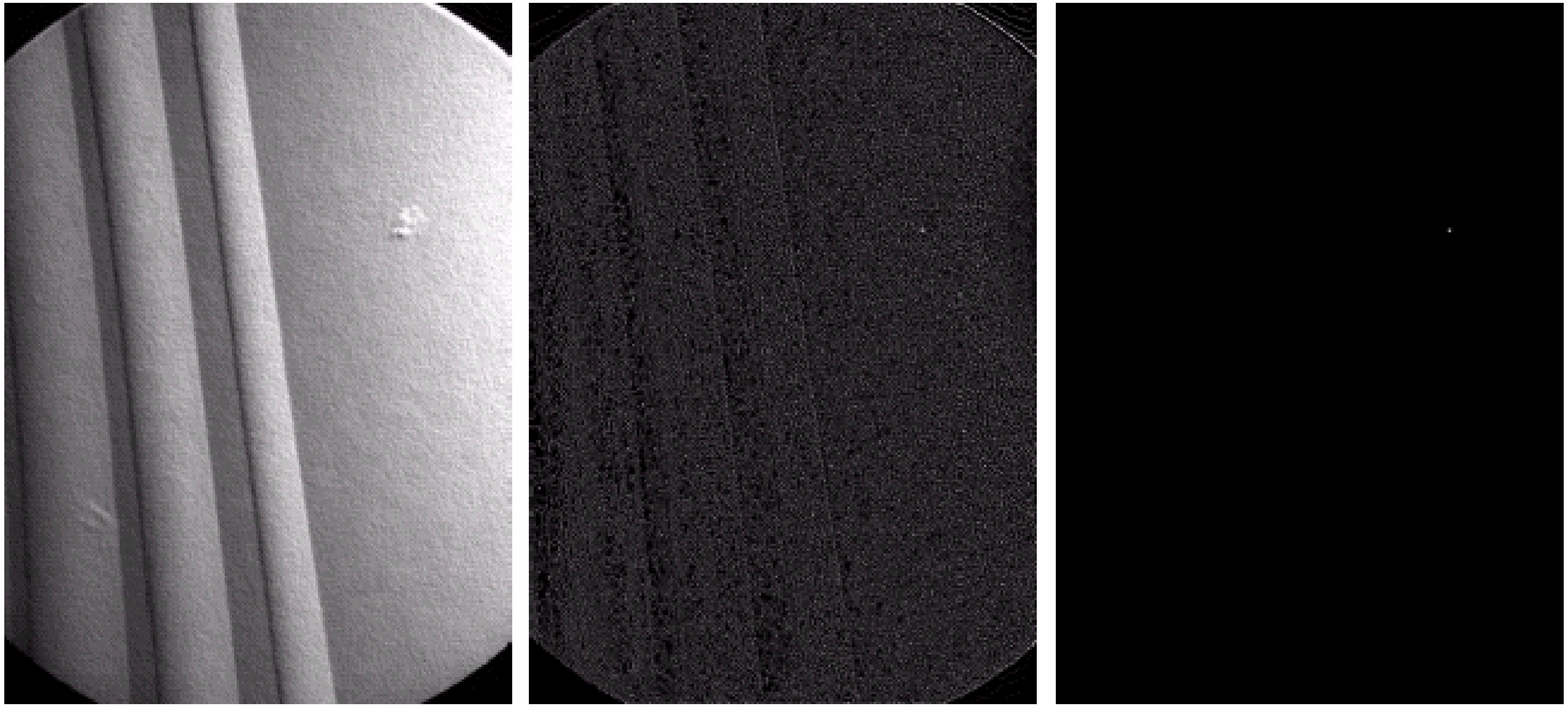
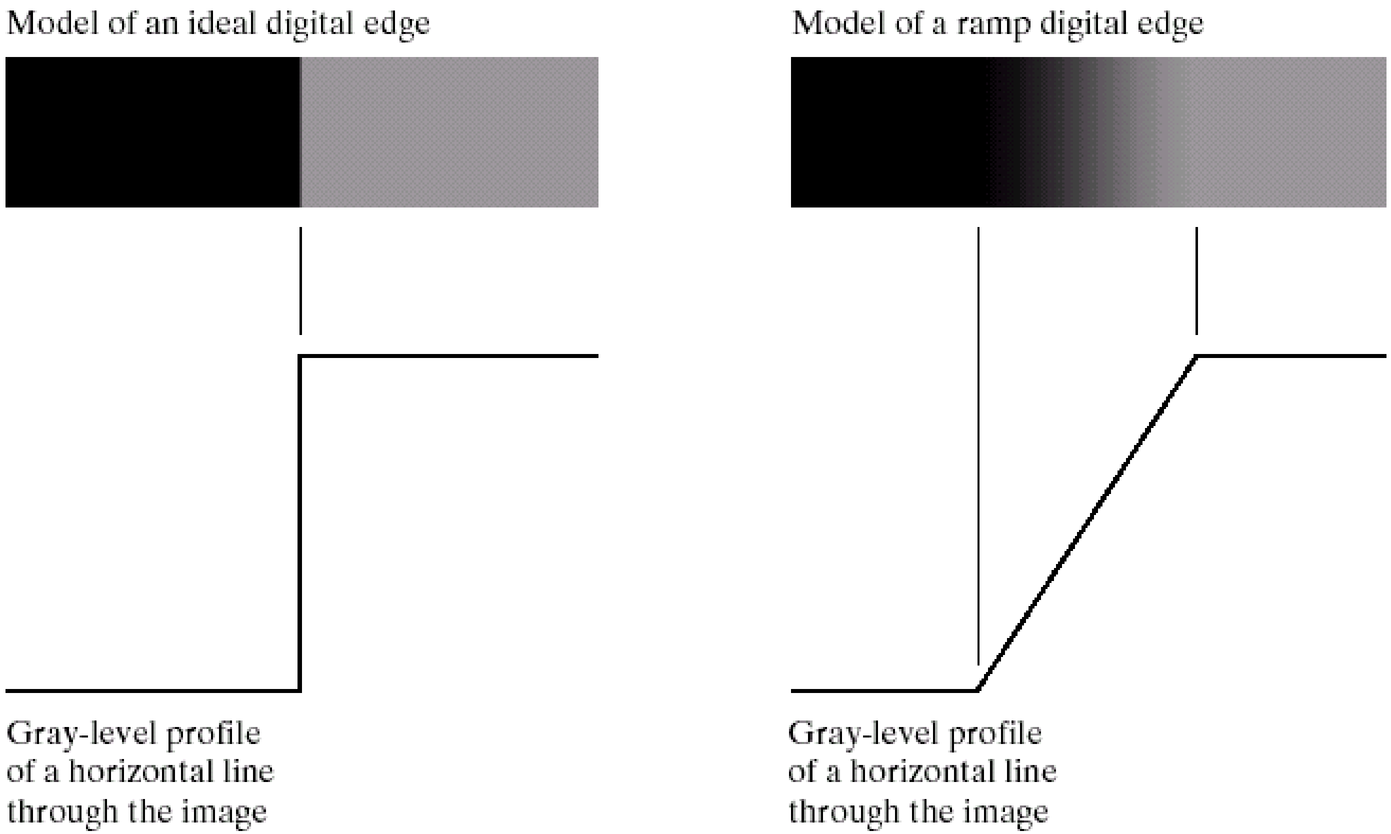


FIGURE 10.2 (a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

Detection of Discontinuities
Edge Detection


Model of an ideal digital edge
Model of a ramp digital edge
Gray-level profile of a horizontal line through the image
Gray-level profile of a horizontal line through the image

a b

FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

Detection of Discontinuities
Edge Detection

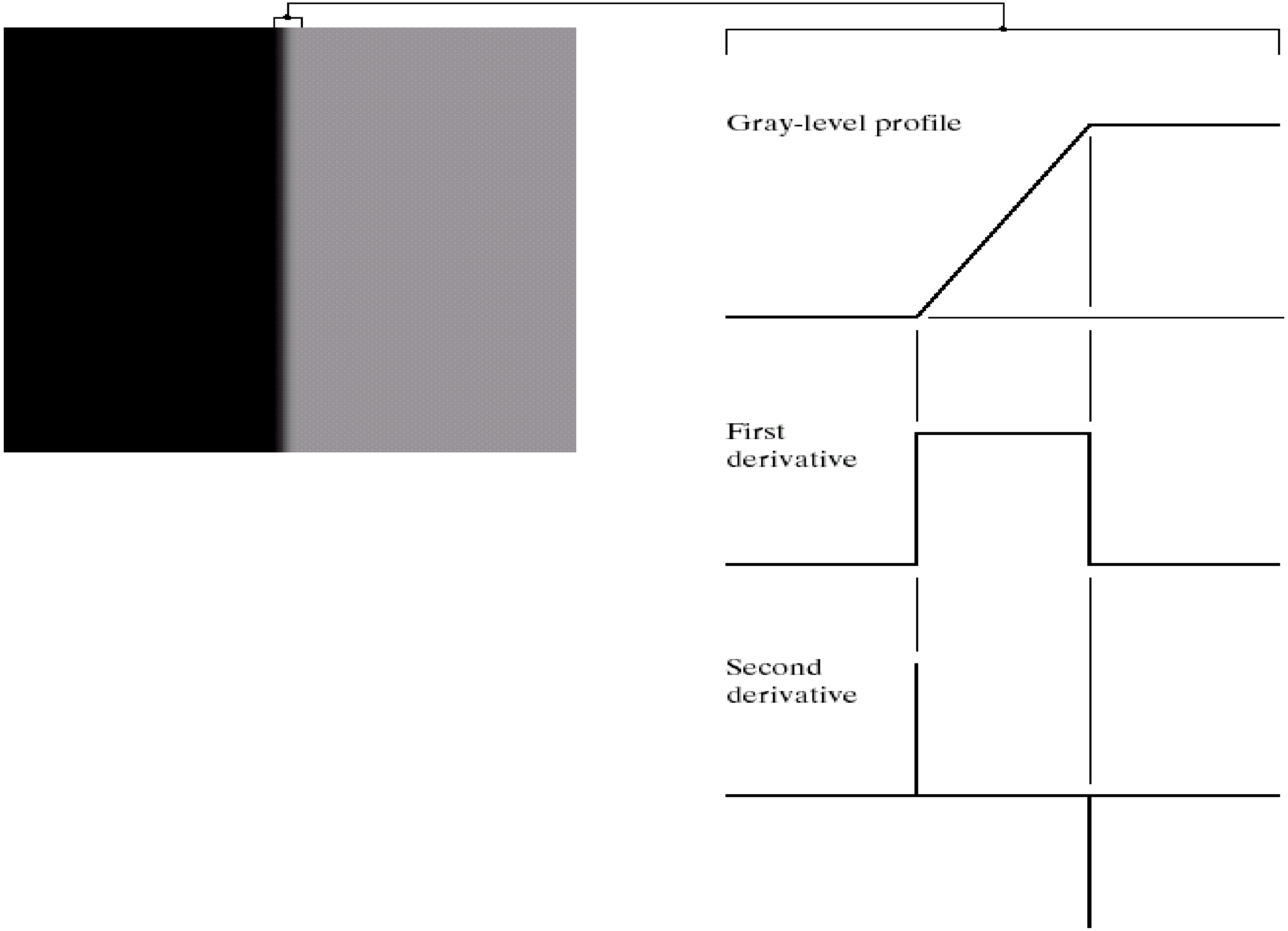


FIGURE 10.6 (a) Two regions separated by a vertical edge. (b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.

Detection of Discontinuities
Edge Detection

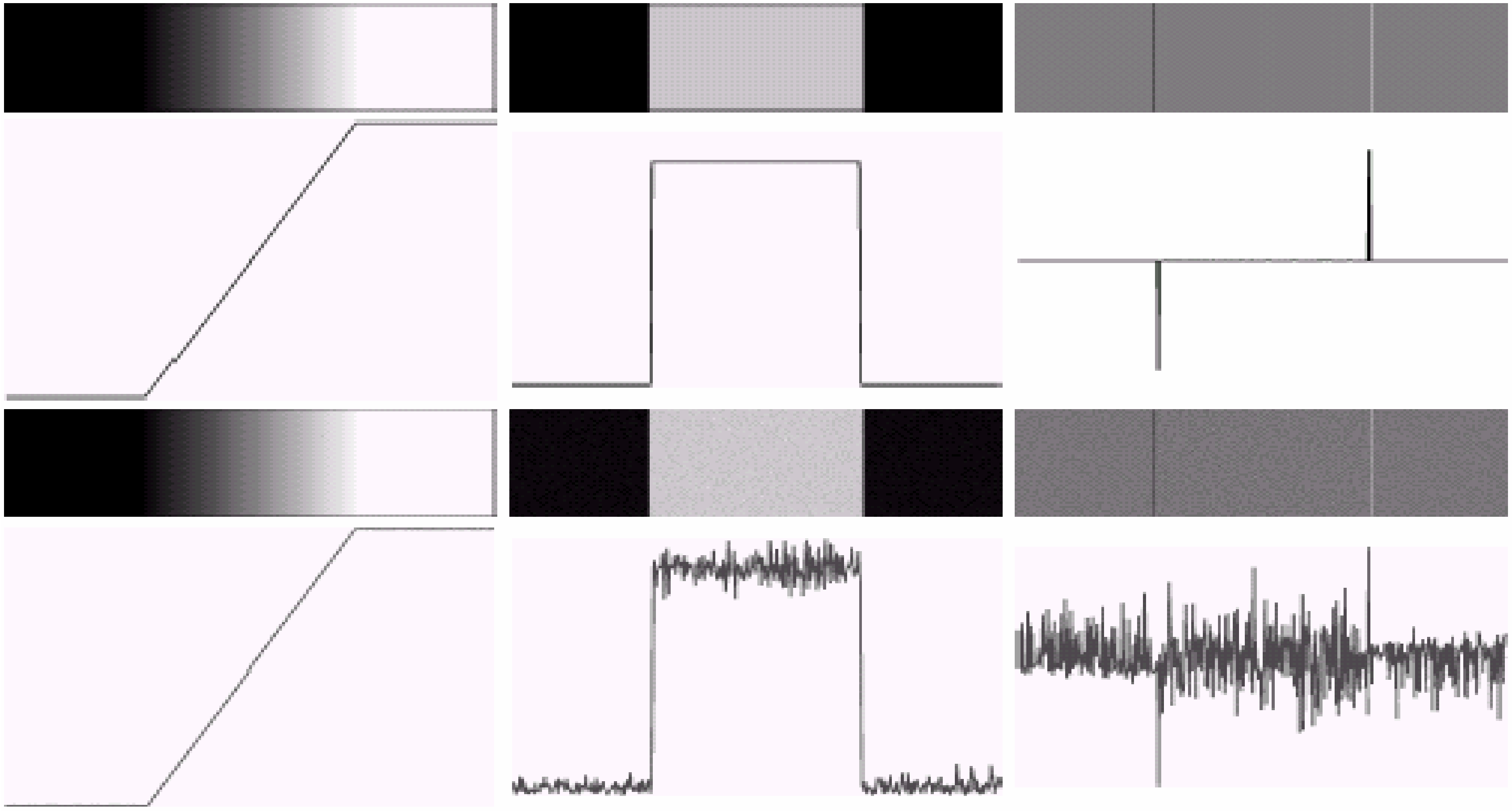


FIGURE 10.7 First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0, 0.1, 1.0$, and 10.0, respectively. Second column: first-derivative images and gray-level profiles. Third column: second-derivative images and gray-level profiles.

