# DIP-Notes-17EC72 - Lecture notes 1-5

Digital Image Processing (Visvesvaraya Technological University)

||Jai Sri Gurudev||

Sri Adichunchanagiri Shikshana Trust(R)

# SJB INSTITUTE OF TECHNOLOGY

**BGS HEALTH AND EDUCATION CITY**

Kengeri, Bengaluru-560060

**Department of Electronics and Communication Engineering**


B.E., VII Semester, Electronics & Communication Engineering

[As per Choice Based Credit System (CBCS) scheme]

# Digital Image Processing
## (17EC72)

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION**

**SJB INSTITUTE OF TECHNOLOGY**

**BENGALURU -560060**

**Faculty incharge: Dr. Supreeth H S G, Associate Professor, ECE, SJBIT**

# DIGITAL IMAGE PROCESSING

**B.E., VII Semester, Electronics & Communication Engineering**

[As per Choice Based Credit System (CBCS) scheme]

| | | | |
|---|---|---|---|
| **Subject Code** | **17EC72** | **IA Marks** | **40** |
| **Number of Lecture Hours/Week** | **04** | **Exam Marks** | **60** |
| **Total Number of Lecture Hours** | **50 (10 Hours / Module)** | **Exam Hours** | **03** |
| CREDITS – 04 | | | |

**Course Objectives:** The objectives of this course are to:

- Understand the fundamentals of digital image processing.
- Understand the image transform used in digital image processing.
- Understand the image enhancement techniques used in digital image processing.
- Understand the image restoration techniques and methods used in digital image processing.
- Understand the Morphological Operations and Segmentation used in digital image processing.

| **Module – 1** | **RBT Level** |
|---|---|
| **Digital Image Fundamentals**: What is Digital Image Processing?, Origins of Digital Image Processing, Examples of fields that use DIP, Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships Between Pixels, Linear and Nonlinear Operations. **[Text: Chapter 1 and Chapter 2: Sections 2.1 to 2.5, 2.6.2]** | L1, L2 |
| **Module – 2** | |
| **Spatial Domain:** Some Basic Intensity Transformation Functions, Histogram Processing, Fundamentals of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters<br>**Frequency Domain**: Preliminary Concepts, The Discrete Fourier Transform (DFT) of Two Variables, Properties of the 2-D DFT, Filtering in the Frequency Domain, Image Smoothing and Image Sharpening Using Frequency Domain Filters, Selective Filtering. **[Text: Chapter 3: Sections 3.2 to 3.6 and Chapter 4: Sections 4.2, 4.5 to 4.10]** | L1, L2, L3 |
| **Module – 3** | |
| **Restoration:** Noise models, Restoration in the Presence of Noise Only using Spatial Filtering and Frequency Domain Filtering, Linear, Position- Invariant Degradations, Estimating the Degradation Function, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Constrained Least Squares Filtering. | L1, L2, L3 |

| | |
|---|---|
| **[Text: Chapter 5: Sections 5.2, to 5.9]** | |
| **Module – 4** | |
| **Color Image Processing:** Color Fundamentals, Color Models, Pseudocolor Image Processing.<br>**Wavelets:** Background, Multi-resolution Expansions.<br>**Morphological Image Processing:** Preliminaries, Erosion and Dilation, Opening and Closing, The Hit-or-Miss Transforms, Some Basic Morphological Algorithms.<br>**[Text: Chapter 6: Sections 6.1 to 6.3, Chapter 7: Sections 7.1 and 7.2, Chapter 9: Sections 9.1 to 9.5]** | L1, L2, L3 |
| **Module – 5** | |
| **Segmentation**: Point, Line, and Edge Detection, Thresholding, Region-Based Segmentation, Segmentation Using Morphological Watersheds.<br>**Representation and Description:** Representation, Boundary descriptors.<br>**[Text: Chapter 10: Sections 10.2, to 10.5 and Chapter 11: Sections 11.1 and 11.2]** | L1, L2, L3 |

**Course Outcomes:** At the end of the course students should be able to:

- Understand image formation and the role human visual system plays in perception of gray and color image data.
- Apply image processing techniques in both the spatial and frequency (Fourier) domains.
- Design image analysis techniques in the form of image segmentation and to evaluate the Methodologies for segmentation.
- Conduct independent study and analysis of Image Enhancement techniques.

**Question paper pattern:**

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of Three sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

**Text Book:**
**Digital Image Processing**- Rafel C Gonzalez and Richard E. Woods, PHI 3rd Edition 2010.

**Reference Books:**
1. **Digital Image Processing**- S.Jayaraman, S.Esakkirajan, T.Veerakumar, Tata McGraw Hill 2014.
2. **Fundamentals of Digital Image Processing**-A. K. Jain, Pearson 2004.

| Module – 1 | RBT Level |
|---|---|
| **Digital Image Fundamentals**: What is Digital Image Processing?, Origins of Digital Image Processing, Examples of fields that use DIP, Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships Between Pixels, Linear and Nonlinear Operations.<br>**[Text: Digital Image Processing**- Rafel C Gonzalez and Richard E. Woods<br>**Chapter 1 and Chapter 2: Sections 2.1 to 2.5, 2.6.2]** | L1, L2 |

### 1.1 What is Digital Image Processing?

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

When $x$, $y$, and $f$ are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer.

A digital image is composed of a finite number of elements, each of which has a location and value. These elements are called pixels.

Unlike humans, who are limited to the visual band of electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves.

There is no general agreement regarding where image processing stops and other related areas, such as image analysis and computer vision, starts.

Although there are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other, one useful paradigm is to consider three types of processes in this continuum:

A low-level process is characterized by the fact that both its inputs and outputs are images.

A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images.

The higher-level processes include object recognition, image analysis, and performing the cognitive functions associated with vision.

### 1.2 Origins of Digital Image Processing

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

Introduction of the Bartlane cable picture transmission system in the early 1920s reduced the time to transport a picture across the Atlantic from more than one week to less than three hours. Examples of fields that use DIP

**FIGURE 1.1** A digital picture produced in 1921 from a coded tape by a telegraph printer with special type faces. (McFarlane.[†])

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

The printing method used to obtain Figure 1.1 was abandoned toward the end of 1921 in favour of a technique based on photographic reproduction made from tapes perforated at the telegraph receiving terminal.

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

**FIGURE 1.2** A digital picture made in 1922 from a tape punched after the signals had crossed the Atlantic twice. (McFarlane.)

**FIGURE 1.3** Unretouched cable picture of Generals Pershing and Foch, transmitted in 1929 from London to New York by 15-tone equipment. (McFarlane.)

The history of digital image processing is tied to the development of the digital computer. The first computers powerful enough to carry out meaningful image processing tasks appeared in the early 1960s.

**FIGURE 1.4** The first picture of the moon by a U.S. spacecraft. *Ranger 7* took this image on July 31, 1964 at 9:09 A.M. EDT, about 17 minutes before impacting the lunar surface. (Courtesy of NASA.)

In parallel with space applications, digital image processing techniques were used in medical imaging, remote Earth resources observations, and astronomy in the late 1960s and early 1970s.

The invention in the early 1970s of computerized axial tomography (CAT), also called computerized tomography (CT), is one of the most important events in the application of image processing in medical diagnosis.

The field of image processing has grown vigorously since 1960s, and the image processing techniques now are used in a broad range of applications.

Other than the processing intended for human interpretation, another important area of applications of digital image processing is in solving problems dealing with machine perception.

Typical problems in machine perception that routinely utilize image processing techniques are automatic character recognition, industrial machine vision, military recognizance, processing of fingerprints, and many other tasks.

The continuing decline in the ratio of computer price to performance and the expansion of networking and communication bandwidth via World Wide Web and the Internet have created unprecedented opportunities for continued growth of digital image processing.

### 1.3 Examples of fields that use DIP

One of the simplest ways to develop a basic understanding of the extent of image processing applications is to categorize images according to their source. The principal energy source for images in use today is the electromagnetic (EM) energy spectrum. Electromagnetic waves can be conceptualized as propagating sinusoidal waves of varying wavelengths, or they can be though of as a stream of massless particles traveling in a wavelike pattern and moving at the speed of light. Each massless particle contains a certain amount (or bundle) of energy.

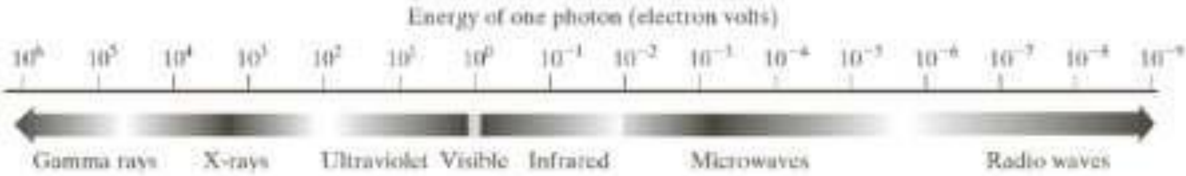
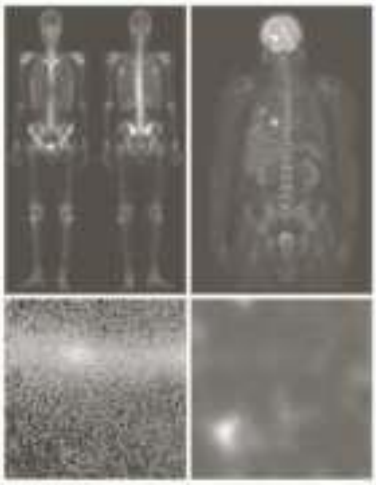



**FIGURE 1.5** The electromagnetic spectrum arranged according to energy per photon.



a b
c d

**FIGURE 1.6** Examples of gamma-ray imaging. (a) Bone scan. (b) PET image. (c) Cygnus Loop. (d) Gamma radiation (bright spot) from a reactor valve. (Images courtesy of (a) G.E. Medical Systems, (b) Dr. Michael E. Casey, CTI PET Systems, (c) NASA, (d) Professors Zhong He and David K. Wehe, University of Michigan.)

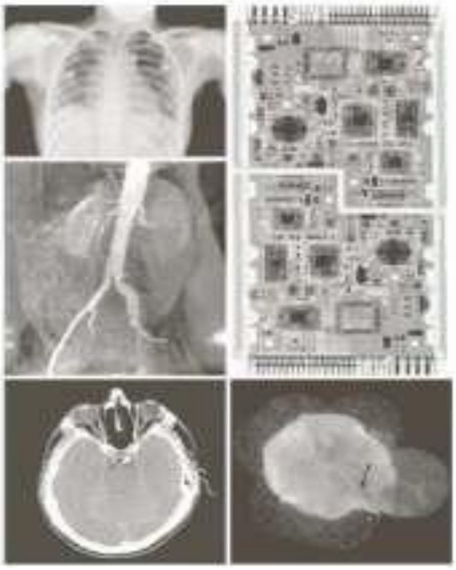
Gamma-Ray Imaging

Major uses of imaging based on gamma rays include nuclear medicine and astronomical observations. Images are produced from emissions collected by gamma ray detectors.

X-Ray Imaging

X-rays are among the oldest sources of EM radiation used for imaging.

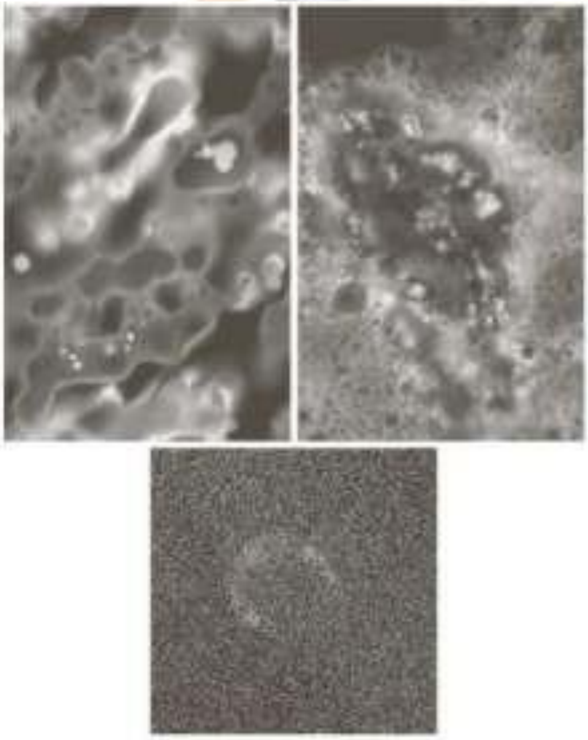
Figure 1.7 shows some examples of X-ray imaging.



## Imaging in the Ultraviolet Band

Applications of ultraviolet ―light‖ are varied. They include lithography, industrial inspection, microscopy, lasers, biological imaging, and astronomical observations.

Figure 1.8 shows some examples of ultraviolet imaging.



a b
c

**FIGURE 1.8** Examples of ultraviolet imaging. (a) Normal corn. (b) Smut corn. (c) Cygnus Loop. (Images courtesy of (a) and (b) Dr. Michael W. Davidson, Florida State University, (c) NASA.)

## Imaging in the Visible and Infrared Bands

The infrared band often is used in conjunction with visual imaging. Figure 1.9 shows some examples of images obtained with a light microscope.

a b c
d e f

**FIGURE 1.9** Examples of light microscopy images. (a) Taxol (anticancer agent), magnified 250×. (b) Cholesterol—40×. (c) Microprocessor—60×. (d) Nickel oxide thin film—600×. (e) Surface of audio CD—1750×. (f) Organic superconductor—450×. (Images courtesy of Dr. Michael W. Davidson, Florida State University.)

Another major area of visual processing is remote sensing, which includes several bands in the visual and infrared regions of the spectrum. Table 1.1 shows the so-called thematic bands in NASA‘s LANDSAT satellite.

The primary function of LANDSAT is to obtain and transmit images of the Earth from space for purposes of monitoring environmental conditions of the planet.

| Band No. | Name | Wavelength (μm) | Characteristics and Uses |
|---|---|---|---|
| 1 | Visible blue | 0.45–0.52 | Maximum water penetration |
| 2 | Visible green | 0.52–0.60 | Good for measuring plant vigor |
| 3 | Visible red | 0.63–0.69 | Vegetation discrimination |
| 4 | Near infrared | 0.76–0.90 | Biomass and shoreline mapping |
| 5 | Middle infrared | 1.55–1.75 | Moisture content of soil and vegetation |
| 6 | Thermal infrared | 10.4–12.5 | Soil moisture; thermal mapping |
| 7 | Middle infrared | 2.08–2.35 | Mineral mapping |

**TABLE 1.1** Thematic bands in NASA's LANDSAT satellite.

Figure 1.10 shows one image for each of the spectrum bands in Table 1.1.

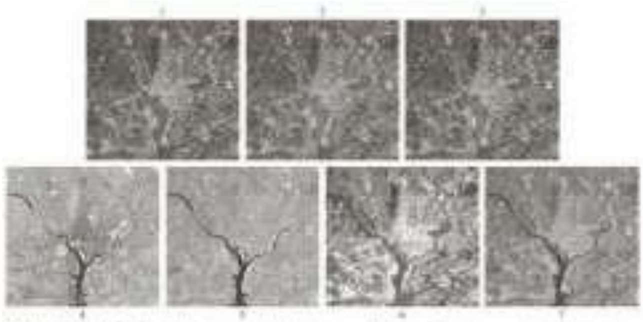


**FIGURE 1.10** LANDSAT satellite images of the Washington, D.C. area. The numbers refer to the thematic bands in Table 1.1. (Images courtesy of NASA.)

Weather observation and prediction also are major applications of multi-spectrum imaging from satellites.

Figure 1.11 is an image of Hurricane Katrina.

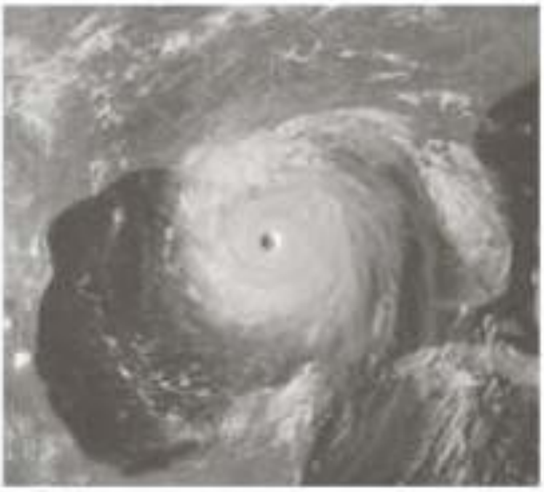


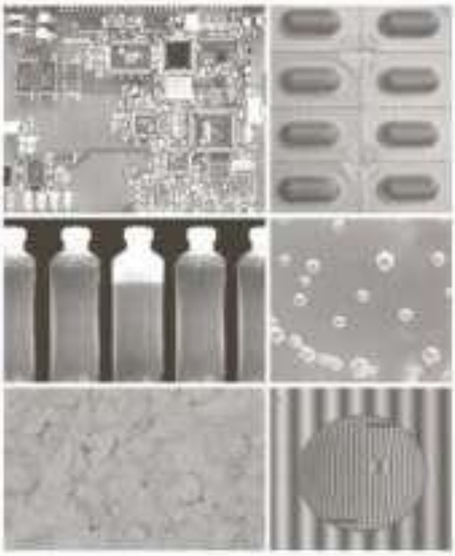
**FIGURE 1.11** Satellite image of Hurricane Katrina taken on August 29, 2005. (Courtesy of NOAA.)

Figure 1.12 and Figure 1.13 show an application of infrared imaging. These images are part of the Nighttime Lights World data set, which provides a global inventory of human settlements.

**FIGURE 1.12** Infrared satellite images of the Americas. The small gray map is provided for reference. (Courtesy of NOAA.)

**FIGURE 1.13** Infrared satellite images of the remaining populated part of the world. The small gray map is provided for reference. (Courtesy of NOAA.)

