Syllabus

DIGITAL IMAGE PROCESSING

[8CS2]

Units	Contents of the subject
I	**Introduction to Image Processing:** Digital Image representation, Sampling & Quantization, Steps in image Processing. Image acquisition, color image representation
II	**Image Transformation & Filtering:** Intensity transform functions, histogram processing, Spatial filtering, Fourier transforms and its properties, frequency domain filters, colour models, Pseudo colouring, colour transforms, Basics of Wavelet Transforms
III	**Image Restoration:** Image degradation and restoration process, Noise Models, Noise Filters, degradation function, Inverse Filtering, Homomorphism Filtering
IV	**Image Compression:** Coding redundancy, Interpixel redundancy, Psychovisual redundancy, Huffman Coding, Arithmetic coding, Lossy compression techniques, JPEG Compression
V	**Image Segmentation & Representation:** Point, Line and Edge Detection, Thresholding, Edge and Boundary linking, Hough transforms, Region Based Segmentation, Boundary representation, Boundary Descriptors, Regional

JPEG compression
Filter numerical
Thresholding
edge detection
Segmentation

Contents

Chapter-1 Introduction to Image Processing ..... 1.1 – 1.46

1.1 Introduction 1.2
1.2 Digital Image 1.2
1.3 Fundamental Components of Image Processing System 1.5
1.3.1 Image Sensors 1.6
1.3.2 Specialized Image Processing Hardware 1.6
1.3.3 Computer 1.6
1.3.4 I.P. Software 1.6
1.3.5 Mass Storage 1.7
1.3.6 Image Displays 1.7
1.3.7 Hard Copy 1.7
1.3.8 Networking 1.7
1.4 Fundamental Steps in Digital Image Processing 1.7
1.5 Elements of Visual Perception 1.8
1.5.1 Image Formation in the Eye 1.8
1.6 Image Sensing and Acquisition 1.9
1.6.1 Image Acquisition Using Single Sensor 1.10
1.6.2 Image Acquisition Using Line Sensor Strip 1.10
1.6.3 Image Acquisition Using Sensor Arrays 1.12
1.7 Image Formation Model 1.13
1.8 Image Sampling and Quantization 1.13
1.8.1 Sampling 1.14
1.9 Representation of Digital Images 1.15
1.10 Spatial and Gray Level Resolution 1.18
1.11 Image Resolution 1.18
1.11.1 Spatial Resolution 1.18
1.11.2 Measuring Spatial Resolution 1.20

1.11.3 Gray Level Resolution 1.21
1.11.4 Bits Per Pixel (BPP) 1.21
1.11.5 Finding BPP from Gray Level Resolution 1.22
1.12 Gray Level Resolution and Quantization 1.23
1.12.1 Spatial Operation 1.23
1.13 Image Registration 1.27
1.14 Vector and Matrix Operations 1.27
1.15 Color Model 1.29
1.16 The RGB Color Model 1.29
1.16.1 Generation of RGB Image 1.31
1.17 The CMY and CMYK Color Models 1.31
1.18 The HSI Color Model 1.32
1.19 Converting Colors from RGB Model to HSI Model 1.33
1.20 Converting Colors from HSI Model to RGB Model 1.34
1.21 Digital Image Representing 1.34
1.21.1 Vector Images 1.35
1.21.2 Coordinate System 1.35
1.21.3 Shapes 1.35
1.21.4 Bitmap Images 1.36
1.22 Sampling 1.37
1.23 Raster Dimensions 1.37
1.23.1 Megapixels 1.38
1.23.2 Scaling / Resampling 1.39
1.23.3 Scaling Image Down 1.39
1.23.4 Scaling Image Up 1.39
1.23.5 Sample Depth 1.39
1.24 Palette/Indexed Images 1.41
1.25 Image Compression 1.42
1.25.1 Lossless Image Compression 1.42
1.25.2 Lossy Image Compression 1.44

1.25.3 Loss through Generations 1.44
1.26 File Formats and Applications 1.45
Review Questions 1.46
Chapter-2 Basic Image Operations ..... 2.1 – 2.38
2.1 Introduction 2.2
2.2 Intensity Transformation Function 2.3
2.2.1 Linear or Image Negative and Identity Transformation . 2.4
2.2.2. Logarithmic [Log and Inverse Log Transformation] 2.4
2.2.3 Power-Law [n^{th} Power and nth Root Transformation or Gamma] 2.5
2.3 Histogram 2.6
2.3.1 Histogram Equalization 2.6
2.3.2 Histogram Matching 2.9
2.3.3 Local Histogram Processing 2.9
2.3.4 Histogram Statistics for Image Enhancement 2.10
2.4 Spatial Filtering 2.11
2.4.1 Mechanism Used is Spatial Filtering 2.11
2.5 Spatial Correlation and Convolution 2.13
2.5.1 Correlation Operation 2.13
2.5.2 Convolution Operation 2.14
2.6 Vector Representation of Linear Filtering 2.15
2.7 Generation of Spatial Filter Masks 2.15
2.7.1 Smoothing Spatial Filters 2.16
2.7.2 Smoothing Linear Spatial Filter 2.16
2.7.3 Smoothening Non-Linear [Order-Statistic] Filters 2.17
2.8 Sharpening Spatial filters 2.18
2.8.1 Basics of Sharpening 2.18
2.8.2 First Order Derivative for Image Sharpening [Non-Linear Filtering] 2.20
2.8.3 Second Order Derivative for Image Sharpening: (The Laplacian Operator) 2.22

2.9 Image Smoothing using Frequency Domain Filters ..... 2.
2.9.1 Ideal Low Pass Filtering ..... 2.
2.9.2 Butterworth Low Pass Filters ..... 2.
2.9.3 Gaussian Low Pass Filters ..... 2.
2.9.4 Image Sharpening using Frequency Domain Filtering ..... 2.
2.10 Ideal Highpass Filters ..... 2.
2.11 Buterworth High Pass Filters ..... 2.
2.12 Gaussian High Pass Filter ..... 2.
2.13 Pseudo Coloring 2.3
2.13.1 Gray Level to Color Transformations 2.3
2.14 Color Transforms 2.3
2.15 Wavelet Transform 2.3
Review Questions 2.3

Chapter-3 Image Restortion ..... 3.1 – 3.3

3.1 Introduction ..... 3.
3.2 Image Degradation/Restoration Model ..... 3.
3.3 Noise Model ..... 3.
3.4 Spatial and Frequency Properties of Noise ..... 3.
3.5 Some Noise Probability Density Function ..... 3.
3.5.1 Gaussian Noise 3.3
3.5.2 Rayleigh Noise 3.4
3.5.3 Erlong (Gamma) Noise 3.5
3.5.4 Exponential Noise 3.6
3.6 Uniform Noise PDF 3.6
3.7 Impulse Noise (Salt-and-Pepper) PDF 3.7
3.8 Periodic Noise 3.8
3.9 Estimation of Noise Parameters 3.8
3.10 Noise Only Spatial Filtering 3.8
3.10.1 Mean Filters 3.9

3.10.2 Order Statistic Filters 3.10
3.10.3 Median Filter 3.10
3.10.4 Maximum and Minimum Filters 3.10
3.10.5 Midpoint Filter 3.11
3.10.6 Alpha-Trimmed Mean Filter 3.11
3.11 Adaptive Filters 3.11
3.11.1 Adaptive Local Noise Reduction Filter 3.11
3.11.2 Adaptive Median Filter 3.12
3.12 Inverse Filtering 3.13
3.13 Minimum Mean Square Error (Wiener) Filtering 3.14
3.14 Homomorphism Filtering 3.16
3.14.1 Homomorphic Filtering Equations 3.18
3.14.2 Implementation of Homomorphic Filtering 3.18
3.14.3 Metrics for Deciding the Performance of Homomorphic Filtering 3.22
3.15 Matlab Source Code 3.27
Review Questions 3.30
Chapter-4 Data Compression and Redundancy ..... 4.1 – 4.18
4.1 Introduction 4.2
4.2 Fundamentals of Visual Data Compression 4.2
4.2.1 Redundancy 4.2
4.2.2 Data Compression 4.5
4.3 Image Compression and Reconstruction 4.5
4.3.1 Image Compression Techniques 4.6
4.4 Measurements for Compression Methods 4.14
4.4.1 Measurements for Lossy Compression Methods 4.14
4.4.2 Measurements for Lossless Compression Methods 4.14
4.5 Quantization 4.15
4.5.1 Quantization in Image Compression 4.15

4.5.2 Measurement of Image Quality 4.15
4.6 JPEG Compression 4.16
Review Question 4.18
Chapter-5 Image Segmentation, Representation and Description 5.1–5.34
5.1 Introduction 5.2
5.2 Edge-Based Segmentation 5.2
5.3 Gradient Operator 5.3
5.4 Edge Linking and Boundary Detection 5.6
5.4.1 Local Processing 5.6
5.4.2 Global Processing 5.6
5.5 Thresholding 5.8
5.5.1 Basic Global Thresholding 5.8
5.5.2 Basic Adaptive Thresholding 5.8
5.5.3 Optimal Global and Adaptive Thresholding 5.9
5.6 Region-Based Segmentation 5.9
5.6.1 Region Growing 5.9
5.7 Use Morphological to Improve Fast Scanning Algorithm 5.12
5.8 Use Geometrical Mathematic to Improve Fast Scanning 5.16
5.8.1 Strategy for Extra Cutting Off 5.16
5.9 Segmentation by Morphological Watersheds 5.18
5.9.1 Dam Construction 5.19
5.9.2 The Use of Markers 5.20
5.10 Representation 5.21
5.10.1 Chain Code 5.21
5.10.2 Polygonal Approximations 5.22
5.11 Signature 5.24
5.12 Boundary Segments 5.24
5.13 Skeletons 5.24
5.14 Descriptor 5.25

5.14.1 Boundary Descriptors 5.25
5.14.2 Regional Descriptors 5.27
5.15 Basic Idea of Region Growing 5.31
5.15.1 A Few Common Approaches 5.31
5.15.2 Region Merging 5.33
Review Questions 5.34
A-1 Appendix A.1–A.24
A-2 Appendix A.1–A.18
E-1 Experiment A.1–A.8

CHAPTER 1

Introduction to Image Processing

Objectives

- Introduction
- Digital Image
- Fundamental Components of Image Processing System
- Fundamental Steps in Digital Image Processing
- Elements of Visual Perception
- Image Sensing and Acquisition
- Image Formation Model
- Image Sampling and Quantization
- Representation of Digital Images
- Spatial and Gray Level Resolution
- Image Resolution
- Gray Level Resolution and Quantization
- Image Registration

1.1 Introduction

An image is a two dimensional function $f(x, y)$, where x and y are plane (spatial) coordinates and its amplitude is called the intensity or gray level of the image.

Figure 1.1 Basic two dimensional image

1.2 Digital Image

When the coordinates of any image, intensity value of the function $f(x, y)$ are all finites, discrete quantities the image is called digital image.

A digital image is composed of a finite number of elements, each having particular location and values. These elements are called picture element, pixels or pets.

Pixel can be defined as an element of digital image.

Digital image processing is a process whose inputs and outputs are images and we can say that it is a process that extract attributes from images up to and including the recognition of individual objects.

The digital image processing includes various processes like acquiring an image, preprocessing that image, extracting the individual characters, segmentations for suitable computer processing, recognization of that individual characters etc.

Origin of Digital Image Processing

Initially, the digital image processing was implemented on newspaper industry. At that time, the first picture was sent by submarine cable between London and New york. In early 1920, the printing equipment

coded picture was transmitted and was reconstructed at the receiving end.

Figure 1.2 A digital picture produced images

Figure 1.2 was received at the receiver end with a halftone pattern. After receiving this picture some problem were observed :

(i) Visual quality was not too much improved.

(ii) Distribution of intensity level was not properly selected.

(iii) Selection of printing procedure.

At the end of 1921, all these problems, were resolved and the quality of picture, intensity level and tonal quality were improved using ***tapes perforated*** method used.

In early 1920s, the Bartlane cable picture transmission system was used to transport a picture across the Altantic within few hours. This system was capable for coding any image, in five distinct levels of gray. Which was further increased upto 15 levels in 1929. So, digital images require storage and computational power. The progress in the field of image processing and supported technologies includes data storage display and transmission.

In 1960s, the computer perform the meaningful and powerful image processing tasks. It was known as the birth of digital image processing and it combines two developments of **John Van Neumann** to bring into focus the potential of digital image processing concepts. After this various improvements were done on images at Jet propulsion laboratory (california). The first image of moon was taken by Ranger 7 on 31 July 1964 and was used for geometric corrections.

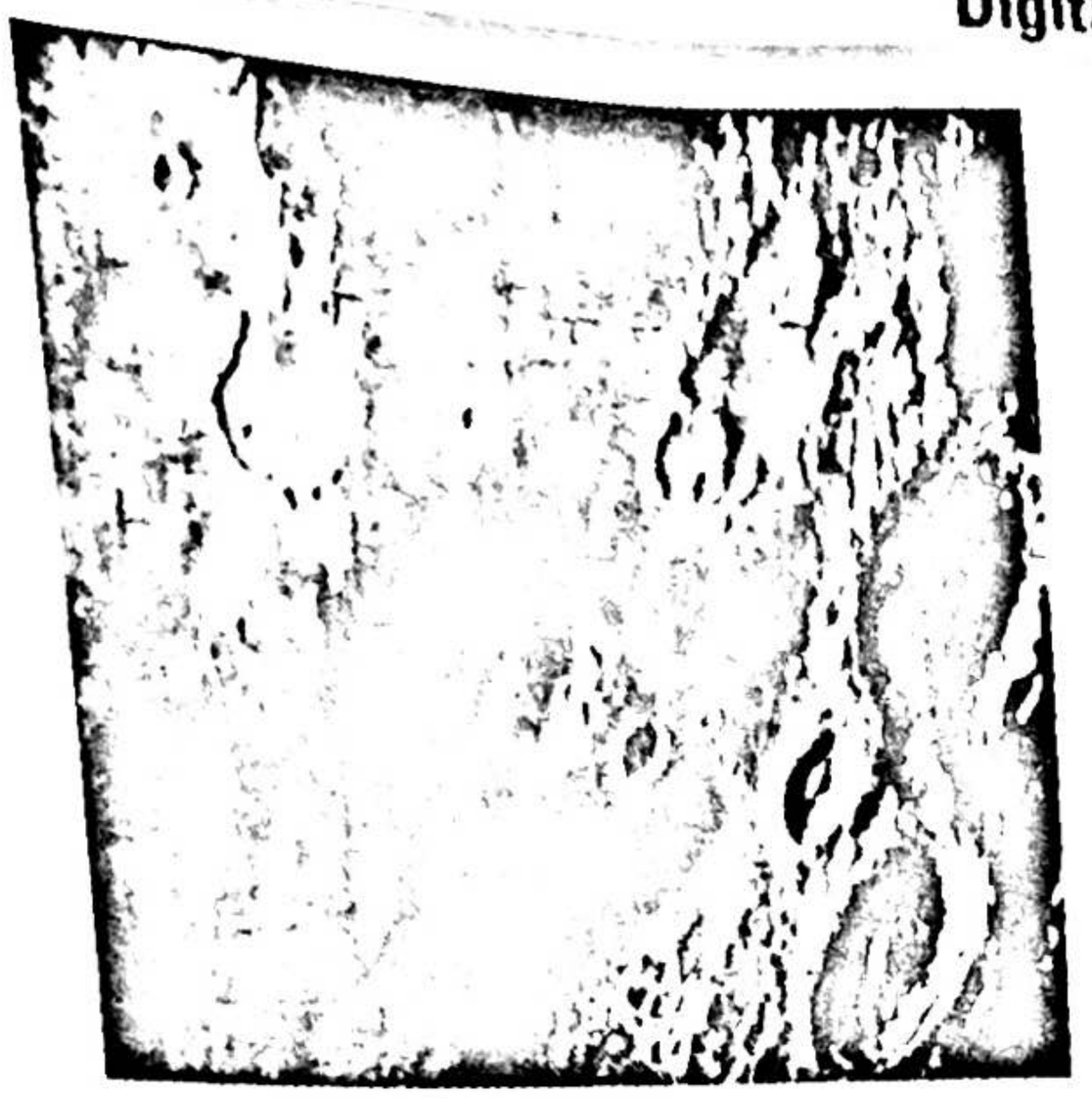


Figure 1.3 The first picture of moon by U.S. spacecraft

Alongwith the space applications, digital image processing is also use in medical imaging, remote earth resources, observations and astronom in 1970s. The Computerized Tomography (CT) also called Computerize Axial Tomography (CAT) is the most important invent in application image processing by Sir **Godfrey N. Hounsfield** and **Prof. Allan M Cormack.** Tomography is a process which consists of an algorithm th uses the sensed data to construct an image which represents a "Slice through object. In this process x-rays passes through the object and a collected at the opposite end by detectors. Besides these application digital image processing has grown vigorously in every fields. Some the major applications of digital image processing are listed below:

(i) Automatic character recognization.

(ii) Industrial machine vision for products assembly and inspection.

(iii) Automatic processing of finger prints.

(iv) Military recognization.

(v) Machine processing of aerial and satellite imagery for weath prediction.

(vi) Screening of x-rays and blood samples.

(vii) Geographer use to study pollution patterns.

(viii) In archeology to restore blurred pictures of rare artifacts.

(ix) In Astronomy, biology, nuctear medicine, law enforcement, defens and industry etc.

Images based on radiation from the EM spectrum are most familia

These EM waves can be considered as propagating sinusoidal waves of varying wavelengths and massless particles. Each particle consists of certain amount of energy and each amount of energy is called a photon.

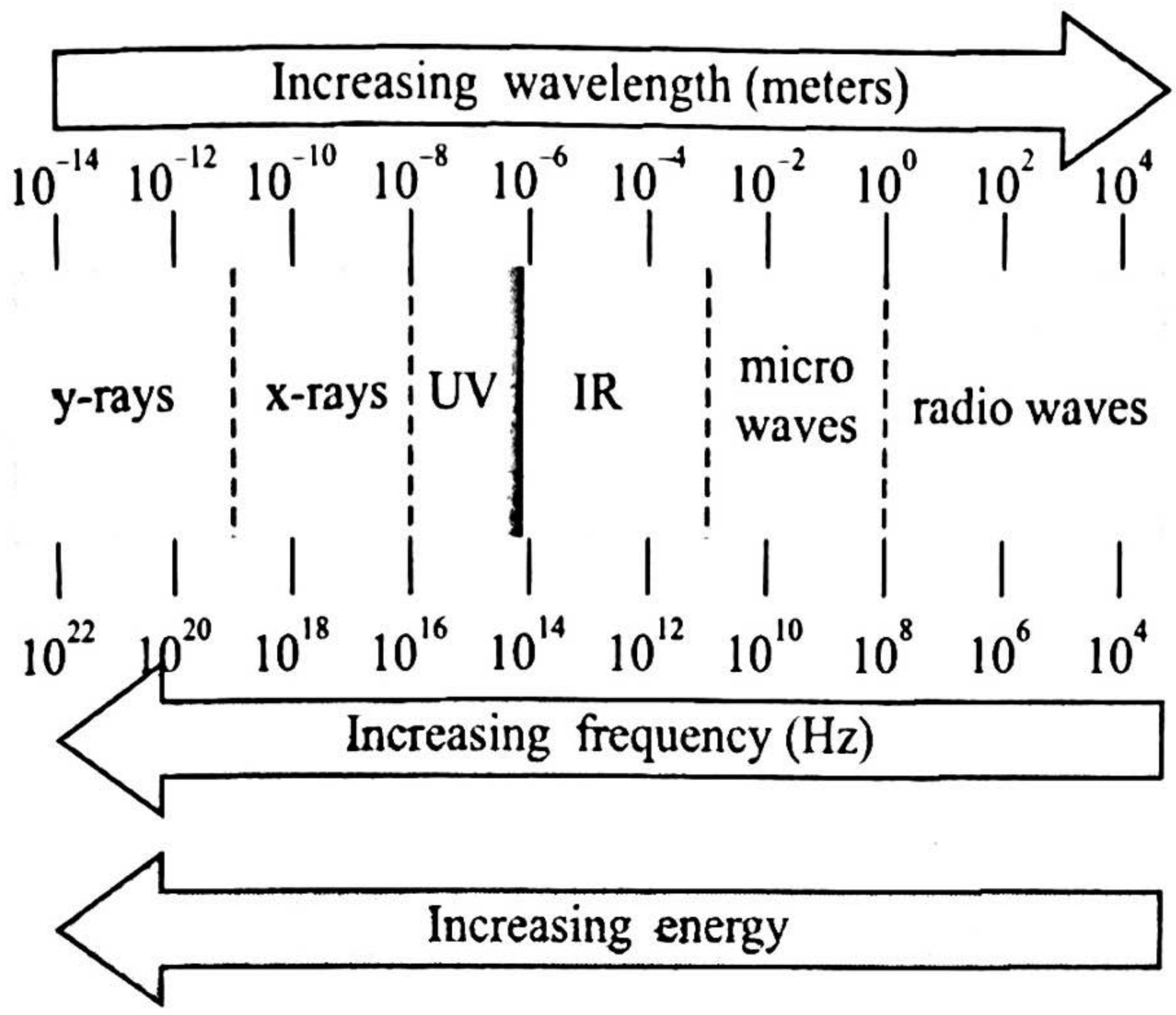


Figure 1.4 EM spectrum

This spectrum is ranging from γ rays having highest energy to radio wave having lowest energy. Depending on this spectrum band there are some fields that uses digital image processing:

(a) X-ray imaging

(b) Gamma ray imaging

(c) Imaging in ultraviolet band

(d) Imaging in visible and IR bands

(e) Imaging in microwave band

(f) Imaging in Radio band

1.3 Fundamental Components of Image Processing System

In the early 1990s it was proposed to design a low cost single board for image processing hardware which is compatible with industry standards and fit into engineering workstation cabinets and personal computers. Figure 1.5 Shows the fundamental components of "General purpose system" for digital image processing.

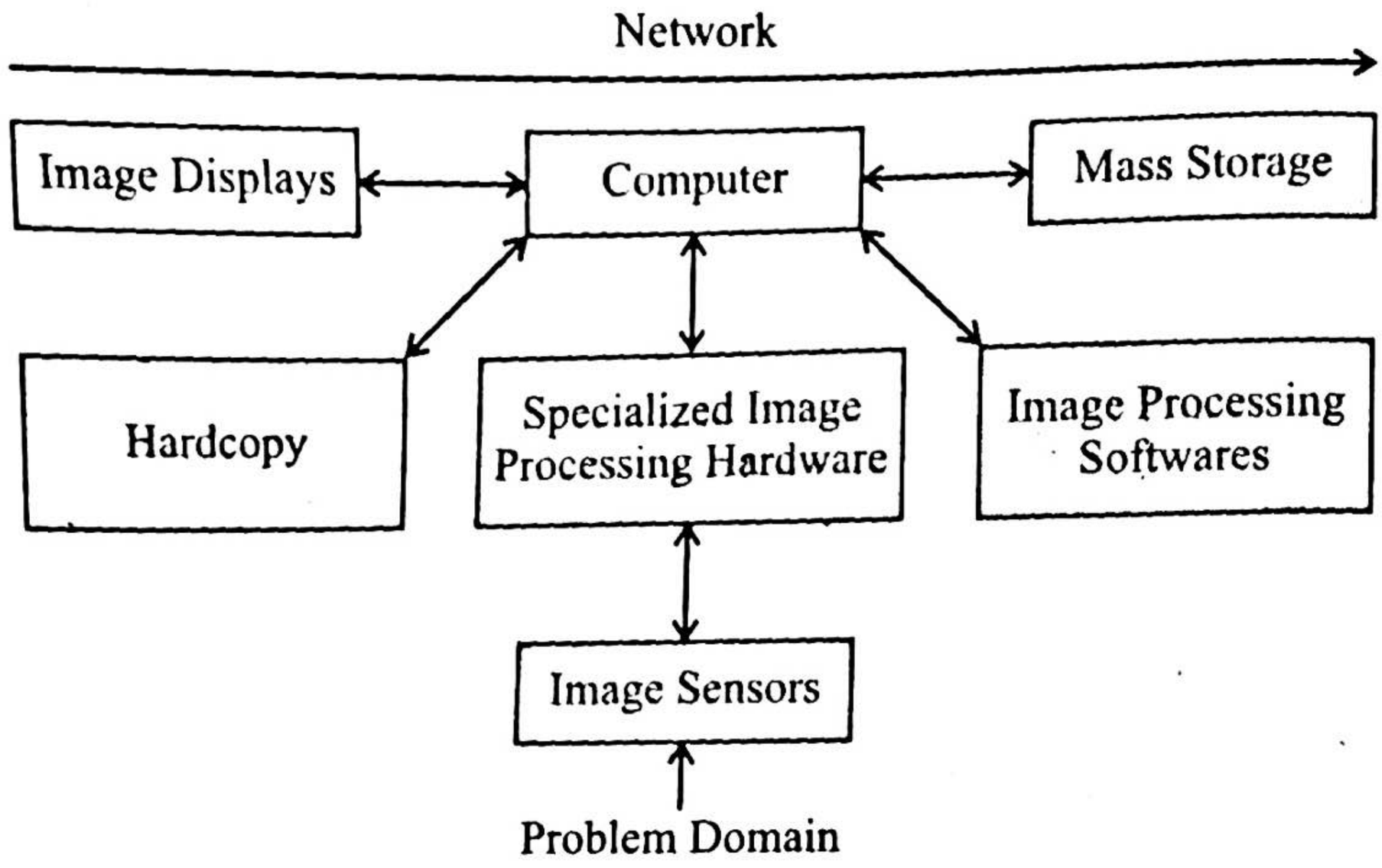


Figure 1.5 Fundamental Components of General Purpose image processing stem

1.3.1 Image Sensors

Two elements are required to sense digital images. First is a physical device which is sensitive to the energy radiated by the object and second is for converting the output of that physical sensing device into digital form is called digitizer.

1.3.2 Specialized Image Processing Hardware

It consists of a hardware which performs some primitive operations like ALU. It is used for noise reductions and are known as front-end-subsystem. This performs all the function at high speed. For example digitation and averaging video images are performed at speed of 30 frames/sec.

1.3.3 Computer

It may be a general purpose computer, PC or supercomputer. Some-time custom computers are used to achieve better performance.

1.3.4 I.P. Software

Image processing softwares are used to perform specific tasks. For this reason a well-designed package is used which have the capability for the user to write correct code.

1.3.5 Mass Storage

To store any image mass storage is required. An image of size 1024 × 1024 pixels, consists of 8 bit each pixel quantity and require 1MB of space.

There are three types of digital mass storage for image processing applications:

(a) **Short term storage** are used for ongoing processings.

(b) **On-line storage** are used for relatively fast recall.

(c) **Archival storage** are used for infrequent access.

1.3.6 Image Displays

Generally, image displays are colour TV monitors, which are driven by the output of image and graphic display card.

If display cards are not available as part of computer system then stereo displays are used which can be implemented in the form of head gear containing two small displays embedded in google worn.

1.3.7 Hard Copy

They are used for recording images. It includes laser printers, heat sensitive devices, film cameras, inkjet units, digital units etc. For presentations, images are displayed on film transparencies if image projection equipment are used.

1.3.8 Networking

Bandwidth is major concern for image transmission. To communicate with remote areas where internet is not so efficiently available, optical fibers and other broadband technologies are widely developed.

1.4 Fundamental Steps in Digital Image Processing

The digital image processing involve some fundamental steps depending on the method use. There are some steps which are categorised on two method. The first method whose input and output are images and the second method whose input may be images but whose outputs are some attributes which are extracted from those images.

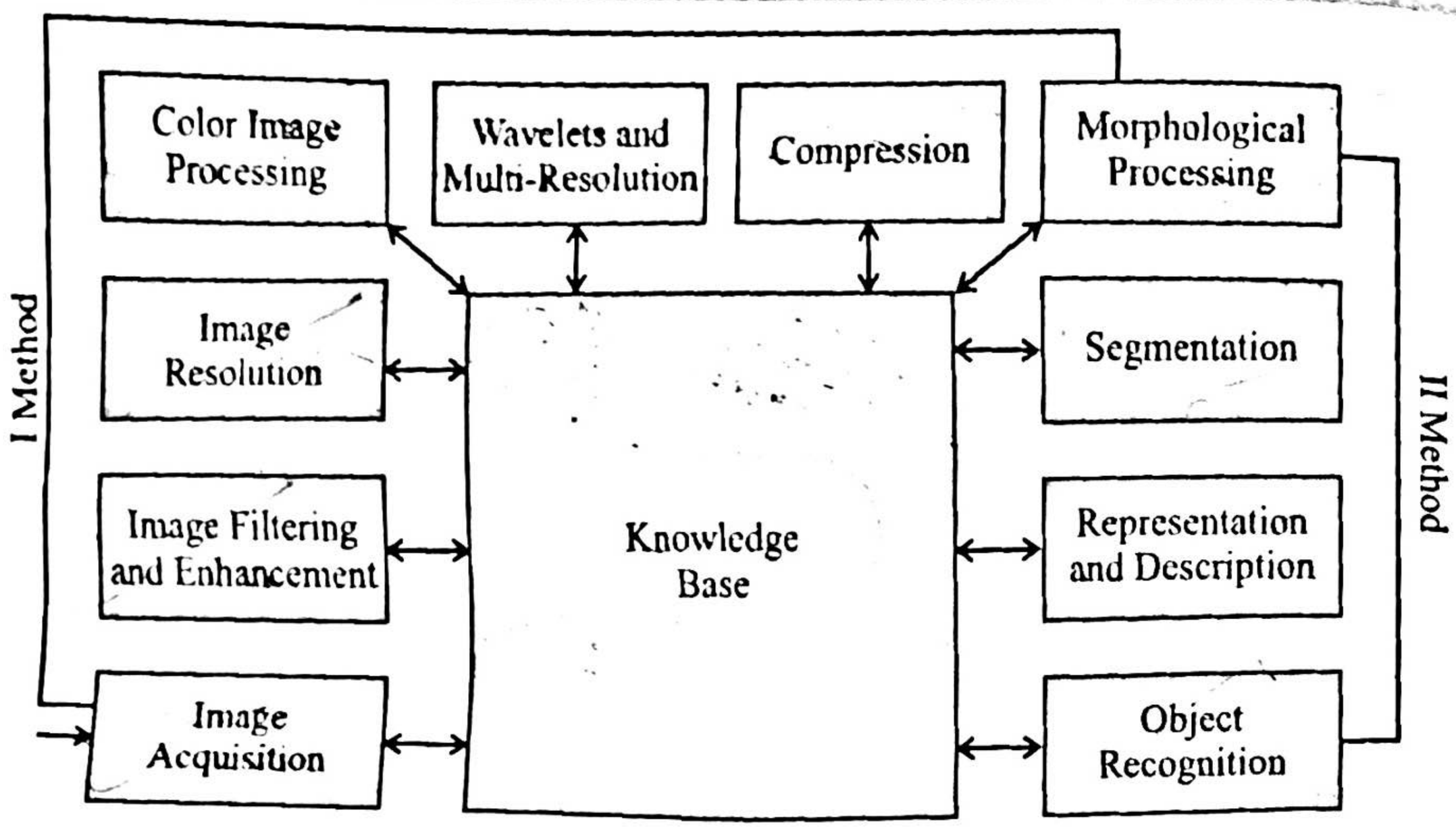


Figure 1.6 Fundamental Steps in digital image processing

All these steps will be discussed in further chapters in this book.

1.5 Elements of Visual Perception

Since the digital image processing is based on mathematical and probabilistic formulations. Besides these human intuition and analysis play an important role in visual judgements. So we have to know how human and electronic imaging devices compare in terms of resolution and ability to adopt to changes in illuminations.

1.5.1 Image Formation in the Eye

In the human eye, the distance between the lens and the imaging region (the retine) is fixed and the focal length needed to achieve proper focus is obtained by varying the shape of the lens.

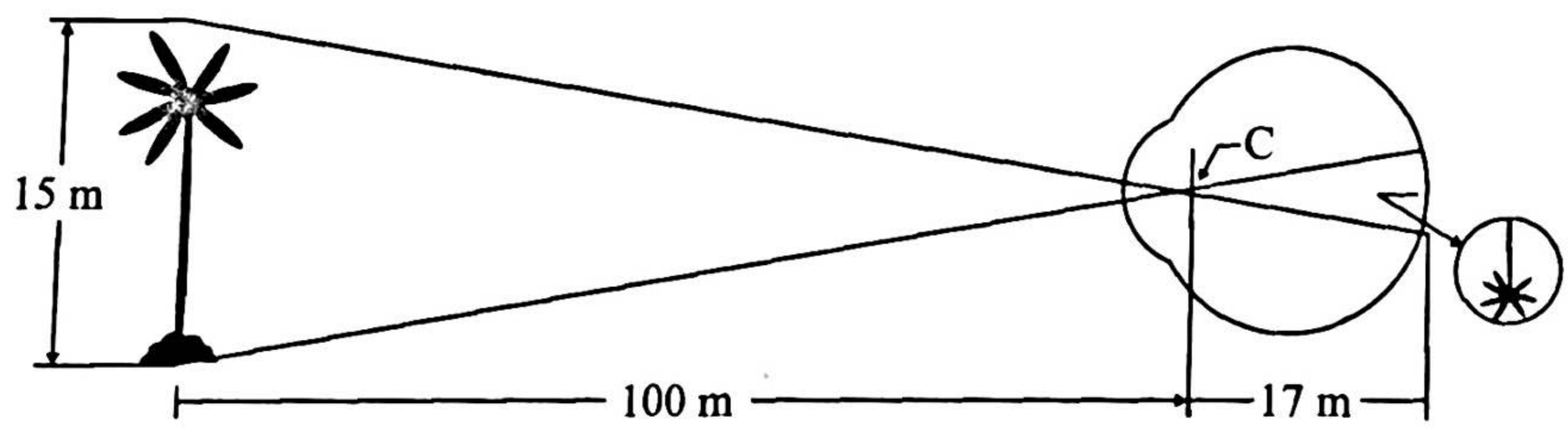


Figure 1.7 Eye working at Palm Tree

The distance between the centre of the lens and the retina along the visual axis is approximately 17 mm. The range of the focal length is 14 mm to 17 mm.

For example suppose a person is working at a tree 15 m high and 100 m distance. Let h is the height of the object in the retinal image then from the geometry.

$$\frac{15}{100} = \frac{h}{17} \qquad \text{...(1.1)}$$

So, $$h = 2.55 \text{ mm} \qquad \text{...(1.2)}$$

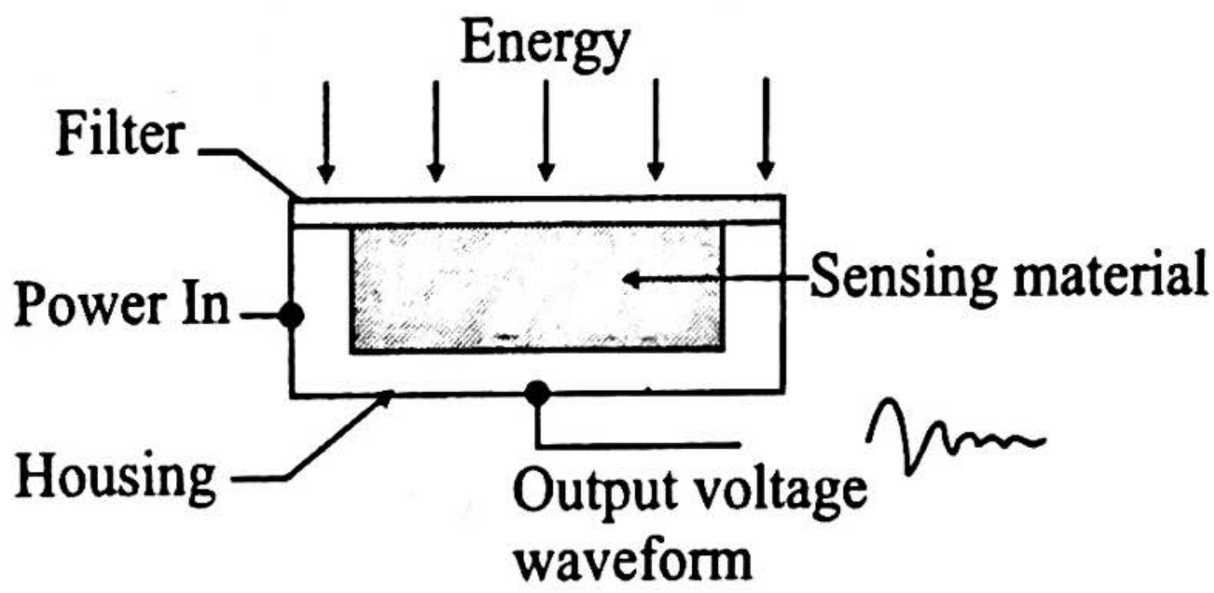
So perception takes place by the relative excitation of light receptors, which are used to transform radiant energy into electrical impulses which are decoded with the use of brain.

1.6 Image Sensing and Acquisition

Since most of the images are generated by the combination of "illumination" and "Scene".

Illumination may originate from a source of electromagnetic energy such as radar, IR or x-ray systems, ultrasound, and computer generated illumination pattern. Similarly, scene are molecules, buried rock formations or a human brain. Illumination energy depends on nature of source through which it become reflected, transmitted or converted.

There are three principal sensors arrangement used to transform energy into digital images. All three use the same idea that "Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material which are detected at output. This output is the response of the sensor used and finally digital image is obtained from each sensor by digitizing its response.



energy
i/p electric pulse
voltage

Figure 1.8 Single imaging sensor

1. Single sensors
2. Line sensors/sensor strips
3. Array sensor

1.6.1 Image Acquisition Using Single Sensor

Image acquisition can be defined as the action of retrieving an image from some source, usually it may be hardware-based source. Performing image acquisition is the first step in image processing workflow sequence.

Photo diode is mostly used sensor in this category for image acquisition. The Photo diode is constructed of Si and its output voltage waveform is proportional to light. The filter used infront of the sensor improves selectivity.

For example, a red filter infront of a light sensor pass the red light stronger than for other components in the visible spectrum.

To generate a 2-D image using single sensor there are some methods among them we have microdensitometers, scanning pattern. In the single sensor image acquisition process there should be relative displacement in both x- and y-direction between the sensor. In this a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension. The single sensor which is mounted on a lead screw provides motion in perpendicular direction.

This is an expensive and slow way to obtain high-resolution images.

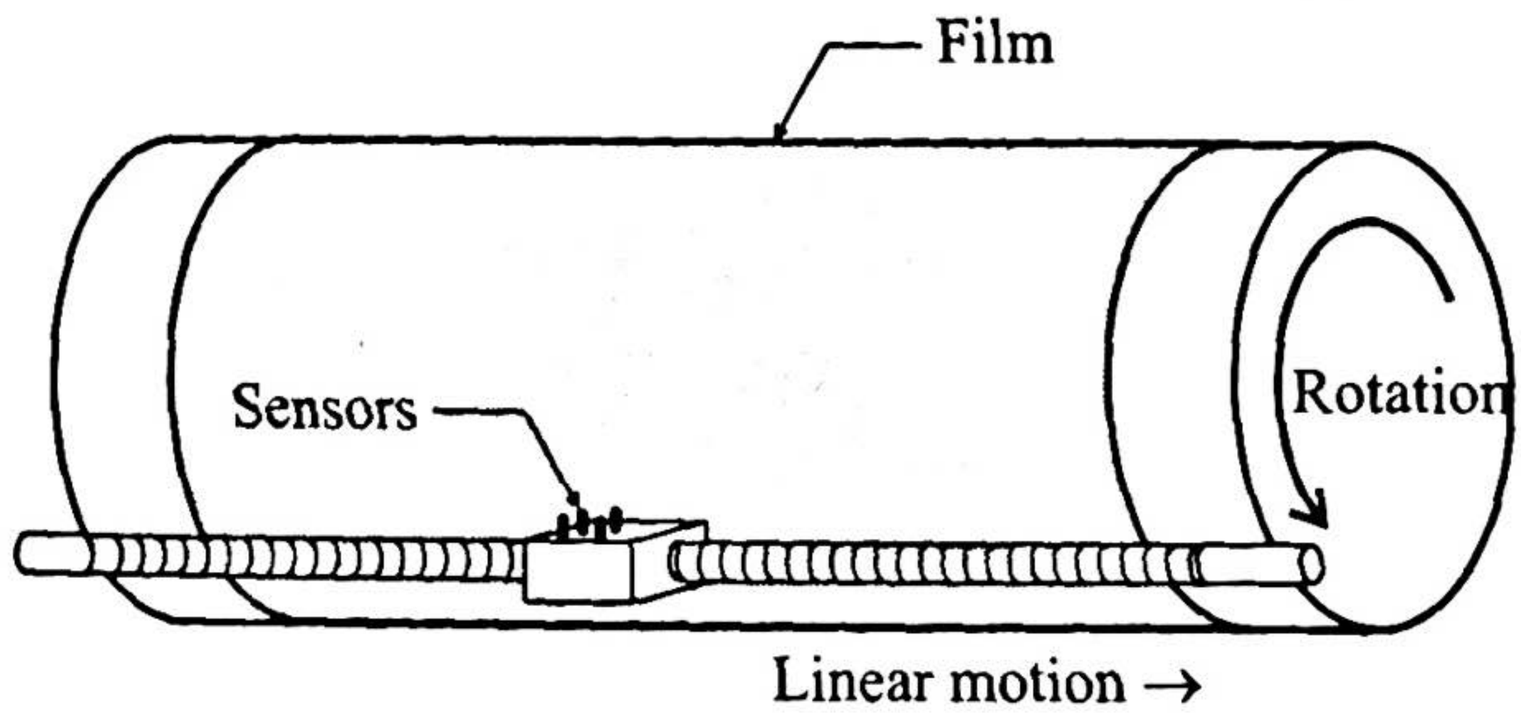


Figure 1.9 Combination of single sensor with motion to generate a 2-D image

1.6.2 Image Acquisition Using Line Sensor Strip

It is the most frequently used geometry. It consists of sensor strip which may be linear or circular. These types of arrangements are basically used in flat bed scanners.

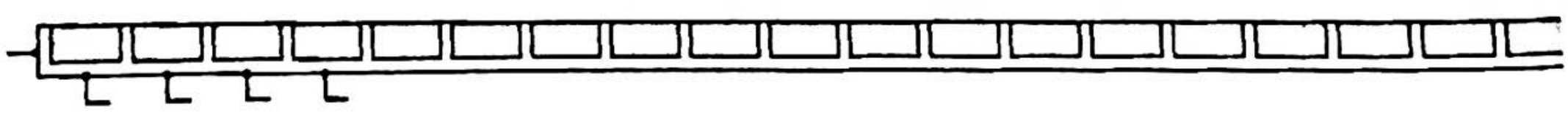


Figure 1.10 Linear Sensor Strip

The in-line sensors are mostly used in airborne imaging applications in which imaging system is mounted on an aircraft which flies at constant altitude and speed. The sensor strips are mounted perpendicular to the direction of flight. This imaging strip gives one line of the image at a time. To project the area to be scanned lenses and other focusing schemes are used on the sensors.

The circular sensor strips are used in medical and industrial imaging to acquire slices/cross-sectional images of 3-D objects.

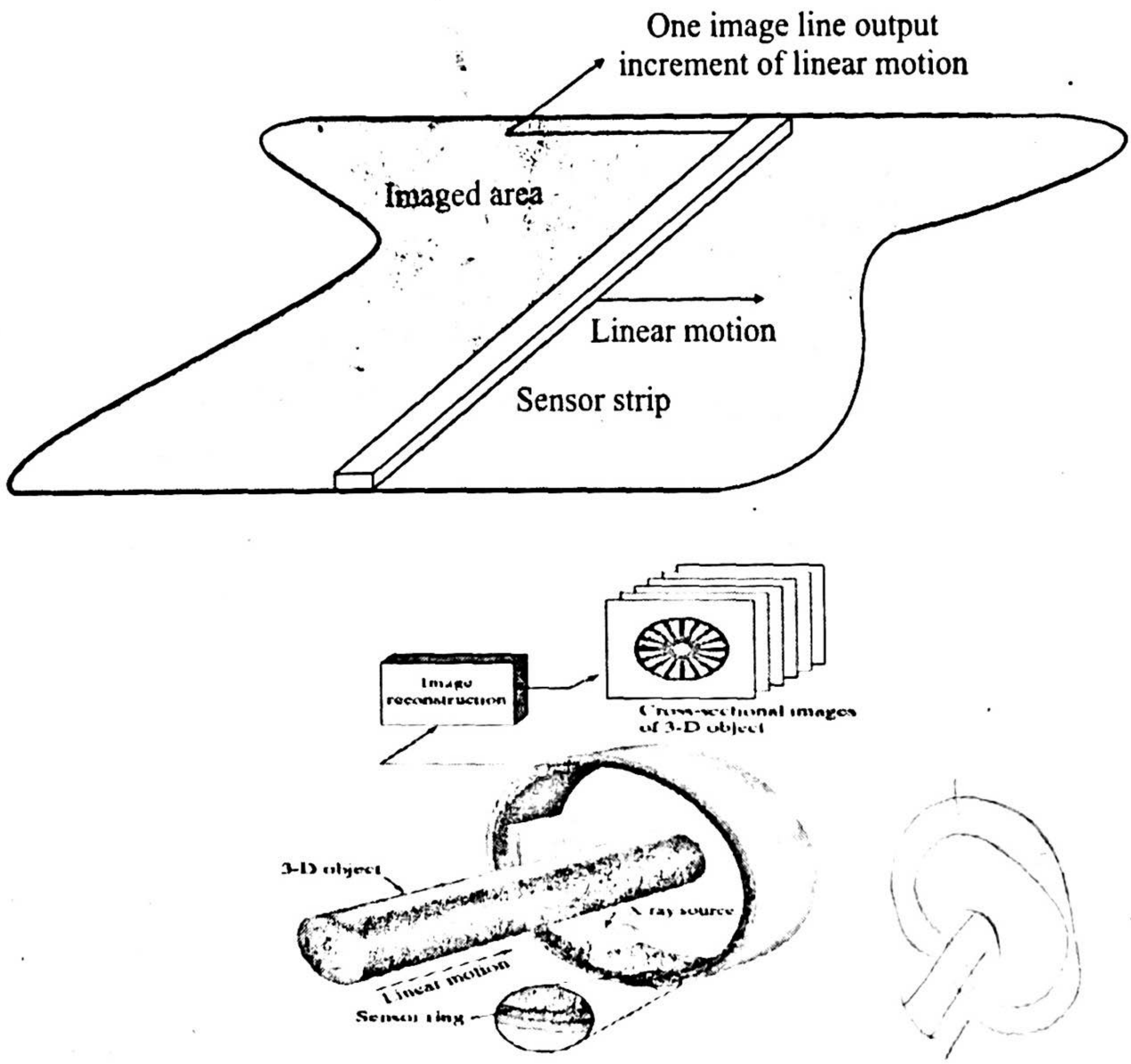


Figure 1.11 Circular Sensor Strip

Figure 1.11 show a image acquisition process using circular sensor strip. In this sensor a rotating x-ray source is used to provides illumination and sensor used at opposite to the source collect the x-ray energy which passes through the object. This is basic principle used in medical and industrial Computerized Axial Tomography (CAT).

1.6.3 Image Acquisition Using Sensor Arrays

In this technique numerous electromagnetic and ultrasonic sensing devices are arranged in an array format figure 1.12 shows this arrangement. This type of arrangements are generally found in digital cameras. CCD sensors are widely used in digital cameras and light sensing instruments.

In such type of sensors the response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor. It help in noise reduction.

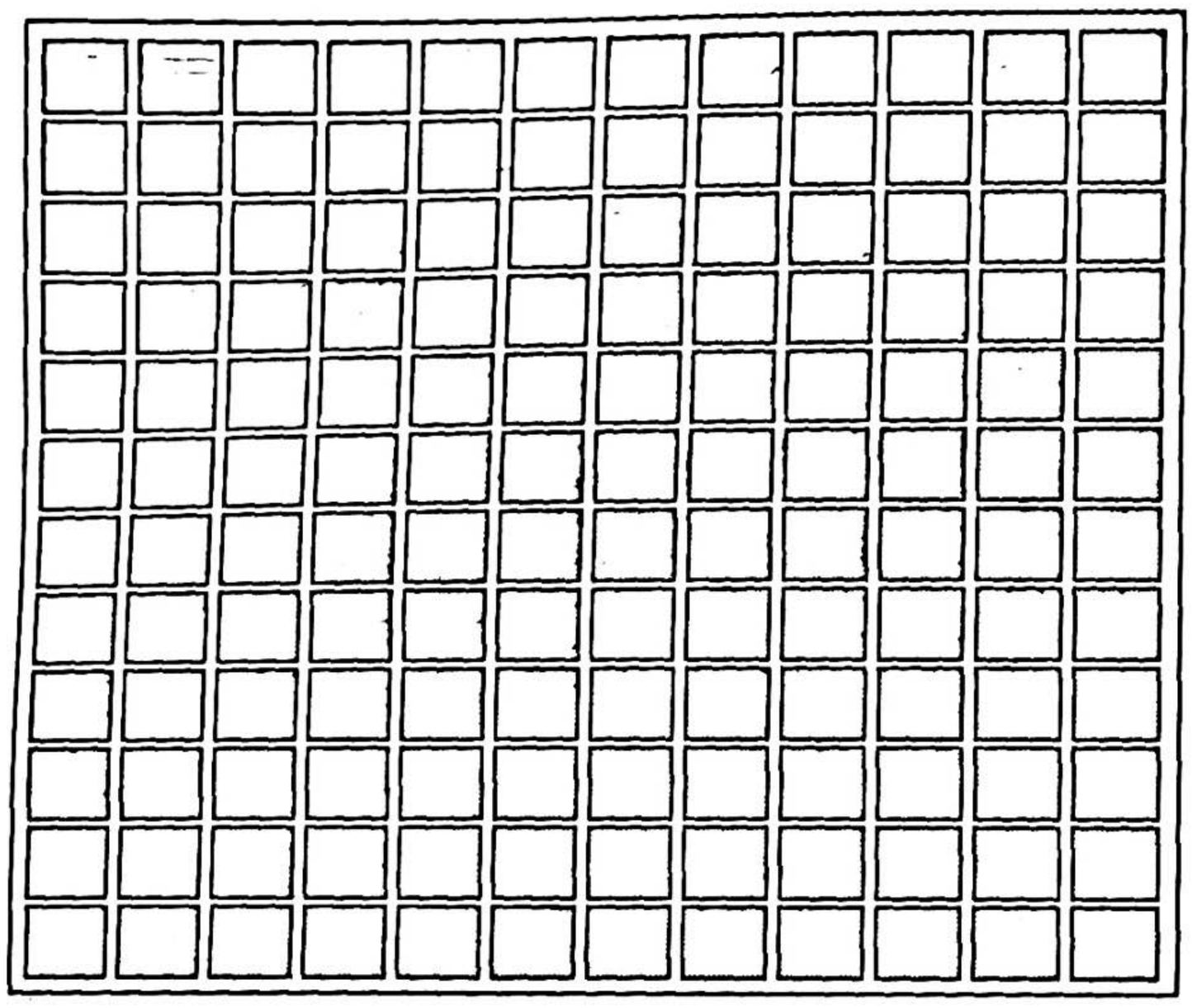


Figure 1.12 Array sensor

Figure 1.12 is an array sensor. In this sensor a complete image can be obtained by focusing the energy pattern onto the surface of the array.

The principle of image acquisition process is explained in figure 1.13. It shows that energy from an illumination source is reflected from a scene element. The imaging system is use to collect the incoming energy and focus it onto the lens image plane. If the illumination is weak then front end of the imaging system is an optical lens and the project viewed scene onto the lens focal plane.

The used sensor array is coincident with the focal plane and produces output which is proportional to the integral of the light received at each sensor.

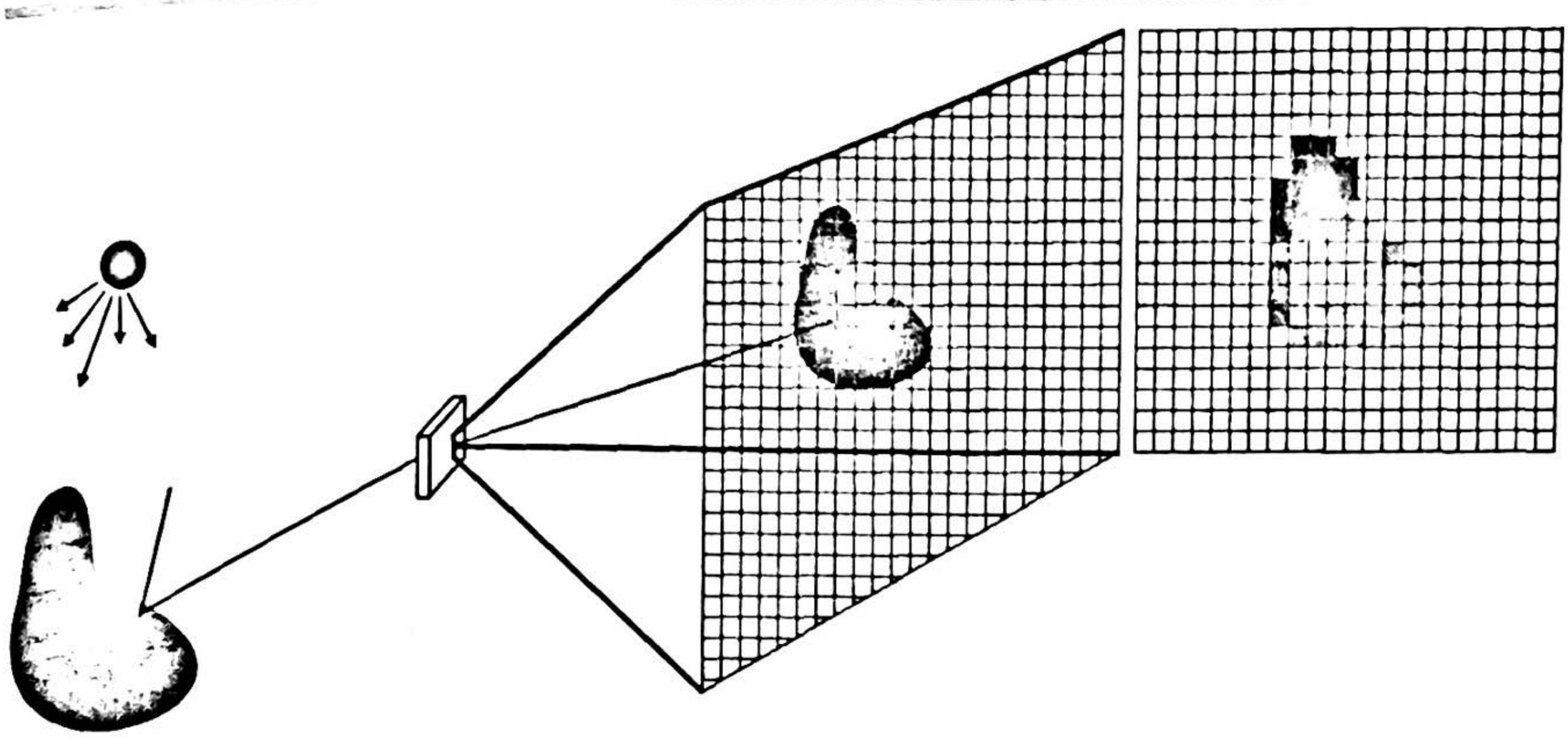


Figure 1.13 Digital Image Acquisition Process

1.7 Image Formation Model

Since an image is a two-dimensional function $f(x, y)$ having an amplitude as positive scalar quantity at spatial coordinates (x,y). The intensity value of this image is proportional to energy radiated by a physical source. So $f(x, y)$ must be non-zero and finite.

i.e., $$0 < f(x, y) < \infty \quad \ldots(1.3)$$

This function $f(x, y)$ have two components

(i) The amount of source illumination incident on the scene and

(ii) The amount of illumination reflected by the object in the scene.

Both of these are called illumination $i(x, y)$ and reflectance $r(x, y)$ components. Both these components are combined to form $f(x, y)$.

i.e., $$f(x, y) = i(x, y)\, r(x, y) \quad \ldots(1.4)$$

where $$0 < i(x, y) < \infty \quad \ldots(1.5)$$

and $$0 < r(x, y) < 1 \quad \ldots(1.6)$$

Nature of $i(x, y)$ is determined by the illumination source and $r(x, y)$ is determined by characteristics of imaged objects.

1.8 Image Sampling and Quantization

Since our main aim is to generate digital image from sensed data. Therefore we need to convert the continuous sensed data into digital form. It involves two processing sampling and quantization.

1.8.1 Sampling

Continuous functions have to be converted into a sequence of discrete values before they can be processed. An image may be continuous with respect to the x-and y-coordinates and in amplitude. To obtain digital form of any image we have to perform sampling in both coordinates and in amplitude.

So digitizing the coordinate values is called sampling and digitizing the amplitude values is called quantization.

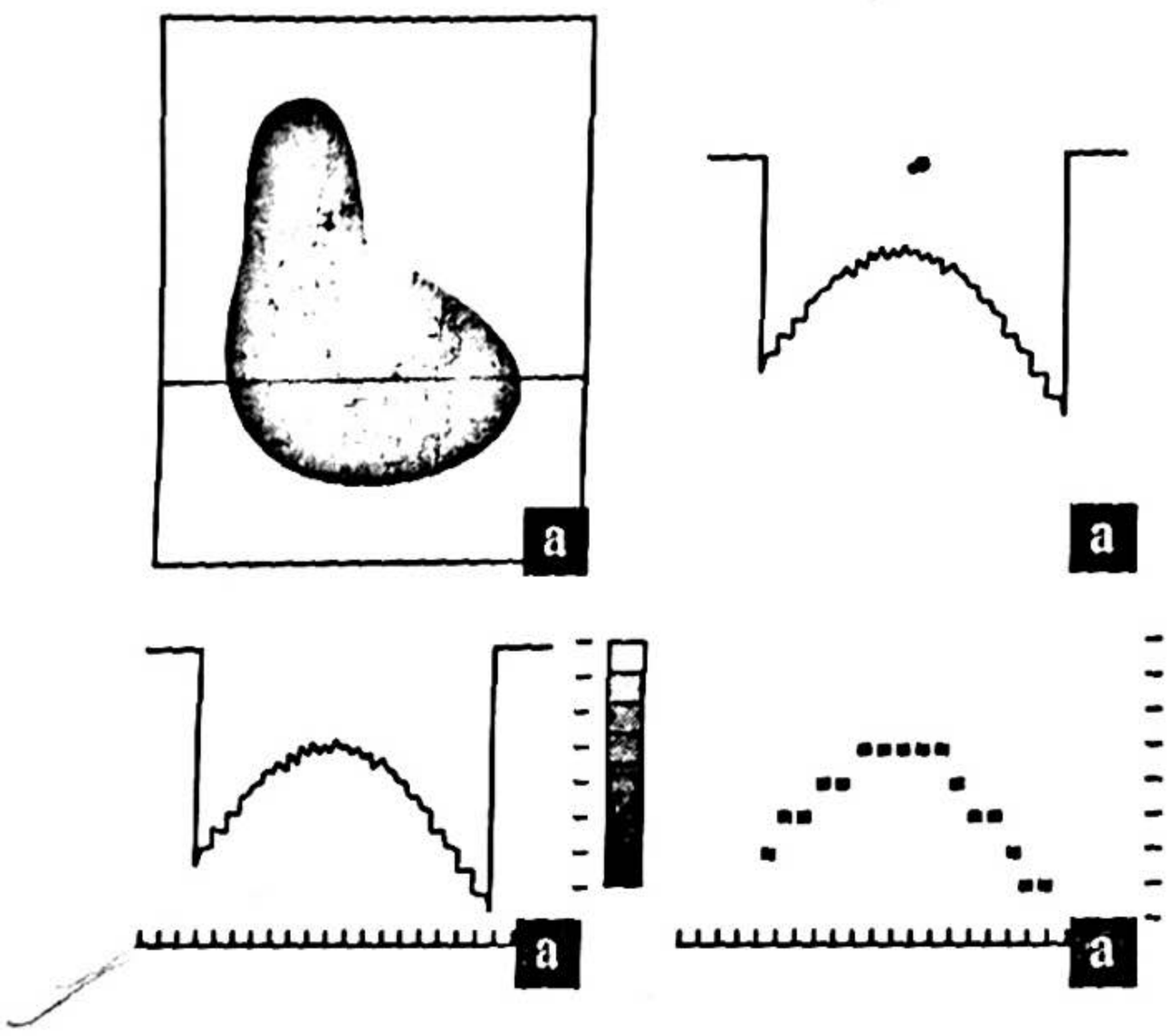


Figure 1.14 Generation of digital image

The samples obtained are very small white squares superimposed on the function and the set of these discrete locations gives the sampled function. To obtain a digital function the intensity values must be converted or quantized into discrete quantities. This intensity is moving from black to white through eight discrete intervals.

The continuous intensity levels are quantized by assigning one of the eight values to each sample. The digital samples obtained from both sampling and quantization are shown in figure 1.14.

The method of sampling is determined by the sensor arrangement which is used to generate the images. First of all image is generated by a single sensor in combined mechanical motion then its output is quantized to complete the process of generating a digital image.

With the use of sensing strip in image acquisition, there are some sampling limitations in one image direction. Mechanical motion can be

controlled more accurately but it makes little sense to achieve sampling density in one direction.

Apart from those when a sensing array is used for image acquisition, then there is no motion and the number of sensors limits the sampling in both directions.

Figure 1.15 shows a sampled and quantized image. It shows that the quality of a digital image is determined to a large degree by the number of samples and discrete intensity levels used in sampling and quantization.

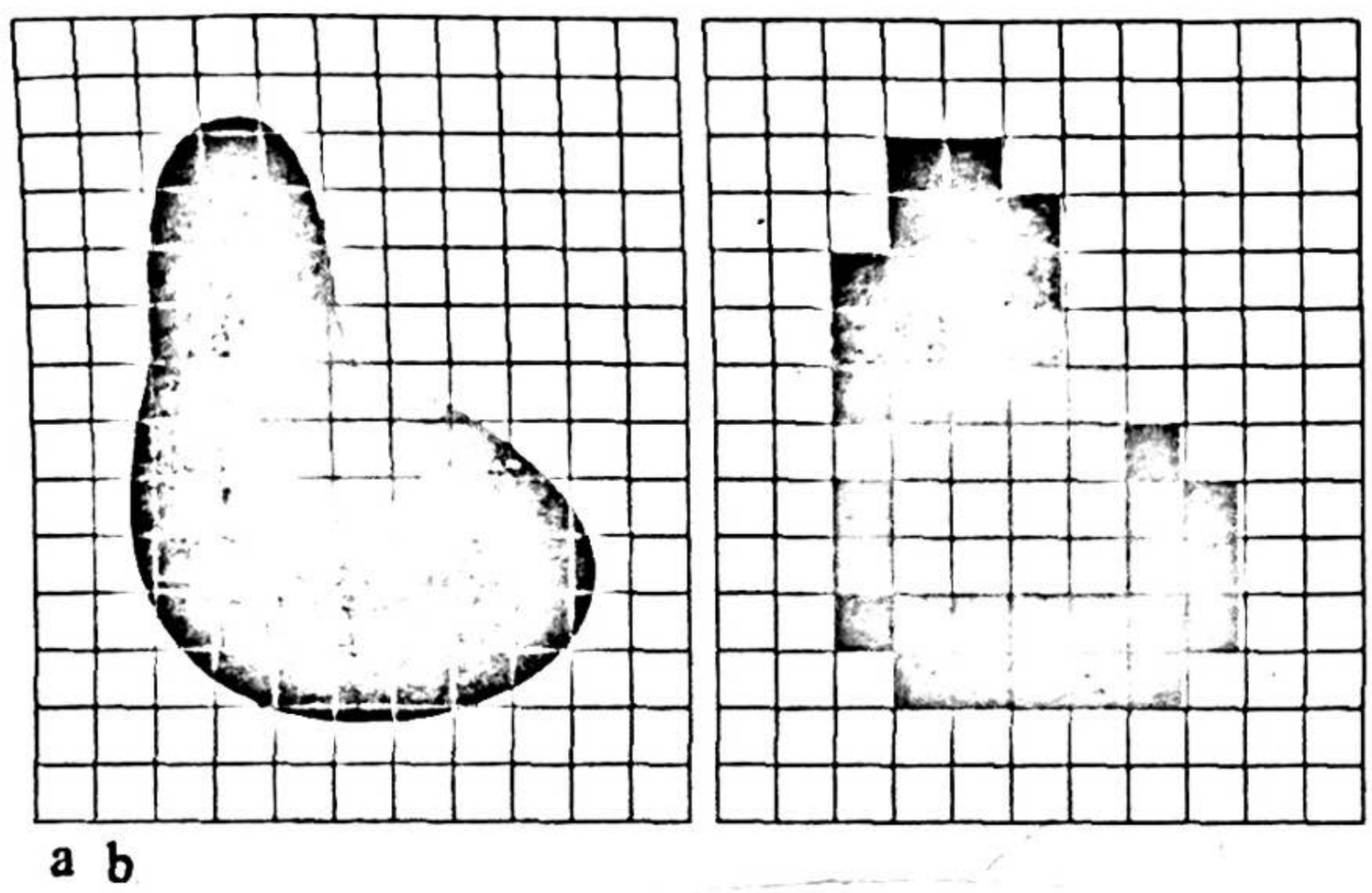


Figure 1.15 Sampled and quantized image

1.9 Representation of Digital Images

The value of the image at any coordinates (x, y) is denoted as $f(x, y)$. The section of real plane spanned by the coordinates of an image is called spatial domain having x and y as spatial variables or spatial coordinates.

Image may be virtual and spatial.

Any function $f(x, y)$ can be represented by three basic ways:

(i) *In this two axis determine spatial location and third axis gives the values of f as a function of two variables i.e. value of intensities.*

(ii) *In this we can assume it as it would appear on a monitor. In this method the intensity of each point is proportional to the value of f at that point.*

For example in the given figure 1.18. If the intensity is normalized to the interval [0, 1], then each point in the images has the value 0, 0.5, or 1. The monitor converts all these value to black, gray and white respectively.

(iii) In this representation, we can display the numerical values of $f(x, y)$ as an array matrix.

For example, let f is having 600×600 elements

i.e. 360,000 numbers

so, printing this array gives little information but when we develop the algorithm it would be quite useful when only part of image is printed.

From this discussions it is concluded that numerical arrays are used for processing and algorithm development.

In matrix for the representation of an $M \times N$ numerical array as

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,1) & \cdots & f(1,N-1) \\ f(2,0) & f(2,1) & f(2,1) & \cdots & f(2,N-1) \\ \vdots & & & \cdots & \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \cdots & f(M-1,N-1) \end{bmatrix} \quad \ldots(1.7)$$

It is an equivalent way to represent a digital image. All the elements of this matrix is called an image element, picture element, pixel or pel.

In general we can use traditional notations to represents digital images with its elements as follows:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,N-1} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,N-1} \\ \vdots & & & & \\ a_{M-1,0} & a_{M-1,1} & a_{M-1,2} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad \ldots(1.8)$$

The origin of digital image is at the top left, from the positive x-axis extending downward and positive y-axis extending to right. This concept was derived from the fact that many image display. Sweep an image starting at the top left and moving to the right one row at a time.

The sampling process may be seen as partitioning the x-y plane into an array or grid, having the coordinates of the center of each cell in the grid being a pair of elements of a Cartesian product z^2. This cartesian

product z^2 is a set of all ordered pairs of elements (z_i, z_j) having z_i and z_j integers from z. In this way $f(x, y)$ is a digital image if $(x, y$ are integers from z^2 and f is a function which gives an intensity value to each distinct pair of coordinates (x, y). This functional assignment is called quantization.

If the intensity levels are integers, function 'z' replaces 'R' where 'R' is set of real numbers then a digital image becomes a 2-D function having coordinates and amplitudes are integers.

The number of intensity levels is an integer power of 2

i.e. $$L = 2^K \qquad ...(1.9)$$

where L is the intensity level or intensity scale and K is an integer

Sometime the integer range $[0, L - 1]$ is spanned by the grey scale which is used in different fields. This range is called dynamic range of an imaging system. It is defined as the ratio of the maximum measurable intensity to the minimum detectable intensity level in the system. Here, the upper limit is called saturation and the lower limit is called noise, depending on this concept an image contrast is defined. It is given as the difference in intensity between highest and lowest intensity levels in an image.

For example if an image have appreciable number of pixels with high dynamic range then we can say the image have high contrast whereas an image with low dynamic range has a dull, washed out gray look.

If 'b' is the bits required to store a digitized image then

$$b = M \times N \times K \qquad ...(1.10)$$

If $M = N$ then

$$b = M^2K \qquad ...(1.11)$$

So to store any square images with various values of M / N and K. We have a table representing number of intensity level (L) corresponding to each value of K.

Table 1.1

M/K	1(L=2)	2(L=4)	3(L=8)	4(L=16)	5(L=32)	6(L=64)	7(L=128)	8(L=256)
32	1,024	2,048	3,072	4,096	5,120	6144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,576	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	372,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

If an image have 2^K intensity levels then is called 'k'-bit image. Similarly an image with 256 discrete intensity values is called an "8-bit image".

It is clear from the table that storage requirement for an 8 bit image having size 1024 × 1024 and higher are not insignificant.

1.10 Spatial and Gray Level Resolution

Spatial resolution is the measure of the smallest recognized (discernible) detail in an image or it can be stated as lines per unit distance, dots per unit distance an image size having 1024 × 1024 pixels is not a meaningful statement without stating the spatial dimensions. Intensity resolution also refers to the smallest discernible charge in intensity level. In intensity resolution the number of bits used to quantized intensity is important.

For example, we can say that if an image is quantized into 256 levels has 8 bits of intensity resolution.

1.11 Image Resolution

Image resolution can be defined in many ways. One type of it which is pixel resolution that has pixel resolution and aspect ratio. Here, we are going to define another type of resolution which is spatial and gray level resolution.

1.11.1 Spatial Resolution

Spatial resolution states that "The clarity of an image cannot be determined by the pixel resolution it does not matter that whatever be the number of pixels in an image".

Spatial resolution can be defined as the smallest discernible (recognizable) detail in an image or we can define spatial resolution as the number of independent pixels values per inch.