a
b
c
d

Detection of Discontinuities
Edge Detection

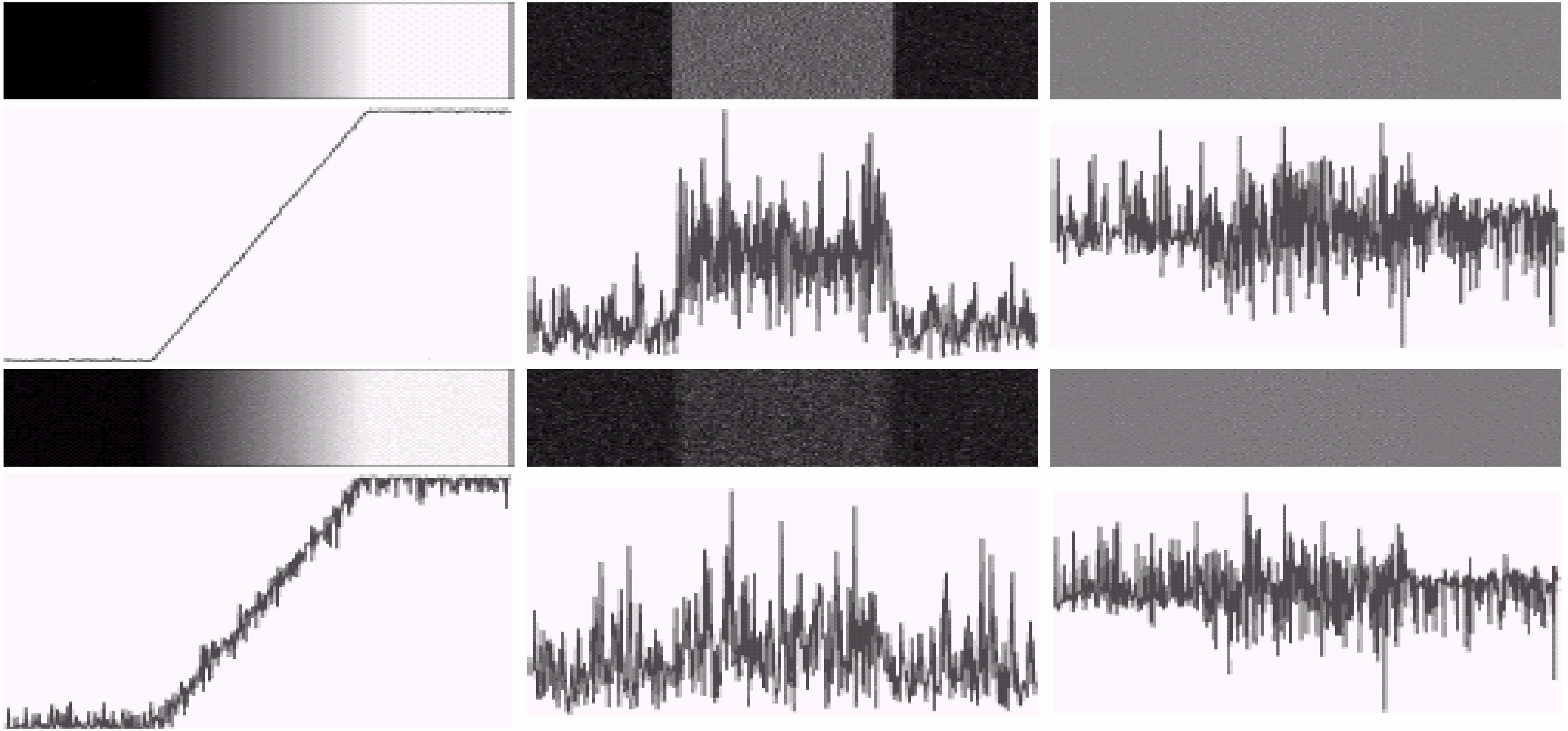


FIGURE 10.7 First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0, 0.1, 1.0,$ and 10.0, respectively. Second column: first-derivative images and gray-level profiles. Third column: second-derivative images and gray-level profiles.

a
b
c
d

Detection of Discontinuities
Gradient Operators

- **First-order derivatives:**
  - **The gradient of an image *f*(*x*,*y*) at location (*x*,*y*) is defined as the vector:**

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

  - **The magnitude of this vector:** $\nabla f = \mathrm{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2}$

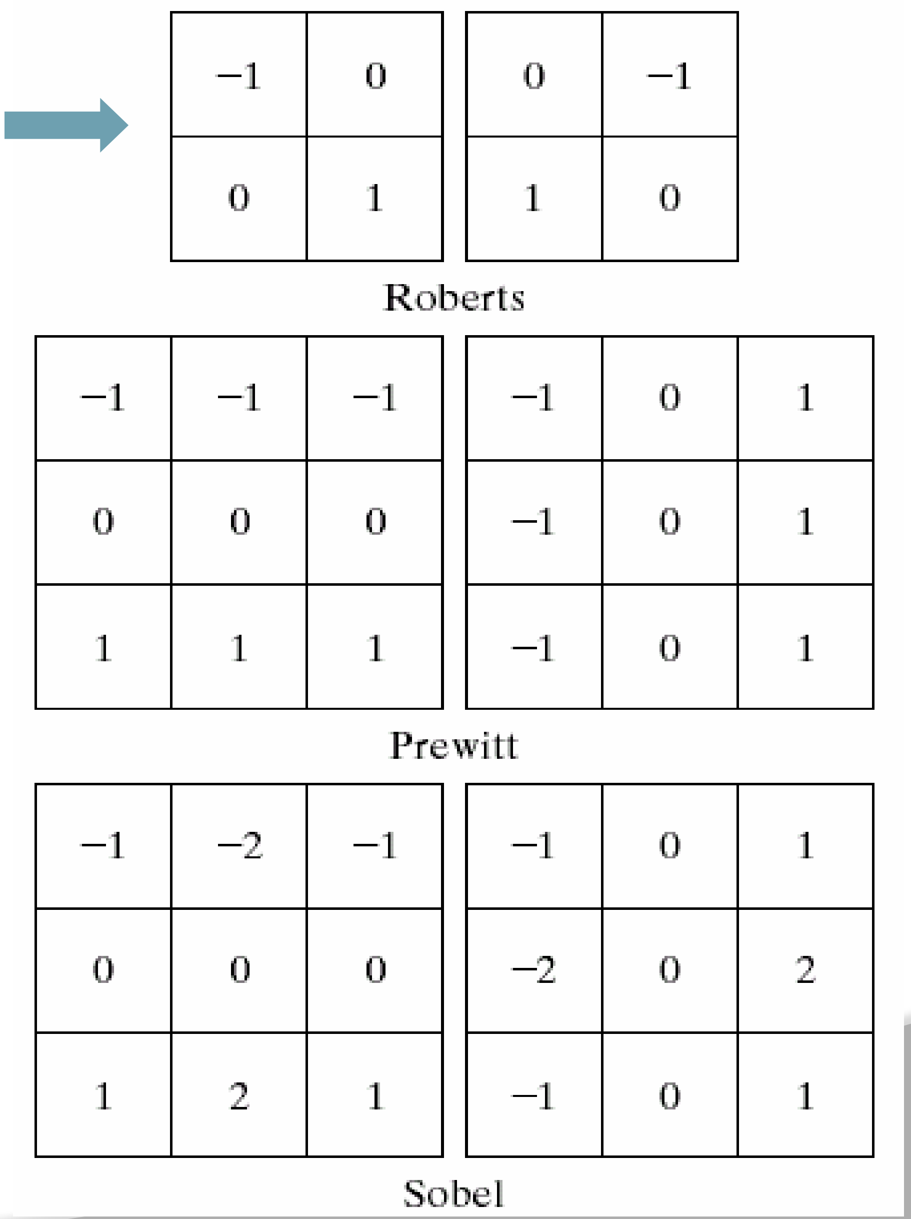
  - **The direction of this vector:** $\alpha(x, y) = \tan^{-1}\left(\frac{G_x}{G_y}\right)$