A major area of imaging in the visual spectrum is in an automated visual inspection of manufactured goods.



a b
c d
e f

**FIGURE 1.14** Some examples of manufactured goods often checked using digital image processing. (a) A circuit board controller. (b) Packaged pills. (c) Bottles. (d) Air bubbles in a clear-plastic product. (e) Cereal. (f) Image of intraocular implant. (Fig. (f) courtesy of Mr. Pete Sites, Perceptics Corporation.)


a b
c
d

**FIGURE 1.15** Some additional examples of imaging in the visual spectrum. (a) Thumb print. (b) Paper currency. (c) and (d) Automated license plate reading. (Figure (a) courtesy of the National Institute of Standards and Technology. Figures (c) and (d) courtesy of Dr. Juan Herrera, Perceptics Corporation.)

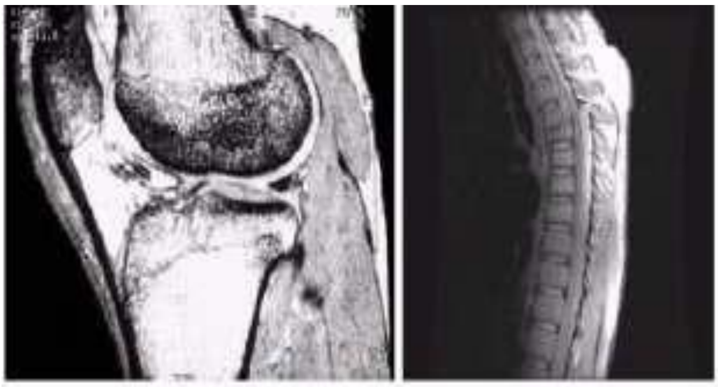
## Imaging in the Microwave Band

The dominant application of imaging in the microwave band is radar. The unique feature of imaging radar is its ability to collect data over virtually any region at any time, regardless of weather or ambient lighting conditions.

**FIGURE 1.16** Spaceborne radar image of mountains in southeast Tibet. (Courtesy of NASA.)

## Imaging in the Radio Band

The major applications of imaging in the radio band are in medicine and astronomy. In medicine, radio waves are used in magnetic resonance imaging (MRI).



a b

**FIGURE 1.17** MRI images of a human (a) knee, and (b) spine. (Image (a) courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School, and (b) Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

## Examples in which Other Imaging Modalities Are Used

Although imaging in the EM spectrum is dominant by far, there are a number of other imaging modalities that also are important. Imaging using –sound‖ finds application in geological exploration, industry, and medicine.

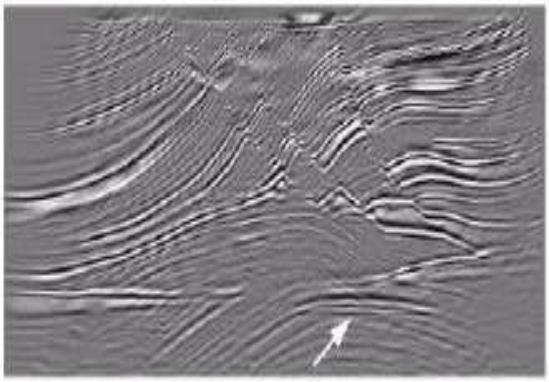


FIGURE 1.19 Cross-sectional image of a seismic model. The arrow points to a hydrocarbon (oil and/or gas) trap. (Courtesy of Dr. Curtis Ober, Sandia National Laboratories.)

In Figure 1.19, the arrow points to a hydrocarbon (oil and/or gas) trap. This target is brighter than the surrounding layers because the change in the target region is larger.

The best known ultrasound imaging applications are in medicine, especially in obstetrics, where unborn babies are imaged to determine the health of their development.

### 1.4 Fundamental Steps in Digital Image Processing

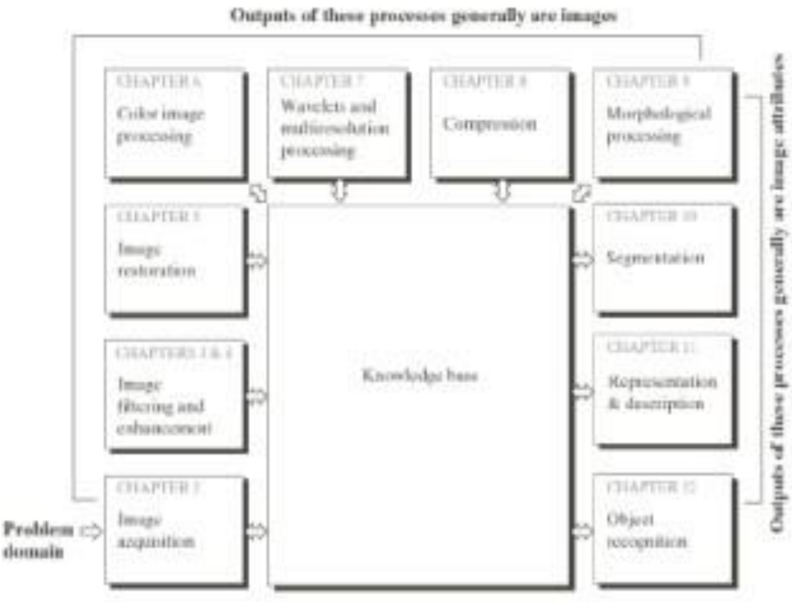



FIGURE 1.23 Fundamental steps in digital image processing. The chapter(s) indicated in the boxes is where the material described in the box is discussed.

**Image acquisition** is the first process shown in Fig., Note that acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling.

**Image enhancement** is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. A familiar example of enhancement is when we increase the contrast of an image because –it looks better.‖ It is important to keep in mind that enhancement is a very subjective area of image processing. Image restoration is an area that also deals with improving the appearance of an image.

However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation. Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a –good‖ enhancement result.

**Color image processing** is an area that has been gaining in importance because of the significant increase in the use of digital images over the Internet.

**Wavelets** are the foundation for representing images in various degrees of resolution. Compression, as the name implies, deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar (perhaps inadvertently) to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

**Morphological processing** deals with tools for extracting image components that are useful in the representation and description of shape.

**Segmentation** procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

**Representation and description** almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. **Boundary representation** is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

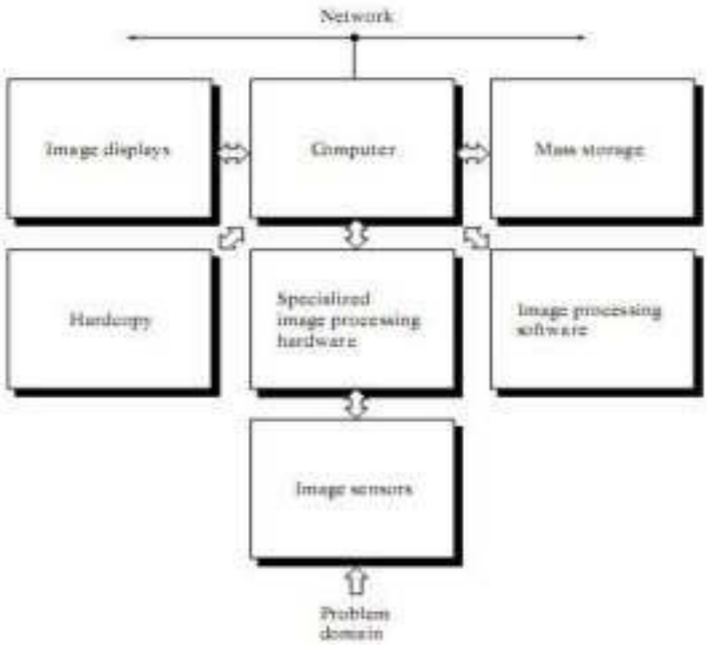
**Recognition** is the process that assigns a label (e.g., –vehicle‖) to an object based on its descriptors. We conclude our coverage of digital image processing with the development of methods for recognition of individual objects.

## 1.5 Components of an Image Processing System

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 3 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.




**Specialized image processing hardware** usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 framess) that the typical main computer cannot handle.

**The computer** in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, some times specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

**Software** for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

**Mass storage** capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications falls into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast re-call, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and Tbytes (meaning tera, or one trillion, bytes). One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers, that store one or more images and can be accessed rapidly, usually at video rates (e.g., at 30 complete images per second).The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig.3.Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in –jukeboxes‖ are the usual media for archival applications.

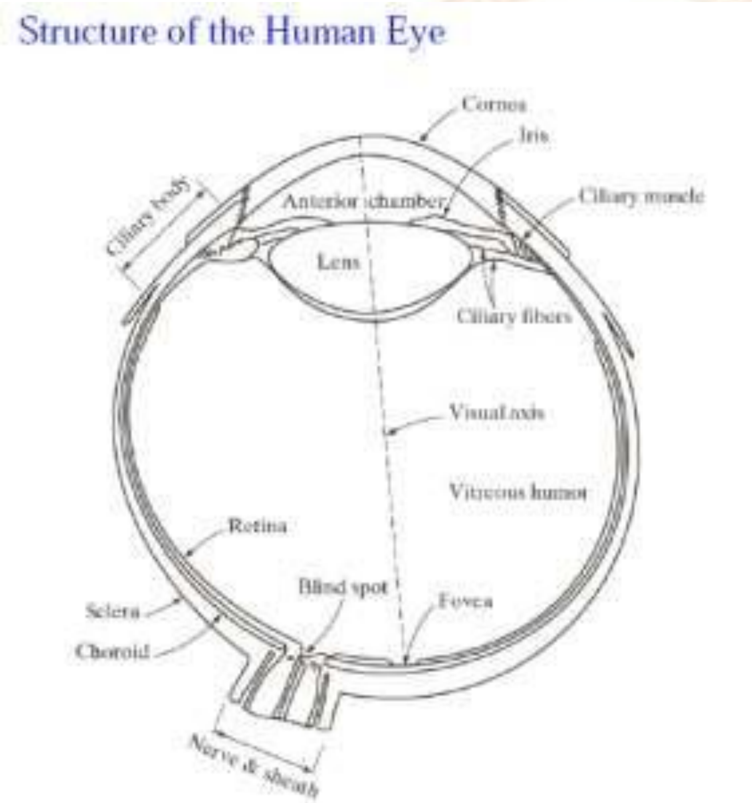
**Image displays** in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

**Hardcopy devices** for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

**Networking** is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

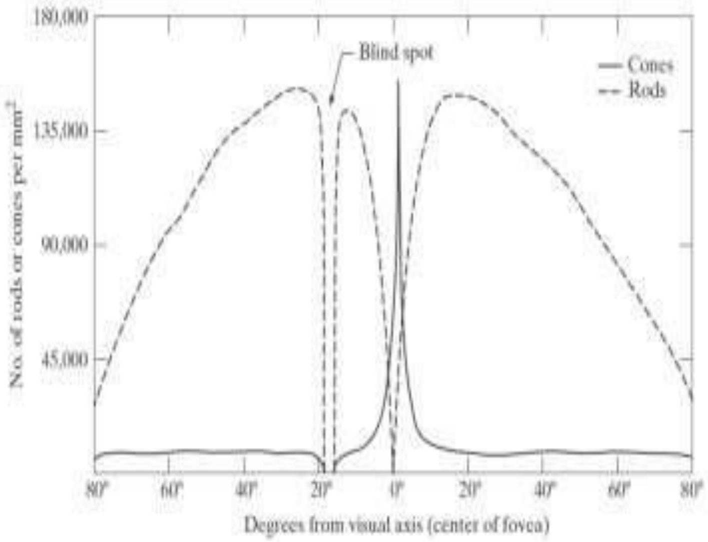
## 1.6 Elements of Visual Perception

Although the field of digital image processing is built on a foundation of mathematical formulations, human intuition and analysis play a central role in the choice of one technique versus another, which often is made based on subjective and visual judgments. We focus our interest in the mechanics and parameters related to how images are formed and perceived by humans.




- Eye is nearly a sphere.
- 20 mm diameter.
- Three membranes enclosed:
  - Cornea & Sclera
  - Choroid
  - Retina
- Cornea – tough transparent tissue.
- Sclera – continuation with cornea opaque membrane that encloses remainder of optic globe.
- Choroid – contains network of blood vessels – serves as major source of nutrition to eye. Divided into
  - Ciliary body
  - Iris (contracts or expands to control amount of light enters the eye)
  - Lens – made up of concentric layers of fibrous cells and suspended by fibres
- Lens absorbs about 8% of the visible light spectrum (higher at shorter wavelengths)
- Retina is the inner most membrane, which covers the posterior surface
- When the eye is properly focused, light from an object is imaged on the **retina**

- There is a distribution of discrete light receptors over retina surface. 2 types: **cones and rods**
- Cones (6-7 million) are mainly around the central part called fovea and sensitive to color
- Rods (75-150 million) are distributed wider and are sensitive to low illumination levels




- We perceive fine detail with cones (called **photopic** or bright-light vision)
- Rods give a general, overall picture of the entire field of view (called **scotopic** or dim-light vision)
- Fovea is a circular indentation of about 1.5mm in diameter
- But, considering it as a square sensor array of 1.5mm x 1.5mm will be more useful for us
- medium resolution CCD imaging chip can have same number of elements in a receptor array no larger than 5mm x 5mm

**Note:** Such a comparison is superficial but, human eye's ability to resolve detail is comparable to current electronic imaging devices.

## Image of Formation in the Eye

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

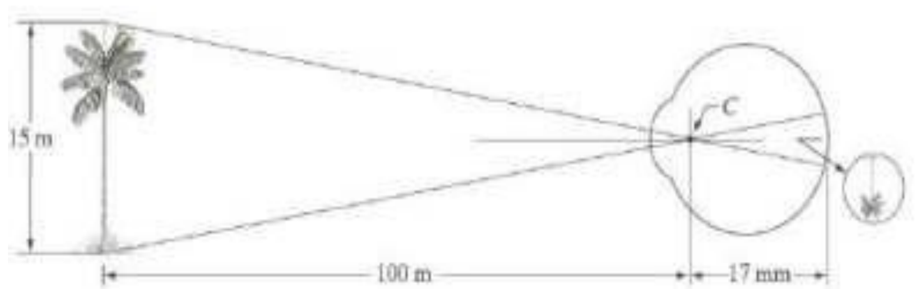



**FIGURE 2.3** Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.

In an ordinary photographic camera, the converse is true. The lens has a fixed focal length, and focusing at various distances is achieved by varying the distance between the lens and the imaging plane.

Interesting Fact!!!

How to obtain dimension of the image formed on the retina?

Sol:

w.r.t above example:
Let ***h*** denote height of the object in retinal image.
Based on the geometry of above fig.,
15/100=h/17
Therefore, h=2.55mm

## Brightness Adaptation and Discrimination

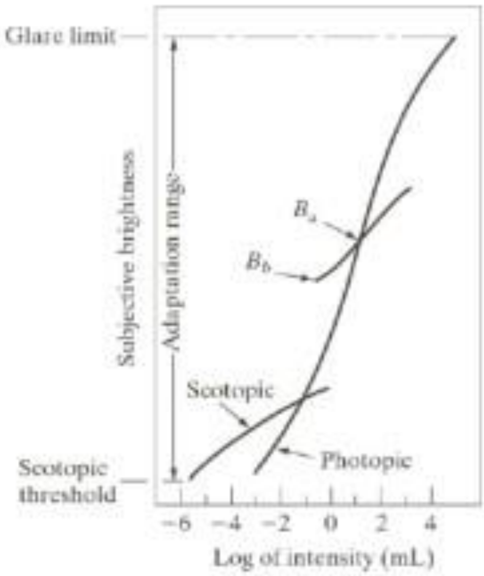



**FIGURE 2.4** Range of subjective brightness sensations showing a particular adaptation level.

Since digital images are displayed as a discrete set of intensities, the eye's ability to discriminate between different intensity levels is an important issue.

The range of light intensity levels adapted by human visual system is enormous, on the order of $10^{10}$, from the scotopic threshold to the glare limit.

Experimental evidence indicates that subjective brightness is a logarithmic function of the light intensity incident on the eye.

The total range of distinct intensity levels the eye can discriminate simultaneously is rather small when compared with the total adaptation range.

For any given set of conditions, the current sensitivity level of the visual system is called the brightness adaptation level, for example, as $B_a$ shown in Figure 2.4. It represents the range of subjective brightness that the eye can perceive when adapted to this level.

Another important issue is the ability of the eye to discriminate between changes in light intensity at any specific adaptation level.

Figure 2.5 shows the idea of a basic experiment to determine the human visual system for brightness discrimination.

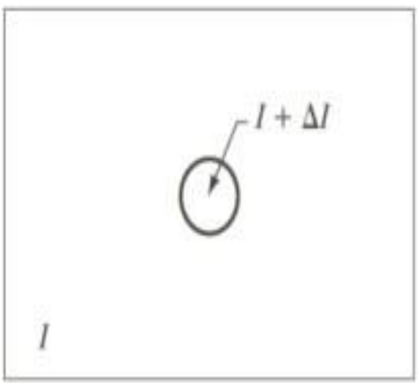



**FIGURE 2.5** Basic experimental setup used to characterize brightness discrimination.

An increment of illumination, $\Delta I$ , is added to the field in the form of a short-duration flash. If $\Delta I$ is not bright enough, the subject says ―no‖. As_$I$ gets stronger, the subject may give a positive response of ―yes‖ Finally, when $\Delta I$ becomes strong enough, the subject will give a positive response of ―yes‖ all the time. Let $\Delta I_c$ denote the increment of illumination discriminable 50% of the time with background illumination $I$ . It is called the Weber ratio.

A small value of $\Delta I_c / I$ means that a small change in intensity is discriminable. It represents –good‖ brightness discrimination. A plot of $\log \Delta I_c / I$ as a function of $\log I$ has the general shape shown in Figure 2.6.