In short what spatial resolution refers to is that we cannot compare two different types of images to see that which one is clear or which one is not. If we have to compare the two images, to see which one is clearer or which has more spatial resolution, we have to compare two images of the same size.

For example:

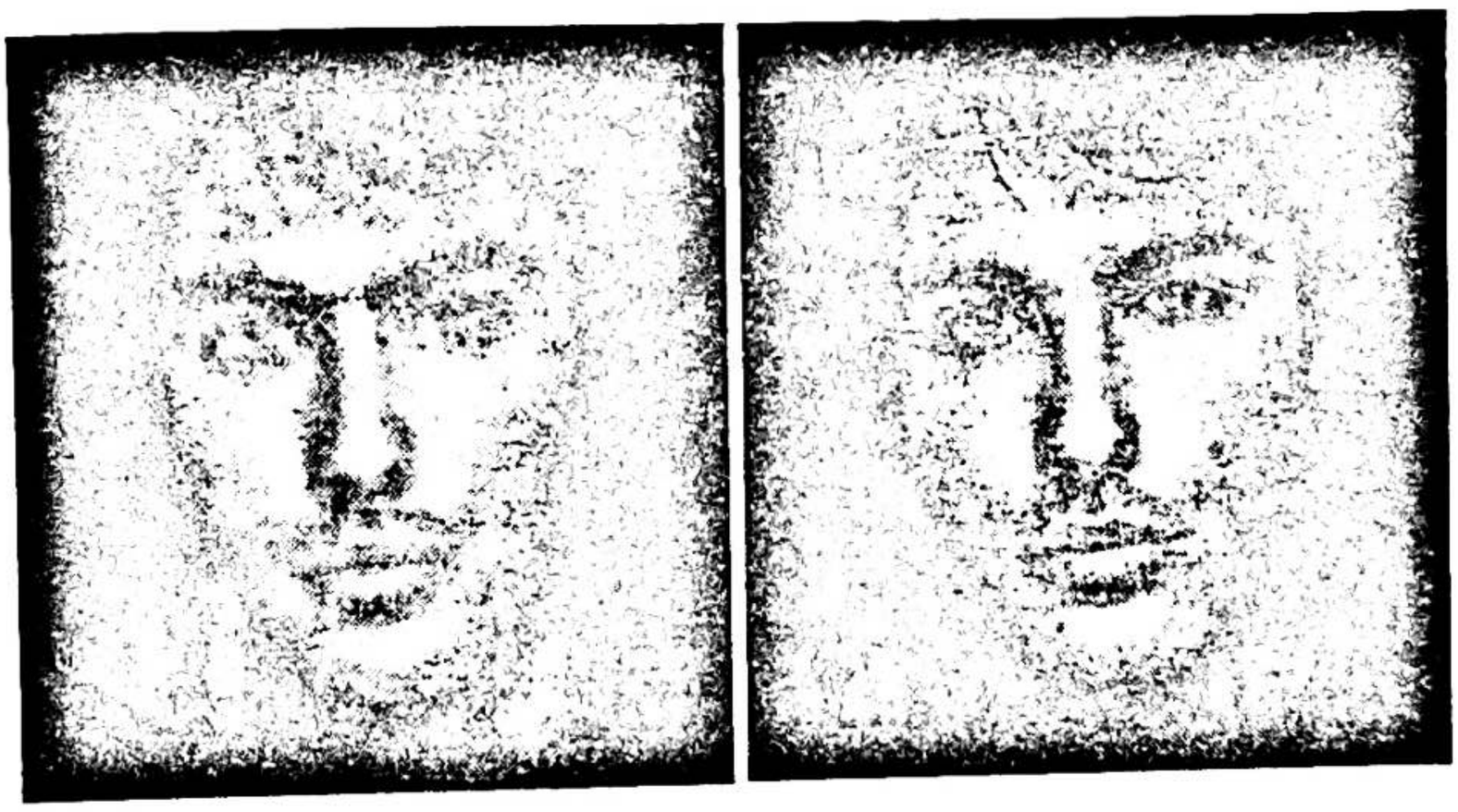


Figure 1.16 Spatial resolution

Image at left has a higher pixel count than the one to the right, but is still of worse spatial resolution.

We cannot compare these two images to see the clarity of the image.

Although both images are of the same person, but that is not the condition we are judging on. The picture on the left is zoomed out picture with dimensions of 227 × 222. Whereas the picture on the right side has the dimensions if 980 × 749 and also it is a zoomed image. We cannot compare them to see that which one is clearer. Remember, the factor of zoom does not matter in this condition, the only thing that matters is that these two pictures are not equal.

So in order to measure spatial resolution, the pictures below would serve the purpose. Now we can compare these two pictures. Both the pictures have same dimensions which are of 227 × 222. Now when we compare them, we will see that the picture on the left side has more spatial resolution or it is more clear than the picture on the right side. That is because the picture on the right is a blurred image.

1.11.2 Measuring Spatial Resolution

Since the spatial resolution refers to clarity, so for different devices, different measure has been made to measure it. There are some resolution measurement techniques:

(a) **Dots per inch:** Dots per inch or DPI is usually used in monitors.

(b) **Lines per inch:** Lines per inch or LPI is usually used in laser printers.

(c) **Pixels per inch:** Pixel per inch or PPI is measure for different devices such as tablets, mobile phones etc.

(a) **Dots per inch:** The DPI is often related to PPI, whereas there is a difference between these two. DPI or dots per inch is a measure of spatial resolution of printers. In case of printer, DPI means that how many dots of ink are printed per inch when an image get printed out from the printer.

Remember, it is not necessary that each Pixel per inch is printed by one dot per inch. There may be many dots per inch used for printing one pixel. The reason behind this that most of the color printers uses CMYK model. The colors are limited. Printer has to choose from these colors to make the color of the pixel whereas within pc, you have hundreds of thousands of colors.

The higher is the DPI of the printer the higher is the quality of the printed document or image on paper. Usually, some of the laser printers have DPI of 300 and some have 600 or more.

(b) **Lines per inch:** When DPI refers to dots per inch, lines per inch refers to lines of dots per inch. The resolution of halftone screen is measured in lines per inch.

The following table shows some of the lines per inch capacity of the printers.

Table 1.2

Printer	LPI
Screen Printing	45–65 lpi
Laser Printer (300 dpi)	65 lpi
Laser Printer (600 dpi)	85–105 lpi
Offset Press (Newsprint Paper)	85 lpi
Offset Press (Coated Paper)	85–185 lpi

(c) **Pixels per inch:** Pixel density or Pixels per inch is a measure of spatial resolution for different devices that includes tables, mobile phones.

The higher is the PPI, the higher is the quality. In order to more understand it, that how it calculated. Lets calculate the PPI of a mobile phone.

For example The Samsung Galaxy s4 has PPI or pixel density of 441. But how does it is calculated?

First of all we use Pythagoras theorem to calculate the diagonal resolution in pixels.

It can be given as:

$$c = \sqrt{a^2 + b^2} \quad \text{...(1.12)}$$

Where a and b are the height and width resolutions in pixel and c is the diagonal resolution in pixels.

For example Samsung Galaxy s4, it is 1080 × 1920 pixels.

So putting those values in the equation gives the result

$$C = 2202.90717 \quad \text{...(1.13)}$$

Now we can calculate PPI

$$\text{PPI} = 2202.90717/5.0$$

$$\text{PPI} = 440.58$$

$$\text{PPI} = 441 \text{ (Approx.)} \quad \text{...(1.14)}$$

That means that the pixel density of Samsung Galaxy s4 is 441 PPI.

1.11.3 Gray Level Resolution

Gray level resolution refers to the predictable or deterministic change in the shades or levels of grey in an image. In short grey level resolution is equal to the number of bits per pixel.

We have already discussed bits per pixel in previous section of bits per pixel and image storage requirements.

1.11.4 Bits Per Pixel (BPP)

The number of different colors in an image is depends on the depth of color or bits per pixel.

The mathematical relation that can be established between grey level

resolution and bits per pixel can be given as:

$$L = 2^k \qquad \ldots(1.15)$$

In this equation L refers number of gray levels. It can also be defined as the shades of gray and k refers to bits per pixel. So the 2 raise to the power of bits per pixel is equal to the gray level resolution.

For example:

Figure 1.17

The above image of Einstein is a grey scale image. Means it is an image with 8 bits per pixel or 8 bpp.

Now if we have to calculate the gray level resolution, then

$$L = 2^k$$

where, $k = 8$

So $\qquad L = 2^8$

So $\qquad L = 256 \qquad \ldots(1.16)$

It means its gray level resolution is 256 or in other way we can say that this image has 256 different shades of gray.

The more is the bits per pixel of an image, the more is its grey level resolution.

1.11.5 Finding BPP from Gray Level Resolution

We can find the bits per pixels from the given gray level resolution. For this, we just have to twist the formula a little.

Since we have

$$L = 2k$$

This formula finds the levels. Now if we have to find the bits per pixel or in this case k, we will simply change it like this.

$$K = \log_2(L) \quad \text{...(1.17)}$$

Let's take an example to find bits per pixel from gray level resolution. If we are given an image of 256 levels. What are the bits per pixel required for it?

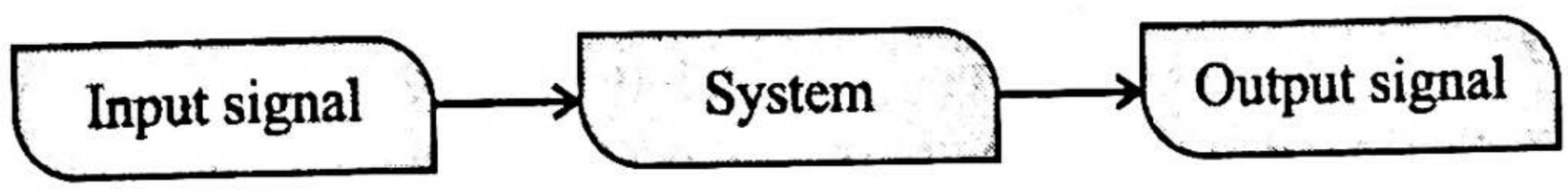
Putting $L = 256$, we get

$$K = \log_2(256)$$

$$K = 8 \text{ BPP} \quad \text{...(1.18)}$$

1.12 Gray Level Resolution and Quantization

Gray level resolution is found on the y-axis of the signal. We studied that digitizing, an analog signal requires two steps. Sampling and quantization.



Sampling is done on x axis. And quantization is done on Y axis.

So that means digitizing the gray level resolution of an image is done in quantization.

1.12.1 Spatial Operation

In digital system there are various operations or tasks which have to be performed. Among them spatial operations are performed directly on the pixels of a given image. They are broadly classified into three categories:

(a) **Single-Pixel Operations:** It is the simplest operation in digital image processing which is used to alter the values of its individual pixels based on their intensity.

In terms of transformation function T

$$S = T(z) \quad \text{...(1.19)}$$

where z is the intensity of a pixel in image and S is the mapped intensity of the corresponding pixel in the processing image.

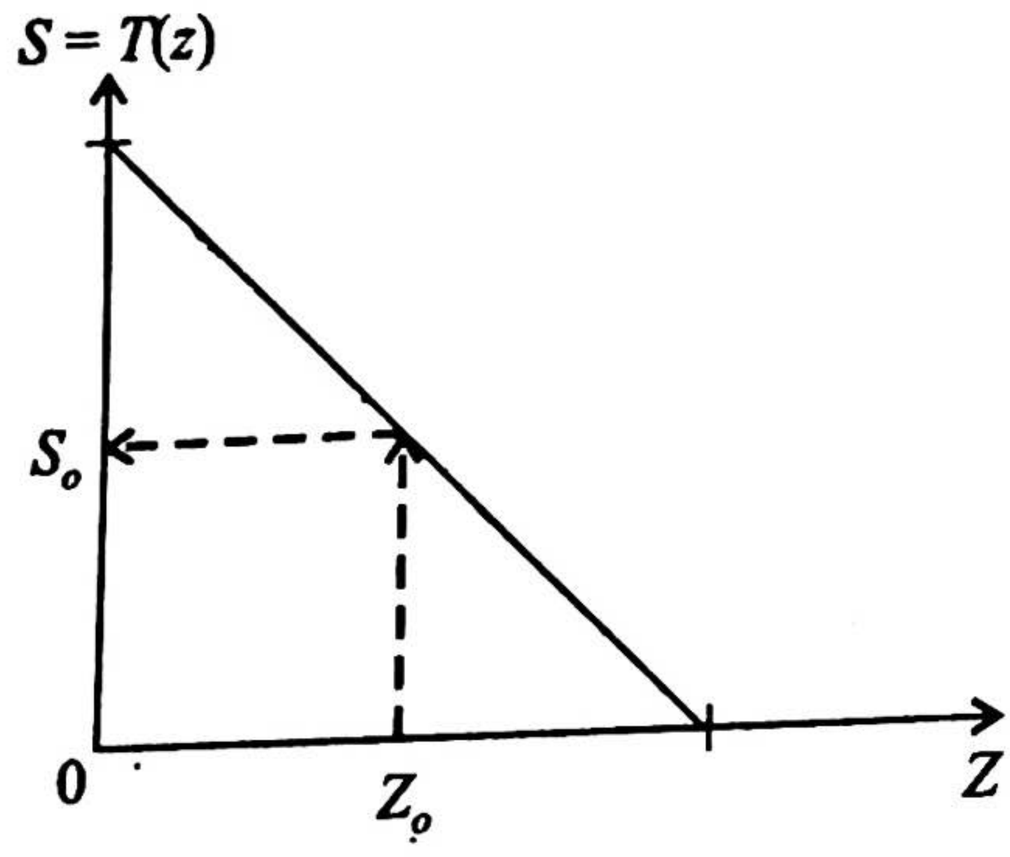


Figure 1.18

Figure 1.18 shows the intensity transformation function used to obtain the negative of an 8-bit image.

In this figure dashed arrows represents transformation of an arbitrary input intensity value z_0 into its corresponding output value,

(b) Neighbourhood Operations: The neighbourhood processing generates a corresponding pixel at the same coordinates in an output image. Let S_{xy} denotes the set of coordinates of a neighbourhood centred on an arbitrary point (x, y) in an image f. Then the value of pixels is determined by a specified operation involving in the input image with coordinates in S_{xy}'.

Let we have to compute the average value of the pixels in a rectangular neighbourhood of size $M \times N$ centred at (x, y). Then locations of pixels in this range constitute a set S_{xy}. This process can be mathematically explained as follows

$$g(x, y) = \frac{1}{MN} \sum_{(r,c)} f(r,c) \qquad \ldots(1.19)$$

Where

r and c are the row and column of the pixel having coordinates in the set S_{xy}.

The averaging neighbourhood image 'g' is created by varying the coordinates (x, y). So that the center of this image moves from pixel to pixel in image f and repeat this operation at each new location.

This operation is used to eliminate small details and thus render (changing its state) "blobs" corresponding to the largest regions of an image.

(c) **Geometric Spatial Transformations and Image Registration:** Geometric transformation is used to modify the spatial relationship between pixels in any image. Such types of transformations are known as "rubber-sheet" transformations because they may be viewed as analogous to printing an image on a sheet of rubber and then stretching the sheet.

In image processing the Geometric transformations have two basic operations. First is the spatial transformation of coordinates and second intensity interpolation which assign intensity values to the spatial transformed pixels.

From the first operation the transformation of coordinates is expressed as

$$(x, y) = T\{(v, w)\} \qquad \ldots(1.20)$$

Here (v, w) are pixel coordinates in original image and (x, y) are pixel coordinates of transformed image.

With this transformation we can shrink or zoom the original image. For example, if we have

$$(x, y) = T\{(v, w)\} = (v/2, w/2) \qquad \ldots(1.21)$$

then original image is shrink to half in both spatial directions.

The most widely used spatial coordinate transformation is the "affine transform" it is given in general as

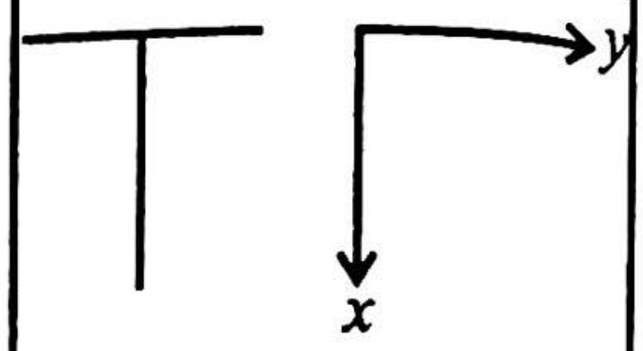
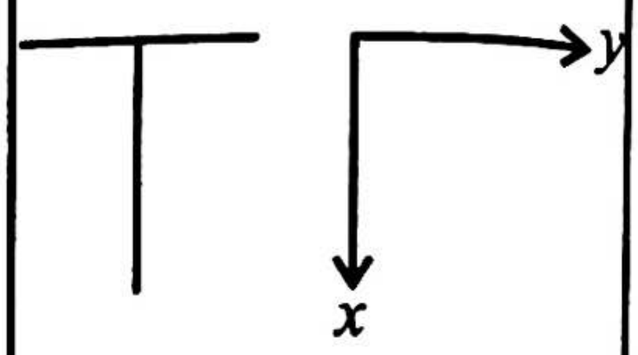
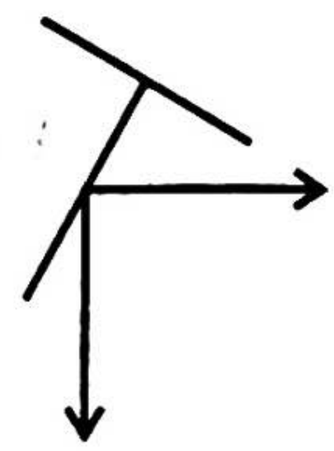
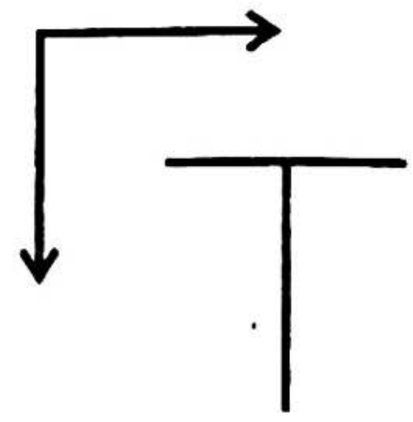
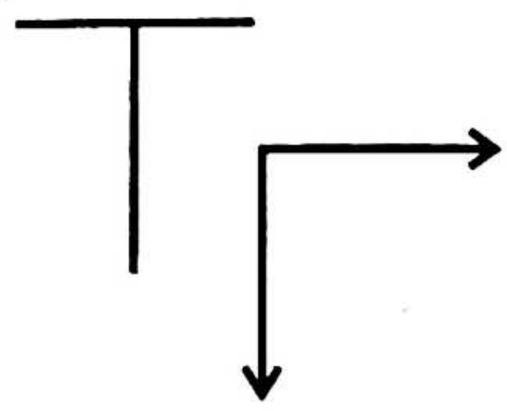
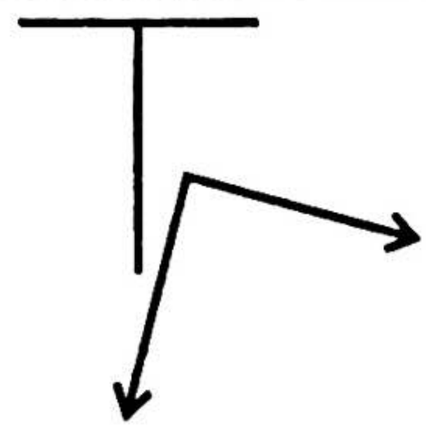
$$[x\ y\ 1] = [v\ w\ 1]T = [v\ w\ 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix} \qquad \ldots(1.22)$$

With the use of equation (1.22) i.e. affine transformation we can scale, translate, rotate or sheer a set of coordinates point for the element of matrix T.

Equation 1.22 can be use in two basic ways. The first is called a forward mapping which consists of scanning the pixels of the input image at (v, w) and computing the spatial location at (x, y) in the processed image.

Table 1.3 shows various transformation using "affine matrix" based on equation (1.22).

Table 1.3

Transformation Name	Affine Matrix T	Coordinate Equations	Example
IDENTIFY	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
SCALING	$\begin{bmatrix} C_x & 0 & 0 \\ 0 & C_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = C_x V$ $y = C_y W$	
ROTATION	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - W$ $\sin\theta$ $y = v\sin\theta + W$ $\cos\theta$	
TRANSLATION	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = V + t_x$ $y = w + ty$	
SHEAR (Vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ S_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = V + S_v W$ $y = w$	
SHEAR (Horizontal)	$\begin{bmatrix} 1 & S_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = V$ $y = S_h + W$	

The second technique or way is called inverse mapping which scans the output pixel location and at each location (x, y). computer the corresponding location in the input image using

$$(v, w) = T^{-1}(x, y) \qquad \ldots(1.23)$$

Finally the intensity of the output pixel value is determined. This technique is widely used in numerous commercial implementations of spatial transformations.

1.13 Image Registration

Image registration is used to align two or more images of the same scene. Here we have input and output images, but the specific transformation which produces the output image from the input is unknown. So here we have to estimate the transformation function to register the image. In short we have an input image that we wish to transform and a reference image against we want to register the input.

In this technique there are some problems arises like compensating for geometric distortions caused by differences in the viewing angle, distance and orientation, sensor resolution, shift in object positions etc.

All these problem can be solved by using a principle approach which is called "tie points" or 'control points'. These points are corresponding points whose locations are known precisely in the input and reference images.

Most of the imaging systems have physical small metallic objects embedded in the imaging sensors, which produce a set of "known points" also called 'Reseau marks' on all images captured by the system which can be used as guides for establishing tie points.

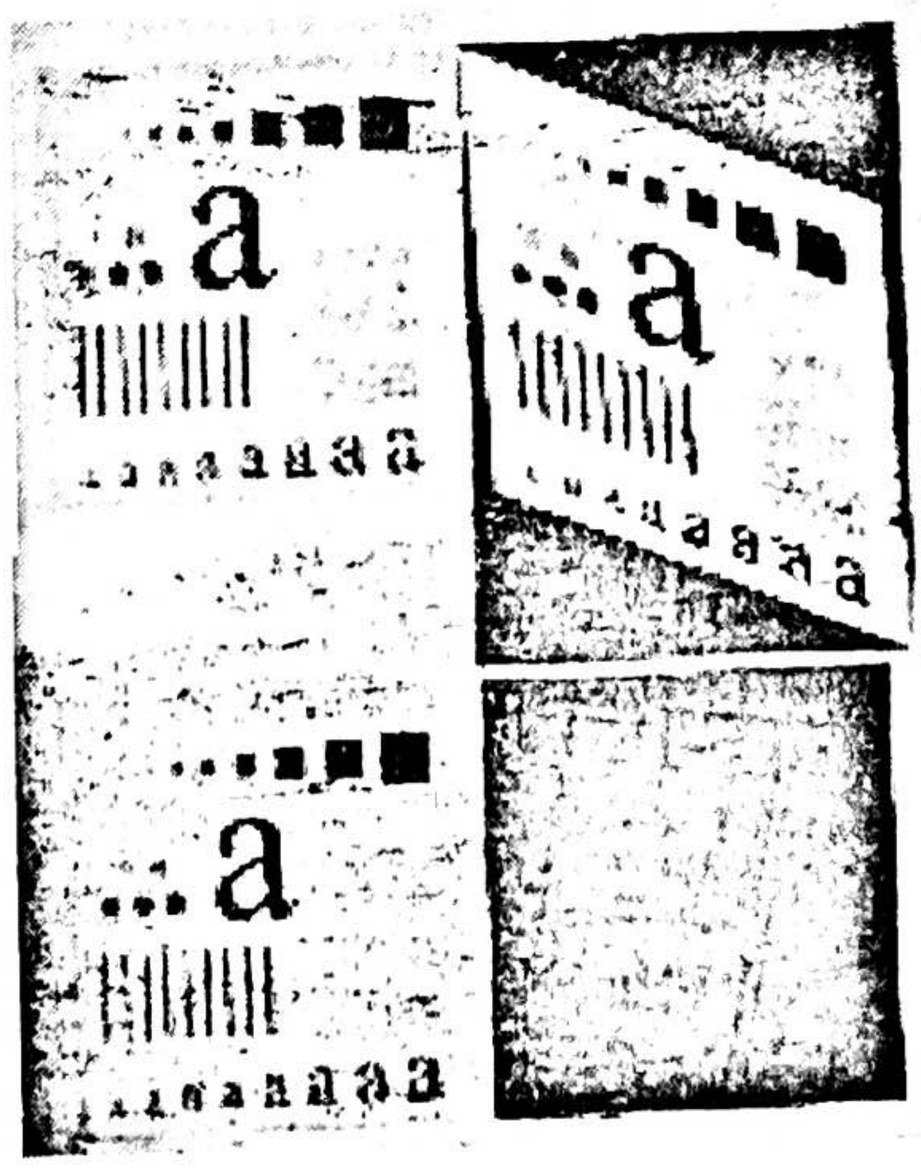


Figure 1.19 Image Registration

1.14 Vector and Matrix Operations

In image processing various operations are performed routinely, among them vector and matrix operation are one.

Since we know that color images are formed in RGB color space by using red, green and blue component images. In this color image each pixel of image has three components in the form of column vector.

i.e.
$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \quad \ldots(1.24)$$

Here Z_1, Z_2 and Z_3 are the intensity of the pixel in the red, green and blue images respectively.

Figure 1.20 shows the formation of a vector from corresponding pixel values in basic RGB color images.

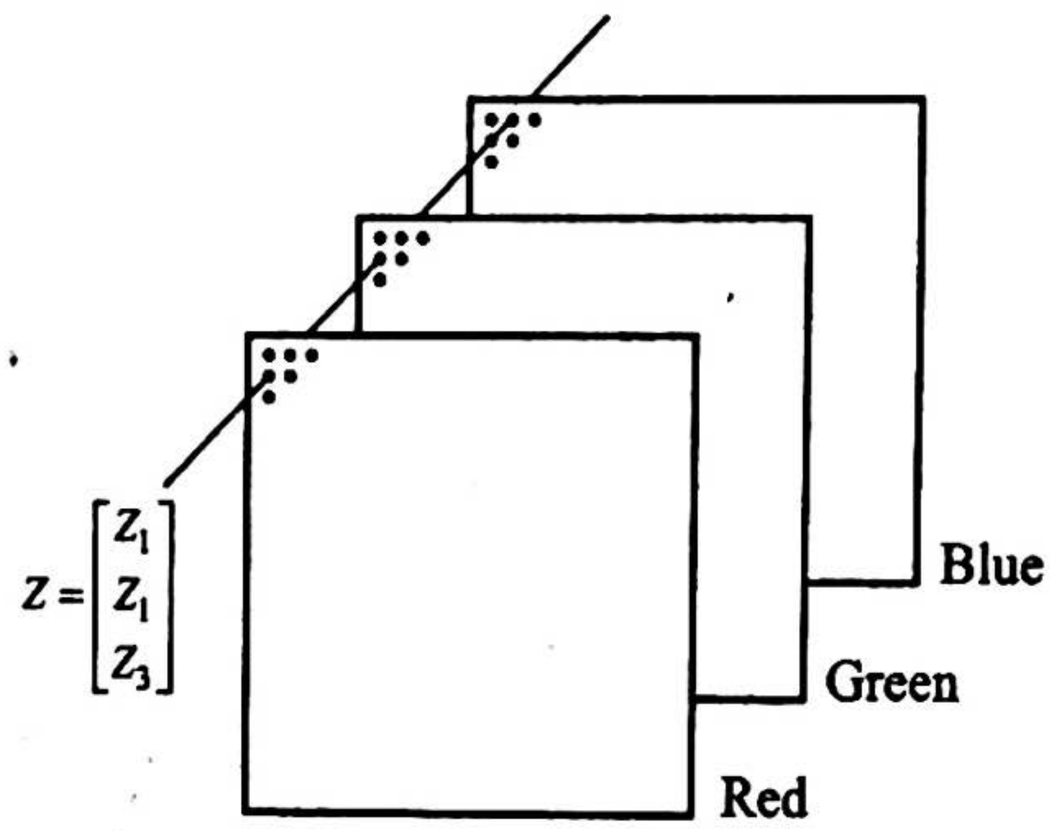


Figure 1.20 Basic RGB color image in vector form

An RGB color image of size $M \times N$ can be represented by three basic components of this size or we can say by a total of MN 3-D vectors.

Let the Euclidean distance D, between a pixel vector Z and an arbitrary point 'a' in n-dimensional space is defined as vector product

i.e.
$$D(z, a) = [(z - a)^T (z - a)]^{1/2}$$
$$= [(z_1 - a_1)^2 + (z_2 - a_2)^2 + (z_3 - a_3)^3 + \ldots (z_n - a_n)^2]^{1/2} \quad \ldots(1.25)$$

Equation (1.25) a generalized form of the 2-D Euclidean distance which is sometime known as "vector norm" denoted by $|z - a|$.

These pixel vectors have importance in Linear transformation. Which is given as

$$W = A(z - a) \quad \ldots(1.26)$$

Here A is a $m \times n$ size Matrix and z and a are colum vector of size $n \times 1$.

For example we can express an image of size $M \times N$ as a vector of dimension $MN \times 1$ by letting the first row of the image be the first N elements of the vector, the second row of the image be Next N elements and so on. By using the complete range of such processes we have

$$g = Hf + n \qquad \ldots(1.27)$$

Here f is an $MN \times 1$ vector showing an input image and n is an $MN \times 1$ vector showing an $M \times N$ noise pattern. g is an $MN \times 1$ vector showing a processed image and H is a $MN \times MN$ matrix showing a linear process applied to the input image.

1.15 Color Model

Color Model is also known as color space or color system. It is used to facilitate the specification of colors in some standard. In general, a color model provides a coordinate system and a subspace within that system where each color is represented by a single point.

All the color model are either hardware oriented or application oriented to achieve color manipulation. The hardware oriented color model are the RGB (Red, Green, Blue) for color monitors and color video cameras. The CMY (Cyan, Magenta, Yellow) and CMYK (Cyan, Magenta, Yellow, Black) for color printing. The HSE (Hue, Saturation, Intensity) which are used to describe and interpret color by humans.

1.16 The RGB Color Model

In this model all the images appear in its primary spectral components i.e. in red, green and blue. It is bases on Cartesian Coordinate System.

Figure 1.21 represents a schematic of the RGB color cube having primary value at three corners, secondary colors cyan, magenta and yellow are at three other corners, black is at the origin and white is at the farthest corner from the origin.

The points of equal RGB value is called gray scale which is extended from black to white along the line joining of these two points.

Here all values of R,G,B are assumed in the range [0,1]. The image consisting of three main color when fed into an RGB monitor, these images combine on the screen to produce a composite color image. In the RGB color model the number of bits used to represent each pixel is called the "pixel depth".

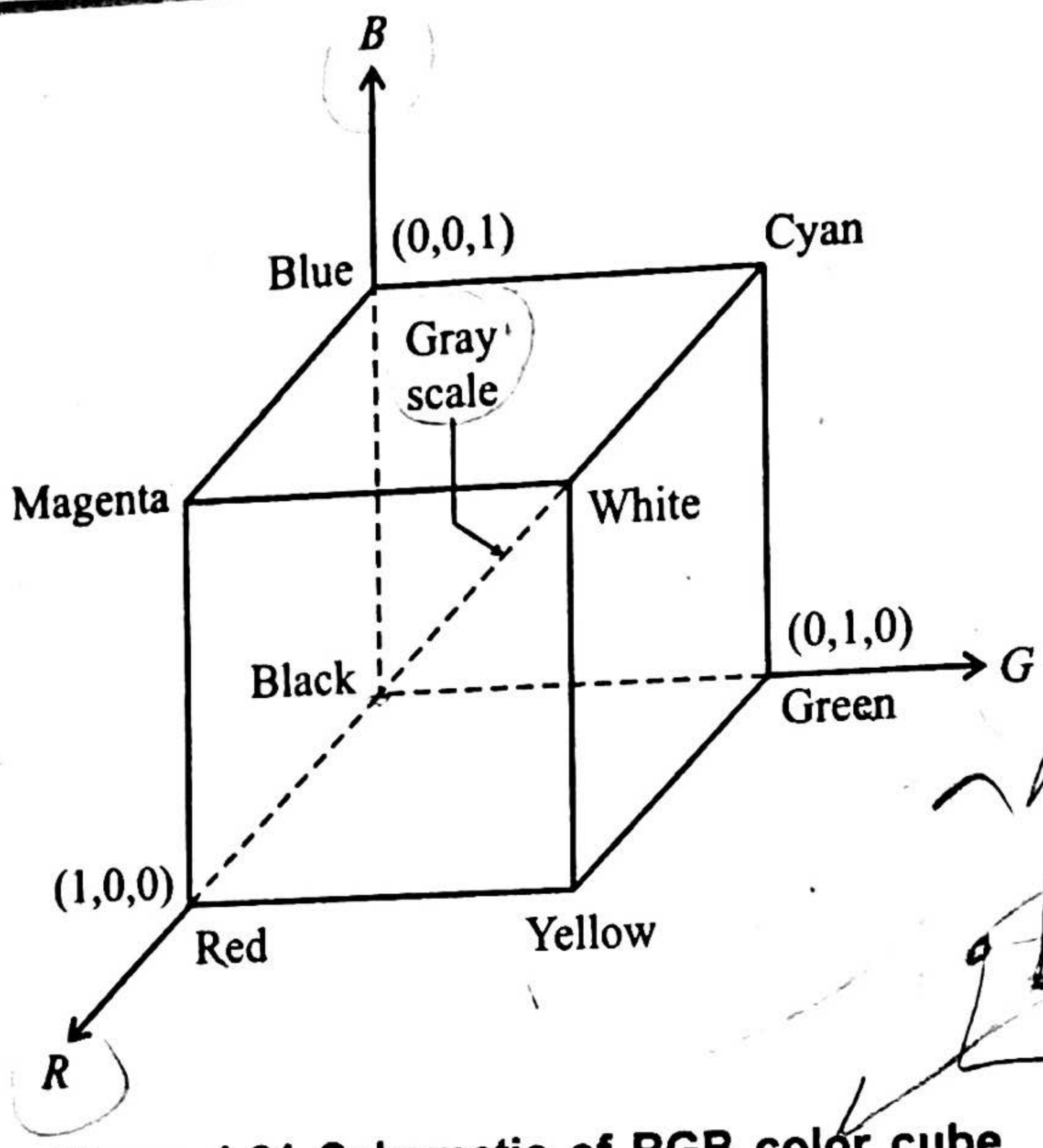


Figure 1.21 Schematic of RGB color cube

For example if we have an 8-bit RGB image then we have depth of 24-bit RGB color pixel. Full color image is used to denote a 24-bit RGB color image which is shown in figure 1.22.

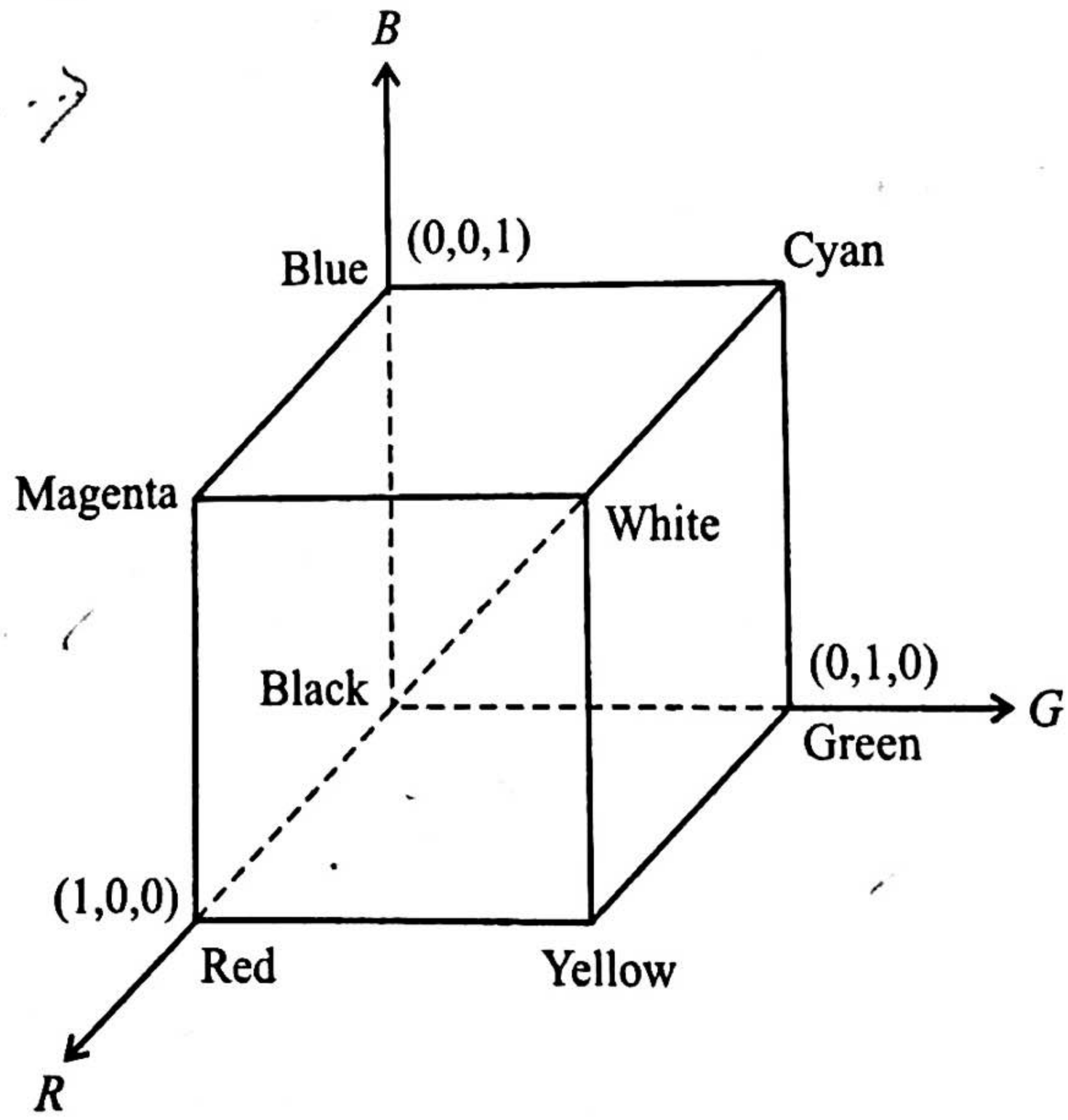


Figure 1.22 RGB full color image of 24-bit

This figure 1.22 is a solid and composed of $(2^8)^3 = 16,777,216$ color

1.16.1 Generation of RGB Image

A color plane is the face or the cross section of the cube. This color plan can be generated by fixing one of the three colors and allowing the other to vary. Figure 1.23 show a generation of the RGB image of cross-sectional color plane (127,G,B) which is placed at centre of the cube and parallel to the GB-plane.

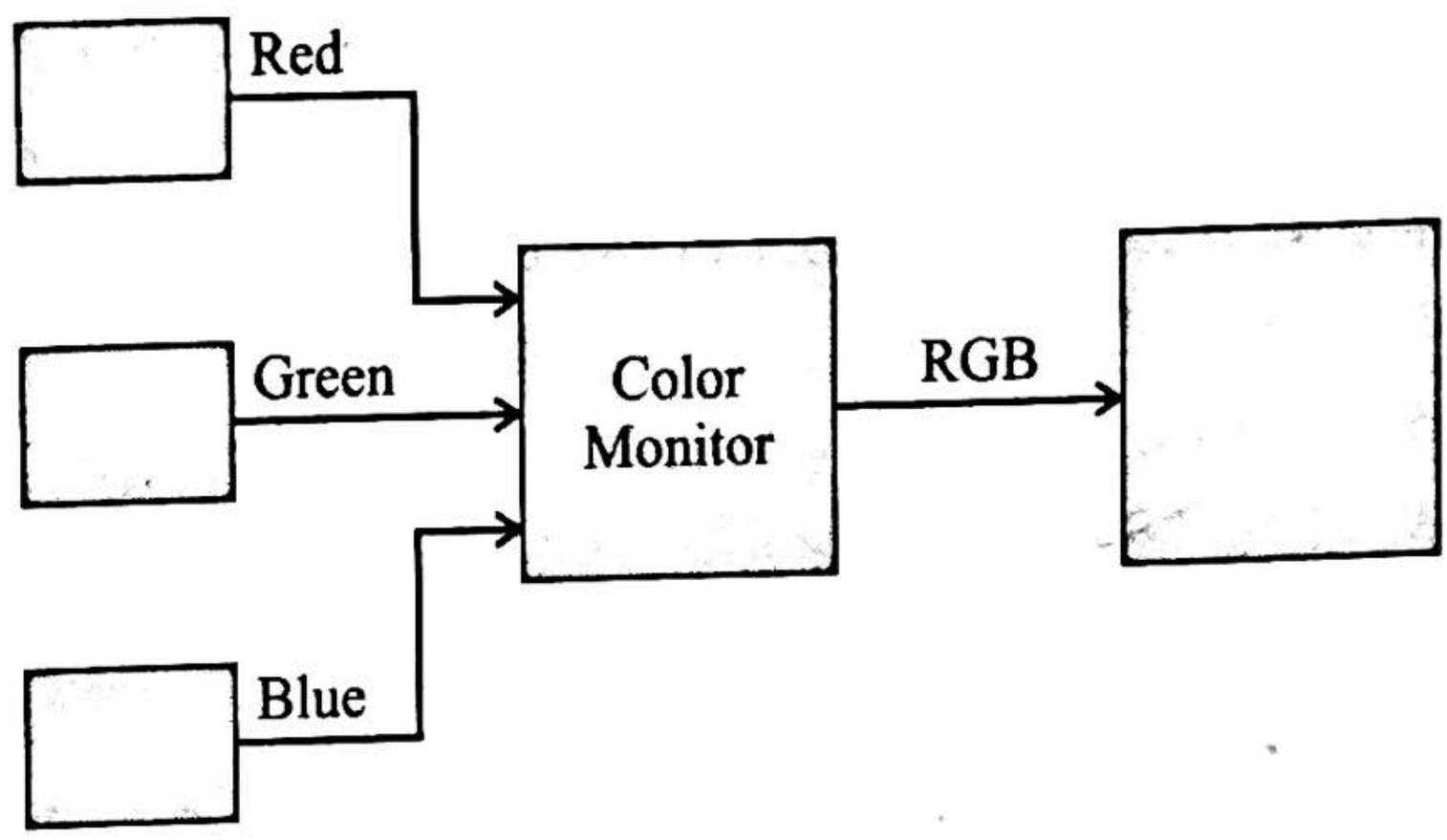


Figure 1.23 Color plan of generation of RGB

1.17 The CMY and CMYK Color Models

The primary colors of the pigments (particular colors) or we can say that the secondary colors of light are cyan, magenta and yellow. Mostly color printers and copiers use CMY data input or perform an RGB to CMY conversion. In mathematical

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \ldots(1.28)$$

Equation 1.28 shows that light reflected from surface coated with pure cyan does not contain red, pure magenta does not contain green and pure yellow does not reflect blue.

When we combine all these three colors for printing it produces a muddy-looking black.

So to produe true black color in the printing fourth color 'Black" is added which produce CMYK color model. So due to this it is also called "four color printing model".

1.18 The HSI Color Model

Since RGB and CMY are the hardware implementations color models but these system are not well suited for describing colors in terms of human interpretation.

When a human being view a color object, it is described by its hue, Saturation and brightness (intensity).

Hue: It is a color attribute which describe a pure colors (Pure yellow, orange or red).

Saturation: It gives a measure of the degree to which a pure color is diluted by white light.

Brightness (Intensity): It is a subjective description which is practically impossible to measure. It is one of the key factor in describing color sensation.

So the HSI color model decouples the intensity component from the color-carrying informations (hue and saturation) in a color image. In general we can say that RGB is ideal for image color generations but it can not be used for color description and HSI models serves these color description applications.

If we take a color cube of figure 1.21 and stand it on the black vertex (0, 0, 0) with the white vertex (1, 1, 1) directly above it then intensity is along the line joining these two vertices. It is shown in figure 1.24(a).

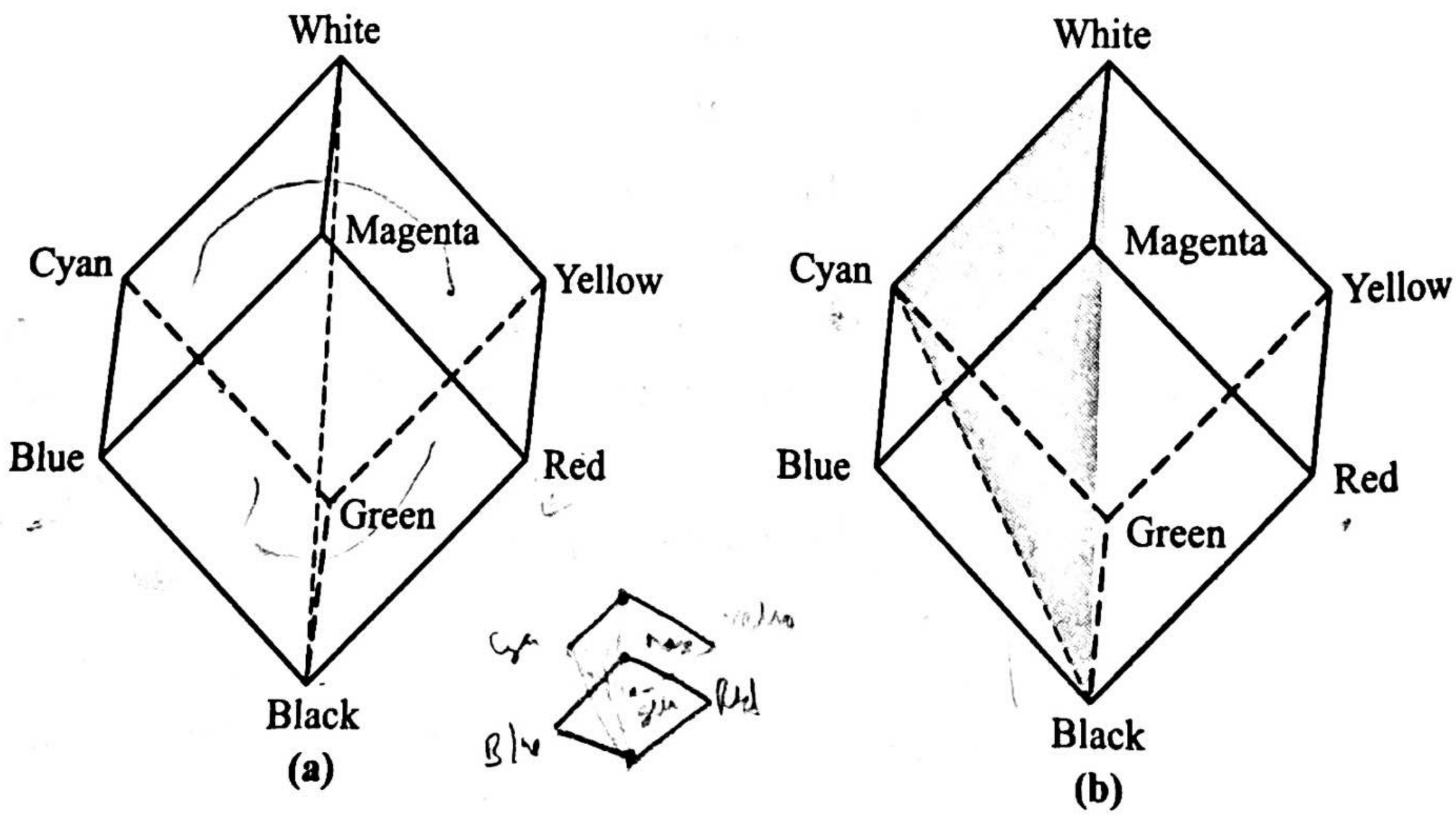


Figure 1.24 Interpretation between RGB and HSI color models

That if we want to determine the intensity of any color point then we have to pass a plane perpendicular to the intensity axis containing the color points.

The intersection of the place with the intensity axis gives us a point with intensity values in the range [0, 1]. It gives that saturation of a color increases as a function of distance from the intensity axis.

Figure 1.24(b) is used to represent/determine hue. In this figure, black and white points are contained in plane of intensity axis and all points on intensity axis and boundaries of the cube have same hue. So we can conculded that image have cyan color hue because black and white components can not change the hue. So in general we can say that we can convert any RGB point to a corresponding point in the HSI color model provided that HSI space is represented by a vertical intensity axis and the locus of color points that lie on planes perpendicular to this axis.

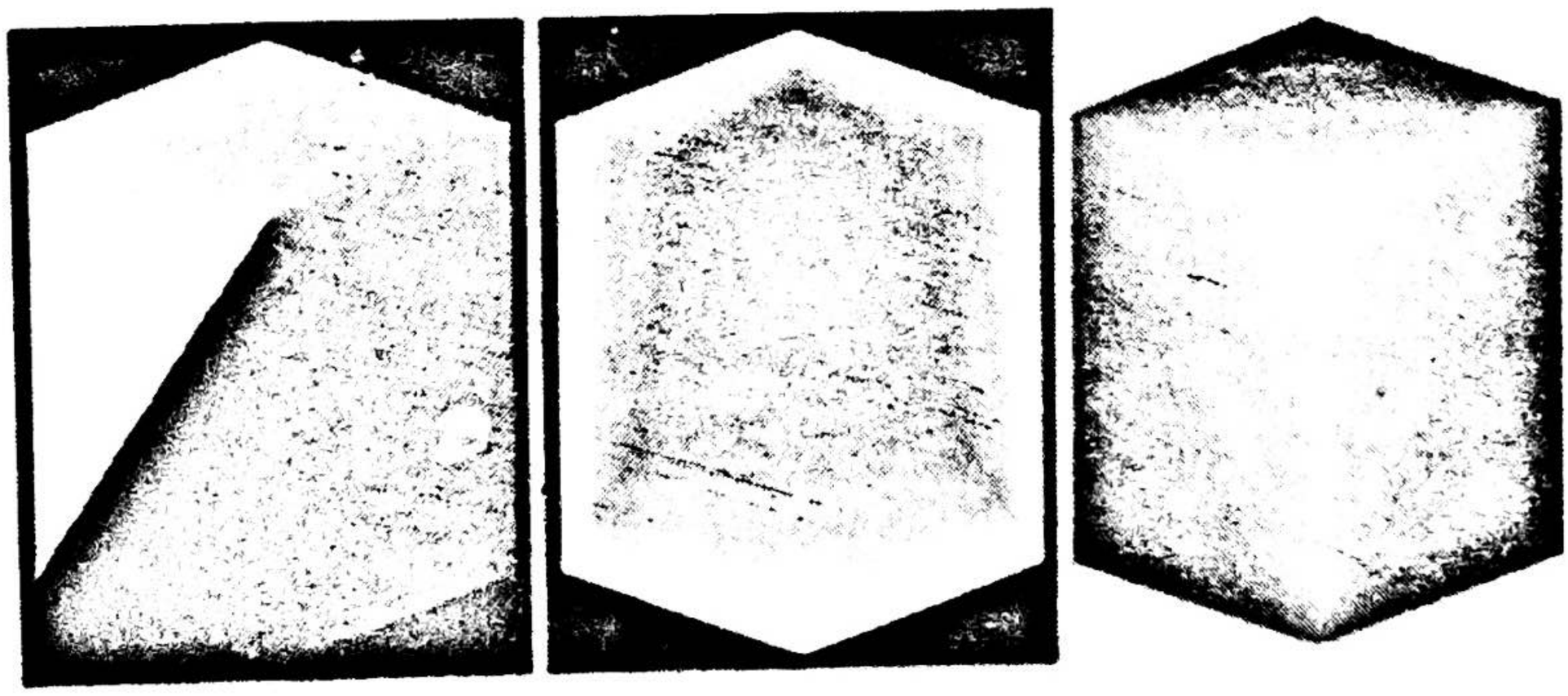


Figure 1.25 Hue/saturation/intensity

1.19 Converting Colors from RGB Model to HSI Model

'*H*' component of each RGB pixel in an RGB color format is

$$H = \begin{cases} \theta & \text{if} \quad B \le G \\ 360 - \theta & \text{if} \quad B > G \end{cases} \qquad \ldots(1.29)$$

$$\frac{\frac{1}{2}(R-G)+(R-B)}{(R-G)^2+(R-B)(G-B)}$$

where

$$\theta = \cos^{-1}\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{(R-G)^2+[(R-B)(G-B)]^{1/2}}\right] \qquad \ldots(1.30)$$

and the saturation component's form RGB model is

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \qquad \ldots(1.31)$$

and the intensity component

$$I = \frac{1}{3}[R + G + B] \quad \ldots(1.32)$$

Provided that RGB values have been normalized to the rang (0, 1) and angle θ is measured with respect to red axis of HSI model.

1.20 Converting Colors from HSI Model to RGB Model

There are three sectors to calculate the value of R, G and B from HSI model.

1. RG Sector ($0° \le H < 120°$)

   RGB components in this sector are given as

$$R = \left\{1 + \left[\frac{S\cos H}{\cos(60° - H)}\right]\right\} I \quad \ldots(1.33)$$

$$G = 3I - (R + B) \quad \ldots(1.34)$$

and $$B = I(1 - S) \quad \ldots(1.35)$$

2. GB Sector ($120° \le H < 240°$)

   First of all H component is subtract 120° from it.

i.e. $$H = H - 120° \quad \ldots(1.36)$$

then $$R = (1 - S)I \quad \ldots(1.37)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right] \quad \ldots(1.38)$$

$$B = 3I - (R + G) \quad \ldots(1.39)$$

3. BR Sector ($240° < H < 360°$)

Here $$H = H - 240° \quad \ldots(1.40)$$

then $$R = 3I - (G + B) \quad \ldots(1.41)$$

$$G = I(1 - S) \quad \ldots(1.42)$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right] \quad \ldots(1.43)$$

1.21 Digital Image Representing

"Virtual image, a point or system of points, on one side of a mirror or lens, which, if it existed, would emit the system of rays which actually exists on the other side of the mirror or lens."

1.21.1 Vector Images

One way to describe an image using numbers is to declare its contents using position and size of geometric forms and shapes like lines, curves, rectangles and circles; such images are called vector images.

1.21.2 Coordinate System

We need a coordinate system to describe an image, the coordinate system used to place elements in relation to each other is called used space, since this is the coordinates the user uses to define elements and position them in relation to each other.

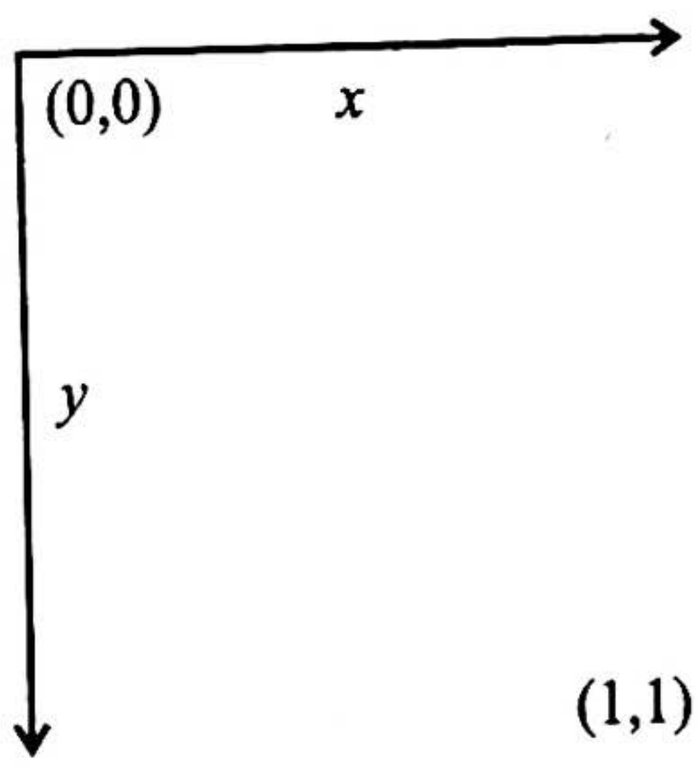


Figure 1.26 Coordinate system

The coordinate system used for all examples has the origin in the upper left, with the x axis extending to the right and y axis extending downwards.

1.21.3 Shapes

It would have been nice to make a smiling face, instead of the dissatisfied face on the left, by using a bezier curve, or the segment of a circle this could be achieved, this being a text focusing mainly on raster graphics, though, that would probably be too complex.

A simple image of a face can be declared as follows:

draw circle

centre	0.5, 0.5
radius	0.4
fill color	yellow
stroke	black
stroke width	0.05

draw circle	
centre	0.35, 0.4
radius	0.05
fill color	black
draw circle	
centre	0.65, 0.4
radius	0.05
fill color	black
draw line	
start	0.3, 0.6
end	0.7, 0.6
stroke-color	black
stroke-width	0.1

A vector image of a face, and the instructions used to create the image.

The preceding description of an image can be seen as a "cooking recipe" for how to draw the image, it contains geometrical primitives like lines, curves and circles describing color as well as relative size, position and shape of elements. When preparing the image for display it has to be translated into a bitmap image, this process is called rasterization.

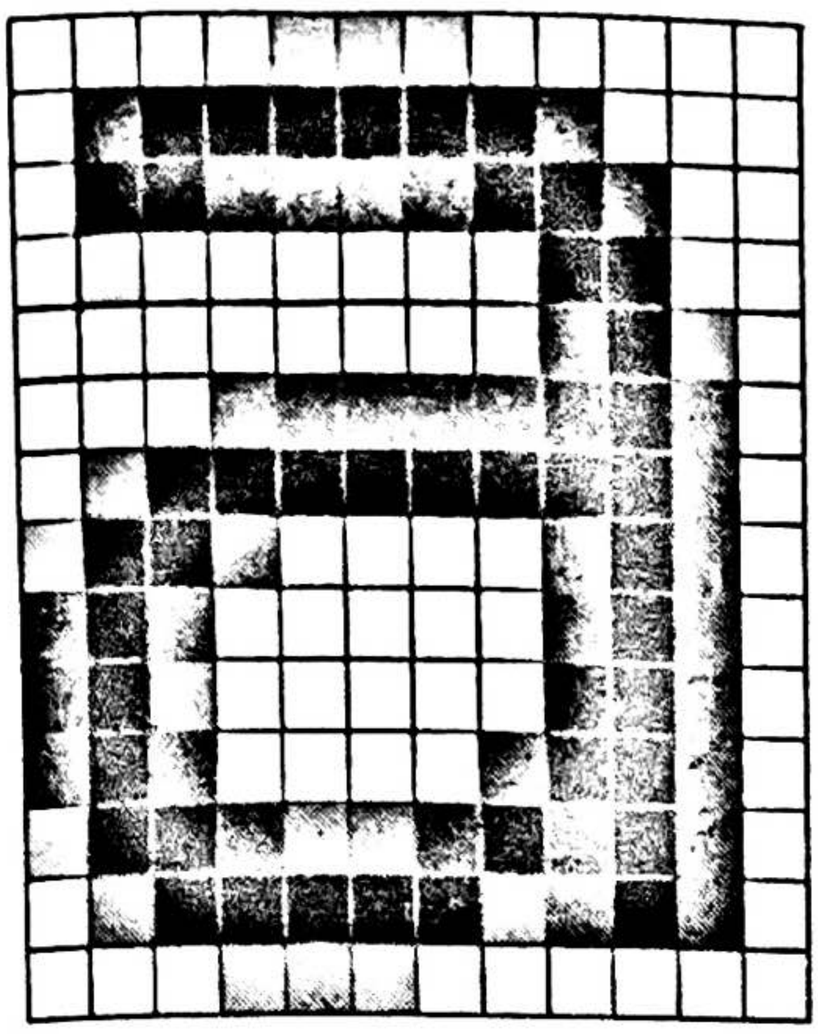
A vector image is resolution independent, this means that we can enlarge or shrink the image without affecting the output quality. Vector images are the preferred way to represent Fonts, Logos and many illustrations.

1.21.4 Bitmap Images

Bitmap or raster images are "digital photographs", they are the most common form to represent natural images and other forms of graphics that are rich in detail. Bitmap images is how graphics is stored in the video memory of a computer. The term bitmap refers to how a given pattern of bits in a pixel maps to a specific color. A rasterized form of the letter 'a' magnified 16 times using pixel doubling.

A bitmap image take the form of an array, where the value of each element, called a pixel picture element, correspond to the color of that portion of the image. Each horizontal line in the image is called a scan line.

a



1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Figure 1.27 Raster Image

The letter 'a' might be represented in a 12 × 14 matrix as depicted in Figure, the values in the matrix depict the brightness of the pixels (picture elements). Larger values correspond to brighter areas whilst lower values are darker.

1.22 Sampling

When measuring the value for a pixel, one takes the average color of an area around the location of the pixel. A simplistic model is sampling a square, this is called a box filter, a more physically accurate measurement is to calculate a weighted Gaussian average (giving the value exactly at the pixel coordinates a high weight, and lower weight to the area around it). When perceiving a bitmap image the human eye should blend the pixel values together, recreating an illusion of the continuous image it represents.

1.23 Raster Dimensions

The number of horizontal and vertical samples in the pixel grid is called Raster dimensions, it is specified as width × height.

Resolution is a measurement of sampling density, resolution of bitmap images give a relationship between pixel dimensions and physical dimensions. The most often used measurement is ppi, (pixels per inch).

A rasterized form of the letter 'a' magnified 16 times, where each pixel is represented as a circle instead of a square.

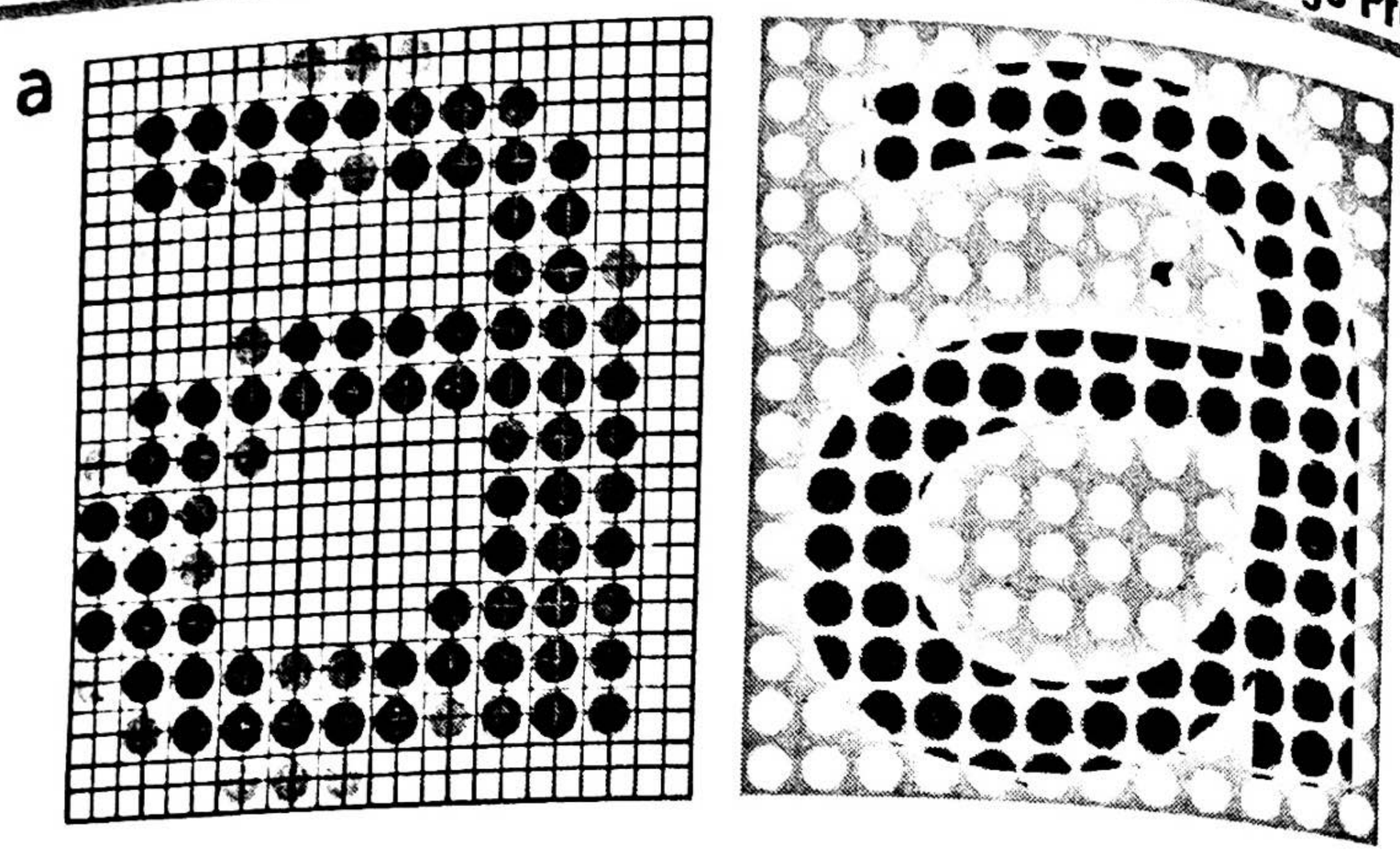


Figure 1.28 Sampling Grid

1.23.1 Megapixels

Megapixels refer to the total number of pixels in the captured imag an easier metric is raster dimensions which represent the number horizontal and vertical samples in the sampling grid. An image with 4 : 3 aspect ratio with dimension 2048 × 1536 pixels, contain a tot of 2048 × 1535 = 3,145,728 pixels; approximately 3 million, thus it a 3 megapixel image.

Table 1.4 Common/Encountered raster dimensions

Dimensions	Megapixels	Name	Comment
640 × 480	0.3	VGA	VGA
720 × 576	0.4	CCIR 601 DV PAL	Dimensions used for PAL DV, and PAL DVDs
768 × 576	0.4	CCIR 601 PAL full	PAL with square sampling grid ratio
800 × 600	0.4	SVGA	
1024 × 768	0.8	XGA	The currently (2004) most common computer screen dimensions.
1280 × 960	1.2		
1600 × 1200	2.1	UXGA	
1920 × 1080	2.1	1080i HDTV	Interlaced, high resolution digital TV format.
2048 × 1536	3.1	2K	Typically used for digital effects in feature films.

1.23.2 Scaling / Resampling

When we need to create an image with different dimensions from what we have we scale this image. A different name for scaling is resampling, when resampling algorithms try to reconstruct the original continuous image and create a new sample grid.

1.23.3 Scaling Image Down

The process of reducing the raster dimensions is called decimation, this can be done by averaging the values of source pixels contributing to each output pixel.

1.23.4 Scaling Image Up

When we increase the image size we actually want to create sample points between the original sample points in the original raster, this is done by interpolation the values in the sample grid, effectively guessing the values of the unknown pixels.

1.23.5 Sample Depth

The values of the pixels need to be stored in the computers memory, this means that in the end the data ultimately need to end up in a binary representation, the spatial continuity of the image is approximated by the spacing of the samples in the sample grid. The values we can represent for each pixel is determined by the sample format chosen.

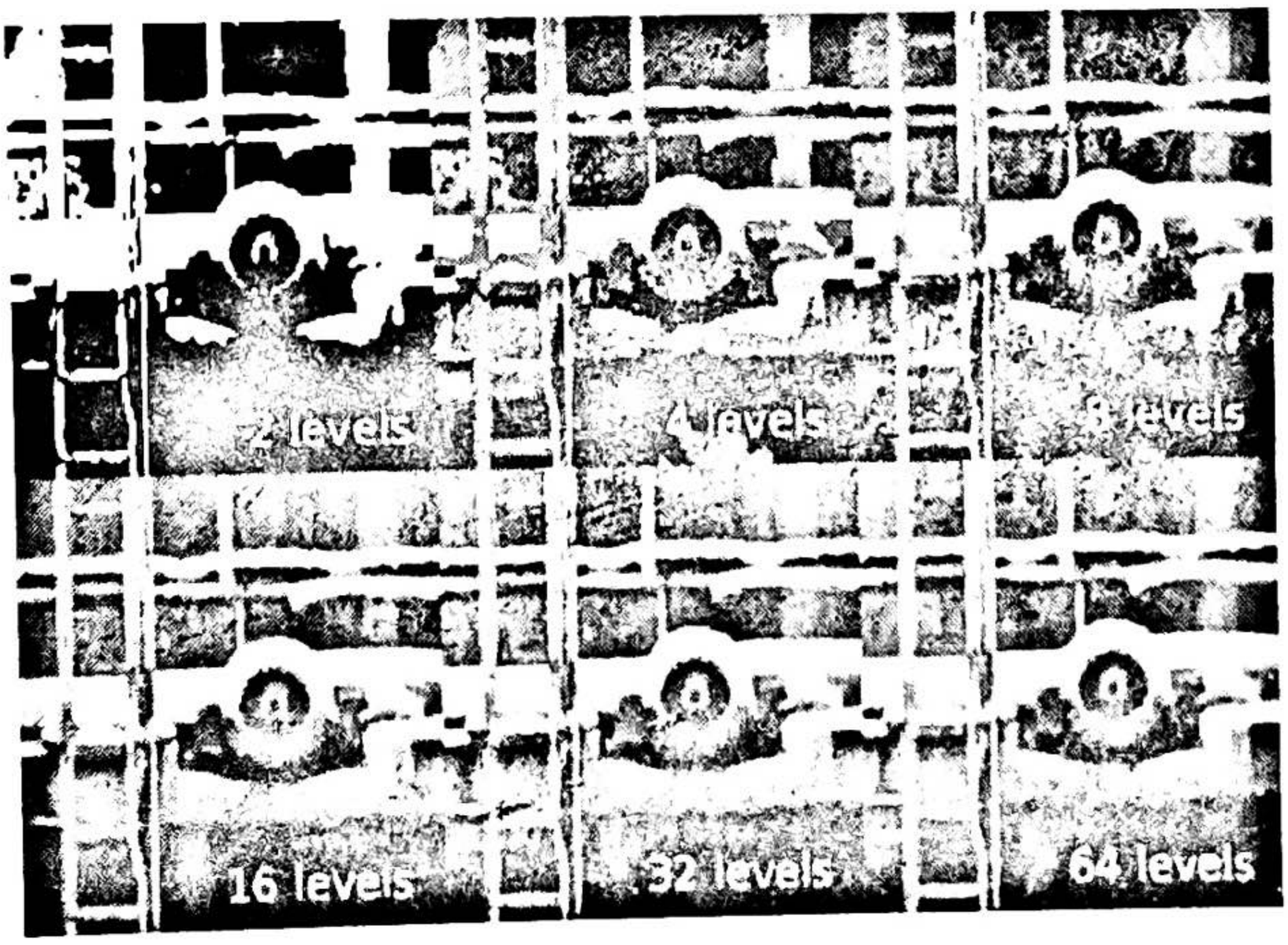


Figure 1.29 Sample depth

The same image width varying sample depths, note that high frequen areas (detailed areas) have an OK look earlier than low frequency are

8 bit

A common sample format is 8 bit integers, 8 bit integers can on represent 256 discrete values ($2^8 = 256$), thus brightness levels a quantized into these levels.