Detection of Discontinuities
Gradient Operators

Roberts cross-gradient operators

Prewitt operators

Sobel operators



−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

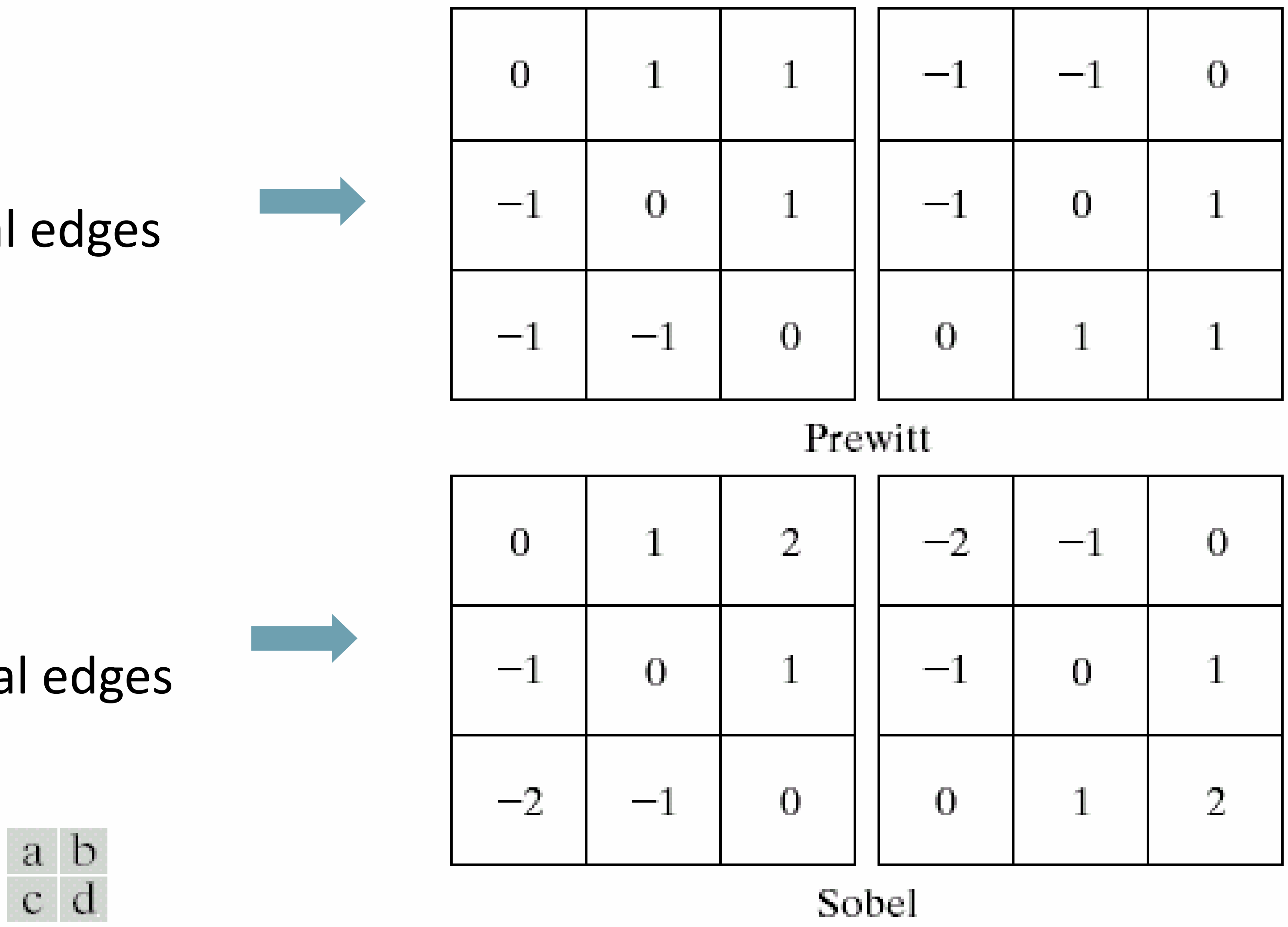
−1	0	1
−2	0	2
−1	0	1

Sobel

Detection of Discontinuities
Gradient Operators

Prewitt masks for detecting diagonal edges

Sobel masks for detecting diagonal edges



0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

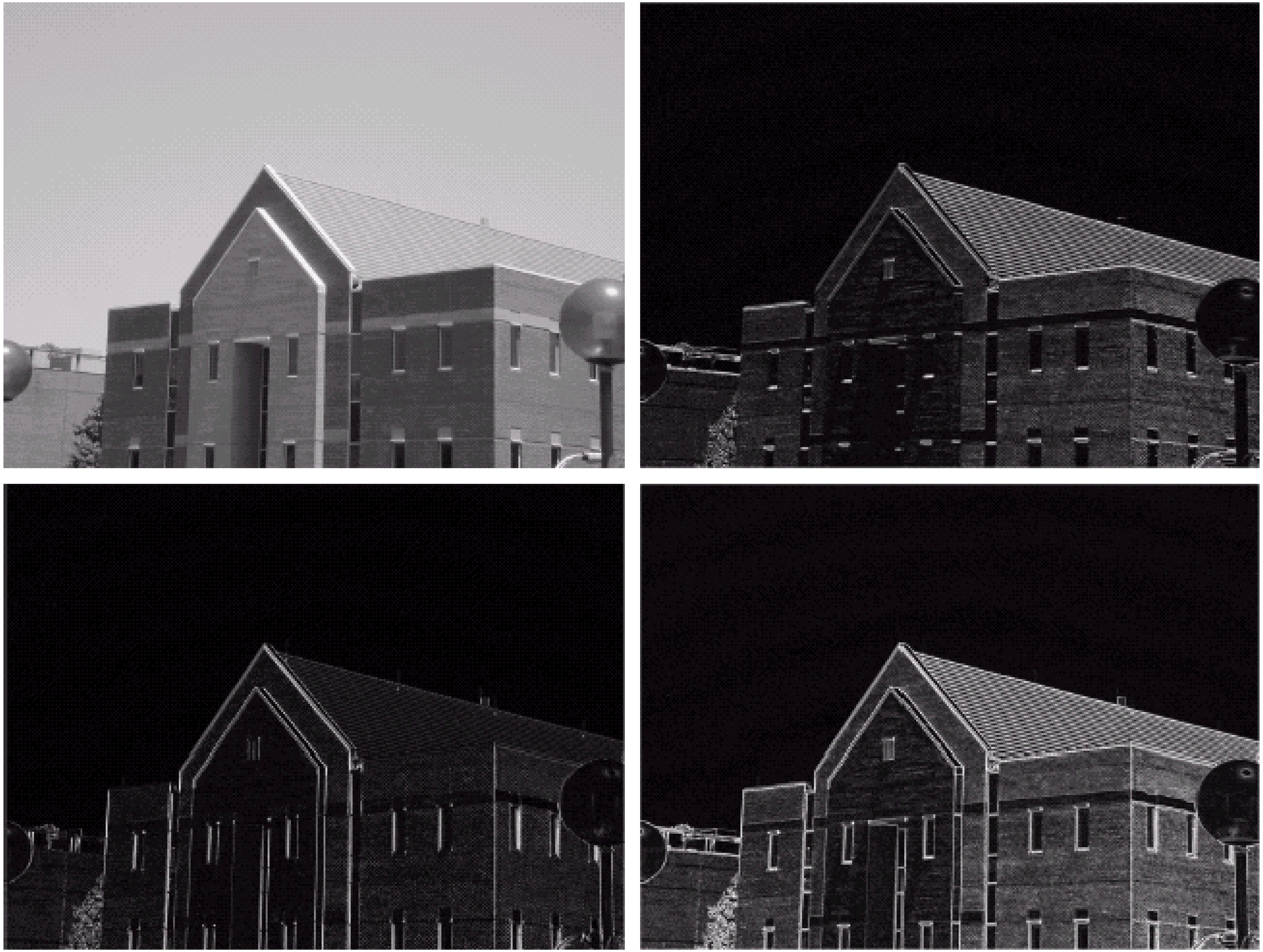
FIGURE 10.9 Prewitt and Sobel masks for detecting diagonal edges.

Detection of Discontinuities
Gradient Operators: Example

a b
c d

FIGURE 10.10 (a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.

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



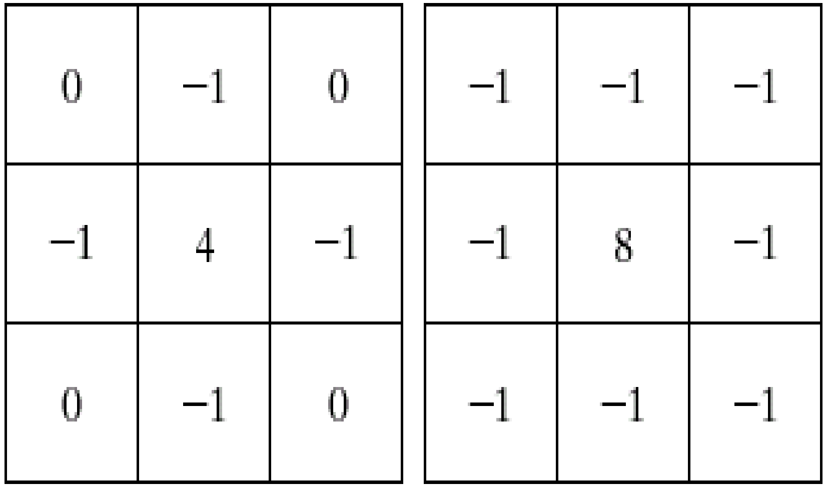
Detection of Discontinuities
Gradient Operators

- **Second-order derivatives: (The Laplacian)**
  - **The Laplacian of an 2D function *f*(*x*,*y*) is defined as**

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

  - **Two forms in practice:**

FIGURE 10.13 Laplacian masks used to implement Eqs. (10.1-14) and (10.1-15), respectively.



0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Detection of Discontinuities
Gradient Operators

- **Consider the function:**

A Gaussian function

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}} \quad \text{where} \quad r^2 = x^2 + y^2$$

$$\text{and} \quad \sigma : \text{the standard deviation}$$

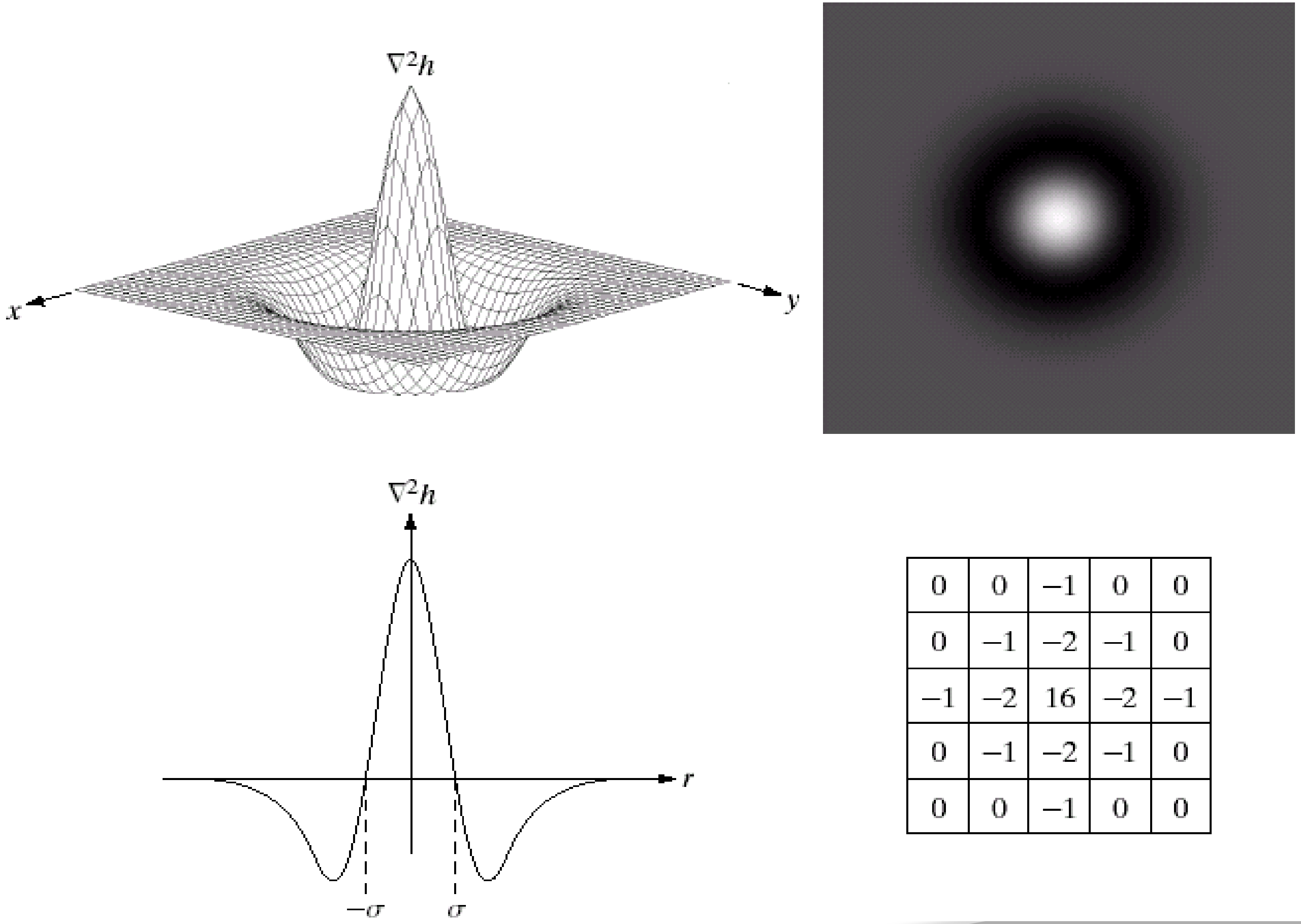
- **The Laplacian of *h* is**

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

The Laplacian of a Gaussian (LoG)

- **The Laplacian of a Gaussian sometimes is called the Mexican hat function. It also can be computed by smoothing the image with the Gaussian smoothing mask, followed by application of the Laplacian mask.**

Detection of Discontinuities
Gradient Operators



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

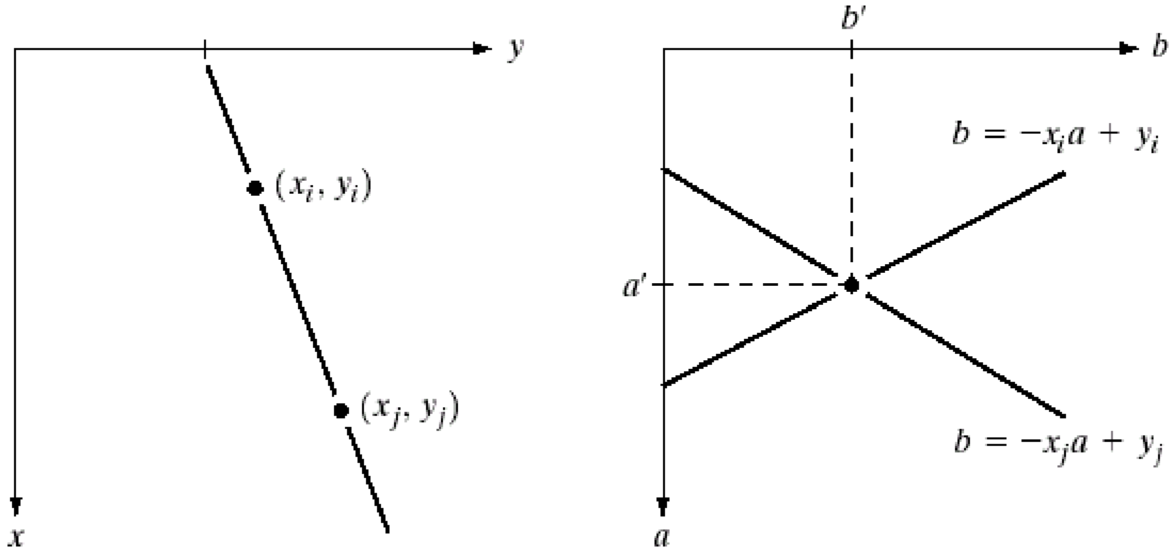
a b
c d

FIGURE 10.14 Laplacian of a Gaussian (LoG). (a) 3-D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross section showing zero crossings. (d) 5 × 5 mask approximation to the shape of (a).

Edge Linking and Boundary Detection
Global Processing via the Hough Transform

- **Hough transform: a way of finding edge points in an image that lie along a straight line.**
- **Example: *xy*-plane v.s. *ab*-plane (parameter space)**

$$y_i = ax_i + b$$

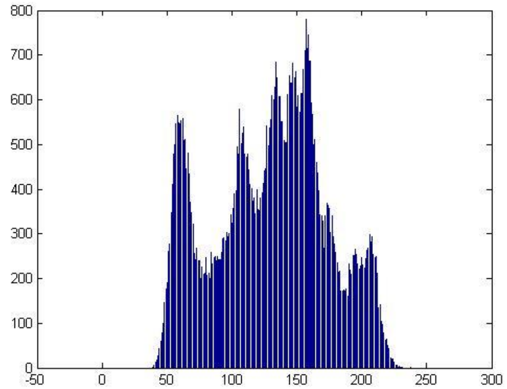


a b

FIGURE 10.17
(a) *xy*-plane.
(b) Parameter space.