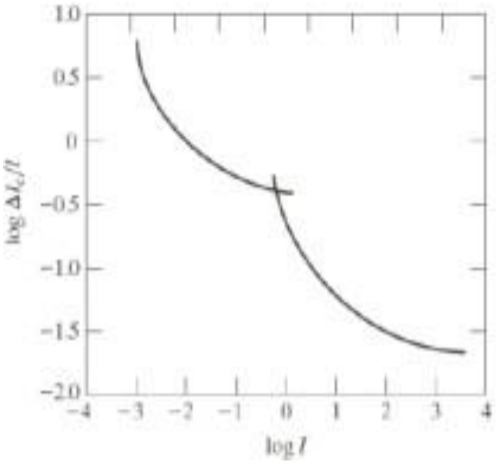
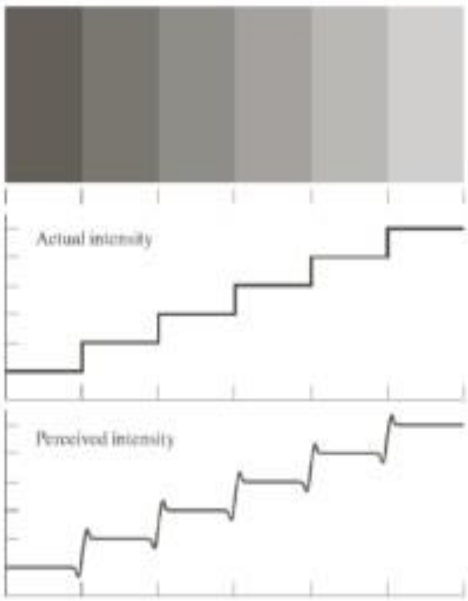



**FIGURE 2.6** Typical Weber ratio as a function of intensity.

Figure 2.6 shows that brightness discrimination improves significantly as background illumination increases. Two phenomena demonstrate that perceived brightness is not a simple function of intensity.




a
b
c

**FIGURE 2.7** Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.

Figure 2.7 (a) shows the fact that the visual system tends to undershoot or overshoot around the boundary of regions of different intensities.

As shown in Figure 2.7 (c), although the intensity of the stripes is constant, we actually perceive a brightness pattern that is strongly scalloped near the boundaries.

Another phenomenon, called simultaneous contrast, is related to the fact that a region‘s perceived brightness does not depend simply on its intensity.

a b c

**FIGURE 2.8** Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

In Figure 2.8, all the center squares have exactly the same intensity, though they appear to the eye to become darker as the background gets lighter.

Other examples of human perception phenomena are optical illusions, in which the eye fills in non-existing information or wrongly perceives geometrical properties of objects.

## Light and the Electromagnetic Spectrum

In 1666, Sir Isaac Newton discovered that when a beam of sunlight is passed through a glass prism, the emerging beam of light is consists of a continuous spectrum of colors from violet at one end to red at the other.

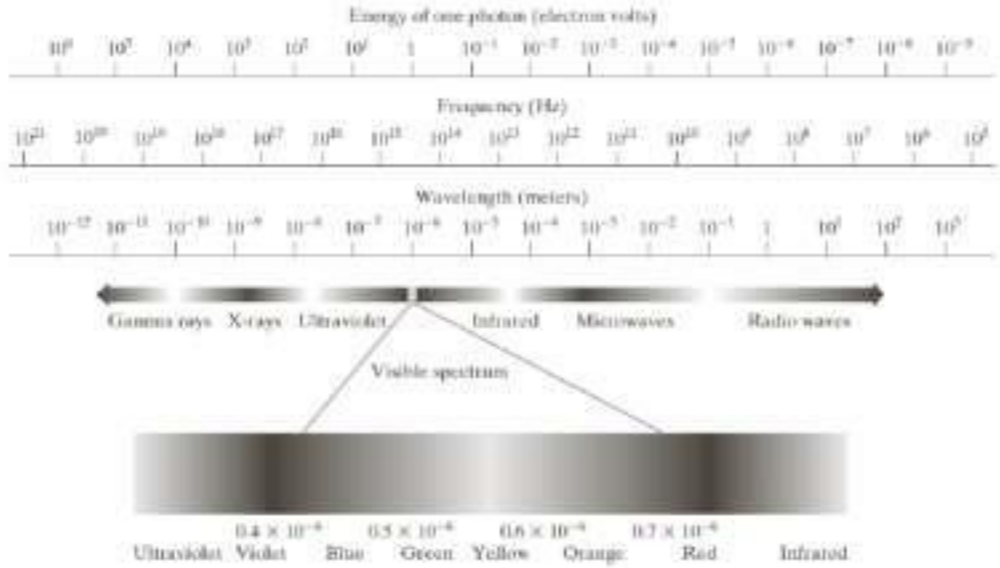



**FIGURE 2.10** The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

As Figure 2.10 shows that the range of colors we perceive in visible light represents a very small portion of the electromagnetic spectrum. Wavelength $\lambda$ and frequency $v$ are related by the expression .

$$\lambda = c/v$$

where $c$ is the speed of light ( $2.998\times10^{8} m/s$ ). The energy of the various components of the electromagnetic spectrum is given by $E = hv.$ where $h$ is Planck's constant.

Electromagnetic wave can be visualized as propagating sinusoidal waves with wavelength$\lambda$ , or they can be thought of as a stream of massless particles traveling in a wavelike pattern. Each massless particle contains a bundle of energy, which is called a photon.

Regarding to (2.2-2), energy is proportional to frequency, so the higher-frequency electromagnetic phenomena carry more energy per photon.

Light is a particular type of electromagnetic radiation that can be sensed by human eye. The visible band of the electromagnetic spectrum spans the range from about $0.43\ \mu m$ (violet) to about $0.79\ \mu m$ (red).

The colors that human perceive in an object are determined by nature of the light reflected from the object.

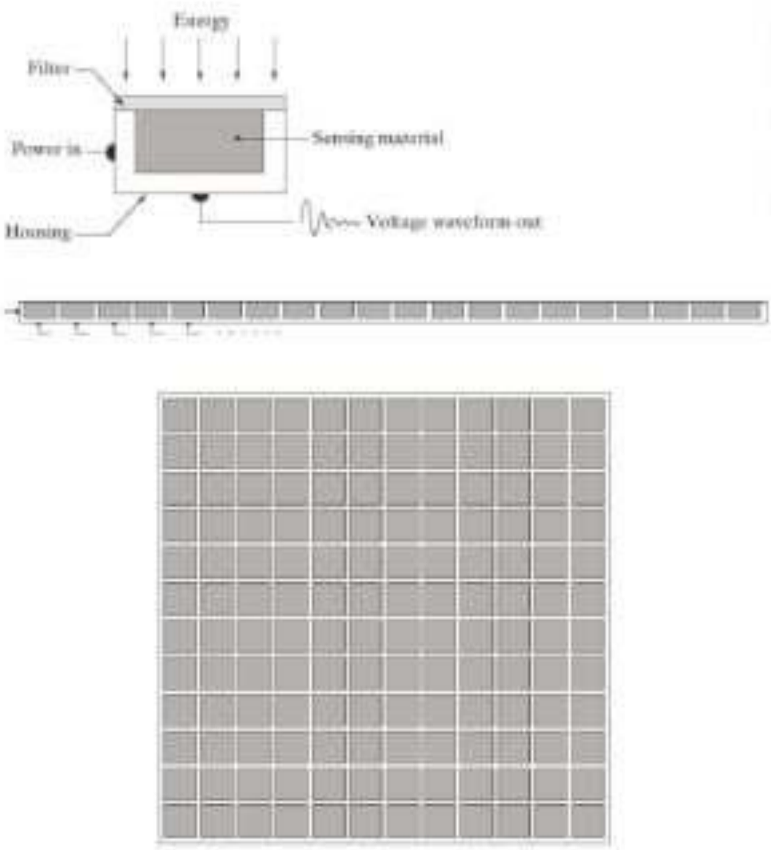
Light that is void of color is called monochromatic light. The only attribute of monochromatic light is its intensity or amount. Because the intensity is perceived to vary from black to white, the term gray level is used to denote monochromatic intensity.

The terms intensity and gray level are used interchangeably.

The range of measured values of monochromatic light from black to white is usually called the gray scale, and monochromatic images are frequently referred to as gray-scale images.

### 1.7 Image Sensing and Acquisition

Figure 2.12 shows the three principal sensor arrangements used to transform illumination energy into digital images.




a
b
c

**FIGURE 2.12**
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected.

The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response.

#### Image Acquisition Using a Single Sensor

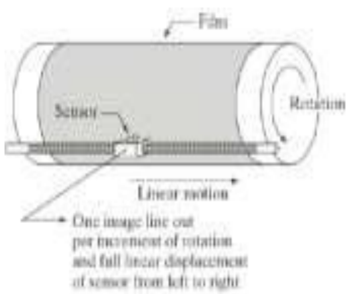



**FIGURE 2.13** Combining a single sensor with motion to generate a 2-D image.

In order to generate a *2-D* image using a single sensor, there has to be relative displacements in both the *x* - and *y*- directions between the sensor and the area to be imaged.

Figure 2.13 shows an arrangement used in high-precision scanning.

Other similar mechanical arrangements use a flat bed, with the sensor moving in two linear directions.

Another example of imaging with a

single sensor places a laser source coincident with the sensor. Moving mirrors are used to control the outgoing beam in a scanning pattern and to direct the reflected laser signal onto the sensor.

## Image Acquisition Using Sensor Strips

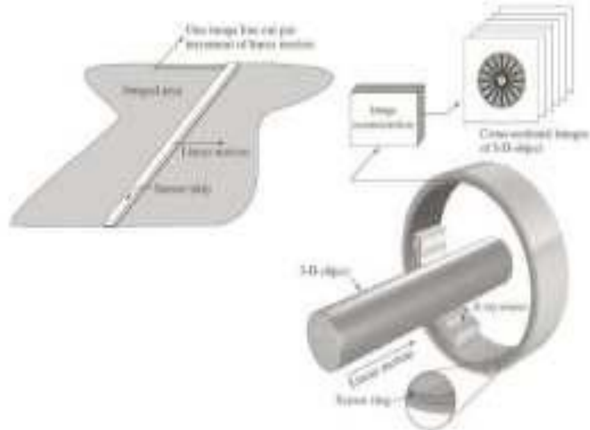


FIGURE 2.14 (a) Image acquisition using a linear sensor strip. (b) Image acquisition using a circular sensor strip.

As Figure 2.12 (b) shows, an in-line arrangement of sensors in the form of a sensor strip is used much more than a single sensor.

Figure 2.14 (a) shows the type of arrangement used in most flat bed scanners.

Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional images of 3-*D* objects, as Figure 2.14 (b) shows.

## Image Acquisition Using Sensor Arrays

Figure 2.12 (c) shows individual sensors arranged in the form of a 2-*D* array. This is the predominated arrangement found in digital cameras. Since the sensor array is two-dimensional, its key advantage is that a complete image can be obtained by focusing the energy pattern onto the surface of the array. The principal manner in which array sensors are used is shown in Figure 2.15.

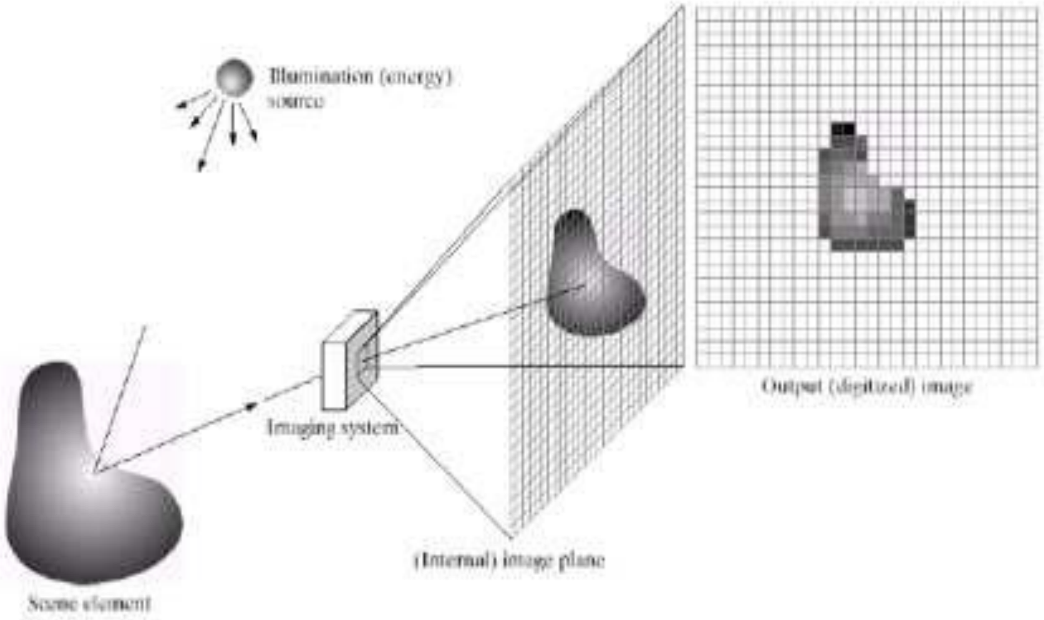



FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

### A Simple Image Formation Model

We denote images by two-dimensional functions of the form $f(x,y)$ . The value of $f$ at coordinates $(x,y)$ is a positive scalar quantity whose physical meaning is determined by the source of the image.

The function $f(x,y)$ must be nonzero and finite

$$0 < f(x,y) < \infty$$

The function $f(x,y)$ may be characterized by two components:

1. The amount of source illumination incident on the scene being viewed;
2. The amount of illumination reflected by the objects in the scene.

These are called illumination and reflectance components and are denoted by $i(x,y)$ and $r(x,y)$. These two functions combine to form $f(x,y)$ :

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

where

$$0 < i(x,y) < \infty$$

and

$$0 < r(x,y) < 1,$$

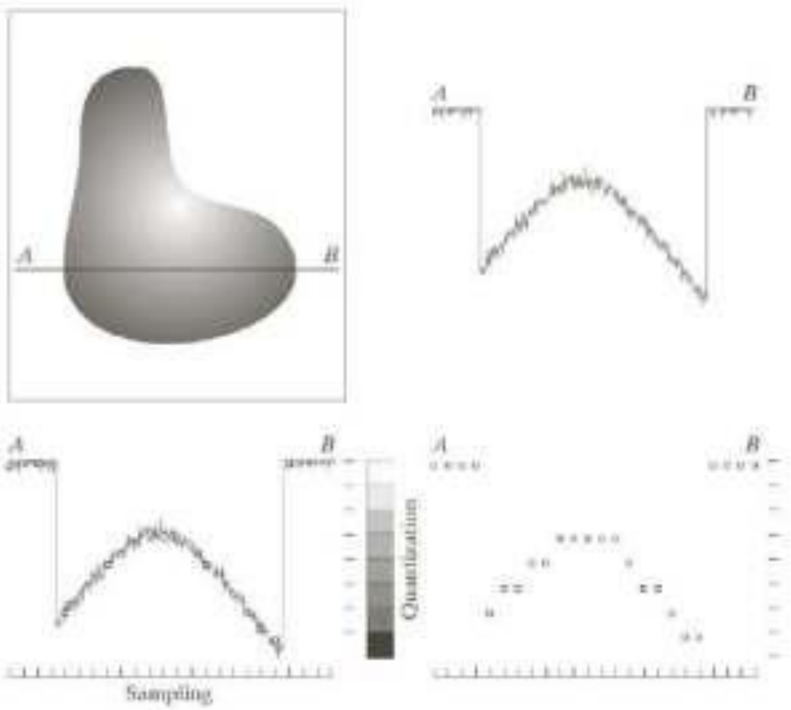
Which means that reflectance is bounded by 0 (total absorption) and 1 (total reflectance).

## 1.8 Image Sampling and Quantization

To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: sampling and quantization.

### Basic Concepts in Sampling and Quantization

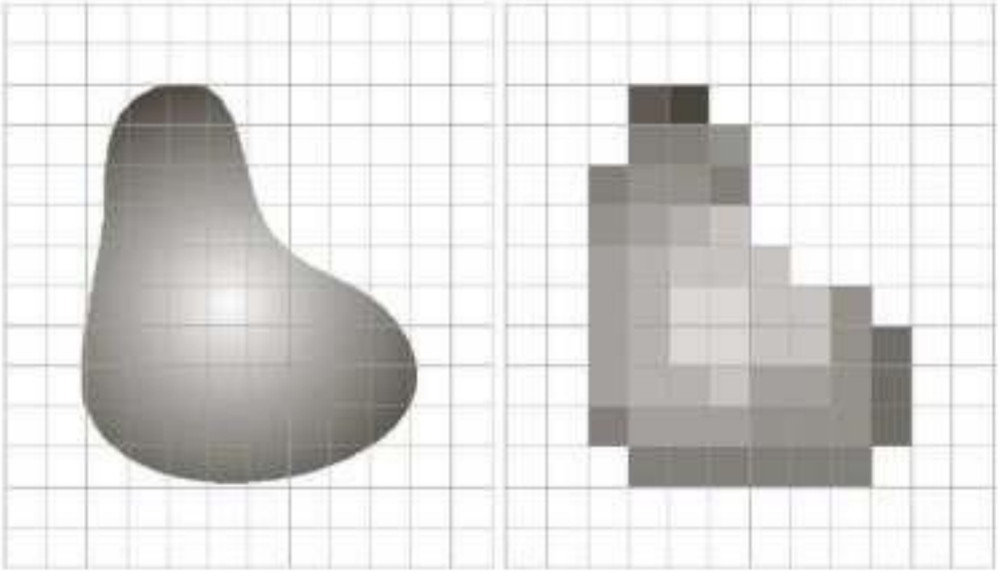
To convert an image to digital form, we have to sample the function in both coordinates and in amplitude. Digitizing the coordinate values is called sampling. Digitizing the amplitude values is called quantization. Figure 2.16 shows the basic idea behind sampling and quantization.




a b
c d

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

When a sensing array is used for image acquisition, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions.



a b

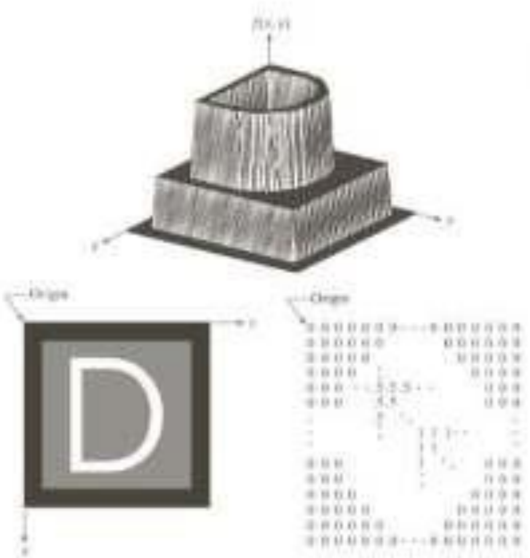
**FIGURE 2.17** (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

## Representing Digital Images

By applying sampling and quantization, we can convert a continuous image function of two continuous variables, $f(s,t)$, into a digital image $f(x,y)$, which contains $M$ rows and $N$ columns. $(x,y)$ are discrete coordinates: $x = 0,1,2,\ldots,M-1$ and $y = 0,1,2,\ldots,N-1$.