12 bit

For high dynamic range images (images with detail both in shadows an highlights) 8 bits 256 discrete values does not provide enough precisio to store an accurate image. Some digital cameras operate with more tha 8 bit samples internally, higher end cameras (mostly SLRs) also provid RAW images that often are 12bit ($2^{12} = 4096$).

16 bit

The PNG and TIF image formats supports 16 bit samples, many image processing and manipulation programs perform their operations in 16 bit when working on 8 bit images to avoid quality loss in processing.

Floating Point

Some image formats used in research and by the movie industry store floating point values. But "normal" 32 bit floating point values and a special format called half which uses 16bits/sample. Floating point is useful as a working format because quantization and computational errors are kept to a minimum until the final render.

Floating point representations often include HDR, (High Dynamic Range). High Dynamic Range images are images include sampling values that are whiter than white (higher values than 255 for a normal 8bit image). HDR allows representing the light in a scene with a greater degree of precision than LDR, (Low Dynamic Range images).

Colors

The most common way to model color in Computer Graphics is the RGB color model, this corresponds to the way both CRT monitors and LCD screens/projectors reproduce color. Each pixel is represented by three values, the amount of red, green and blue. Thus an RGB color image will use three times as much memory as a grey-scale image of the same pixel dimensions.

Color image built up of bands of red, green and blue color. (This is image illustrates how a laptop display is constructed, note that this image is preferred to be viewed on a computer screen, and not in print.)

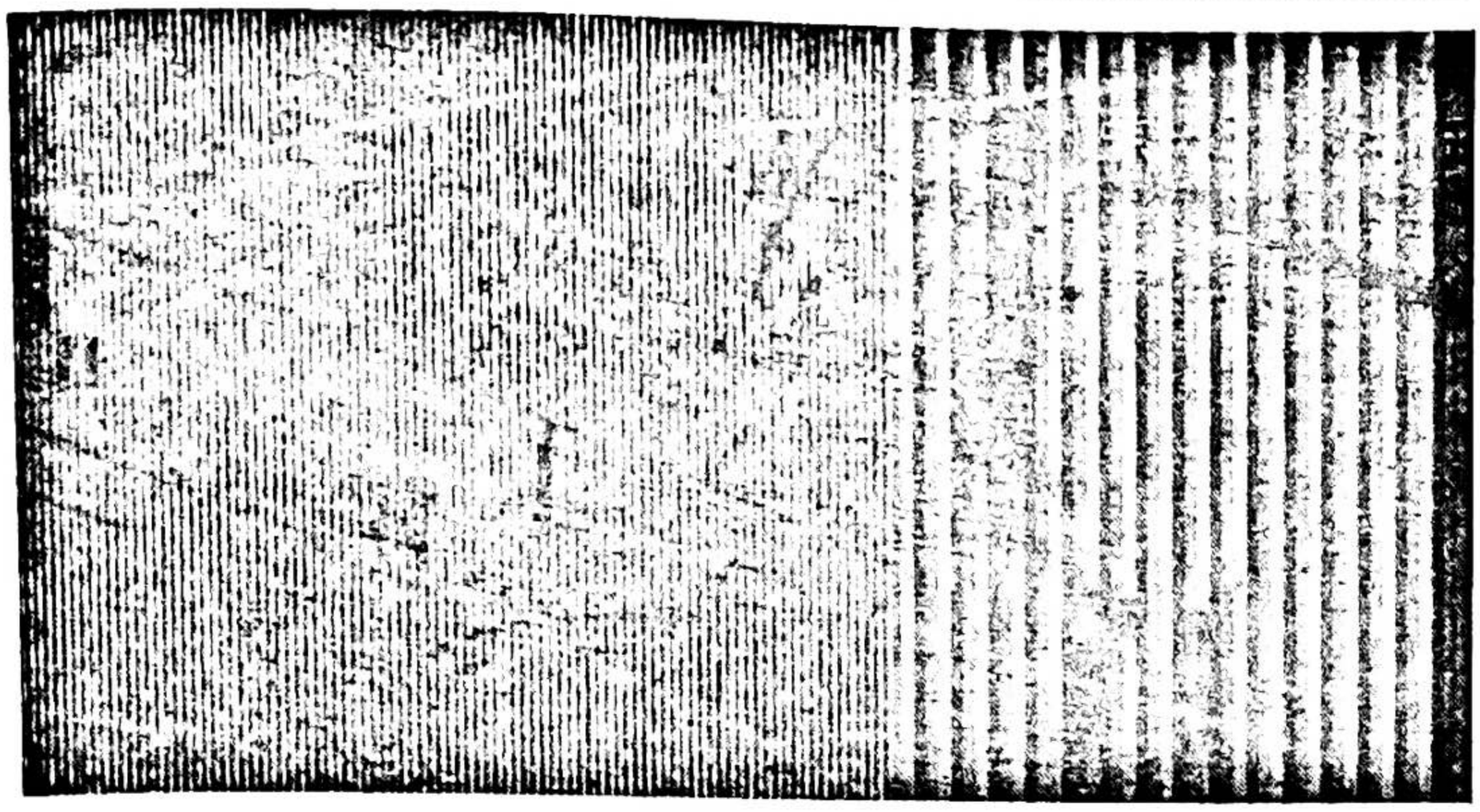


Figure 1.30 RGB bands

One of the most common pixel formats used is 8 bit RGB where the red, green and blue values are stored interleaved in memory. This memory layout is often referred to as chunky, storing the components in seperate buffers is called planar, and is not as common.

1.24 Palette/Indexed Images

It was earlier common to store images in a palletized mode, this works similar to a paint by numbers strategy. We store just the number of the palette entry used for each pixel. And for each palette entry we store the amount of red, green and blue light.

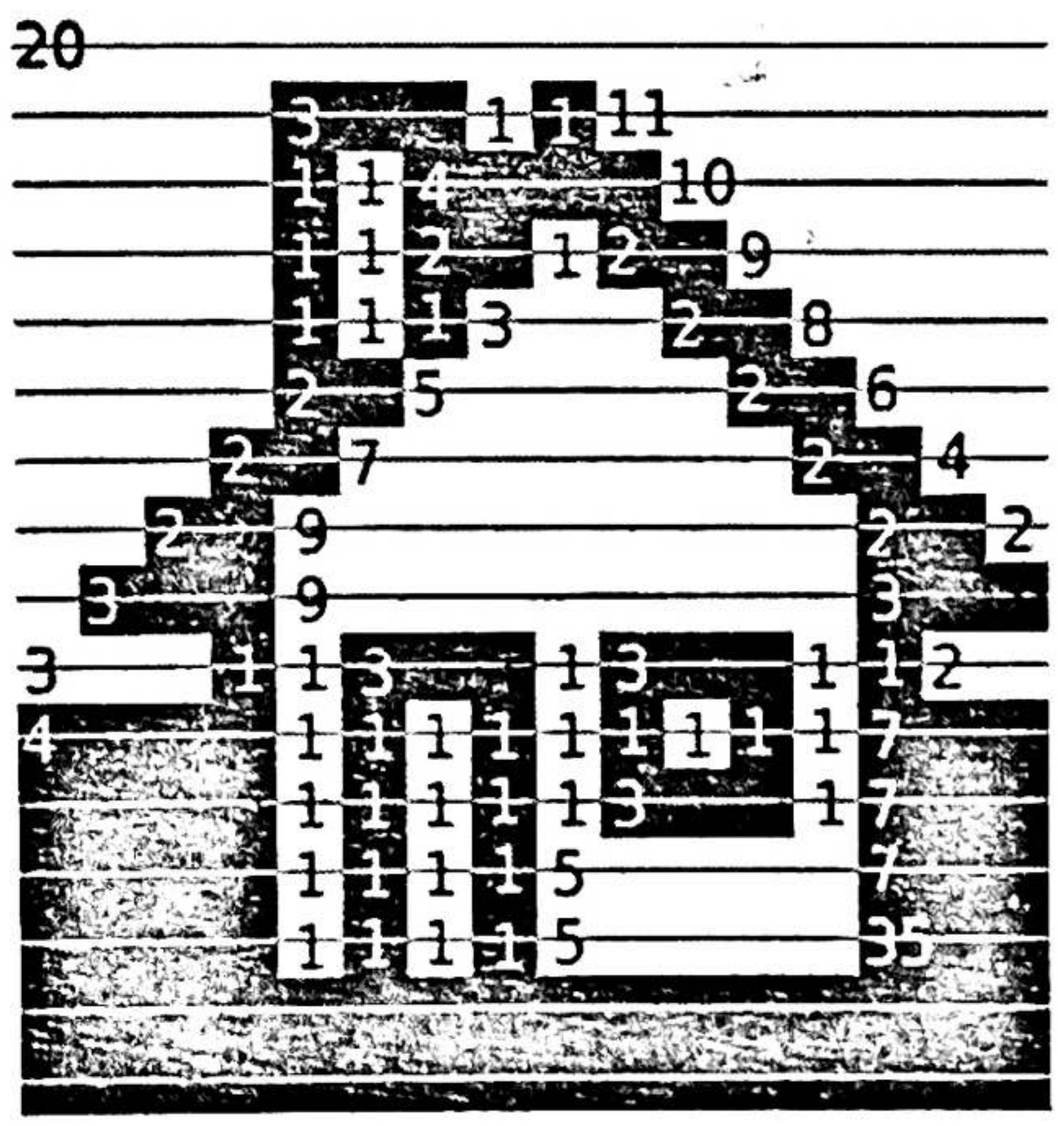


Figure 1.31 Indexed Image

On the left, image using just 16 colors, on the right the palette used for this image. The way an indexed/paletted image works is similar to how paint by number work.

1.25 Image Compression

Bitmap images take up a lot of memory, image compression reduces the amount of memory needed to store an image. For instance a 2.1 megapixel, 8bit RGB image (1600 × 1200) occupies 1600 × 1200 × 3 bytes = 5760000 bytes = 5.5 megabytes, this is the uncompressed size of the image. Compression ratio is the ratio between the compressed image and the uncompressed image, if the example image mentioned above was stored as a 512 kb jpeg file the compression ratio would be 0.5 mb : 5.5 mb = 1 : 11.

1.25.1 Lossless Image Compression

When an image is losslessly compressed, repetition and predictability is used to represent all the information using less memory.

70,
5, 25
5, 27,
4, 26,
4, 25,
6, 24,
6, 23,
3, 2, 3, 22,
3, 2, 3, 21,
3, 5, 2, 20,
3, 5, 2, 19,
3, 7, 2, 18,
3, 7, 2, 17,
14, 16,
14, 15,
3, 11, 2, 14,
3, 11, 2, 13,
3, 13, 2, 12,
3, 13, 2, 11,
3, 15, 2, 10,
3, 15, 2, 8,
6, 12, 6, 6
6, 12, 6, 64

Figure 1.32 Run-length encoding

The original image can be restored. One of the simplest lossless image compression methods is run-length encoding. Run-length encoding encodes consecutive similar values as one token in a data stream.

In Figure 1.32 Run-length encoding" a black and white image of a house has been compressed with run length encoding, the bitmap is considered as one long string of black/or white pixels, the encoding is how many bytes of the same color occur after each other. We'll further reduce the amount of bytes taken up by these 72 numerical values by having a maximum span length of 15, and encoding longer spans by using multiple spans separated by zero length spans of the other color.

70,	15, 0, 15, 0, 15, 0, 10,
5, 25	5, 15, 0, 10,
5, 27,	6, 15, 0, 12,
4, 26,	4, 15, 0, 11,
4, 25,	4, 15, 0, 10,
6, 24,	6, 15, 0, 9,
6, 23,	6, 15, 0, 8,
3, 2, 3, 22,	3, 2, 3, 15, 0, 7,
3, 2, 3, 21,	3, 2, 3, 15, 0, 6,
3, 5, 2, 20,	3, 5, 2, 15, 0, 5,
3, 5, 2, 19,	3, 5, 2, 15, 0, 4,
3, 7, 2, 18,	3, 7, 2, 15, 0, 3,
3, 7, 2, 17,	3, 7, 2, 15, 0, 2
14, 16,	14, 15, 0, 1
14, 15,	14, 15,
3, 11, 2, 14,	3, 11, 2, 14,
3, 11, 2, 13,	3, 11, 2, 13,
3, 13, 2, 12,	3, 13, 2, 12,
3, 13, 2, 11,	3, 13, 2, 11,
3, 15, 2, 10,	3, 15, 2, 10,
3, 15, 2, 8,	3, 15, 2, 8,
6, 12, 6, 6	6, 12, 6, 6
6, 12, 6, 64	6, 12, 6, 15, 0, 15, 0, 15, 0, 15, 0, 4

Figure 1.33

The new encoding is 113 nibbles long, a nibble is 4bit and can represent the value 0-4, thus we need 57 bytes to store all our values, which is

less than the 93 bytes we would have needed to store the image as a 1bit image, and much less than the 750 bytes needed if we used a byte for each pixel. Run length encoding algorithms used in file formats would probably use additional means to compress the RLE stream achieved here.

1.25.2 Lossy Image Compression

Lossy image compression takes advantage of the human eyes ability to hide imperfection and the fact that some types of information are more important than others. Changes in luminance are for instance seen as more significant by a human observer than change in hue.

JPEG is a file format implementing compression based on the Discrete Cosine Transform DCT, together with lossless algorithms this provides good compression ratios. The way JPEG works is best suited for images with continuous tonal ranges like photographs, logos, scanned text and other images with lot's of sharp contours/lines will get more compression artifacts than photographs.

1.25.3 Loss through Generations

Lossy compression algorithms should not be used as a working format, only final copies should be saved as jpeg since loss accumulates over generations.

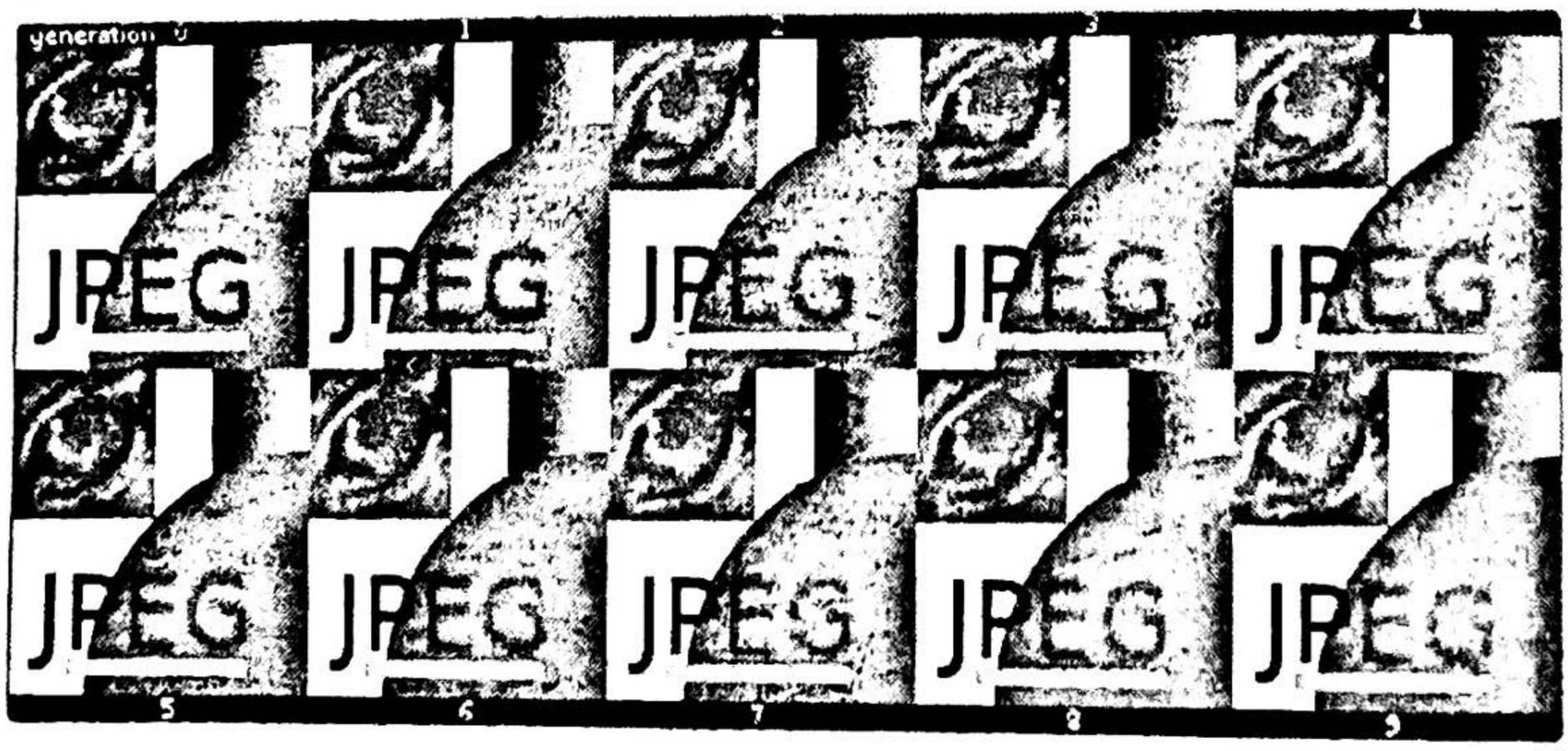


Figure 1.34 JPEG generation loss

An image specially constructed to show the deficiencies in the JPEG compression algorithm, saved, reopened and saved again 9 times.

JPEG is most suited for photographics content where the adverse effect of the compression algorithm is not so evident.

JPEG is not suited as an intermediate format, only use JPEG for final distribution where filesize actually matters.

1.26 File Formats and Applications

Many applications have their own internal file format, while other formats are more suited for interchange of data. Table refer lists some common image formats.

Table 1.5 Vector File Formats

Extension	Name	Notes
.ai	Adobe Illustrator Document	Native format of Adobe Illustrator (based on .eps)
.eps	Encapsulated Postscript	Industry standard for including vector graphics in print
.ps	PostScript	Vector based printing language, used by many Laser printers, used as electronic paper for scientific purposes
.pdf	Portable Document Format	Modernized version of ps, adopted by the general public as 'electronic print version'
.svg	Scalable Vector Graphics	XML based W3C standard, incorporating animation, gaining adoption
.swf	Shockwave Flash	Binary vector format, with animation and sound, supported by most major web browsers.

Table 1.6 Raster File Formats

Extension	Name	Notes
.jpg	Joint Photographic Experts Group	Lossy compression format well suited for photographic images.
.png	Portable Network Graphics	Lossless compression image, supporting 16bit sample depth, and Alpha channel.
.gif	Graphic Interchange Format	8 bit indexed bitmap format, is superceded by PNG on all accounts but animation.
.exr	EXR	HDR, High Dynamic Range format, used by movie industry.
.raw, .raw	Raw Image File	Direct memory dump from a digital camera, contains the direct imprint from the imaging sensor without processing with whitepoint and gamma correction. Different cameras use different extensions, many of them derivatives of TIFF, examples are .nef, .raf and .crw

.dgn	Digital Negative	A subset/clarification of TIFF, created by Adobe to provide a standard for storing RAW files, as well as exchanging RAW image data between application
.tiff, .tiff	Tagged Image File Format	
.psd	Photoshop Document	Native format of Adobe Photoshop, allows layers and other structural elements
.xcf	Gimp Project File	GIMP's native image format

REVIEW QUESTIONS

Q.1 What is an image. What are the characteristics of an image operation.

Q.2 What are the fundamental components of general purpose image processing system.

Q.3 What is image sampling and quantization.

Q.4 What do you mean by image registration.

Q.5 What is colour model or colour system. Explain RGB and CMY and HSI colour model.

Q.6 What are the various applications of digital?

$\lambda \nu = c$

$1nm = 10^{-9} m$

$c = 3 \times 10^{8} m/s$

000

CHAPTER 2

Basic Image Operations

Objectives

- Introduction
- Intensity Transformation Function
- Histogram
- Spatial Filtering
- Spatial Correlation and Convolution
- Vector Representation of Linear Filtering
- Generation of Spatial Filter Masks
- Sharpening Spatial filters
- Image Smoothing using Frequency Domain Filters
- Ideal Highpass Filters
- Buterworth High Pass Filters
- Gaussian High Pass Filter

2.1 Introduction

Spatial domain simply implies an image plane containing the pixels of an image. Spatial Domain can be implement directly on the pixel of an image to the frequency domain, and operation are performed on fourier transform of an image. Generally these are more efficient and require less processing resources to implement.

The spatial domain process in given as :

$$g(x,y) = T[f(x,y)] \quad \ldots(2.1)$$

Here $f(x,y)$ is the input image

$g(x,y)$ is the output image and

T is transform operator

Equation 2.1 can be implemented by using the following figure 2.1

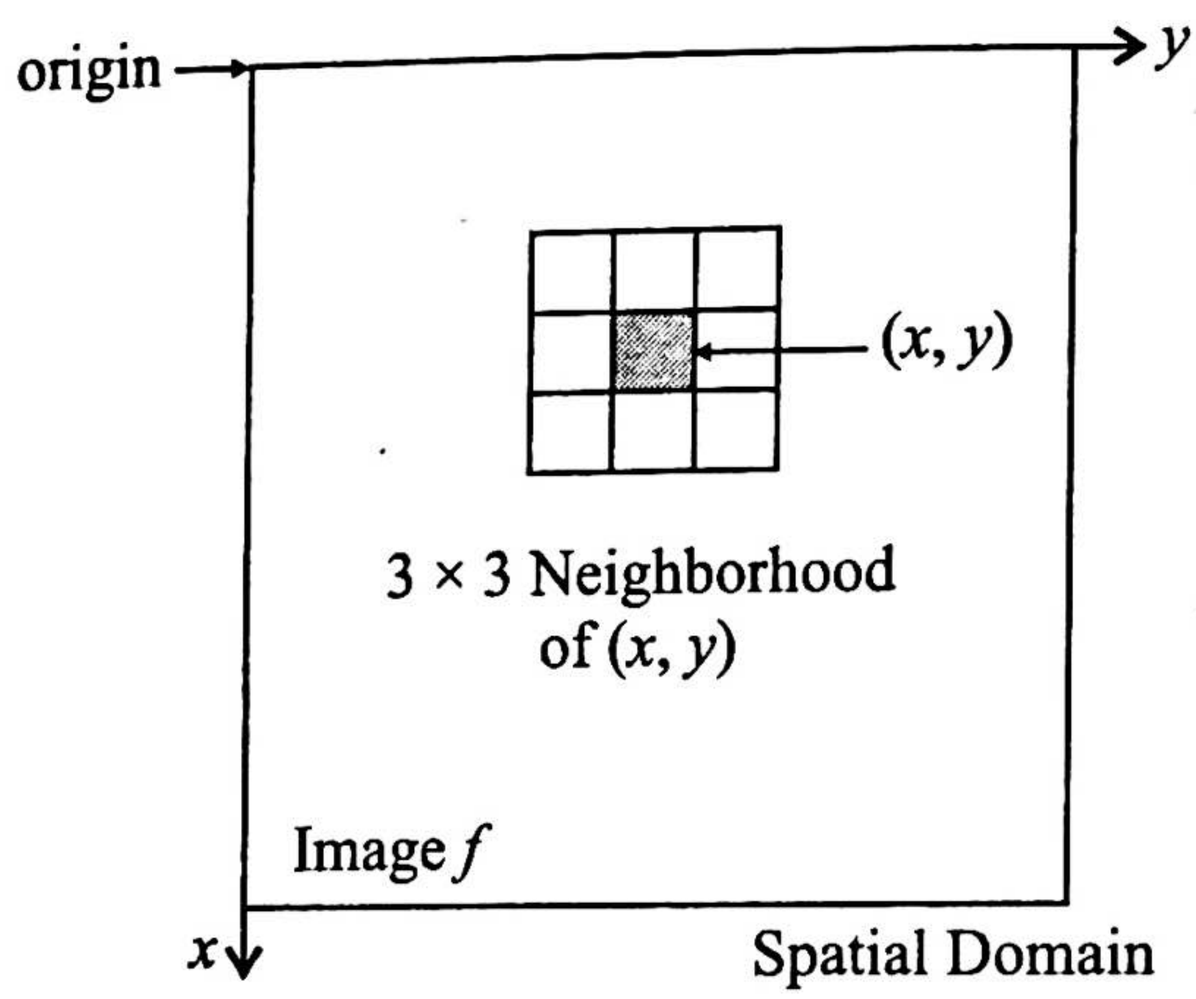


Figure 2.1 Implementation of spatial image

Point (x,y) is an arbitrary location in the image is a neighbourhood of (x,y). It shows that origin is moving from pixel to pixel and then apply the operator to get the output at that location.

For example, let the neighbourhood is a square of size 3×3 then operator T is given as "compute the average intensity of the neighbourhood."

The process starts at the top left of the input image and proceeds pixel by pixel in horizontal scan, one row at a time.

The spatial filtering is a procedure in which the neighbourhood along with a predefined operation is filtered. This process determines the nature of the filtering process.

2.2 Intensity Transformation Function

If we are considering equation (2.1) and assume that g depends only on the value of f at any single point (x, y) and T then it becomes intensity or gray-level or mapping transformation function.

It is given as

$$S = T(r) \qquad \ldots(2.2)$$

Here S and r be the variables for denoting Intensity of g and f at any point (x,y) respectively.

Let us understand this with an example in which $T(r)$ has a form as shown in figure 2.2 and produce an image of higher contrast than the original by darkening the intensity level below K and brightening the level above K.

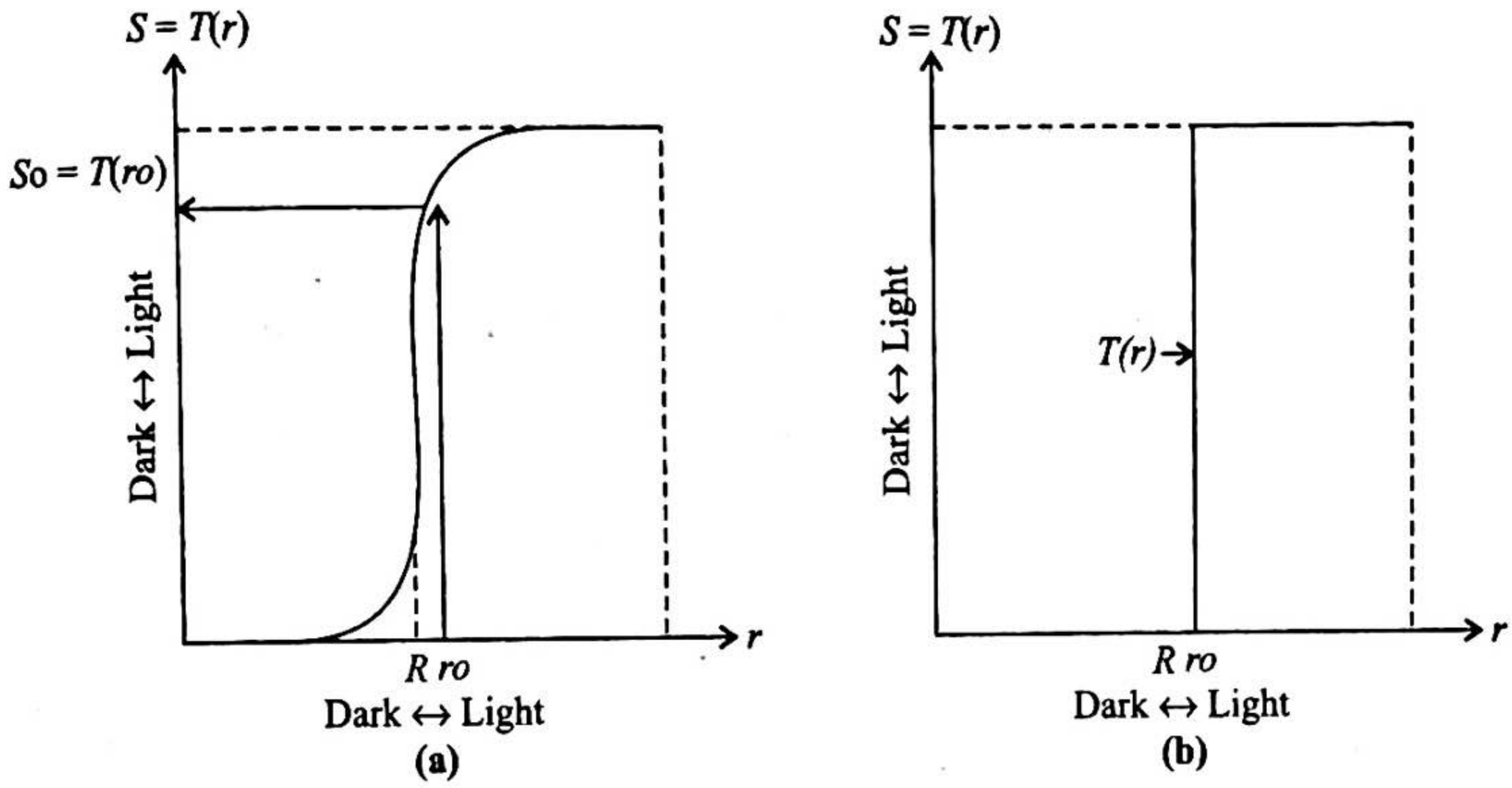


Figure 2.2 Intensity (gray-level) Transformation function

(a) Contrast stretching function (b) Thresholding function

$T(r)$ Produces a two-level binary image which is called thresholding function. While in contrast stretching function values of r lower than K are compressed towards black and opposite is true for the value of r higher than K.

In intensity Transformation there are some basic function which are frequently used in image enhancement are :

(*i*) Linear (Negative and identity transformation) function.

(*ii*) Logarithmic [Log and inverse log transformation] function.

(*iii*) Power Law (n^{th} power and n^{th} root transformation) functions.

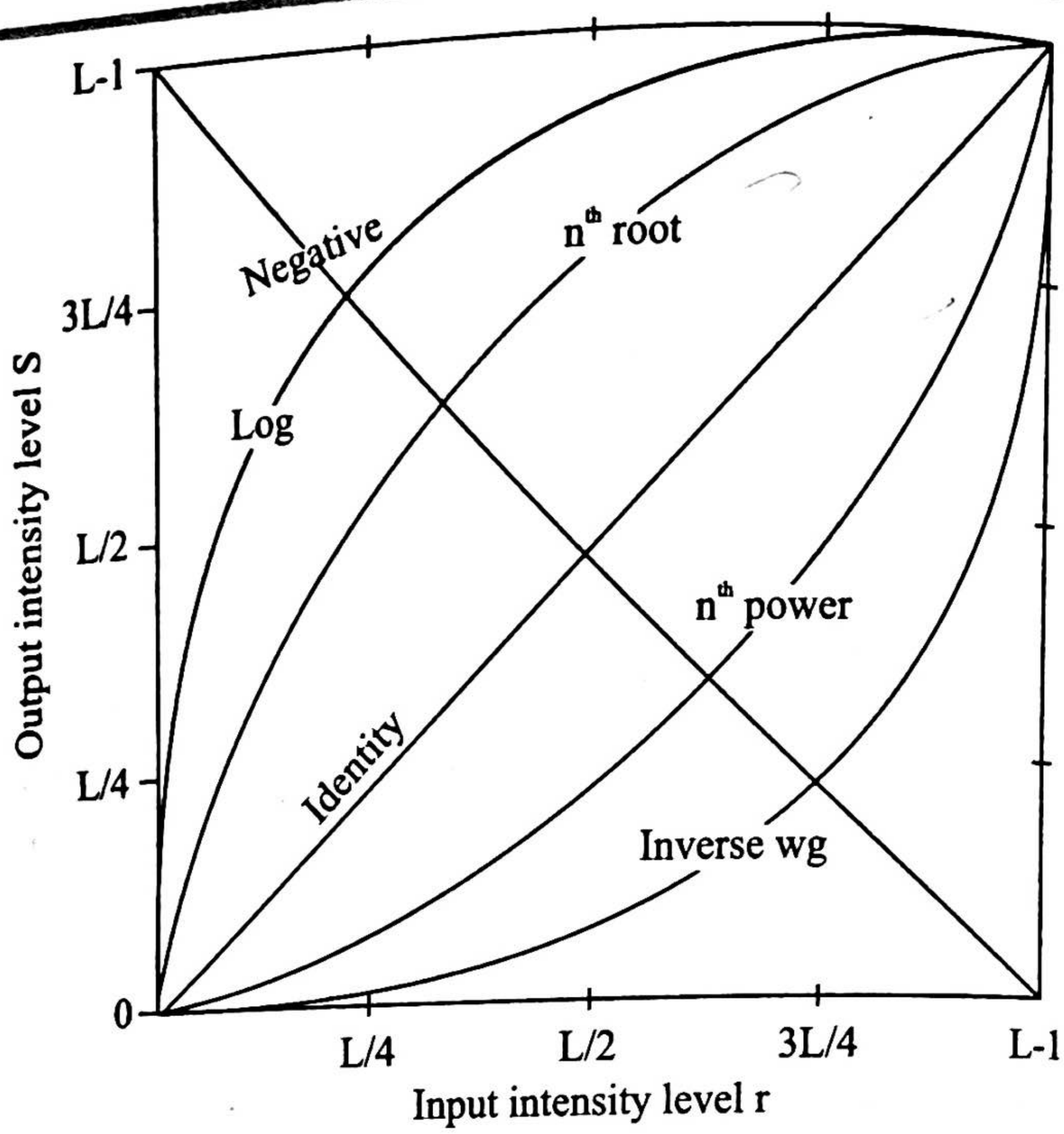


Figure 2.3 Basic Intensity Transformation functions

2.2.1 Linear or Image Negative and Identity Transformation

The image negative in the range $[0, L-1]$ can be obtained by this transformation and is given as

$$S = L - 1 - r \qquad ...(2.3)$$

This transformation is particularly suited for enhancing white or gray detail embedded in dark regions of an image, when the black areas are dominant in size.

2.2.2. Logarithmic [Log and Inverse Log Transformation]

The log transformation functions is given as

$$S = c\log(1 + r) \qquad ...(2.4)$$

Where c is constant and assume that $r \geq 0$.

The graph shows that log transformation maps a narrow range of low intensity values in the input into a wider range of output levels.

This type of transformation is suited to expand the values of dark pixels in an image while compressing the higher-level values. The opposite in also valid for inverse log transformation.

The log function has the important characteristic that it compresses the dynamic range of images with large variation in pixel values.

2.2.3 Power-Law [n^{th} Power and n^{th} Root Transformation or Gamma]

The power law transformation function is given as

$$S = cr^{\gamma} \quad \ldots(2.5)$$

or

$$S = c[r + \epsilon]^{\gamma} \quad \ldots(2.6)$$

Where c and γ are positive constants.

This type of functions maps a narrow range of dark input values into a wider range of output values and vice-versa. Most of the devices are based on power law transformation function for image capturing, printing and displaying.

Figure 2.4 shows various values of γ for $S = cr^{\gamma}$.

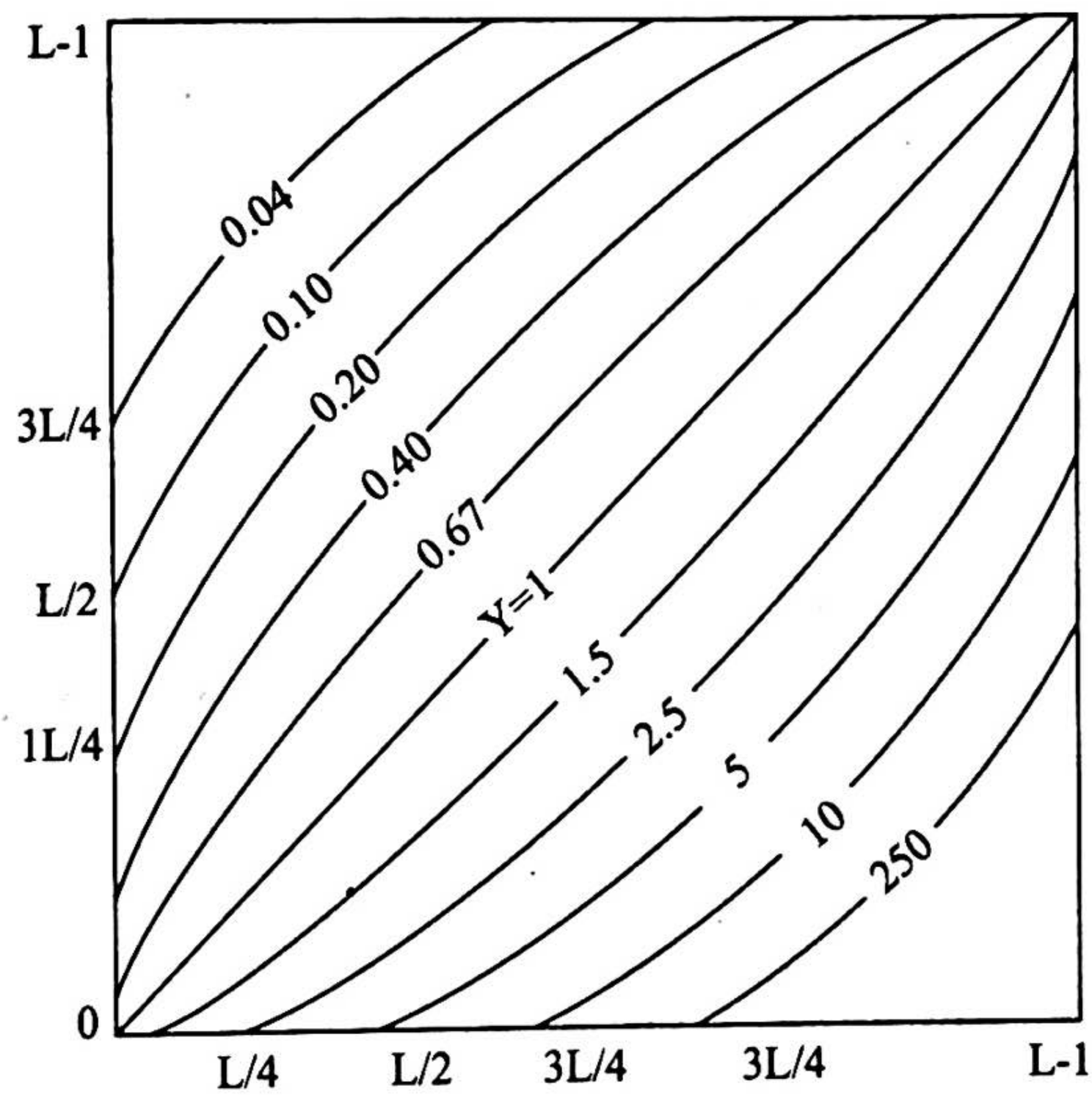


Figure 2.4 Plot of $S = cr^{\gamma}$ for various values of γ

The exponent in the power-law equations is known as gamma and the elimination/correction process of power-law response is called gamma correction. This gamma correction is important if displaying an image accurately on a computer screen is of concern.

2.3 Histogram

Histogram is a discrete function of any digital image having a intensity level in [0, L – 1]. It is given as

$$h(r_K) = n_K \qquad \ldots(2.7)$$

Where r_K is the kth intensity value and n_K is the number of pixels is the image having intensity r_K.

The normalized histogram is given by

$$P(r_R) = \frac{n_R}{MN} \qquad \text{for } K = 0, 1, 2 \ldots L\text{-}1$$

and M and N are the row and column dimension of the image and the sum of all components of a normalized histogram is equal to 1. This histogram manipulation is used for image enhancement.

The histogram may be viewed graphically simply as plots of $h(r_K) = n_K$ versus r_K or.

or $$P(r_K) = \frac{n_K}{MN} \text{ versus } r_K \qquad \ldots(2.8)$$

An image with low contrast has a narrow histogram located towards the middle of the intensity scale. While the component of the histogram in the high contrast image covers a wide range of intensity scale.

2.3.1 Histogram Equalization

Let r be the variable denoting the intensity of an image in the range [0, $L - 1$]. Where $r = 0$ is for black and $r = (L - 1)$ is for white.

Intensity mapping

$$S = T(r) \quad 0 \le r \le L - 1 \qquad \ldots(2.9)$$

Some assumption are considered to produce an output intensity level S. Which is shown in figure 2.5.

(*i*) $T(r)$ is strictly monotonically increasing function in the interval $0 \le r \le L - 1$, and

(*ii*) $0 \le T(r) \le (L - 1)$ for $0 \le r \le (L - 1)$

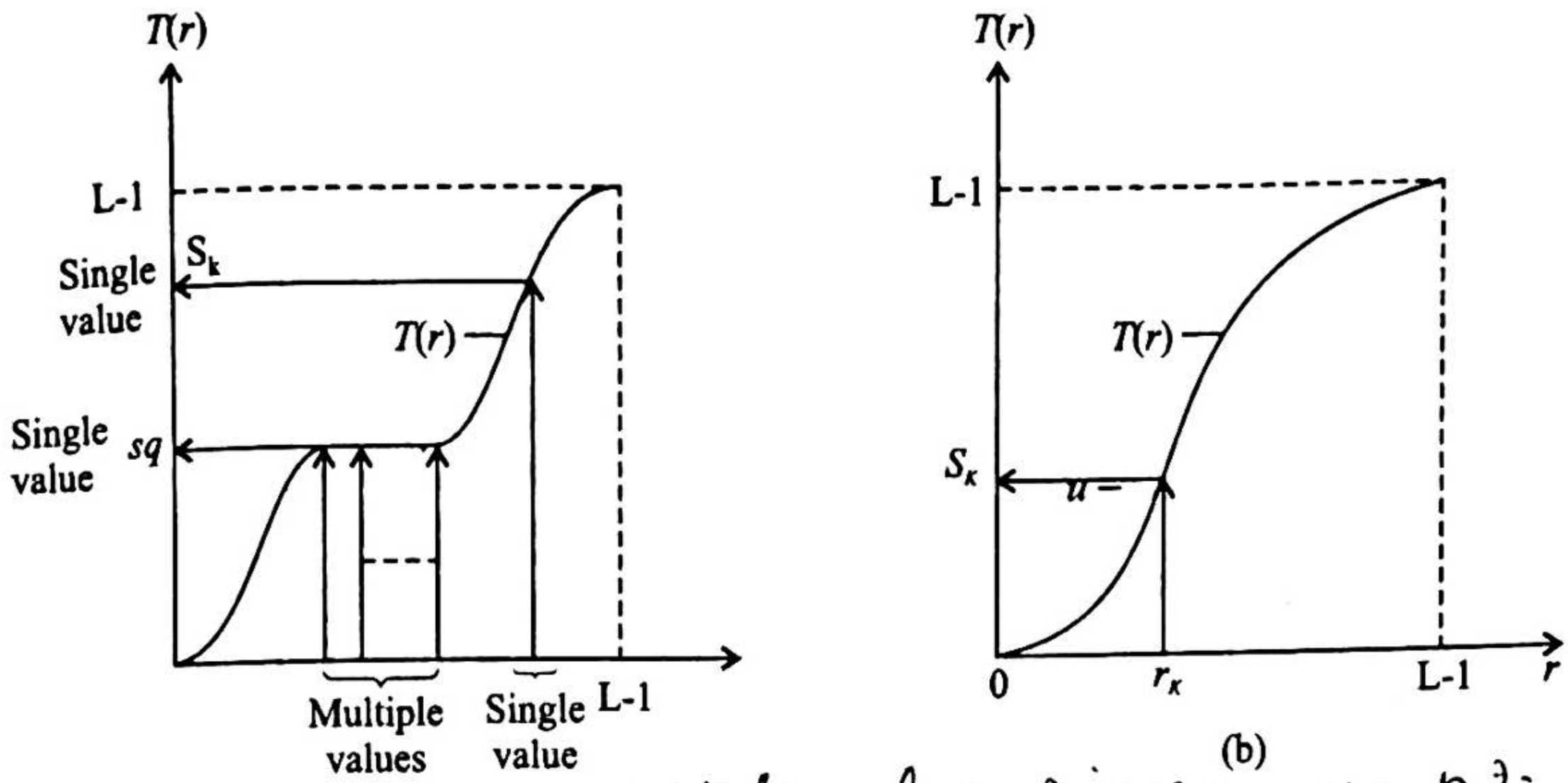
Some time we can use its inverse

$$r = T^{-1}(S) \qquad 0 \le S \le L - 1 \qquad \ldots(2.10)$$

for this assumption (*i*) can be charged as

[0, L-1]

(*i*) $T(r)$ is strictly monotonically increasing function in the interval $0 \leq r \leq (L-1)$.



→ multiple value given are p.f. so it leads in bad image

Figure 2.5 (a) Monotonically function representing multiple value to map single value, (b) Strictly monotonically function mapping in both ways

Figure 2.5 (a) & (b) satisfy both the assumptions and implies that monotonic transformation function performs one-to-one or many-to-one mapping.

Let $Pr(r)$ and $Ps(s)$ be the probability density functions of r and s where probability density function (PDF) is a fundamental descriptor of random variable from the basic probability theory if $Pr(r)$ and $T(r)$ are known, and $T(r)$ is continuous and differentiable then the PDF of transformed variable S can be obtained as :

$$Ps(S) = Pr(r)\left|\frac{dr}{ds}\right| \qquad \ldots(2.11)$$

It shows that PDF of output intensity variable is obtained by the PDF of input intensity and the transformation function used.

This transformation function has the form

$$S = T(r) = (L-1)\int_0^r p_r(w)dw \qquad \ldots(2.12)$$

Where w is a dummy variable of integration, and the right side of this equation is known as cumulative distribution function (CDF) of random variable r.

from the Leibniz's basic calculus the derivative of a definite integral with respect to its upper limit is the integrated evaluated at the limit.

i.e.

$$\frac{ds}{dr} = \frac{dt(r)}{dr}$$

$$\frac{ds}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r Pr(w)dw\right] \quad \ldots(2.13)$$

To find $P_s(S)$ corresponding to the transformation function, substitute $\frac{dr}{ds}$ from equation (2.13) to equation (2.11).

i.e.

$$P_s(s) = P_r(r)\left|\frac{dr}{ds}\right|$$

$$P_s(s) = P_r(r)\left|\frac{1}{(L-1)P_r(r)}\right| \quad \ldots(2.14)$$

$$P_s(s) = \frac{1}{(L-1)} \quad 0 \le S \le (L-1) \quad \ldots(2.15)$$

This is known as uniform probability density function and implies that $P_s(s)$ always is uniform, independent of the form of $Pr(r)$.

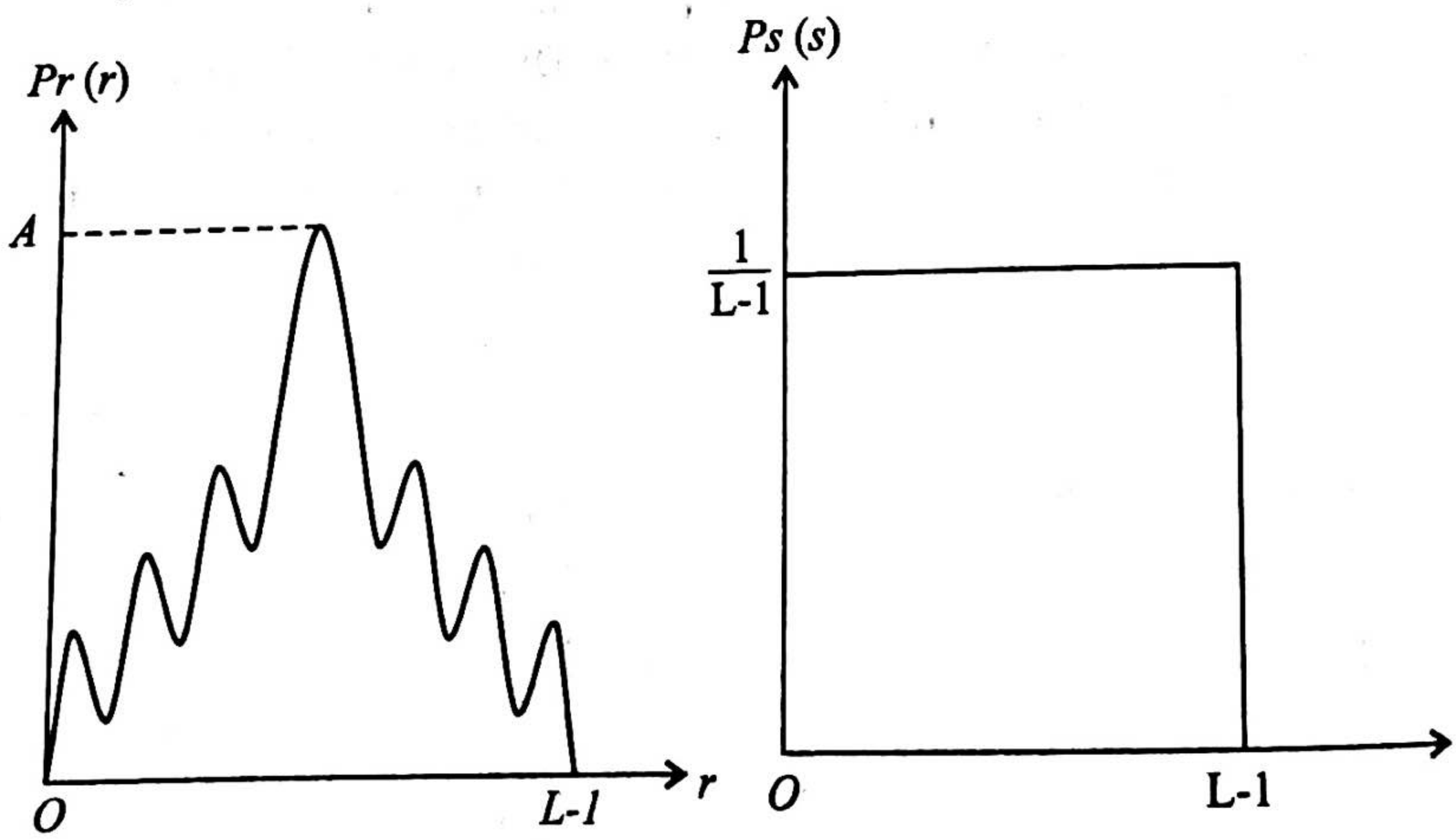


Figure 2.6 (a) Arbitrary PDF
(b) Result of equation (2.12) for all intensity values

For discrete values we deal with histogram values and summations instead of probability density functions and integrals. The probability

of occurrence of intensity level r_k in a digital image is given as

$$P_r(r_k) = \frac{n_K}{MN}, \quad K = 0, 1, 2 \ldots L-1 \qquad \ldots(2.16)$$

Here MN is the total number of pixels in the image, n_R is the number of pixels that have intensity r_K and L is the possible intensity levels in the image, since histogram is the plot of $P_r(r_K)$ versus r_K so that.

$$S_K = T(r_K) = (L-1)\sum_{J=0}^{R} Pr(r_j)$$

$$S_K = \frac{(L-1)}{MN}\sum_{J=0}^{R} n_j \quad K = 0, 1, 2\ (L-1) \qquad \ldots(2.17)$$

The mapped or transformed $T(r_k)$ is known as histogram equalizations or histogram linearization transformation. Which is obtained by mapping each pixel in the input image with intensity r_K into a corresponding pixel with level s_K in the output image.

2.3.2 Histogram Matching

Histogram equalization determines a transformation function that seeks to produce an output image. It is useful to specify the shape of the histogram. So the method to generate a processed image which has a specified histogram is called histogram matching on histogram specification.

Histogram specification or matching process includes following procedure:

(*i*) Compute the histogram $P_r(r)$ of the given image then calculate histogram equalization transformation in the range $[0, L-1]$.

(*ii*) Compute all values of transformation function and obtained $P_z(z_i)$ for value of specified histogram, store these values.

(*iii*) Value obtained in step 1 and used with values obtained in step 2 to find corresponding value of z_q.

(iv) Form the histogram-specified image with histogram equalization.

2.3.3 Local Histogram Processing

The histogram processing are easily adapted to local enhancement. In which first of all, define a neighbourhood and move its centre from pixel to pixel. The histogram equalization or histogram specification is obtained at each location in the neighbourhood. Which is further used

to map the intensity of the pixel. It have an advantage over repeatedly computing the histogram but it produces an undesirable "blocky" effect.

2.3.4 Histogram Statistics for Image Enhancement

The statistic obtained from an image histogram are used for image enhancement. Let r denote a discrete random variable in the range $(0, L-1)$ and $P(r_i)$ be the normalized histogram corresponding to r_i.

The nth moment of r about its mean is

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n P(r_i) \qquad \ldots(2.18)$$

Here m is the mean average intensity value of r and

$$m = \sum_{i=0}^{L-1} r_i \, P(r_i) \qquad \ldots(2.19)$$

It is observed that second moment is particularly important having

$$\mu_2(r) = \sum_{i=0}^{L-1} (ri - m)^2 P(ri) \qquad \ldots(2.20)$$

It is given as intensity variance and denoted by σ^2.

We can directly estimate the mean & variance without computing the histogram. These estimations are known as sample mean and sample variance and we have sample mean.

$$m = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{Y=0}^{N-1} f(x,y) \qquad \ldots(2.21)$$

and sample variance

$$\sigma^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{Y=0}^{N-1} [f(x,y) - m]^2 \qquad \ldots(2.22)$$

It gives that mean intensity of an image can be given by summing the values of all its pixels and dividing the sum by total number of pixels in the image.

Let (x, y) be the coordinate of any pixel in an image and let S_{xy} be the neighbourhood of specified size. Then mean values of the pixel in the neighbourhood is :

$$mS_{xy} = \sum_{i=0}^{L-1} r_i \, ps_{xy}(r_i) \qquad \ldots(2.23)$$

Where PS_{xy} the histogram of the pixel in S_{xy} and the variance of the pixels in the neighbourhood is

$$\sigma^2 S_{xy} = \sum_{i=0}^{L-1} (r_i - ms_{xy})^2 PS_{xy}(ri) \quad \ldots(2.24)$$

This gives that local mean is a measure of average intensity in neighbour S_{xy} and local variance is a measure of intensity contrast in that neighbourhood.

So we can say that image enhancement can be obtained by using global histogram equalization.

2.4 Spatial Filtering

Spatial filtering is one of the most important tool used for broad spectrum of applications. The term filtering represents the accepting or rejecting certain frequency components. For example a low pass filtering in used to blur or smooth an image, any such smoothing can be accomplished by spatial filtering. Which is also known as spatial mask, kernels, templates and window.

2.4.1 Mechanism Used is Spatial Filtering

Since, a spatial filter consists of a neighbourhood, generally a rectangle and a predefined operation, which is performed operation, which is performed on the image pixels by the neighbourhood.

This filtering creates a new pixel having coordinates equal to the coordinates of the center of the neighbourhood. This processed or filtered image visits each pixel in the input image. Depending on the operation performed this filtering can be defined. If the linear operation is performed on the image, then the filter in known as linear spatial filter else it is termed as nonlinear.

Here we emphasis on linear spatial filters.

Let a linear spatial filtering uses a 3 × 3 neighbourhood. The response $g(x,y)$ of the filter is the sum of product of the filter coefficient and the image pixels enclosed by the filter i.e.

$$g(x,y) = w(-1,-1)\, f(x-1,y-1) + w(-1,0)\, f(x-1, y) + w(0,0)\, f(x,y) + \ldots\, w(1,1)\, f(x+1,y+1) \quad \ldots(2.25)$$

It have been observed that centre coefficient of the filter $w(0, 0)$ align with the pixel at (x,y).

So any linear spatial filtering of an image of size *MXN* with a filter of size *MXN* is given as

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t) \qquad \ldots(2.26)$$

and we have x and y are varies so that each pixel in w visiting y every pixel of f.

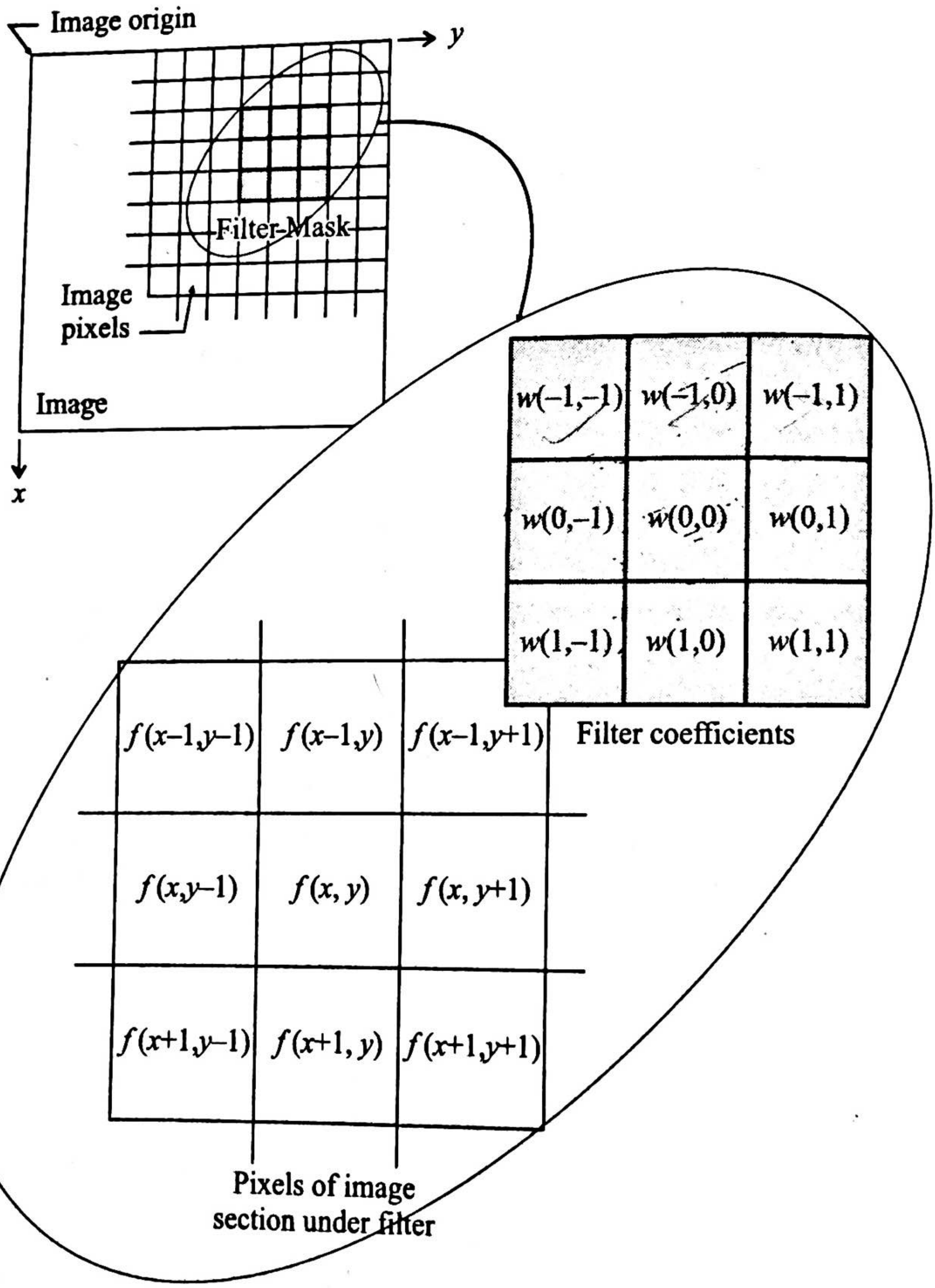


Figure 2.7 Linear spatial filtering

2.5 Spatial Correlation and Convolution

Two basic concepts are used to perform linear spatial filtering.

Correlation is process of moving any filter mask over the image and then computing the sum of product at each location. It is a function of displacement of the filter.

Convolution is similar process to correlation in which filter is first rotated by 180°. If we pre-rotate the filter and perform the same sliding sum of product operation then we can get the desired result.

2.5.1 Correlation Operation

origin f w

↓

0 0 0 1 0 0 0 0 1 2 3 2 8 (a)

↓

0 0 0 1 0 0 0 0 (b)

1 2 3 2 8

↑ starting position

zero padding

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (c)

1 2 3 2 8

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (d)

1 2 3 2 8

↑ one shift position

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 (e)

1 2 3 2 8

↑ two shift position

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (f)

1 2 3 2 8

2328 final position

0 0 0 8 2 3 2 1 0 0 0 0 (g)

full correlated result

0 8 2 3 2 1 0 0 (h)

cropped correlated result

2.5.2 Convolution Operation

origin f w rotated 180

↓

0 0 0 1 0 0 0 0 8 2 3 1 (i)

0 0 0 1 0 0 0 0 starting position (j)

8 2 3 2 1

zero padding

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)

8 2 3 2 1

One shift 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (*l*)

8 2 3 2 1

Two shift 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)

8 2 3 2 1

Final position 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)

8 2 3 2 1

Full convolution result 0 0 0 1 2 3 2 8 0 0 0 0 (o)

Cropped convolution result 0 1 2 3 2 8 0 0 (p)

The correlation of a filter $w(x,y)$ of size $m \times n$ with an image $f(x,y)$ is given as

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \qquad \ldots(2.27)$$

It can evaluate all values of displacement variables x, and y.

Similarly, convolution of $w(x,y)$ and $f(x,y)$ can be given as

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t) \qquad \ldots(2.28)$$

Negative sign is due to rotation by 180°.

2.6 Vector Representation of Linear Filtering

The characteristic response R of a mask either for convolution or correlation can be write as sum of product.

i.e. $$R = w_1 z_1 + w_2 z_2 + w_3 z_3 + ...w_{mn} z_{mn}$$

or $$R = \sum_{k=1}^{mn} w_k z_k \quad ...(2.29)$$

or $$R = w^T z \quad ...(2.30)$$

Here W_s are the coefficients of $m \times n$ filter and z_s are the corresponding image intensities.

Equation 2.30 can be used for correlation and for convolution we simply rotate the mask by 180°. for example let a 3 × 3 mask with coefficients labelled is given as:

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figure 2.8 Representation of 3 × 3 filter mask

So $$R = w_1 z_1 + w_2 z_2 + ...w_9 z_9$$

or $$R = \sum_{k=1}^{9} w_k z_k \quad ...(2.31)$$

or $$R = w^T z$$

Here w and z are 9-D vector.

2.7 Generation of Spatial Filter Masks

To generate an $m \times n$ linear spatial filter it is required that we have to specify mn mask coefficients. These can be selected on the basis of what the filter is supposed to do.

For example, if we replace the pixels in an image by the average intensity of a 3 × 3 neighbourhood centred on those pixels. Let z_i, i = 1, 2, 3, ...9 denotes the intensities then the average is

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

Which is similar to equation 2.31 having coefficient values $w_i = \frac{1}{9}$. So, it is concluded that an linear filtering operation with a 3 × 3 mask having coefficient equal to $\frac{1}{9}$ can be used to implement the desired averaging.

Some other functions like Gaussian function of two variables can also be used to generate spatial filter mask.

i.e., gaussian function $h(x,y) = e\frac{-x^2+y^2}{2\sigma^2}$ ...(2.32)

Where σ is the standard deviation. For generating a non-linear filter it is required that the size of neighbourhood and the operations which are to be performed on the image pixel contained is the neighbourhood.

2.7.1 Smoothing Spatial Filters

Blurring and noise reduction can be accomplished in image processing by use of smoothing spatial filters. It may be or linear of non-linear filter.

The blurring can be used in preprocessing tasks such as removing of some details from an image prior to image extraction, bridging a small gap in linear or curves etc. The noise reduction can be achieved by blurring the image using any of the spatial filtering operation.

2.7.2 Smoothing Linear Spatial Filter

The response of smoothing linear spatial filter is defined as the average of the pixels contained in the neighbourhood of the filter mask. It is also known as averaging filters.

It can be achieved by replacing the value of every pixel in an image by the average of the intensity levels in the neighbourhood explained by the filter mask. The processed image is obtained as reduced sharp transition is intensities. However, edges are characterized by sharp intensity transitions. So, with this operation the edges become blurred.

Let we have an example of 3 × 3 smoothing filters in which all the coefficients of the filters are is.?

So at the end of filtering process the entire image is divided by 9.

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

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

Figure 2.9 3 × 3 smoothing filter mask

So, an spatial averaging filter in which all the coefficients are equal is known as a "box filter".

Let an another example of smoothing linear spatial filtering is having a 3 × 3 filter mask.

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

Figure 2.10 3 × 3 Smoothing linear filter mask

This type of filter is also known as "weighted average" mask. In this example, the center coefficient is 4 and having the multiple value of highest coefficient. The reason behind weighing the center point the highest and then reducing the values of the coefficient as a function of increasing distance from the origin. It can reduce blurring in the smoothing process.

In general, the implementation for filtering an M × N image with weighted average filter of size M × N is given as

$$g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{s=-b}^{b} w(s,t)} \qquad \ldots(2.33)$$

Here $x = 0, 1, 2 \ldots\ldots\ldots\ldots M-1$

and $y = 0, 1, 2 \ldots\ldots\ldots\ldots N-1$

The denominator in this equation is the sum of the mask coefficients.

2.7.3 Smoothening Non-Linear [Order-Statistic] Filters

Order-statistic filters are termed as non-linear filter and its response is based on ordering of the pixels containing the image area enclosed by the filter. It is then replacing the value of the centre pixel with the resulted

ranking value. Mostly median filter is used for such category. Such filters provide excellent noise reduction capabilities and comparatively less blurred images than linear smoothing filter of same size. Such types of filters are very effective in the presence of impulse noise also known as salt and pepper noise due to its appearance as white and black dots superimposed on an image.

For example let a 3 × 3 neighbourhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). To get the median filtering first sort them as (10, 15, 20, 20, 20, 20, 25, 100) which gives medium of 20. The isolated clusters are eliminated by an m × n median filter. Elimination means forced to the median intensity of the neighbours. The median represents the 50th percentile of a ranked set of numbers.

For example the 100th percentile ranked is called "Max Filter" which is used for finding the brightest point in an image. While the 0th percentile is minimum filter".

2.8 Sharpening Spatial filters

Sharpening can be done for highlight transition in intensity. It is accomplished by spatial differentiation. There are various ways for defining and implementing operators for sharpening. In general the strength of the response of a derivative operator is proportional to the degree of intensity discontinuity of that image at which the operator is applied.

So we can say that image differentiation enhance edges and noise and de-emphasizes area with slow varying intensities.

2.8.1 Basics of Sharpening

To understand the basic concept various derivatives are used. It is defined in terms of differences.

For a first derivative

(*i*) It must be zero in areas of constant intensity.

(*ii*) It must be none-zero at the onset of an intensity step or ramp and

(*iii*) It must be none-zero along ramps.

Similarly for a second order derivative

(*i*) It must be zero in constant areas.

(*ii*) It must be none-zero at the onset and end of intensity step.

(*iii*) It must be none-zero along ramps of constant slope.

A basic definition of first order derivatives for 1-D function is

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \qquad \ldots(2.34)$$

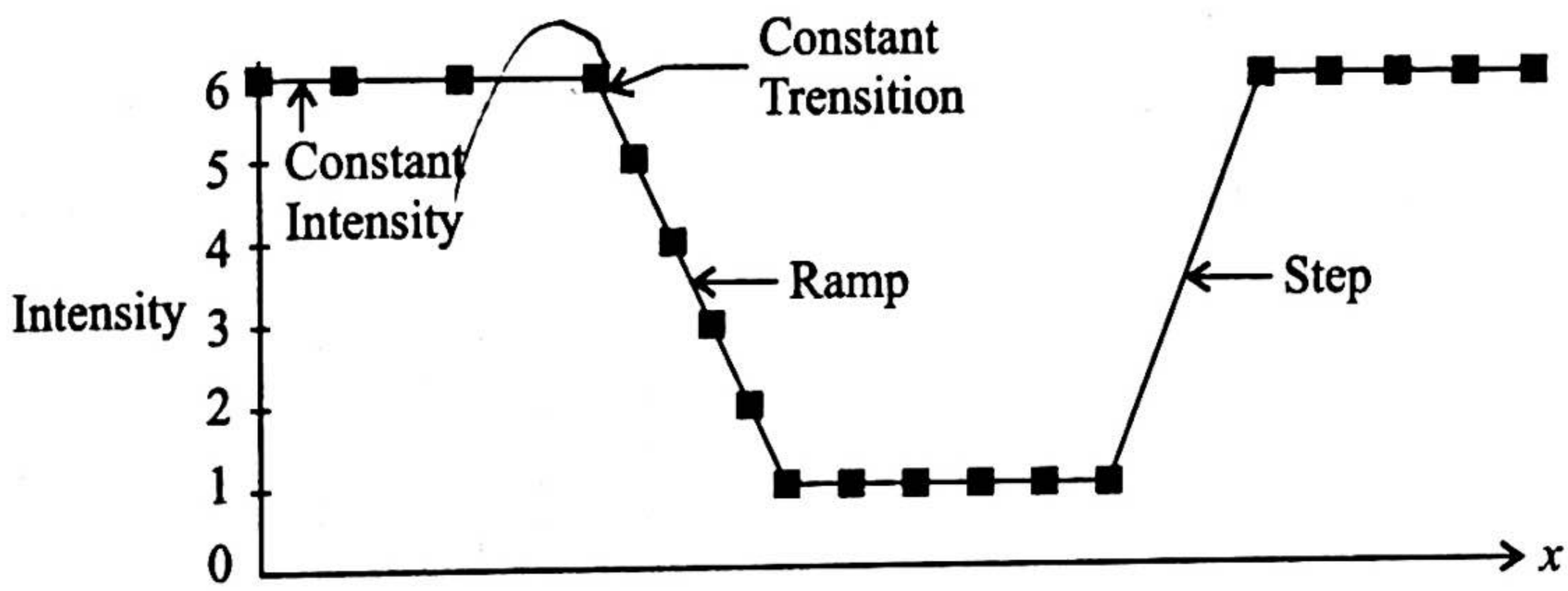
$$\frac{\partial f}{\partial x} = \frac{df}{dx}$$

when there are only one variable in the function

So the second order derivative of $f(x)$ is

$$\frac{d^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x) \qquad \ldots(2.35)$$

Figure 2.11 show first and second order derivative of a 1-D digital function representing a section of a horizontal intensity profile.