Thresholding

- **Finding histogram of gray level intensity.**

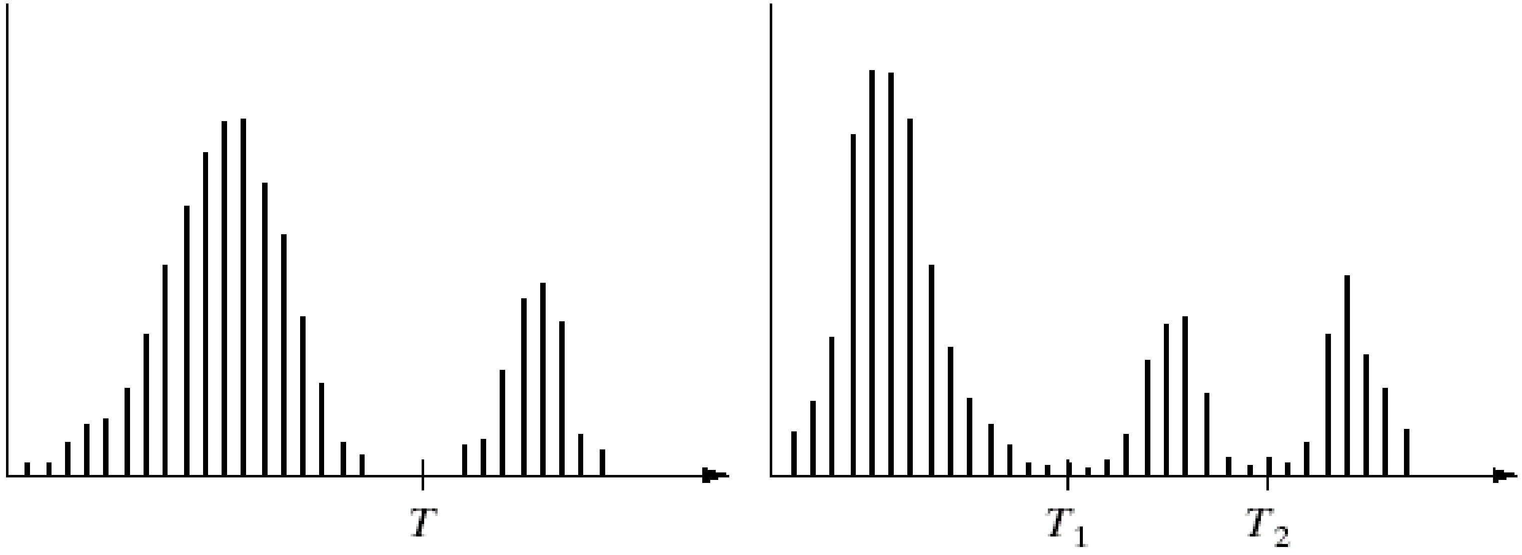


- ✓ **Basic Global ng**
- ✓ **Otsu’s Method**
- ✓ **Multiple Threshold**
- ✓ **Variable Thresholding**

Thresholding

- **Assumption: the range of intensity levels covered by objects of interest is different from the background.**

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

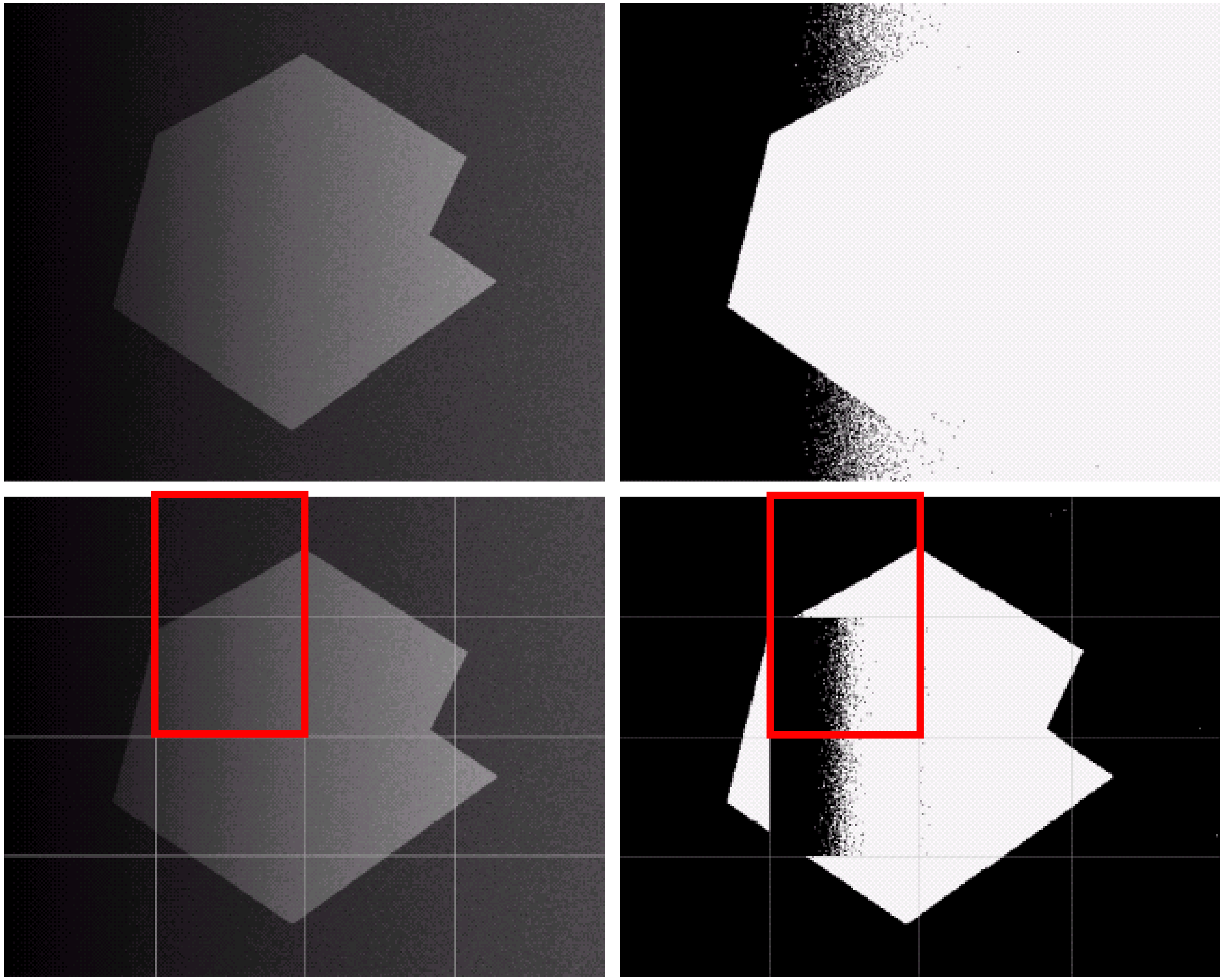


a b

FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

Thresholding
Basic Adaptive Thresholding

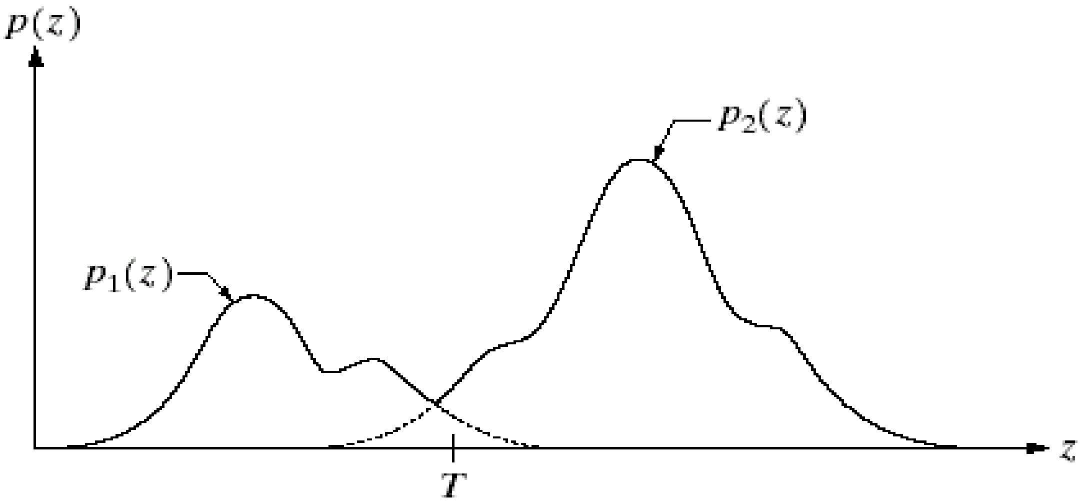
FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.



Thresholding
Optimal Global and Adaptive Thresholding

- This method treats pixel values as **probability density functions**.
- The goal of this method is to **minimize the probability of misclassifying pixels** as either object or background.
- There are two kinds of error:
  - mislabeling an object pixel as background, and
  - mislabeling a background pixel as object.

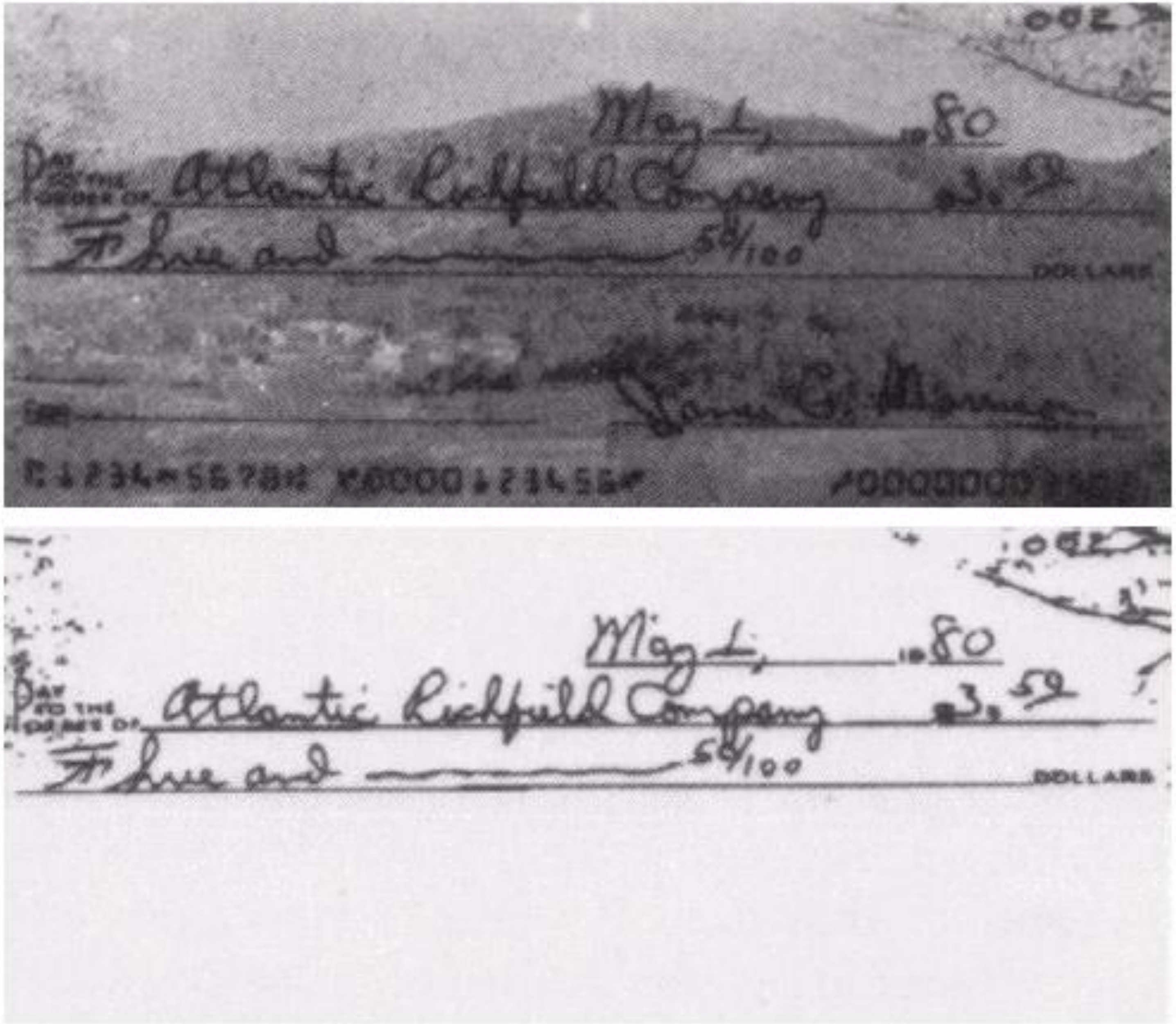
FIGURE 10.32 Gray-level probability density functions of two regions in an image.



Thresholding
Use of Boundary Characteristics

a
b

FIGURE 10.37 (a) Original image. (b) Image segmented by local thresholding. (Courtesy of IBM Corporation.)



Region-Based Segmentation

- Edges and thresholds sometimes do not give good results for segmentation.

- Region-based segmentation is based on the connectivity of similar pixels in a region.
  - Each region must be uniform.
  - Connectivity of the pixels within the region is very important.
  -

- There are two main approaches to region-based segmentation: region growing and region splitting.