In general, the value of the image at any coordinates $(x,y)$ is denoted by $f(x,y)$, where $x$ and $y$ are integers.

The section of the real plane spanned by the coordinates of an image is called the spatial domain.



a
b c

**FIGURE 2.18** (a) Image plotted as a surface. (b) Image displayed as a visual intensity array. (c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

Figure 2.18 shows three basic ways to represent $f(x,y)$. The representations in Figure 2.18 (b) and (c) are the most useful. Image displays allow us to view results at a glance, and numerical arrays are used for processing and algorithm development.

In equation form, we write the representation of an $M \times N$ numerical array as

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

In some discussions, we use a more traditional matrix notation to denote a digital image as its elements:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Due to storage and quantizing hardware considerations, the number of intensity levels typically is an integer power of 2:

$$L = 2^k$$

We assume that the discrete levels are equally spaced and that they are integers in the interval $[0, L-1]$.

We define the dynamic range of an imaging system to be the ratio of the maximum measurable intensity to the minimum detectable intensity level in the system. As a rule, the upper limit is determined by saturation and the lower limit by noise.


**FIGURE 2.19** An image exhibiting saturation and noise. Saturation is the highest value beyond which all intensity levels are clipped (note how the entire saturated area has a high, constant intensity level). Noise in this case appears as a grainy texture pattern. Noise, especially in the darker regions of an image (e.g., the stem of the rose) masks the lowest detectable true intensity level.

Closely associated with the concept of dynamic range is image contrast, which is defined as the difference in intensity between the highest and lowest intensity levels in an image. The number of bits required to store a digitized image is

$b = M \times N \times k$. When $M = N$, becomes $b = N^2k$.

Number of storage bits for various values of $N$ and $k$.

| $N/k$ | 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 | 6,144 | 7,168 | 8,192 |
| 64 | 4,096 | 8,192 | 12,288 | 16,384 | 20,480 | 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 | 327,680 | 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 | 29,360,128 | 33,554,432 |
| 4096 | 16,777,216 | 33,554,432 | 50,331,648 | 67,108,864 | 83,886,080 | 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 |

## Spatial and Intensity Resolution

Spatial resolution is a measure of the smallest discernible detail in an image.

- Spatial resolution: Measure of the smallest discernible detail in an image.
  - Varying MxN or N (if M=N)

[Sampling is a principal factor of determining the spatial resolution of an image.]

- Gray Level Resolution: Smallest discernible change in gray level.
  - Note - it is a highly subjective process.
  - Varying –k‖

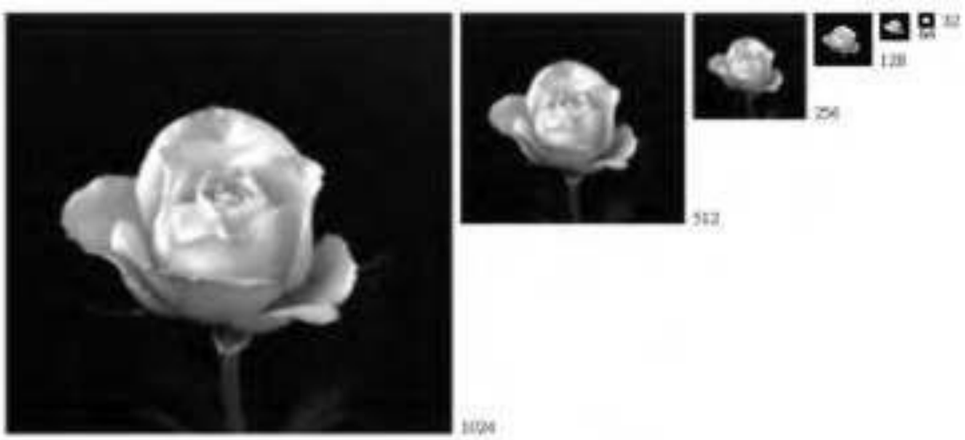



FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

Example of Spatial resolution - down to size Varying –N‖

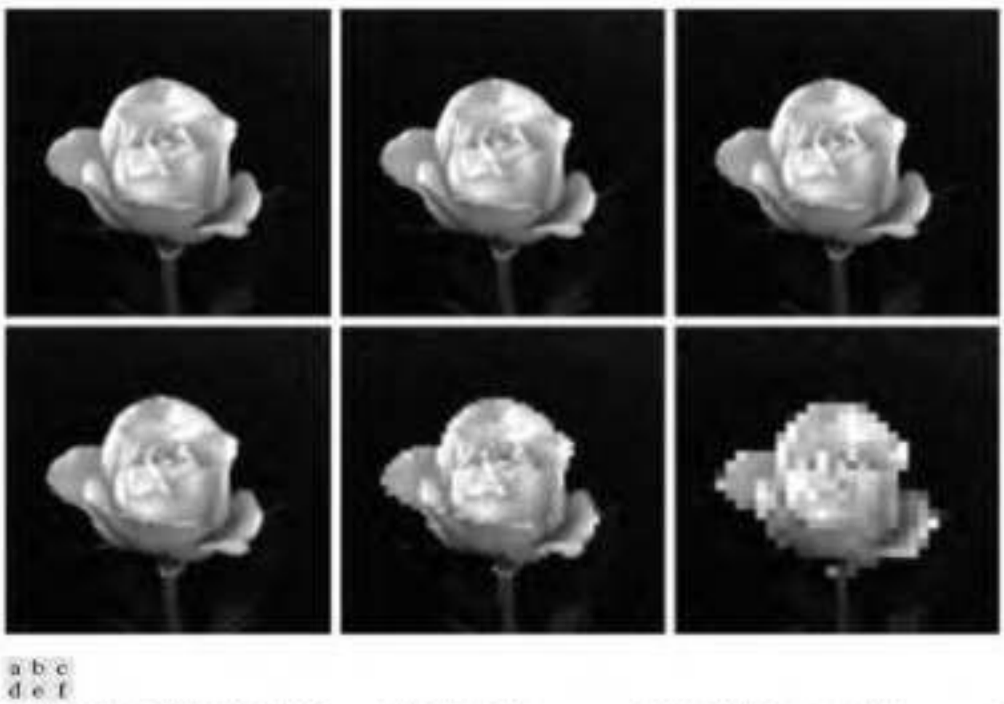
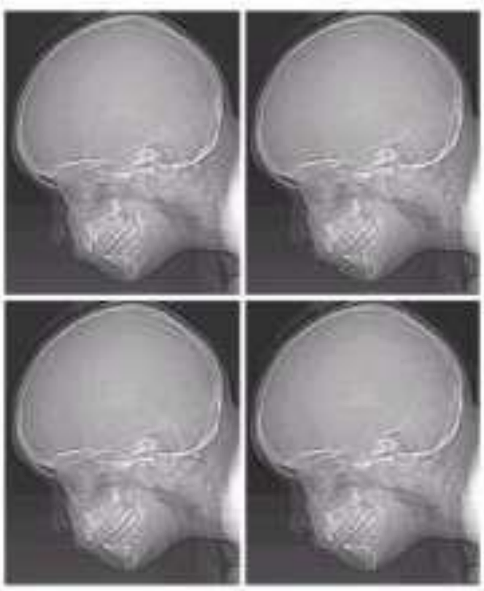
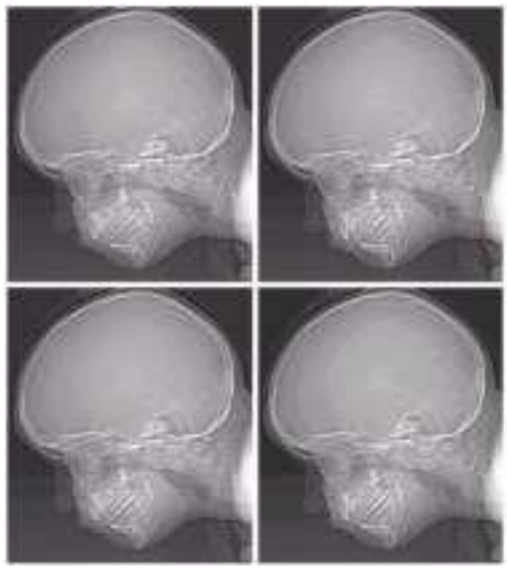



FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

Example of Spatial resolution - upsampled

Example 2.3: Typical effects of varying the number of intensity levels in a digital image.





Example of Gray-level resolution - down to size Varying –k‖

**Varying "N" & "k" simultaneously**

Huang [1965] attempted experiment to quantify by varying N & k on images shown in fig.

a b c

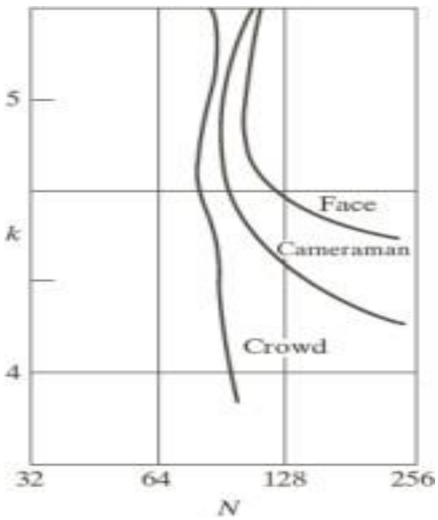
**FIGURE 2.22** (a) Image with a low level of detail. (b) Image with a medium level of detail. (c) Image with a relatively large amount of detail. (Image (b) courtesy of the Massachusetts Institute of Technology.)

Experiment:

1. N & k were varied

2. Observers were asked to rank them according to their subjective quality

3. Results were summarized in the form of ***–isopreference curves‖***




- Curves tends to become more vertical as details in the image increases.
- Images with large amount of detail needs only few gray levels.
- For Face and cameraman images:
  As N increases – number of gray level decreases

## Aliasing and Moire Patterns

- Shannon sampling theorem [Bracewell A995)]
- If the function is sampled at a rate equal to or greater than twice its highest frequency, it is possible to recover completely the original function from its samples.

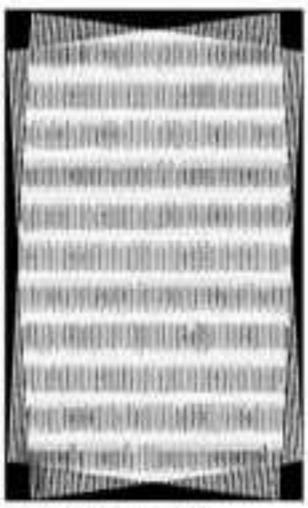


FIGURE 2.24 Illustration of the Moiré pattern effect.

- If the function is under-sampled, then a phenomenon called ***aliasing*** corrupts the sampled image.
- However, aliasing is always present in a sampled image.
- The effect of aliased frequencies can be seen under the right conditions in the form of Moire patterns.
- A Moire pattern, caused by a breakup of the periodicity, is seen in Fig. 2.24 as a 2-D sinusoidal (aliased) waveform (which looks like a corrugated tin roof) running in a vertical direction

Figure shows two identical periodic patterns of equal-equally spaced vertical bars, rotated in opposite directions and then superimposed on each other by multiplying the two images.

A Moire pattern, caused by a breakup of the periodicity, is seen in Fig.

## Zooming and Shrinking Digital Images

Sampling and quantization can be concluded using Zooming and Shrinking

1. Zooming may be viewed as over sampling
2. Shrinking may be viewed as under sampling.

Zooming requires two steps: the creation of new pixel locations, and the assignment of gray levels to those new locations.

Nearest neighbor interpolation:

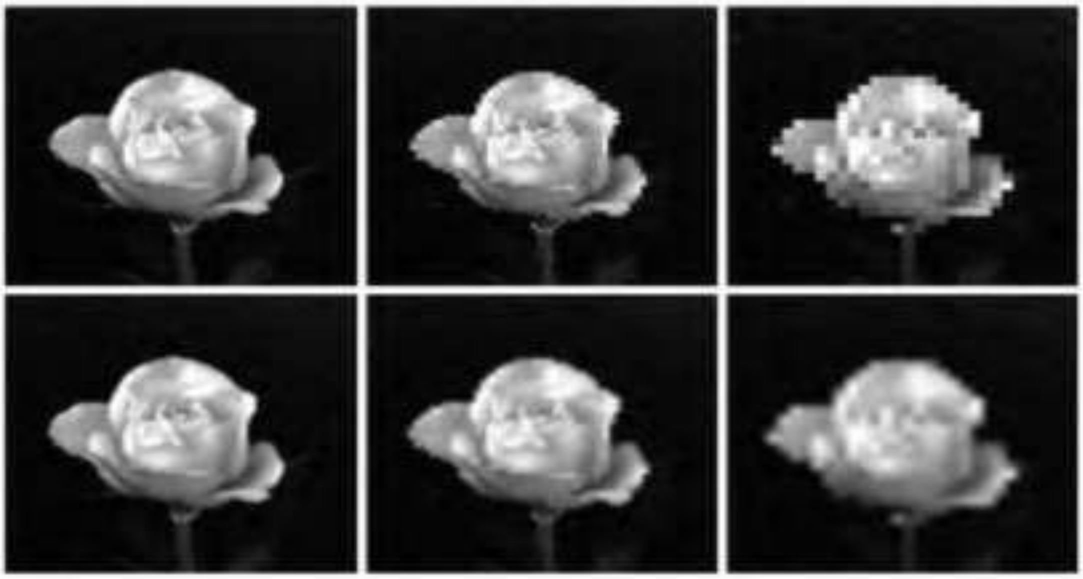
- Increase the size of an image e.g 1.5 times 500X500 = 750X750
- Lay an imaginary grid on the original image
- Assign the nearest pixel gray level to the new pixel in overlay grid

Pixel replication:

Pixel replication is applicable when we want to increase the size of an image an ***integer*** number of times.

We can duplicate each ***column***. This doubles the image size in the ***horizontal direction.***

Then, we duplicate each ***row*** of the enlarged image to double the size in the ***vertical direction***.



a b c
d e f

**FIGURE 2.25** Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

Image shrinking is done in a similar manner as just described for zooming. The equivalent process of pixel replication is row-column deletion. For example, to shrink an image by one-half, we delete every other row and column.

**1.9 Some Basic Relationships Between Pixels**

**<u>1.9.1 Neighbors of a Pixel</u>**

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

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$$

**4 – Neighbors**

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

**D – Neighbors**

The four *diagonal* neighbors of $p$ have coordinates

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

denoted by $N_D(p)$

**8 – Neighbors**

$$\mathbf{N_8 (P) = N_4 (P) + N_D (P)}$$

## 1.9.2 Adjacency, Connectivity, Regions, and Boundaries

Let V be the set of gray-level values used to define adjacency. e.g. V = {1}

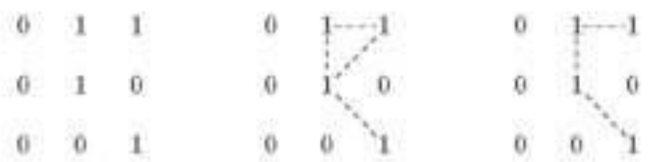
**4-adjacency:** Two pixels p and q with values from V are 4-adjacent if q is in the set N4 (P).

**8-adjacency:** Two pixels p and q with values from V are 8-adjacent if q is in the set N8 (P).

**m-a djacency (mixed adjacency):**

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

1. q is in N4(p),or
2. q is in ND(p) and the set N4(p) ∩ N4(q) has no pixels whose values are from V.



a b c

**FIGURE 2.26** (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.

**A {digital} path** (or curve) from pixel p with coordinates (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates.

**Connected set**:

Let ***S*** represent a subset of pixels in an image. Two pixels p and q are said to be connected in ***S*** if there exists a path between them consisting entire pixels in ***S***

**Region:** Region R is a subset of pixels in an image and if it is a connected set.

**Boundary:** The boundary (also called border or contour) of a region R is the set of pixels in the region that have one or more neighbors that are not in R.

### 1.9.3 Distance Measures

Let pixels be p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively,

D is a distance Junction or metric if:

(a) $D(p, q) \geq 0$ (D{p,q)=0 iff p = q),

(b) $D(p, q) = D(q, p)$ and

(c) $D(p, z) \leq D(p, q) + D(q, z)$.

The *Euclidean distance* between $p$ and $q$ is defined as

$$D_e(p, q) = \left[(x-s)^2 + (y-t)^2\right]^{\frac{1}{2}}.$$

The $D_4$ *distance* (also called *city-block distance*) between $p$ and $q$

$$D_4(p, q) = |x-s| + |y-t|.$$

The $D_8$ *distance* (also called *chessboard distance*) between $p$ and $q$ is

$$D_8(p, q) = \max(|x-s|, |y-t|).$$

### 1.9.4 Image Operations on a Pixel Basis

e.g. –Dividing one image by another,"

- Division is carried out between corresponding pixels in the two images.