6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
0 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
0 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0

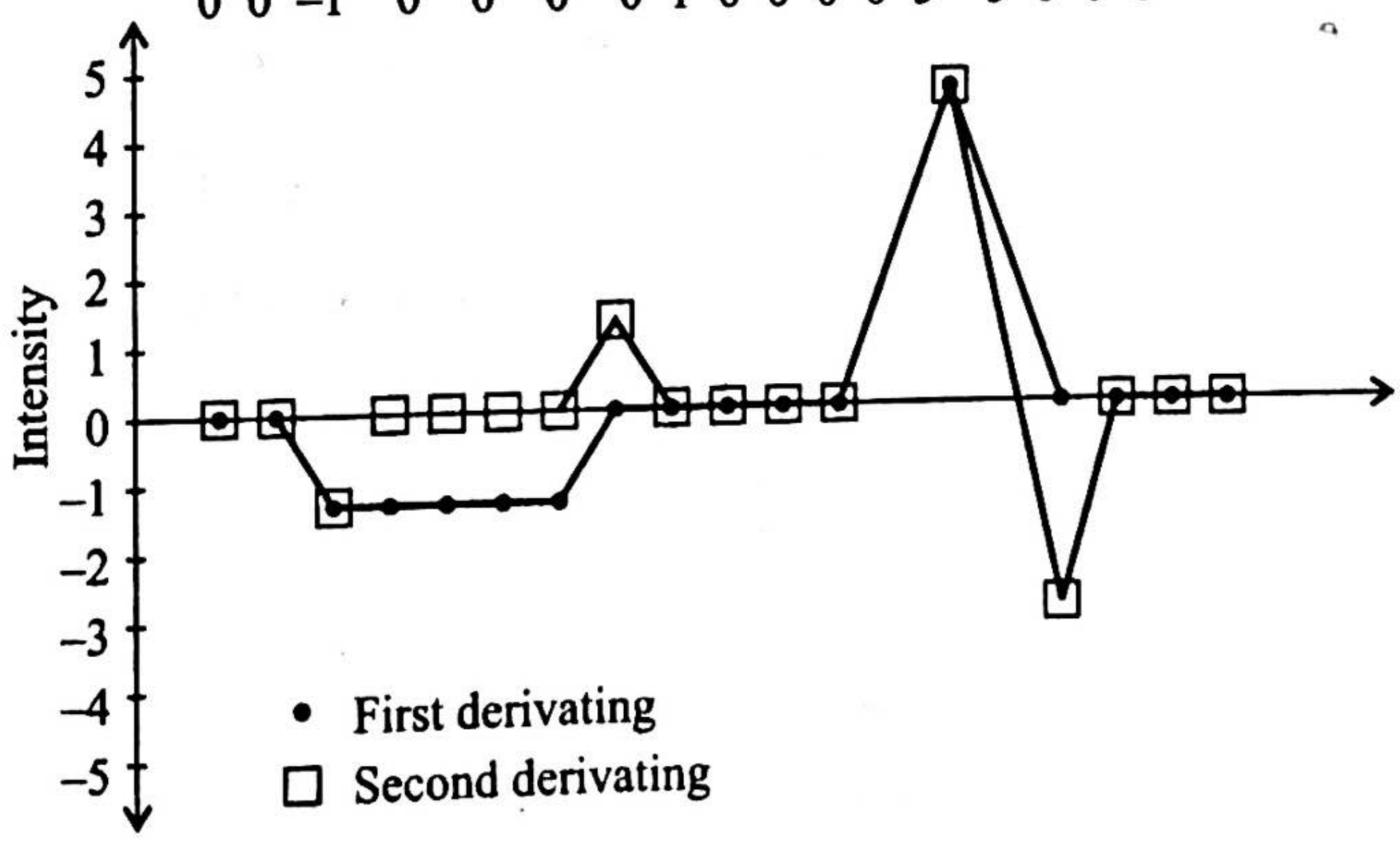


Figure 2.11 First & second derivatives (a), (b) and (c)

6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
0 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
0 0 −1 0 0 0 0 1 0 0 0 0 0 5 −5 0 0 0

From the above figure when we traverse the profile from left to right we have an area of constant intensity for both derivatives having zero values. After that we encountered an intensity ramp followed by a step and we note that the first order derivative is non-zero at the on set of the ramp and the step and the second order derivatives is non-zero at the onset and ends of both the ramp & the step. Finally, to get the first derivative is non-zero and the second is zero along the ramp. It is to be note down that the sign of the second derivative changes at the on set and end of a step or ramp. So in a step transition a line joining these two values crosses the horizontal axis midway between the two extremes. With this zero crossing property we can define the edges of the image.

2.8.2 First Order Derivative for Image Sharpening [Non-Linear Filtering]

The first order derivative can be implemented by magnitude of gradient. Let a function $f(x,y)$ is the gradient of f at (x,y) so the gradient for two-dimensional column vector is

$$\nabla f = grad\,(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} \qquad \ldots(2.36)$$

This vector point in the direction of the greatest rate of change of f at location (x,y).

The magnitude of vector ∇f is given as :

$$M(x,y) = \text{Mag. } (\nabla f) = \sqrt{(g_x)^2 + (g_y)^2} \qquad (2.37)$$

This magnitude is of the same size as the original image. So we simply implies this image as gradient image.

These components of gradient are linear operators while the magnitude of this vector is not. In general

$$M(x,y) \simeq |g_x| + |g_y| \qquad \ldots(2.38)$$

In case of laplacian operation consider a 3 × 3 region of image having z_s as the intensity values.

(a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(b)

–1	0
0	1

(c)

0	–1
1	0

(d)

–1	–2	–1
0	0	0
1	2	1

(e)

–1	0	1
–2	0	2
–1	0	1

Figure 2.12 Illustration of sharpening image using first derivative

Let the center point z_5 Gives $f(x,y)$ at any location (x,y), z_1 gives $f(x-1, y-1)$ and so on. So, the first order derivative gives $g_x = (z_8 - z_5)$ and $g_y = (z_6 - z_5)$.

The Roberts in 1965 proposed cross differences which are known as roberts cross gradient operators.

Since $$g_x = (z_9 - z_5) \quad \ldots(2.39)$$

and $$g_y = (z_8 - z_6)$$

So the magnitude of these

$$M(x,y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{1/2} \quad \ldots(2.40)$$

or $$M(x,y) \simeq |z_9 - z_5| + |z_8 - z_6| \quad \ldots(2.41)$$

The smallest mask can be implemented with z_5 as centered and

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \ldots(2.42)$$

and $$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \ldots(2.43)$$

Above these equations can be implemented using the mask of figure mentioned earlier and the magnitude of the mask is given as

$$M(x,y) \simeq |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \; |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \quad \ldots(2.44)$$

In the sobel operator, roberts cross gradient operator the sum of all the coefficients is equal to zero which indicates that they give a response of zero in an area of constant intensity.

The g_x and g_y can be implemented as a sum of product using spatial mask. This nonlinear aspect of image sharpening with gradient is the computation of $M(x,y)$ involvers squaring and square roots which are non-linear operations.

2.8.3 Second Order Derivative for Image Sharpening : (The Laplacian Operator)

The laplacian operator is used for image sharpening as a second order derivative of spatial linear filtering. It consists of a discrete formulation of the second-order derivative and then construction a filter mask based on that formulation.

Here Isotropic filters are preferred which are rotational invariant. It means if we are rotating the image and then applying the filter will gives the same result as applying the filter to that image first and then rotating the result.

The laplacian operator for a function $f(x,y)$ is given as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad \text{...(2.45)}$$

Since derivative of any order are linear so the laplacian is a linear operator. Using equation (2.35) in x-direction we have

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y) \qquad \text{...(2.46)}$$

and in y-direction

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y) \qquad \text{...(2.47)}$$

From these two equations the discrete laplacian of two variables is

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \qquad \text{...(2.48)}$$

Equation 2.48 can be implemented using filter mask as shown in below figure 2.13 It gives an isotropic resultant values for rotation in increments of 90°.

(a)

0	1	0
1	–4	1
0	1	0

(b)

1	1	1
1	–8	1
1	1	1

(c)

0	–1	0
–1	4	–1
0	–1	0

(d)

–1	–1	–1
–1	8	–1
–1	–1	–1

Figure 2.13 Illustration of laplacian operator

Each diagonal term contains a $-2f(x,y)$, the total subtracted from the difference term would be $-8f(x,y)$. Since the laplacian is a derivative operator so it gives intensity discontinuities in an image. It provides greyish edge lines images and some discontinuities which are superimposed on dark background.

Depending on the definition used for laplacian operator the add and subtract is defined. If it use negative center coefficient then we subtract other wise we add this image to obtain the resultant sharpened image.

So in general the laplacian operator for image sharpening is

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right] \qquad \ldots(2.49)$$

Here $f(x,y)$ is an input image and $g(x,y)$ is resultant sharpened image.

2.9 Image Smoothing using Frequency Domain Filters

Smoothing is achieved in frequency domain by high-frequency attenuation i.e. by lowpass filtering. various low pass filtering are performed among them ideal, butterworth and gaussian are to be discussed. Ideal filtering is used for very sharp while Gaussian is for very smoothing. butterworth filtering is used for ordering any filter for higher order values butterworth is ideal while for lower order values butterworth is Gaussian filter.

2.9.1 Ideal Low Pass Filtering

An ideal low pass filter is defined as a filter that "Passes" all the frequencies within a circle of radius D_o from the origin without attenuation and "cut off" all the frequencies outside this circle with some attenuation.

i.e.
$$H(u,v) = \begin{cases} 1 & for \quad D(u,v) \leq Do \\ 0 & for \quad D(u,v) > Do \end{cases} \qquad \ldots(2.50)$$

Here D_o is a positive constant and $D(u,v)$ is the distance between a point (u,v) in given frequency domain and center of the frequency rectangle.

i.e.
$$D(u,v) = \left[\left(\frac{u-p}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2\right]^{1/2} \qquad \ldots(2.51)$$

Here P and Q are padded sizes.

Figure 2.14 shows the plot of $H(u,v)$ with its filtered image. The resultant is radially symmetric about the origin. When we rotate this cross section filtered by 360° then it gives 2-D image of the filter.

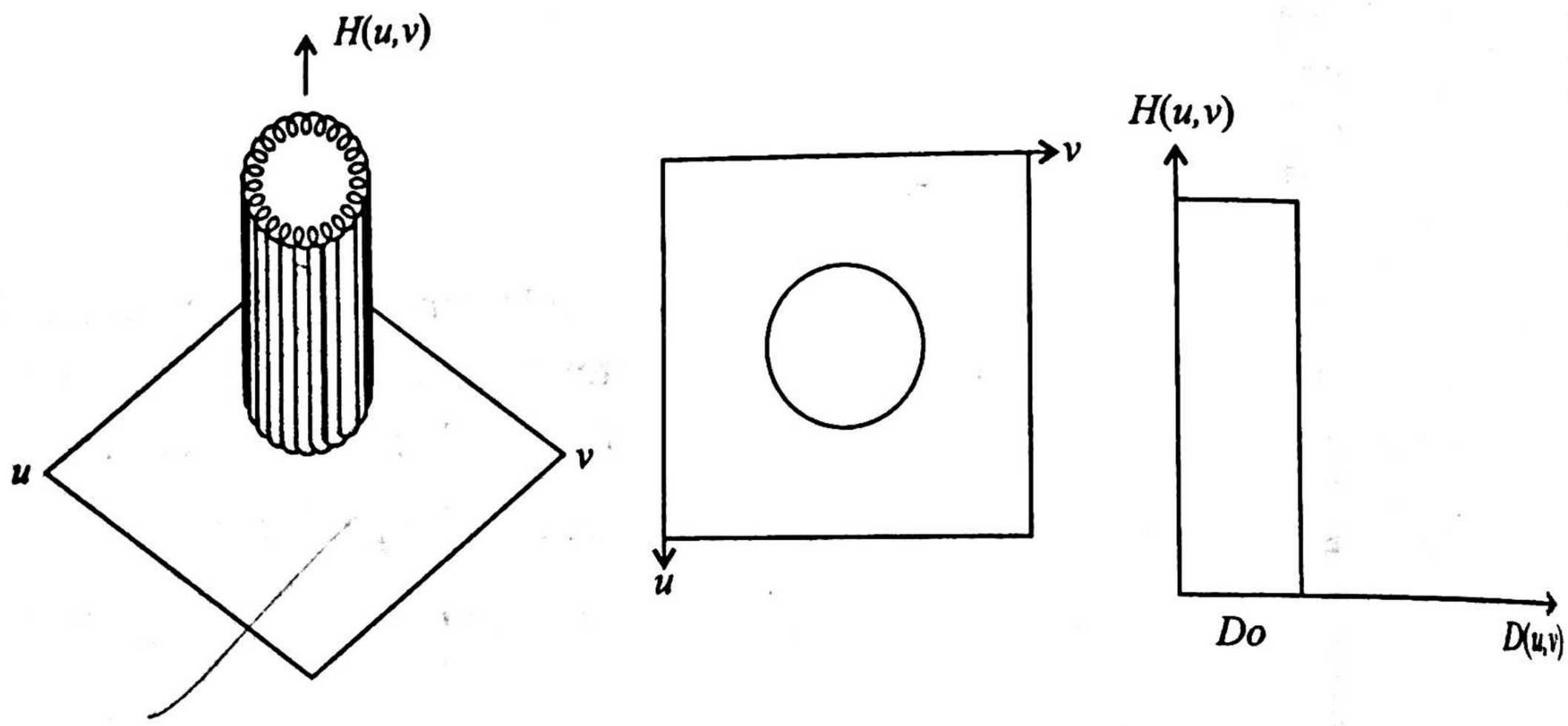


Figure 2.14 Plot of ideal low pass filter, its displayed image and its radial cross section rotational by 360°

In any ideal LPF, the point of transition between $H(u,v) = 1$ and $H(u,v) = 0$ is called the "cut off frequency." Here D_o is assumed to be cut off frequency.

If we are considering total image power P_T for each point (u,v) where $u = 0, 1, 2.... P\text{-}1$ and $v = 0, 1, 2 .....Q\text{-}1$ then

$$P_T = \sum_{u=0}^{P-1} \sum_{v=0}^{Q-1} P(u,v) \qquad \ldots(2.52)$$

Where
$$P(u,v) = |F(u,v)|^2 \qquad \ldots(2.53)$$

or
$$P(u,v) = R^2(u,v) + I^2(u,v) \qquad \ldots(2.54)$$

$F(u,v)$ is the DFT of $F(x,y)$ is a complex function and R & I be the real and imaginary part of $F(u,v)$ respectively.

The blurring and ringing properties of ILPF can be explained by convolution property. The spatial domain and intensity profile is explained is figure 2.15.

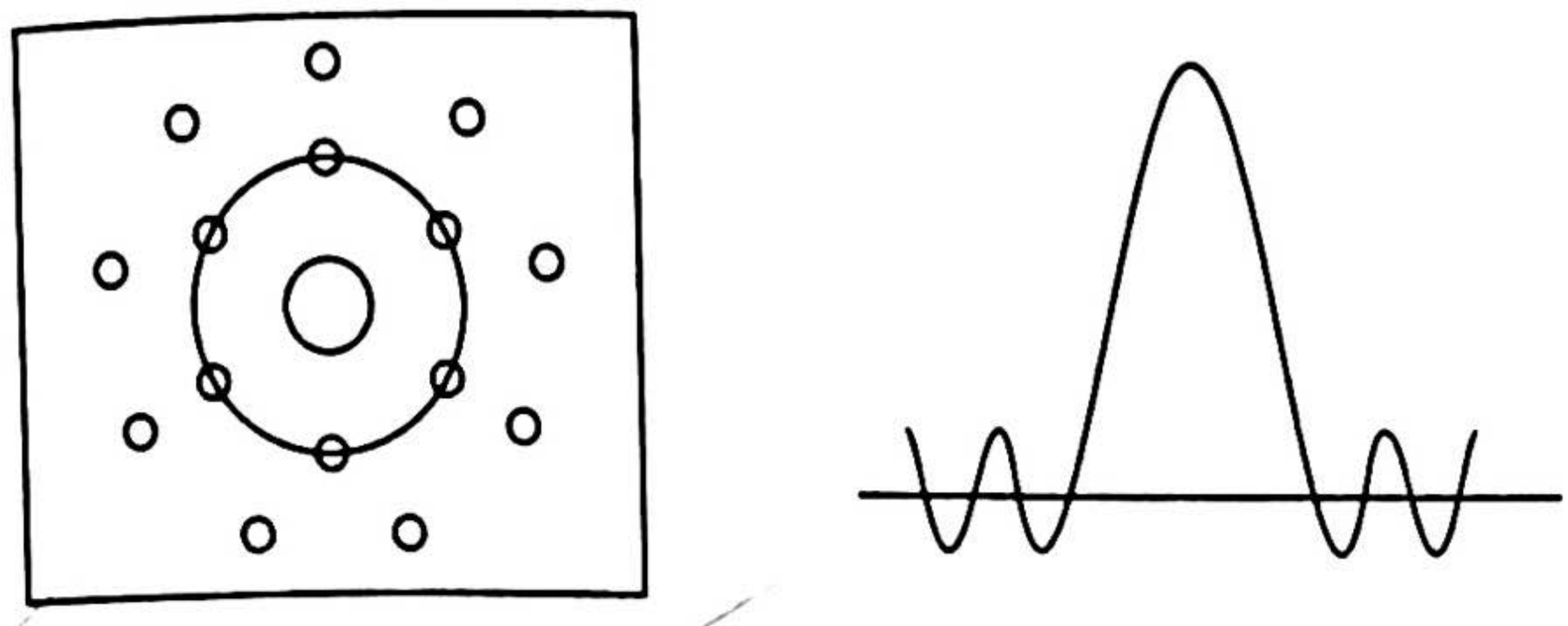


Figure 2.15

[Representation of spatial domain of *ILPF* and its intensity profile.]

Since the cross section of ideal LPF is frequency domain is a "box filter" so it is not unexpected that a cross section of corresponding spatial filter is of a sinc function. The center lobe of this sinc is main cause of blurring while the outer smaller lobe are responsible for ringing.

2.9.2 Butterworth Low Pass Filters

Unlike ILPF the butterworth low pass filter (BLPF) transfer function does not have sharp discontinuity which gives a clear cut off between passed and filtered frequency.

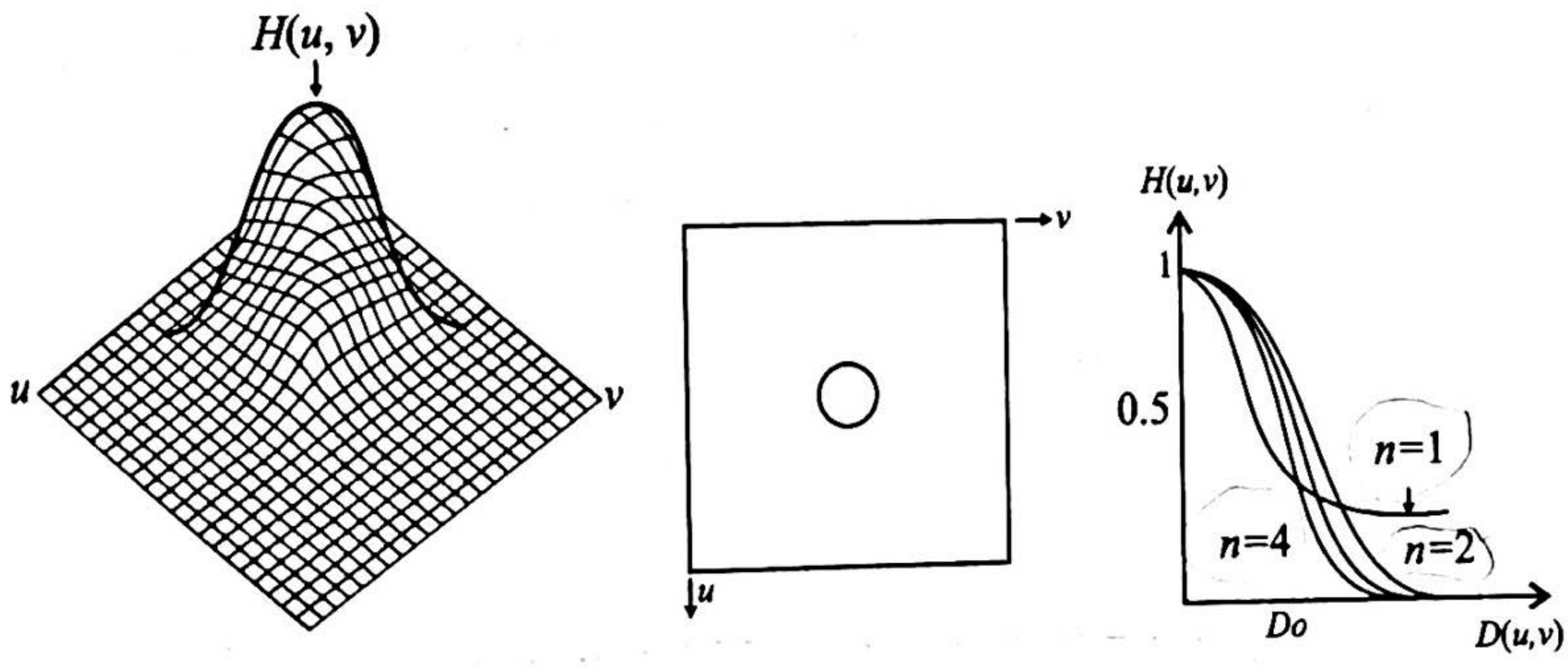


Figure 2.16

The transfer function of a BLPF of order n, at a distance D_o from the center with cut off frequency is given as

$$H(u,v) = \frac{1}{1+\left[D(u,v)/Do\right]^{2n}} \qquad \ldots(2.55)$$

The plot of BLPF, its image display and its cross section is shown below be in figure 2.16.

It is observed that a BLPF of order $n = 1$ has no ringing in the spatial domain but it is generally imperceptible for the $BLPF_S$ of order $n = 2$.

We have consider an example to show comparison between spatial representations of $BLPF_S$ having order 1, 2, 5 and 20 respectively in figure 2.17.

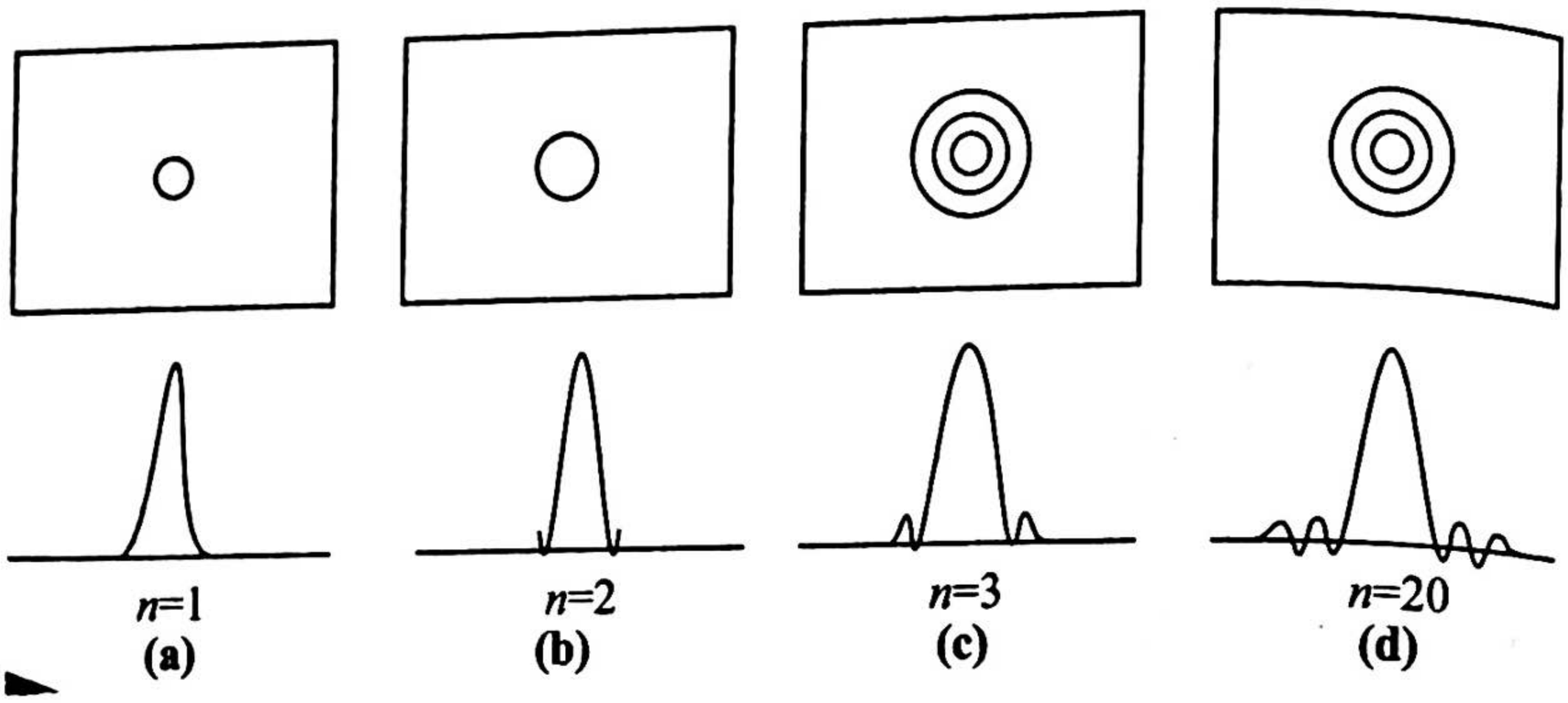


Figure 2.17 Representation of spatial filtering for various order of $BLPF_S$

2.9.3 Gaussian Low Pass Filters

The very smoothing filtering can be achieved by Gaussian low pass filtering. It is suitable for low ordered values.

The Gaussian low pass filters in two dimension is gives as

$$H(u, v) = e^{-D^2(u,v)/2\sigma^2}$$

Where $D(u, v)$ is the distance from the center of the frequency rectangle and σ is a measure of spread about the center.

Let $\sigma = D_o$ than

Then $$H(u, v) = e^{-D^2(u,v)/2Do^2} \quad \ldots(2.56)$$

and D_o is the cut off frequency.

When $D(u, v) = Do$ then gaussian low pass filtering GLPF is

$$H(u, v) = e^{-Do^2/2Do^2}$$

$$H(u,v) = e^{-1/2}$$

$$H(u,v) = 0.607 \qquad \ldots(2.57)$$

The inverse fourier transform of GLPF is also Gaussian. It show that a spatial Gaussian filter will have no ringing.

Figure 2.18 shows plot of Gaussian LPF, its image display and radial cross section.

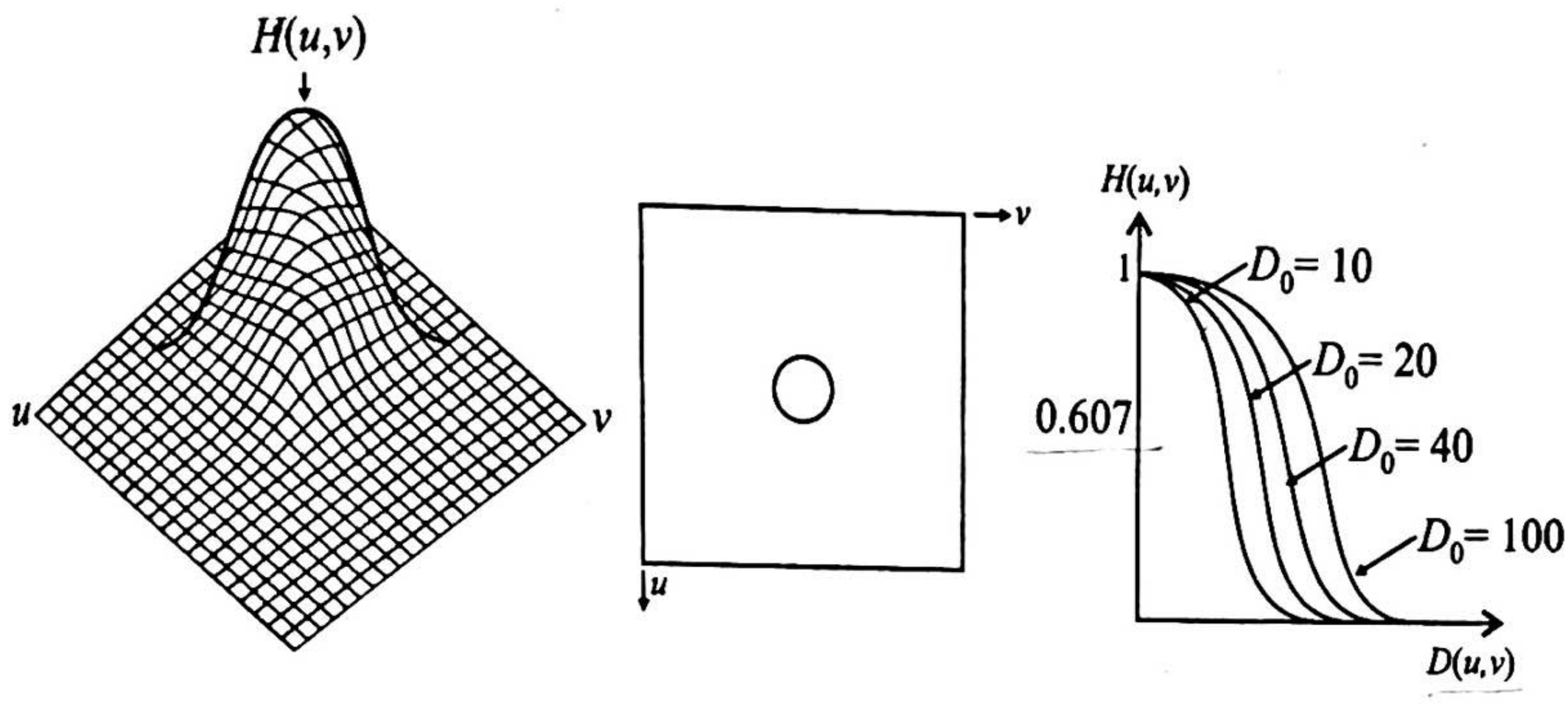


Figure 2.18

2.9.4 Image Sharpening using Frequency Domain Filtering

Since image smoothing can be achieved by attenuating high-frequency component. Similarly the image sharpening can be achieved by attenuating low frequency components without disturbing high frequency informations is Fourier transform. These low frequency components can be attenuated by high-pass filtering.

This high-pass filter is obtained from low pass filter as

$$H_{HP}(u,v) = 1 - H_{LP}(u,v) \qquad \ldots(2.58)$$

Here H_{HP} is the transfer function of high pass filter and H_{LP} is the transfer function of low pass filter.

The butterworth filter represents a transition between the sharpness of ideal LPF and broad smoothness of the Gaussian LPF.

2.10 Ideal Highpass Filters

The ideal High pass filter (IHPF) is the opposite of the ILPF gives that it sets to zero all frequencies inside a circle of radius D_o.

A two Dimensional ideal HPF is given as :

$$H(u,v) = \begin{cases} 0 & for \quad D(u,v) \leq Do \\ 1 & for \quad D(u,v) > Do \end{cases}$$

Where D_0 is the cut-off frequency and

$$D(u,v) = \left[\left(u - P/2\right)^2 \left(v - Q/2\right)^2\right]^{1/2}$$

The IHPFs have same ringing properties at ILPFs have. It is shown is figure 2.19.

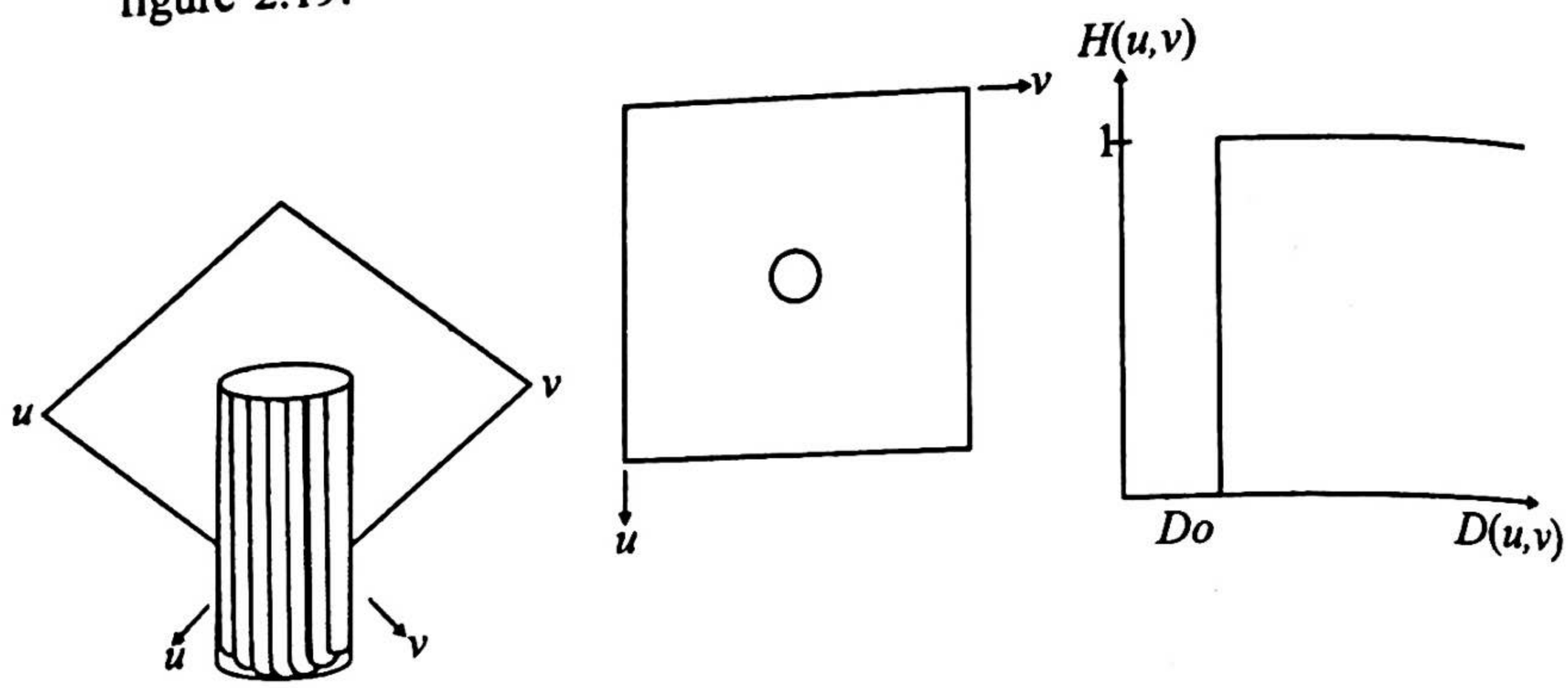


Figure 2.19 IHPFs

2.11 Buterworth High Pass Filters

The two-dimensional butterworth HPF of order n and cut-off frequency D_o is given as

$$H(u,v) = \frac{1}{1 + \left[Do / D(u,v)\right]^{2n}} \qquad \ldots(2.59)$$

The butterworth HPF have is to be smoother than IHPFs. Since the spot size is the center areas of the IHPF and BHPF are similar. So the performance of the two filters on smaller object are comparable.

So it is concluded with figure 2.20 that transition into higher values of cut off frequencies is much smoother with butter worth HPF.

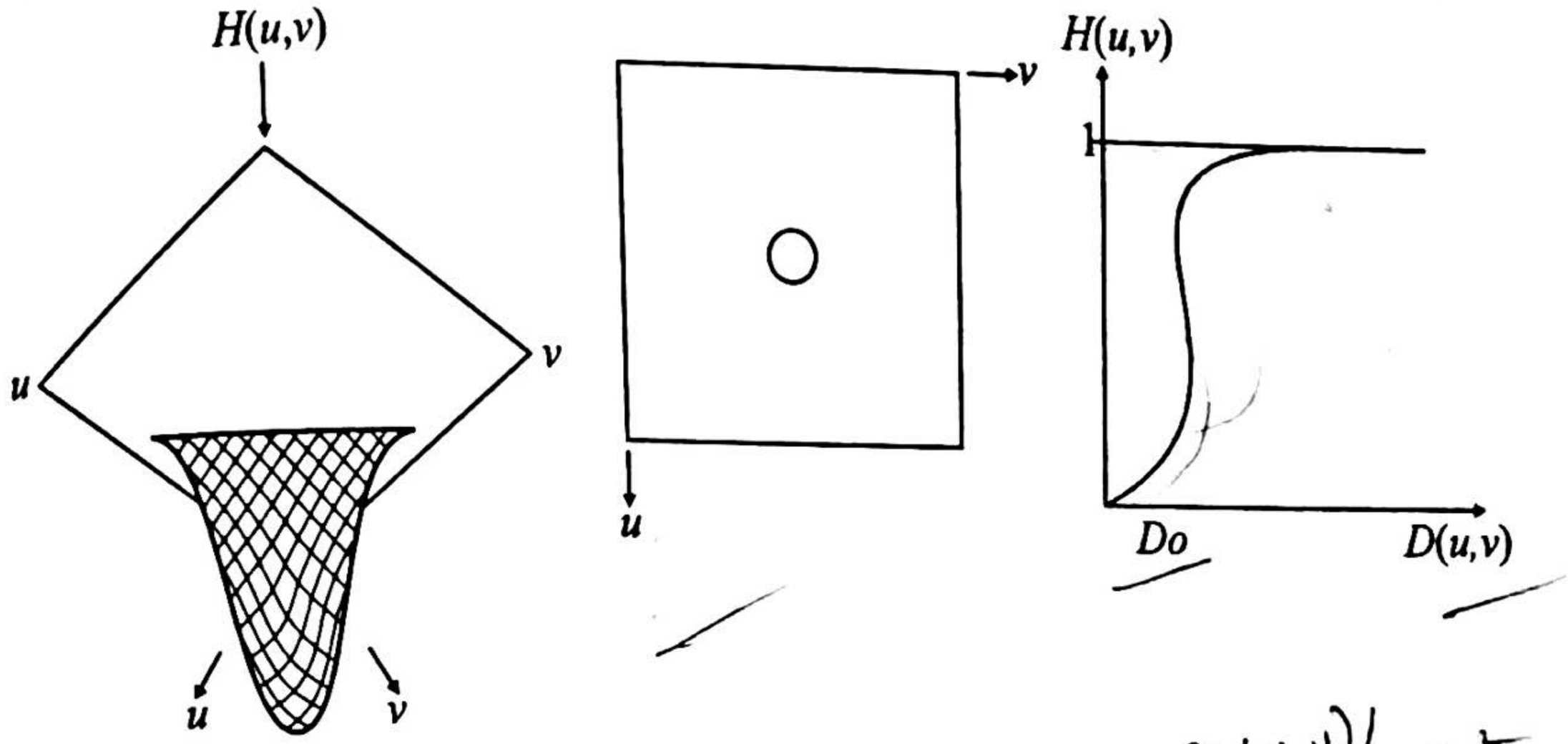


Figure 2.20

$1-e^{-D^2(u,v)/2D_0^2}$

2.12 Gaussian High Pass Filter

Transfer function of the Gaussian HPF with cut off frequency D_0, at a distance $D(u,v)$ from the center of the frequency rectangle is

$$H(u,v) = 1-e^{-D^2(u,v)/2Do^2} \qquad \ldots(2.60)$$

This Gaussian filter provides very low sharpen image as compared to previous two filters.

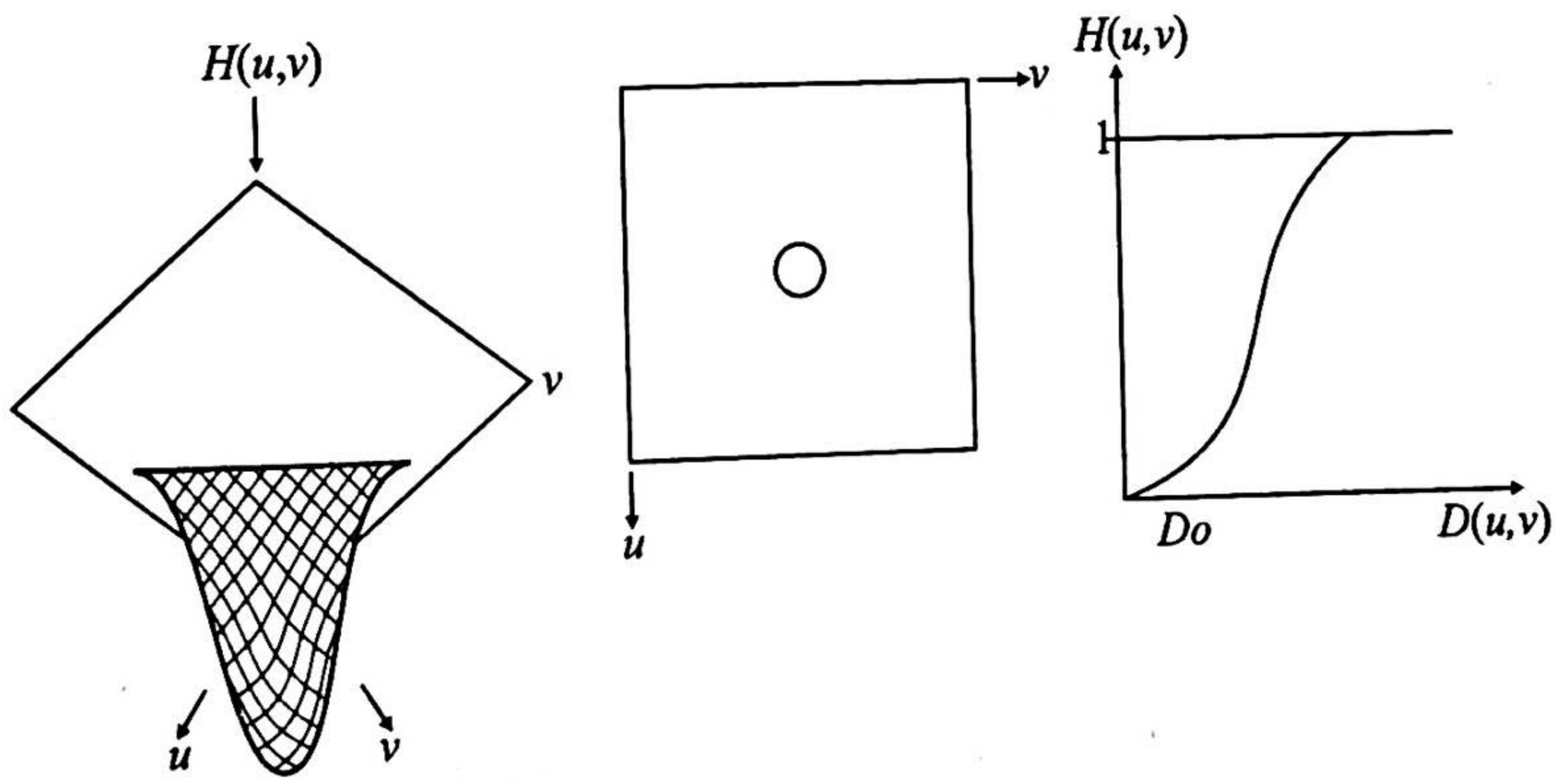


Figure 2.21

Result obtained are more gradual than with the previous two filters. while the filtering of smaller objects and their bans is cleaner.

From the above all discussion it is clear that high pass filtered image lost its gray tones because the *dc* term to reduced to zero. The obtained image is dark tone predominate in HPF images.

the pr... intensity value available in a gray scale image is known as

2.13 Pseudo Coloring

processing.

Pseudo coloring is an important tool in digital image processing. Pseudo-color refers to using color to visualize things that are not inherently colored. If two variables of information have to be displayed at a time in an image, the Pseudo coloring comprises of assigning colors to gray values based on some specific criterion. Pseudo coloring is generally applied on the 2D images. The term Pseudo color or false color is used to differentiate the process of assigning colors to monochrome images from the process associated with true color images.

Pseudo coloring can be effectively used to label or identify particularly significant sections of the image as determined by some other image processing or vision algorithms. Mostly, the Pseudo coloring is used for the detection of presumptive breast cancer. The most important use of Pseudo color is in human visualization and interpretation of gray scale events in an image or sequent of images.

The widely used example of pseudo coloring is intensity slicing or density slicing and color coding.

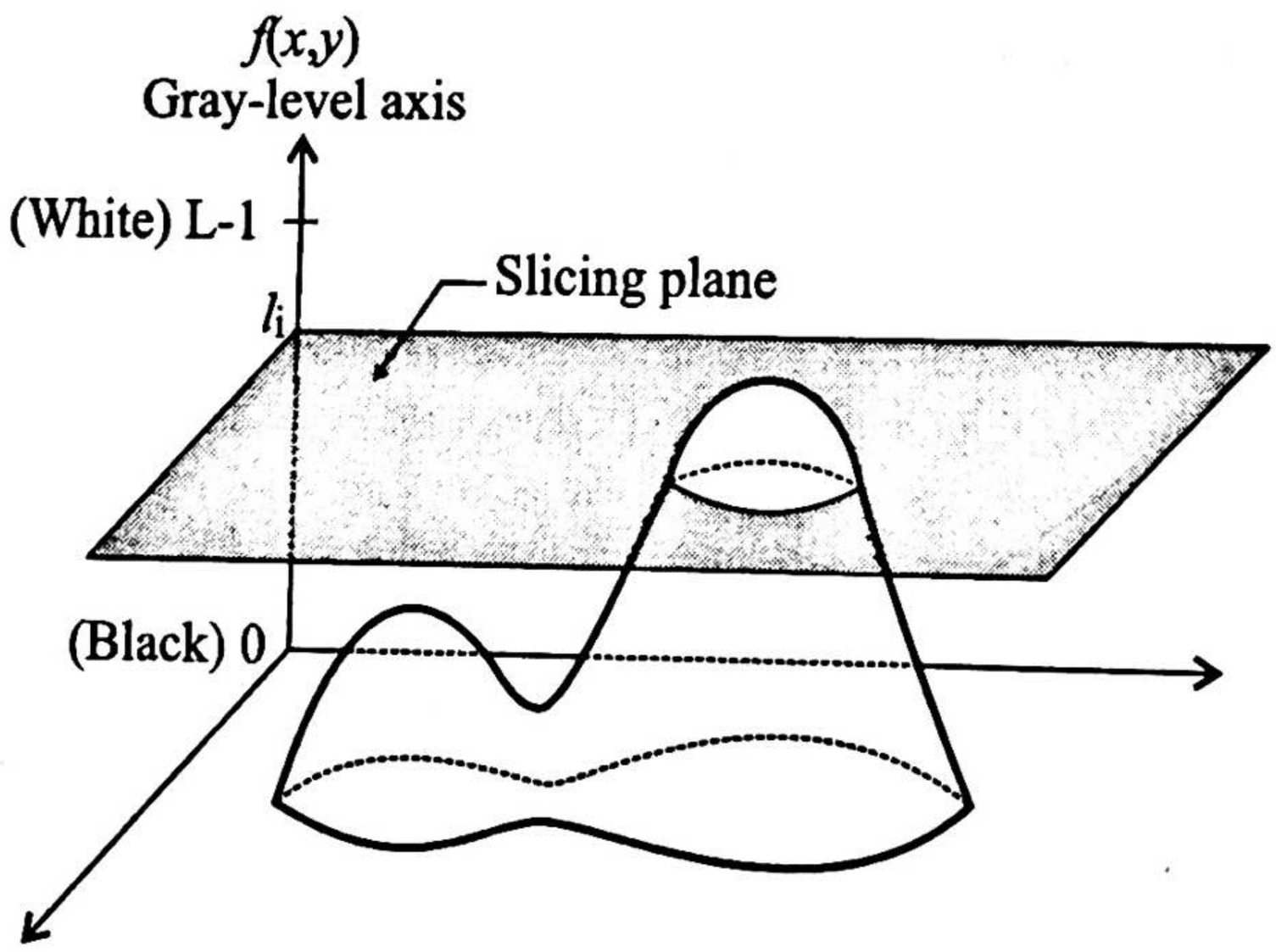


Figure 2.22 Geometric interpretation of the intensity-slicing

Figure 2.22 shows a plane at $f(x,y)$ equals to l_i, to slice the image function into two levels which interprets that any pixel whose gray scale is above the plane will be coded with one color, and any pixel below the plane will be coded with the other. It results in a two color image whose relative appearance can be controlled by moving the slicing plane up and down the gray level axis.

in general we can use multi level slicing (color coding

It is assumed that P planes perpendicular to intensity axis given as level l_i, $i = 1,2,3.....P$. These P plane divides the gray level into P+1 interval.

When more levels are used, the mapping function takes on a staircase form.

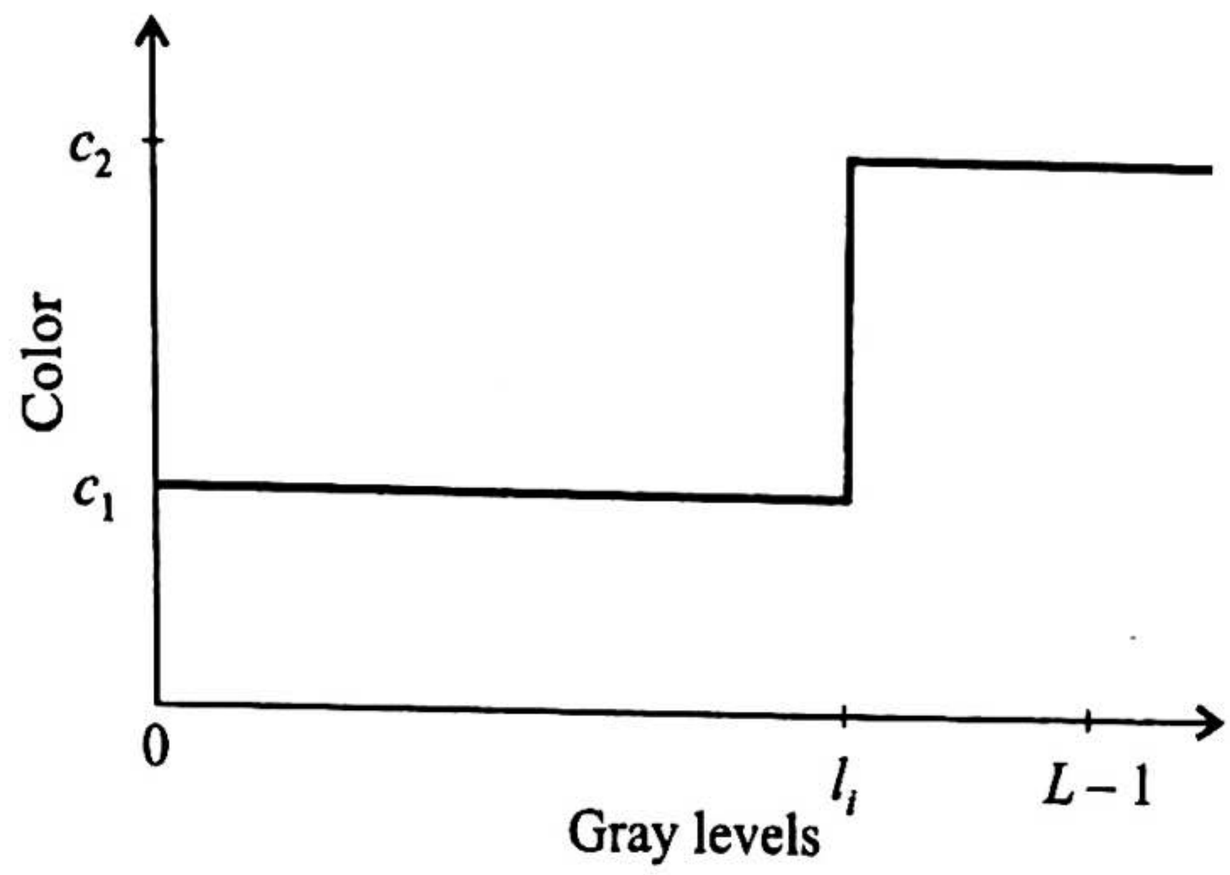


Figure 2.23 Pseudo image processing

Here $f(x, y) = \text{Ci}$ and $f(x, y)\ \varepsilon\ V_k$; $k = 1, 2, 3, P$+1.

Where C_i is the color associated with the kth intensity level and V_k is defined as the partition lanes at $l = (k - 1)$ and $l = k$.

2.13.1 Gray Level to Color Transformations

There are various types of transformations which are capable of achieving a wide range of pseudo color enhancement among the one is described here.

This is used to perform three independent transformations on the gray level of any input pixel. The three results are then fed separately into the Red, Green, and Blue channels of a color television monitor. This produces a composite image, whose color content is modulated by the nature of the transformations on the gray-level values of an image, and is not a functions of position.

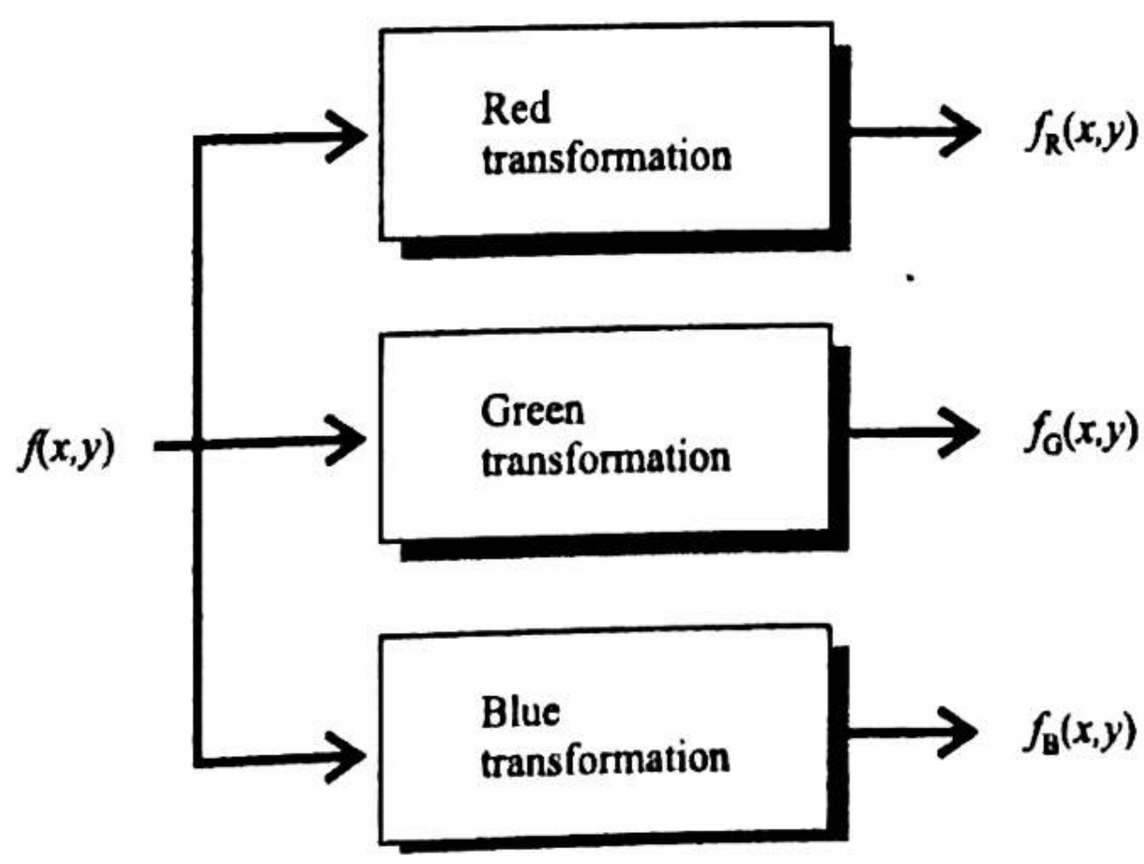
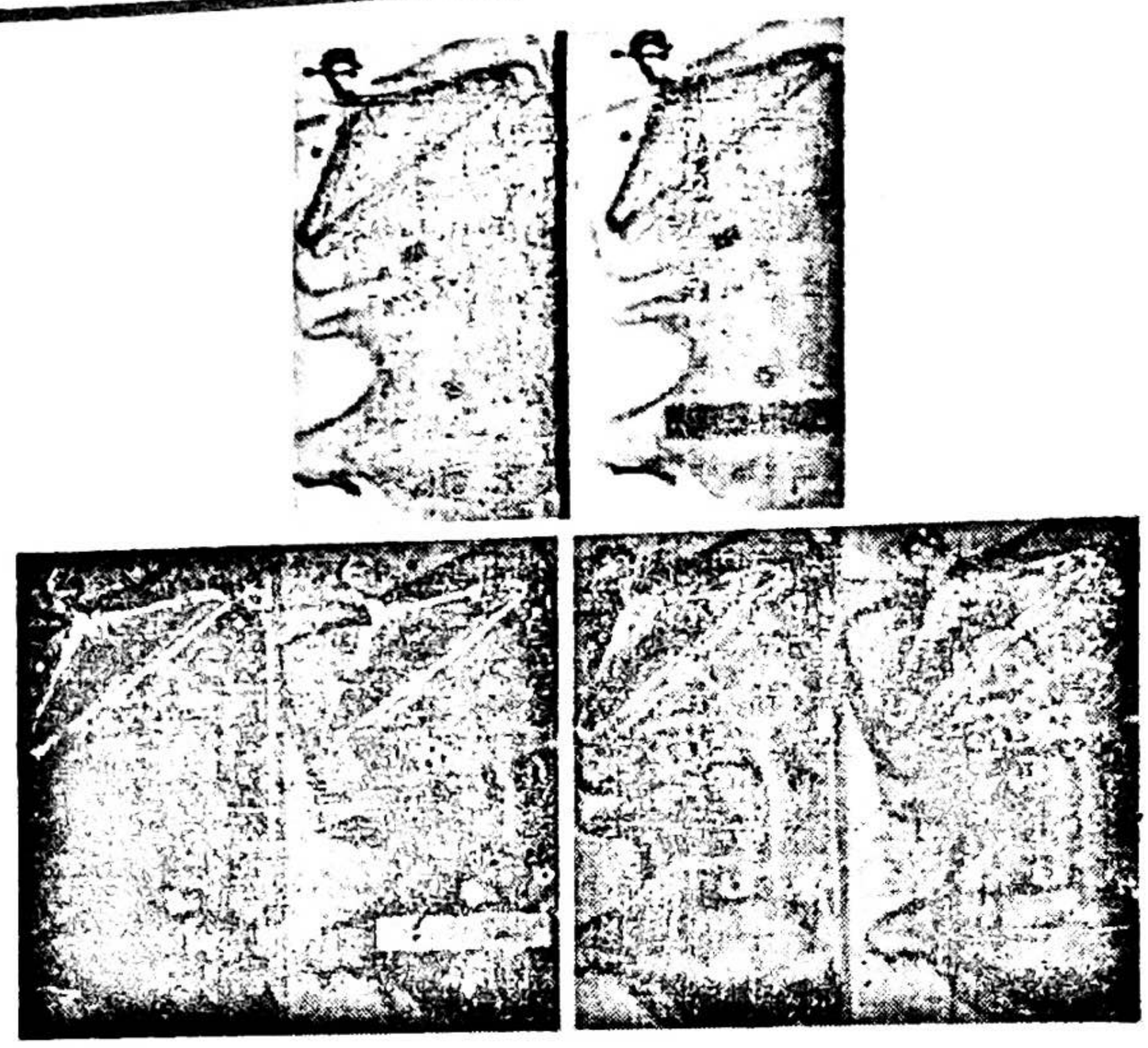
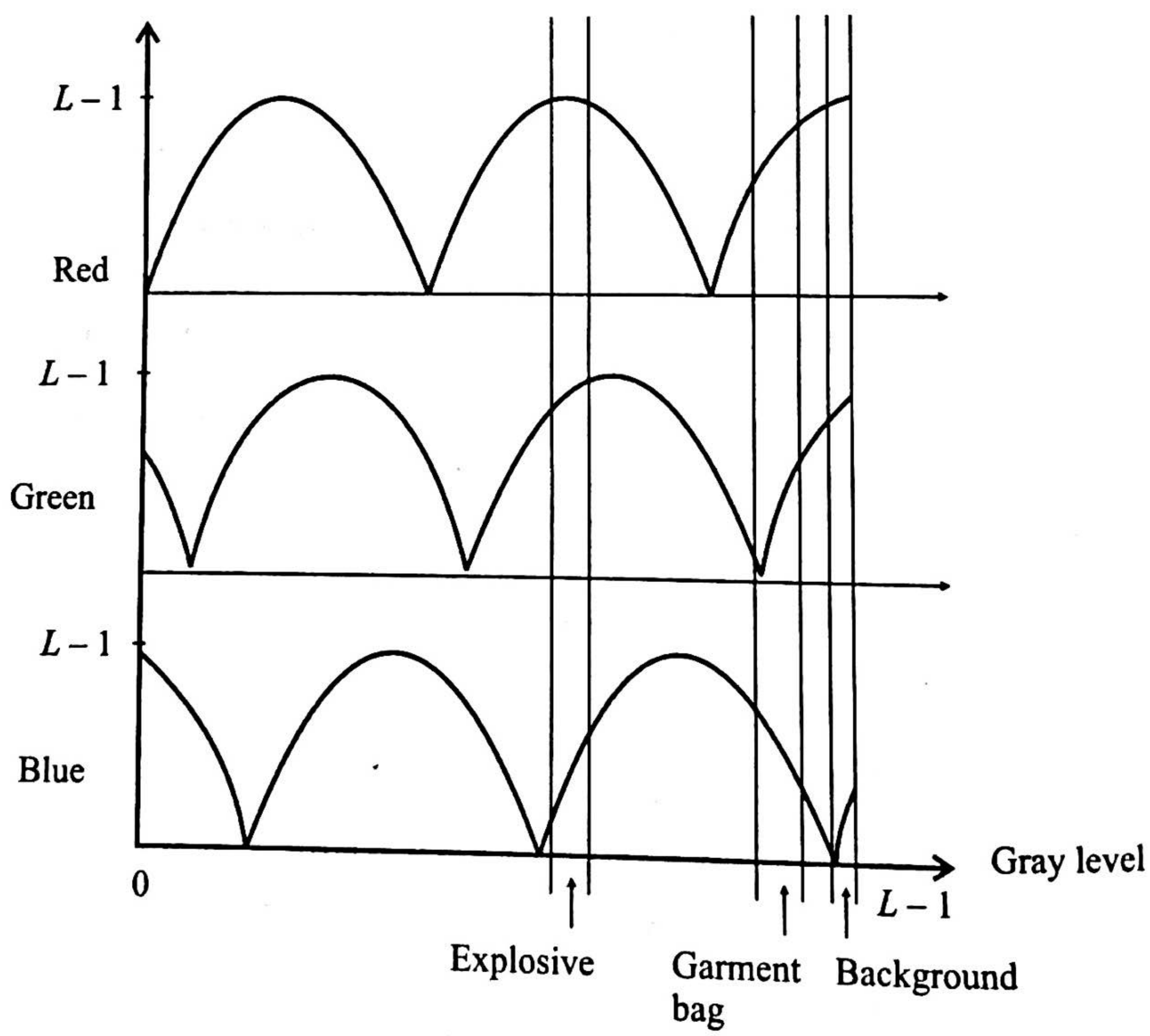


Figure 2.24 Functional block diagram of pseudo color image processing.

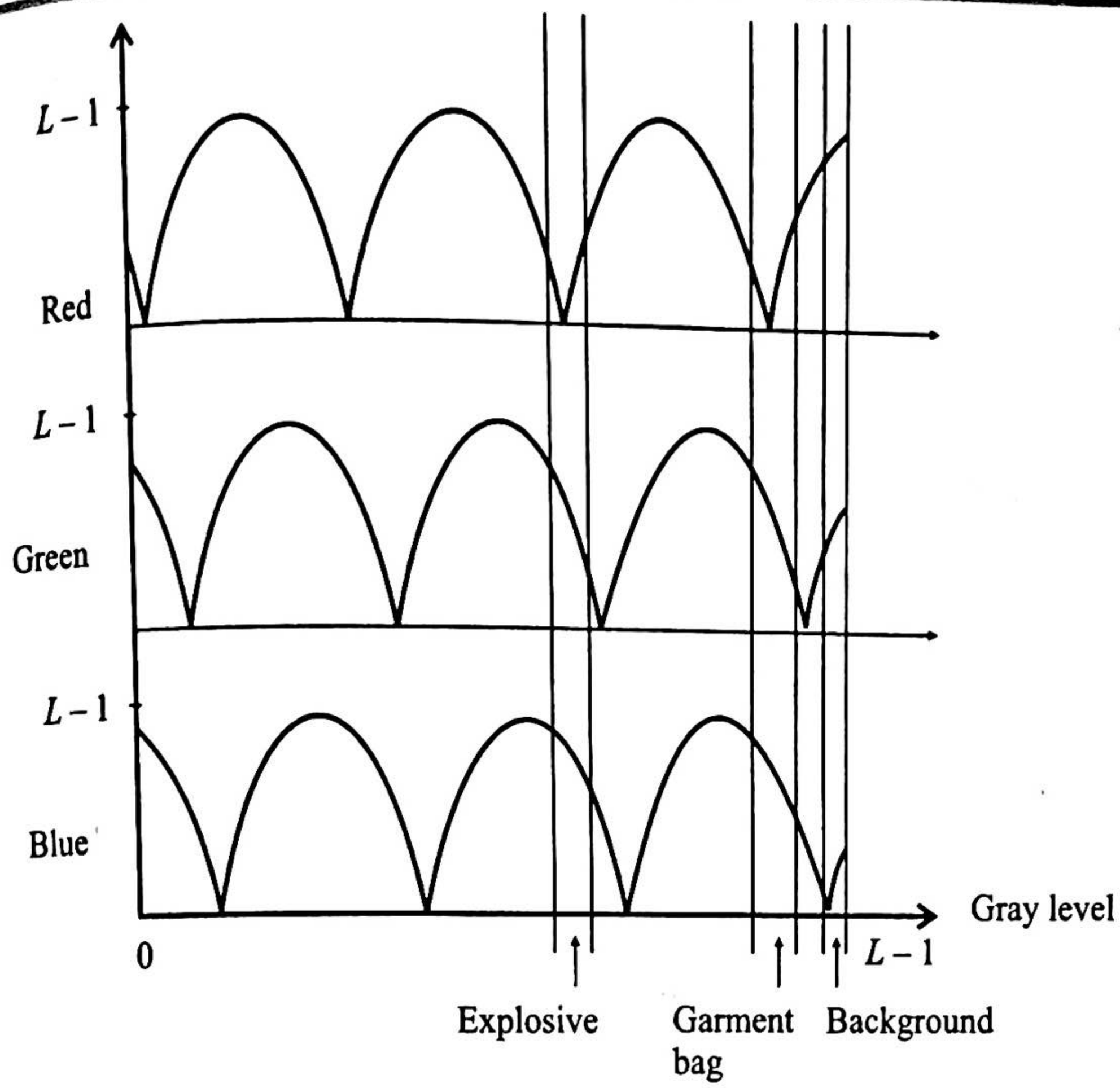


(a)



(b)

(a) Pseudo color enhancement by using gray to color transformation



(c)

(b) and (c) Transformation functions used to obtain image of (a)

Figure 2.25

Above Figure shows that Change in the phase and frequency of each sinusoid can emphasize (in color) ranges in the gray scale. Here Peak represents the constant color region while the valley represents the rapid change of color region. The small change in the phase between the three transforms produces a little change in pixels whose gray level corresponding to the peaks in the sinusoidal.

2.14 Color Transforms

Color transformation simply includes following steps:

1. **Formulation:** It includes gray level transformations and color transformations with the color components.

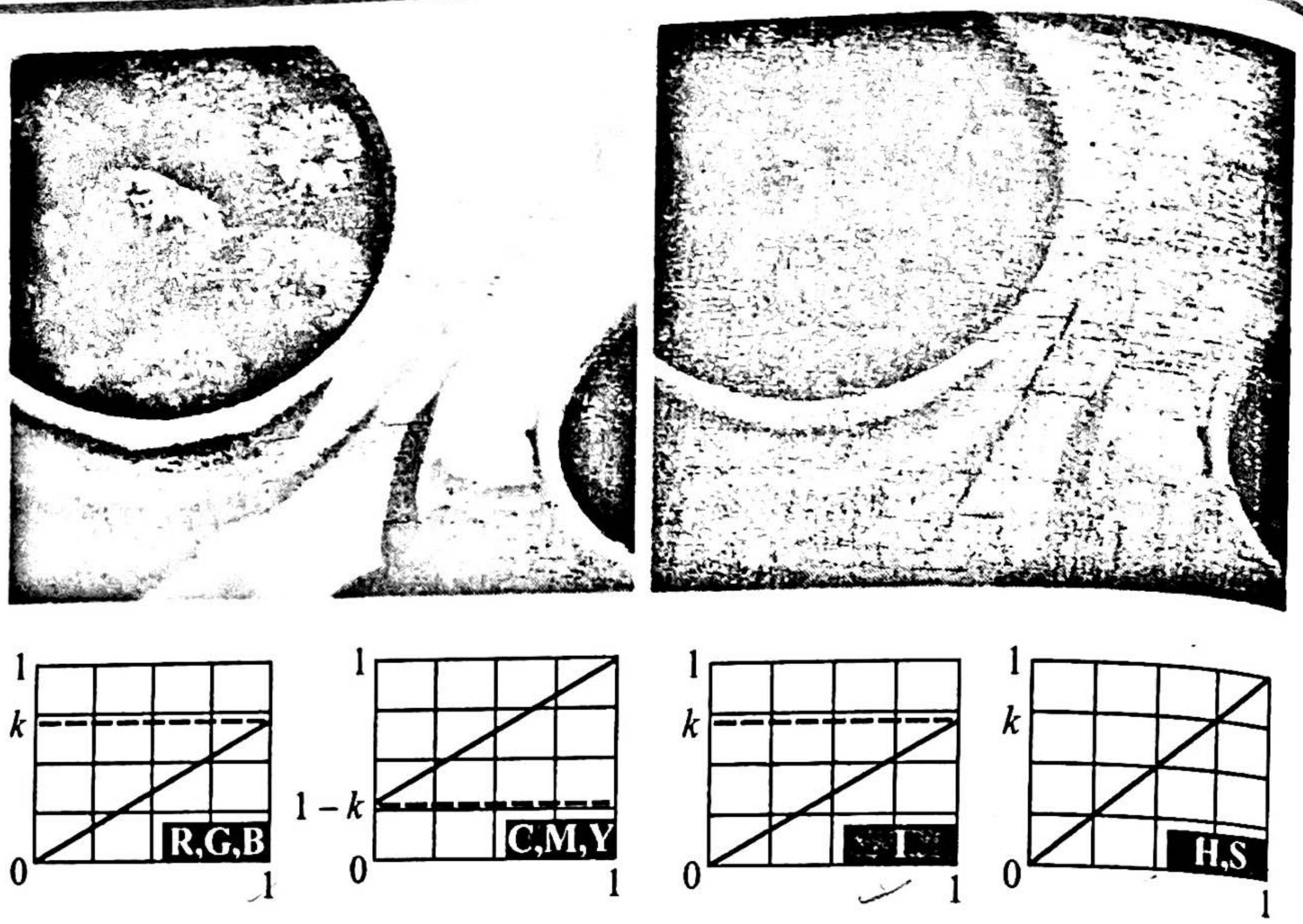


Figure 2.26 Adjustment of intensity of image using color transformation

Modification of intensity is given as:

$$g(x, y) = kf(x, y), \quad 0<k<1$$

2. **Color complements:** The hues directly opposite one another on the color circle are called color complements. Color complements are useful for enhancing detail that is embedded in dark regions of a color image.