Region-Based Segmentation
Basic Formulation

- Let *R* represent the entire image region.
- Segmentation is a process that partitions *R* into subregions, $R_1, R_2, \ldots, R_n$, such that

$$\text{(a)}\ \bigcup_{i=1}^{n} R_i = R$$

$$\text{(b)}\ R_i \text{ is a connected region, } i = 1, 2, \ldots, n$$

$$\text{(c)}\ R_i \cap R_j = \phi \ \text{ for all } i \text{ and } j, i \neq j$$

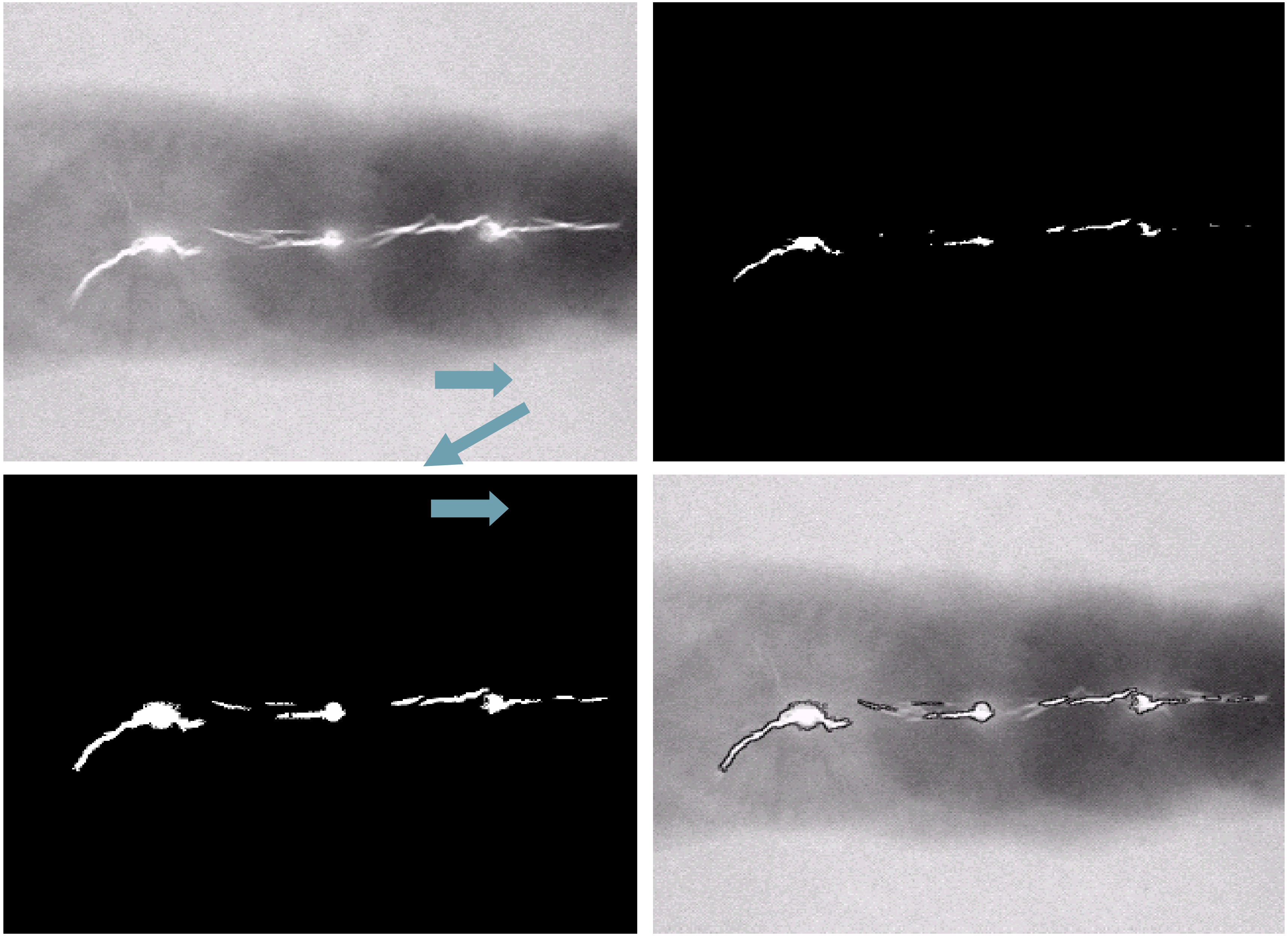
$$\text{(d)}\ P(R_i) = \text{TRUE for } i = 1, 2, \ldots, n$$

$$\text{(e)}\ P(R_i \cup R_j) = \text{FALSE} \ \text{ for any adjacent regions } R_i \text{ and } R_j$$

where $P(R_k)$: a logical predicate defined over the points in set R_k

For example: $P(R_k)$=TRUE if all pixels in R_k have the same gray level.

Region-Based Segmentation
Region Growing



a b
c d

FIGURE 10.40 (a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).

Region-Based Segmentation
Region Growing

- **Fig.1(a) shows the histogram of Fig.1(a) .It is difficult to segment the defects by thresholding methods. (Applying region growing methods are better in this case.)**

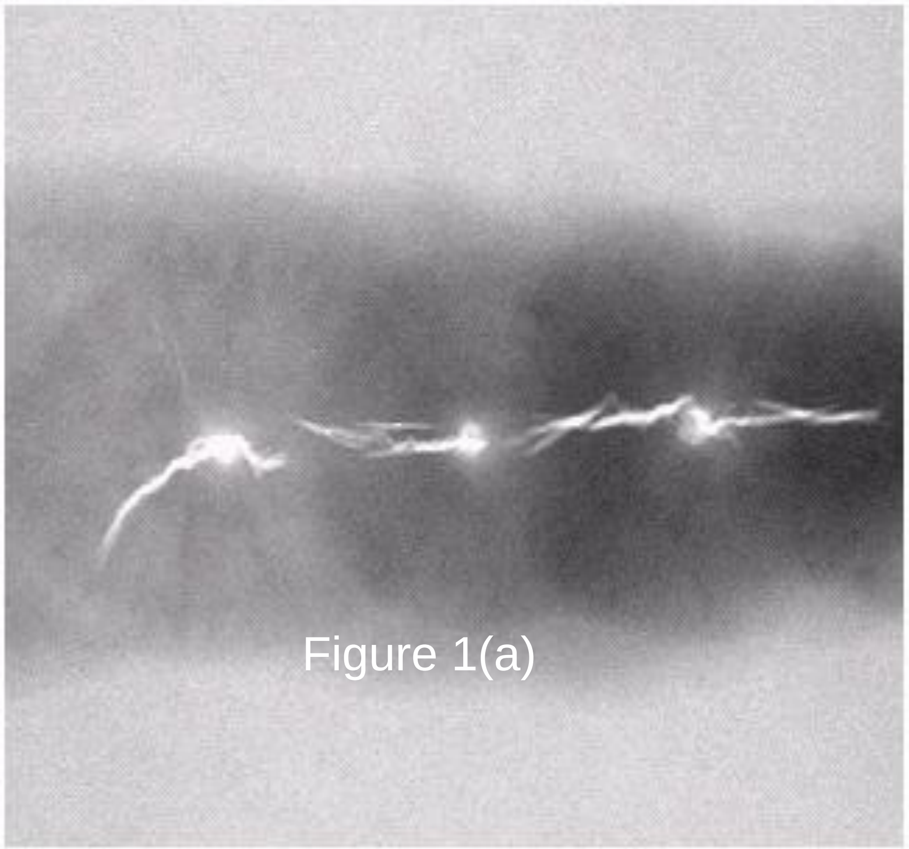


Figure 1(a)

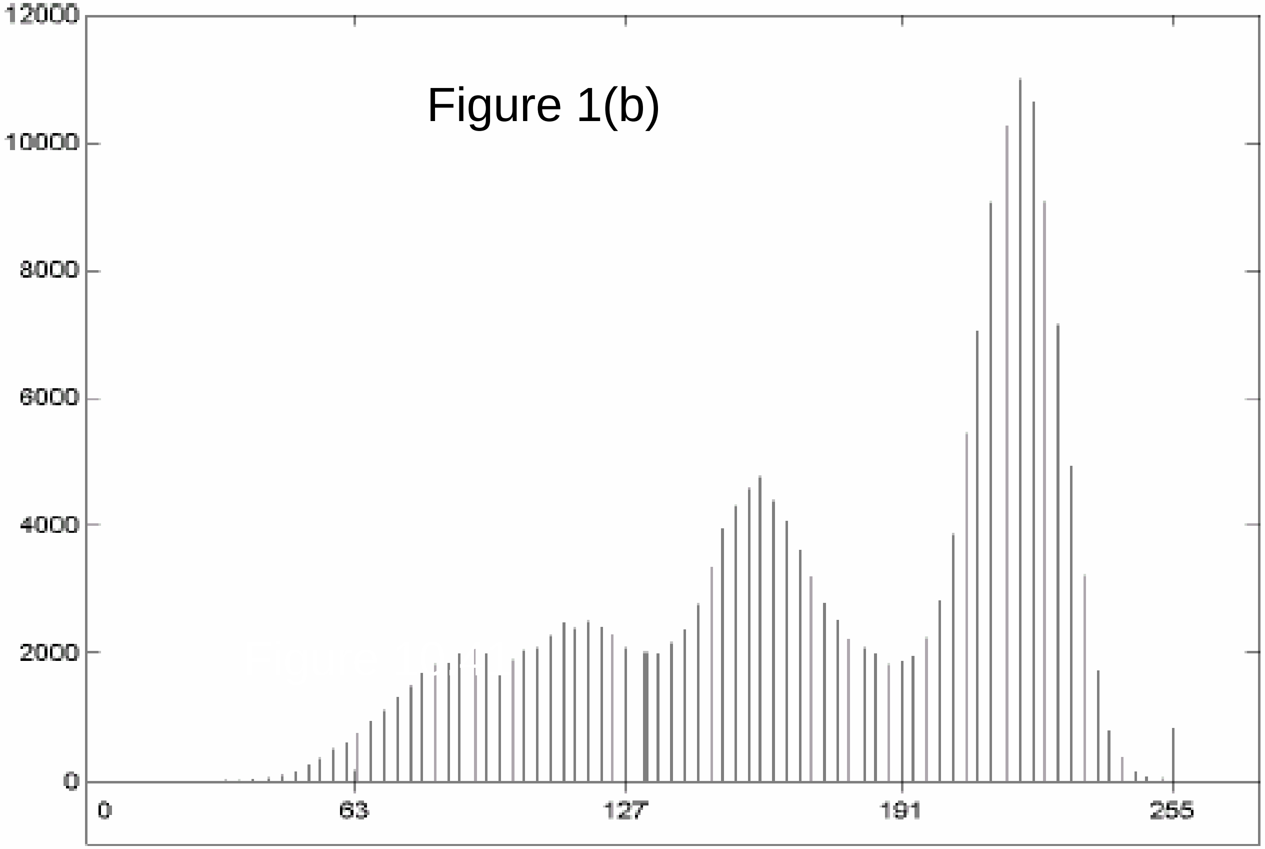


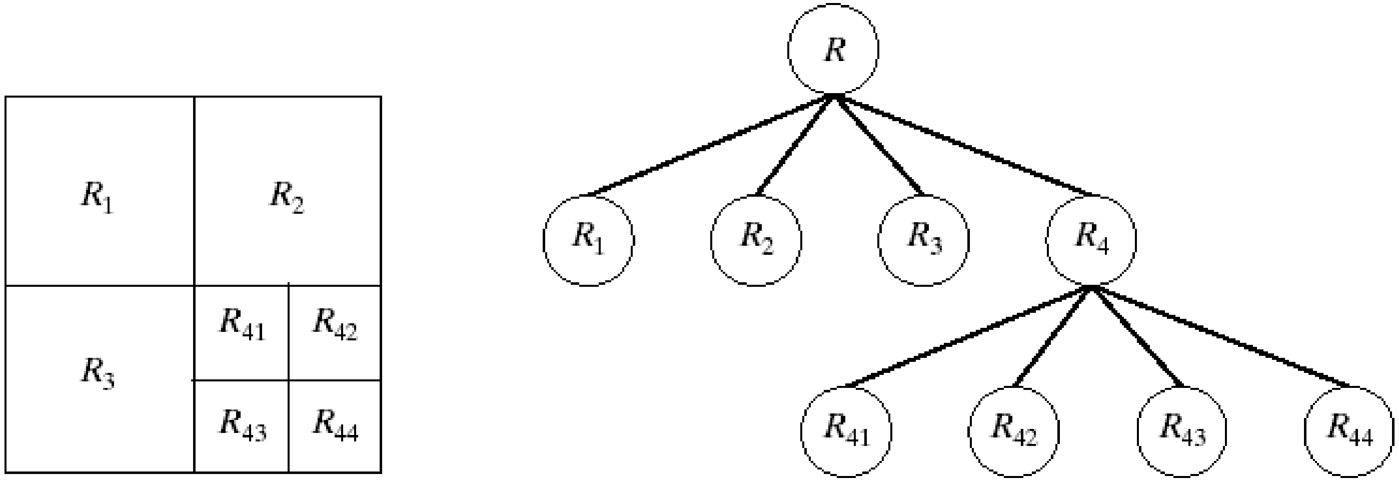
Figure 1(b)

Region-Based Segmentation
Region Splitting and Merging

- **Region splitting is the opposite of region growing.**
  - **First there is a large region (possible the entire image).**
  - **Then a predicate (measurement) is used to determine if the region is uniform.**
  - **If not, then the method requires that the region be split into two regions.**
  - **Then each of these two regions is independently tested by the predicate (measurement).**
  - **This procedure continues until all resulting regions are uniform.**

Region-Based Segmentation
Region Splitting

- The main problem with region splitting is determining where to split a region.
- One method to divide a region is to use a quad tree structure.
- Quadtree: a tree in which nodes have exactly four descendants.



a b

FIGURE 10.42 (a) Partitioned image. (b) Corresponding quadtree.

Region-Based Segmentation
Region Splitting and Merging

- **The split and merge procedure:**
  - **Split into four disjoint quadrants any region R_i for which $P(R_i)$ = FALSE.**
  - **Merge any adjacent regions R_j and R_k for which $P(R_j U R_k)$ = TRUE. (the quadtree structure may not be preserved)**
  - **Stop when no further merging or splitting is possible.**

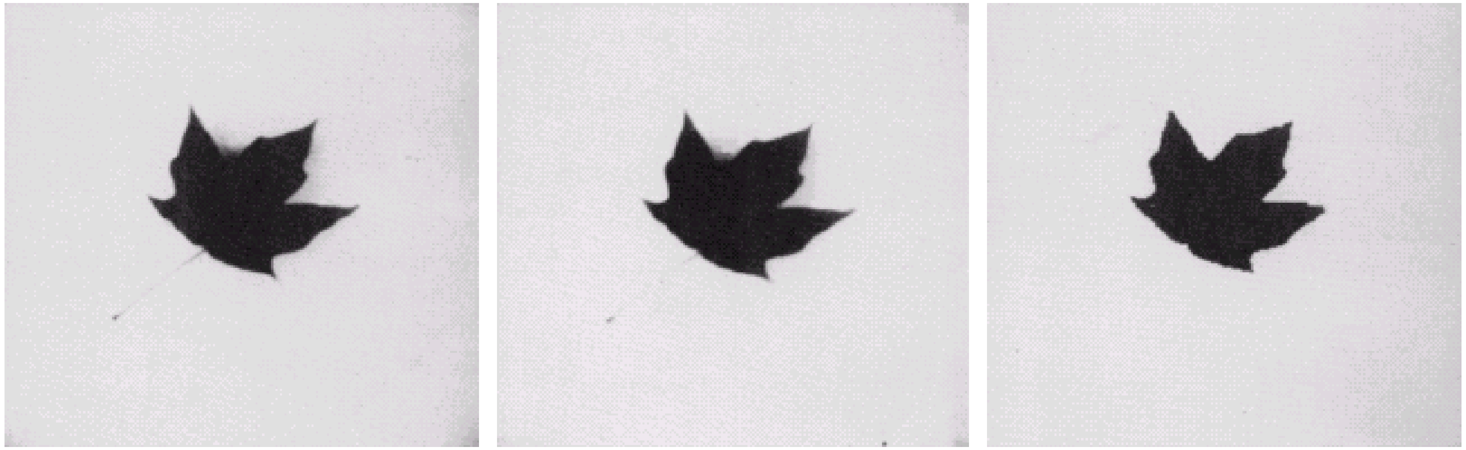


FIGURE 10.43 (a) Original image. (b) Result of split and merge procedure. (c) Result of thresholding (a).