### 1.10 Linear and Nonlinear Operations.

For any two images f and g and any two scalars a and b,

$$H(af + bg) = aH(f) + bH(g).$$

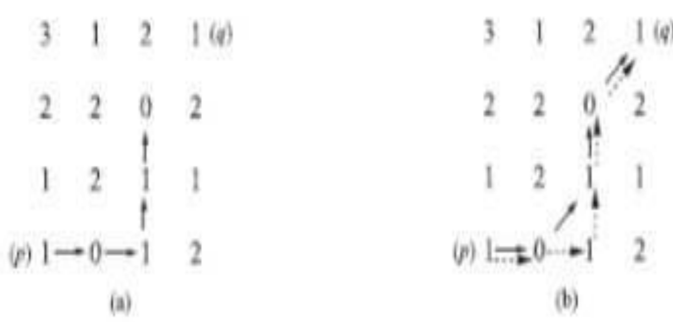
Linear – if it satisfies above equation, Non – Linear, If it doesn‘t

**Example 1.** Consider the image segment shown. Let V={0, 1} and compute the lengths of the shortest 4-, 8-, and m-path between p and q. If a particular path does not exist between these two points, explain why.

| | | | | |
|---|---|---|---|---|
| | 3 | 1 | 2 | 1 (q) |
| | 2 | 2 | 0 | 2 |
| | 1 | 2 | 1 | 1 |
| (p) | 1 | 0 | 1 | 2 |

Sol.

- When $V = \{0,1\}$, *4*-path does not exist between $p$ and $q$ because it is impossible to get from $p$ to $q$ by traveling along points that are both 4-adjacent and also have values from $V$. Fig. *a* shows this condition; it is not possible to get to $q$.
- The shortest *8*-path is shown in Fig. *b* its length is **4**.
- The length of the shortest *m*- path (shown dashed) is **5**.
- Both of these shortest paths are unique in this case.

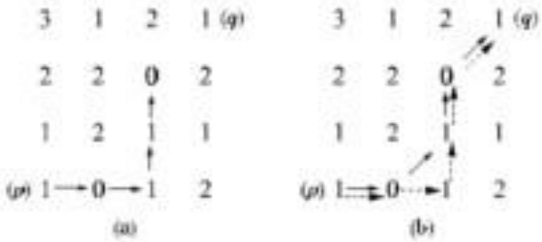



**Example 2:**

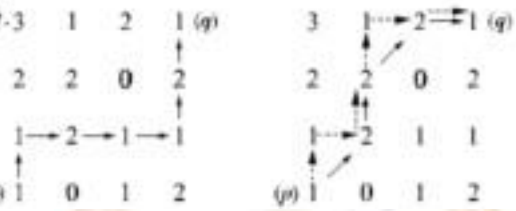
Consider the image segment shown.
**(a)** Let V={0, 1} and compute the lengths of the shortest 4-, 8-, and m-path between p and q. If a particular path does not exist between these two points, explain why.
**(b)** Repeat for V={1, 2}.

| | | | | |
|---|---|---|---|---|
| | 3 | 1 | 2 | 1 (q) |
| | 2 | 2 | 0 | 2 |
| | 1 | 2 | 1 | 1 |
| (p) | 1 | 0 | 1 | 2 |

- When $V = \{0,1\}$, 4-path does not exist between $p$ and $q$ because it is impossible to get from $p$ to $q$ by traveling along points that are both 4-adjacent and also have values from $V$ . Figure (a) shows this condition; it is not possible to get to $q$.
- The shortest 8-path is shown in Figure (b); its length is 4.
- The length of the shortest $m$- path (shown dashed) is 5. Both of these shortest paths are unique in this case.




**(b)** Repeat for V={1, 2}.



**Example 3:** Define 4-, 8- and m-adjacency. Compute the lengths of the shortest 4-, 8- and m-path between p and q in the image segment shown in Fig. by considering V = {2, 3, 4}

| | | | | | |
|---|---|---|---|---|---|
| | 3 | 4 | 1 | 2 | 0 |
| | 0 | 1 | 0 | 4 | 2 (q) |
| | 2 | 2 | 3 | 1 | 4 |
| (p) | 3 | 0 | 4 | 2 | 1 |
| | 1 | 2 | 0 | 3 | 4 |

4-, 8- and m-adjacency
4-, m- adjacency doesn't exist
Length of path – 8 adjacency = 4

**(p) → (q) : 3 – 2 – 3 – 4 – 2**

| Module – 2 | RBT Level |
|---|---|
| **Spatial Domain:** Some Basic Intensity Transformation Functions, Histogram Processing, Fundamentals of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters<br>**Frequency Domain**: Preliminary Concepts, The Discrete Fourier Transform (DFT) of Two Variables, Properties of the 2-D DFT, Filtering in the Frequency Domain, Image Smoothing and Image Sharpening Using Frequency Domain Filters, Selective Filtering.<br>**[Text: Digital Image Processing**- Rafel C Gonzalez and Richard E. Woods<br>**Chapter 3: Sections 3.2 to 3.6 and Chapter 4: Sections 4.2, 4.5 to 4.10]** | L1, L2, L3 |

## Image Enhancement in Spatial Domain

Spatial domain refers to the image plane itself, and image processing methods in this category are based on direct manipulation of pixels in an image.

Two principal categories of spatial processing are intensity transformations and spatial filtering.

Intensity transformations operate on single pixels of an image for the purpose of contrast manipulation and image thresholding.

Spatial filtering deals with performing operations, such as image sharpening, by working in a neighbourhood of every pixel in an image.

### 2.1 Some Basic Intensity Transformation Functions

Generally, spatial domain techniques are more efficient computationally and require less processing resources to implement. The spatial domain processes can be denoted by the expression

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

Where $f(x, y)$ is the input image, $g(x, y)$ is the output image, and $T$ is an operator on $f$ defined over a neighbourhood of point $(x, y)$. The operator can apply to a single image or to a set of images.

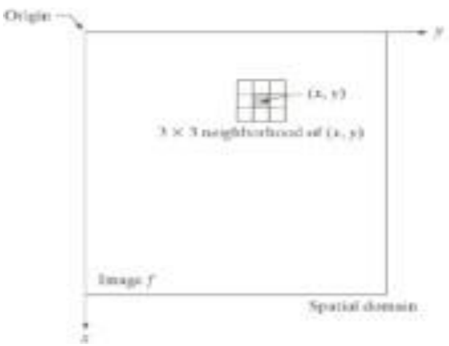



**FIGURE 3.1** A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

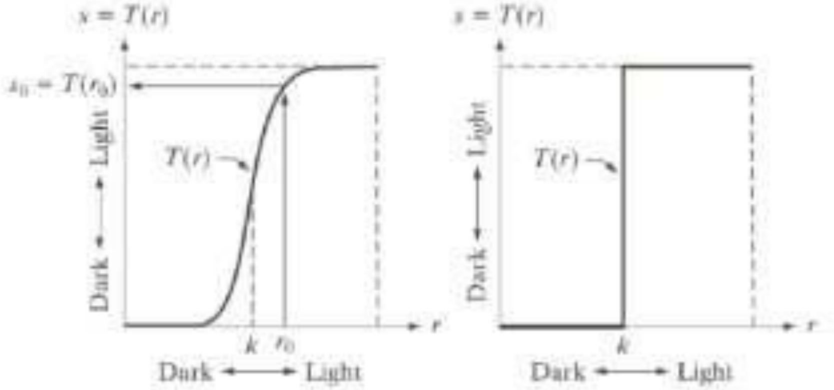
Typically, the neighbourhood is rectangular, centered on *(x, y)*, and much smaller than the image.
Example:

Suppose that the neighbourhood is a square of size 3×3 and the operator *T* is defined as ―compute the average intensity of the neighbourhood.‖

At an arbitrary location in an image, say *(10, 15)*, the output *g(10, 15)* is computed as the sum of *f(10, 15)* and its 8-neighbourhood is divided by *9*.

The origin of the neighbourhood is then moved to the next location and the procedure is repeated to generate the next value of the output image *g*.

The smallest possible neighbourhood is of size 1×1.




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

**Intensity transformations** are among the simplest of all image processing techniques. We use the following expression to indicate a transformation

$$s = T(r)$$

Where *T* is a transformation that maps a pixel value *r* into a pixel value *s*.

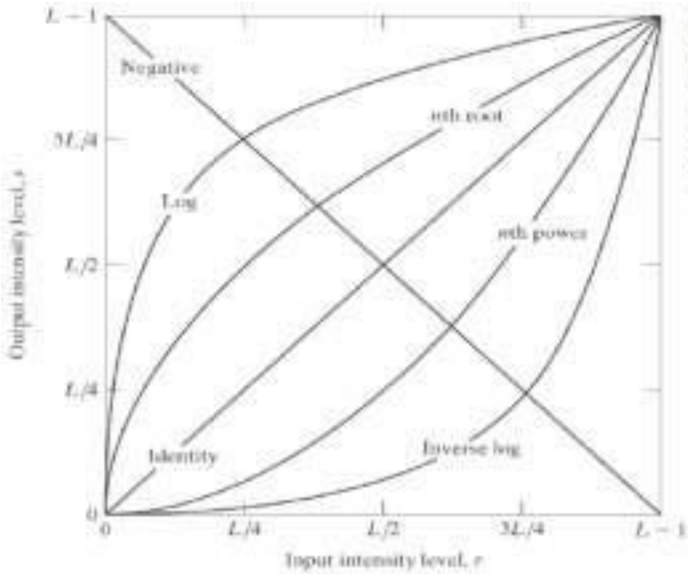



**FIGURE 3.3** Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

- **Image Negatives**

The negative of an image with intensity levels in the range $[0, L-1]$ is obtained by using the negative transformation shown in Figure 3.3, which is given by

$$s = L - 1 - r$$

The negative transformation can be used to enhance white or gray detail embedded in dark regions of an image.

- **Log Transformations**

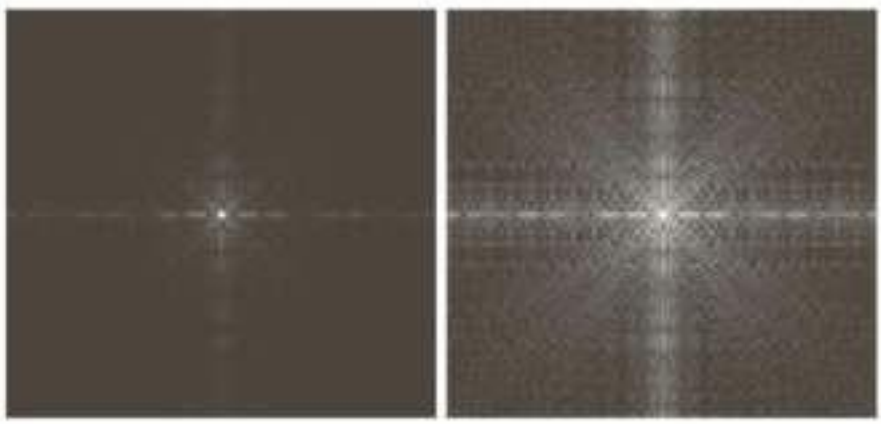
The general form of the log transformations is

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

where $c$ is a constant, and $r \geq 0$

The log transformation maps a narrow range of low intensity values in the input into a wider range of output levels. We use the transformation of this type to expend the values of dark pixels in an image while compress the higher-level values.

The opposite is true of the inverse log transformation.



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

Figure 3.5(a) shows a Fourier spectrum with values in the range $0$ to $1.5\times106$.

Figure 3.5(b) shows the result of applying (3.2-2) to the spectrum values, which will rescale the values to a range of $0$ to $6.2$, and displaying the results with an 8-bit system.

- **Power-Law (Gamma) Transformations**

Power-law transformations have the basic form

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

Where $c$ and $\gamma$ are positive constants. A variety of devices used for image capture, printing, and display according to a power-law. By convention, the exponent in the power-law equation is referred to as gamma.

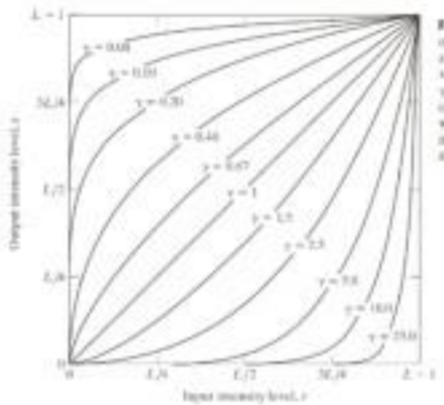
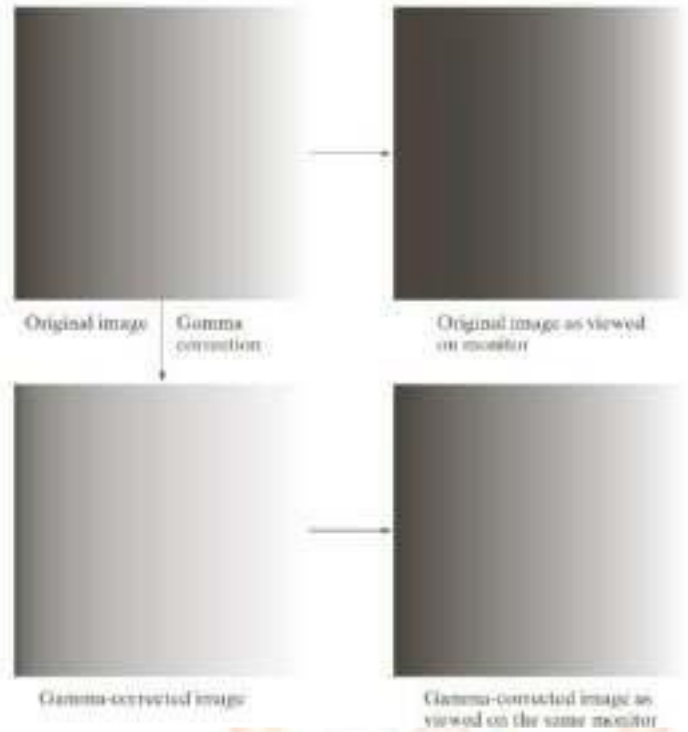


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of $\gamma$ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Unlike the *log* function, changing the value of $\gamma$ will obtain a family of possible transformations. As shown in Figure 3.6, the curves generated with values of $\gamma > 1$ have exactly the opposite effect as those generated with values of $\gamma < 1$.

The process used to correct these power-law response phenomena is called gamma correction.




a b
c d

**FIGURE 3.7** (a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

Gamma correction is important if displaying an image accurately on a computer screen is of concern.

Gamma correction has become increasingly important as the use of digital images over the Internet has increased.

In addition to gamma correction, power-law transformations are very useful for general-purpose contrast manipulation.

Example 3.1: Contrast enhancement using power-law transformations.

Example 3.2: Another illustration of power-law transformations.

Figure 3.9(a) shows the opposite problem of Figure 3.8(a).

## Piecewise-Linear Transformation Functions

A complementary approach to the abovementioned methods is to use piecewise linear functions.

- **Contrast stretching**
  One of the simplest piecewise linear functions is a contrast stretching transformation.
  Contrast-stretching transformation is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.

Example:

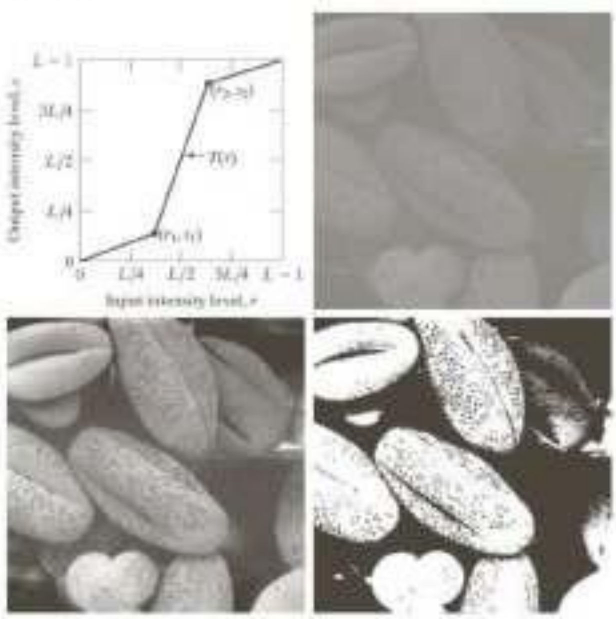
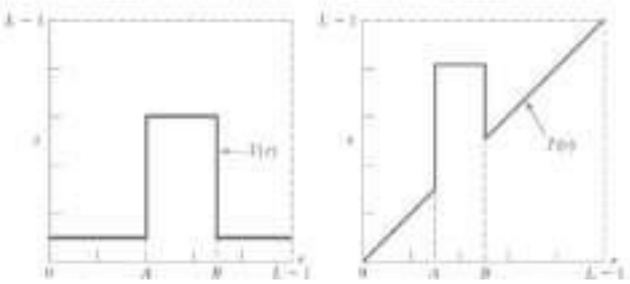


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

- **Intensity-level slicing**
  Highlighting a specific range of intensities in an image often is of interest. The process, often called intensity-level slicing, can be implemented in several ways, though basic themes are mostly used. One approach is to display in one value all the values in the range of interest and in another all other intensities, as shown in Figure 3.11 (a).

FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.



Another approach is based on the transformation in Figure 3.11(b), which brightens (or darkens) the desired range of intensities but leaves all other intensities levels in the image unchanged.

Example 3.3: Intensity-level slicing

a b c

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

Figure 3.12 (b) shows the result of using a transformation of the form in Figure 3.11 (a), with the selected band near the top of the scale, because the range of interest is brighter than the background.

Figure 3.12 (c) shows the result of using the transformation in Figure 3.11 (b) in which a band of intensities in the mid-gray region around the mean intensity was set to black, while all other intensities were unchanged.

- **Bit-plane slicing**

Instead of highlighting intensity-level ranges, we could highlight the contribution made to total image appearance by specific bits.

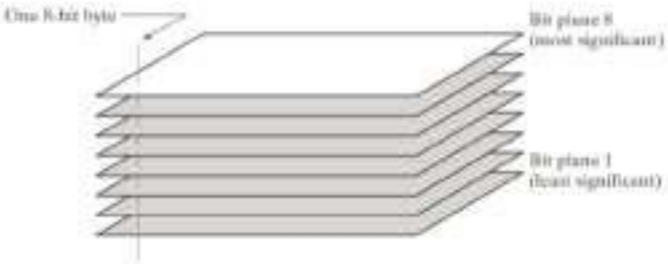



FIGURE 3.13 Bit-plane representation of an 8-bit image.