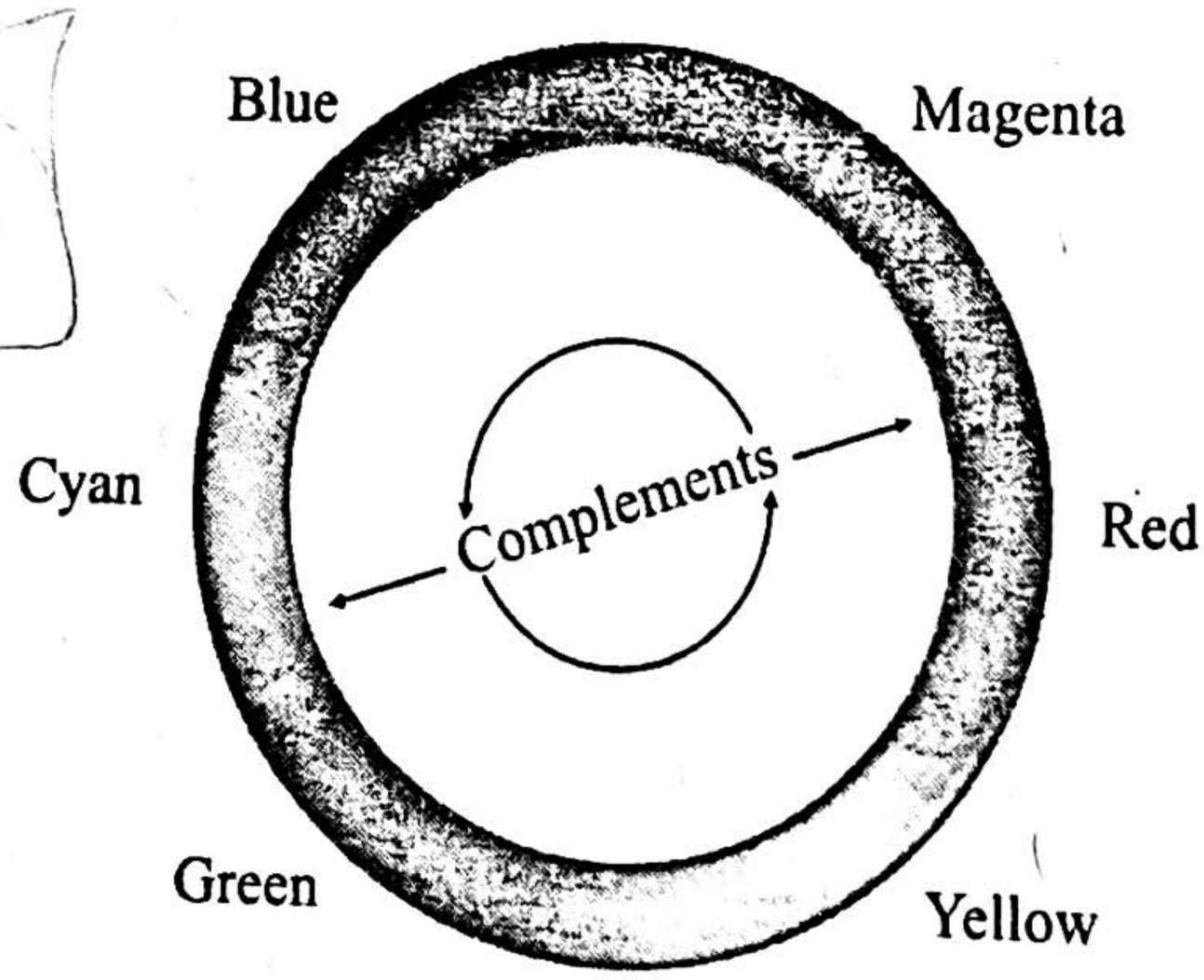


Figure 2.27 Complement on the color circle

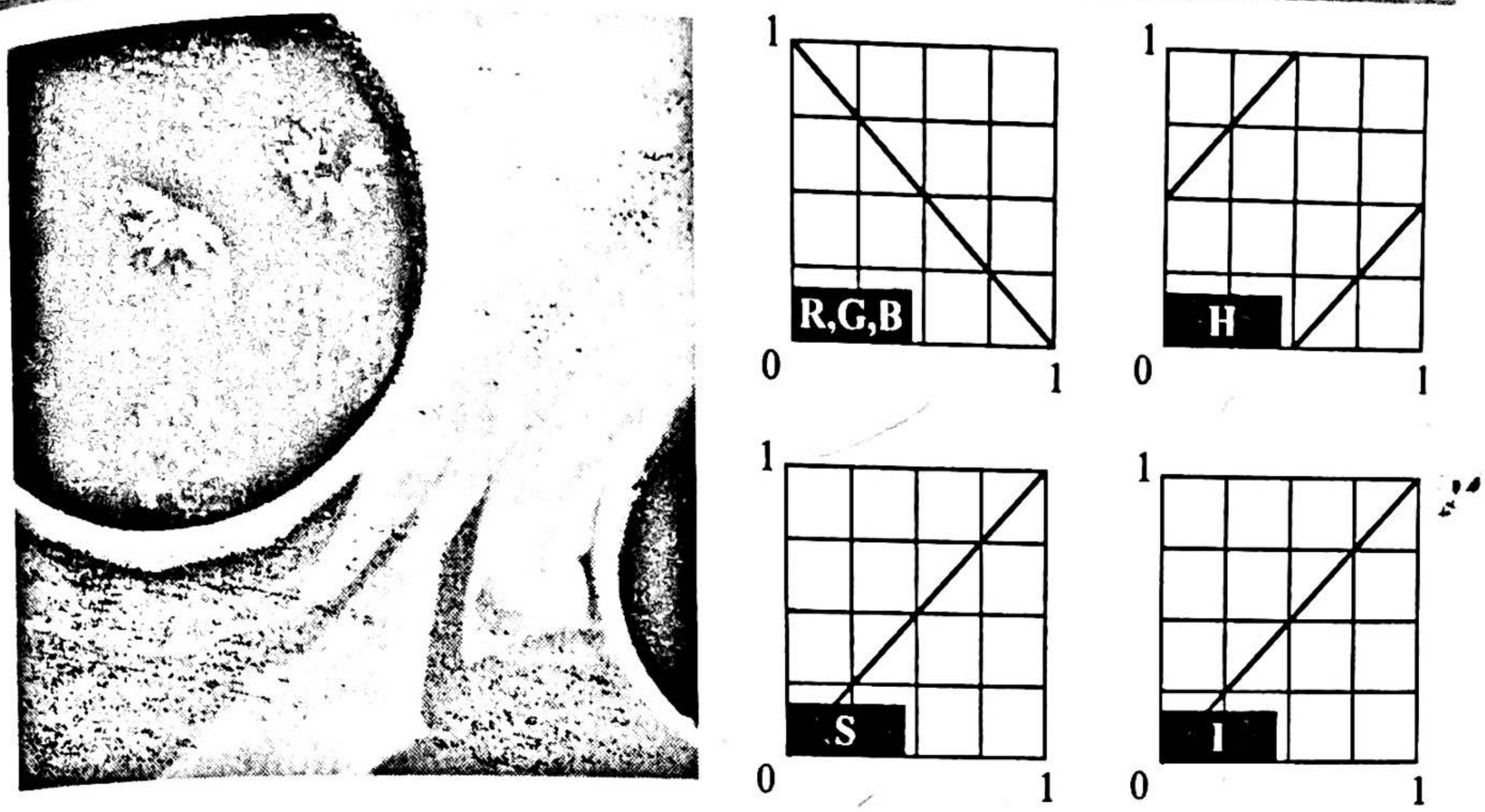


Figure 2.28 Color complements transformation in color models

3. **Color Slicing:** Color slicing is used to highlight a specific range of colors in an image to separate objects from surroundings. It displays just the colors of interest, or use the regions defined by specified colors for further processing. Color slicing is more complex than gray-level slicing, due to multiple dimensions for each pixel. It depends on the color space chosen.

2.15 Wavelet Transform

Wavelets are a more general way to represent and analyze multi resolution images which can also be applied to 1-D signals. In conventional Fourier transform, we use sinusoid for basis functions. It can only provide the frequency information. Temporal information is lost in this transformation process. In some applications, we need to know the frequency and temporal information at the same time, such as a musical score, we want to know not only the notes (frequencies) we want to play but also when to play them. Unlike conventional Fourier transform, wavelet transforms are based on small waves, called wavelets. It can be shown that we can both have frequency and temporal information by this kind of transform using wavelets. Moreover, images are basically matrices. For this reason, image processing can be regarded as matrix processing. Due to the fact that human vision is much more sensitive to small variations in color or brightness, that is, human vision is more sensitive to low frequency signals. Therefore, high frequency components in images can be compressed without distortion. Wavelet transform is

one of a best tool for us to determine where the low frequency area and high frequency area is. These kinds of applications will be discussed later. These are useful for image compression and removing noise.

The fundamental idea behind wavelets is to analyze according to scale. Indeed, some researchers in the wavelet field feel that, by using wavelets, one is adopting a whole new mindset or perspective in processing data.

Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. This idea is not new. Approximation using superposition of functions has existed since the early 1800's, when Joseph Fourier discovered that he could superpose sines and cosines to represent other functions. However, in wavelet analysis, the scale that we use to look at data plays a special role. Wavelet algorithms process data at different scales or resolutions. If we look at a signal with a large "window," we would notice gross features. Similarly, if we look at a signal with a small "window," we would notice small features. The result in wavelet analysis is to see both the forest and the trees, so to speak.

This makes wavelets interesting and useful. For many decades, scientists have wanted more appropriate functions than the sines and cosines which comprise the bases of Fourier analysis, to approximate choppy signals . By their definition, these functions are non-local (and stretch out to infinity). They therefore do a very poor job in approximating sharp spikes. But with wavelet analysis, we can use approximating functions that are contained neatly in finite domains. Wavelets are well-suited for approximating data with sharp discontinuities.

The wavelet analysis procedure is to adopt a wavelet prototype function, called an analyzing wavelet or mother wavelet. Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet. Because the original signal or function can be represented in terms of a wavelet expansion (using coefficients in a linear combination of the wavelet functions), data operations can be performed using just the corresponding wavelet coefficients. And if you further choose the best wavelets adapted to your data, or truncate the coefficients below a threshold, your data is sparsely represented. This sparse coding makes wavelets an excellent tool in the field of data compression.

Other applied fields that are making use of wavelets include astronomy, acoustics, nuclear engineering, sub-band coding, signal and image processing, neurophysiology, music, magnetic resonance imaging, speech discrimination, optics, fractals, turbulence, earthquake-prediction, radar, human vision, and pure mathematics applications such as solving partial differential equations.

Wavelet related transform

1. **Continuous Wavelet Transform:** A continuous wavelet transform is used to divide a continuous-time function into wavelets. Unlike Fourier transform, the continuous wavelet transform possesses the ability to construct a time frequency represented of a signal that offers very good time and frequency localization.
2. **Multiresolution Analysis:** A multiresolution analysis (MRA) or multiscale approximation (MSA) is the design methods of most of the practically relevant discrete wavelet transform (DWT) and the justification for the algorithm of the fast Fourier wavelet transform (FWT).
3. **Discrete Wavelet Transform:** In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information.
4. **Fast Wavelet Transform:** The Fast Wavelet Transform is a mathematical algorithm designed to turn a waveform or signal in the time domain into a sequence of coefficients based on an orthogonal basis of small finite waves, or wavelets. The transform can be easily extended to multidimensional signals, such as images, where the time domain is replaced with the space domain.
5. **Applications of Wavelet Transform:** As mentioned, most of the applications of Wavelet Transform is about image processing such as image compression, edge detection, noise removal, etc. In last chapter, we see that images can be decomposed into four parts by two-dimensional Wavelet Transform. In fact, the decomposition can continue until the size of the sub-image is as small as you want. By setting some parts of its sub-images, we can reduce the quantity of information, in other words we can compress the image by setting

the useless data. Edge detection and noise removal are based on the same idea. When we want to detect the edges of the image, we can simply set the diagonal sub-images to zero, then we can obtain the output image with edges clearly.

REVIEW QUESTIONS

Q.1 What are the Gray level transformation function of an image.

Q.2 What do you mean by histogram. Explain histogram equalization of an image.

Q.3 What is spatial filtering? Define spatial correlation and convolution with an example.

Q.4 What is image sharpening? Explain first and second order derivatives of image sharpening.

○○○

CHAPTER 3

Image Restortion

Objectives

- Introduction
- Image Degradation/Restoration Model
- Noise Model
- Spatial and Frequency Properties of Noise
- Some Noise Probability Density Function
- Uniform Noise PDF
- Impulse Noise (Salt-and-Pepper) PDF
- Periodic Noise
- Estimation of Noise Parameters
- Noise Only Spatial Filtering
- Adaptive Filters
- Inverse Filtering
- Minimum Mean Square Error (Wiener) Filtering

3.1 Introduction

The restoration technique is used to improve an image in some sense. This technique recover an image which has been degraded. So the restorations are oriented towards modelling the degradation and then applying the inverse process to recover original image. In this unit we explore the restoration mode and various noise model.

3.2 Image Degradation/Restoration Model

The restoration is to obtain an estimated image of the original image. The restoration model is explained with degradation model. It consists of a degradation function *H*, with an additive noise *n* and a restoration filter.

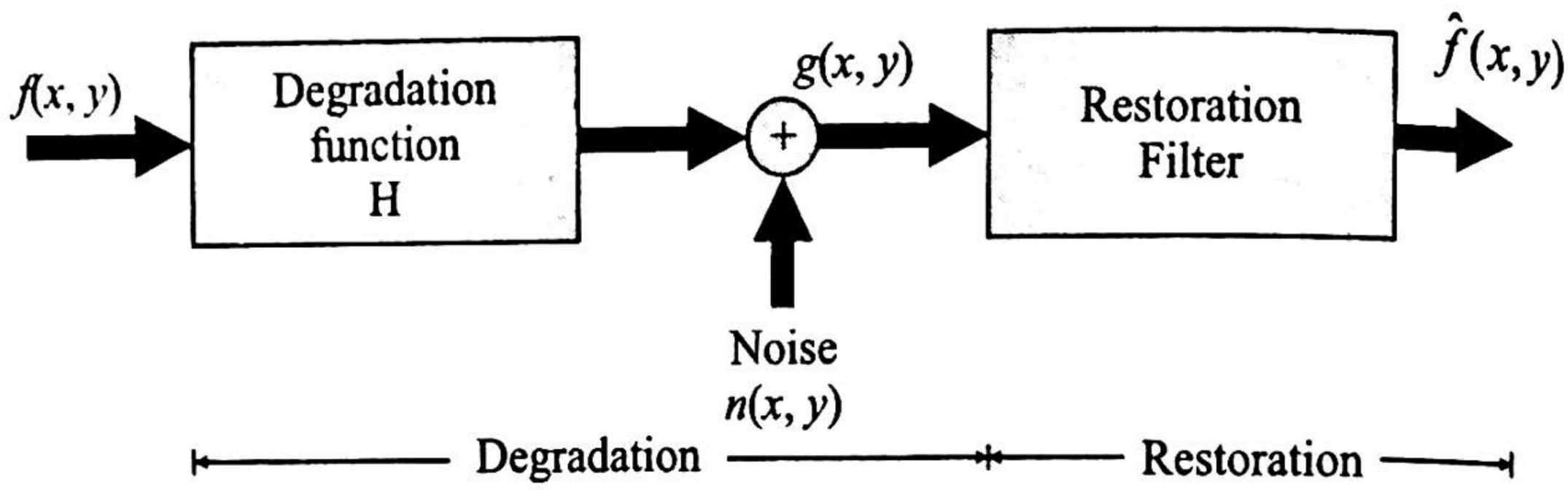


Figure 3.1 Image degradation/restoration model

The image restoration process consists of degrading the input image with the degradation function *H*, which is added with an additive noise $n(x,y)$ to produce degraded image $g(x,y)$. The restoration gives the estimated image $\hat{f}(x,y)$ of the original image. The restorated image (estimated image) should be as much closer as possible to the original input image. There are various restoration filters used to restore or estimate the image. The degraded image in the spatial domain is given as

$$g(x,y) = h(x,y)*f(x,y) + n(x,y) \qquad \ldots(3.1)$$

Here $h(x,y)$ is the spatial representation of degradation function, $f(x,y)$ is input to the degradation function, $n(x,y)$ is additive noise and '*' indicates convolution function.

The equivalent frequency domain representation of this model is given as

$$G(u,v) = H(u,v)\, F(u,v) + N(u,v) \qquad \ldots(3.2)$$

In this chapter it is to be assumed that H is the identity operator and we study only with degradation due to presence of noise.

3.3 Noise Model

At the image acquisition and transmission process in any digital image processing the noise is introduced. This noise affects the performance of imaging sensors. If we acquiring images with a CCD camera, the noise developed is mainly V affected due to light levels and sensor temperature.

So in whole we can say that images are corrupted or affected during image transmission or acquiring due to interference. This noise model consists of image acquisition and transmission, wireless network and sensors through which a resultant or corrupted image is acquired.

3.4 Spatial and Frequency Properties of Noise

The frequency properties refer to the frequency constant of noise as opposed to frequency of the electromagnetic spectrum. Similarly the spatial characteristics of noise are correlated with the image.

If we consider that fourier spectrum of noise is constant than noise is known as "white noise". It is carryover to white light which contains all frequency component in the visible spectrum is equal proportion.

This noise is independent of spatial coordinates and thus it is uncorrected with the image itself, which gives that there is no correlation between pixel values on noise contents.

3.5 Some Noise Probability Density Function

The spatial noise is the statistical behavior of the intensity values in the noise component of the restoration model. It may be viewed as random variable, which can be explained by probability density function (PDF). Some basic and important PDFs are explained here for image processing application. noise i model

3.5.1 Gaussian Noise

Gaussian or normal noise model is widely used due to its mathematical tractability in spatial and frequency domain.

The PDF of a Gaussian random variable Z is given as

$$P(z) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-(z-\bar{z})/2\sigma^2} \qquad \ldots(3.3)$$

Where, Z is the intensity,

$\overline{Z}$ is the mean (average) value of Z.

and σ is the standard deviation of Z.

and σ^2 is the variance of Z

about and 70% value of Z lies in $[(\bar{Z}-\sigma),(\bar{Z}+\sigma)]$ while the 95%, lies in $[(\bar{Z}-2\sigma), (\bar{Z}+2\sigma)]$.

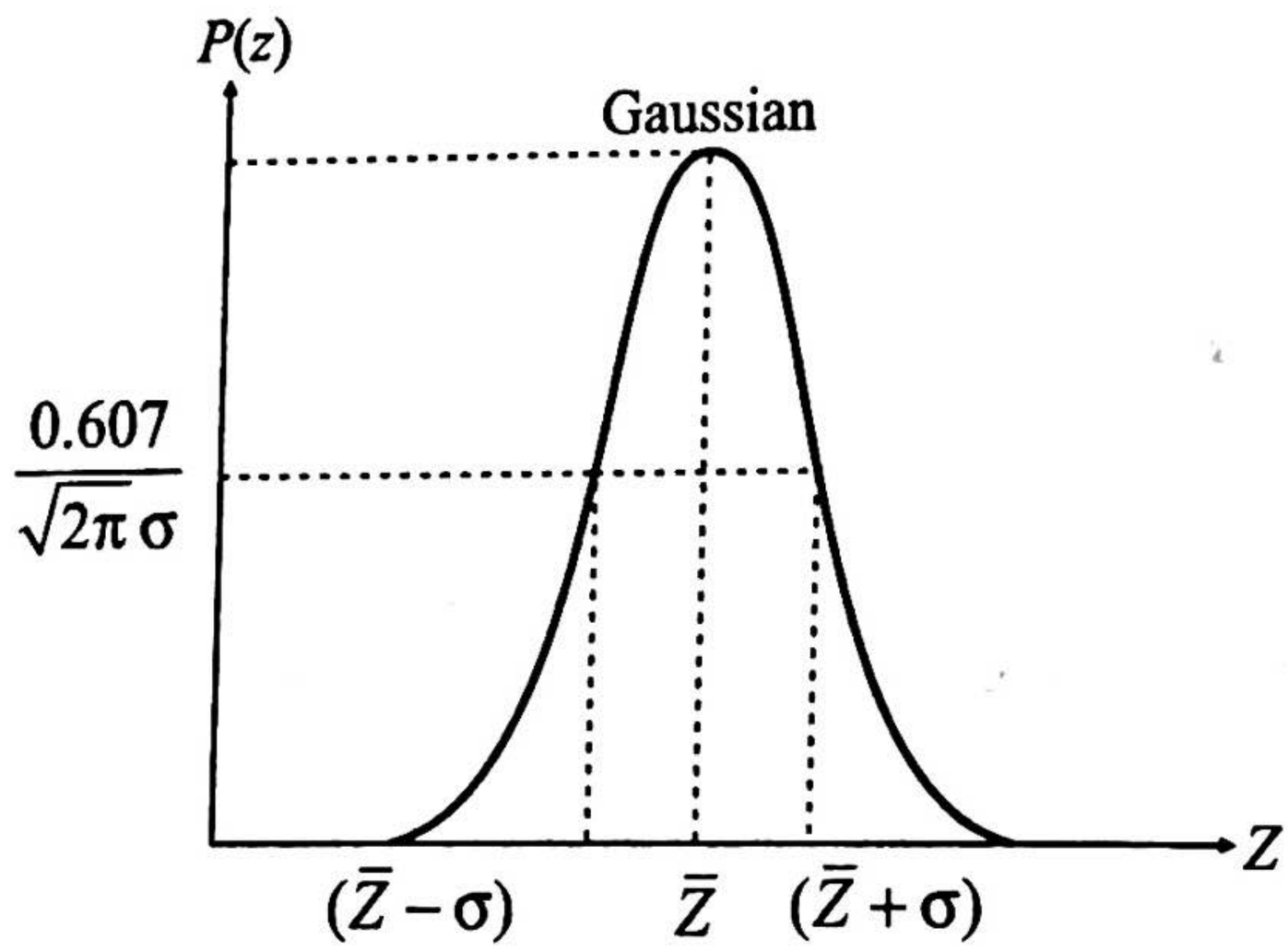


Figure 3.2 PDF of gaussian noise

3.5.2 Rayleigh Noise

The rayleigh PDF is useful for approximating skewed histogram images the PDF of rayleigh noise for variable Z is given as

$$P(Z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & for\, Z \geq a \\ 0 & for\, Z < a \end{cases} \qquad \ldots(3.4)$$

the mean $$\overline{Z} = a + \sqrt{\pi b/4} \qquad \ldots(3.5)$$

and variance $$\sigma^2 = \frac{b+(4-\pi)}{4} \qquad \ldots(3.6)$$

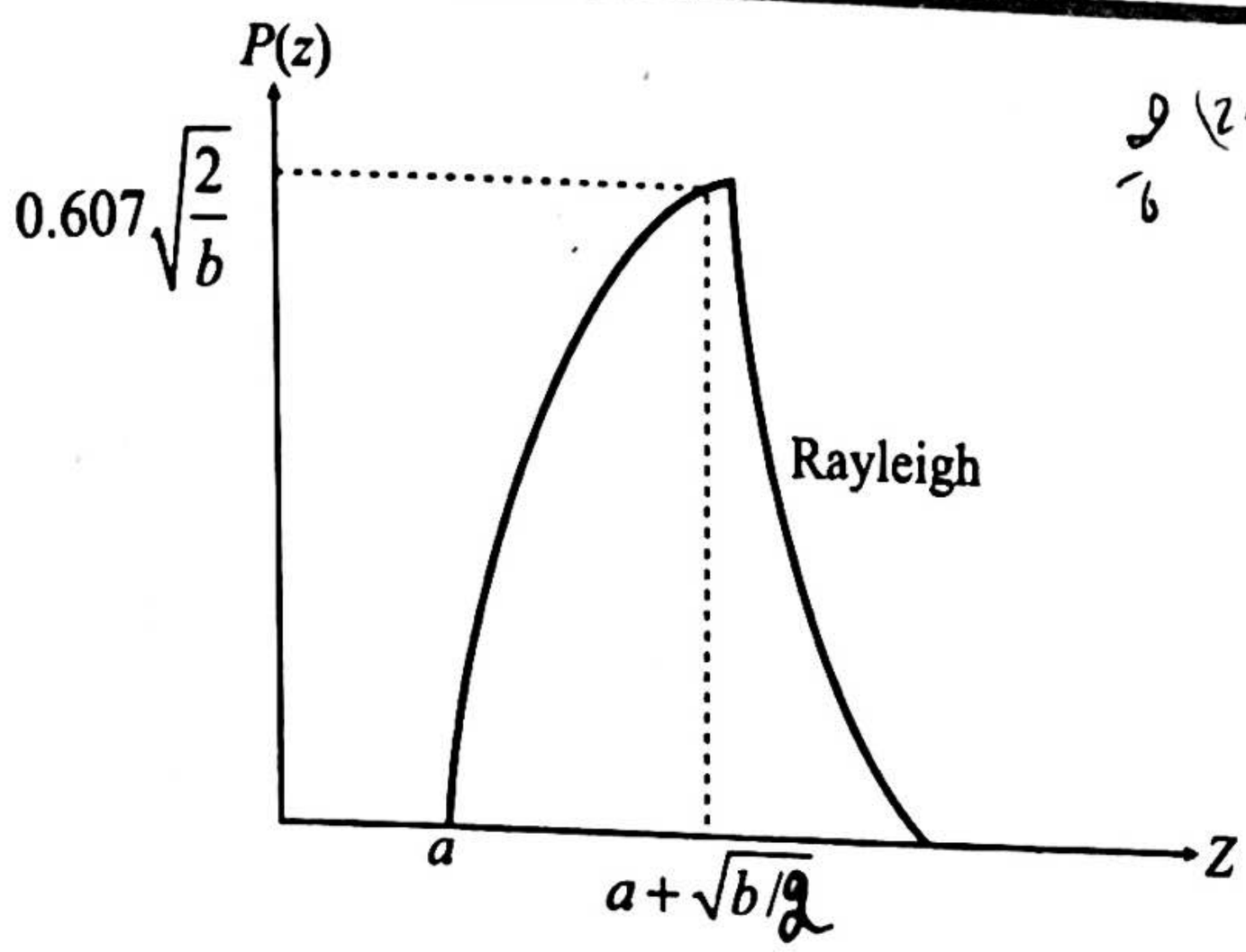


Figure 3.3 PDF on rayleigh

It is cleared that basic shape of the density function is skewed to the right.

3.5.3 Erlong (Gamma) Noise

The PDf of erlang noise for variable Z *is* given as

$$P(Z) = \begin{cases} \dfrac{a^b Z^{b-1}}{(b-1)!} & for\, Z \geq 0 \\ 0 & for\, Z < 0 \end{cases} \qquad \ldots(3.7)$$

Here for $a > 0$, b is a positive integrer.

the mean of this density is

$$\bar{Z} = \frac{b}{a} \qquad \ldots(3.8)$$

and variance $$\sigma^2 = \frac{b}{a^2} \qquad \ldots(3.9)$$

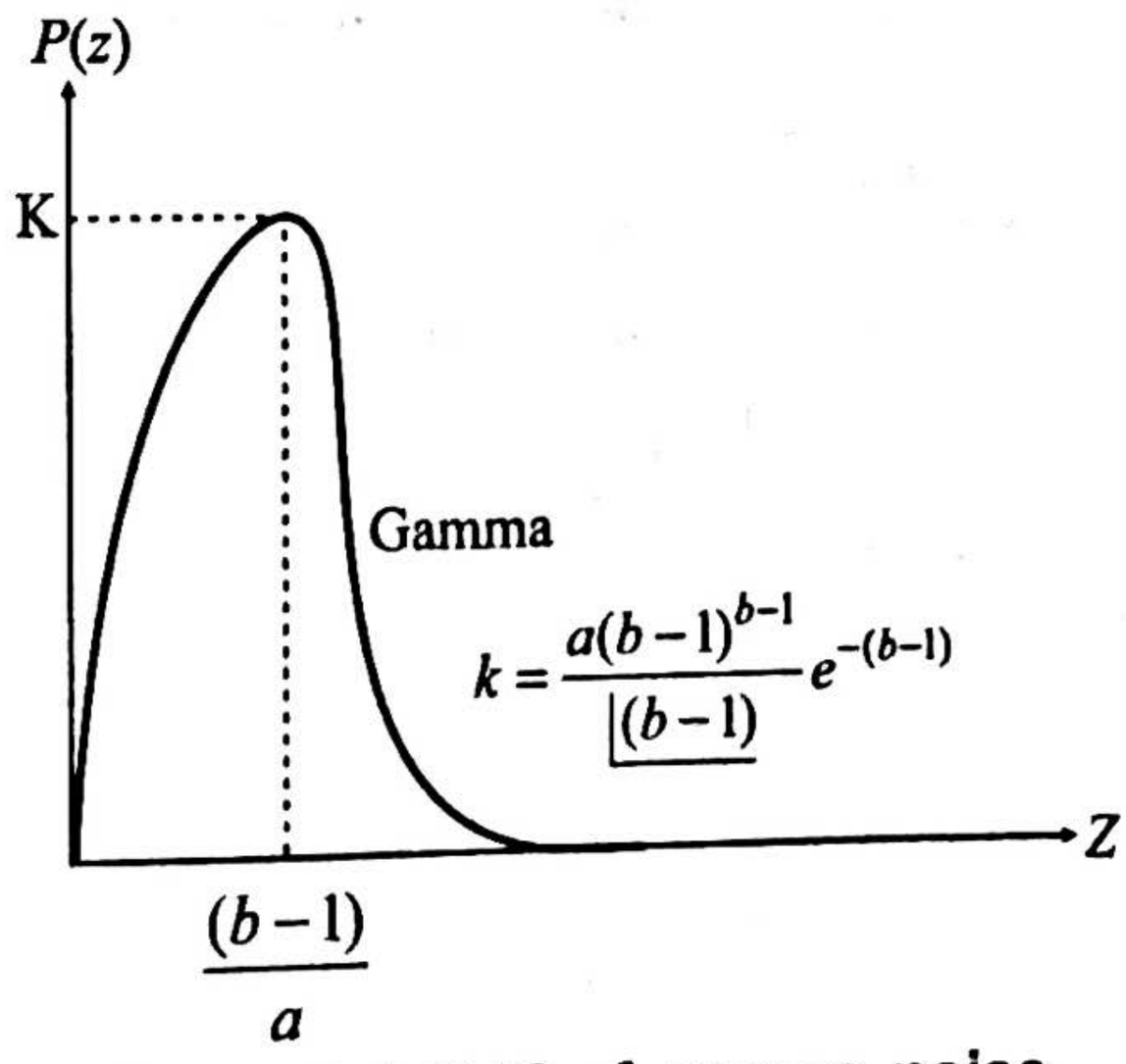


Figure 3.4 PDF of gamma noise

This is gamma PDF and shows that denominates is the gamma function $\Gamma(b)$.

3.5.4 Exponential Noise

In the Earlong PDF if we consider $b = 1$ than it gives exponential noise PDF which is given as

$$P(z) = \begin{cases} ae^{-az} & for\, z \geq 0 \\ 0 & for\, z < 0 \end{cases} \quad \ldots(3.10)$$

where $a > 0$.

The mean value for exponential PDF is

$$\bar{Z} = \frac{1}{a} \quad \ldots(3.11)$$

and variance $$\sigma^2 = \frac{1}{a^2} \quad \ldots(3.12)$$

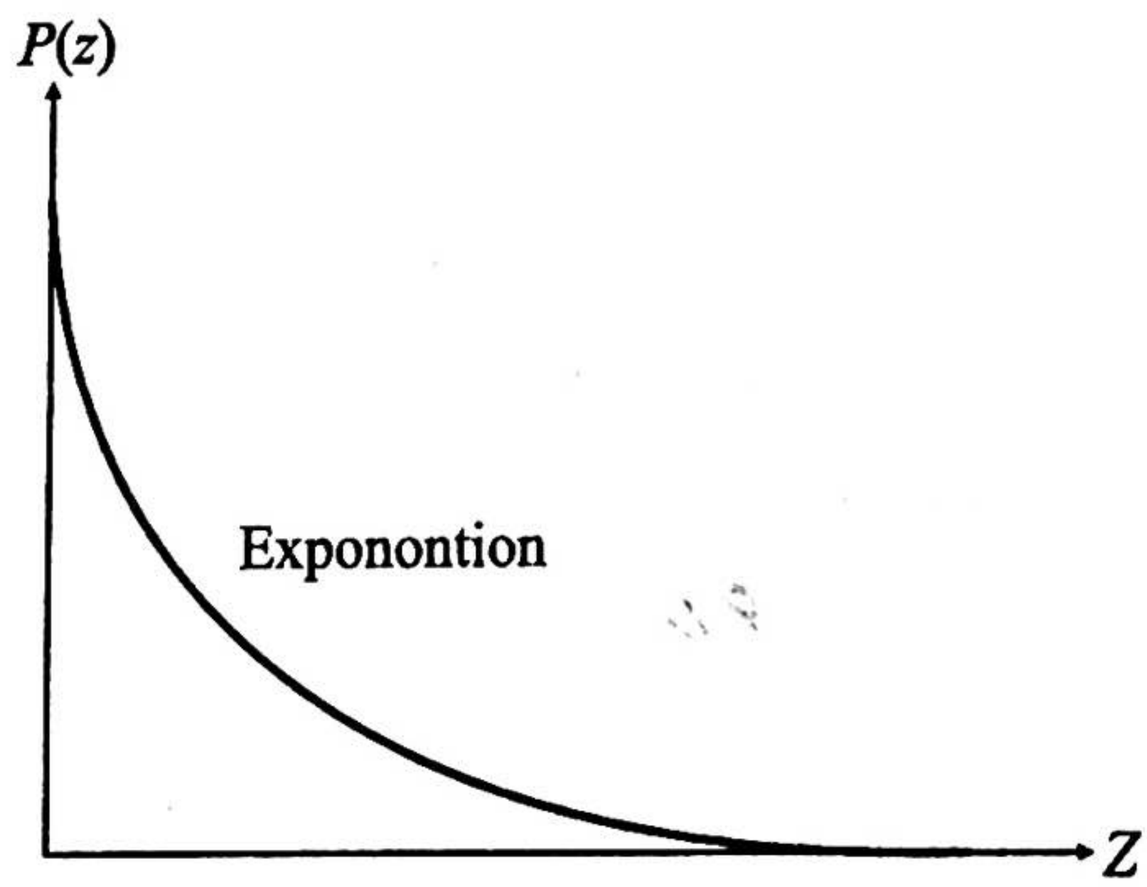


Figure 3.5 Exponential noise PDF

3.6 Uniform Noise PDF

The power spectral density of uniform noise is given as

$$P(z) = \begin{cases} \dfrac{1}{b-a} & a \leq z \leq b \\ 0 & otherwise \end{cases} \quad \ldots(3.13)$$

having means as $$\bar{Z} = \frac{(b-a)^2}{12} \quad \frac{a+b}{2} \quad \ldots(3.14)$$

and variance $$\sigma^2 = \frac{(b-a)^2}{12} \quad \ldots(3.15)$$

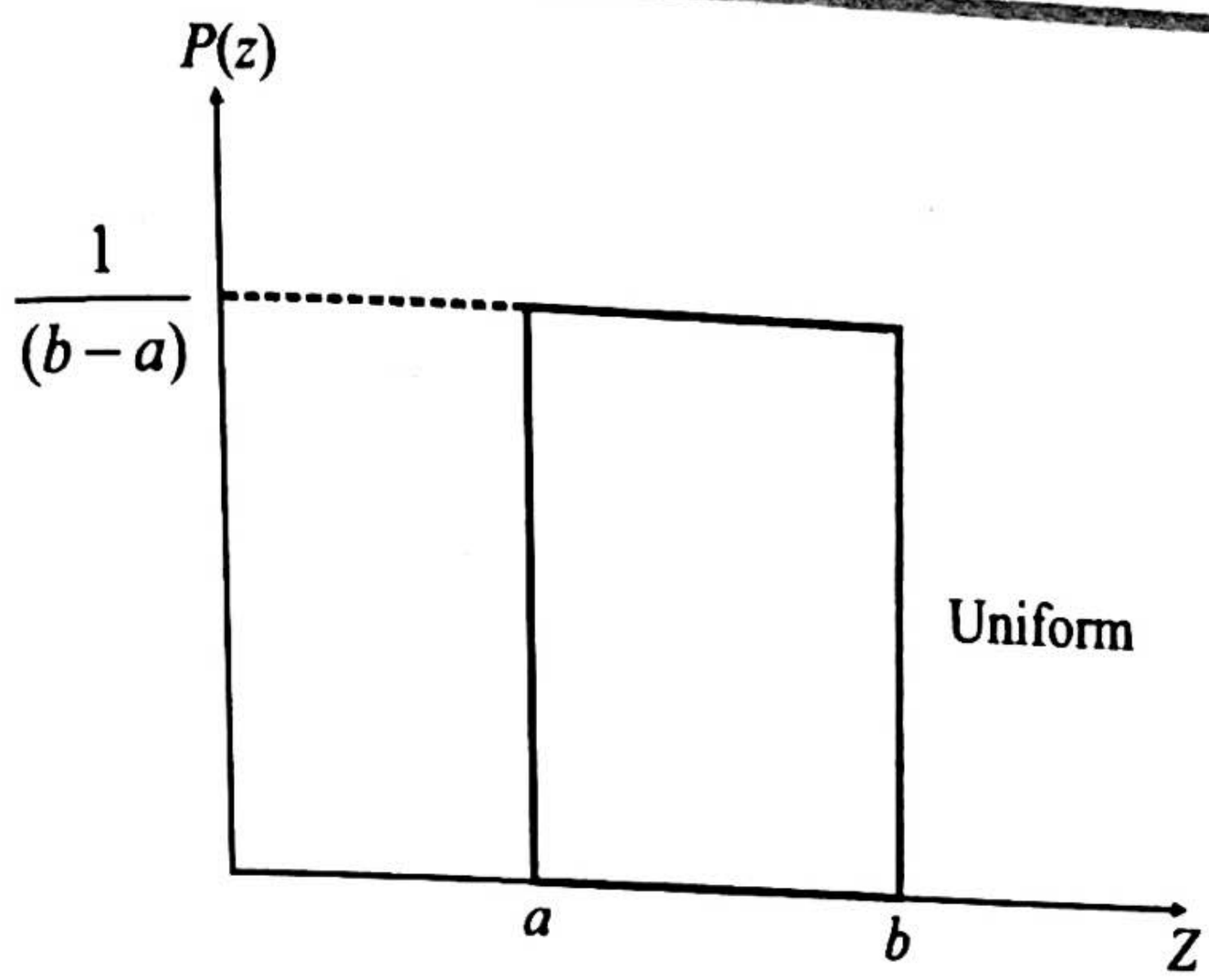


Figure 3.6 Uniform noise PDF

3.7 Impulse Noise (Salt-and-Pepper) PDF

The power density function of bipolar impulse noise is given as

$$P(Z) = \begin{cases} P_a & \text{for } Z = a \\ P_b & \text{for } Z = b \\ 0 & otherwise \end{cases} \qquad \ldots(3.16)$$

If either P_a or P_b is zero then impulse noise is turned as unipolar.

If none of probability is zero and $P_a \simeq P_b$ then impulse noise resembles salt-and-pepper granules due to this the bipolar impulse noise are also known as salt-and-pepper noise. It is also know as data-drop-out and spike noise. It may be negative or positive. In general the negative impulses appear as black (pepper) point and positive impulses appear as white (salt) noise.

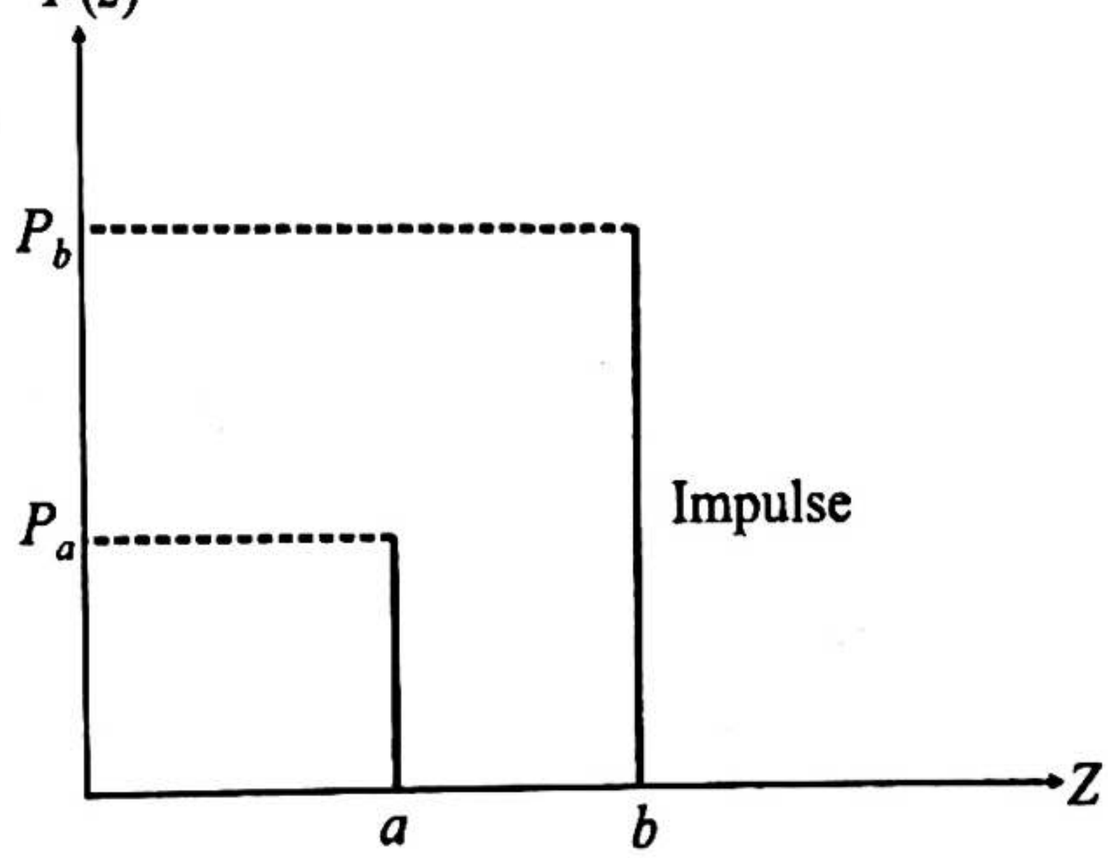


Figure 3.7 PDF of impulse noise

For example it is an 8-bit image it a=0 if give black and b=255 it gives white points.

After considering the PDFs of all noise it is to be noted down that gaussian noise occurs due to electronic circuit noise and sensor noise. Rayleigh is used for characterizing noise phenomena. The exponential and Gamma PDFs are used in laser imaging. Whereas the uniform

PDFs are least used for practical consideration rather then it can use for various random number generators which can be used for simulations. The impulses noises are mostly used for quick transients responses where faulty switching take place in imaging.

3.8 Periodic Noise

In image acquisition process the periodic noise occurs due to electrical or electromechanical interference. This type of noise can be reduced significantly through frequency domain filtering. Such types of noise tends to produce frequency skills that after can be detected, by visual analysis.

3.9 Estimation of Noise Parameters

The noise parameter can be estimated by inspection of the fourier spectrum of the image. These parameters of noise PDF may be knowns from sensor specification. If we have an imaging system then it is easy to calculate or study the characteristics of system noise. Consider a strip or sub-image (s) and $P_s(z_i)$ be the probability estimate then.

Means of the pixels in S is

$$\overline{Z} = \sum_{i=0}^{L-1} z_i P_s(z_i) \qquad \ldots(3.17)$$

and variance of the pixel in S is

$$\sigma^2 = \sum_{i=0}^{L-1} (Z_i - \overline{Z})^2 P_s(Z_i) \qquad \ldots(3.18)$$

where L is the number of possible intensities is the image.

3.10 Noise Only Spatial Filtering

In the restoration model if only degradation present in an image is noise than degraded images in spatial domain can be given as

$$g(x, y) = f(x, y) + n(x, y) \qquad \ldots(3.19)$$

In frequency domain it is given as

$$G(u,v) = F(u,v)+N(u,v) \qquad \ldots(3.20)$$

In the case of periodic noise we can estimate $N(u,v)$ from spectrum of $G(u,v)$. So we can subtract $N(u,v)$ from $G(u,v)$ to estimate the original image. Such types of filters are discussed below.

3.10.1 Mean Filters

The noise reduction capabilities can be understood very well with such filters. It includes:

(a) **Arithmetic Mean Filter:** The simplest mean filter is arithmetic mean filter. It is capable of computing the average value of the corrupted image $g(x,y)$ in the defined area.

Let S_{xy} is an $m \times n$ sized rectangular subimage window. Then the value of the restored image $\hat{f}(x,y)$ is simply considered as arithmetic mean computed for the region S_{xy}.

i.e.,
$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(x,t) \qquad \ldots(3.21)$$

This filter smooths local variations in an image and noise can be reduced as a result of blurring.

(b) **Geometric Mean Filter:** In geometric mean filter each restored pixel is given by the product of the pixels in the subimage rectangular window, raised to the power $1/mn$.

The restored image is geometric mean filter is given as

$$\hat{f}(x,y) = \left[\prod_{(s,t) \in S_{xy}} g(x,y) \right]^{\frac{1}{mn}} \qquad \ldots(3.22)$$

It gives smoothing comparable to arithmetic mean filter but the image details are less lossed.

(c) **Harmonic Mean Filter:** Harmonic mean filters are best suited for salt noise and can be used with other noise gaussian.

It is given as

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}} \qquad \ldots(3.23)$$

(d) **Contraharmonic Mean Filter:** The contraharmonic filters are well suited for reducing or untruly eliminating the effects of salt and pepper noise. It is given as

$$\hat{f}(x,y) = \frac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q}} \qquad \ldots(3.24)$$

Where Q is the order of the filter. For positive values of Q filter eliminates pepper noise while for negative values of Q filter eliminates salt noise. It is possible to eliminate both noises simultaneously.

Any contraharmonic filter can be reduces to an arithmetic mean filter if $Q=0$ while to an harmonic mean filter if $Q=-1$.

After having discussion of all mean filters it is observed that arithmetic and geometric mean filters are well suited for random noise i.e, for gaussian or uniform noise. The harmonic mean filter are best suited for salt noise but not for pepper noise the other hand contraharmonic filters are suited for impulse noise with proper selection of dark or light noise to select proper sign of Q.

3.10.2 Order Statistic Filters

Order statistic or nonlinear filters are spatial filters whose response is based on ordering of the values of the pixels contained in an image area enclosed by the filter. The ordering (ranking) determines the response of the filters.

3.10.3 Median Filter

The median filter replaces the value of a pixel by the median of the intensity levels is the subimages (neighbourhood) pixel. It is mostly used order-statistic filter available.

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{Median} \{g(s,t)\} \qquad \ldots(3.25)$$

These median filters are used to provide excellent noise-reduction capabilities which gives considerably less blurring. Such filters are more effective in presence of bipolar and unipolar impulse noise.

3.10.4 Maximum and Minimum Filters

The median filters are used to represent upto 50th percentile of a ranked set of number.

If we use 100th percentile results in the ranked set of a number than filter is called max filter and it is given as

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{Median}} \{g(s,t)\} \qquad \ldots(3.26)$$

This max filters are used for finding brightest point in an image and reduces pepper noise.

On the other hand the 10th percentile results in the ranked set of number is called min filter.

And min filter is given as

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{Min}} \{g(s,t)\} \qquad \ldots(3.27)$$

These min filter are used to find the darkest point in an image and reduces salt noise.

3.10.5 Midpoint Filter

These types of filters are used to compute midpoint of max and min filters.

i.e.,

$$\hat{f}(x,y) = \frac{1}{2}\left[\underset{(s,t)\in S_{xy}}{\max\{g(s,t)\}} + \underset{(s,t)\in S_{xy}}{\min\{g(s,t)\}}\right] \qquad \ldots(3.28)$$

Such filters are best suited for random distributed noise i.e., for gaussian or uniform noise.

3.10.6 Alpha-Trimmed Mean Filter

If we delete some intensity values from the subimage neighbourhood then averaging these remaining pixel is called alpha-trimmed mean filters. Let we delete $d/2$ lowest and $d/2$ highest intensity values in S_{xy} then remaining $(mn\text{-}d)$pixels can be represented by $g_r(s,t)$. So, for a alpha trimmed mean filter

$$\hat{f}(x,y) = \frac{1}{(mn-d)} \sum_{(s,t)\in S_{xy}} g_x(s,t) \qquad \ldots(3.29)$$

where d is from 0 to $(mn\text{-}1)$.

This alpha trimmed mean filter can be reduces to arithmetic mean filter if we consider d=0, and to median filter if we consider d=$(mn\text{-}1)$.

3.11 Adaptive Filters

The adaptive filters are those filters whose behaviour changes as per statistical characteristic of the image at region defined by the $m\times n$ rectangular window S_{xy}. They have superior performance with respect to other filters.

3.11.1 Adaptive Local Noise Reduction Filter

Any adaptive filter is based on two reasonable parameters i.e., on its mean and variance. Mean gives a measure of average intensity in the region over which the mean is computed while the variance gives a measure of contrast in that region.

The adaptive filter operates on a local region S_{xy} and the response of these filter is based on following quantities.

(a) At $g(x,y)$ which is the value of the noisy image.

(b) At σ_n^2 which is the variance of the noise.

(c) m_L which is the local mean of the pixels is S_{xy}

(d) σ_L^2 which is the local variance of the pixel.

Our prime concern is to get the behavior of the filter as stated below:

(a) It should satisfy trivial, zero noise case in which $g(x,y)$ is equal to $f(x,y)$ and σ_n^2 is zero i.e., filter should return the value of $g(x,y)$.

(b) High local variance should be preserved and should return a value close to $g(x,y)$.

(c) If there are two equal variances then filter return the arithmetic mean value of the pixels in S_{xy}.

Based on these assumptions the adaptive local noise reduction filter is given as

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_n^2}{\sigma_L^2}\left[g(x,y) - m_L\right] \quad \ldots(3.30)$$

It is to be assumed that $\sigma_n^2 \leq \sigma_L^2$.

3.11.2 Adaptive Median Filter

The adaptive median filtering can handle impulse noise with larger probabilities. Such types of filter seeks to present details while smoothing non-impulse noise. Which can not be made by some traditional median filters. Any adaptive median filtering works on two stages to compile an algorithm.

Stages A consists of

$$A_1 = Z_{med} - Z_{min}$$

Z_{min} Z_{med} Z_{max} $A_2 = Z_{med} - Z_{max}$

If $A_1 > 1$ and $A_2 < 0$, goto stage B

Else

Increase the window size.

If window size $\leq S_{max}$, repeat stage B

Else

Output Z_{med}

and

Stage B consists of

$$B_1 = Z_{xy} - Z_{min}$$

Zmin Zxy Zmax $B_2 = Z_{xy} - Z_{max}$

If $B_1 > 0$ and $B_2 < 0$,

Output Z_{xy}

Else

Output Z_{med}

Here Z_{min} = Minimum intensity value is S_{xy}

Z_{max} = Maximum intensity value is S_{xy}

Z_{med} = Median of intensity values is S_{xy}

Z_{xy} = Intensity value at coordinates (x,y)

and S_{max} = Maximum allowed size of S_{xy}

This also has three main purposes

(i) To remove salt and pepper noise.

(ii) To provide smoothness of other noise which may not be impulse and

(iii) To overcome distortion i.e., to remove excessive thinning or thickening of object boundaries.

The stage A serves the purpose of determining whether medium filter output Z_{med} is an impulse or not.

If $Z_{min} < Z_{med} < Z_{max}$ then Z_{med} may not be an impulse at this condition move on to stage B and verify if the point in the centre of the window Z_{xy} is itself an impulse.

If $B_1 > 0$ and $B_2 < 0$ is valid then

$Z_{min} < Z_{xy} < Z_{max}$ and Z_{xy} may not be an impulse.

If above condition is not valid then either $Z_{xy} = Z_{min}$ or $Z_{xy} = Z_{max}$. In either case value of pixels is an extreme and output is Z_{med}.

Each time the algorithm outputs a values, the window S_{xy} is moved to the next location in the image. The algo is reinitialize and applied to the pixels in the new location.

3.12 Inverse Filtering

The inverse filtering is used to restore an image directly. Here we compute an estimate $\hat{F}(u,v)$ of the transform of the original image by

dividing the transform of the degraded image $G(u,v)$.

So,
$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \quad \ldots(3.31)$$

Since $G(u,v) = H(u,v)\,F(u,v) + N(u,v)$ is the degradation in frequency domain.

So divide this equation by $H(u,v)$ gives

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

$$\Rightarrow \quad \hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \quad \ldots(3.32)$$

This expression give us that even if we know the degradation function we can not recover the undegraded image since $N(u,v)$ is unknown.

The small value problems or get zero is to limit the filter frequency to values near the origin.

3.13 Minimum Mean Square Error (Wiener) Filtering

The inverse filtering does not gives the provision for handling noise. So the wiener filtering is an approach for the degradations function and statistical characteristics of noise into the restoration process.

The main objective of wiener filtering is to find an estimate $\hat{f}$ of uncorrupted image f by considering images and noise as random variables and mean square error between them is minimized.

The mean square error is

$$e^2 = E\left\{(f - \hat{f})^2\right\} \quad \ldots(3.33)$$

Here we have assumed that noise and images are uncorrelated and intensity levels in the estimate are linear function of the levels in the degraded images.

The minimum of the error function is

$$\hat{F}(u,v) = \left[\frac{H^*(u_1 v)S_f(u,v)}{S_f(u,v)\,|H(u,v)|^2 + S_n(u,v)}\right]G(u,v)$$

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{S_n(u,v)}{S_f(u,v)}}\right]G(u,v)$$

or

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{s_n(u,v)}{S_f(u,v)}}\right] G(u,v) \quad \ldots(3.34)$$

Where

$H(u,v) \rightarrow$ Degradation function

$H^*(u,v) \rightarrow$ Complex Conjugate of $H(u,v)$

$(H(u,v)^2 \rightarrow H^*(u,v)H(u,v)$

$S_n(u,v) = |N(u,v)|^2 =$ Power spectrum of noise.

$S_f(u,v) = |F(u,v)|^2 =$ Power spectrum of undegraded image.

It gives that product of a complex quantity with its conjugate is equal to the magnitude of the complex quantity squared. It is known as wiener filter.

The term inside the bracket is known as minimum mean square error filter or the least square error filter.

The restored image in spatial domain is given by the inverse fourier transform, of the frequency domain estimate $\hat{F}(u,v)$.

In this expression, the wiener filter reduces to inverse filter if noise is zero. The signal to noise ratio is frequency domain is

$$\text{SNR} = \frac{\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}|F(u,v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u,v|^2} \quad \ldots(3.35)$$

It gives the measure of the level of information bearing signal power to the level power.

Image with low noise tend to have a high SNR and high noise leads to have lower SNR.

Equation (3.33) in terms of summation is

Mean square error

$$\text{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2 \quad \ldots(3.36)$$

If we consider a restored image to be signal and difference between this image and original image as noise then SNR in spatial domain

$$\text{SNR} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)-\hat{f}(x,y)\right]^2} \quad \ldots(3.37)$$

If we are consider square root then it is termed as root mean square signal to noise ratio and root mean square error.

For spectrally white noise the spectrum $|N(u,v)|^2$ is constant.

If we don't have any information about the noise then we can estimate this approximation by

$$\hat{f}(u,v) = \left[\frac{1}{H(u,v)}\cdot\frac{|H(u,v)|^2}{|H(u,v)|^2+K}\right]G(u,v) \quad \ldots(3.38)$$

Where k is a specified constant.

Above these discussion we find that noise in the inverse filtered image is so strong that its structure is in the direction of the deblurring filter. While the wiener filter five hint as to image content.

3.14 Homomorphism Filtering

Images are sometimes been acquired under poor illumination. Under this condition, the same uniform region will appears brighter on some areas and darker on others. This undesired situation will leads to several problem in computer vision based system. The pixels might be misclassified, leading to wrong segmentation results, and therefore contribute to inaccurate evaluation or analysis from the system. Therefore, it is very crucial to process this type of images first before they are fed into the system.

One of the popular methods used to enhance or restore the degraded images by uneven illumination is by using homomorphic filtering. This technique uses illumination-reflectance model in its operation. This model consider the image is been characterized by two primary components. The first component is the amount of source illumination incident on the scene being viewed $i(x, y)$. The second component is the reflectance component of the objects on the scene $r(x, y)$. The image $f(x, y)$ is then defined as

$$f(x, y) = i(x, y)\, r(x, y) \quad \ldots(3.39)$$

In this model, the intensity of $i(x, y)$ changes slower than $r(x, y)$. Therefore, $i(x, y)$ is considered to have more low frequency components than $r(x, y)$. Using this fact, homomorphic filtering technique aims to reduce the significance of $i(x, y)$ by reducing the low frequency components of the image. This can be achieved by executing the filtering process in frequency domain. In order to process an image in frequency domain, the image need first to be transformed from spatial domain to frequency domain. This can be done by using transformation functions, such as Fourier transform. However, before the transformation is taking place, logarithm function has been used to change the multiplication operation of $r(x, y)$ with $i(x, y)$ in equation (3.39) into addition operation.

In general, homomorphic filtering can be implemented using five stages, as stated as follows:

Stage 1. Take a natural logarithm of both sides to decouple $i(x, y)$ and $r(x, y)$ components

$$z(x, y) = ln\ i(x, y) + ln\ r(x, y) \qquad ...(3.40)$$

Stage 2. Use the Fourier transform to transform the image into frequency domain:

$$Z(u, v) = F_i(u, v) + F_r(u, v) \qquad ...(3.41)$$

Where $F_i(u, v)$ and $F_r(u, v)$ are the Fourier transforms of $lni\ (x, y)$ and $lnr(x, y)$ respectively.

Stage 3. High pass the $Z(u, v)$ by means of a filter function $H(u,v)$ in frequency domain, and get a filtered version S(u,v) as the following:

$$S(u, v) = H(u, v)\ Z(u, v) = H(u, v)$$
$$F_i(u, v) + H(u, v)\ F_r(u, v) \qquad ...(3.42)$$

Stage 4. Take an inverse Fourier transform to get the filtered image in the spatial domain:

$$s(x, y) = F^{-1}\ \{S(u, v)\} = F^{-1}\ \{H(u, v)F_i(u, v)$$
$$+ H(u, v)\ F_r(u, v)\}z \qquad ...(3.43)$$

State 5. The filtered enhanced image $g(x, y)$ can be obtained by using the following equations:

$$g(x, y) = \exp\{s(x, y)\} \qquad ...(3.44)$$

However, there are several variations of equations that have been used to present $H(u, v)$ in equation (3.43).

3.14.1 Homomorphic Filtering Equations

This filter has circularly symmetric curve shape, centred at $(u, v) = (0,0)$ coordinates in frequency domain. This filter is modified from Gaussian highpass filter, which is known as Difference of Gaussian (DoG) filter.

(a) **Homomorphic Filtering:** An image as a function can be expressed as the product of illumination and reflectance components as follows:

$$F(x, y) = I(x, y) * R(x, y) \quad \ldots(3.46)$$

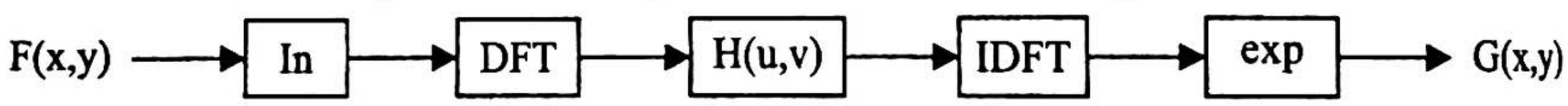
Equation (3.46) cannot be used directly to operate separately on the frequency components of illumination and reflectance because the Fourier transform of the product of two functions is not separable. Instead the function can be represented as a logarithmic function wherein the product of the Fourier transform can be represented as the sum of the illumination and reflectance components as shown below:

$$ln(x, y) = ln(I(x, y)) + ln(R(x, y)) \quad \ldots(3.47)$$

The Fourier transform of equation (3.47) is

$$Z(u, v) = Fi(u, v) + Fr(u, v) \quad \ldots(3.48)$$

The fourier transformed signal is processed by means of a filter function $H(u,v)$ and the resulting function is inverse fourier transformed. Finally, inverse exponential operation yields an enhanced image. This enhancement approach is termed as homomorphic filtering. The whole operation is expressed as a block diagram below:



3.14.2 Implementation of Homomorphic Filtering

Consider an image with 256 × 256 pixels where the pixels have varying intensity. The pattern goes from dark to light as we go from left to right. This is also called a horizontal intensity and it can be visualized by creating a matrix of size 256 with non-uniform illumination. This is the first illumination pattern under investigation. figure 3.8 shows the horizontal illumination pattern of 256 × 256 pixels.

Illumination Pattern

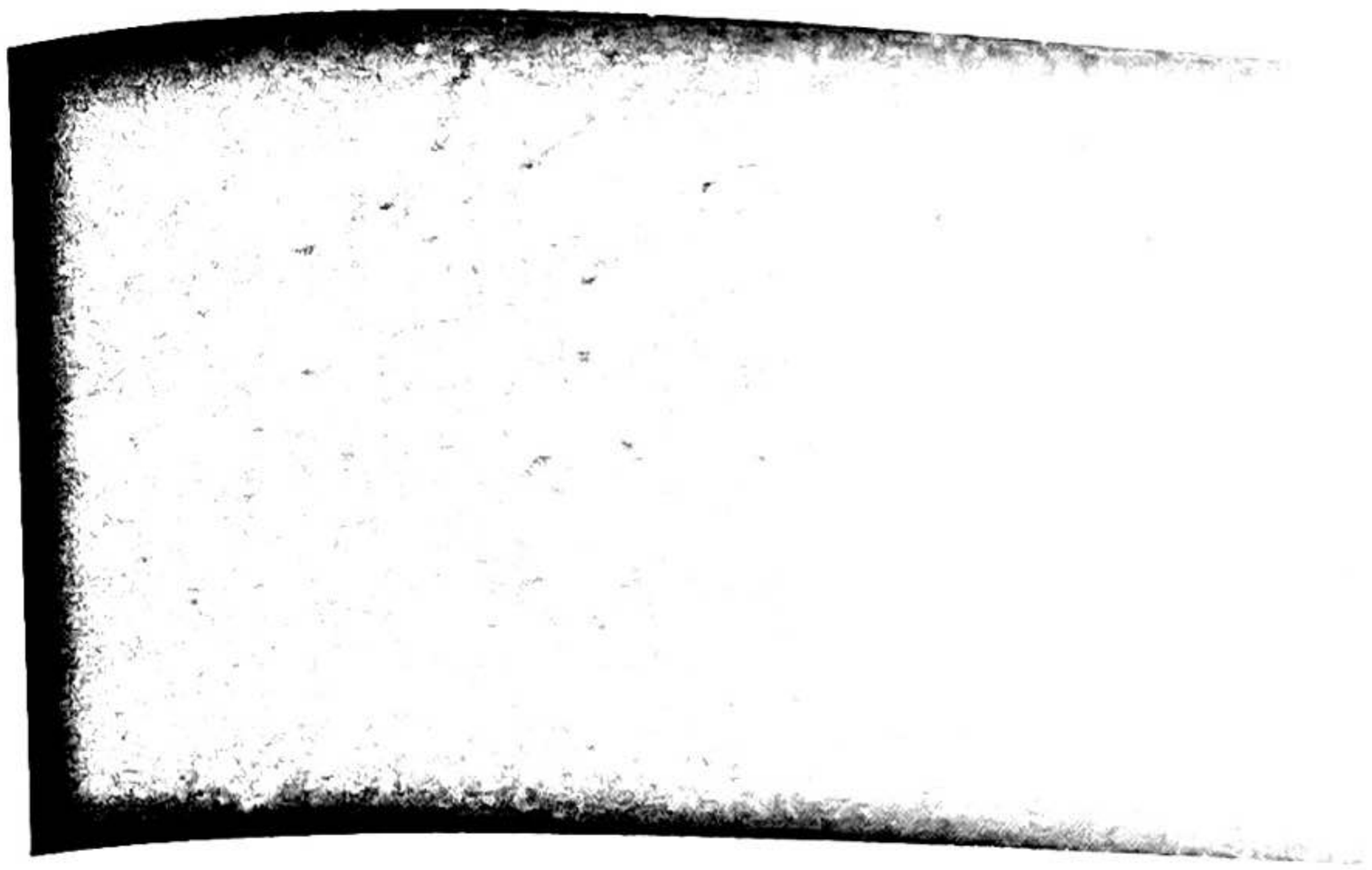


Figure 3.8 Illumination pattern, Image with varying intensity

The image is then multiplied with the non-uniform illumination. In the real world the corrupted images are expected to be the way the image that is corrupted by this type of non-uniform illumination. The corrupted image under investigation is as shown in Figure 3.9. The figure shows a comparison of the original image and the non-uniformly illuminated image. The aim of the project is to homomorphicly filter out the illumination component to retain the original image.

Original Image

Corrupted Image

Figure 3.9 Left panel - Original image, Right panel - Corrupted image

The histogram of the corrupted image is shown in figure 3.10. As we can see, the histogram is very non-uniform. The areas where it is corrupted densely i.e., with more darkness intensity, the histogram has a greater value than the lesser dense area. This is due to the contribution from the non-uniform illumination component of the image.

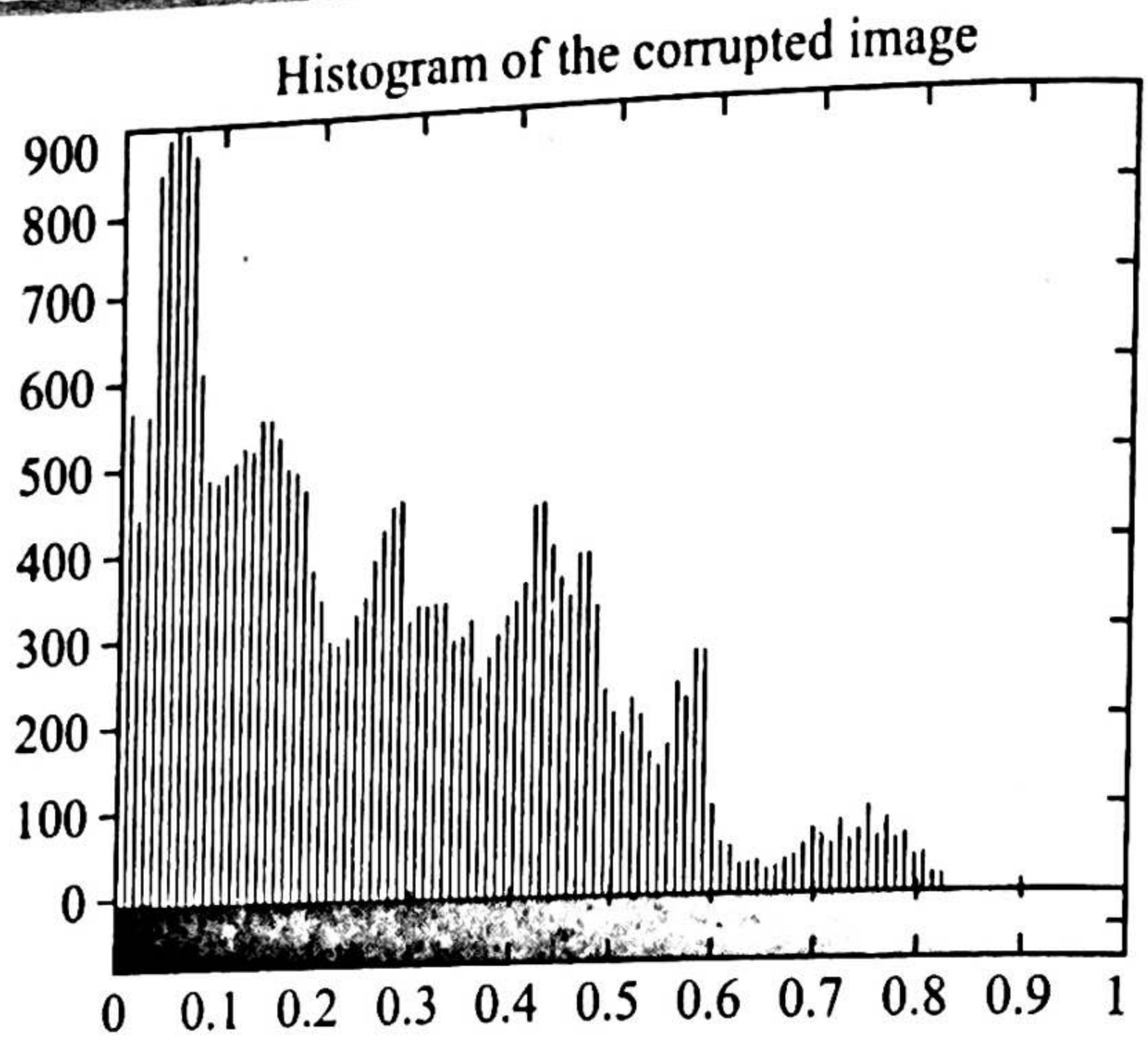


Figure 3.10 Histogram of the non-uniformly illuminated image

Histogram equalization is a technique that divides the bins of the histogram and spreads the image uniformly to some extent. The restored image as a result of the histogram equalization is shown in figure 3.11. The overall intensity is much lighter so that the figure appears better than the corrupted image i.e., the darker portions of the corrupted images are visible clearly. However, the results are not perfect due to the continued presence of the non-uniform illumination.