Segmentation by Morphological Watersheds

- The concept of watersheds is based on visualizing an image in three dimensions: two spatial coordinates versus gray levels.
- In such a topographic interpretation, we consider three types of points:
  - (a) points belonging to a regional minimum
  - (b) points at which a drop of water would fall with certainty to a single minimum
  - (c) points at which water would be equally likely to fall to more than one such minimum
- The principal objective of segmentation algorithms based on these concepts is to find the watershed lines.

Binary Morphological Processing

- Non-linear image processing technique
  - Order of sequence of operations is important
    - Linear: (3+2)*3 = (5)*3=15

      3*3+2*3=9+6=15
    - Non-linear: $(3+2)^2 + (5)^2 = 25$ [sum, then square]

      $(3)^2 + (2)^2 = 9+4=13$ [square, then sum]
- Based on geometric structure
- Used for edge detection, noise removal and feature extraction

  → Used to ‘understand’ the shape/form of a binary image

Introduction

- **Structuring Element**
- **Erosion**
- **Dilation**
- **Opening**
- **Closing**
- **Hit-and-miss**
- **Thinning**
- **Thickening**

Image – Set of Pixels

- **Basic idea is to treat an object within an image as a set of pixels (or coordinates of pixels)**
- **In binary images, pixels that are ‘off’, set to 0, are background and appear black.**
- **Foreground pixels (objects) are 1 and appear white**

- Neighborhood
- Set of pixels defined by their location relation to the pixel of interest
  - Defined by structuring element
  - Specified by connectivity

- Connectivity-
  - ‘4-connected’
  - ‘8-connected’

Translation of Object A by vector b

- **Define Translation ob object A by vector b:**

  $A_t = \{ t \in I^2 : t = a+b, a \in A \}$

  Where I^2 is the two dimensional image space that contains the image

- **Definition of DILATION is the UNION of all the translations:**

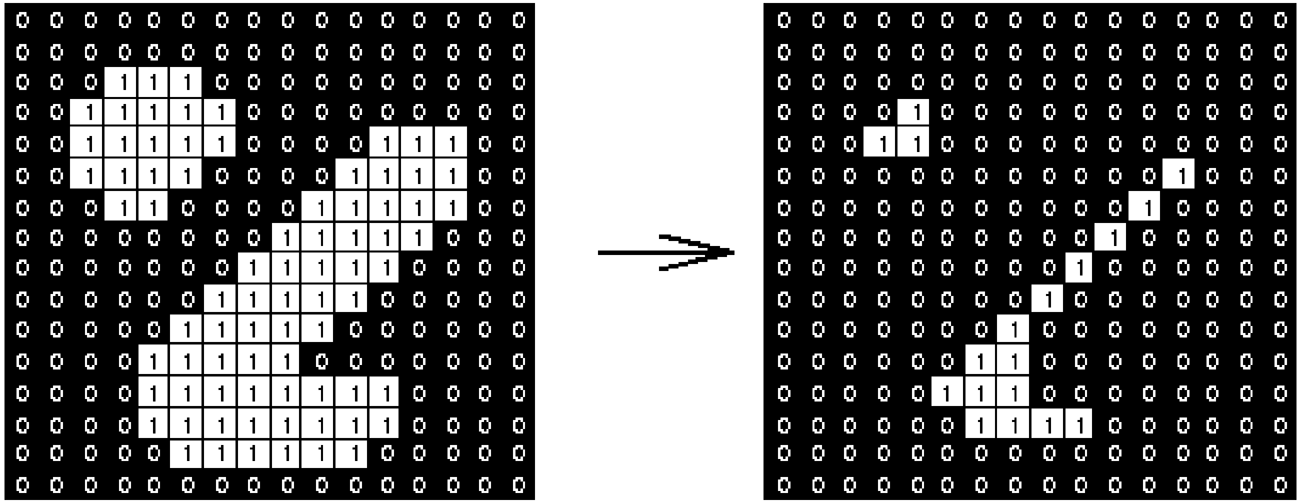
  $A\oplus B = \cup \{ t \in I^2 : t = a+b, a \in A \}$ **for all b's in B**

DILATION

- **The shape of B determines the final shape of the dilated object.**

- **B acts as a geometric filter that changes the geometric structure of A**

- **Erosion is an important morphological operation**
- **Applied Structuring Element:**



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

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),

(-1, 0), (0, 0), (1, 0),

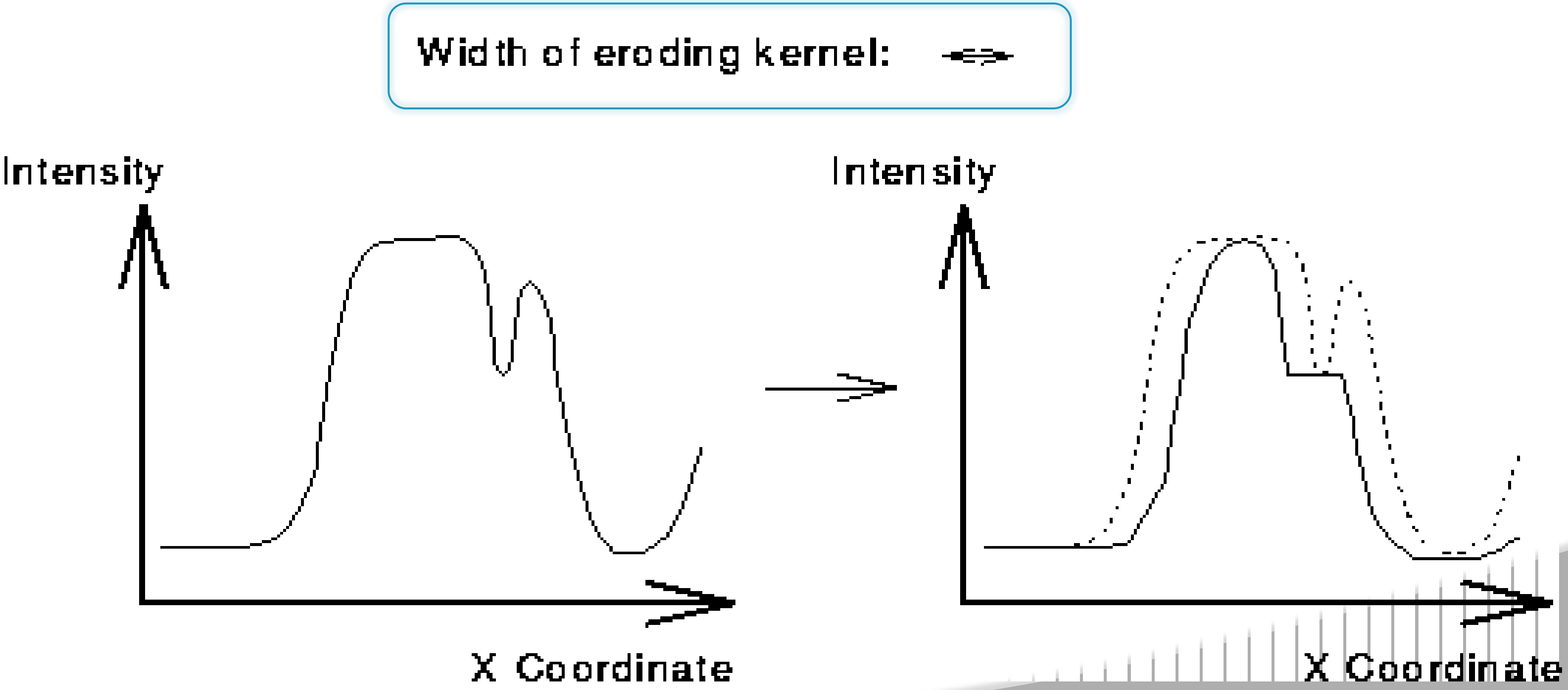
(-1, 1), (0, 1), (1, 1) }

- Basic operations
- Are dual to each other:
  - Erosion shrinks foreground, enlarges Background
  - Dilation enlarges foreground, shrinks background

Erosion

- Erosion is the set of all points in the image, where the structuring element “fits into”.
- Consider each foreground pixel in the input image
  - If the structuring element fits in, write a “1” at the origin of the structuring element!
- Simple application of pattern matching
- Input:
  - Binary Image (Gray value)
  - Structuring Element, containing only 1s!

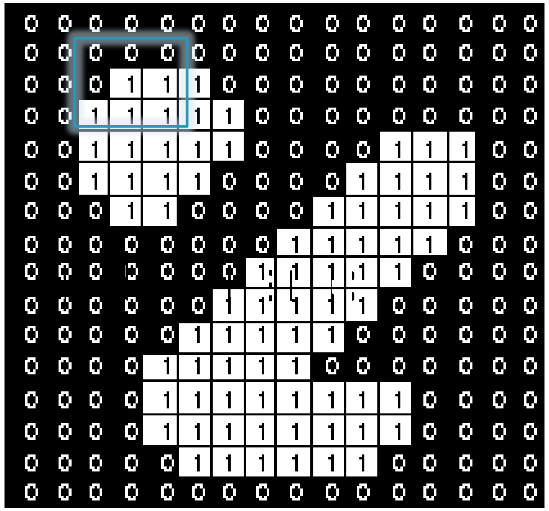
- View gray value images as a stack of binary images!



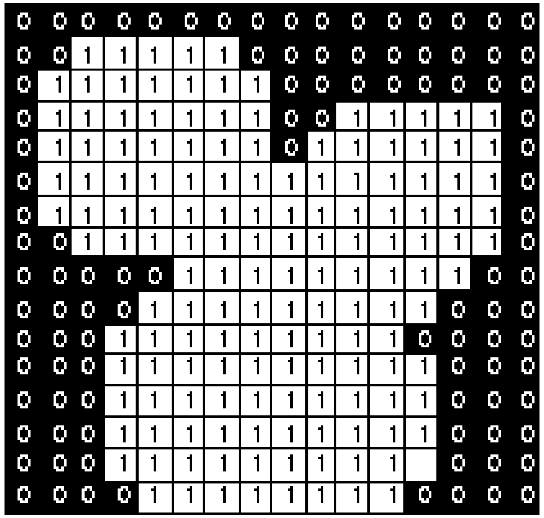
Intensity is lower so the image is darker

Example: Dilation

- Dilation is an important morphological operation
- Applied Structu



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

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),

(-1, 0), (0, 0), (1, 0),

(-1, 1), (0, 1), (1, 1) }

Dilation

- Dilation is the set of all points in the image, where the structuring element “touches” the foreground.
- Consider each pixel in the input image
  - If the structuring element touches the foreground image, write a “1” at the origin of the structuring element!
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!!

Dilation and Erosion

- **DILATION: Adds pixels to the boundary of an object**
- **EROSIN: Removes pixels from the boundary of an object**
- **Number of pixels added or removed depends on size and shape of structuring element**

- Important operations
- Derived from the fundamental operations
  - Dilatation
  - Erosion
- Usually applied to binary images, but gray value images are also possible
- Opening and closing are dual operations

Opening

- Similar to Erosion
  - Spot and noise removal
  - Less destructive
- Erosion next dilation
- the *same structuring element for both operations.*
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

Opening

- **Take the structuring element (SE) and slide it around *inside* each foreground region.**
  - **All pixels which can be covered by the SE with the SE being entirely within the foreground region will be preserved.**
  - **All foreground pixels which can *not* be reached by the structuring element without lapping over the edge of the foreground object will be eroded away!**
- **Opening is idempotent: Repeated application has no further effects!**

- **Structuring element: 3x3 square**

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

Closing

- Similar to Dilation
  - Removal of holes
  - Tends to enlarge regions, shrink background
- Closing is defined as a Dilatation, followed by an Erosion *using the same structuring element for both operations.*
- Dilation next erosion!
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

Closing

- Take the structuring element (SE) and slide it around *outside* each foreground region.

  - All background pixels which can be covered by the SE with the SE being entirely within the background region will be preserved.
  - All background pixels which can *not* be reached by the structuring element without lapping over the edge of the foreground object will be turned into a foreground.

- Opening is idempotent: Repeated application has no further effects!

DIGITAL IMAGE PROCESSING

UNIT-V

IMAGE COMPRESSION

- Introduction
- Huffman coding
- Run-length coding
- Modified READ coding
- LZW compression
- Predictive coding
- Transform image coding

INTRODUCTION

Digital image coding and compression

Techniques and algorithms concerned with the minimization of the memory needed to represent and store digital images

- *Compression factors*

  Transmission and storing of large images.

  Reduce of baud rate.

  Reduce of transmission time.

TYPES OF COMPRESSION TECHNIQUES

Lossless compression techniques

- These are used when raw image data are difficult to obtain or contain vital information that may be destroyed
- by compression, e.g. in medical diagnostic imaging.

Lossy compression techniques

- These can be used when raw image data can be easily reproduced or when the information loss can be tolerated at the receiver site, e.g. in Digital Television, Teleconferencing

ENTROPY CODING

Pulse Coding Modulation (PCM) using *B* bits/pixel.

The average number of bits per pixel can be reduced by assigning binary codes of different bit length to the various image intensities.