Figure 3.13 shows an 8-bit image, which can be considered as being composed of eight 1-bit planes, with plane 1 containing the lowest-order bit of all pixels in the image and plane 8 all the highest-order bits.

Example:

a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size $500 \times 1192$ pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Note that each bit plane is a binary image. For example, all pixels in the border have values 1 1 0 0 0 0 1 0, which is the binary representation of decimal 194. Those values can be viewed in Figure 3.14 (b) through (i). Decomposing an image into its bit planes is useful for analysing the relative importance of each bit in the image.

Example:

a b c

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

## 2.2 Histogram Processing

The histogram of a digital image with intensity levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$ , where $r_k$ is the $k^{th}$ intensity value and $n_k$ is the number of pixels in the image with intensity $r_k$.

It is common practice to normalize a histogram by diving each of its components by the total number of pixels in the image, denoted by $MN$, where $M$ and $N$ are the row and column dimensions of the image.

A normalized histogram is given by

$$p(r_k) = \frac{n_k}{MN}, \text{ for } k = 0, 1, 2, \dots, L-1.$$

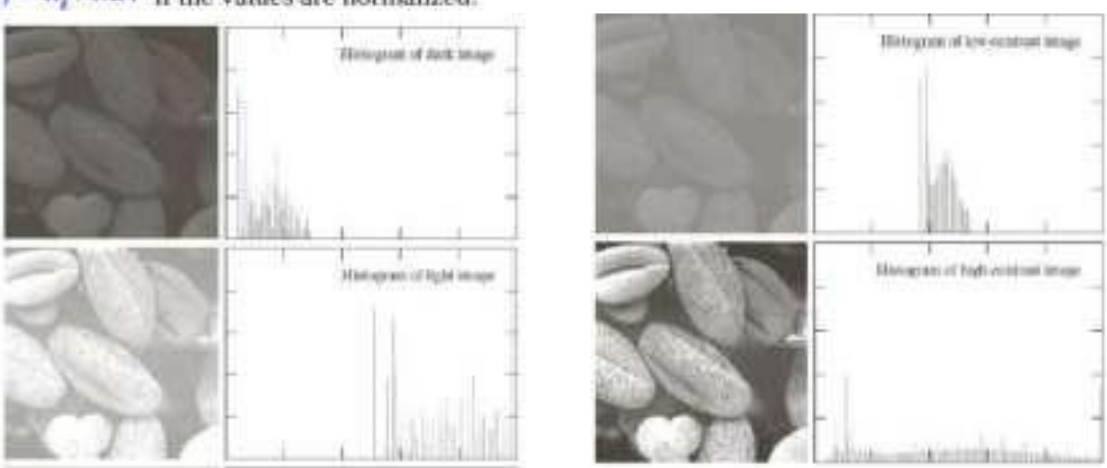
$P(r_k)$ can be seen as an estimate of the probability of occurrence of intensity level $r_k$ in an image. The sum of all components of a normalized histogram is equal to 1.

Histograms are the basic for numerous spatial domain processing techniques.

Example:

Figure 3.16, which is the pollen image of Figure 3.10 shown in four basic intensity characteristics: dark, light, low contrast, and high contrast, shows the histograms corresponding to these image.

The vertical axis corresponds to value of $h(r_k) = n_k$ or $p(r_k) = n_k / MN$ if the values are normalized.



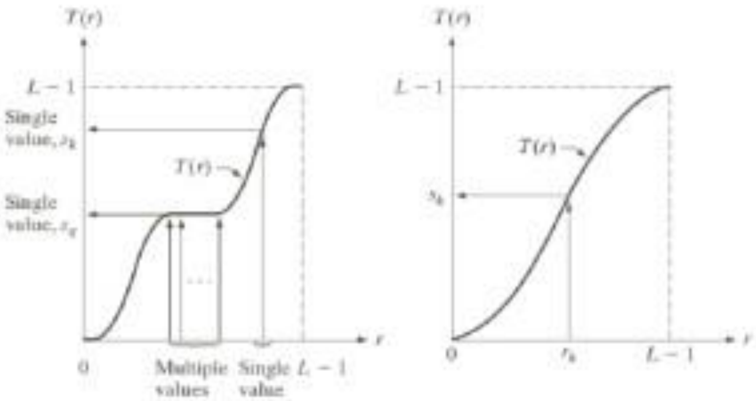
### 2.2.1 Histogram Equalization

We consider the continuous intensity values and let the variable $r$ denote the intensities of an image. We assume that $r$ is in the range $[0, L-1]$.
We focus on transformations (intensity mappings) of the form

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

that produce an output intensity level $s$ for every pixel in the input image having intensity $r$.
Assume that
(a) $T(r)$ is a monotonically increasing function in the interval $0 \le r \le L-1$, and
(b) $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.
In some formations to be discussed later, we use the inverse

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




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

From Figure 3.17 (a), we can see that it is possible for multiple values to map to a single value and still satisfy these two conditions, (a) and (b). That is, a monotonic transformation function can perform a one-to-one or many-to-one mapping, which is perfectly fine when mapping from $r$ to $s$.

However, there will be a problem if we want to recover the values of $r$ uniquely from the mapped values.

As Figure 3.17 (b) shows, requiring that $T(r)$ be strictly monotonic guarantees that the inverse mappings will be single valued. This is a theoretical requirement that allows us to derive some important histogram processing techniques.

**Prove that result of applying the transformation to all intensity levels „r". The resulting intensities „s" have a uniform PDF independently of the form of the PDF of the r"s**

Solution:
The intensity levels in an image may be viewed as random variables in the interval [0, $L$ -1].
A fundamental descriptor of a random variable is its probability density function (PDF).

Let $p_r(r)$ and $p_s(s)$ denote the probability density functions of $r$ and $s$. A fundamental result from basic probability theory is that if $p_r(r)$ and $T(r)$ are known, and $T(r)$ is continuous and differentiable over the range of values of interest, then the PDF of the transformed variable $s$ can be obtained using the formula

$$p_s(s) = p_r(\text{r})\left|\frac{dr}{ds}\right| \qquad \rightarrow (1)$$

A transformation function of particular importance in image processing has the form

$$s = T(\text{r}) = (L-1)\int_0^r p_r(\omega)\,d\omega \qquad \rightarrow (2)$$

where $\omega$ is a dummy variable of integration.

RHS in equation 2 is recognized as cumulative distribution function of random variable *r*

Since PDFs always are positive, the transformation function satisfies → condition (a)

Upper limit in equation is $r = (L-1)$ , the integral evaluates to 1, satisfies → condition (b)

Differentiating equation wrt _dr' we get:

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

$$= (L-1)\frac{d}{dr}\left[\int_0^r p_r(\omega)\,d\omega\right]$$

$$= (L-1)\,p_r(r)$$

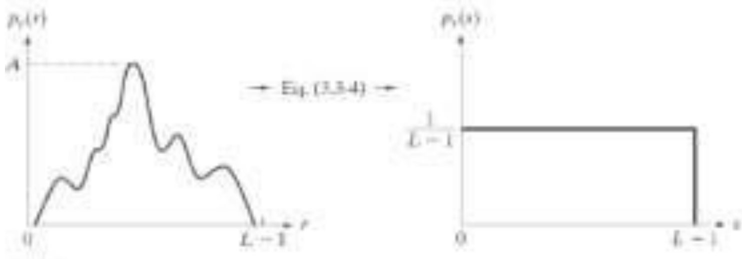
Substituting this result in equation 1

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

$$= p_r(r)\left|\frac{1}{(L-1)\,p_r(r)}\right|$$

$$= \frac{1}{L-1} \qquad 0 \le s \le L-1$$

which shows the that $p_s(s)$ always is uniform, independently of the form of $p_r(r)$ .




a b

which shows the that $p_s(s)$ always is uniform, independently of the form of $p_r(r)$ .

**Histogram equalization transformation**

Continuous case:

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

Discrete values:

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

**Example1. (Refer to Class Notes)**

**Given 64X64 image having 3 bit gray scales, perform histogram equalization and draw the histogram of image before and after equalization. Intensity distributions are shown in Table.**

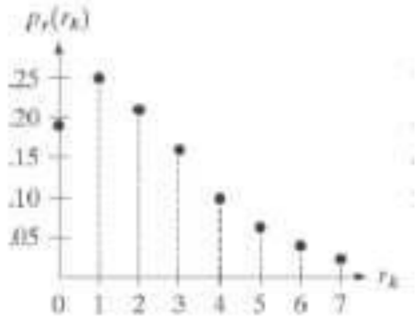
| $r_k$ | $n_k$ | $p_r(r_k) = n_k/MN$ |
|---|---|---|
| $r_0 = 0$ | 790 | 0.19 |
| $r_1 = 1$ | 1023 | 0.25 |
| $r_2 = 2$ | 850 | 0.21 |
| $r_3 = 3$ | 656 | 0.16 |
| $r_4 = 4$ | 329 | 0.08 |
| $r_5 = 5$ | 245 | 0.06 |
| $r_6 = 6$ | 122 | 0.03 |
| $r_7 = 7$ | 81 | 0.02 |

**TABLE 3.1** Intensity distribution and histogram values for a 3-bit, $64 \times 64$ digital image.

Solution:

3-bit image ( $L = 8$) of size 64´64 pixels ($MN = 4096$ )

The histogram of our hypothetical image is sketched in Figure

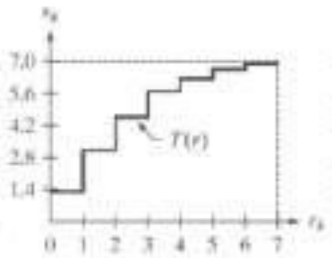



By using Trasformation equation, we can obtain values of the histogram equalization function:

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

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

$s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, and $s_7 = 7.00$. This function is shown in Figure 3.19 (b).



Then, we round them to the nearest integers:

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

which are the values of the equalized histogram.

Observe that there are only five distinct levels:

$s_0 \to 1$: 790 pixels
$s_1 \to 3$: 1023 pixels
$s_2 \to 5$: 850 pixels
$s_3 \to 6$: 985 (656+329) pixels
$s_5 \to 7$: 448 (245+122+81) pixels
Total: 4096

Dividing these numbers by $MN = 4096$ would yield the equalized histogram shown in Figure 3.19 (c).

Example 2:

For a given 4X4 image having 0 – 9 gray scales, perform histogram equalization and draw the histogram of image before and after equalization. 4X4 image is shown in Fig.

$$\begin{bmatrix} 2 & 3 & 3 & 2 \\ 4 & 2 & 4 & 3 \\ 3 & 2 & 3 & 5 \\ 2 & 4 & 2 & 4 \end{bmatrix}$$

Solution:

0 – 9 levels => L=10 levels

| **r** | **n** | **P=n/16** | **C_P Cumulative** | **L-1*C_P 9*C_P** | **Round up values- Equalized Intensity values** |
|---|---|---|---|---|---|
| 2 | 6 | 0.375 | 0.375 | 3.375 | 3 |
| 3 | 5 | 0.312 | 0.687 | 6.183 | 6 |
| 4 | 4 | 0.25 | 0.937 | 8.433 | 8 |
| 5 | 1 | 0.062 | 1 | 9 | 9 |

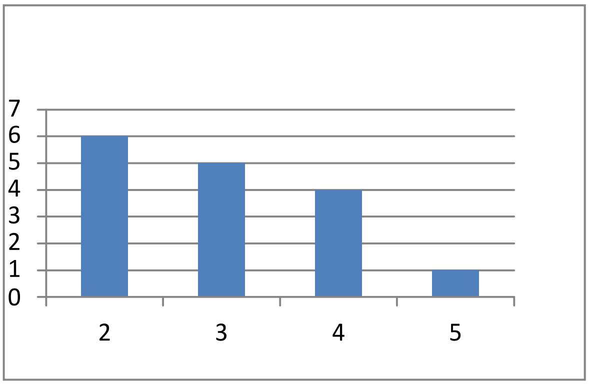
Equalized values 2→ 3 (6 pixels), 3→6 (5 pixels), 4→8 (4 pixels), 5→9 (1 pixel)

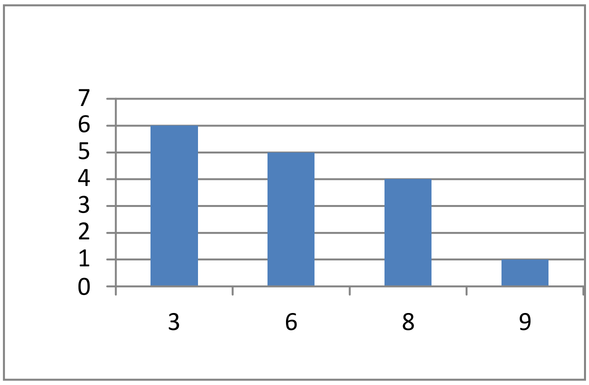
| | | | |
|---|---|---|---|
| 2 | 3 | 3 | 2 |
| 4 | 2 | 4 | 3 |
| 3 | 2 | 3 | 5 |
| 2 | 4 | 2 | 4 |

| | | | |
|---|---|---|---|
| 3 | 6 | 6 | 3 |
| 8 | 3 | 8 | 6 |
| 6 | 3 | 6 | 9 |
| 3 | 8 | 3 | 8 |

Given

Equalized




### 2.2.2 Histogram Matching

Histogram matching (histogram specification)—A processed image has a specified histogram

Let $p_r(r)$ and $p_z(z)$ denote the continous probability density functions of the variables $r$ and $z$. $p_z(z)$ is the specified probability density function.

Let $s$ be the random variable with the probability

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

Obtain a transformation function G

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

Histogram Matching: Discrete Cases

- Obtain $p_r(r_j)$ from the input image and then obtain the values of $s_k$, round the value to the integer range [0, L-1].

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

- Use the specified PDF and obtain the transformation function $G(z_q)$, round the value to the integer range [0, L-1].

- Mapping from $s_k$ to $z_q$

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) = s_k$$

$$z_q = G^{-1}(s_k)$$

Note: Refer to the Problems in Class Work

### 2.2.3 Local Histogram Processing:

Procedure:

1. Define a neighborhood and move its center from pixel to pixel
2. At each location, the histogram of the points in the neighborhood is computed. Either histogram equalization or histogram specification transformation function is obtained.
3. The function is used to Map the intensity of the pixel centered in the neighborhood
4. Move to the next location and repeat the procedure

### 2.2.4 Using Histogram Statistics for Image Enhancement

So far, image histogram used for enhancement.

We can also use- statistical parameters obtained from the histogram

**Mean value of r:**

$$m = \sum_{i=0}^{L-1} r_i p(r_i) \qquad = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x, y)$$

**Moment of r about its mean:**

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

**Variance**

$$\sigma^2 = u_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$$

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

Local variance

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

Local average intensity

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

$s_{xy}$ denotes a neighborhood

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

$m_G$ :global mean; $\sigma_G$ : global standard deviation

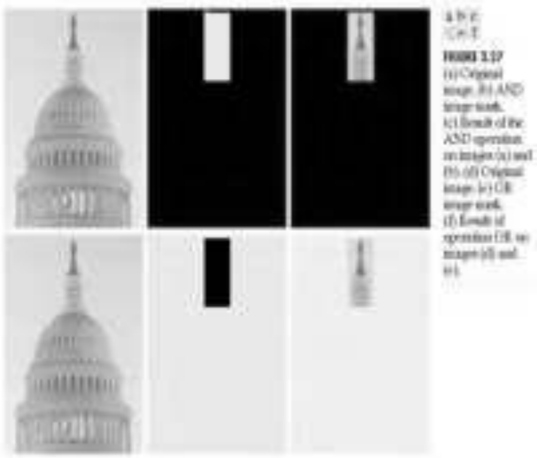
$k_0 = 0.4;\quad k_1 = 0.02;\ k_2 = 0.4;\ E = 4$

### 2.2.5 Enhancement using Arithmetic/logic operations

- Pixel by pixel operation between two or more images (except NOT operator)
- Addition/subtraction operations can be done sequentially or parallely (simultaneously)
- AND, OR, NOT – performed pixel by pixel.
- Other logical operators can be implemented using above operators.
- Pixels are represented in binary values e.g. 8 bit binary data
    - E.g NOT operations on 00000000 → 11111111

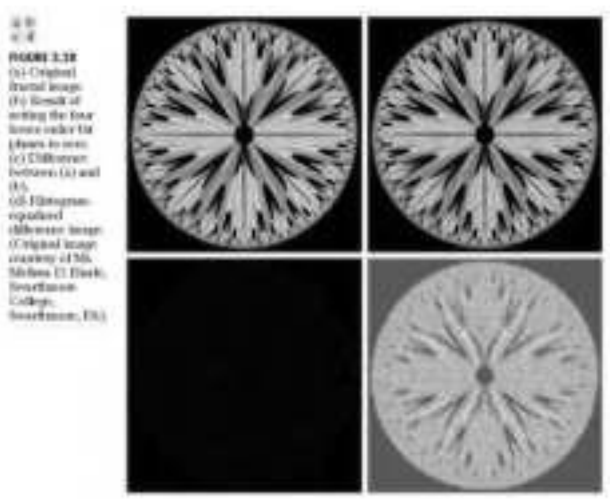
**AND and OR used for masking**

- Masking also known as – Region of Interest processing
- Masking is used to isolate an area for processing
- Highlights the area and differentiate from the rest



**Arithmetic operations**

- Addition and subtraction – widely used for image enhancement
- Division – same as multiplication of image by its reciprocal



### Image subtraction

Note: higher order bit-planes carry significant amount of viusally relevant information

g{x,y)=f{x,y)-h{x,y)

**Image Averaging:**

- g(x, y) be the noisy image formed by the addition of noise η(x, y) to an original image f(x, y) $$g(x, y) = f(x, y) + \eta(x, y)$$
- Noise is uncorrelated and has zero average value at each pair of coordinates, Then

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

an image $\bar{g}(x, y)$ is formed by averaging $K$ different noisy images,

$E\{\bar{g}(x, y)\}$ is the expected value of $\bar{g}$, Is equal to enhanced image

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

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

Note:

1. As K increases → variability (noise) decreases
2. $\bar{g}(x, y)$ approaches f(x, y) as number of noisy images (k) increases in averaging process

## 2.3 Fundamentals of Spatial Filtering

- Some operations work with the values of neighbourhood pixels.
- Create a Mask – sub image , filter, kernel, template or window
- Move mask from point to point in an image
- Result / response of mask = sum of products of the mask coefficients with the corresponding pixels directly under the mask.

$$R = w(-1,-1)f(x-1, y-1) + w(-1,0)f(x-1, y) + \cdots + w(0,0)f(x, y) + \cdots + w(1,0)f(x+1, y) + w(1,1)f(x+1, y+1)$$

So, A spatial filter consists of

(a) **a neighborhood**, and

(b) **a predefined operation** (mask)

Linear spatial filtering of an image of size MxN with a filter of size mxn is given by the expression

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

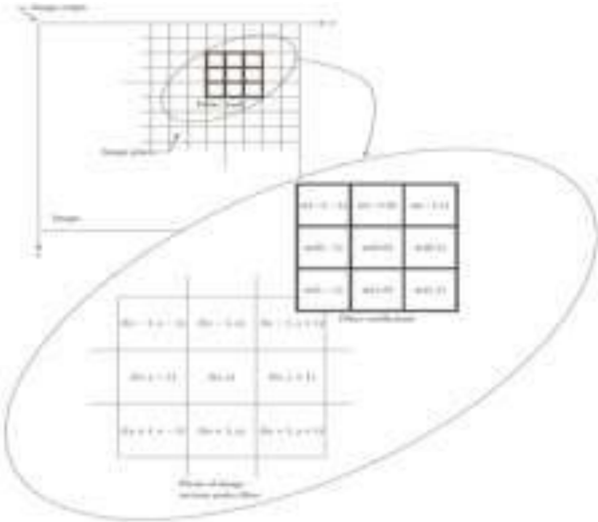
$$m = 2a + 1; \quad n = 2b + 1$$

a, b are non-negative values producing

Masks of odd size. E.g. 3, 5, 7, 9

m = no. of rows in mask

n = no. of columns in mask



## 2.4 Smoothing Spatial Filters

Smoothing filters are used for

- Blurring
- Noise reduction

Blurring is used in removal of small details and bridging of small gaps in lines or curves

2 ways of Smoothing spatial filters

1. linear filters
2. Non-linear filters.