Original Image

Image restored after histogram equalization

Figure 3.11 Result of histogram equalization, Left pane – Original image,

(a) **Right Panel-Restored Image:** The illumination component of an image generally is characterized by slow spatial variations, while reflectance component tends to vary abruptly. These characteristics lead to associating the low frequencies of the Fourier transform of the

logarithm of an image with illumination and high frequencies with reflectance, which are used as an advantage in image enhancement.

Using the technique of homomorphic filtering, assuming that we know the image, we can filter out all kinds of illumination and restore the image. The frequency characteristics of the illumination component are shown in figure 3.12. The frequency component of the log of the illumination is then subtracted from the frequency characteristics of the corrupted image.

A good deal of control can be gained over the illumination and reflectance components with a homomorphic filter assuming that we know the image. This control requires specification of a filter function $H(u, v)$ that affects the low and high frequency components of the fourier transform in different ways.

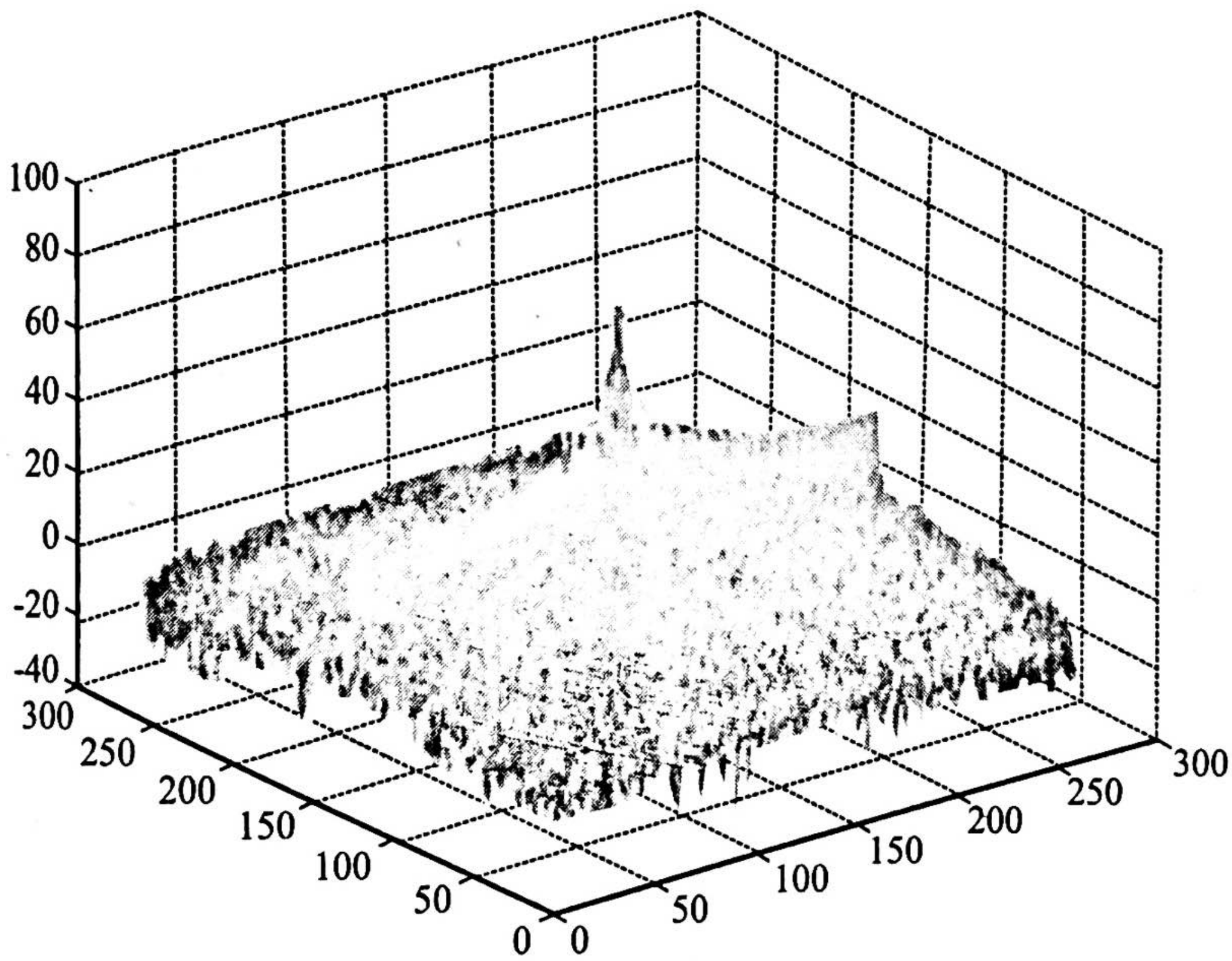


Figure 3.12 Frequency characteristics of illumnination

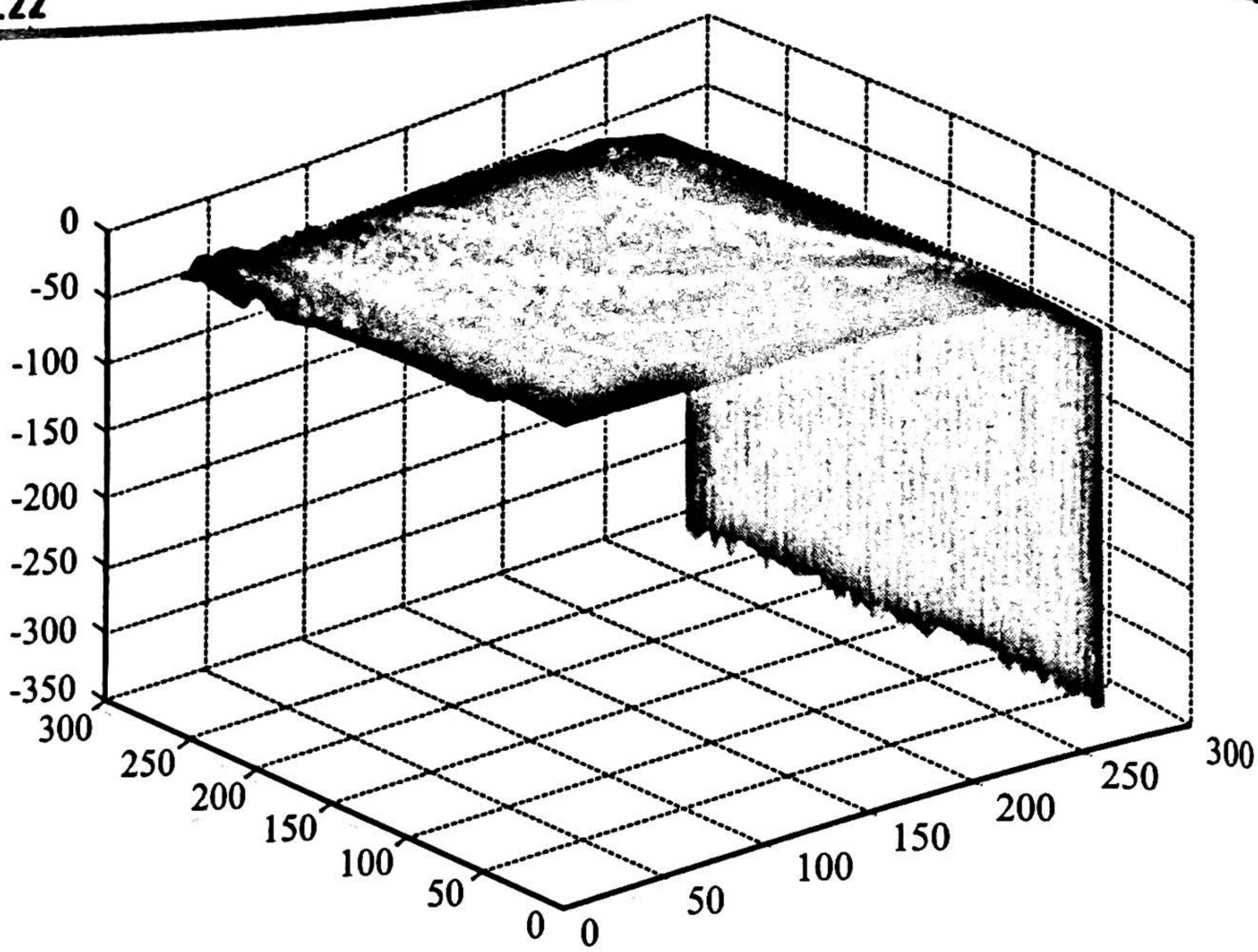


Figure 3.13 Frequency characteristic of homomorphic filter

The parameters of the filter are chosen in such a way so that the filter function tends to decrease the contribution made by the low frequencies (illumination) and amplify the contribution made by high frequencies (reflectance). The net result is simultaneous dynamic range compression and contrast enhancement. One such filter designed for this type of non-uniform illumination is shown in figure 3.13.

3.14.3 Metrics for Deciding the Performance of Homomorphic Filtering

The comparison of frequency characteristics of the original image and the image restored with the help of the homomorphic filtering is chosen as a metric. The resulting image displays frequency characteristics similar to that of the original filter. Figure 3.14 shows the frequency characteristics of the original image in the left end compared to the frequency characteristics of the restored image in the right end of the plot and frequency characteristics of the corrupted image in the bottom.

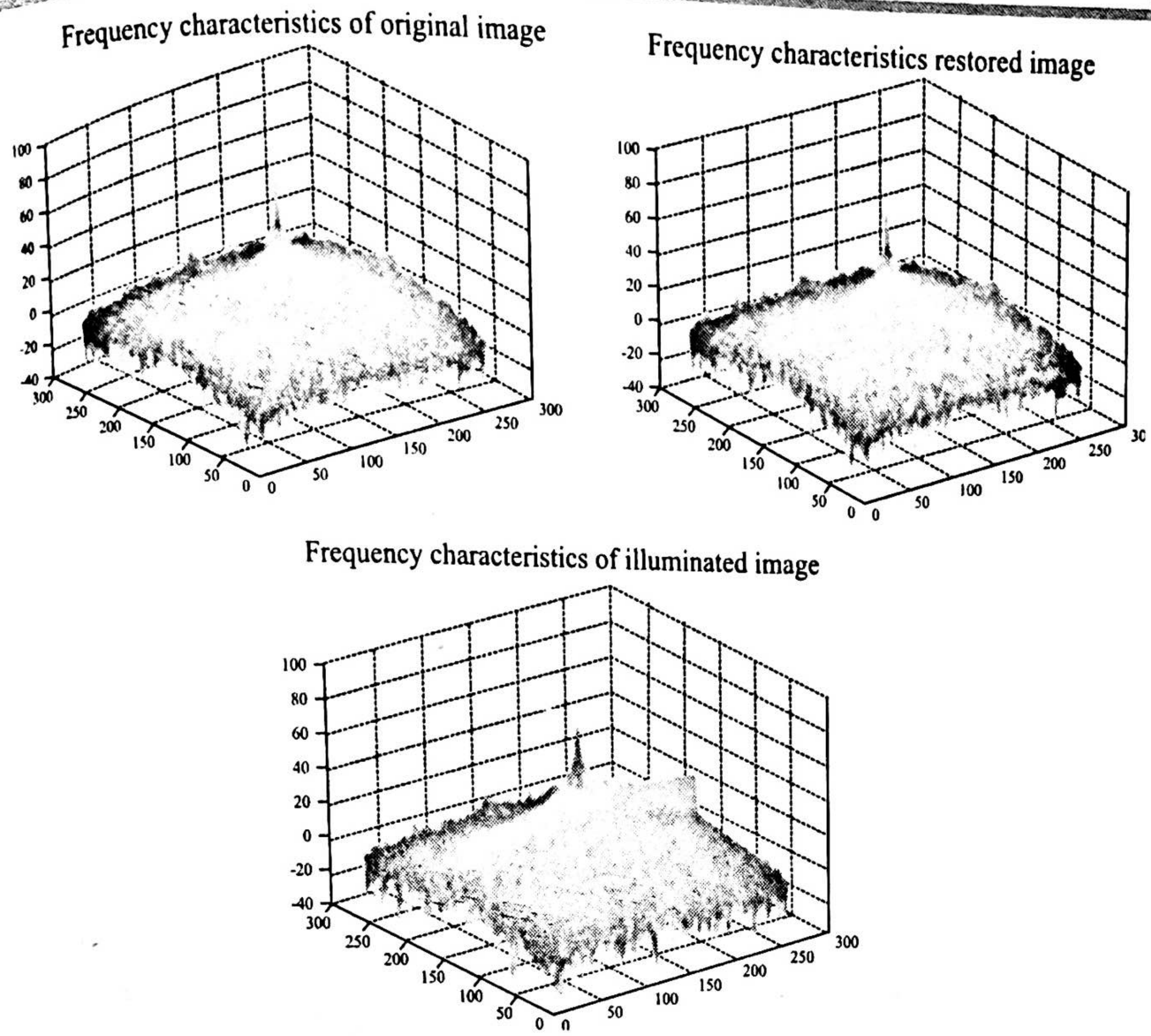


Figure 3.14 Original image frequency characteristics (left) with restored image frequency characteristics (right) in comparison with the frequency characteristics of the illuminated image (bottom).

The restored images are also compared in the spatial domain and the images are displayed in the figure 3.15. The restored image looks very close to the original image i.e., the illumination component is completely filtered out so that the non-uniform pattern is no more visible to the naked eye. This metric proves the method of homomorphic filtering to be a very efficient technique.

Figure 3.15 Original image (left panel), Restored image (right panel)

Another metric in this case is the error performance matrix or the error plot of the images, which decides the performance of the homomorphic filtering. The error plot is shown in figure 3.16.

Difference of original and restored images

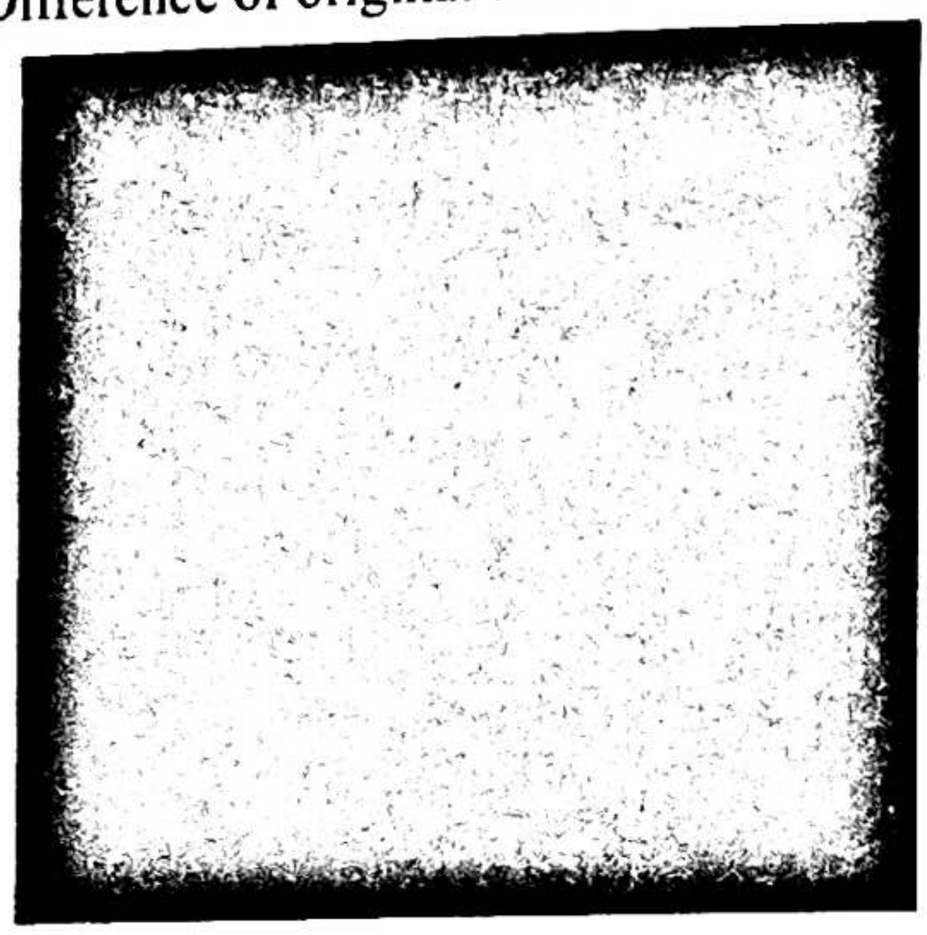
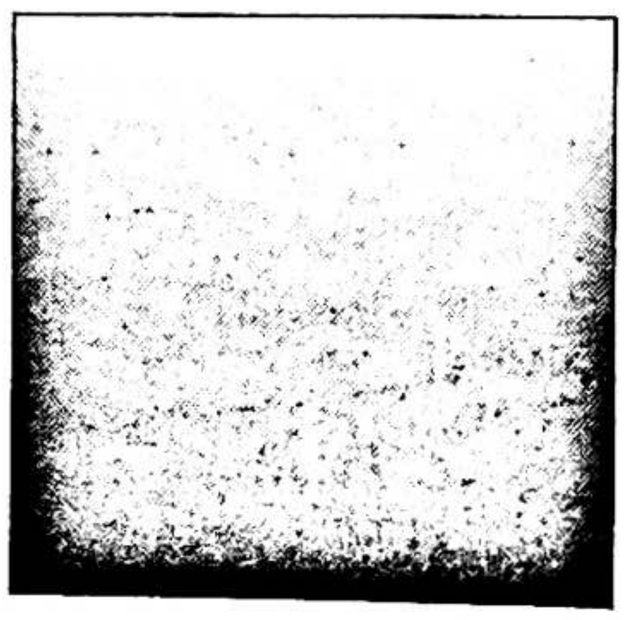


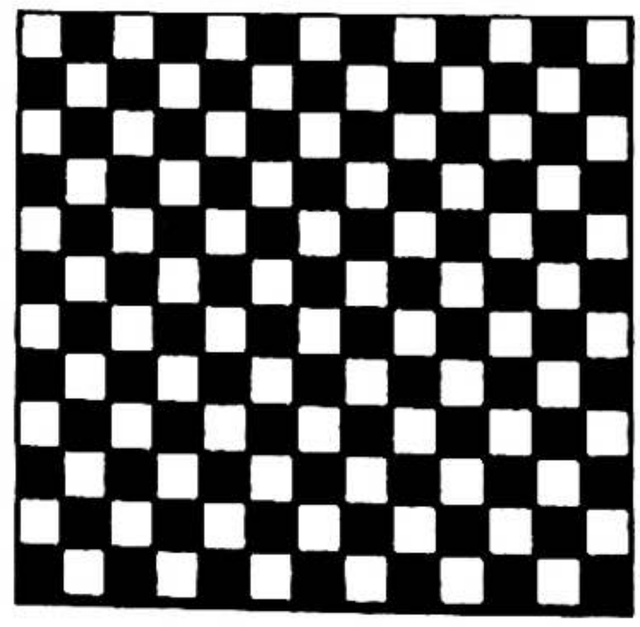
Figure 3.16 Difference of original and restored images

Similar, experiments were done with different types of illumination patterns. The previous illumination pattern was rotated 90 degrees to obtain a vertical illumination and also 45 degrees tilt and a sine wave pattern was tried as illumination. The following figure 3.17 shows the different types of illumination patterns tried with the technique of homomorphic filtering.

Illumination pattern



Illumination pattern



Rotated illumination pattern

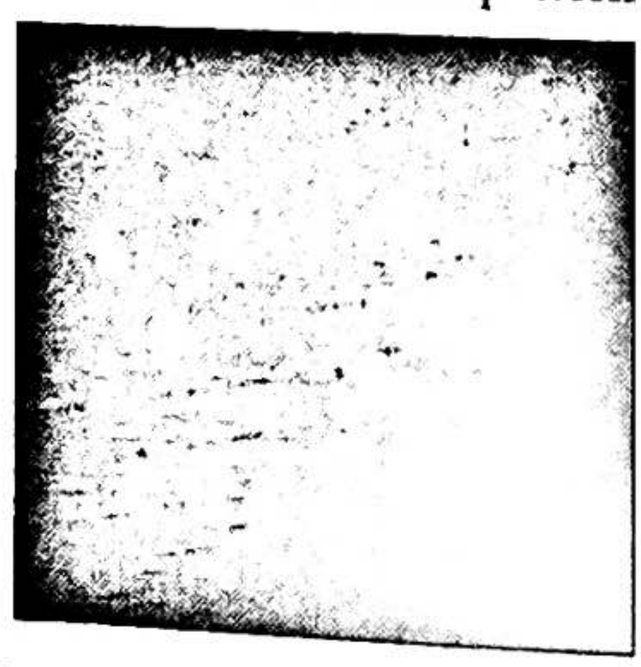


Figure 3.17 Different illumination patterns, Top panel left – vertical illumination, Top right panel – sine wave pattern, Bottom panel – 45 degrees rotated illumination pattern

(a) **Experiments with Sine Wave Illumination Pattern:** The image with the illumination pattern in the form of sine wave was investigated and is shown in the figure 3.18. Here also, the results of homomorphic filtering are found to be much effective than other conventional methods. Figure 3.18 displays the frequency characteristics of the original image in the left end and frequency characteristics of the restored image in the right end along with the frequency characteristics of the corrupted image in the bottom.

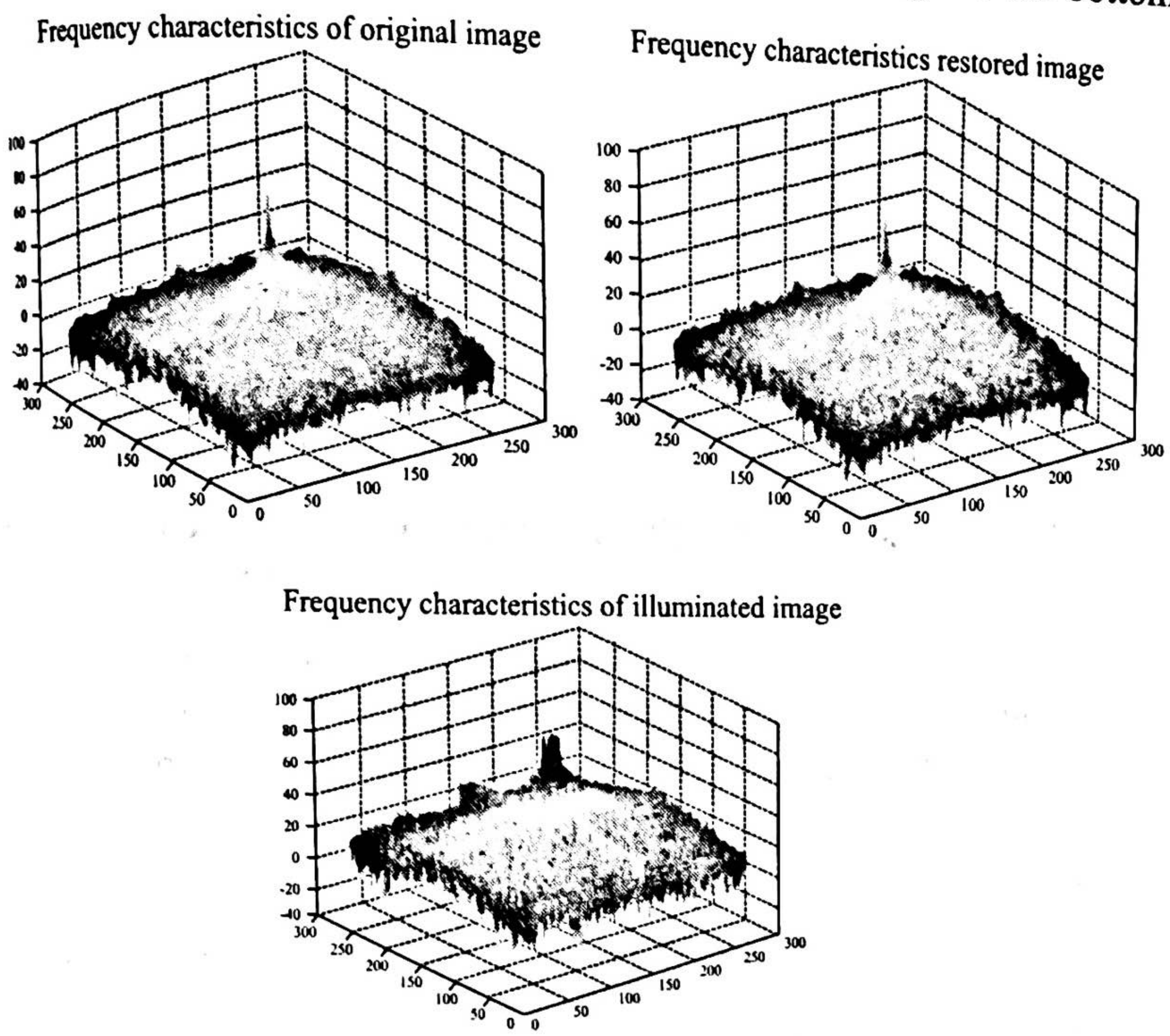


Figure 3.18

homomorphic filter was designed using the multiplicative and logarithmic techniques. The net result is simultaneous dynamic range compression and contrast enhancement. One such filter designed for sinusoidal type of non-uniform illumination is shown in figure 3.19.

(b) Frequency characteristics of filter

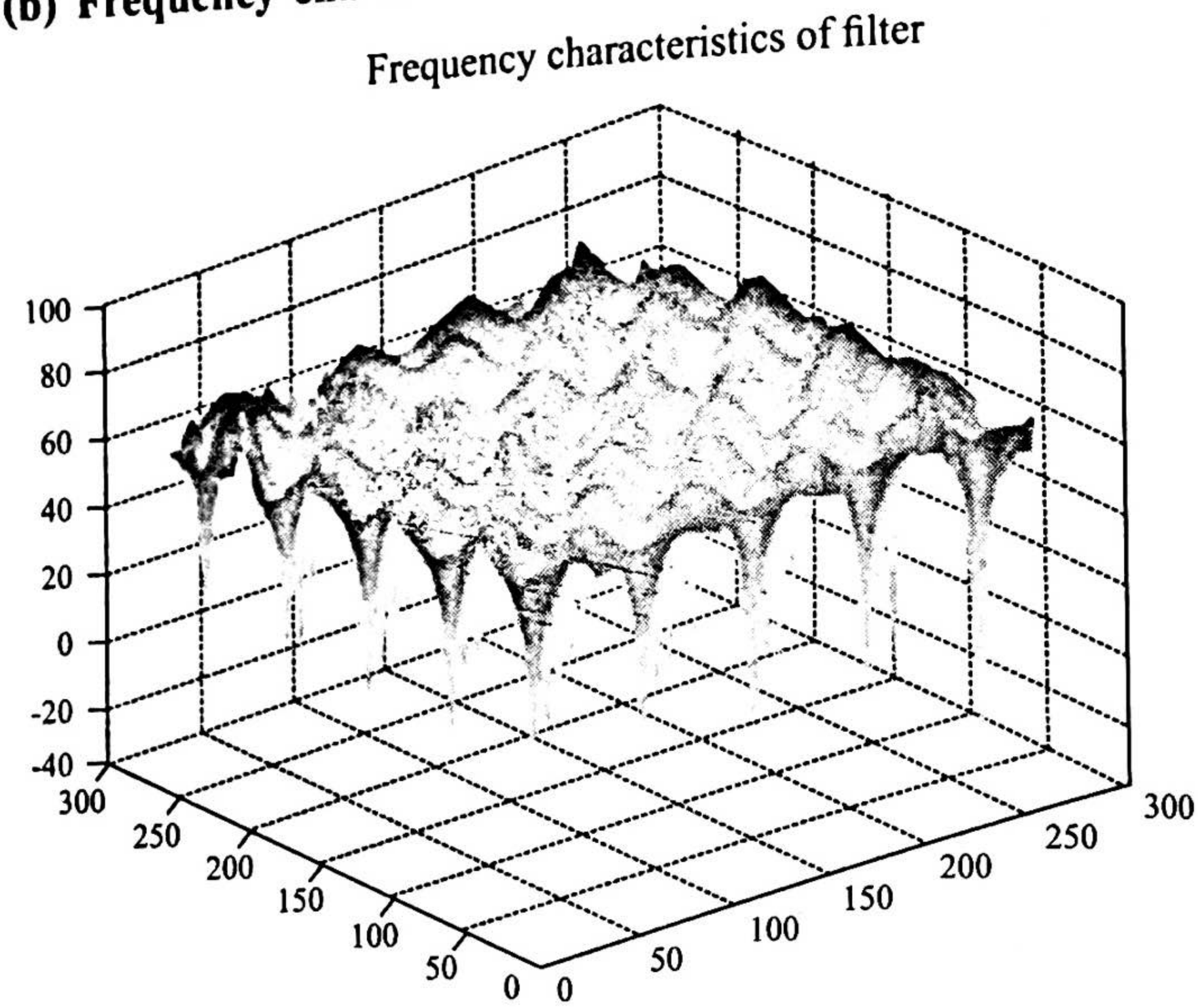


Figure 3.19 Frequency characteristics of homomorphic filter designed for a sinusoidal non-uniform illumination

The original image and the image restored using the homomorphic filtering is also compared in the spatial domain. The result shows that the homomorphic filtering gets rid of the whole non-uniform illumination and the restored image looks much closer to the original image. The results are shown in figure 3.20.

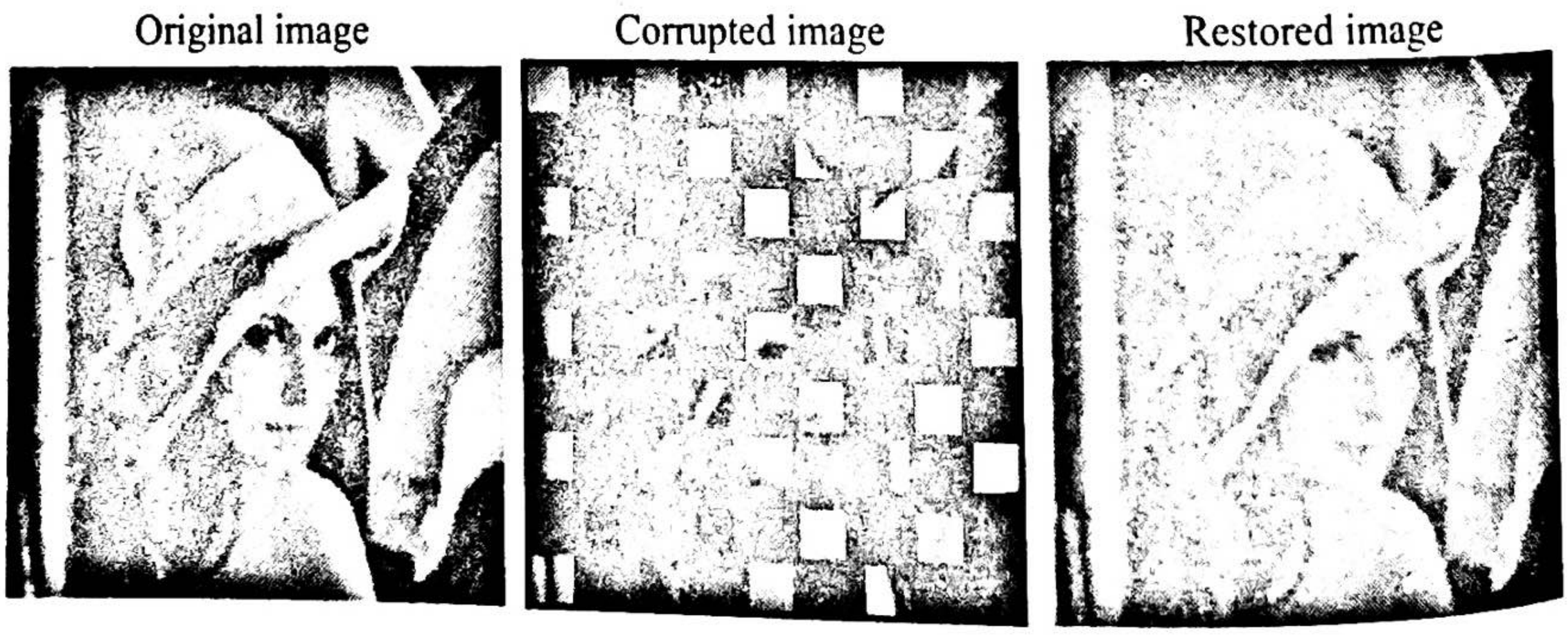


Figure 3.20 Results of homomorphic filtering, Top left – Original, Top right – Corrupted,

3.15 Matlab Source Code

```
clear all; close all;
I = imread('C:\MATLAB6p1\work\lena.jpg');
G = rgb2gray(I);
GM = im2double(G);
GN = 1-GM;
GN = im2uint8(GN);
t = 256;
ill = illumination_v1(t);
product = GM.*ill;
figure;
imshow(GM);
title('Original Image');
figure;
imshow(product);
title('Corrupted Image');
figure;
kk = imhist(product);
imhist(product);
title('Histogram of the corrupted image');
figure;
imhist(kk);
title('Histogram after equalization');
figure;
jj = histeq(product);
imshow(jj);
title('Image restored after histogram
equalization');
log_GM = log(GM + 1);
log_product = log(product + 1);
fft_GM = (fftshift(fft2(log_GM)));
figure;
original_value = (20*log10(abs(fft_GM)));
mesh(20*log10(abs(fft_GM)));
title('Frequency characteristics original image');
fft_product = (fftshift(fft2(log_product)));
figure;
mesh(20*log10(abs(fft_product)));
title('Frequency characteristics of illuminated
```

```
image');
diff_ill = (fft_product - fft_GM);
figure;
mesh(20*log10(abs(diff_ill)));
title('Frequency characteristics ofillumination');
filt_filt = ifft2(diff_ill);
figure;
mesh(20*log10(abs(filt_filt)));
title('Frequency characteristics of filter');
restored = (fft_product - diff_ill);
figure;
desired_value = (20*log10(abs(restored)));
mesh(20*log10(abs(restored)));
title('Frequency characteristics of restored
image');
restored_im = ifft2(restored);
restored_image = (exp(restored_im)) - 1;
figure;
abs_restored_image = abs(restored_image);
imshow(abs_restored_image);
title('Restored Image');
%metrics for testing the performance
final = original_value - desired_value;
sum_ave = det(final);
%error image
error = restored_image - GM;
error1 = im2uint8(error);
figure;
imshow(error1);
title('Error image');
Function file
function[ill] = illumination_v1(t);
p = 1/t:1/t:1;
for k = 1:t,
ill(k,:) = p(:)';
end
figure;
imshow(ill);
title('Illumination Pattern');
p = 1/t:1/t:1;
```

```
for k = 1:t,
ill(k,:) = p(:)';
end
ill = rot90(ill);
figure;
imshow(ill);
title('illumination pattern');
p = 1/t:1/t:1;
for k = 1:t,
il(k,:) = sin(9*p*pi + 0.2);
end
for k = 1:t,
il_1(k,:) = sin(9*p*pi + 0.2);
end
il_1 = rot90(il_1);
ill = il.*il_1;
figure;
imshow(ill);
title('Illumination Pattern');
p = 1/t:1/t:1;
for k = 1:t,
il(k,:) = p(:)';
end
figure;
imshow(il);
title('Illumination Pattern');
ill = imrotate(il,-45,'bilinear','crop');
imshow(ill);
title('Rotated Illumination Pattern');
dia = sqrt(2*(t^2));
p = 1/dia:1/dia:1;
for k = 1:dia,
il(k,:) = p(:)';
end
size_il = size(il)
figure;
imshow(il);
title('Illumination Pattern');
J = imrotate(il,-45,'bilinear','crop');
ill = imcrop(J,[0,0,t,t]);
```

```
size_ill = size(ill)
figure;
imshow(ill);
title('Rotated Illumination Pattern');
p = 1/t:1/t:1;
for k = 1:t,
il(k,:) = exp(p*pi/10);
end
for k = 1:t,
il_1(k,:) = exp(p*pi/10);
end
il_1 = rot90(il_1);
ill = il.*il_1;
figure;
imshow(ill);
title('Illumination Pattern');
```

REVIEW QUESTIONS

Q.1 Explain image restoration model in brief.

Q.2 What is noise PDF. Explain erelong noise.

Q.3 What is impulse noise PDF.

Q.4 What is minimum mean square error filtering?

Q.5 Differentiate between order statistic and adaptive filters.

○○○

CHAPTER 4

Data Compression and Redundancy

OBJECTIVES

- Introduction
- Fundamentals of Visual Data Compression
- Image Compression and Reconstruction
- Measurements for Compression Methods
- Quantization
- JPEG Compression

4.1 Introduction

Image compression deals with reducing the amount of data required to represent a digital image by removing of redundant data.

Images can be represented in digital format in many ways. Encoding the contents of a 2-D image in a raw bitmap (raster) format is usually not economical and may result in very large files. Since, raw image representations usually require a large amount of storage space (and proportionally long transmission times in the case of file uploads/downloads). The need to save storage space and shorten transmission time, as well as the human visual system tolerance to a modest amount of loss, have been the driving factors behind image compression technique.

Compression methods can be lossy, when a tolerable degree of deterioration in the visual quality of the resulting image is acceptable, or lossless, when the image is encoded in its full quality. The overall results of the compression process, both in terms of storage savings usually expressed numerically in terms of Compression Ratio (CR) or bits per pixel (bpp) as well as resulting quality loss (for the case of lossy techniques) may vary depending on the technique, format, options such as the quality setting for JPEG, and the image contents. As a general guideline, lossy compression should be used for general purpose photographic images, whereas lossless compression should be preferred when dealing with line art, technical drawings, cartoons, etc. or images in which no loss of detail may be tolerable.

4.2 Fundamentals of Visual Data Compression

The general problem of image compression is to reduce the amount of data required to represent a digital image or video and the underlying basis of the reduction process is the removal of redundant data. Mathematically, visual data compression typically involves transforming (encoding) a 2-D pixel array into a statistically uncorrelated data set. This transformation is applied prior to storage or transmission. At some later time, the compressed image is decompressed to reconstruct the original image information.

4.2.1 Redundancy

A digital image, or "bitmap", consists of a grid of dots, or "pixels", with each pixel defined by a numeric value that gives its colour. The term

data compression refers to the process of reducing the amount of data required to represent a given quantity of information. Different amounts of data might be used to communicate the same amount of information. If the same information can be represented using different amounts of data, it is reasonable to believe that the representation that requires more data contains what is technically called data redundancy.

Data redundancy is one of central issue in digital image compression. If $n1$ and $n2$ denote the number of information carrying units in two data sets that represent the same information, the relative data redundancy R_D of the first data set can be defined as

$$R_D = 1-1/C_R$$

Where C_R, commonly called the compression ratio and it is given as

$$C_R = n1/n2$$

Case 1. $n2 = n1$ and $R_D = 0$, indicating that the first representation of the information contains no redundant data.

Case 2. When $n2 << n1$, $C_R > \infty$ and $R_D > 1$, implying significant compression and highly redundant data.

Case 3. When $n2 >> n1$, $C_R > 0$ and $R_D > \infty$, indicating that the second data set contains much more data than the original representation.

Generally, C_R=10(10:1) defines that the first data set has 10 information carrying units for every 1 unit in the second or compressed data set. Thus, the corresponding redundancy of 0.9 means 90 percent of the data in the first data set is redundant with respect to the second one.

(a) **Coding Redundancy:** It consists in using variable-length codewords selected as to match the statistics of the original source, in this case, the image itself or a processed version of its pixel values. This type of coding is always reversible and usually implemented using Look-Up Tables (LUTs). Examples of image coding schemes that explore coding redundancy are the Huff-man codes and the arithmetic coding technique.

If the gray levels of an image are coded in a way that uses more code symbols than absolutely necessary to represent each gray level, the resulting image is said to contain coding redundancy. It is almost always present when an image's gray levels are represented with a straight or natural binary code. Let us assume that a random variable r_K lying in the interval [0, 1] represents the gray levels of an image and that each r_K occurs with probability $P_r(r_K)$.

$$P_r(r_K) = N_{k/n}$$

where $k = 0, 1, 2....L-1$

L = No. of gray levels.

N_k =No. of times that gray appears in that image

N = Total no. of pixels in the image

If no. of bits used to represent each value of r_K is l (r_K), the average no. of bits required to represent each pixel is

$$L_{avg} = l(r_K)\, P_r(r_K)$$

That is average length of code words assigned to the various gray levels is found by summing the product of the no. of bits used to represent each gray level and the probability that the gray level occurs. Thus, the total no of bits required to code an M×N image is $M \times N \times L_{avg}$.

(b) **Interpixel Redundancy:** This type of redundancy sometimes called spatial redundancy, interframe redundancy, or geometric redundancy exploits the fact that an image very often contains strongly correlated pixels, in other words, large regions whose pixel values are the same or almost the same. This redundancy can be explored in several ways, one of which is by predicting a pixel value based on the values of its neighboring pixels. In order to do so, the original 2-D array of pixels is usually mapped into a different format, e.g., an array of differences between adjacent pixels. If the original image pixels can be reconstructed from the transformed data set the mapping is said to be reversible. Examples of compression techniques that explore the interpixel redundancy include: Constant Area Coding (CAC), (1-D or 2-D) Run-Length Encoding (RLE) techniques, and many predictive coding algorithms such as Differential Pulse Code Modulation (DPCM).

(c) **Psychovisual Redundancy:** Many experiments on the psychophysical aspects of human vision have proven that the human eye does not respond with equal sensitivity to all incoming visual information; some pieces of information are more important than others. The knowledge of which particular types of information are more or less relevant to the final human user have led to image and video compression techniques that aim at eliminating or reducing any amount of data that is psychovisually redundant. The end result of applying these techniques is a compressed image file, whose size and quality are smaller than the original information, but whose resulting quality is still acceptable for the application at hand. The loss of quality that ensues as a byproduct of such techniques is frequently called quantization, as to indicate that a wider range of

input values is normally mapped into a narrower range of output values thorough an irreversible process. In order to establish the nature and extent of information loss, different fidelity criteria (some objective such as Root Mean Square (RMS) error, some subjective, such as pairwise comparison of two images encoded with different quality settings) can be used. Most of the image coding algorithms in use today exploit this type of redundancy, such as the Discrete Cosine Transform (DCT) based algorithm at the heart of the JPEG encoding standard.

4.2.2 Data Compression

Data compression is defined as the process of encoding data using a representation that reduces the overall size of data. This reduction is possible when the original data set contains some type of redundancy. Digital image compression is a field that studies methods for reducing the total number of bits required to represent an image. This can be achieved by eliminating various types of redundancy that exist in the pixel values.

During the past two decades, various compression methods have been developed to address major challenges faced by digital imaging. These compression methods can be classified broadly into lossy or lossless compression. Lossy compression can achieve a high compression ratio, 50:1 or higher, since it allows some acceptable degradation. Yet it cannot completely recover the original data. On the other hand, lossless compression can completely recover the original data but this reduces the compression ratio to around 2:1.

In medical applications, lossless compression has been a requirement because it facilitates accurate diagnosis due to no degradation on the original image. Furthermore, there exist several legal and regulatory issues that favor lossless compression in medical applications.

4.3 Image Compression and Reconstruction

The principle of image compression algorithm is

- Reducing the redundancy in the image data and
- Producing a reconstructed image from the original image with the introduction of error that is insignificant to the intended applications

The aim here is to obtain an acceptable representation of digital image while preserving the essential information contained in that particular data set.

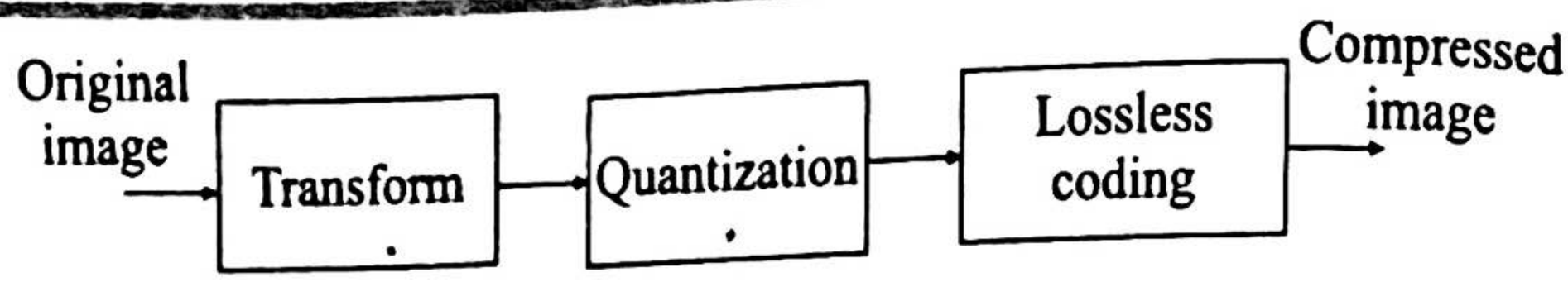


Figure 4.1 Image Compression System

The problem faced by image compression is very easy to define, as in above figure 4.1.

First ; the original digital image is usually transformed into another domain, where it is highly de-correlated by using some transform. This de-correlation concentrates the important image information into a more compact form. The compressor then removes the redundancy in the transformed image and stores it into compressed file or data stream.

In second stage, the quantization block reduces the accuracy of transformed output in accordance with some pre-established fidelity criterion. Also this stage reduces the psycho-visual redundancy of input image. Quantization operation is a reversible process and thus may be omitted when there is a need of error free or lossless compression. In the final stage of the data compression model the symbol coder creates a fixed or variable-length code to represent the quantizer output and maps the output in accordance with the code. Generally, a variable-length code is used to represent the mapped and quantized data set. If assigns the shortest code words to the most occurring output values and thus reduces coding redundancy. The operation in fact is a reversible one.

The decompression reverses the compression process to produce the recovered image as shown in figure 4.2. The recovered image may have lost some information due to the compression, may have error or distortion compared to the original image.

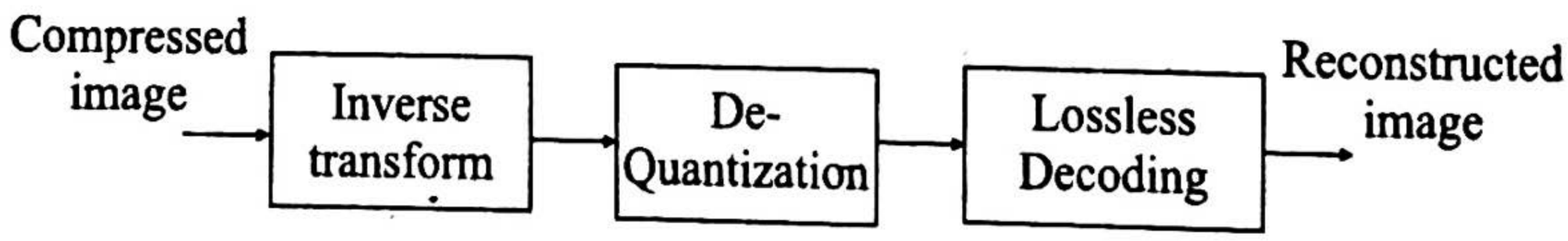


Figure 4.2 Image Decompression System

4.3.1 Image Compression Techniques

There are basically two methods of Image Compression:

(a) **Lossless Coding Techniques:** In Lossless Compression schemes, the reconstructed image, after compression, is numerically identical

to the original image. However, Lossless Compression can achieve a modest amount of Compression.

Lossless coding guaranties that the decompressed image is absolutely identical to the image before compression. This is an important requirement for some application domains, e.g. Medical Imaging, where not only high quality is in the demand, but unaltered archiving is a legal requirement. Lossless techniques can also be used for the compression of other data types where loss of information is not acceptable, e.g. text documents and program executables. Lossless compression algorithms can be used to squeeze down images and then restore them again for viewing completely unchanged.

Lossless compressor as shown in figure 4.3 are usually two-step algorithms. The first step transforms the original image to some other format in which the inter-pixel redundancy is reduced. The second step uses an entropy encoder to remove the coding redundancy. The lossless decompressor is a perfect inverse process of the lossless compressor.

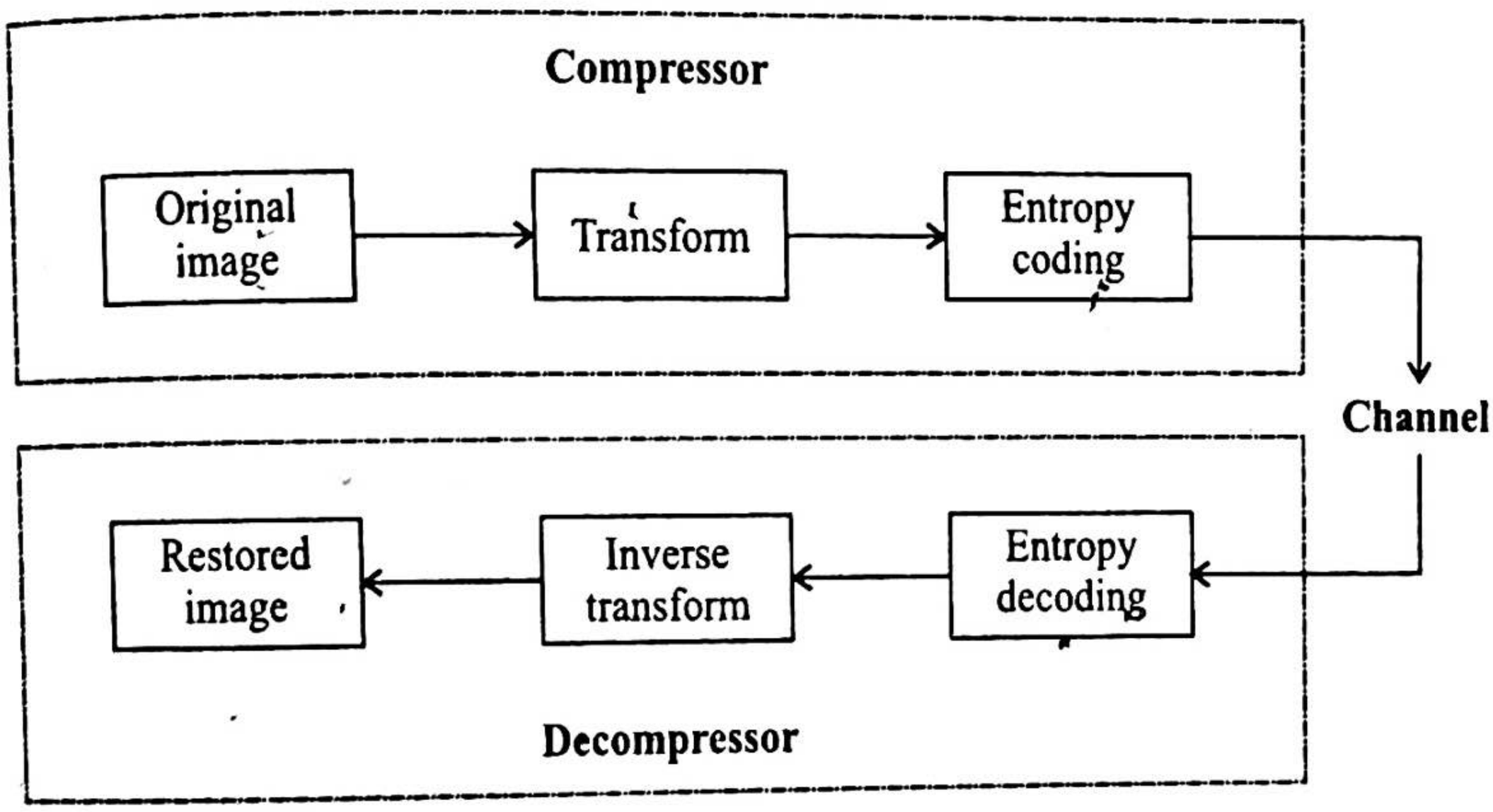


Figure 4.3 Lossless image compression

Typically, medical images can be compressed losslessly to about 50% of their original size. Boncelet etc. all investigated the use of three entropy coding methods for lossless compression with an application to digitized radiographs and found that a bit rate of about 4 to 5 bpp was best. Tavakoli applied various lossless coding techniques to MR images and reported a compression down to about 5 to 6 bpp, with LZ (Lempel-Ziv) coding achieving the best results.

Lossless compression works best with decorrelated data. Roose et. al. investigated prediction, linear transformation, and multiresolution methods for decorrelating medical image data before coding them. The compression result was 3:1 and less than 2:1 for angiograms and MRI respectively. Kuduvalli and Rangayyan studied similar techniques and found linear prediction and interpolation techniques gave the best results with similar compression ratios.

Here, we summarize the lossless compression methods into four categories.

(i) **Run length coding:** Run length coding replaces data by a (length, value) pair, where "value" is the repeated value and "length" is the number of repetitions. This technique is especially successful in compressing bi-level images since the occurrence of a long run of a value is rare in ordinary gray-scale images. A solution to this is to decompose the gray-scale image into bit planes and compress every bit-plane separately. Efficient run-length coding method is one of the variations of run length coding.

(ii) **Lossless predictive coding:** Lossless predictive coding predicts the value of each pixel by using the values of its neighboring pixels. Therefore, every pixel is encoded with a prediction error rather than its original value. Typically, the errors are much smaller compared with the original value so that fewer bits are required to store them.

DPCM (Differential Pulse Code Modulation) is a predictive coding based lossless image compression method. It is also the base for lossless JPEG compression. A variation of the lossless predictive coding is the adaptive prediction that splits the image into blocks and computes the prediction coefficients independently for each block to achieve high prediction performance. It can also be combined with other methods to get a hybrid coding algorithm with higher performance.

(iii) **Entropy coding:** An entropy encoding is a coding scheme that assigns codes to symbols so as to match code lengths with the probabilities of the symbols. Typically, entropy encoders are used to compress data by replacing symbols represented by equal-length codes with symbols represented by codes proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes.

According to Shannon's source coding theorem, the optimal code length for a symbol is

$\log_b P$,

Where b is the number of symbols used to make output codes and P is the probability of the input symbol. $\log_2 P$

Three of the most common entropy encoding techniques are Huffman coding, range encoding, and arithmetic coding. If the approximate entropy characteristics of a data stream are known in advance (especially for signal compression), a simpler static code such as unary coding, Elias gamma coding, Fibonacci coding, Golomb coding, or Rice coding may be useful.

There are three main techniques for achieving entropy coding:

- **Huffman coding:** One of the simplest variable length coding schemes.
- **Run-length Coding (RLC):** Very useful for binary data containing long runs of ones of zeros.
- **Arithmetic coding:**A relatively new variable length coding scheme that can combine the best features of Huffman and run-length coding, and also adapt to data with non-stationary statistics.

Entropy represents the minimum size of dataset necessary to convey a particular amount of information. Huffman coding, LZ (Lempel-Ziv) coding and arithmetic coding are the commonly used entropy coding schemes.

Huffman coding utilizes a variable length code in which short code words are assigned to more common values or symbols in the data, and longer code words are assigned to less frequently occurring values. Modified Huffman coding and dynamic Huffman coding are two examples among many variations of Huffman's technique.

LZ coding replaces repeated substrings in the input data with references to earlier instances of the strings. It often refers to two different approaches to dictionary-based compression: the LZ77 and the LZ78. LZ77 utilizes a sliding window to search for the substrings encountered before and then substitutes them by the (position, length) pair to point back to the existing substring. LZ78 dynamically constructs a dictionary from the input file and then

replaces the substrings by the index in the dictionary. Several compression methods, among which LZW (Lempel-Ziv-Welch) is one of the most well known methods, have been developed based on these ideas. Variations of LZ coding are used in the Unix utilities Compress and Gzip.

Arithmetic coding represents a message as some finite intervals between 0 and 1 on the real number line. Basically, it divides the intervals between 0 and 1 into a number of smaller intervals corresponding to the probabilities of the message's symbols. Then the first input symbol selects an interval, which is further divided into smaller intervals. The next input symbol selects one of these intervals, and the procedure is repeated. As a result, the selected interval narrows with every symbol, and in the end, any number inside the final interval can be used to represent the message. That is to say, each bit in the output code refines the precision of the value of the input code in the interval. A variation of arithmetic coding is the Q-coder, developed by IBM in the late 1980's. Two references are provided for the latest Q-coder variation.

(iv) **Multiresolution coding:** HINT (Hierarchical Interpolation) is a multiresolution coding scheme based on sub-samplings. It starts with a low-resolution version of the original image, and interpolates the pixel values to successively generate higher resolutions. The errors between the interpolation values and the real values are stored, along with the initial low-resolution image. Compression is achieved since both the low-resolution image and the error values can be stored with fewer bits than the original image.

Laplacian Pyramid is another multiresolution image compression method developed by Burt and Adelson. It successively constructs lower resolution versions of the original image by down sampling so that the number of pixels decreases by a factor of two at each scale. The differences between successive resolution versions together with the lowest resolution image are stored and utilized to perfectly reconstruct the original image. But, it cannot achieve a high compression ratio because the number of data values is increased by 4/3 of the original image size.

In general, the image is reversibly transformed into a group of different resolution sub-images in multi-resolution coding. Usually, it reduces the entropy of the image.

Some kinds of tree representation could be used to get more compression by exploiting the tree structure of the multi-resolution methods.

(b) **Lossy Coding Techniques:** Lossy techniques cause image quality degradation in each Compression /De-compression step. Careful consideration of the Human Visual perception ensures that the degradation is often unrecognizable, though this depends on the selected compression ratio. An image reconstructed following Lossy compression contains degradation relative to the original. Often this is because the compression schemes are capable of achieving much higher compression. Under normal viewing conditions, no visible loss is perceived (visually Lossless).

Generally, most lossy compressors figure 4.4 are three-step algorithms, each of which is in accordance with three kinds of redundancy mentioned below.

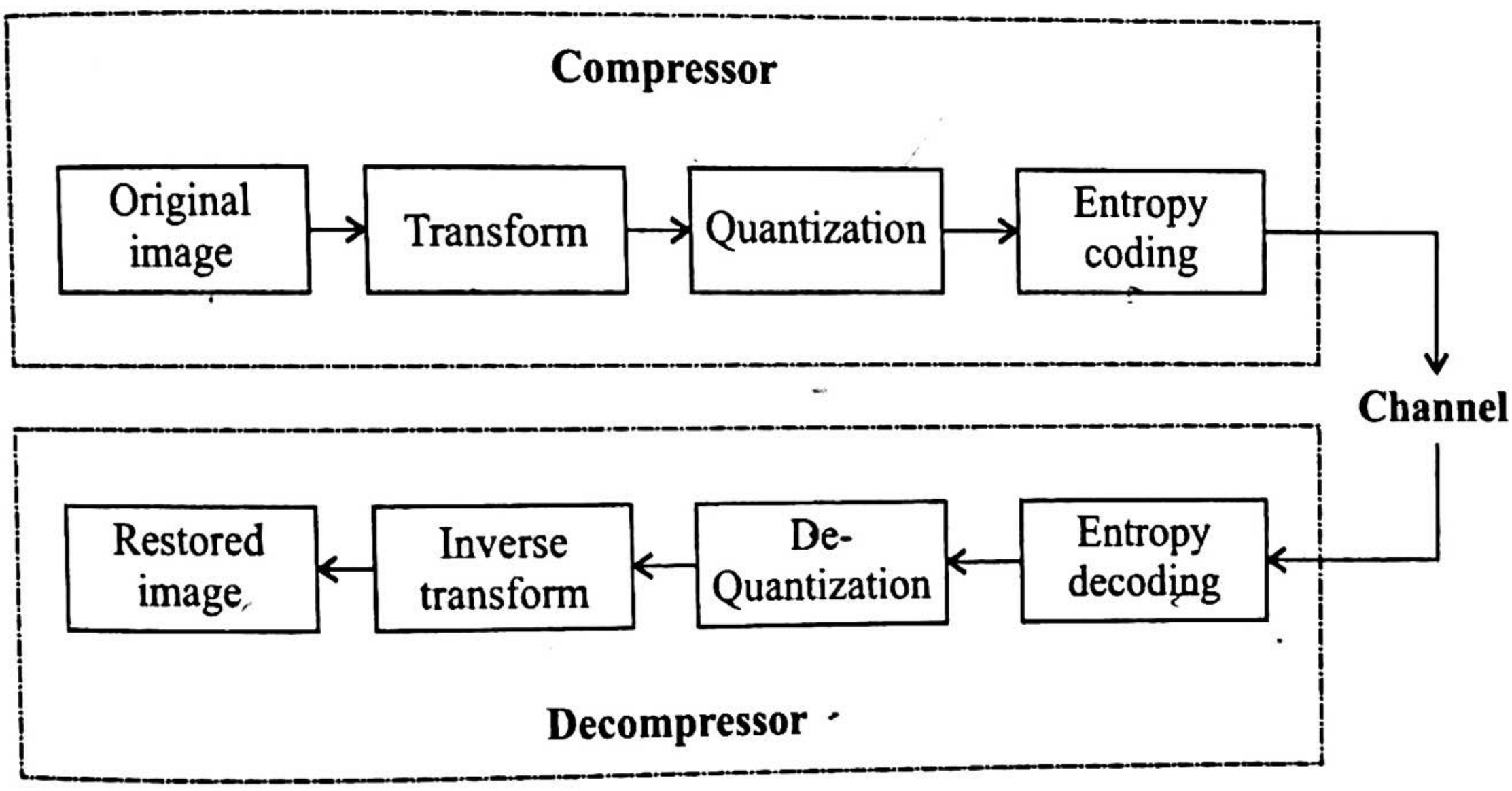


Figure 4.4 Lossy compressors

Lossy image compression

The first stage is a transform to eliminate the inter-pixel redundancy to pack information efficiently. Then a quantizer is applied to remove psycho-visual redundancy to represent the packed information with as few bits as possible. The quantized bits are then efficiently encoded to get more compression from the coding redundancy.

(i) **Transform coding:** Transform coding is a general scheme for lossy image compression. It uses a reversible and linear transform to decorrelate the original image into a set of coefficients in transform domain. The coefficients are then quantized and coded sequentially in transform domain.

Numerous transforms are used in a variety of applications. The discrete KLT (Karhunen-Loeve Transform), which is based on the Hotelling transform, is optimal with its information packing properties, but usually not practical since it is difficult to compute. The DFT (Discrete Fourier Transform) and DCT (discrete cosine transform) approximate the energy-packing efficiency of the KLT, and have more efficient implementation. In practice, DCT is used by most practical transform systems, since the DFT coefficients require twice the storage space of the DCT coefficients.

(ii) **Block transform coding:** To simplify the computations, block transform coding exploits correlation of the pixels within a number of small blocks that divide the original image. As a result, each block is transformed, quantized and coded separately. This technique, using square 8*8 pixel blocks and he DCT followed by Huffman or arithmetic coding, is utilized in the ISO JPEG (Joint Photographic Expert Group) draft international standard for image compression. The disadvantage of this scheme is the blocking (or tiling) artifacts appear at high compression ratios.

Since, the adoption of the JPEG standard, the algorithm has been the subject of considerable research. Collins et. al. studied the effects of a 10:1 lossy image compression scheme based on JPEG, with modifications to reduce the blocking artifacts. Baskurt et. al used an algorithm similar to JPEG to compress mammograms with a bit rate as low as 0.27 bpp (bits per pixel) while retaining detection ability of pathologies by radiologists. Kostas et. al. used JPEG modified for use with 12-bit images and custom quantization tables to compress mammograms and chest radiographs.

Moreover, the ISO JPEG committee is currently developing a new still-image compression standard called JPEG-2000 for delivery to the marketplace by the end of the year 2000. The new JPEG-2000 standard is based upon wavelet decompositions combined with more powerful quantization and encoding strategies such as embedded quantization and context-based arithmetic. It provides the

potential for numerous advantages over the existing JPEG standard. Performance gains include improved compression efficiency at low bit rates for large images, while new functionalities include multi-resolution representation, scalability and embedded bit stream architecture, lossy to lossless progression, ROI (Region of Interest) coding, and a rich file format.

(iii) **Full-frame transform coding:** To avoid the artifacts generated by block transforms, full-frame methods, in which the transform is applied to the whole image as a single block, have been investigated in medical imaging research. The tradeoff is the increased computational requirements and the appearance of ringing artifacts (a periodic pattern due to the quantization of high frequencies).

Subband coding is one example among full-frame methods. It will produce a number of sub-images with specific properties such as a smoothed version of the original plus a set of images with the horizontal, vertical, and diagonal edges that are missing from the smoothed version according to different frequencies. Rompelman applied subband coding to compress 12-bit CT images at rates of 0.75 bpp and 0.625 bpp without significantly affecting diagnostic quality.

Recently, much research has been devoted to the DWT discrete Wavelet Transform for subband coding of images. DWT is a hierarchical subband decomposition particularly suited to image compression. Many different wavelet functions can be applied to different applications. In general, more complicated wavelet functions provide better performance. The wavelet transform can avoid the blocking artifacts presented in block transform methods and allow easy progressive coding due to its multiresolution nature

Bramble et. al. used full-frame Fourier transform compression on 12 bpp digitized hand radiographs at average rates from about 0.75 bpp to 0.1 bpp with no significant degradation in diagnostic quality involving the detection of pathology characterized by a lack of sharpness in a bone edge. However, Cook et. al. investigated the effects of full-frame DCT compression on low-contrast detection of chest lesions and found significant degradation at rates of about 0.75 bpp. These results illustrate that both imaging modality and the task play an important role in determining achievable compression.

4.4 Measurements for Compression Methods

4.4.1 Measurements for Lossy Compression Methods

Lossy compression methods result in some loss of quality in the compressed images. It is a trade off between image distortion and the compression ratio. Some distortion measurements are often used to quantify the quality of the reconstructed image as well as the compression ratio (the ratio of the size of the original image to the size of the compressed image). The commonly used objective distortion measurements, which are derived from statistical terms, are the RMSE (Root Mean Square Error), the NMSE (Normalized Mean Square Error) and the PSNR (Peak Signal-to-Noise Ratio).

These measurements are defined as follows:

$$3\text{RMSE} = \sqrt{\frac{1}{N*M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[f(i,j)-f'(i,j)\right]^2}$$

$$\text{NMSE} = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[f(i,j)-f'(i,j)\right]^2 \Bigg/ \left[\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} f(i,j)\right]^2$$

$$\text{PSNR} = 20*\log_{10}\left(\frac{255}{RMSE}\right)$$

where the images have $N*M$ pixels (8 bits per pixel), $f(i, j)$ represents the original image, and $f(i, j)$ represents the reconstructed image after compression-decompression.

Since, the images are for human viewing, it leads to subjective measurements based on subjective comparisons to tell how "good" the decoded image looks to a human viewer. Sometimes, application quality can be used as a measure to classify the usefulness of the decoded image for a particular task such as clinical diagnosis in medical images and meteorological prediction in satellite images and so on.

When comparing two lossy coding methods, we may either compare the qualities of images reconstructed at a constant bit rate, or, equivalently, we may compare the bit rates used in two constructions with the same quality, if it is accomplishable.

4.4.2 Measurements for Lossless Compression Methods

Lossless compression methods result in no loss in the compressed images so that it can perfectly restore the original images when

applying a reversible process. The frequently used measurement in lossless compression is the compression ratio. This measurement can be misleading, since it depends on the data storage format and sampling density. For instance, medical images containing 12 bits of useful information per pixel are often stored using 16 bpp.

A better measurement of compression is the bit rate due to its independence of the data storage format. A bit rate measures the average number of bits used to represent each pixel of the image in a compressed form. Bit rates are measured in bpp, where a lower bit rate corresponds to a greater amount of compression.

4.5 Quantization

Quantization involved in image processing. Quantization techniques generally compress by compressing a range of values to a single quantum value. By reducing the number of discrete symbols in a given stream, the stream becomes more compressible. For example seeking to reduce the number of colors required to represent an image. Another widely used example are DCT data quantization in JPEG and DWT data quantization in JPEG 2000.

4.5.1 Quantization in Image Compression

The human eye is fairly good at seeing small differences in brightness over a relatively large area, but not so good at distinguishing the exact strength of a high frequency brightness variation. This fact allows one to get away with greatly reducing the amount of information in the high frequency components. This is done by simply dividing each component in the frequency domain by a constant for that component, and then rounding to the nearest integer. This is the main lossy operation in the whole process. As a result of this, it is typically the case that many of the higher frequency components are rounded to zero, and many of the rest become small positive or negative numbers.

4.5.2 Measurement of Image Quality

The design of an imaging system should begin with an analysis of the physical characteristics of the originals and the means through which the images may be generated. For example, one might examine a representative sample of the originals and determine the level of detail that must be preserved, the depth of field that must be captured, whether they can be placed on a glass platen or require a custom book-edge

scanner, whether they can tolerate exposure to high light intensity, and whether specular reflections must be captured or minimized. A detailed examination of some of the originals, perhaps with a magnifier or microscope, may be necessary to determine the level of detail within the original that might be meaningful for a researcher or scholar. For example, in drawings or paintings it may be important to preserve stippling or other techniques characteristic.