The pdf (probability density function) *p(i)* can be estimated by calculating the digital image histogram.

Assignment of short code words to intensities having a hgh probability of occurrence and larger code words to less frequent image intensity levels.

HUFFMAN CODING

The image intensity levels are coded by using variable length codewords.

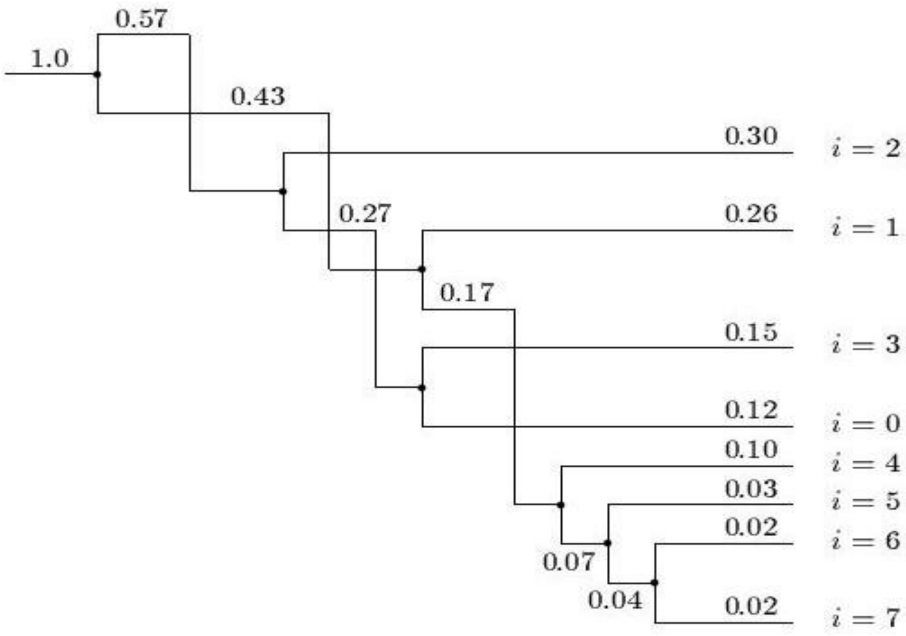
No codeword can be the prefix of another codeword

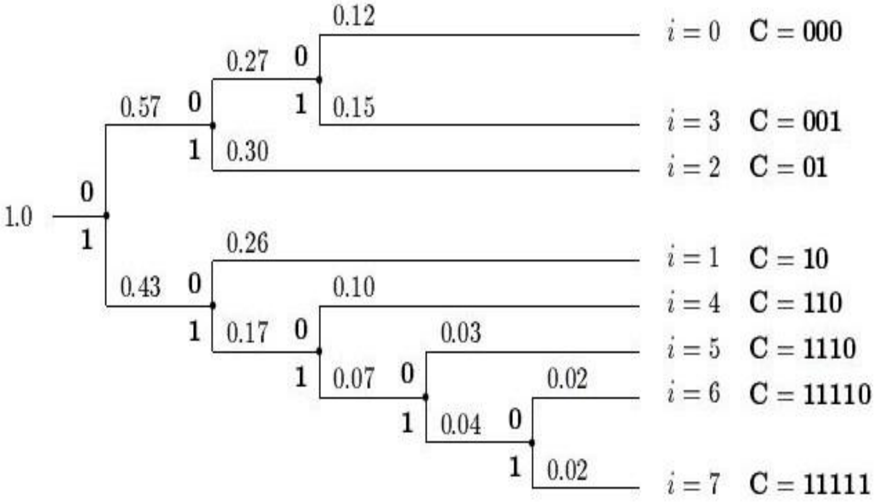
Average codeword length

$$H(B) \le L \le H(B) + 1$$

HUFFMAN CODING

Figure1: (a) Construction of Huffman code tree,
(b) tree rearrangement





Run-length coding

Each image line can be represented as follows:

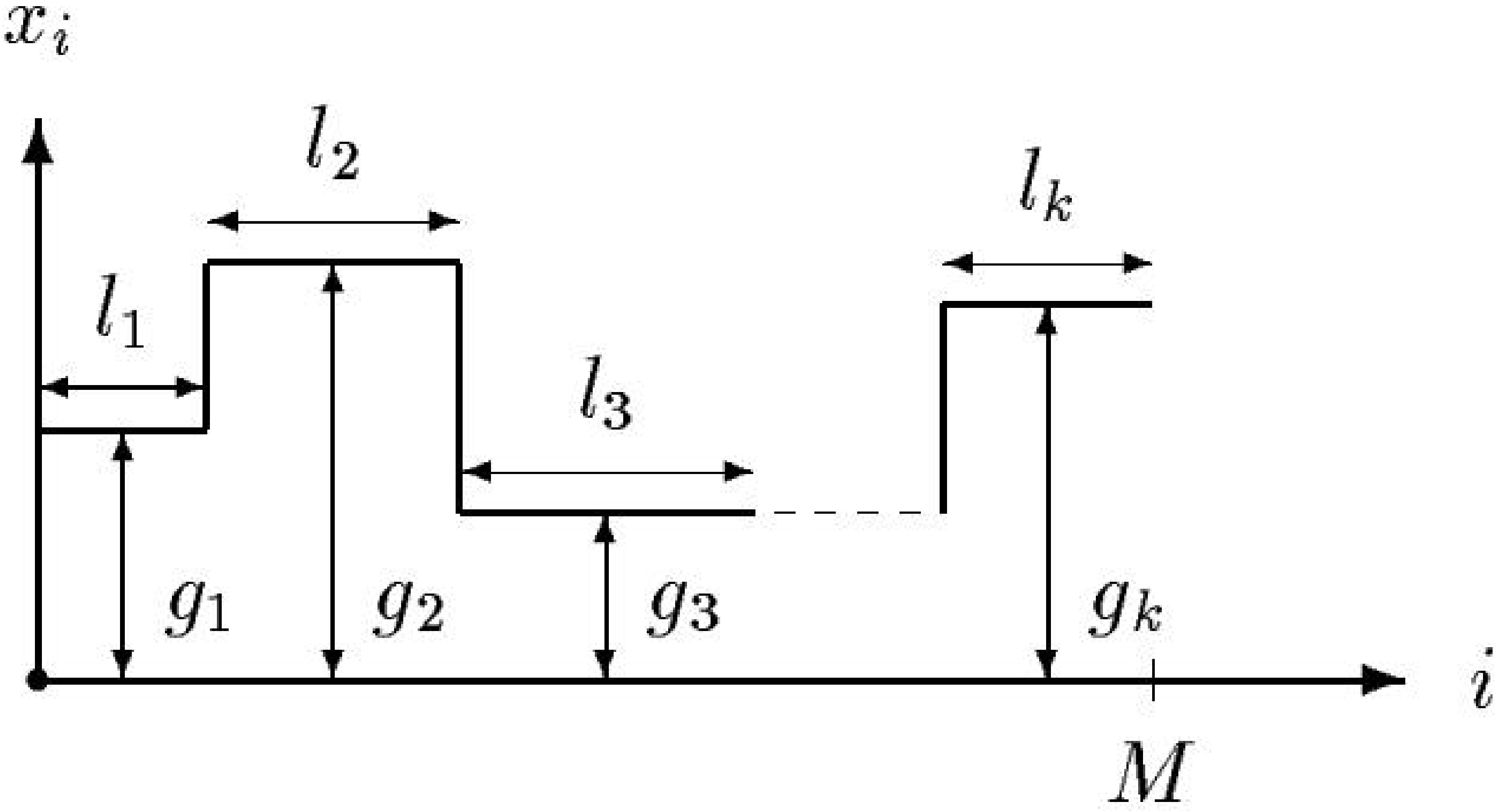
$(x_1, \ldots, x_M)$ ⍰ $(g_1, l_1), (g_2, l_2), \ldots, (g_k, l_k)$

where:

$g_1 = x_1, \; g_k = x_M$

Each couple (g_i, l_i) is called gray-level run.

Run-length coding



RUN-LENGTH CODING

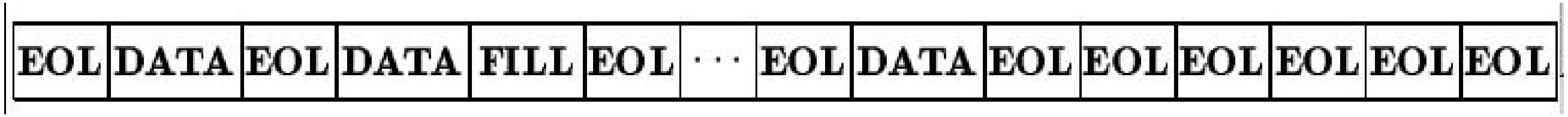
- The resulting compression is considerable if the grey-level runs are relatively large.

- The savings are even larger when the image is binary.

- An end of line (EOL) code word indicates the start of an image and the end of a line.

RUN-LENGTH CODING

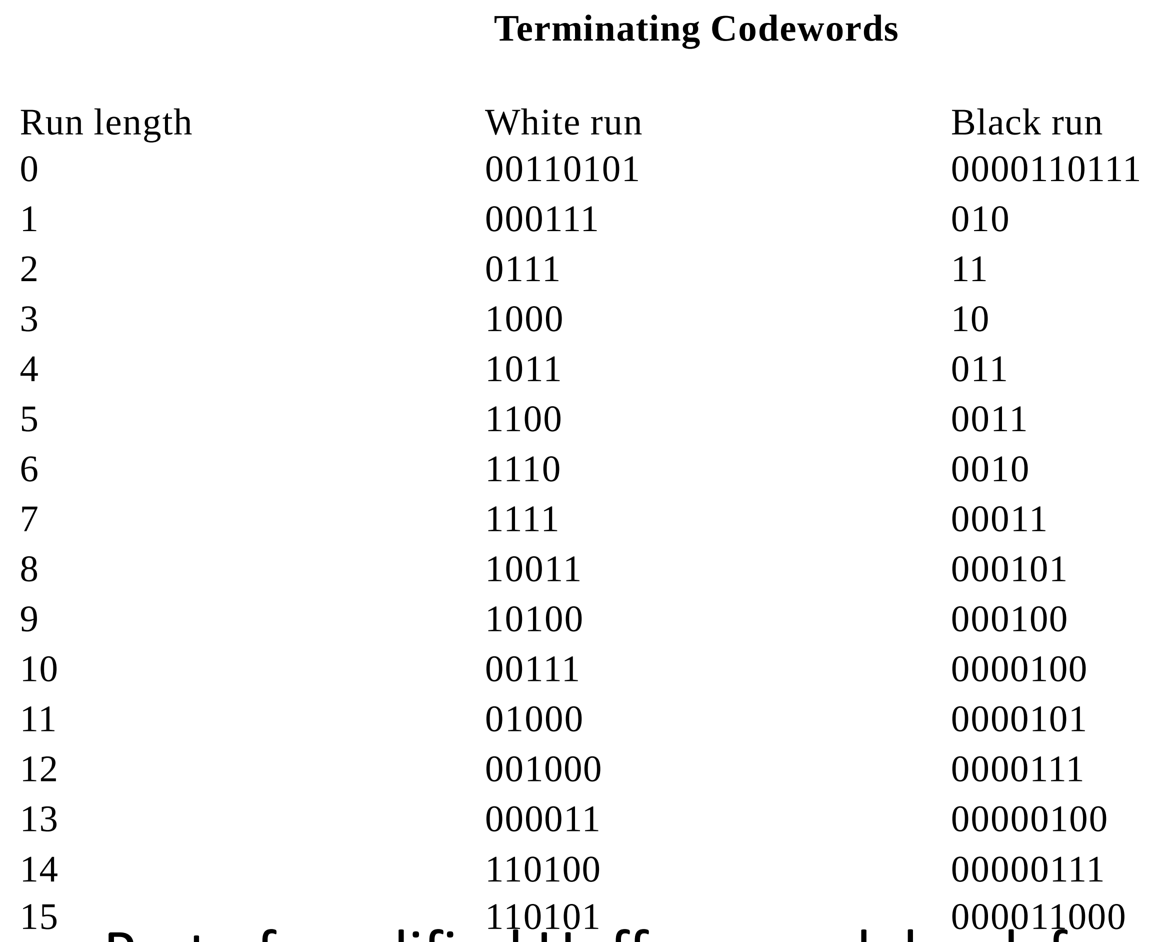
- The run-length coding has been standardized by CCITT.
- It has been included in Group 3 coding schemes (FAX transmission).



EOL	DATA	EOL	DATA	FILL	EOL	· · ·	EOL	DATA	EOL	EOL	EOL	EOL	EOL	EOL

Figure: Format of an image coded by a run-length code.

RUN-LENGTH CODING



Terminating Codewords

Run length	White run	Black run
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111
13	000011	00000100
14	110100	00000111
15	110101	000011000

Figure : Part of modified Huffman codebook for run-length coding (CCITT).

LZW COMPRESSION

◆General-purpose compression scheme proposed by Lempel-Ziv and Welch.

◆It can be used for the compression of any binary data file.

◆It is incorporated in several de facto image storage standards (e.g. TIFF, GIF).

LZW COMPRESSION

- It is a lossless, fast and effective algorithm and can operate on images of any bit depth.

- LZW compression is based on the construction of a code table that maps frequently encountered bit strings to output codewords.

- The digital image as well as the coded one is treated as a one-dimensional bit string.

LZW COMPRESSION

- The speed of both LZW compression and decompression depends on the implementation of the code table and on the efficient searching in it.

- The decompression is usually faster because no search is needed.

- The compression ratio ranges from 1:1.5 to 1:3.

- Substantial compression can be obtained for binary or bitmap images (moderate compression for raw greyscale images).

PREDICTIVE CODING

- One way to describe information redundancy in digital images is to use predictability in local image neighbourhoods.