### 2.4.1 Linear filter:

– ***Filter/Mask*** is Average of the pixels in the neighbourhood of the mask

– Idea is to, replace value of **every pixel** in an image by the average of its neighbourhood defined by the mask

– Reduces sharp transitions in the intensities

- Two types of masks (linear filters):
  – Averaging filter
  – Weighted averaging filter

***Averaging filter:***

– Also known as **low pass filter**

– Filter coefficients are equal is called ***Box Filter***

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

***Weighted Averaging filter:***

– Pixels multiplied by different coefficients

– Center pixel more weightage

– Other pixels are inversely weighted (distance from center of mask)

### 2.4.2 Order-statistic (Nonlinear) Filters

Nonlinear

— Based on ordering (ranking) the pixels contained in the filter mask

— Replacing the value of the center pixel with the value determined by the ranking result

- Edges are better preserved

E.g., median filter, max filter, min filter

Min filter:

- Filter selects the smallest value in window and replaces the center
- Enhances dark areas of image

Max filter:

- Replaces the center by largest value
- Enhances bright areas

Median Filter:

- Sort the neighbouring pixels
- Replaces center by median value

## 2.5 Sharpening Spatial Filters

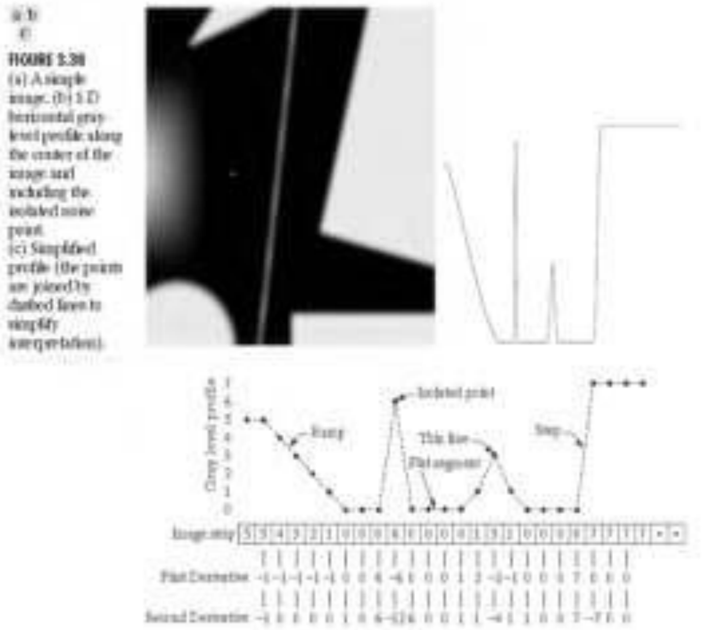
Foundation:

- ► Highlights fine details and enhance details that is blurred
- ► The first-order derivative of a one-dimensional function f(x) is the difference

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

- ► The second-order derivative of f(x) as the difference

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



- First order derivatives →
  - Produces thicker edges
  - Stronger response to a gray level step (…..7…..)
- Second order derivatives →
  - Stronger response to fine detail (such as thin lines and isolated points)
  - Produces double response at step changes in gray level (….7 -7..).

### 2.5.1 Sharpening Spatial Filters: Laplace Operator

Sharpened image = original image – laplacian operator

The second-order isotropic derivative operator is the Laplacian for a function (image) f(x,y)

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

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

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

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

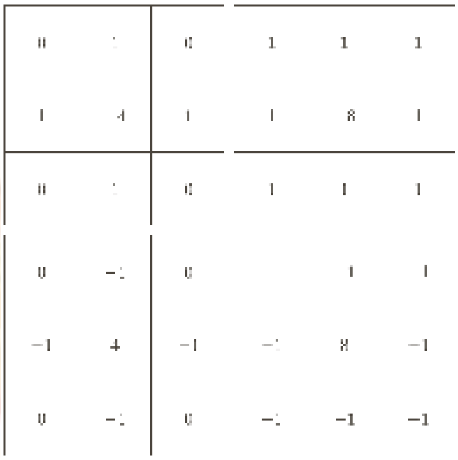
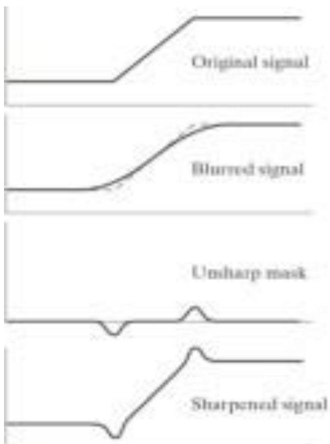


Image sharpening in the way of using the Laplacian:

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

### 2.5.2 Unsharp Masking and Highboost Filtering




► Unsharp masking

Sharpen images by subtracting blurred version of an image from image itself

e.g., printing and publishing industry

► Steps

1. Blur the original image
2. Subtract the blurred image from the original (Resulting difference image – called as mask)
3. Add the mask to the original

First, we obtain the mask:

mask(x,y)=f(x,y)- f_blur(x,y)

Then we add weighted portion of the mask to original image:

g(x, y) = f(x, y) + k * mask(x, y)

Where k>=0

If K = 1 → Unsharp Masking

If k > 1 → Highboost Filtering

a b

**FIGURE 3.42** The high-boost filtering technique can be implemented with either one of these masks, with $A \geq 1$.

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

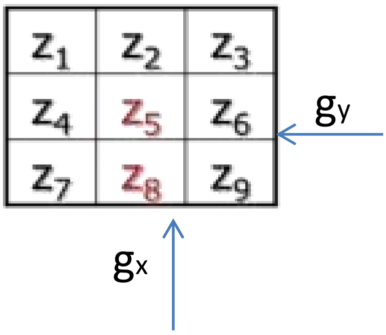
### 2.5.3 Image Sharpening based on First-Order Derivatives

For function $f(x, y)$, the gradient of $f$ at coordinates $(x, y)$ is defined as

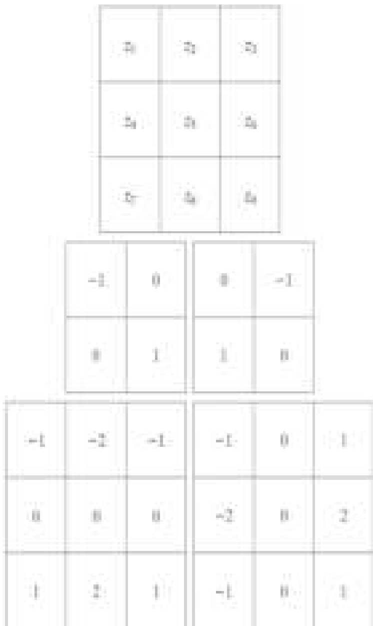
$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The *magnitude* of vector $\nabla f$, denoted as $M(x, y)$

Gradient Image $M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$




$$M(x, y) = |z_8 - z_5| + |z_6 - z_5|$$




Sobel Operators

Roberts Cross-gradient Operators

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

$$M(x, y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

# Image Enhancement in Frequency Domain

Joseph Fourier (21 March 1768 – 16 May 1830) was a French mathematician and physicist best known for initiating the investigation of Fourier series and their application to problems of heat transfer.

One of the most important Fourier's contributions states that any periodic function can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient. We now call this sum a Fourier series.

It does not matter how complicated the function is, if it is periodic and satisfies some mild mathematical conditions, it can be represented by Fourier series.

FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Even functions that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighting function.

The formulation in this case is the Fourier transform, and its utility is even greater than the Fourier series in many theoretical and applied disciplines.

One of the most important characteristics of these representations is that a function, expressed in either a Fourier series or transform, can be reconstructed completely via an inverse process with no loss of information. This characteristic allows us to work in the Fourier domain and then return to the original domain of the function without losing any information.

The initial application of Fourier's ideas was in the field of heat diffusion. The advent of digital computers and the ―discovery‖ of a fast Fourier transform (FFT) algorithm in the early 1960s revolutionized the field of signal processing.

We will show that Fourier techniques will provide a meaningful and practical way to study and implement a host of image processing approaches.

## 2.6 Preliminary Concepts

### 2.6.1 Complex numbers

A complex number, $C$, is defined as

$$C = R + jI$$

where $R$ and $I$ are real numbers, and $j$ is an imaginary number equal to $\sqrt{-1}$.
The conjugate of a complex number $C$, denoted $C*$, is defined as

$$C^* = R - jI$$

Sometimes, it is useful to represent complex numbers in polar coordinates,

$$C = C\,(\cos\theta + j\sin\theta)$$

where $|C| = \sqrt{(R^2 + I^2)}$ is the length of the vector extending from the origin of the complex plane to the point $(R, I)$ , and $\theta$ is the angle between the vector and the real axis.

we have the following familiar representation of complex numbers in polar coordinates

$$C = C\, e^{j\theta}$$

$$\theta(u) = \arctan[I(u)/R(u)]$$

### 2.6.2 Fourier Series

A function $f(t)$ of a continuous variable $t$ that is periodic with period, $T$ , can be expressed as the sum of sines and cosines multiplied by appropriate coefficients. The sum, known as a Fourier series, has the form.

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{T}t}$$

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t) e^{-j\frac{2\pi n}{T}t} dt \quad \text{for } n = 0, \pm 1, \pm 2, \ldots$$

### 2.6.3 Impulses and Their Sifting Property

Central to the study of linear systems and the Fourier transform is the concept of an impulse and its sifting property. A unit impulse of a **continuous variable** $t$ located at $t = 0$ , denoted $\delta(t)$ , is defined as

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0 \end{cases}$$

It also satisfies the identity

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

An impulse has the sifting property with respect to integration

$$\int_{-\infty}^{\infty} f(t)\delta(t) dt = f(0)$$

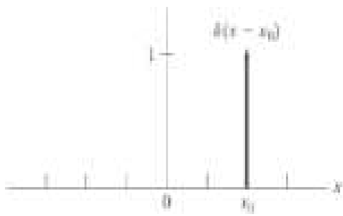
$f(t)$ is continuous at $t = 0$.

A more general statement of the sifting property (wrf to integration)

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_0) dt = f(t_0)$$

The **unit discrete impulse**, $\delta(x)$, is defined as

$$\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$$



Sifting property for discrete variables is given by

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x) = f(0)$$

General Form:

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x - x_0) = f(x_0)$$

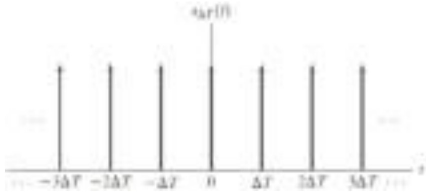



**FIGURE 4.3** An impulse train.

An impulse train, $s_{\Delta T}(t)$, defined as the sum of infinitely many periodic impulses $\Delta T$ units apart:

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T)$$

### 2.6.4 The Fourier Transform of Functions of One Continuous Variable

Fourier transform of a continuous function $f(t)$

$$\mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}dt \qquad \mathcal{F}\{f(t)\} = F(\mu)$$

Therefore, Fourier transform may be written as

$$F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}dt$$

Using Euler's formula

$$F(\mu) = \int_{-\infty}^{\infty} f(t)[\cos(2\pi\mu t) - j\sin(2\pi\mu t)]dt$$

Inverse Fourier transform may be written as $f(t) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t}d\mu$

### 2.6.5 Convolution

The convolution of $f(t)$ & $h(t)$ is defined as: $f(t)\star h(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau$

By a few steps, we can find the so-called Fourier transform pair:

$$f(t)\star h(t) \Leftrightarrow H(\mu)F(\mu)$$

and

$$f(t)h(t) \Leftrightarrow H(\mu)\star F(\mu)$$

The convolution theorem is the foundation for filtering in the frequency domain.

## 2.7 The Discrete Fourier Transform (DFT) of Two Variables

### 2.7.1 The 2-D Impulse and its Shifting property

- The impulse, $\delta(t, z)$, of ***two continuous variables***, $t$ and $z$,

$$\delta(t,z) = \begin{cases} \infty & \text{if } t = z = 0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(t,z)\,dt\,dz = 1.$$

2-D impulse exhibits the *sifting property* under integration, and given by

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t,z)\,dt\,dz = f(0,0)$$

More generally for an impulse located at ( $t_0$, $z_0$)

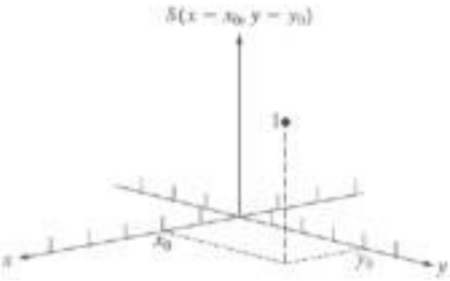
$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t-t_0,z-z_0)\,dt\,dz = f(t_0,z_0).$$

- For discrete variables $x$ and $y$ , the 2-D discrete impulse $\delta(x, y)$

$$\delta(x,y) = \begin{cases} 1 & \text{if } x = y = 0 \\ 0 & \text{otherwise} \end{cases}$$

and its sifting property is

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x,y) = f(0,0)$$




For an impulse located at ($x_0$, $y_0$) , as shown in Figure

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x-x_0,y-y_0) = f(x_0,y_0)$$

### 2.7.2 The 2-D Continuous Fourier Transform Pair

Let $f(t, z)$ be a continuous function of two continuous variables

2-D continuous Fourier transform pair is given by:

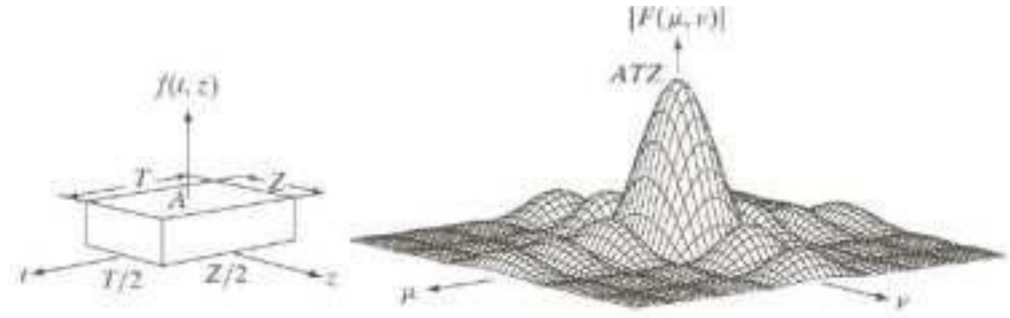
$$F(\mu,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-j2\pi(\mu t+\nu z)}dt\,dz$$

and

$$f(t,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,\nu)e^{j2\pi(\mu t+\nu z)}d\mu\,d\nu$$

where $\mu$ and $\nu$ are the frequency variables.

**Example: Obtaining the 2-D Fourier transform of a simple function**




$$\begin{aligned} F(\mu,\nu) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-j2\pi(\mu t+\nu z)}dt\,dz \\ &= \int_{-T/2}^{T/2}\int_{-Z/2}^{Z/2} Ae^{-j2\pi(\mu t+\nu z)}dt\,dz \\ &= ATZ\left[\frac{\sin(\pi\mu T)}{(\pi\mu T)}\right]\left[\frac{\sin(\pi\nu Z)}{(\pi\nu Z)}\right] \end{aligned}$$

The magnitude is given by the expression

$$|F(\mu,\nu)| = ATZ\left|\frac{\sin(\pi\mu T)}{(\pi\mu T)}\right|\left|\frac{\sin(\pi\nu Z)}{(\pi\nu Z)}\right|$$

### 2.7.3 Two-Dimensional Sampling and Sampling Theorem

Sampling in 2-D can be modeled using the sampling function (2-D impulse train):

$$s_{\Delta T\Delta Z}(t,z) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\delta(t-m\Delta T, z-n\Delta Z)$$

Where *T* and *Z* are the separations between samples along and *z*-axis.