4.6 JPEG Compression

The Joint Photographic Expert Group (JPEG), under both the International Standards Organization (ISO) and the International Telecommunications Union-Telecommunication Sector (ITU-T) has published several standards:

- **JPEG:** lossycoding of continuous tone still images Based on DCT
- **JPEG - LS:** lossless and near lossless coding of continuous tone still images based on predictive coding and entropy coding
- **JPEG 2000:** scalable coding of continuous tone still images (from lossyto lossless) based on wavelet transform.

1. **Basic advantages of JPEG 2000 are**
   (a) Improved coding efficiency
   (b) Full quality scalability
   (c) From lossless to lossyat different bit rate
   (d) Spatial scalability
   (e) Improved error resilience
   (f) Timing
   (g) Region of interests
   (h) More demanding in memory and computation time

65536 Bytes
8 bpp

4839 Bytes
0.59 bpp
CR = 13.6

Figure 4.5 Performance for B/W images

2. **JPEG for color images are**

   (a) Color images are typically stored in (R,G,B) format

   (b) JPEG standard can be applied to each component separately

   (c) Does not make use of the correlation between color components

   (d) Does not make use of the lower sensitivity of the human eye to chrominance samples

3. **Alternate Approach**

Convent (R,G,B) representation to a YCbCr representation.

(Y: luminance, Cb, Cr: chrominance)

Down-sample the two chrominance components.

Because the peak response of the eye to the luminance component occurs at a higher frequency.

JPEG standard is designed to handle an image consists of many (up to 100) components.

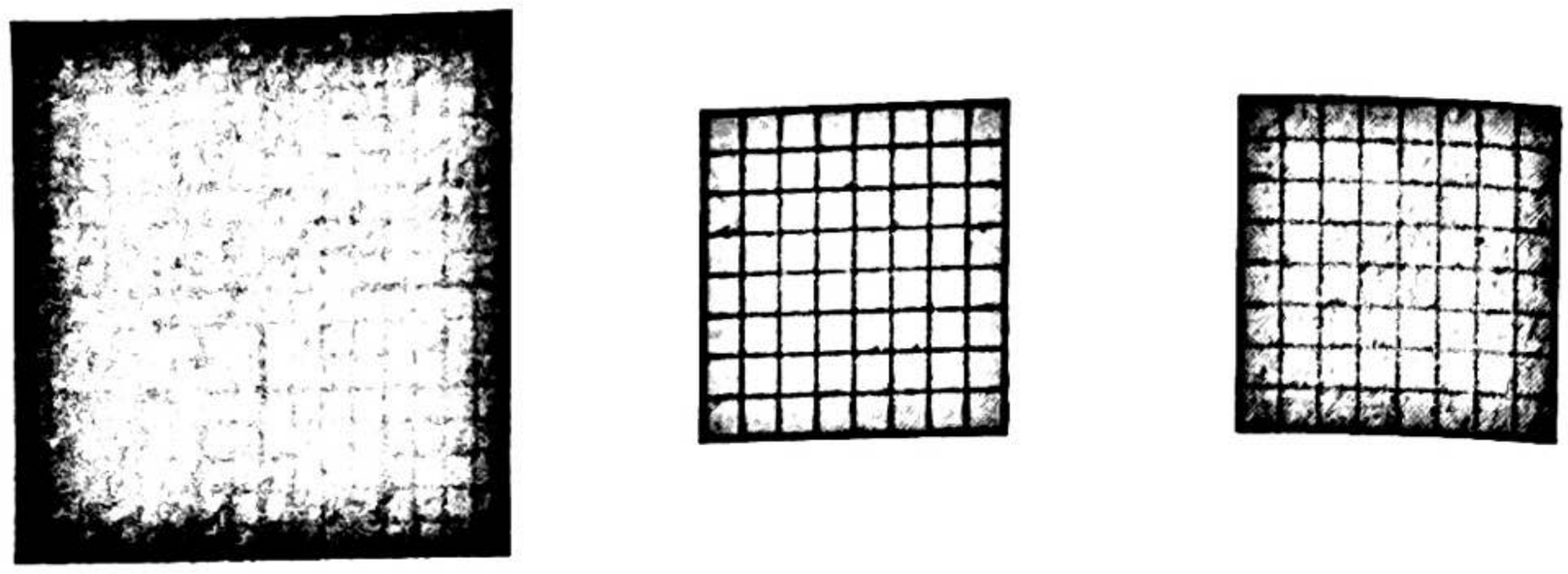
(a) RGB to YCbCr Conversion

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & -0.001 & 1.402 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.722 & 0.001 \end{bmatrix} \begin{bmatrix} Y \\ C_b - 128 \\ C_r - 128 \end{bmatrix}$$

Note : Cb~Y-B, Cr~Y-R, are known as color difference signal.

(b) Coding Unit in JPEG



4 8×8 Y blocks 1 8×8 Cb blocks 1 8×8 Cr blocks

Each basic coding unit (called a data unit) is a 8x8 block in any color component. In the interleaved mode, 4 Y blocks and 1 Cband 1 Cr blocks are processed as a group (called a Minimum Coding Unit or MCU) for a 4:2:0 image.

(c) 1992 JPEG Standard Contains several modes

- The Baseline system (what is commonly known as JPEG) : lossy
- It Can handle gray scale or color images (8bit)
- Extended system: lossy
- It Can handle higher precision (12 bit) images, providing progressive streams, etc.

4. Performance of JPEG

(a) For color images at 24 bits/pixel (bpp)

- 0.25-0.5 bpp: moderate to good
- 0.5-0.75 bpp: good to very good
- 0.75-1.5 bpp: excellent, sufficient for most applications
- 1.5-2 bpp: indistinguishable from original

(b) For grayscale images at 8 bpp 0.5 bpp: excellent quality

REVIEW QUESTIONS

Q.1 What is data compression ?

Q.2 What is data redundancy ? Also write its take.

Q.3 Write about Lossy compassion Methods.

Q.4 Explain Image Compression & Reconstruction Technique.

Q.5 Explain JPEG compression ?

Q.6 What do you mean by Quantization ? Explain it.

○○○

CHAPTER 5

Image Segmentation, Representation and Description

OBJECTIVES

- Introduction
- Edge-Based Segmentation
- Gradient Operator
- Edge Linking and Boundary Detection
- Thresholding
- Region-Based Segmentation
- Use Morphological to Improve Fast Scanning Algorithm
- Use Geometrical Mathematic to Improve Fast Scanning
- Segmentation by Morphological Watersheds
- Representation
- Signature
- Boundary Segments
- Skeletons

5.1 Introduction

Segmentation is to subdivide an image into constituent regions or objects. Image segmentation algorithms are generally based on one of the two basic properties of intensity value: discontinuity and similarity. In this section, we discuss many different approaches to detect the boundary choose of threshold, and so on. After an image is segmented into regions, each region is represented and described in a form suitable for further computer processing. Basically, there are two ways to represent a region involves two choices. The external representation is used when the primary focus is on shape characteristics.

5.2 Edge-Based Segmentation

There are three basic types of gray-level discontinuities in a digital image: points, lines, and edges. The most common way to look for discontinuities is to run a mask through the image. And the response of the mask at any point in the image is given by

$$R = w_1 z_1 + w_2 z_2 + \ldots\ldots + w_9 z_9 = \sum_{i=1}^{9} z_i \qquad \ldots(5.1)$$

1. **Point Detection:** We say that a point has been detected at the location on which the mask is centred if

$$|R| \geq T$$

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 5.1 A point detection mask

2. **Line Detection:** In the line detection, we use the same way of point to detect different angle of line. Figure 5.2 illustrates the line detection mask at horizontal direction.

-1	-1	-1
2	2	2
-1	-1	-1

Figure 5.2 A mask of horizontal direction

3. **Edge Detection:** Although, point and line detection certainly are important in any discussion on segmentation, edge detection is by far the most common approach for detecting meaningful discontinuities in gray level. About the edge detection, we discuss approaches for implementing first- and second-order digital derivatives for the detection if edges in an image.

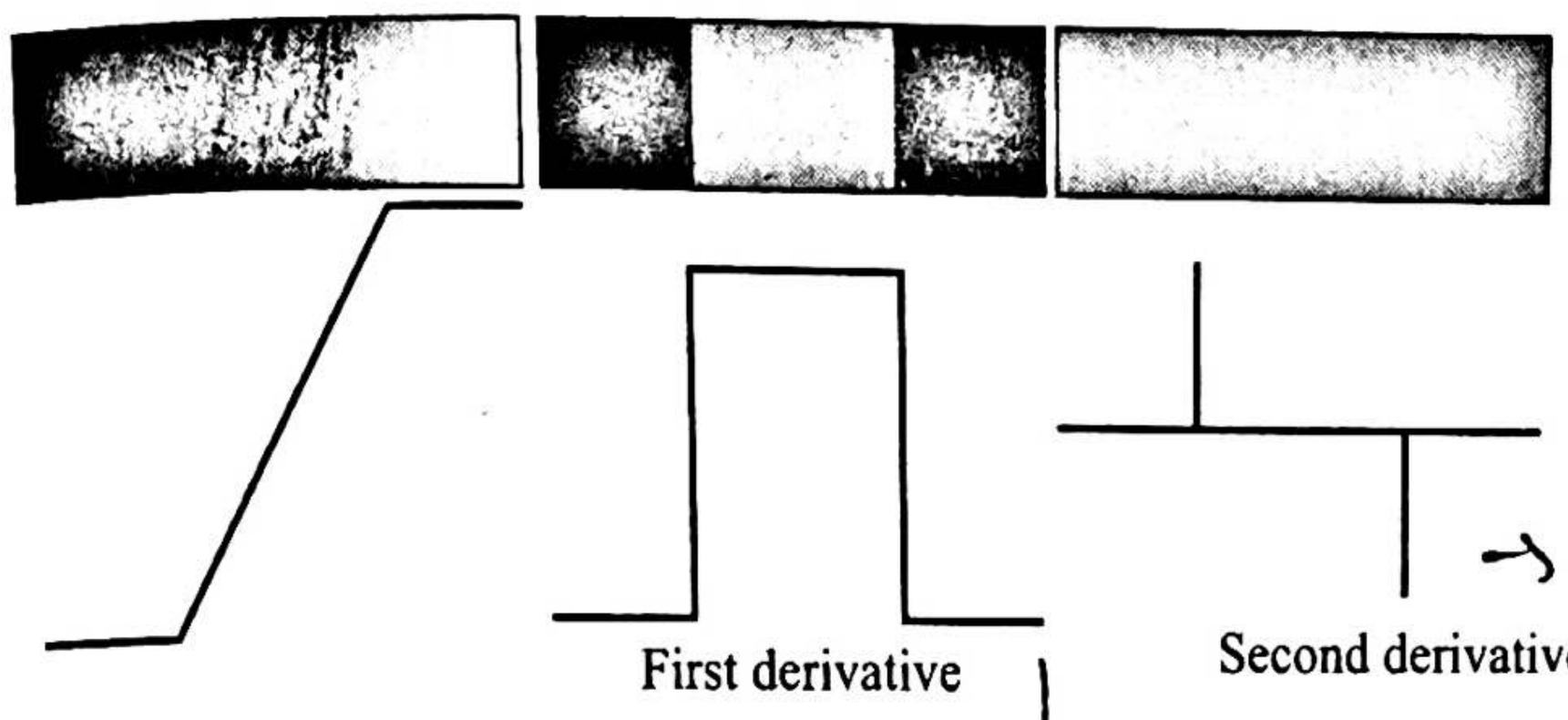


Figure 5.3 Gray-level profile

5.3 Gradient Operator

Filter masks used to compute the derivatives needed for the gradient are often called gradient operators difference operators, edge operators or edge detectors.

Digital approximative of the partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ over a neighbourhood point is required. Since we have

$$g_x = \frac{\partial f(x,y)}{\partial x} = f(x+1, y) - f(x, y) \qquad ...(5.2)$$

and

$$g_y = \frac{\partial f(x,y)}{\partial y} = f(x, y+1) - f(x, y) \qquad ...(5.3)$$

Equation (5.2) and (5.3) can be used to implement for all relative values of x and y by filtering $f(x, y)$ with one dimensional mask as shown in figure 5.4.

−1	
1	

−1	1

Figure 5.4 One dimensional mask for implementation of equation (5.2) and (5.3)

If we want to filter some diagonal edges then 2-D mask is useful as compare to 1-D mask. The Roberts cross-gradient operator is an example of 2-D masks with a diagonal preferences.

Let a 3 × 3 region is used here and various operator (masks) are used to compute the gradient at some pertient point say z_5. Then

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5) \qquad \text{...(5.4)}$$

and

$$g_y = \frac{\partial f}{\partial x} = (z_8 - z_6) \qquad \text{...(5.5)}$$

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Roberts mask

-1	0
0	1

0	-1
1	0

Prewitt mask

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

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

$Z_7 + 2Z_8 + Z_9 - (Z_1 + Z_2 + Z_3)$

Sobel mask

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

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

Figure 5.5 3×3 region of image with various masks to compute the gradient at z_5

All these masks based on the nature of the dates on opposite side of the center point. Due to this they carry more information regarding the direction of an edge.

So, the simplest digital approximation to the partial derivatives is

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \quad ...(5.6)$$

and

$$g_y = \frac{\partial f}{\partial x} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \quad ...(5.7)$$

It shows that the difference between the third and first rows of the 3×3 region approximates the derivatives in the x-direction and the difference between third and first column approximates the derivatives in y-direction.

Equation (5.6) and (5.7) can be implement over an entire image by filtering f with the Prewitt operators. Slightly variation in these two equations can leads to another mask which is known as "Sobel operators". It have just weight of 2 in the center coefficient i.e.,

Sobel

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad ...(5.8)$$

and

$$g_y = \frac{\partial f}{\partial x} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad ...(5.9)$$

The Prewitt masks are simple to implement as compare to Sobel masks. But Sobel masks have better noise suppression (smoothing) characteristics due to which Sobel masks are preferred method for implementations. All these mask operators are used to obtain the gradient components g_x and g_y at every pixel location in an image.

The magnitude of these gradient can be obtained by following equation.

$$M(x, y) = | g_x | + | g_y | \quad ...(5.10)$$

The Prewitt and Sobel masks can also be used for detecting diagonal edges. Figure 5.6 shows additional masks for diagonal edge detections.

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

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

Prewitt masks

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

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

Sobel masks

Figure 5.6 Detection of diagonal edges using Prewitt and Sobal mask operators

Gradient operation

The gradient of an image at location (x, y) is defined as the vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \ldots(5.11)$$

We can use this to find the magnitude direction angle

$$\nabla f = mag(\nabla f) = \left[G_x^{\,2} + G_y^{\,2}\right]^{1/2} \quad \ldots(5.12)$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad \ldots(5.13)$$

5.4 Edge Linking and Boundary Detection

5.4.1 Local Processing

The two principal properties used for establishing similarity of edge pixels in this kind of analysis are : (1) The strength of the response of the gradient operator used to produce the edge pixel; and (2) The direction of the gradient vector. If

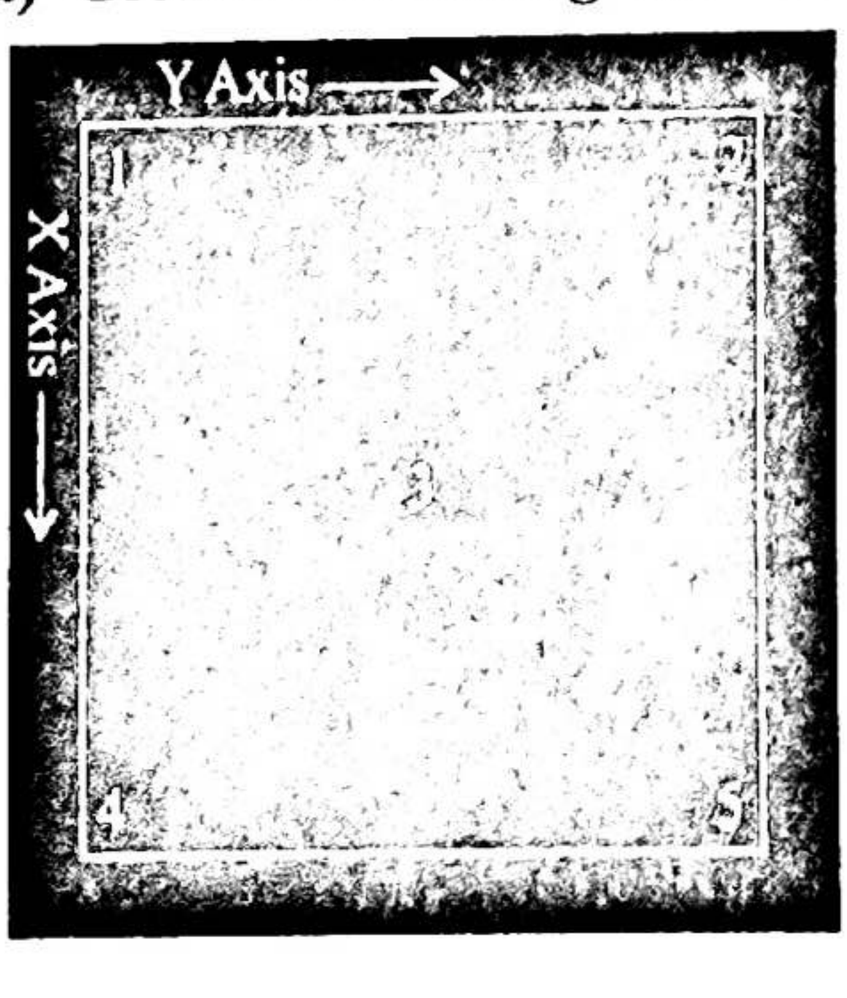
$$\left|\nabla f(x,y) - \nabla f(x_0,y_0)\right| \le E \quad \ldots(5.14)$$

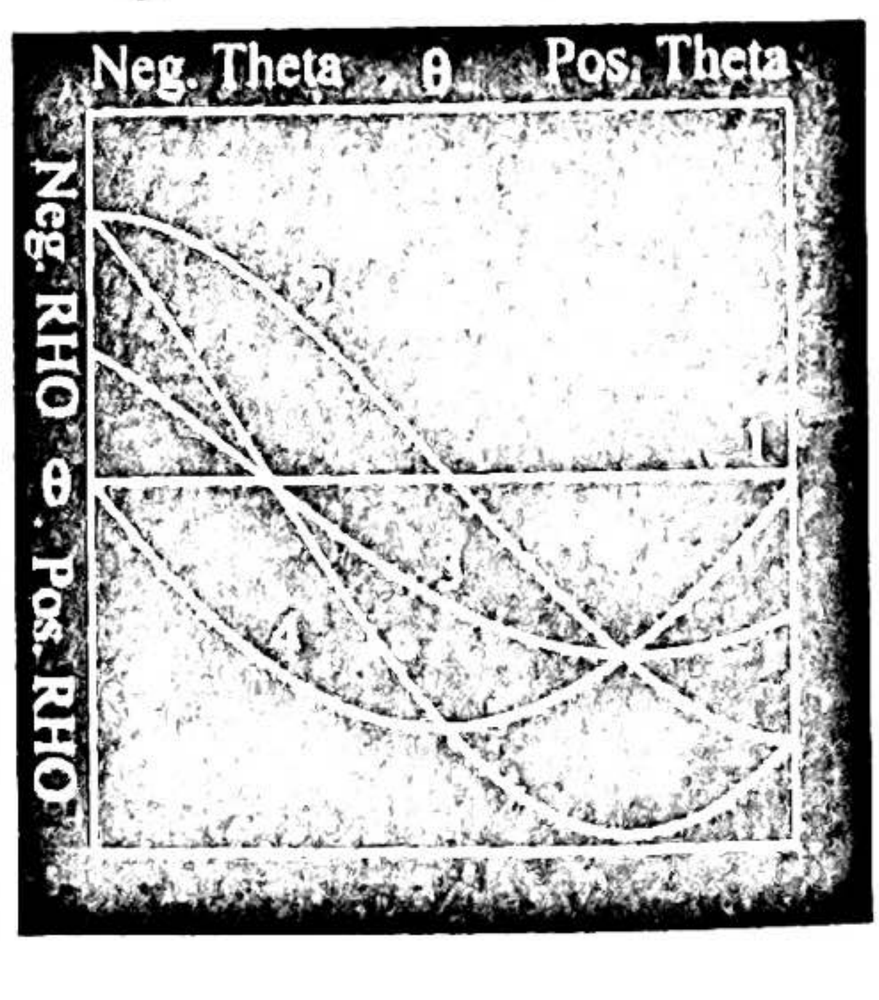
$$\left|\alpha(x,y) - \alpha(x_0,y_0)\right| < A \quad \ldots(5.15)$$

the point is detected.

5.4.2 Global Processing

(a) Global Processing via the Hough Transform





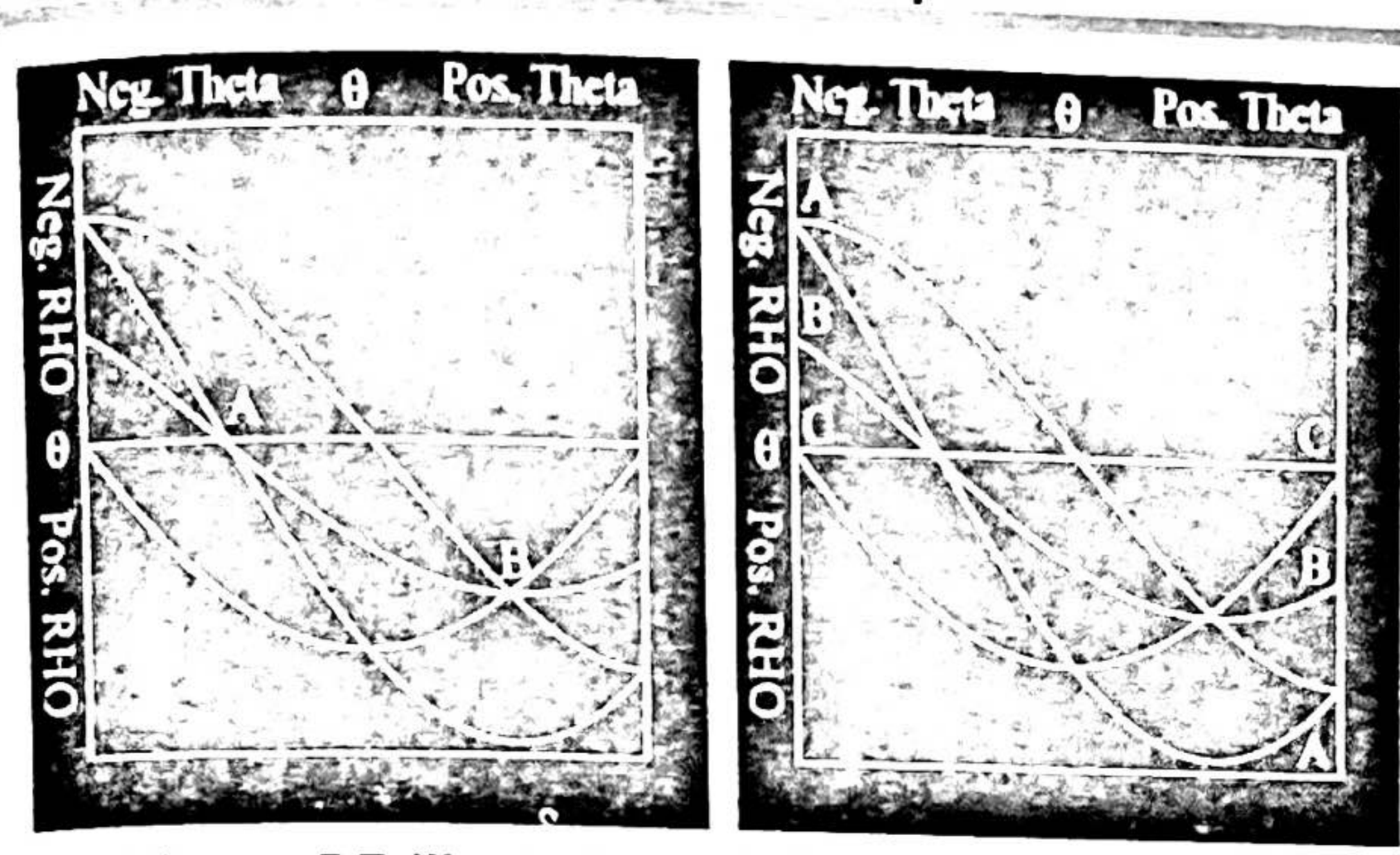


Figure 5.7 Illustration of the Hough transform

(b) Global Processing via Graph-Theoretic Techniques

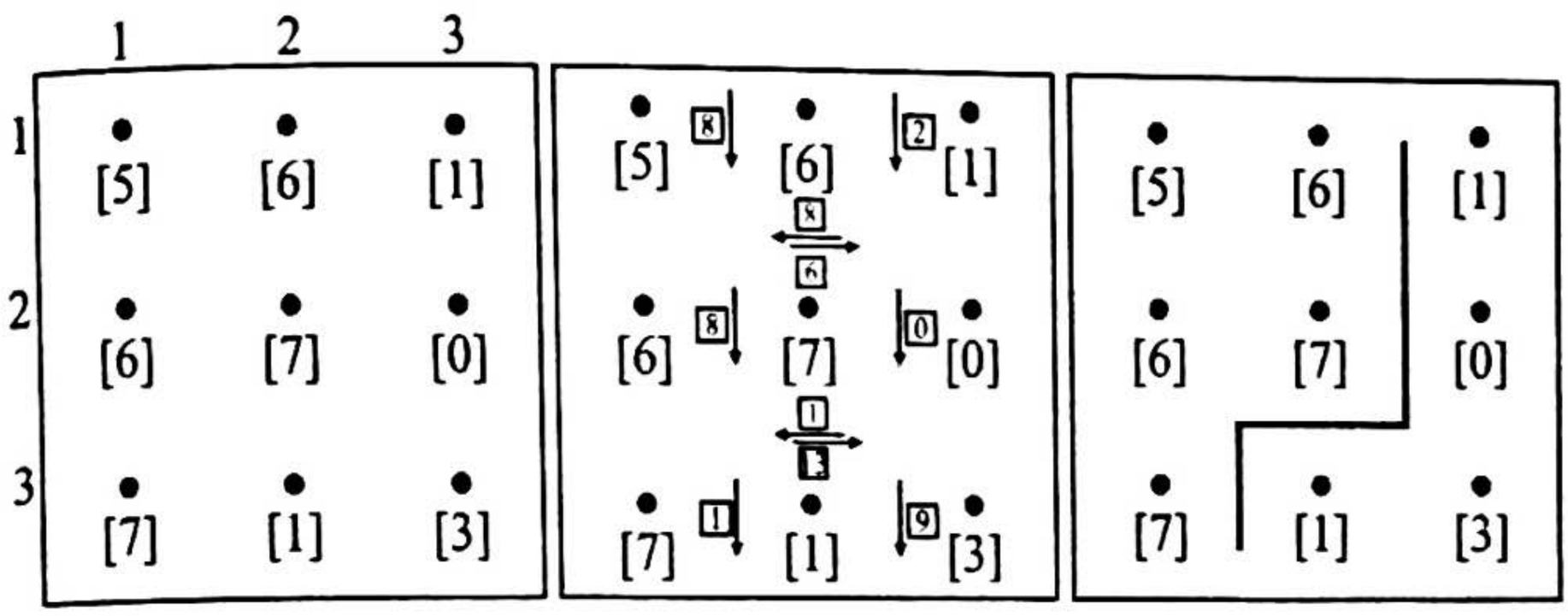


Figure 5.8 (a) A 3 3 image region (b) Edge segments and their costs (c) Edge corresponding to the lowest-cost path graph

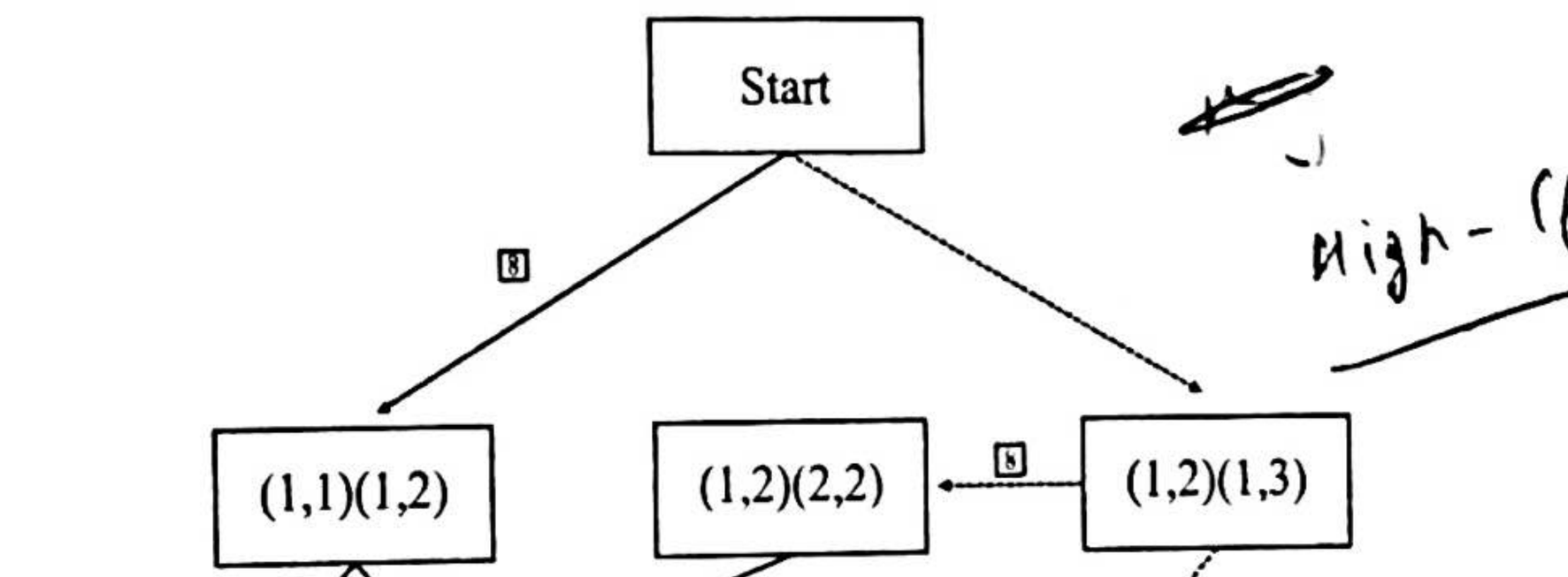


Figure 5.9 Graph for the image. (The lowest-cost path is shown dashed)

5.5 Thresholding

In this section, we introduce an approach to produce the two-level image from the gray level image. After we decide a threshold ψ, we make the gray level value become 1 if the value more than T, and vice versa. The formula is shown as following.

$$g(x,y) = \begin{cases} 1 \text{ if } f(x,y) > T \\ 0 \text{ if } f(x,y) \le T \end{cases} \quad \ldots(5.16)$$

5.5.1 Basic Global Thresholding

The following algorithm can be used to obtain T automatically:

(a) Select an initial estimation for T

(b) Segment the image using T. This will produce two groups of pixels: $G1$ consisting of all pixels with gray level values > T and G2 consisting of pixels with values <=T

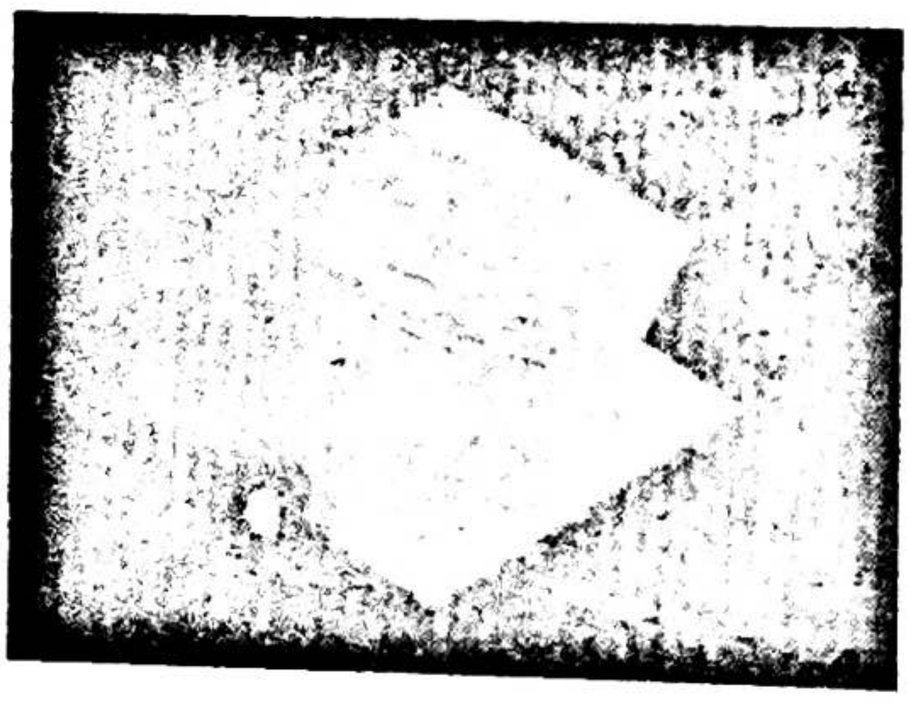
(c) Compute the average gray level values u1 and u2 for the pixels in regions $G1$ and $G2$.

(e) Compute a new threshold value: T=1/2(u1+ u2).

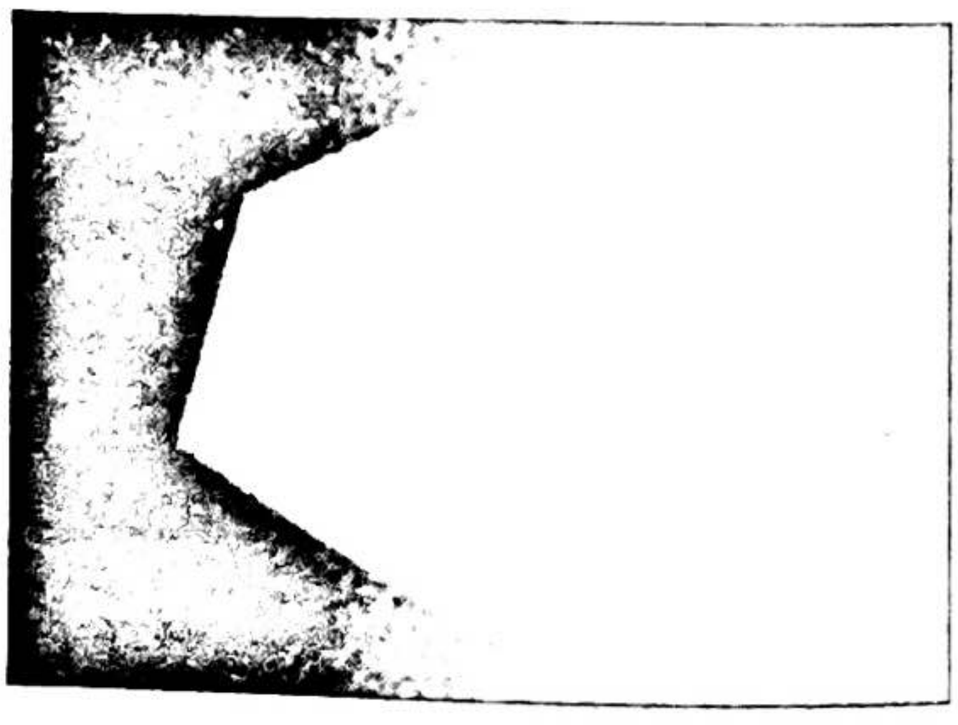
(f) Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T0

5.5.2 Basic Adaptive Thresholding

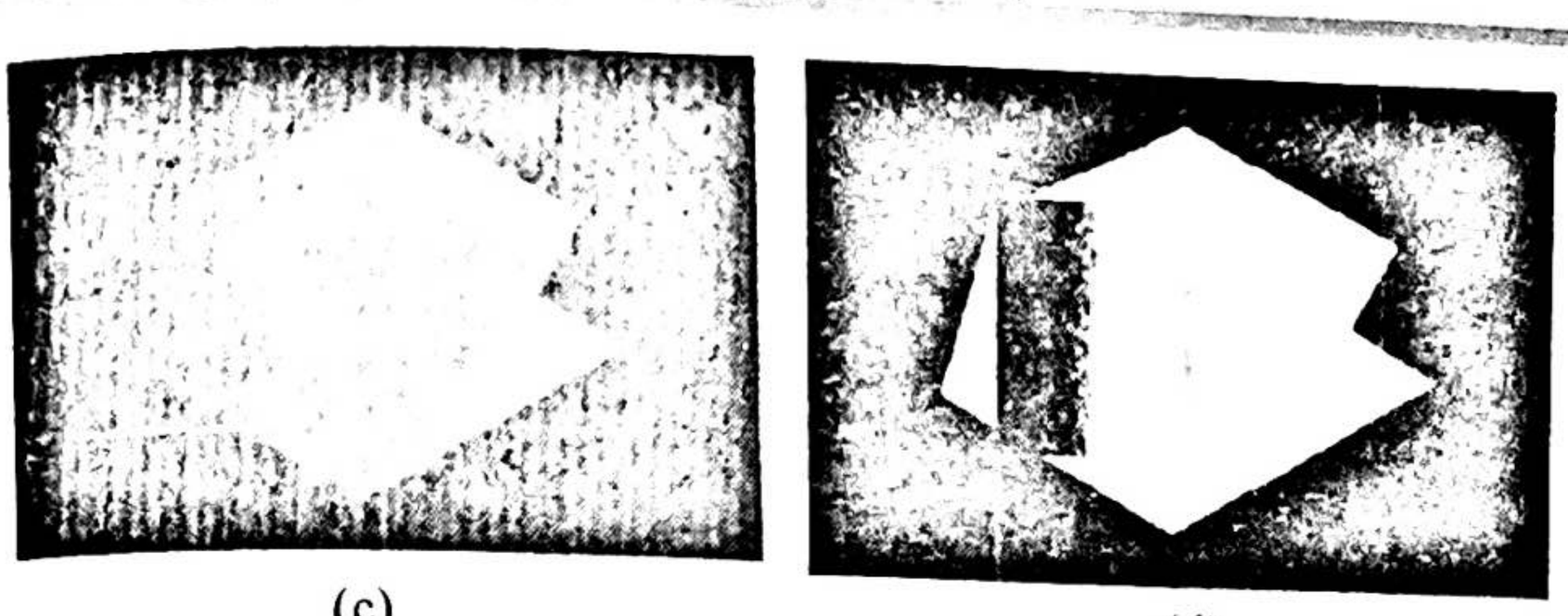
Because of the role of illumination, we find an approach to solve this problem that is subdividing Image into individual subimages. The following figure 5.10 shows the result of the adaptive thresholding.



(a)



(b)



(c) (d)

Figure 5.10 (a) Original image (b) Result of global thresholding (c) An image subdivided into subimages (d) Result of adaptive thresholding

5.5..3 Optimal Global and Adaptive Thresholding

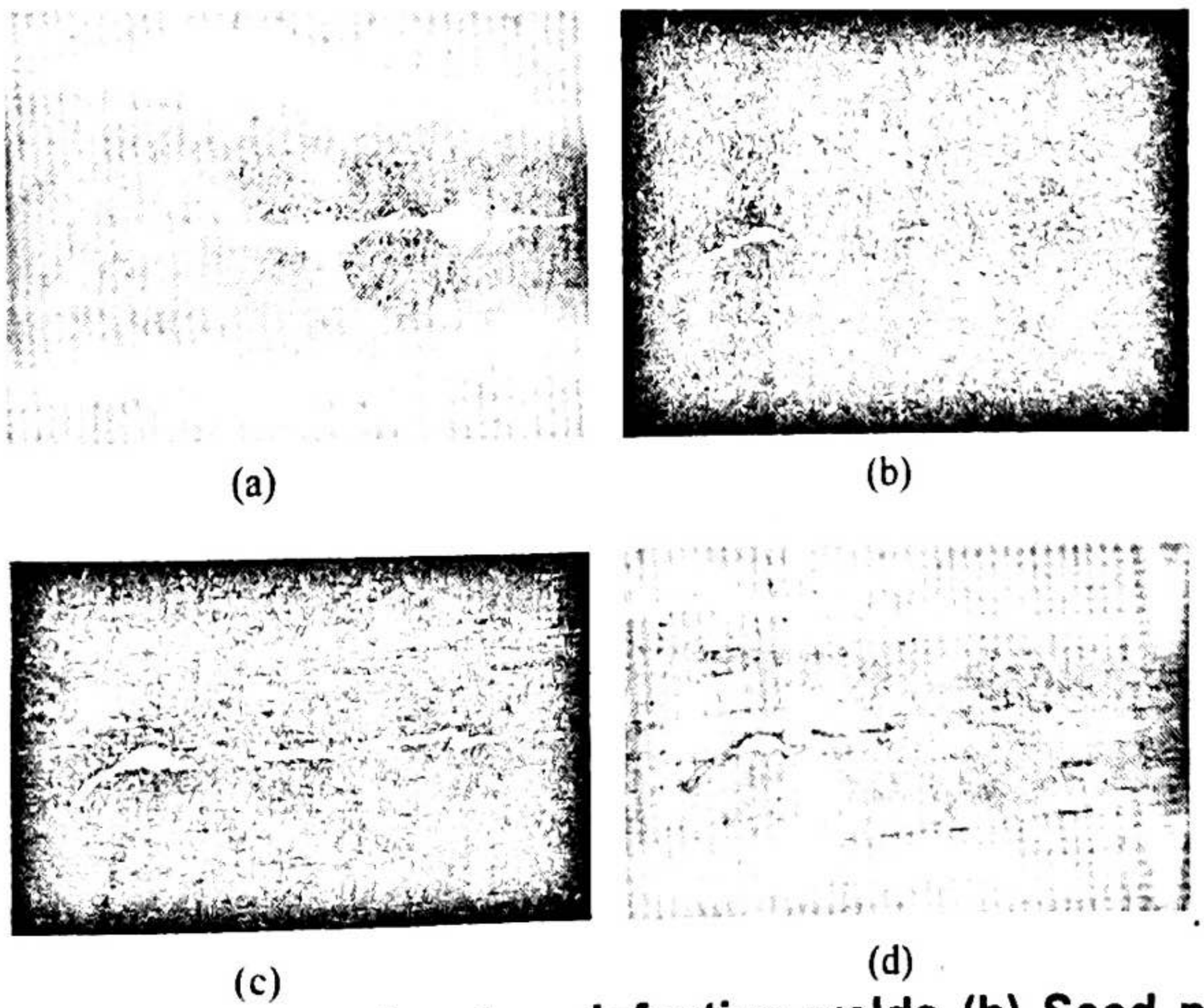
If gray level values belong to Gaussian distribution. Threshold is calculated as following when error rate is minimized.

$$T = \frac{u_1 + u_2}{2} + \frac{\sigma^2}{u_1 - u_2} \ln\left(\frac{P_2}{P_1}\right) \qquad \ldots(5.17)$$

5.6 Region-Based Segmentation

5.6.1 Region Growing

Region growing is a procedure that groups pixels or subregion into larger regions based on predefined criteria. The basic approach is to start with a set of "seed" points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed.



(a) (b)

(c) (d)

Figure 5.11 (a) Image showing defective welds (b) Seed points (c) Result of region growing (d) Boundaries of segmented defective welds

Region and merging in region growing is an important step. The result of region splitting and merging is better than threshold. Because the image segmentation by threshold just have two gray level value, but the image use region splitting and merging have similar sharp with original image. There is a procedure of region splitting and merging below.

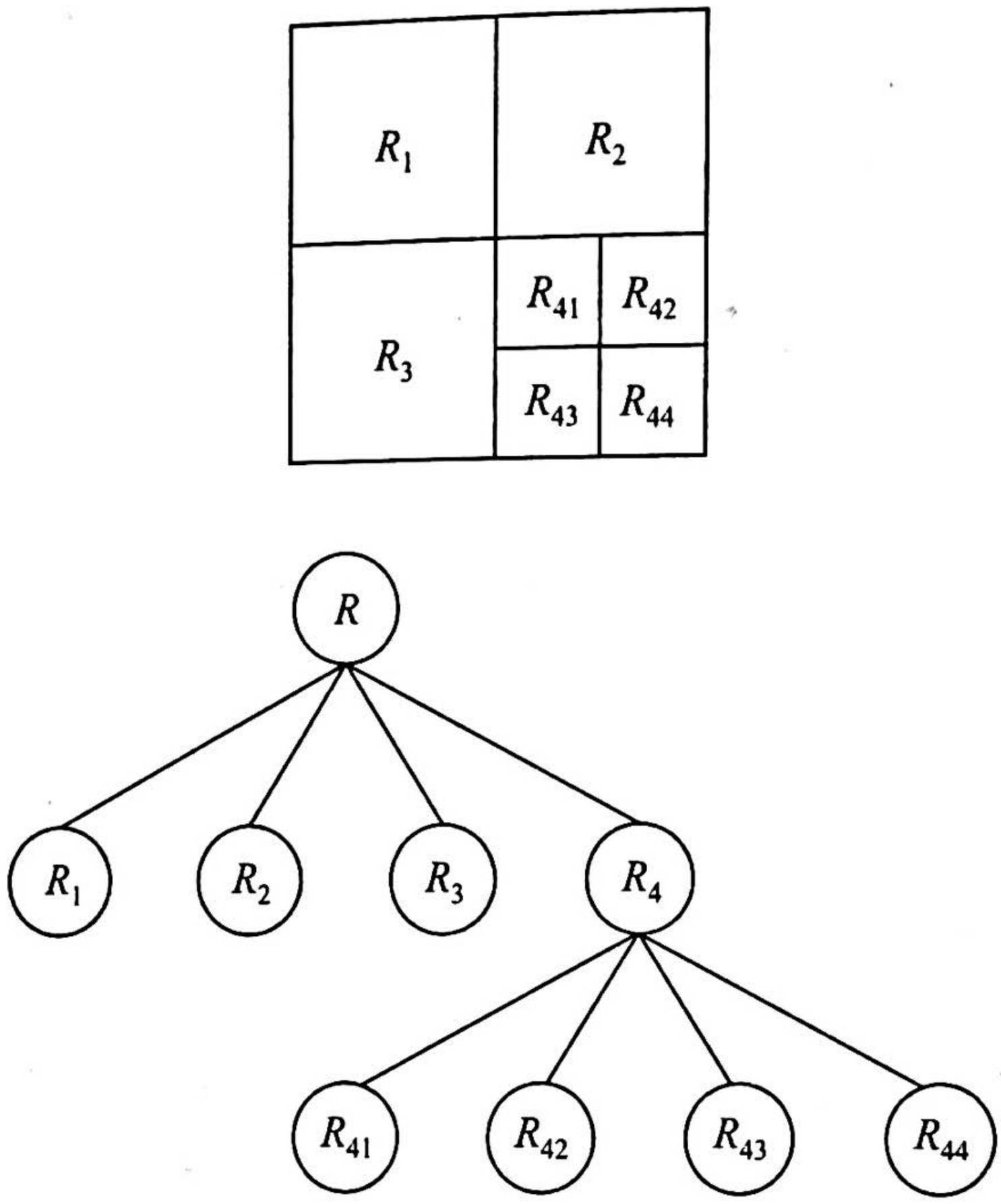


Figure 5.12 (a) Partitioned image (b) Corresponding quadtree

(a) **Unseeded Region Growing:** A characteristic of USRG different to SRG is no explicit seed selection is necessary, the seeds can be generated by the segmentation procedure automatically. The automatic segmentation is by added benefit of robustness from region-based segmentation. Advantages of it are easy to use and can readily incorporate higher level knowledge of the image composition through region threshold.

(b) **Mean Shift Algorithm:** The mean shift algorithm is a nonparametric clustering technique which does not require the prior knowledge of clusters, and does not constrain the shape of the windows and clusters. A advantage is good in shape of match and a disadvantage is the slow speed.

(c) **Fast scanning Algorithm:** The fast scanning algorithm somewhat resembles Un-Seeded Region Growing (USRG) we had mentioned above. Both two algorithms are region-based approaches and the number of clusters of both results would not be decided before an image passing through them as well. We list the steps of algorithm as below.

Step 1. Make the first pixel (Mostly, it will be the top-left one) we scan as the first clustering. See the pixel (x,1) in the image as one cluster C_i. See the pixel which we are scanning as C_j.

Step 2. From the first column, scan the next pixel (x,1+1) and make a decision with the threshold if it will be merged into the first clustering or to be a new clustering.

If C_j – centroid (c_i) ≤ threshold, we merge C_j into C_i and recount the centroid of C_i.

If, C_j - centroid (C_i) > threshold, we make C_j as a new cluster,

Step. 3. Repeat Step 2 until all the pixels in the same column have been scanned.

Step. 4. Scan the next column with pixel (x+1,1) and compare it to the region C_u which is in the upper side of it. And make the merge decision see if we have to merge pixel (x+1,1) to the region C_u.

If C_j – centroid (C_u) ≤ threshold, we merge C_j into C_u and recount the centroid of C_u.

If, C_j – centroid (C_u) > threshold, we make C_j as a new cluster C_n, where n is the cluster number so far.

Step. 5. Scan the next pixel (x+1,1+1) and compare it to the region C_u, C_l which is upper to it and in the left side of it, respectively. And make the merge decision see if we have to merge pixel (x+1,1) to anyone of them.

If C_j – centroid (C_u) ≤ threshold, and C_j - centroid (c_l) ≤ threshold,

We merge C_j into C_u, merge C_j into.

Combine the region C_u and C_l to be region C_n, where n is the cluster number so far.

Recount the centroid of C_u.

else if C_j – centroid (C_u) ≤ threshold, and C_j - centroid(C_l) > threshold,

We merge C_j into C_u, and recount the centroid of C_u.

else if C_j – centroid $(c_u) \leq$ threshold and C_j - centroid $(c_l) \leq$ threshold,

We merge C_j into C_l and recount the centroid of C_l.

else,

We make C_j as a new cluster C_n, where n is the cluster number so far.

Step. 6. Repeat Step 4 - Step 5 until all the pixels in the image has been scanned.

Step. 7. Process the small regions which are classified from step 1- step 4.

It is important to deal with the isolated small regions carefully. The algorithm provides an approach in step 5 to disappear many fragmentary results after segmenting the image. The concept is to classify the isolated small regions into the big region which is already classified and is adjacent to these isolated small regions.

5.7 Use Morphological to Improve Fast Scanning Algorithm

There are some morphological operations like erosion, dilation, thinning, opening, closing. In the below we briefly illustrate the results of those operations and how to using these ones.

Binary dilation is a primary one of morphological operations. It is a combination of two sets by vector addition of set elements. If a digital image A referred as set and kernel B is structuring element. It is said that dilation of A by B :

$$A \oplus B = \{c \in E^N \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\} \quad \ldots(5.18)$$

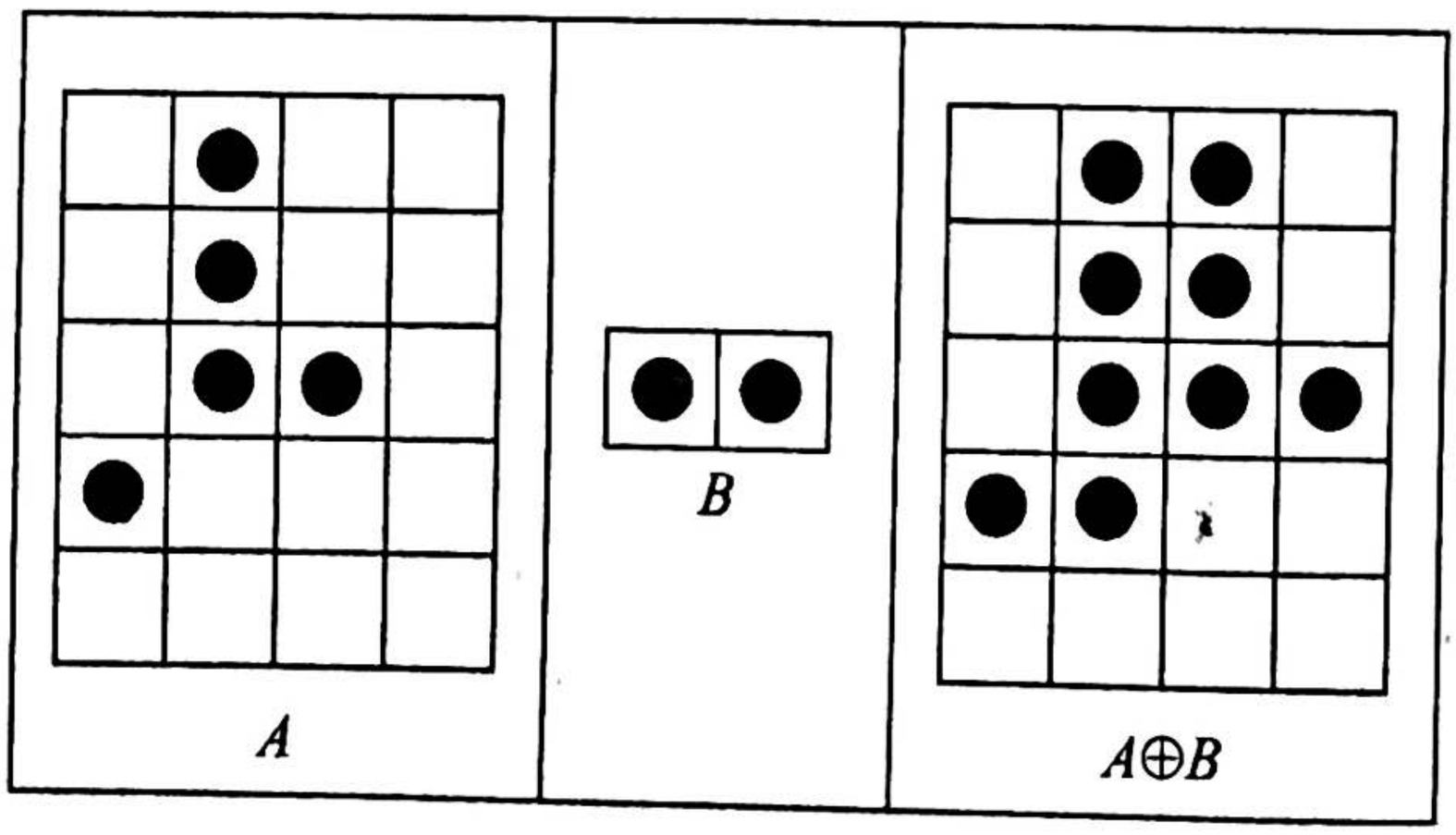


Figure 5.13 Dilation operation and their corresponding binary images. The red point is origin of *B*

Binary dilation is with below properties :

1. Dilation A by kernel with origin guaranteed to contain A.
2. Extensive: operators whose output contains input dilation extensive when kernel contains origin.
3. Dilation preserves order

$$A \subseteq B \rightarrow A \oplus K \subseteq B \oplus K \qquad \ldots(5.19)$$

4. Increasing: preserves order.
5. Binary erosion is another primary morphological operation. If a set represented by "A", and a kernel set by B. The definition of erosion of A by B is

$$A \ominus B \quad = \{x \in E^N \mid x + b \in A \text{ for every } b \in B\}$$

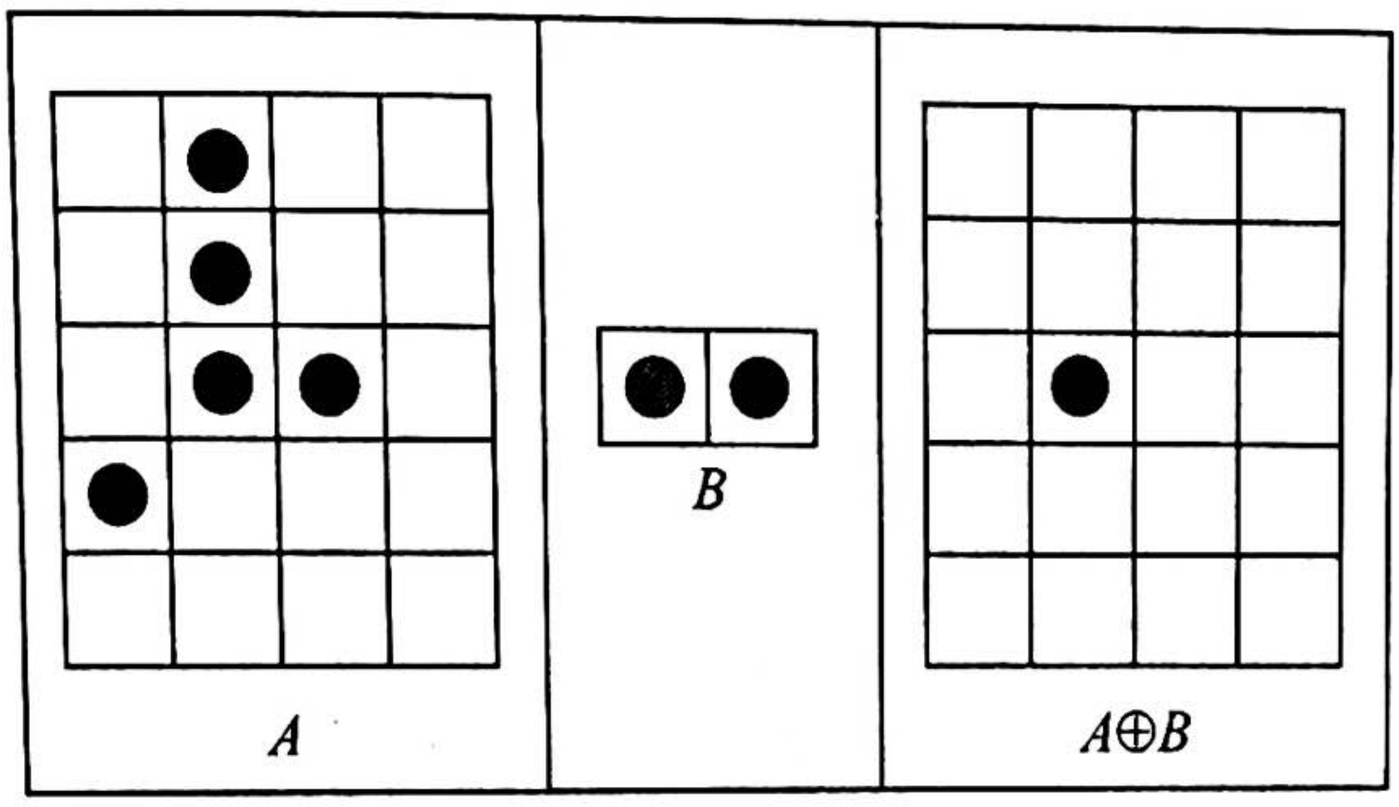


Figure 5.14 Binary erosion of a set A by a structuring element B.

The property of binary erosion is

$$K \subseteq L \Rightarrow A \ominus L \subseteq A \ominus K \qquad \ldots(5.20)$$

1. **Binary Opening and Closing:** Binary opening and closing are composed from dilation and erosion. Opening of image B by kernel K is represented as

$$B \circ K \;=\; (B \ominus K) \oplus K \qquad \ldots(5.21)$$

Opening with disk kernel would produce smoothes contours and breaks narrow isthmuses, or neck, moreover, eliminates small islands, sharp peaks, and capes. Closing operator is represented as

$$B \bullet K \;=\; (B \oplus K) \ominus K \qquad \ldots(5.22)$$

Closing with disk kernel would also produce smoothes contours but fuses narrow breaks, long, thin gulfs, in the same time, eliminates small holes and fill gaps on the contours.

2. **Gray Scale Morphology:** Not only binary image but also gray level one could be processed by morphological operations. It is obvious that definitions of gray scale morphological operations differ from binary ones. Gray scale morphological operations introduce minimum or maximum operation to perform dilation and erosion.

Grayscale dilation is represented by

$$A \oplus K(x,y) = \max(A(x-z,y-w)+K(z,w) \mid z,w \in K, x-z, y-w \in A) \quad \ldots(5.23)$$

Grayscale erosion is similarly represented by

$$A \ominus K(x,y) = \min(A(x+z,y+w)-K(z,w) \mid z,w \in K, x+z, y+w \in A) \quad \ldots(5.24)$$

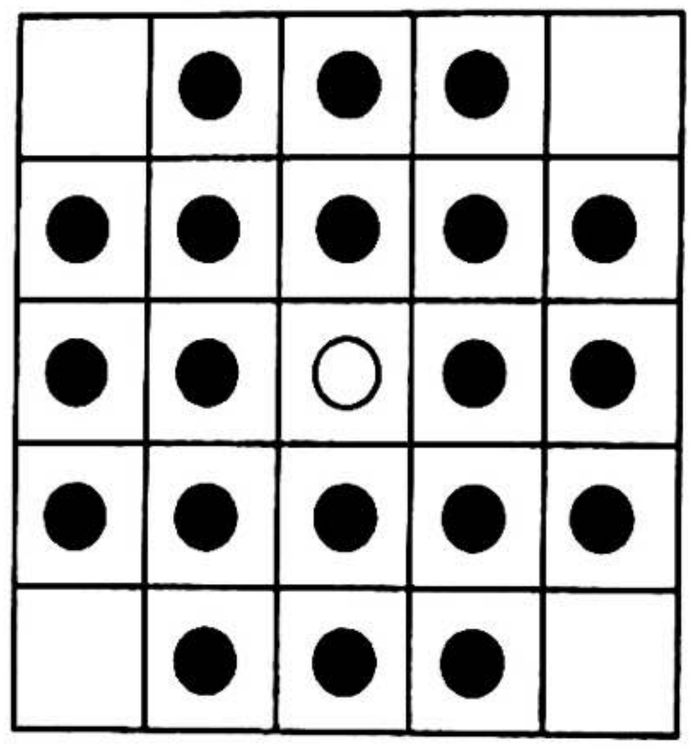


Figure 5.15 Structuring elements, the values of all points are zero

Improvement of Fast Scanning with Morphological Operation:

The outcome of fast scanning algorithm is not enough matching physical objects. For improving the result of fast scanning, we use the operations of morphology to diminish some flaws. The step of first scheme is described at figure 5.15.

Because of binary and dilation for every region are time consumptive. To shorten the consumption time, we calculate the binary erosion of every region in one time instead of one by one and consider the duality property between binary erosion and dilation, and it is second scheme. The step of second scheme is described at figure 5.16.

The flow chart of first scheme is

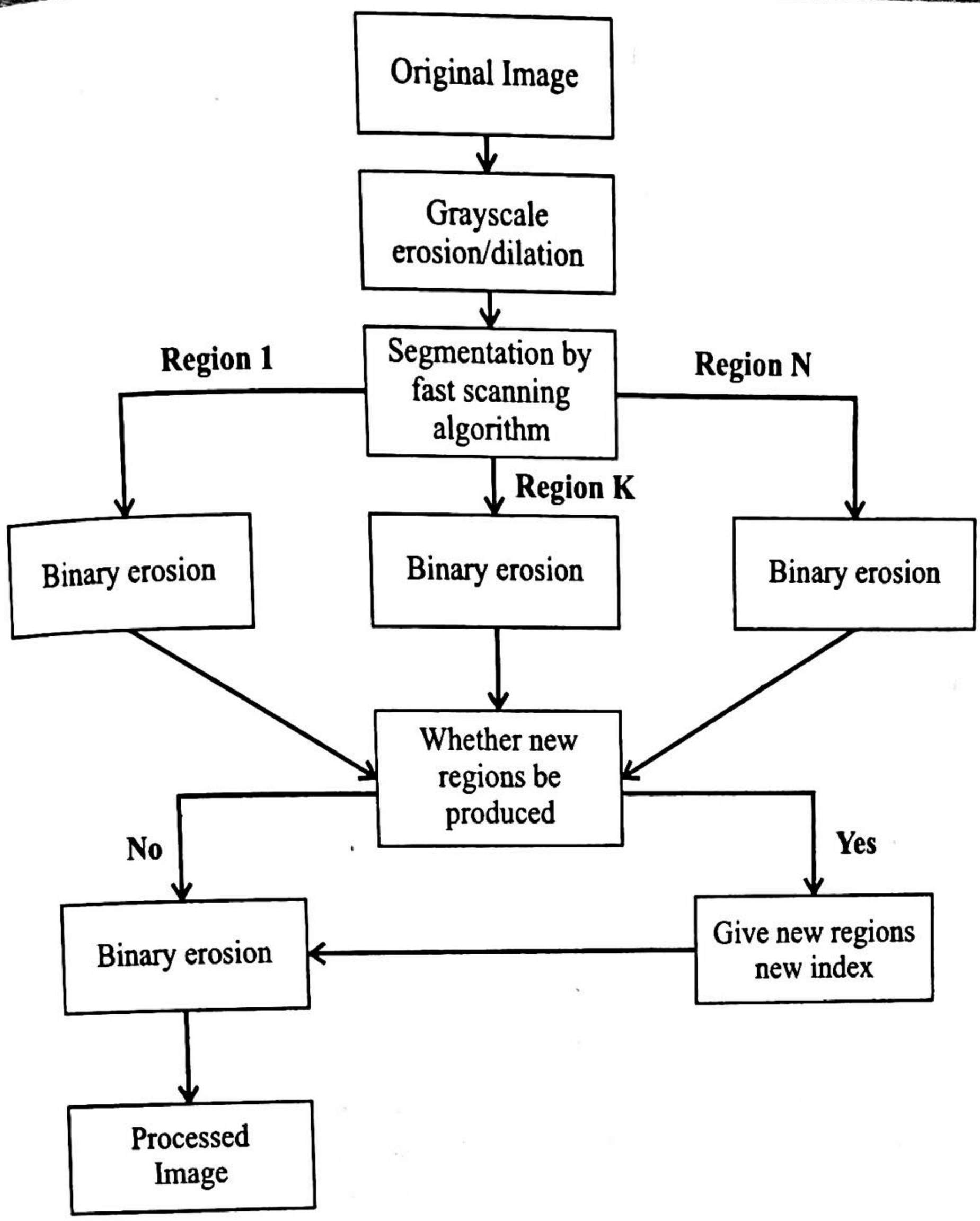


Figure 5.16 Flow chart of improvement approach morphological image

The flow chart of second scheme is

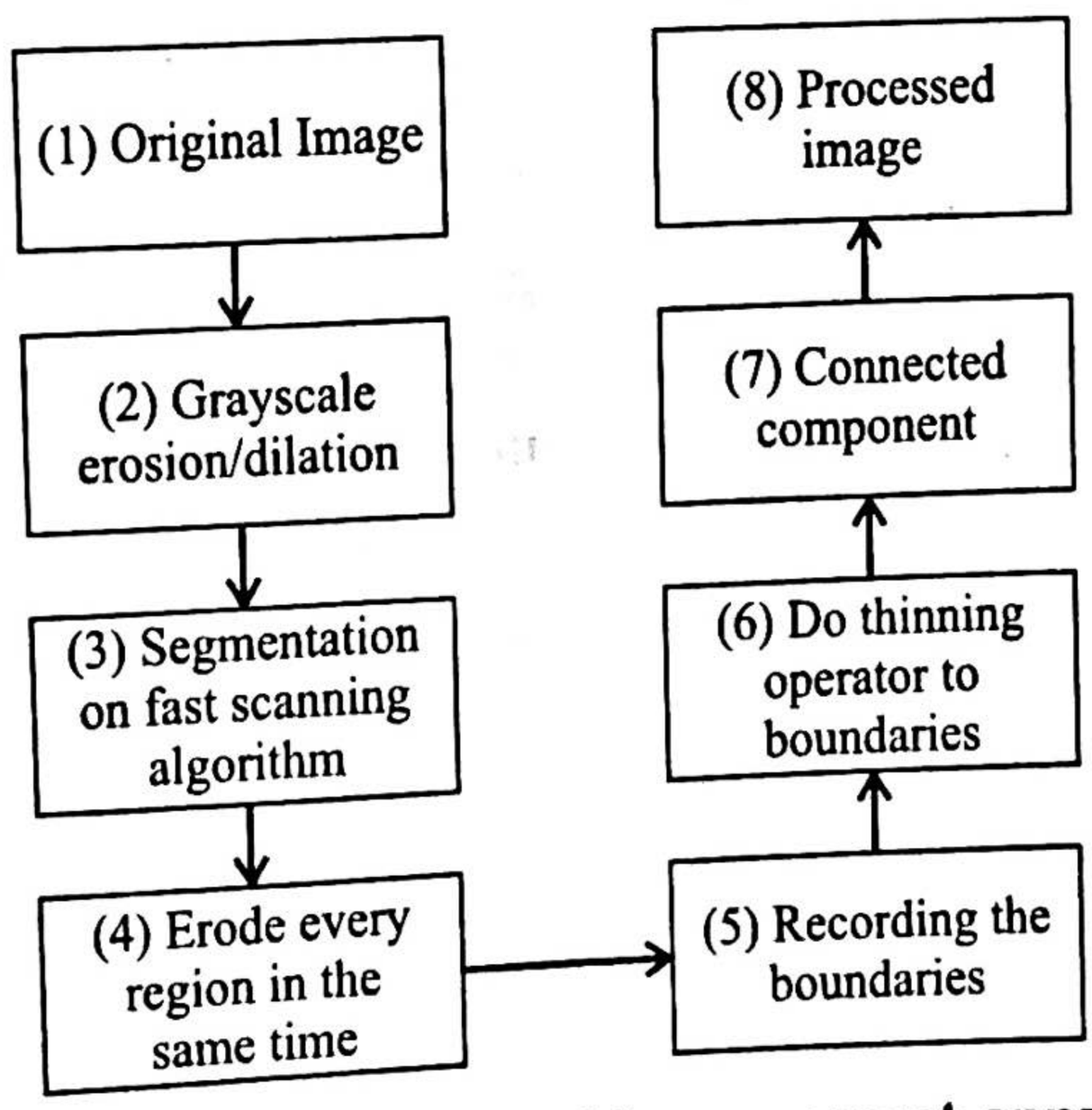


Figure 5.17 Flow chart of improvement approach

5.8 Use Geometrical Mathematic to Improve Fast Scanning

A potential issue of morphological technique is might be sustained a kernel size problem in which a too small kernel makes whole improvement structure trivial. It will confuse us that how to decide the size of kernel. In this chapter, we replace morphological technique with making use of other concept: geometrical mathematics. The approach can improve matching physical objects of fast scanning algorithm.

5.8.1 Strategy for Extra Cutting Off

While an image had been segmented, we could hypothesize that every segmented regions are all polygons. Before transforming segmented regions to polygons, we should firstly know which points in boundary of region could be seen as vertices. Supposing that all picked points of one region from split approach are vertices of polygon. Then, the following process is finding out the vertices with their interior angle which is bigger than 180 degree.