- The pixel intensity *f(n,m)* can be predicted from the intensities of its neighbourhooding pixels *Á*:

$$\hat{f}(n,m) = L\left[f_r(n-i,m-j), \qquad (i,j)\in A\right]$$

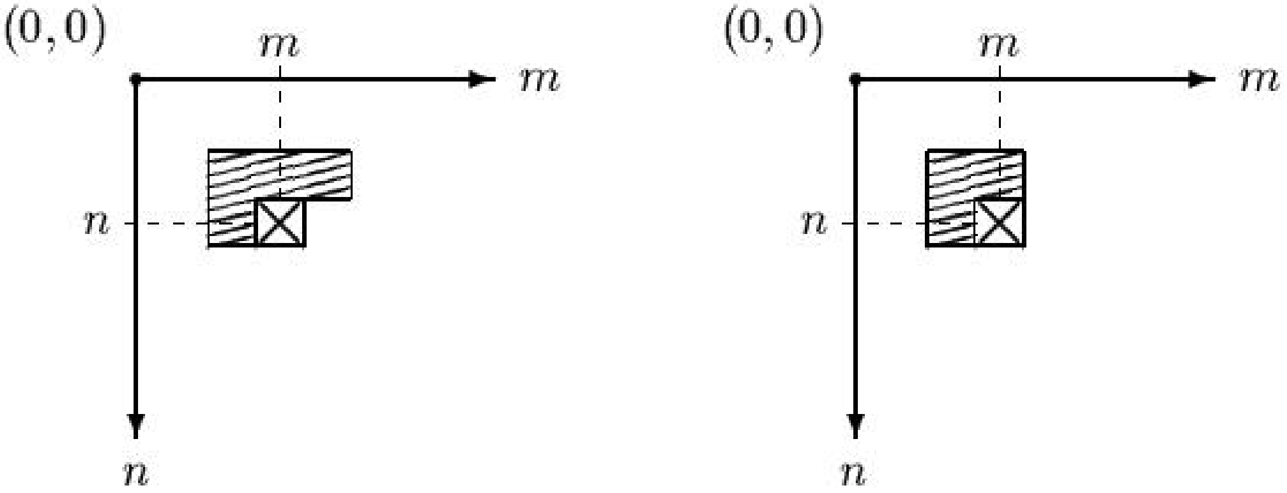


Figure: Causal windows used in image prediction

PREDICTIVE CODING

- Let us suppose that it is sufficient to code the error:

$$e(n,m) = f(n,m) - \hat{f}(n,m)$$

- If $e_q(n,m)$ is the quantized and coded value of the error e(n,m), the pixel value can be reconstructed as follows:

$$f_r(n,m)\ L\ f_r(n-i,m-j),$$

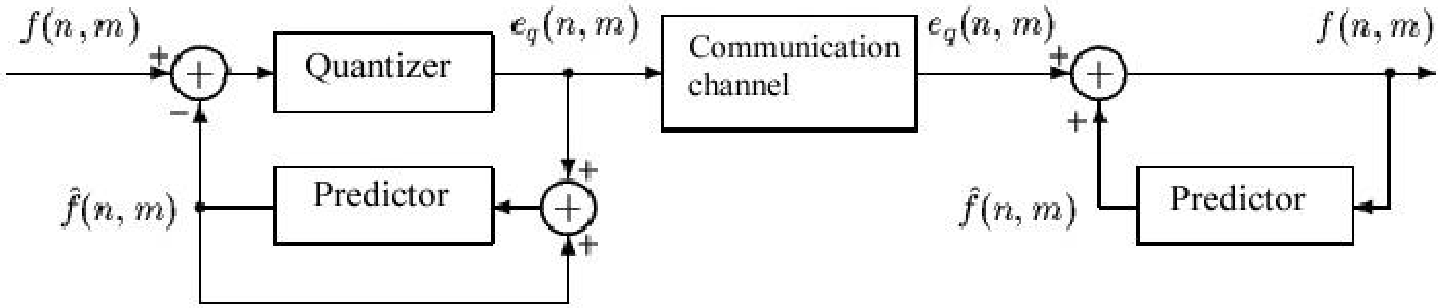
If the prediction is good, the error term has a small dynamic range and

a substantial compression can be achieved.

- For the reconstruction, the transmission of the prediction coefficients and of the coded error is needed.

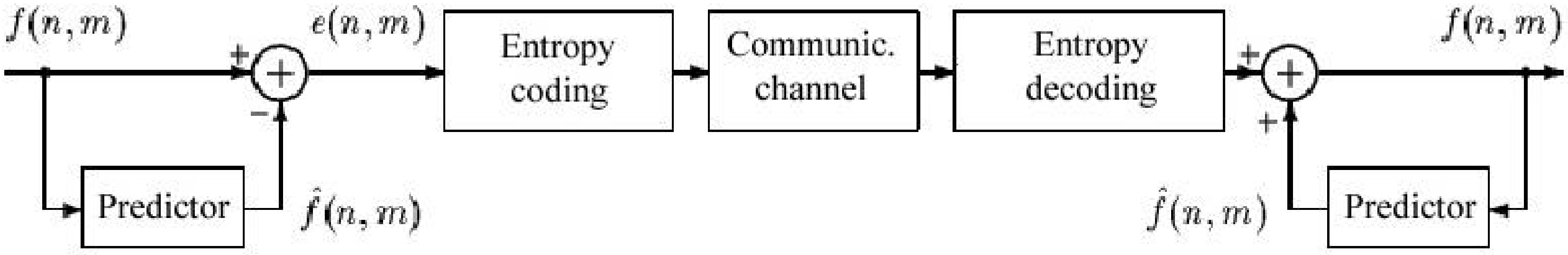
PREDICTIVE CODING

- Extensively used in telecommunications.
- It is a lossy coding scheme.
- The quantization of the error signal always creates an irrecoverable amount of distortion.



Predictive Differential Pulse Code Modulation with entropy coding

- It is a lossless coding scheme.
- The performance of the DPCM depends greatly on the predictor used and on the choice of its coefficients.



PREDICTIVE CODING

Let us suppose that the image line *f*(*m*) can be modeled as a stationary AR process:

$$f(m)=\sum_{k=1}^{p}a(k)f(m-k)+\sigma(m),\qquad E\left[\sigma^{2}(m)\right]=o^{2}$$

where *å*(*m*) is a white additive Gaussian noise uncorrelated to *f*(*m*).

A natural choice for the prediction scheme is:

$$\hat{f}(m)=\sum_{k=1}^{p}a(k)f_{r}(m-k)$$

PREDICTIVE CODING

- The quantized error signal $e_q(m)$ is transmitted to the receiver:

$$e_q(m) = Q[e(m)] = Q[f(m) - \hat{f}(m)]$$

- The image row is reconstructed by using:

$$f_r(m) = \sum_{k=1}^{p} a(k) f_r(m-k) + e_q(m)$$

- The prediction coefficients can be chosen by solving the set of normal equations (where $R(k)$ is the autocorrelation function):

$$\begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & R(0) & \dots & R(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ R(p-1) & R(p-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(p) \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}$$

PREDICTIVE CODING

- The error image can be obtained and quantized, once the coefficients $a(i,j)$ are known:

$$e(n,m) = f(n,m) - \hat{f}(n,m)$$

- The digital image can be reconstructed at the receiver as follows: $f_r(n,m) = \sum\sum_{(i,j)\in A} a(i,j)\, f_r(n-i, m-j) + e_q(n,m)$

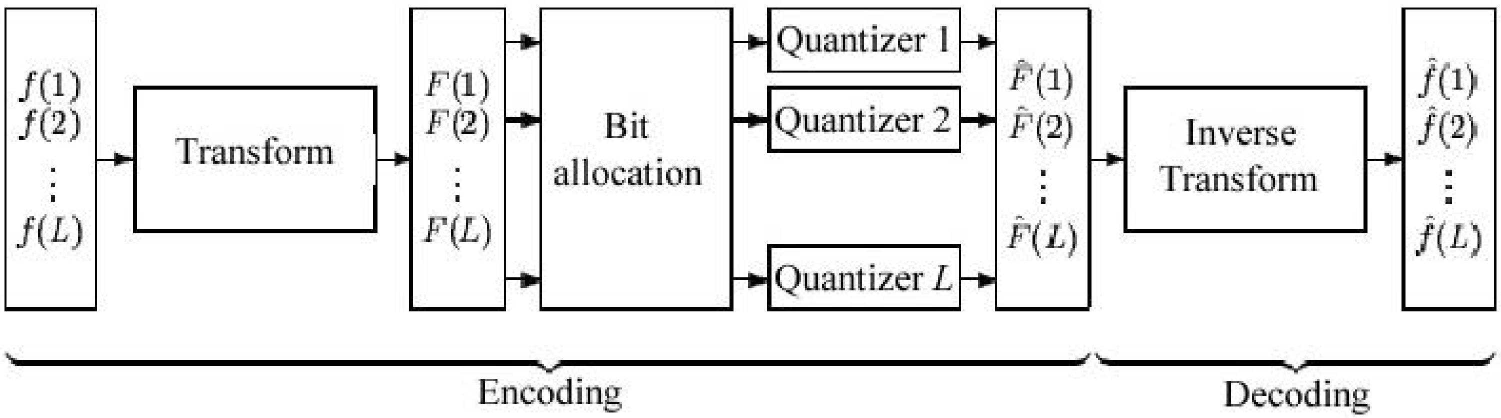
- The autocorrelation coefficients can be estimated by using:

$$R(i,j) = \frac{1}{(2N+1)(2M+1)} \sum_{\substack{i=-N \\ j=-M}}^{N} \sum^{M} f(k,l)\, f(k+i, l+j)$$

TRANSFORM IMAGE CODING

Transform image coding

We try to use image transforms in an effort to concentrate the image energy in a few transform coefficients.



TRANSFORM IMAGE CODING

Let **f** be the vector representing an image of size *L* = *N* x*M.* The transform vector **F** is given by:

$$\mathbf{F} = \mathbf{Af}$$

where **Á** is the transform matrix. The inverse transform is defined as follows:

$$\mathbf{f} = \mathbf{A}^{-1}\mathbf{F}$$

Unitary transform definition:

$$\mathbf{AA}^{*T} = \mathbf{A}^{T}\mathbf{A}^{*} = \mathbf{I}$$

TRANSFORM IMAGE CODING

The DC coefficient and some other “low-frequency” coefficients tend to concentrate most of the signal energy.

A varying number of bits can be allocated to the remaining coefficients.

The transform coefficients $F(k)$, $1<k<K$ are quantized using K quantizers.

The decoding is done by applying the inverse transform to the encoded coefficient vector.

TRANSFORM IMAGE CODING

Problems to be solved for a good transform coding *algorithm*

The choice of the transform to be used (DFT, WHT, DCT, DST etc.).

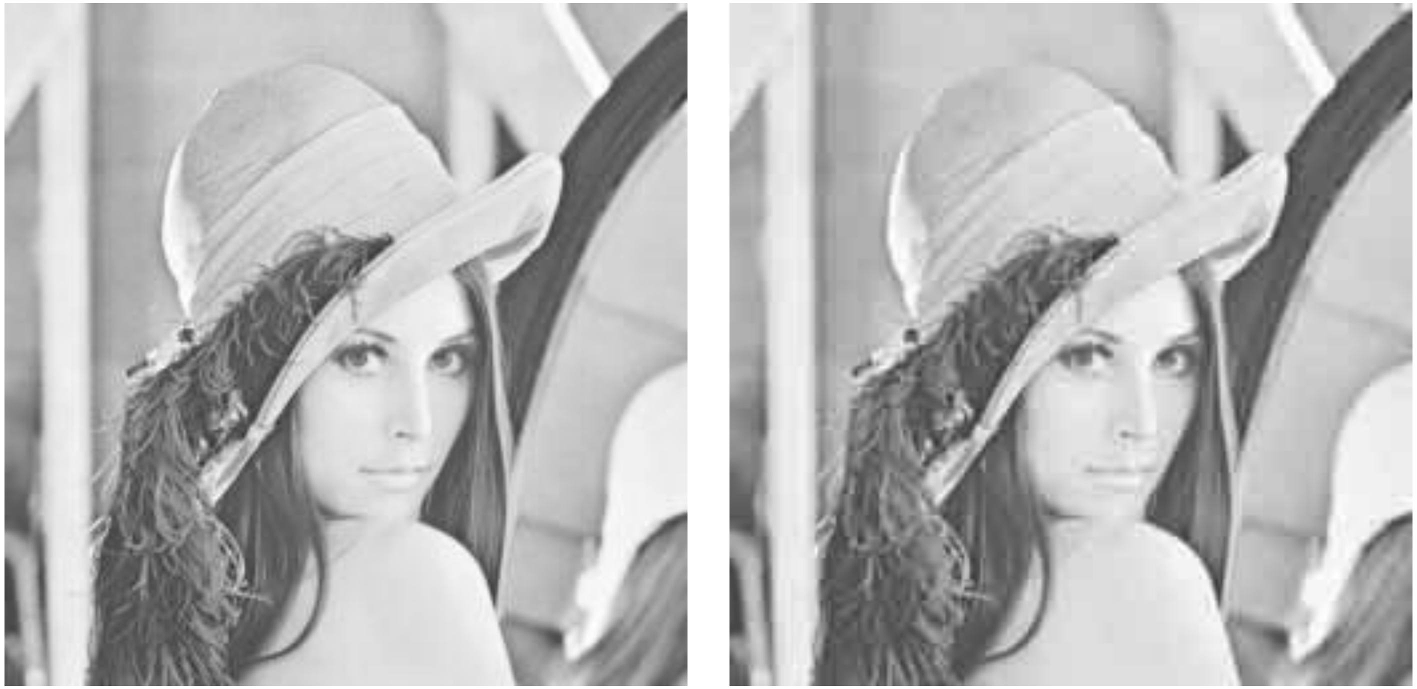
The choice of image block size.
The determination of bit allocation.

TRANSFORM IMAGE CODING

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

Figure : Binary allocation of a 16×16 block cosine transform.

TRANSFORM IMAGE CODING



(a) (b)

Figure : (a) Original image. (b) JPEG compressed image.