Next, we will descript the criterion of extra cutting off. Before, we had introduced the tactic of improvement in previous section, but a core issue is in which condition the region must be cut off. In previous tactic, we want to find the vertices with interior angle over 180 degree. Now, we call this "vertex". And that we introduce three matrixes below. First matrix below is called as straight line distance matrix. Second and third matrixes are individually clockwise and counterclockwise distances along boundaries of region.

$$\begin{bmatrix} 0 & 10 & 9 \\ 10 & 0 & 3 \\ 9 & 3 & 0 \end{bmatrix} \begin{bmatrix} 0 & 80 & 35 \\ 10 & 0 & 45 \\ 55 & 45 & 0 \end{bmatrix} \begin{bmatrix} 0 & 10 & 55 \\ 80 & 0 & 45 \\ 35 & 45 & 0 \end{bmatrix}$$

(a) (b) (c)

Figure 5.18 (a) A matrix of mutual straight line distances (b) a matrix of clockwise distances along boundaries of region (c) a matrix of counterclockwise distances along boundaries of region

After this three type matrixes are found. We calculate the minimum elements matrix.

$$\begin{bmatrix} 0 & 10 & 35 \\ 10 & 0 & 45 \\ 35 & 45 & 0 \end{bmatrix} = \min_{element\ by\ element}\left(\begin{bmatrix} 0 & 80 & 35 \\ 10 & 0 & 45 \\ 55 & 45 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 10 & 55 \\ 80 & 0 & 45 \\ 35 & 45 & 0 \end{bmatrix}\right) \quad \ldots(5.25)$$

The final step is to calculate the ratio matrix of minimum of distance equation (5.25) to mutual distance. The matrix is showed in below :

$$\begin{bmatrix} NaN & 1 & 3.89 \\ 1 & NaN & 15 \\ 3.89 & 15 & NaN \end{bmatrix} = \left(\begin{bmatrix} 0 & 10 & 35 \\ 10 & 0 & 45 \\ 35 & 45 & 0 \end{bmatrix} \Big/ \begin{bmatrix} 0 & 10 & 9 \\ 10 & 0 & 3 \\ 9 & 3 & 0 \end{bmatrix} \right) \quad \text{...(5.26)}$$

This matrix is the final determination matrix, in this thesis we define a pre-threshold. The geometric meaning of threshold is the acceptable minimum ratio of boundary-along distance to straight geometric distance among two points. And the threshold equal to 10.

Compare every element in equation 5.26 with the threshold. The following is a determination rule, indeed, as equation 5.26 is symmetric matrix, we could just focus on upper triangle elements. Also, according to the definition of ratio matrix, diagonal elements are concerning relation between vertex and itself. It is trivial for discussion.

(a) If exist just only one element (i,j) is bigger than threshold, where, $i>j$ then picking up point i and point j. And connect these two vertices in order to cutting off the objective region into two parts.

(b) If there exist more than or equal to two elements (i_1,j_1), (i_2,j_2) , ... (i_n, j_n) are bigger than threshold, where $i_k>j_k$ for k =2,3,....n, then attempting to find out the maximum :

$$(i_M, j_M) = \text{Max } [(i_1, j_1), (i_2, j_2) , \ldots\ldots (i_n, j_n)] \quad \text{...(5.27)}$$

Connecting these two vertices, vertex i_M and vertex j_M, to cut region off into parts.

(c) If these exist no element (i, j) are bigger than threshold, then the objective region would be reserved.

In the case, the ratio distance matrix of this region is showed in figure (5.18). It agree with condition a, only element is bigger than threshold. The connection among B-vertex and C-vertex must be linked so that this region would be decomposed into two disjoint parts as showed in figure 5.18.

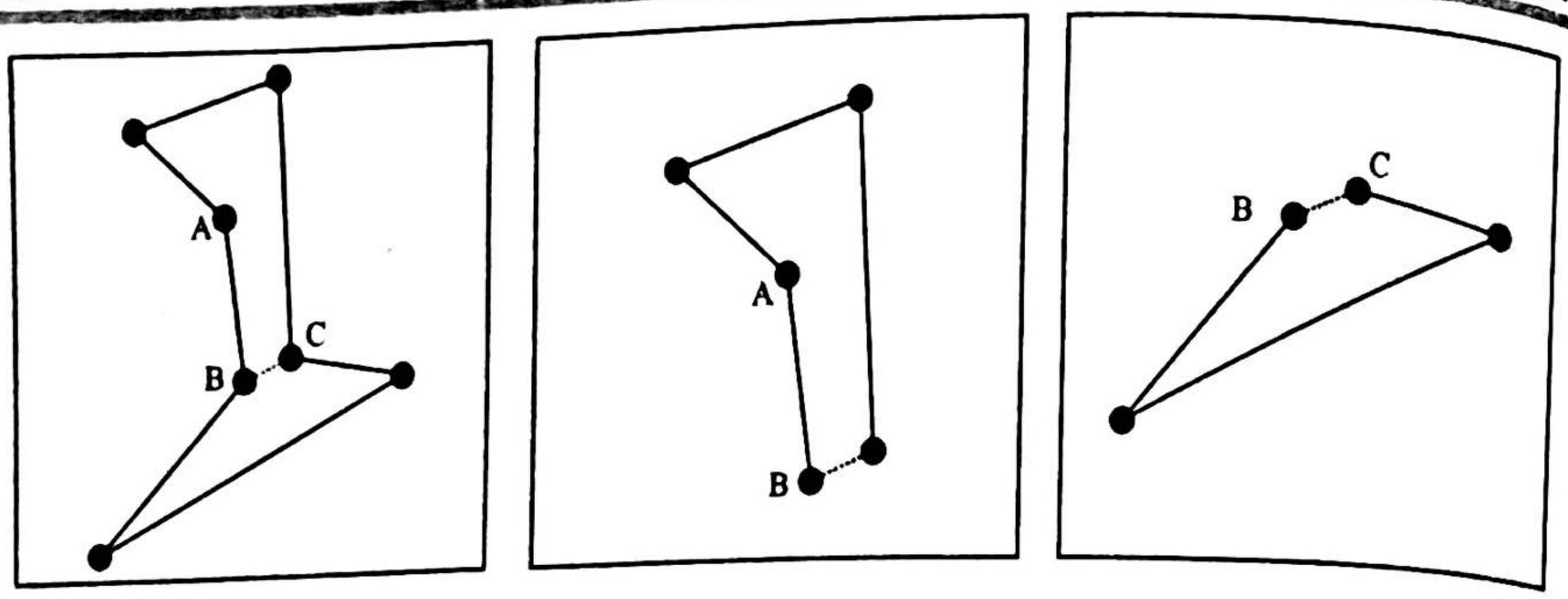


Figure 5.18 Region in (a) is separated into two parts (b), (c).

5.9 Segmentation by Morphological Watersheds

When the rising water in distinct catchment basins is about to merge, a dam is built to prevent the merging. The flooding will eventually reach a stage when only the tops if the dams are visible above the water line.

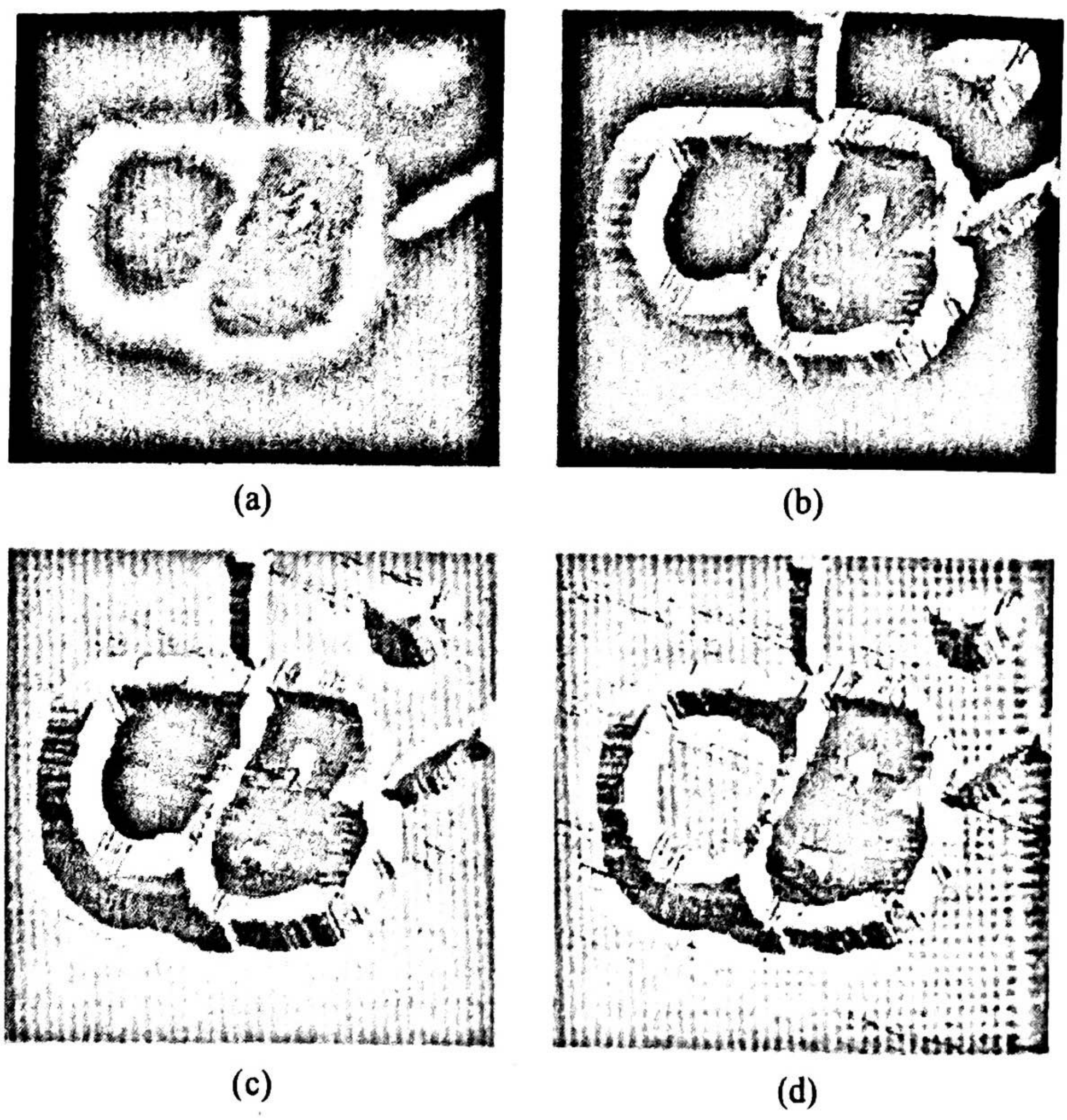


Figure 5.19 (a) Original image (b) Topographic view (c) - (d) Two stages of flooding

5.9.1 Dam Construction

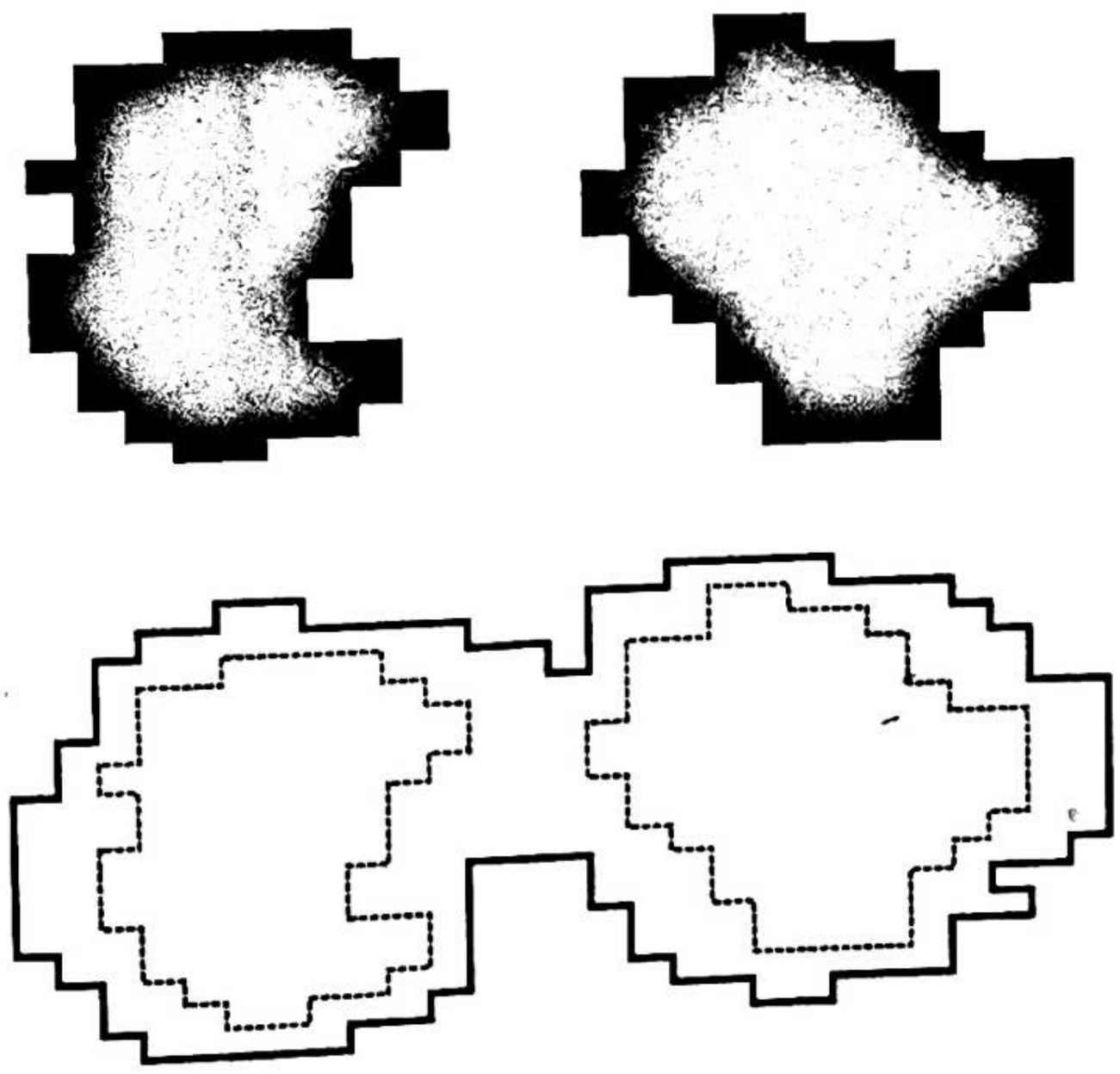
Dam or watershed lines construction is based on binary images which are members of 2-D integer space z^2. It can be achieved by separating sets of binary points using dilation.

Figure 5.20 gives the construction of dams using dilation process. Initially achieve the portions of two catchment basins at flooding step n-1, after that obtain the next flooding step n. Dam is required to avoid water spilled from one basin to another.

The two regional minimas $c_{n-1}(m_1)$ and $c_{n-1}(m_2)$ are used to represent gray regions. It is to be noted that all points belonging to an individual catchment basin form a single connected component. These connected components is to be dilated by the structuring element.

This dilation can be done on two ways dilation has to be constrained to connected components i.e., the center of the structuring element can be located only at points in connected components and the dilation cannot be performed on point that would cause the sets being dilated to merge.

Figure 5.20 shows that first condition is satisfy by every point during dilation and second condition didn't apply to any point during dilation; that why boundary of each region was expanded uniformly.



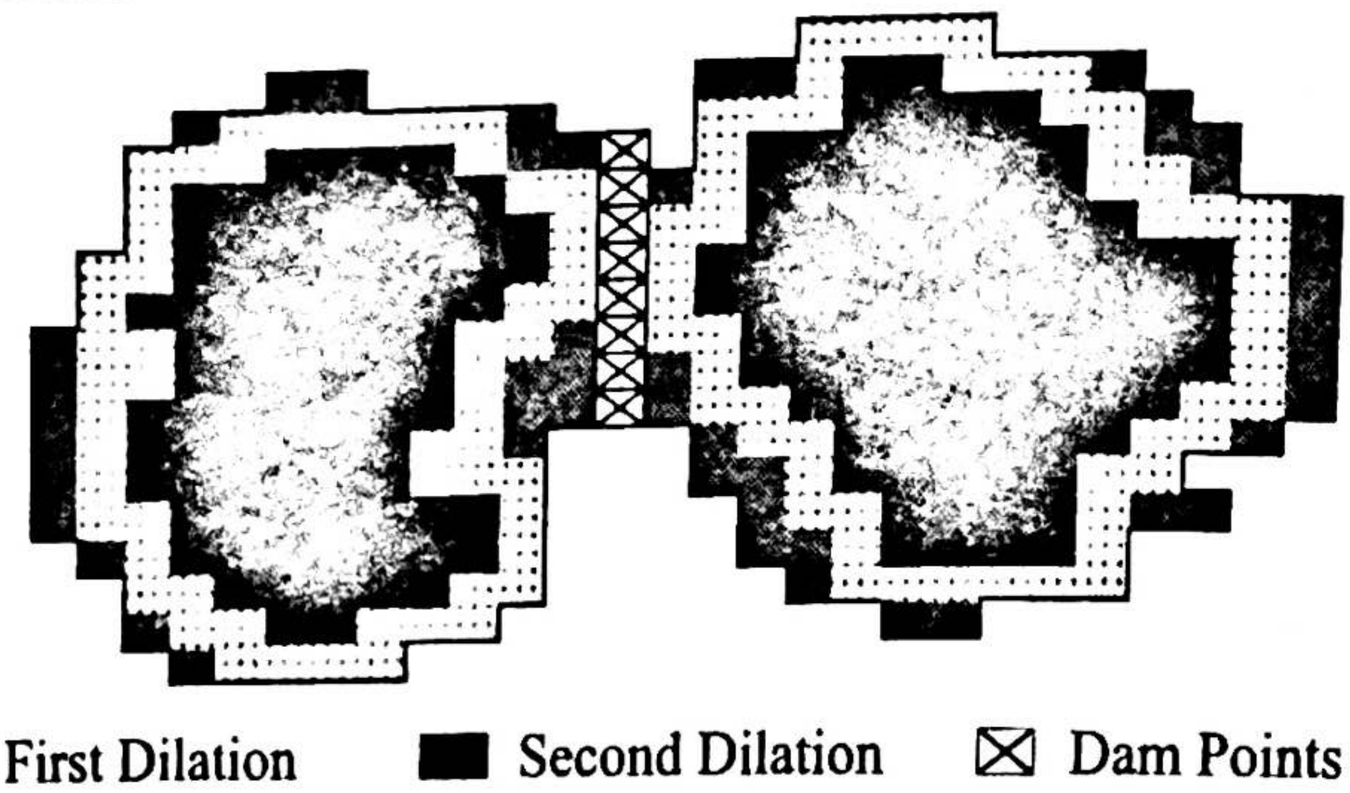
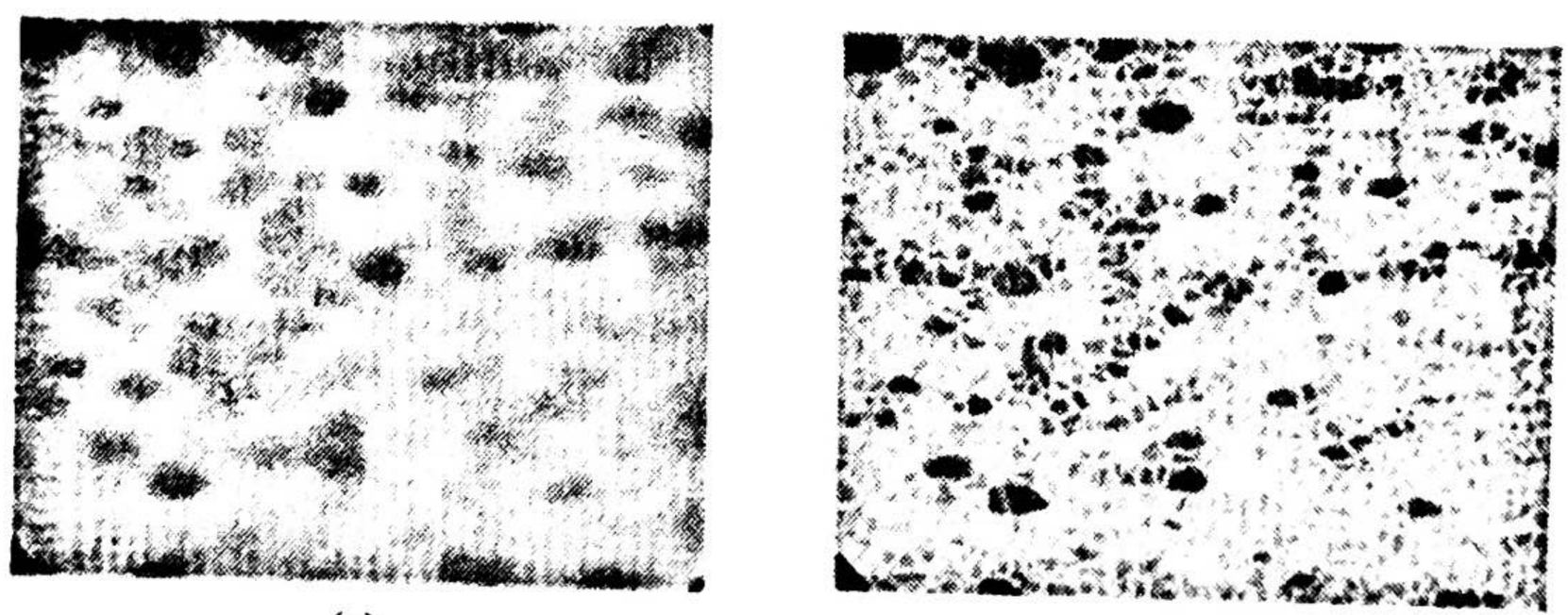


Figure 5.20 (a) Two partially flooded catchment basins at stage n-1 of flooding (b) Flooding at stage n, showing that water has spilled between basins (c) Structuring element used for dilation (d) Result of dilation and dam construction

5.9.2 The Use of Markers

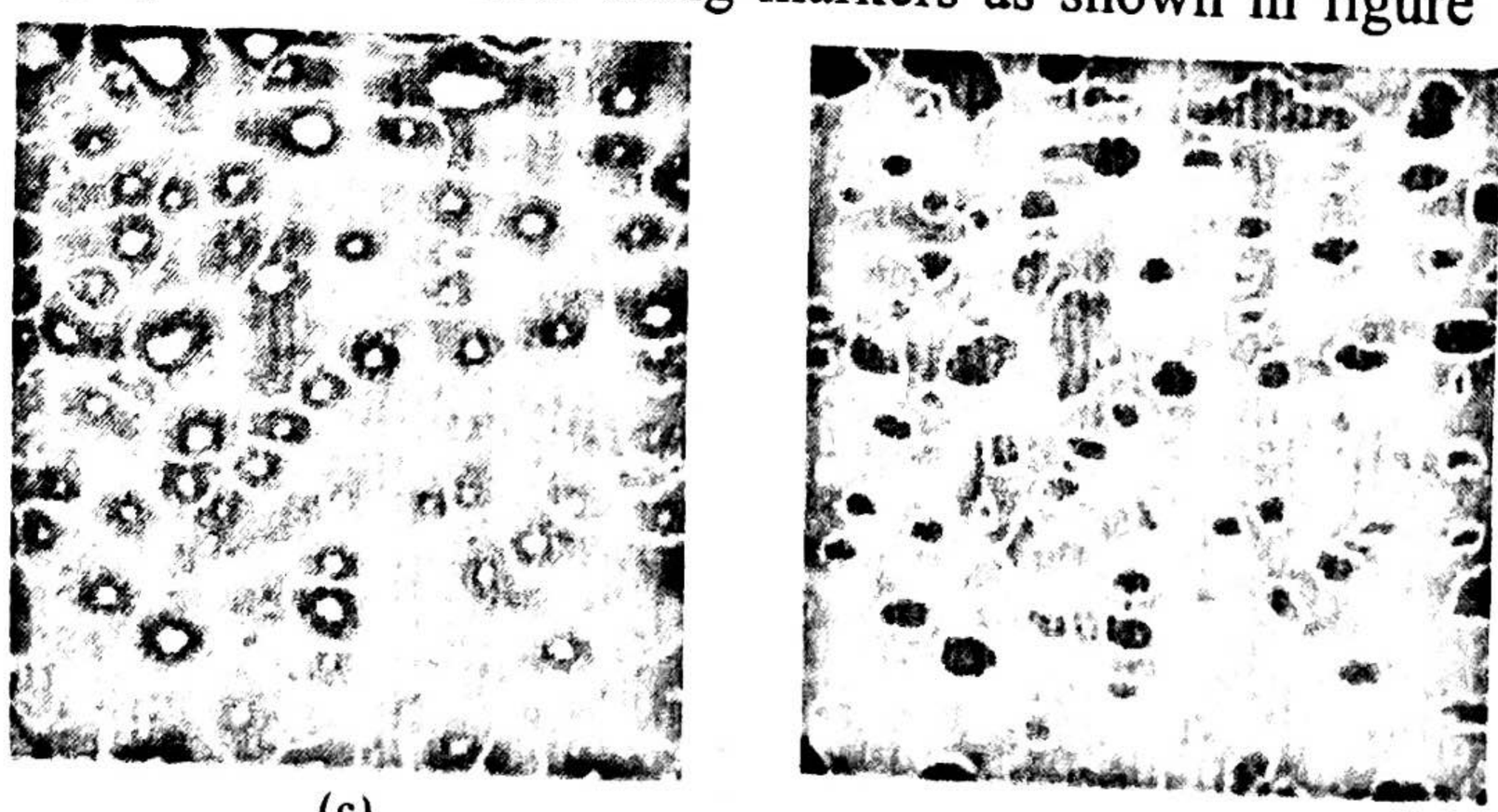
The graph below is after using markers as shown in figure 5.21:



(a) (b)

Figure 5.21 (a) Electrophoresis image (b) Result of applying the watershed segmentation algorithm to the gradient image

The graph below is after using markers as shown in figure 5.22 :



(c) (d)

Figure 5.22 : (c) Image showing internal markers and external markers (d) Result of segmentation

5.10 Representation

The segmentation techniques usually consider the pixel along a boundary and pixel contained in the region. And an approach to obtain the descriptor that is compact the data into representation. There are several approaches above.

5.10.1 Chain Code

The chain code is used to represent a boundary by the length and the direction of straight-line segments. Typically, this representation is based on 4- or 8- connectivity of the segments.

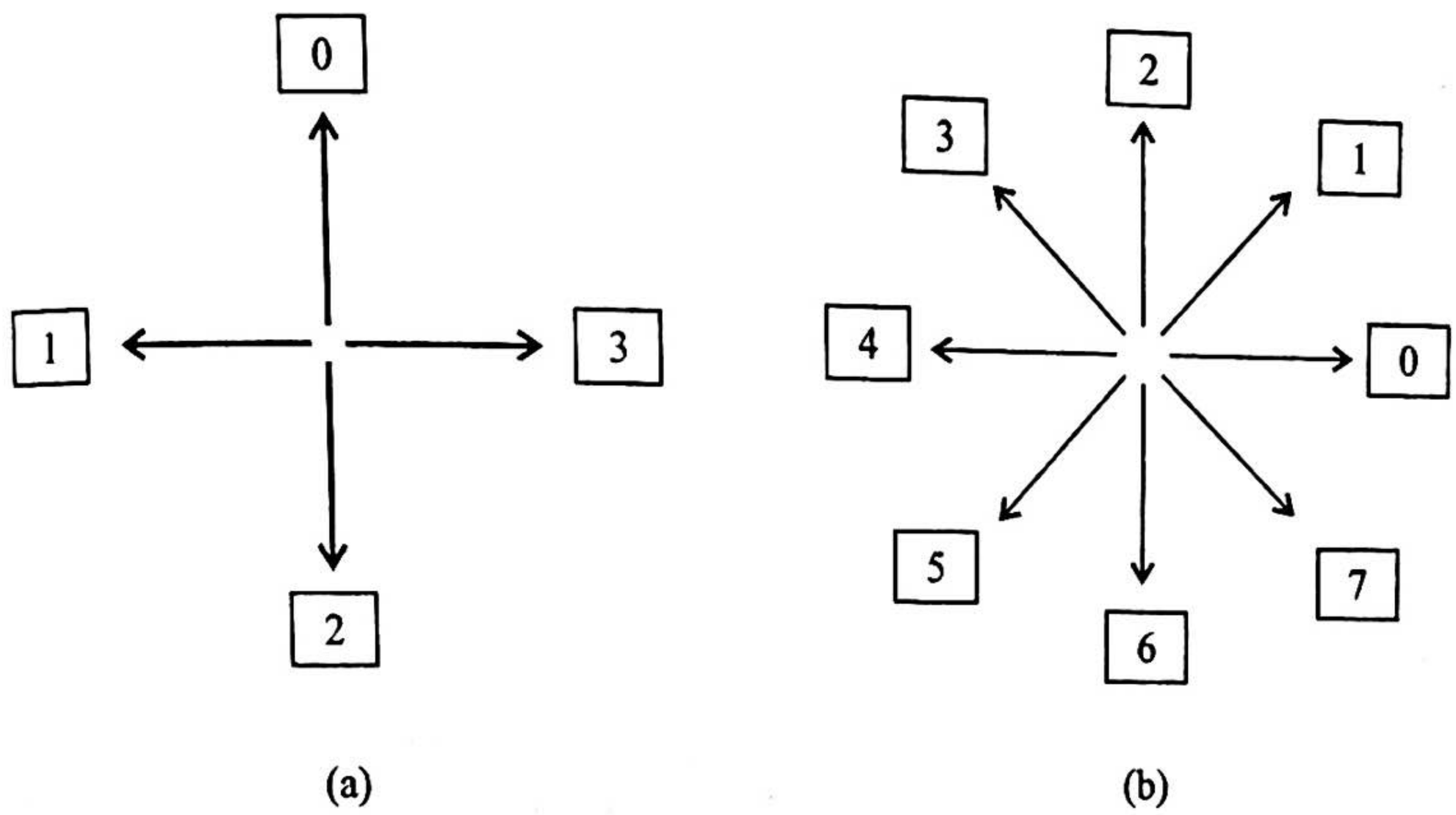


Figure 5.23 (a) 4-directional chain code (b) 8-directional chain code.

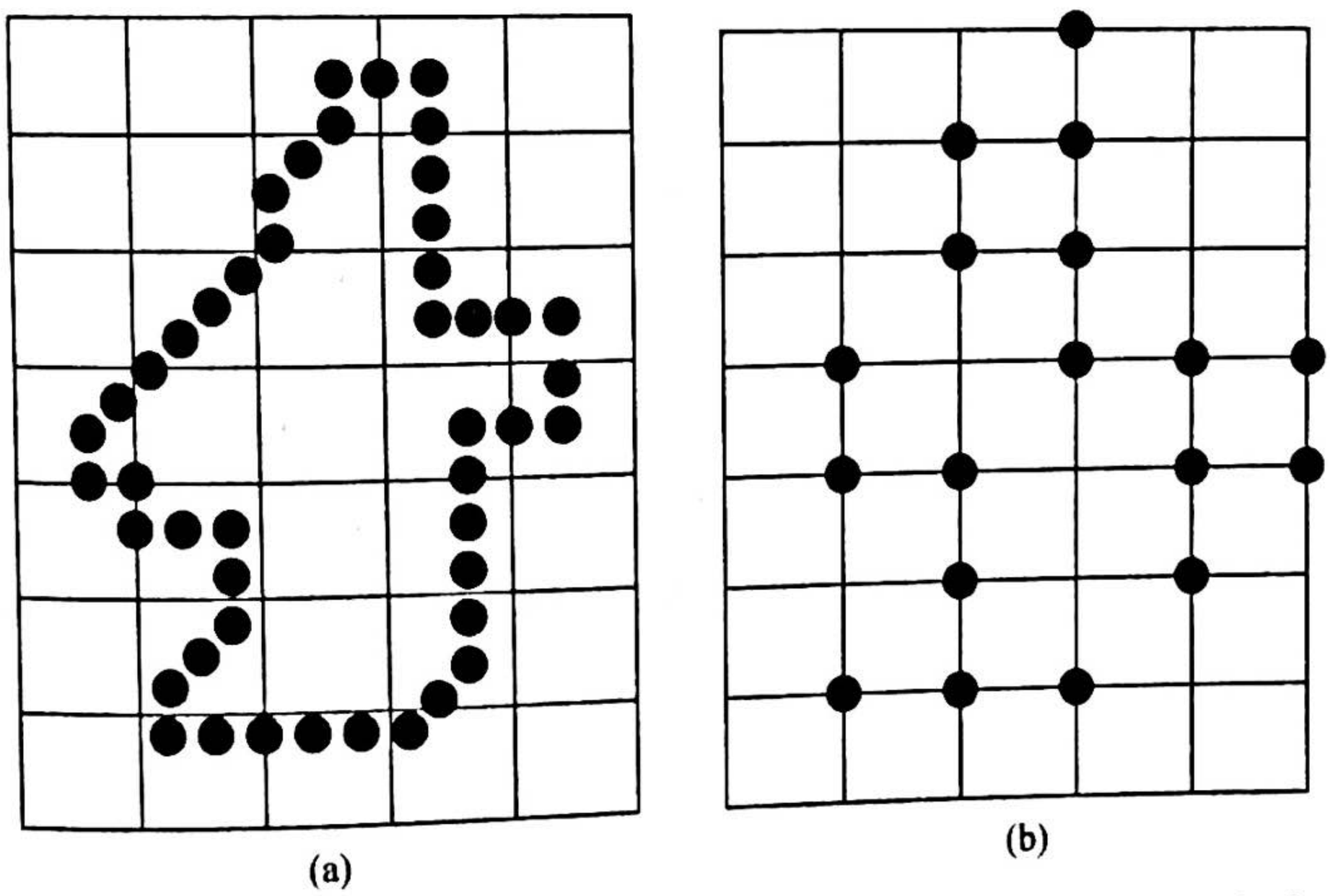


Figure 5.24 (a) 4 - directional chain code (b) 8 - directional chain code.

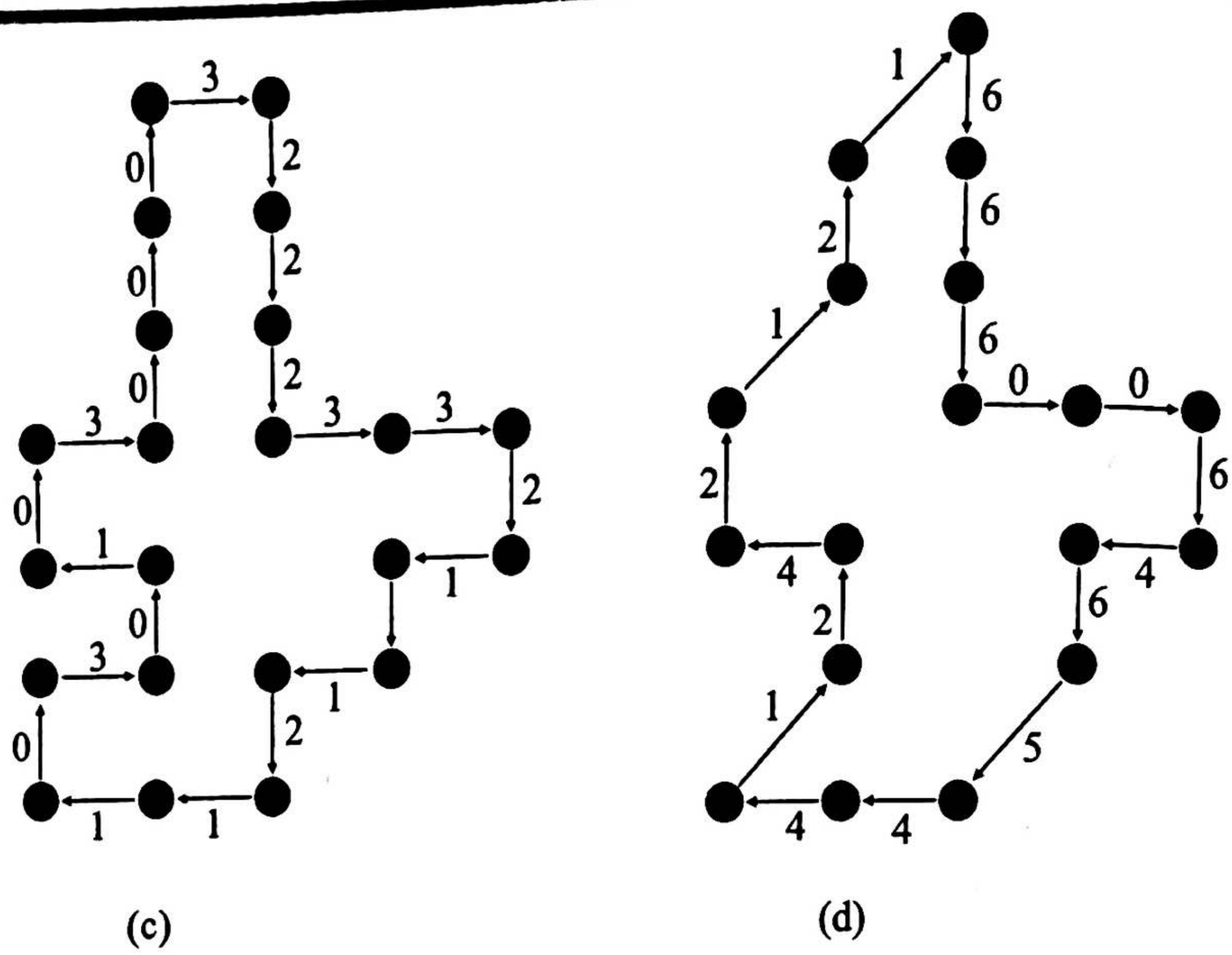


Figure 5.25 : (a) Digital boundary with resampling grid srperimposed. (b) Result of resample. (c) 4-directional chain code. (d) 8-directional chain code

Digital images are usually processed in a grid format, but it may not match shape of the boundary if the chain of codes is quite long or the boundary is distributed by the noise. An approach frequently used to circumvent the problems just discussed is to resample the boundary by selecting a larger grid spacing. It can be seen that the accuracy and samples is related to grid spacing.

5.10.2 Polygonal Approximations

The goal of polygonal approximation is to capture the essence of the boundary shape with the fewest possible polygonal segments. Several polygonal approximation techniques of middle complexity and processing requirements are suitable for image processing applications.

(a) Minimum Perimeter Polygons: We visualize this enclosure as two walls corresponding to the outside and inside boundaries of the strip of cells, and think of the object boundary as a rubber band contained within the wall. If the rubber band is allowed to shrink, it takes the shape shown in figure 5.26.

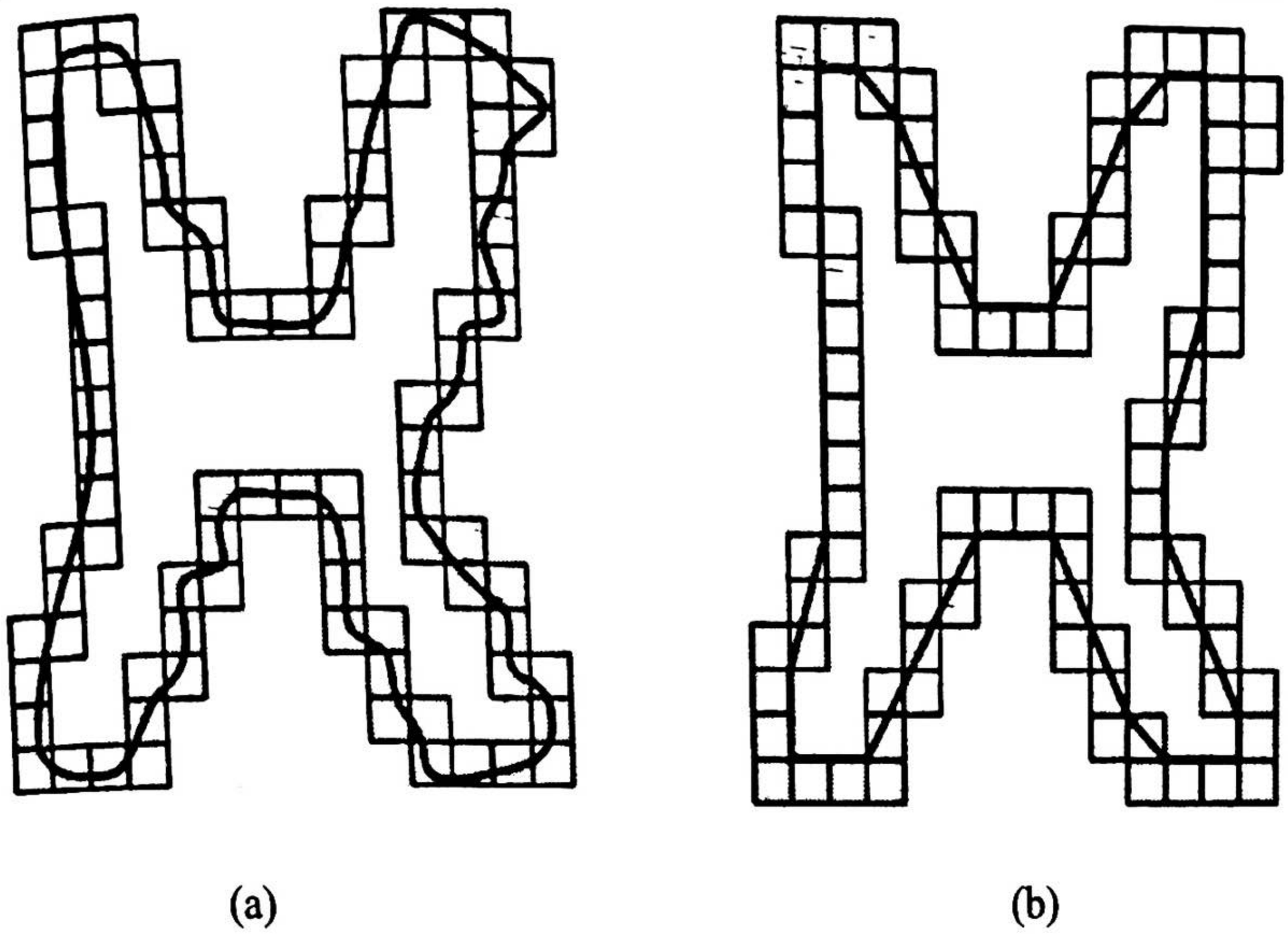


Figure 5.26 (a) Object boundary enclosed by cells, and (b) Minimum perimeter polygon.

(b) **Merging Techniques:** Merging techniques based on average error or other criteria have been applied to the problem of polygonal approximation. The approach is to merge points along a boundary until the least square error line fit of the points merged so far exceeds a preset threshold.

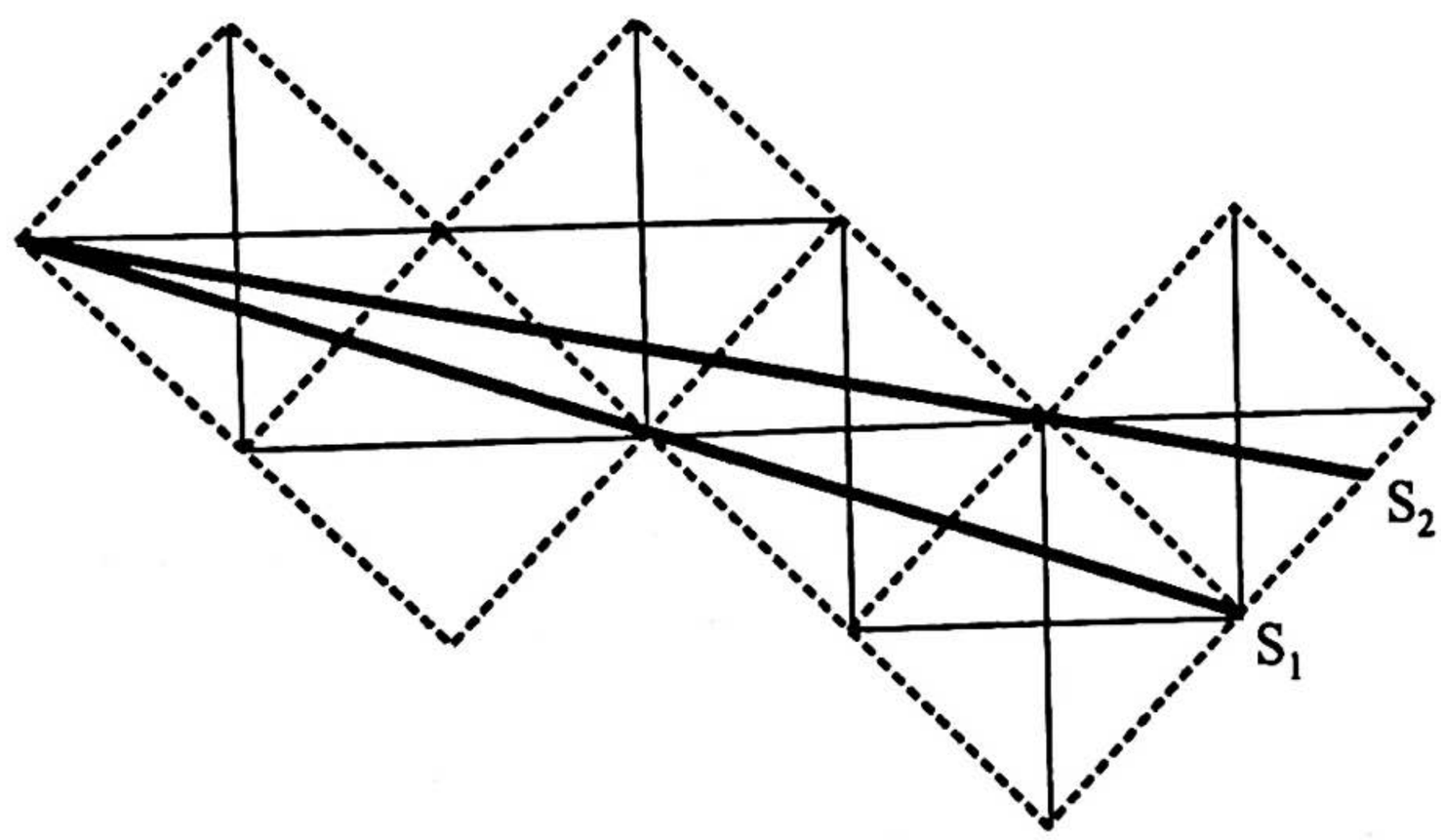


Figure 5.27 Merge approach of a set of digital boundary

(c) **Splitting Techniques:** An approach of the splitting techniques is subdivided a segment successively into two part until a criterion is satisfied. For instance, a requirement might be that the maximum perpendicular distance from a boundary segment to the line joining its two end point not exceed a preset threshold.

5.11 Signature

Signature is a approach that translate 2-D function to 1-D function. One of the simplest is to plot the distance from the center to the boundary as a function of angle, as illustrated in figure 5.28.

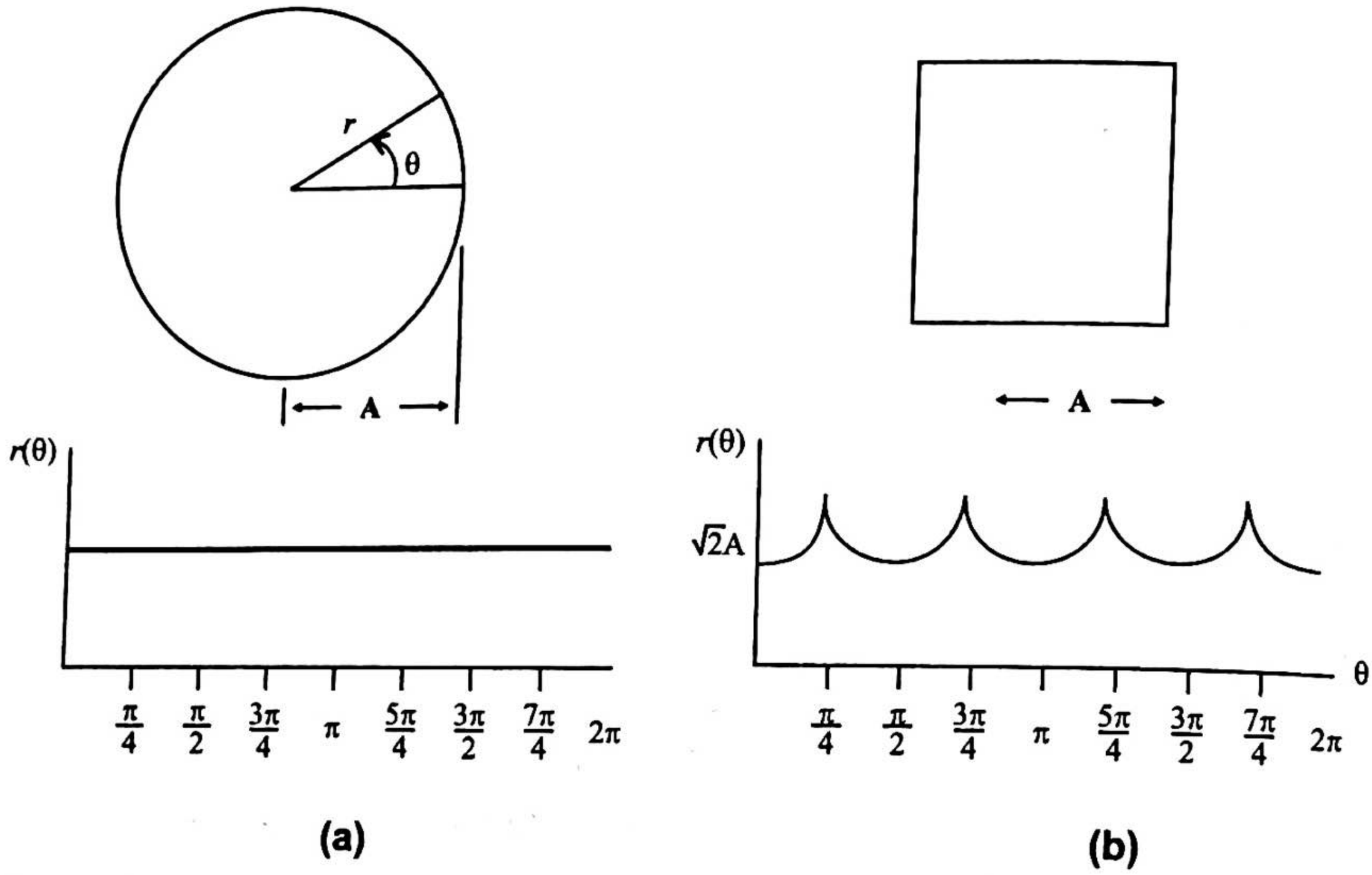


Figure 5.28 (a), (b): Distance signature of circle and rectangular shapes.

5.12 Boundary Segments

Decomposing a boundary into segments often is useful. Decomposition reduces the boundary's complexity. This approach is useful when the boundary contains one or more concavities. In this case use of the convex hull of the region enclosed by the boundary is a powerful tool for robust decomposition of the boundary.

5.13 Skeletons

Skeletons mean thinning. The skeleton of a region may be defined via the MAT (Medial Axis Transformation) proposed by Blum (1967). According to the definition, the step 1 flags a contour point for deletion if the following conditions are satisfied :

1. $2 \leq N(p_1) \leq 6$
2. $T(p_1) = 1$

3. $p_2 \cdot p_4 \cdot p_6 = 0$

4. $p_4 \cdot p_6 \cdot p_8 = 0$

...(5.28)

Where $N(p_1)$ is the is the number of nonzero neighbors of (p_1) : that is,

$$N(p_1) = p_2 + p_3 \ldots\ldots\ldots + p_8 + p_9 \quad \ldots(5.29)$$

p_1	p_2	p_3
p_4	p_5	p_6
p_7	p_8	p_9

Figure 5.28 Neighborhood arrangement used by the thinning algorithm

In section 2, conditions (1) and (2) remain the same, but conditions (3) and (4) are changed to

$$p_2 \cdot p_4 \cdot p_8 = 0$$

$$p_2 \cdot p_6 \cdot p_8 = 0$$

The iteration of the thinning algorithm consists of (1) applying step 1 to flag border points for deletion; (2) deleting the flagged points; (3) applying step 2 to flag the remaining border points for deletion; and (4) deleting the flagged points.

5.14 Descriptor

In this section, we discuss boundary descriptors and regional descriptors.

5.14.1 Boundary Descriptors

There are several approaches to describe the boundary of a region. We illustrate as follows.

(a) **Some Simple Descriptors:** There are several simple descriptors. Just like basic rectangular, eccentricity, curvature and so on. Basic rectangular is the rectangular that generate by major axis and minor axis. Eccentricity is the ratio of major axis and minor axis. Curvature is defined as the rate of the boundary that use convex and concave to describe the boundary.

(b) **Shape Numbers:** The first difference of a chain-coded boundary depends on the starting point. The shape number of such a boundary, based on the 4-Directional code is defined as the first difference of

smallest magnitude. For a desired shape order, we find the rectangle of order n whose eccentricity best approximates that of the basic rectangle and use this new rectangle to establish the grid size.

(c) **Fourier Descriptors:** The Fourier descriptors are starting at an arbitrary point. Each coordinate pair can be treated as a complex number so that

$$s(k) = x(k) + jy(k) \qquad \ldots(5.30)$$

This representation has one great advantage that it reduces a 2-D to a 1-D problem. The Discrete Fourier Transform (DFT) of $s(k)$ is

$$a(u) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/} \qquad \ldots (5.31)$$

For $u = 0, 1, 2, \ldots, K-1$. The complex coefficients $a(u)$ are called the Fourier descriptors of the boundary. The inverse Fourier transform of these coefficients restore $s(k)$. That is,

$$s(k) = \sum_{u=0}^{K-1} a(u)e^{j2\pi uk/} \qquad \ldots(5.32)$$

For $k = 0,1,2,\ldots,k-1$. Suppose, however, that instead of all the Fourier coefficients, only the first P coefficient is used. This is equivalent to setting $a(u) = 0$ for $u > P-1$. The result is the following approximation to $s(k)$:

$$\hat{s}(k) = \sum_{u=0}^{P-1} a(u)e^{j2\pi uk/} \qquad \ldots(5.33)$$

The smaller P becomes, the more detail that is lost on the boundary. And the bigger P becomes, it more similar to boundary.

(d) **Statistical Moments:** The shape of boundary segments can be described quantitatively by using simple statistical moments, such as the mean, variance, and higher-order moments. Let us treat the amplitude of g as a discrete random variable v and form an amplitude histogram $p(v_i)$, $i = 0,1,2,\ldots A-1$, where A is the number of discrete amplitude increments in which we divide the amplitude scale, and $p(\ )$ is the probability of value v_i. The equation of nth moment about its mean is

$$u_n(v) = \sum_{i=0}^{A-1} (v_i - m)^n p(v_i) \qquad \ldots (5.34)$$

Where

$$m = \sum_{i=0}^{A-1} v_i p(v_i) \qquad \ldots(5.35)$$

An alternative approach is to normalize $g(r)$ to unit area and treat it as a histogram. In other words, is now treated as the probability of value occurring. In this case, $\underline{r}$ is treated as the random variable and the moments are

$$u_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i) \qquad \ldots(5.36)$$

where

$$m = \sum_{i=0}^{K-1} r_i g(r_i) \qquad \ldots(5.37)$$

In this notation, K is the number of points on the boundary, and is directly related to the shape of . Basically, what we have accomplished is to reduce the description task to that of describing 1-D functions. The advantage of moments over other techniques is that implementation of moments is carry a physical explain of boundary shape and this approach is insensitive to rotation.

5.14.2 Regional Descriptors

In this section, we consider various approaches for describing image regions.

(a) **Some Simple Descriptor:** The area and the perimeter are sometimes used as descriptors. A more frequent use of these two descriptors is in measuring compactness of a region.

(b) **Topological Descriptor:** Topological is the study of properties of a figure that are unaffected by any deformation, as long as there is no tearing. It can be defined by the number of holes, and connected component. In this section, we define the Euler formula :

$$E = V - Q - F = C - H \qquad \ldots(5.38)$$

Where

V = the number of vertices

Q = the number of edges

F = the number of faces

C = the number of connected component

H = the number of holes

(c) **Texture:** An important approach to region description is to quantify its texture content. There are three approaches introduced above.

(i) **Statistical approaches:** Statistical approaches yield characterizations of textures as smooth, coarse, and grainy, and so on. One of the approaches for describing texture is to use statistical moments of the gray-level histogram of an image or region. The nth moment of z about the mean is

$$u_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \qquad \ldots(5.39)$$

Where

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \qquad \ldots(5.40)$$

The second moment is important. It is a measure of gray-level contrast that can be used to establish descriptors of relative smoothness. For example, the measure of constant intensity

$$R = 1 - \frac{1}{1+\sigma^2(z)} \qquad \ldots(5.41)$$

The third moment is a measure of the skewness of the histogram. The fourth moment is a measure of its relative flatness. The fifth and higher moments are not so easily related to histogram shape. Some useful approaches established in the histogram is measuring of uniformity

$$U = \sum_{i=0}^{L-1} p^2(z_i) \qquad \ldots(5.42)$$

and measuring of an average entropy

$$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \qquad \ldots(5.43)$$

(ii) **Structural Approaches:** Structural techniques deal with the arrangement of image primitives. Suppose that we have a rule of the form $S \rightarrow aS$, which indicates that the symbol S may be rewritten as aS. If a represents a circle as shown in figure 5.29 (a) and the meaning of "Circles to the Right" is assigned to a string of the form aaa...., the rule $S \rightarrow aS$ allows generation of the texture pattern shown in figure 5.30 (b).

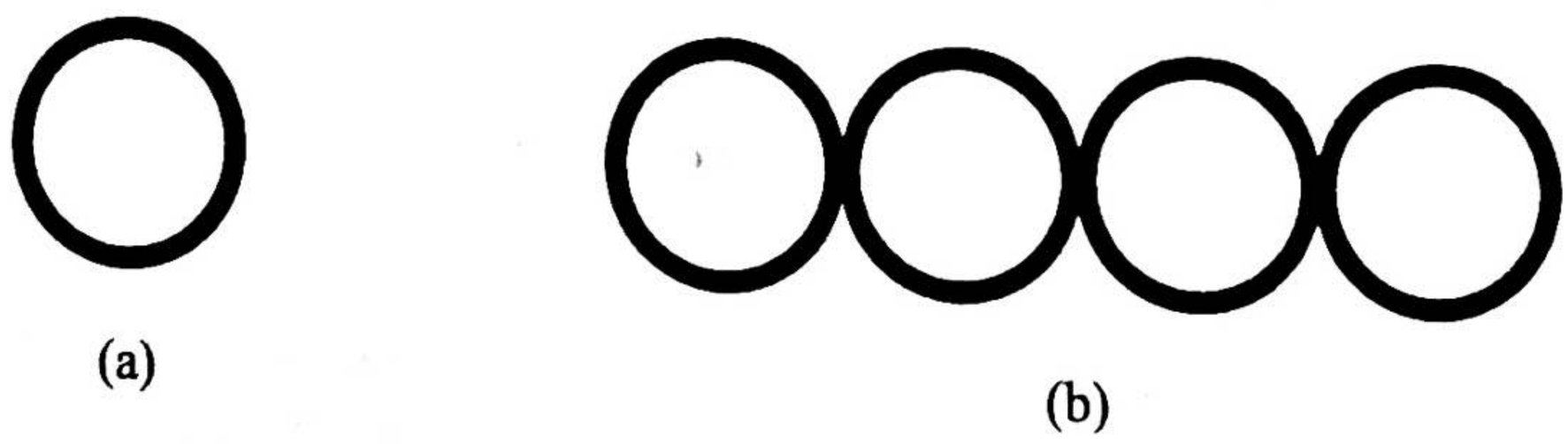


Figure 5.30 (a) Texture primitive. (b) Pattern generated by the rule $S \rightarrow aS$

(iii) **Spectral approaches:** Spectral techniques are based on properties of the Fourier spectrum and are used primarily to detect global periodicity in an image by identifying high energy, narrow peaks in the spectrum. Interpretation of spectrum features just mentioned often are simplified by expressing the spectrum in polar coordinates to yield a function $S(r,\theta)$, where S is the spectrum function and r and θ are the variables in this coordinate system. Analyzing $S_\theta(r)$ for a fixed value of θ yield the behaviour of the spectrum along a radial direction from the origin, whereas analyzing $S_r(\underline{\theta})$ for a fixed value of r yields the behaviour along a circle centered on the origin.

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \qquad \text{...(5.44)}$$

$$S(\theta) = \sum_{r=1}^{R_0} Sr(\theta) \qquad \text{...(5.45)}$$

Discriminating the different texture patterns by analyzing their corresponding $S(\theta)$ and $S(r)$ waveforms would be straightforward.

(d) **Use of Principal Components for Description:** The material discussed is applicable to boundaries and regions. Suppose that we are given the n component images of such a color. The n images can be treated as a unit by expressing each group of n corresponding pixels as a vector. These n elements con be expressed in the form of a 3-D column vector, x where

$$x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix} \qquad \text{...(5.46)}$$

We discuss the mean vector and covariance matrix

$$m_x = E\{x\} \qquad \text{...(5.47)}$$

$$C_x = E\{(x\text{-}m_x)(x\text{-}m_x)^T\} \qquad \text{...(5.48)}$$

The diagonal of C_x is the variance of vector. And we can find the correlation of any tow elements. Let A be a matrix whose rows are formed form the eigenvectors of C_x, ordered so that the last row is the eigenvector corresponding to the smallest eigenvalue.

Suppose that we use A as a transformation matrix to map the x into vectors denoted by y, as follows

$$y = A(x\text{-}m_x) \qquad \text{...(5.49)}$$

This expression is called the Hotelling transform. The covariance matrix of y is

$$C_y = AC_xA^T = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \cdot & \\ & & & \cdot \\ 0 & & & \lambda_n \end{bmatrix} \quad \ldots(5.50)$$

The off-diagonal elements of this covariance matrix are 0, so the elements of the y vectors are uncorrelated.

The concept of aligning a 2-D object with its principal eigenvectors plays an important role in description. As noted earlier, description should be as independent as possible to variations in size, translation, and rotation. The ability to align the object with its principal axes provides a reliable means for removing the effects of rotation. The eigen values are the variances along the eigen axes, and can be used for size normalization. The effects of translation are account mind the fact that the method of description derived in this section is equally applicable to both regions and boundaries.

(e) **Relational Descriptors:** Relational descriptors apply equally well to boundaries or regions, and their main purpose is to capture in the form of rewriting rules basic repetitive patterns in a boundary or region. By defining the two primitive elements a and b, we use it to establish the code, and use rewriting rules to transform it.

$S \rightarrow aA$

$A \rightarrow bS$

$A \rightarrow b$

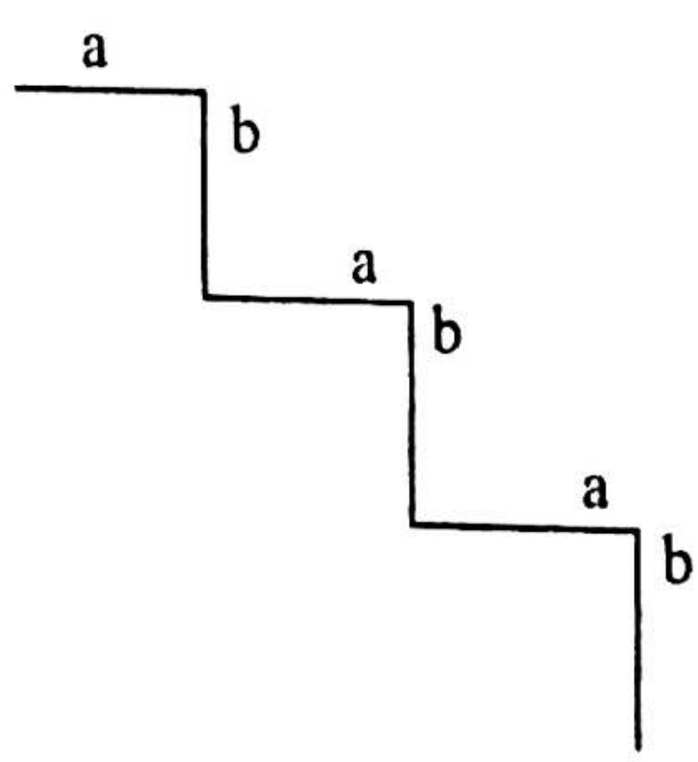


Figure 5.31 (1,2,1,2,1,3) is coded structure

Because strings are 1-D structures, their application to image description requires establishing an appropriate method for reducing 2-D positional relations to 1-D form.

An approach is to follow the contour of an object and code the result with segments of specified direction and/or length. Another approach is to describe sections of an image by directed line segments. String descriptions are best suited for applications in which connectivity of primitives can be expressed in a head-tail or other continuous manner. Sometimes regions that are similar in terms of texture or other descriptor may not be contiguous, and techniques are required for describing such situations. An approach is use tree descriptors.

5.15 Basic Idea of Region Growing

Suppose that we start with a single pixel p and wish to expand from that seed pixel to fill a coherent region. Let's define a similarity measure $S(i, j)$ such that it produces a high result if pixels i and j are similar and a low one otherwise. First, consider a pixel q adjacent to pixel p. We can add pixel q to pixel p's region if $S(p, q) > T$ for some threshold T. We can then proceed to the other neighbours of p and do likewise.

Suppose that $S(p, q) > T$ and we added pixel q to pixel p's region. We can now similarly consider the neighbours of q and add them likewise if they are similar enough. If we continue this recursively, we have an algorithm analogous to a "flood fill" but which works not on binary data but on similar gray scale data.

5.15.1 A Few Common Approaches

(a) **Similarity Measures:** One obvious similarity measure is to compare individual pixel intensities. This can be sensitive to noise, though. We can reduce our sensitivity to noise by comparing neighbourhood characteristics between pixels. For example, we could compare the average intensities over a neighbourhood around each pixel. Notice, though, that this is simply the same as applying an averaging kernel through convolution and then doing single-pixel comparison.

One can also gather other features from the neighbourhood, such as texture (which we'll cover in a later lecture), gradient, or geometric properties.

(b) **Comparing to Original Seed Pixel:** One approach is to always compare back to the seed point p by using $S(p, r)$ when considering adding pixel r to the growing region. This the advantage of using a single basis for comparison across all pixels in the region. However, it means that the region produced is very sensitive to the choice of seed pixel. Does it make sense that the region produced by growing pixel p is different than that produced by its neighbour q also in the same region?

(c) **Comparing to Neighbor in Region:** One way of removing this effect is to compare pixel r to the neighboring pixel q already in the region of p: $S(q, r)$. In this way, each pixel that is already in the region can bring in neighbors who are like it. The advantage of this method is that it produces transitive closures of similarity. If p is similar to q, and if q is similar to r, p and r end up in the same region. Of course, this method can cause significant drift as one grows farther away from the original seed pixel. Indeed, the original seed is of no significance once one grows out more than one pixel. What started out finding green pixels ends up adding yellow ones if the transition is gradual enough.

(d) **Comparing to Region Statistics:** A third approach is to compare candidate pixel r to the entire region already collected. Initially, this region consists of pixel p alone, so pixel p dominates. As the region grows, aggregate statistics are collected, and any new pixel r which may be added to the region is compared not to pixel p or to its already-included neighbor q, but to these aggregate statistics

One simple such statistic is to keep an updated mean of the region pixels. As each new pixel is added, this mean is updated. Although gradual drift is still possible, the weight of all previous pixels in the region act as a damper on such drift. Some texts refers to this as centroid region growing.

(e) **Multiple Seeds:** Another approach is to initialize the region with not only a single pixel but a small set of pixels to better describe the region statistics. With such initialization, not only a region mean is suggested but the variance as well. Candidate points can be compared to the region mean with respect to the region variance. This gives us at least some hope of producing identical regions under varying noise conditions.

Multiple seeds can be selected either through user interaction or by sampling a small area around the initial seed point. In other words, if we assume that no desired region is indeed smaller than, say, 5 pixels across, we can sample our initial statistics from a 5 × 5 area around the seed and grow outwards from there.

(f) **Cumulative Differences:** Another approach is to use a similarity measure that involves not only comparison to a single pixel or to a region statistic, but by calculating cumulative differences as one follows a path from the seed to the candidate point.

In other words, if point q is a neighbor or seed p, and candidate point r is a neighbor of q, instead of using $S(p, r)$ or $S(q, r)$, we can use $S(p, q) + S(q, r)$. This is equivalent to finding the minimum-cost path from p to r and using this as the basis for the addition or rejection of point r.

5.15.2 Region Merging

The limit of the multiple-seed approach is to let every pixel be a seed. If two adjacent pixels are similar, merge them into a single region. If two adjacent regions are collectively similar enough, merge them likewise. This collective similarity is usually based on comparing the statistics of each region. Eventually, this method will converge when no further such merging are possible.

(a) **Split and Merge Algorithm:** Pure merging methods are, however, computationally expensive because they start from such small initial regions (individual points). We can make this more efficient by recursively splitting the image into smaller and smaller regions until all individual regions are coherent, then recursively merging these to produce larger coherent regions.

First, we must split the image. Start by considering the entire image as one region.

(i) If the entire region is coherent (i.e., if all pixels in the region have sufficient similarity), leave it unmodified.

(ii) If the region is not sufficiently coherent, split it into four quadrants and recursively apply these steps to each new region.

The "splitting" phase basically builds a quad-tree. Notice that each of the regions (squares) so produced is now coherent. However, several adjacent squares of varying sizes might have similar characteristics.

We can thus merge these squares into larger regions from the bottom up, much as we merged regions earlier. Since we are starting with regions (hopefully) larger than single pixels, this method is more efficient.

REVIEW QUESTIONS

Q.1 What is segmentation ? Write about point, Line and Edge detection ?

Q.2 Explain gradient operator.

Q.3 What si Edge linking ? Define local and global processing.

Q.4 Explain fast scanning algorithm.

Q.5 How to improve fast scanning algorithm by using Morphological?

Q.6 Explain representation.

Q.7 Write about descriptor.

○ ○ ○