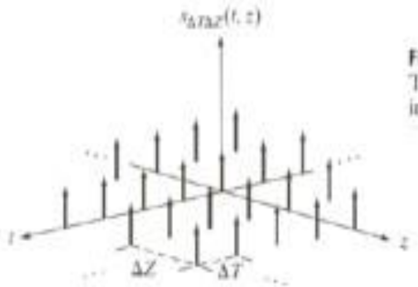



**FIGURE 4.14** Two-dimensional impulse train.

Function $f(t, z)$ is said to be band-limited if its Fourier transform is 0 outside a rectangle established by the intervals $[-\mu_{max}, \mu_{max}]$ and $[-\nu_{max}, \nu_{max}]$:

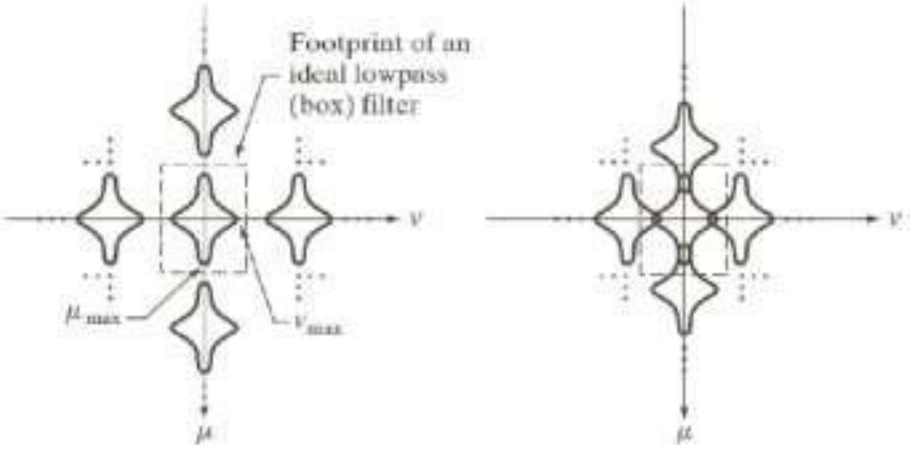
$$F(\mu, \nu) = 0 \quad \text{for } |\mu| \geq \mu_{max} \quad \text{and} \quad |\nu| \geq \nu_{max}$$

The two-dimensional sampling theorem states that a continuous, band-limited function $f(t, z)$ can be recovered with no error from a set of its samples if the sampling intervals are

$$\frac{1}{\Delta T} > 2\mu_{max}$$

and

$$\frac{1}{\Delta Z} > 2\nu_{max}.$$




a b
**FIGURE 4.15** Two-dimensional Fourier transforms of (a) an over-sampled, and (b) under-sampled band-limited function.

**Aliasing in Images:**

There are two principal manifestations of aliasing in images:

- Spatial aliasing, which is due to under-sampling;
- Temporal aliasing, which is related to time intervals between images in a sequence of images.

**Assignment**

### 2.7.4 The 2-D Discrete Fourier Transform and Its Inverse

The 2-D discrete Fourier transform (DFT) is

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)},$$

for $u = 0, 1, 2, \ldots, M-1$ and $v = 0, 1, 2, \ldots, N-1$.

Given $F(u, v)$, we can obtain $f(x, y)$ by using the inverse discrete Fourier transform (IDFT):

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)},$$

for $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, N-1$.

## 2.8 Properties of the 2-D DFT

### 2.8.1 Relationships between Spatial and Frequency Intervals

- $f(t, z)$ continuous function
- $f(x,y)$ sampled digital image consisting $(M \times N)$ samples taken in the $t$ and $z$-directions.
- $\Delta T$ and $\Delta Z$ separations between samples
- The corresponding discrete, frequency domain variables are given by
- $$\Delta u = \frac{1}{M\Delta T}$$ and $$\Delta v = \frac{1}{N\Delta Z}.$$

### 2.8.2 Translation and Rotation

The Fourier transform pair satisfies the following translation properties:

$$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$$

and

$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(x_0u/M+y_0v/N)}$$

Using the polar coordinates

$$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi,$$

we have the following transform pair:

$$f(r, \theta+\theta_0) \Leftrightarrow F(\omega, \varphi+\theta_0).$$

Equation indicates that rotating $f(x,y)$ by an angle $\theta_0$ will rotate $F(u,v)$ by the same angle. Conversely, rotating $F(u,v)$ will rotate $f(x,y)$ by the same angle.

### 2.8.3 Periodicity

The 2-D Fourier transform and its inverse are infinitely periodic in the $u$ and $v$ directions:

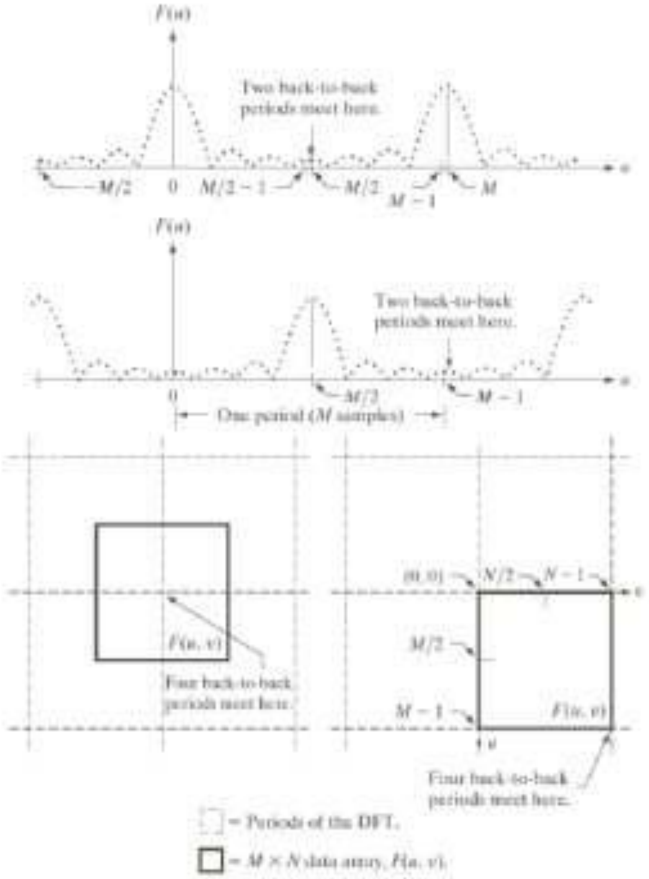
$$F(u,v) = F(u+k_1M, v) = F(u, v+k_2N) = F(u+k_1M, v+k_2N)$$

and

$$f(x,y) = f(x+k_1M, y) = f(x, y+k_2N) = f(x+k_1M, y+k_2N)$$

where $k_1$ and $k_2$ are integers.

$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$$




a
b
c d

**FIGURE 4.23** Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$. (c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained with Eq. (4.5-15). This array consists of four quarter periods. (d) A Shifted DFT obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).

### 2.8.4 Fourier Spectrum and Phase Angle

- Since the 2-D DFT is complex in general, it can be expressed in polar form:

$$F(u,v) = |F(u,v)| e^{j\phi(u,v)},$$

where

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

is the Fourier (frequency) spectrum, and

$$\phi(u,v) = \arctan\left[\frac{I(u,v)}{R(u,v)}\right]$$

is the phase angle.

The power spectrum is defined as

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v),$$

The spectrum has even symmetry about the origin

$$|F(u,v)| = |F(-u,-v)|, \qquad ($$

and phase angle has the odd symmetry about the origin $\phi(u,v) = -\phi(-u,-v)$,

### 2.8.5 The 2-D Convolution Theorem

1–D Convolution

$$f(x)\star h(x) = \sum_{m=0}^{M-1} f(m)h(x-m)$$

2–D Circular Convolution

$$f(x,y)\star h(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)$$

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

The 2-D convolution theorem is given by

$$f(x,y)\star h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

and, conversely,

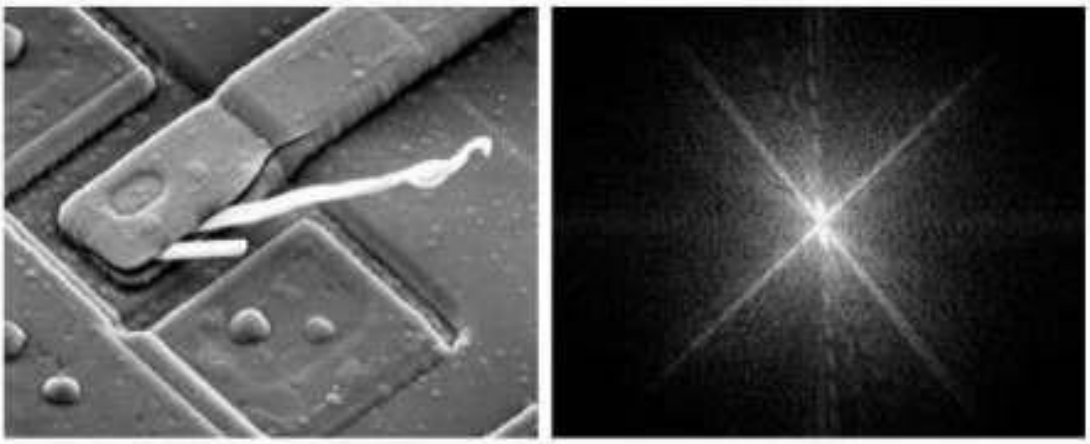
$$f(x,y)h(x,y) \Leftrightarrow F(u,v)\star H(u,v)$$

## 2.9Filtering in the Frequency Domain

- Filtering Technique in frequency domain based on
  - Modifying DFT to achieve specific objectives
  - Compute IDFT to get back to image domain

  e.g. $F(u,v) = |F(u,v)|e^{j\phi(u,v)}$

  We have access to magnitude (spectrum) and the phase angle
- Visual analysis of the Magnitude (spectrum)
  - Phase component – not useful
  - Spectrum – provides gross characteristics of the image



| Two notable features: strong edges that run approximately at ±45 degrees and two white oxide protrusions. | Vertical component that is off-axis slightly to the left is caused due to the edges of the white oxide protrusions |
|---|---|

Given a digital image $f(x, y)$, of size $M \times N$, filtering Equation is given by:

$$g(x, y) = \mathcal{F}^{-1}[H(u, v)F(u, v)]$$

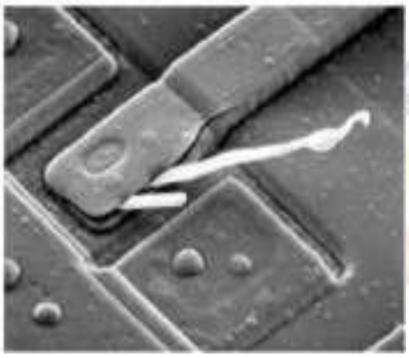
$\mathcal{F}^{-1}$ is the IDFT, $F(u, v)$ is the DFT

$H(u, v)$ is a filter function

$g(x, y)$ is the filtered image

- One of the simplest filters we can construct is A filter $H(u,v)$
  - 0 at the center of the transform and
  - 1 elsewhere

**(This filter will reject the *DC* term and pass all other terms)**



**SEM image of damaged IC**

**Filtered image by Setting F(M/2, N/2) to 0 in the Fourier Transform**

**Lowpass filter:**

- High Frequencies → Due to Sharp transitions (edges & Noise)
- Filter $H(u,v)$ should attenuate high frequencies
- low frequencies → would blur an image

**Highpass filter:**

- Enhances sharp details
- But Causes, reduction in contrast

## Steps for Filtering in the Frequency Domain

1. Given an input image $f(x, y)$ of size $M \times N$, obtain the padding parameters $P$ and $Q$ from

$$P \geq 2M - 1 \qquad (4.6\text{-}31)$$

and

$$Q \geq 2N - 1. \qquad (4.6\text{-}32)$$

Typically, we select $P = 2M$ and $Q = 2N$.

2. Form a padded image, $f_p(x, y)$, of size $P \times Q$ by appending the necessary number of zeros to $f(x, y)$.

3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center its transform.

4. Compute the DFT, $F(u, v)$, of the image from Step 3.

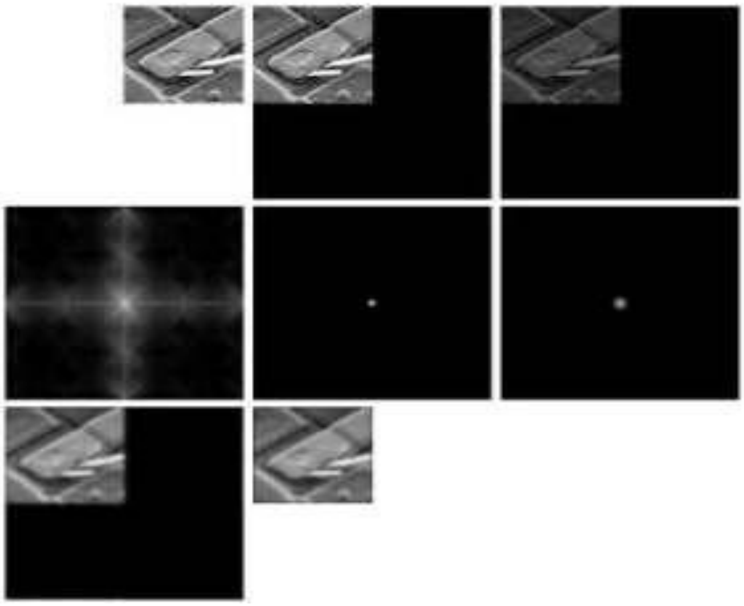
5. Generate a real, symmetric filter function, $H(u, v)$, of size $P \times Q$ with center at $(P/2, Q/2)$. Form the product $G(u, v) = H(u, v)F(u, v)$ using array multiplication $G(i, k) = H(i, k)F(i, k)$.

6. Obtain the processed image:

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

where the real part is selected in order to ignore parasitic complex components resulting from computational in accuracies.

7. Obtain the final processed result, $g(x, y)$, by extracting the $M \times N$ region from the top, left quadrant of $g_p(x, y)$.



a b c
d e f
g h

**FIGURE 4.36** (a) An $M \times N$ image, $f$. (b) Padded image, $f_p$ of size $P \times Q$. (c) Result of multiplying $f_p$ by $(-1)^{x+y}$. (d) Spectrum of $F_p$. (e) Centered Gaussian lowpass filter, $H$, of size $P \times Q$. (f) Spectrum of the product $HF_p$. (g) $g_p$, the product of $(-1)^{x+y}$ and the real part of the IDFT of $HF_p$. (h) Final result, $g$, obtained by cropping the first $M$ rows and $N$ columns of $g_p$.

## 2.10 Image Smoothing

Smoothing (blurring) is achieved in the frequency domain by lowpass filtering.

### 2.10.1 Ideal Lowpass Filters

Passes all frequencies within a circle of radius $D0$ from the origin and ―cuts off‖ all frequencies outside this circle is called an ideal lowpass filter (ILPF)

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

where $D_0$ is a positive constant, and $D(u,v)$ is the distance between a point $(u,v)$ in the frequency domain and the center of the frequency rectangle

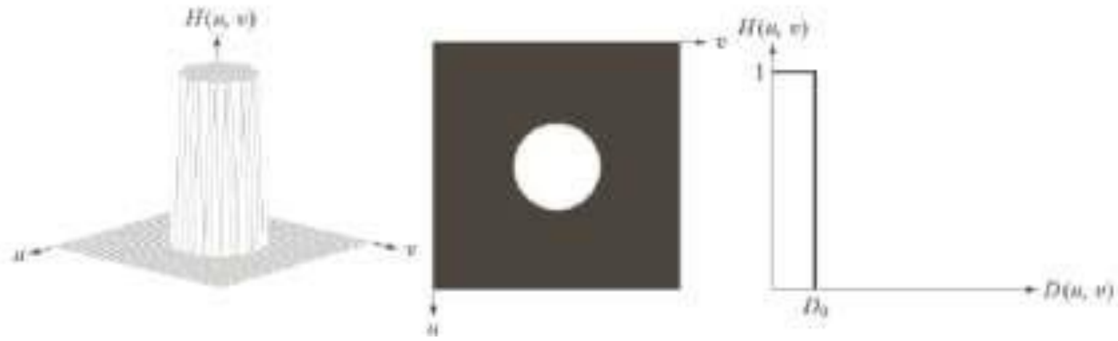
$$D(u,v) = \sqrt{(u-P/2)^2 + (v-Q/2)^2},$$

where $P$ and $Q$ are the padded sized from

$$P \ge 2M - 1$$

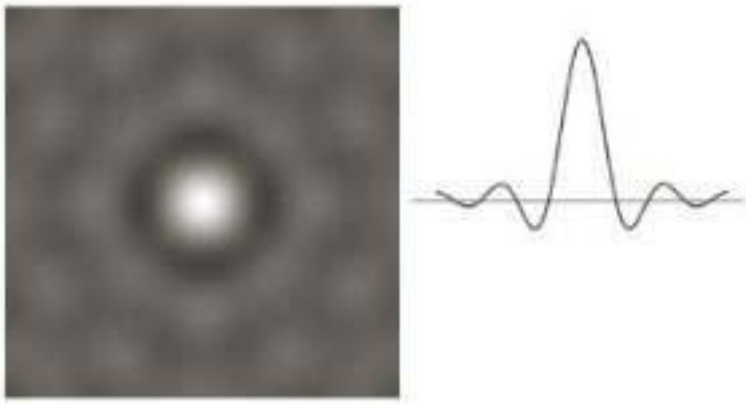
and

$$Q \ge 2N - 1.$$




a b
c d
e f

**FIGURE 4.42** (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.



a b

**FIGURE 4.43** (a) Representation in the spatial domain of an ILPF of radius 5 and size 1000 × 1000. (b) Intensity profile of a horizontal line passing through the center of the image.

- ILPF in the frequency domain looks like a box filter
- Corresponding spatial filter has the shape of a sinc function.
- The center lobe of the sinc is the principal cause of blurring,
- While the outer, smaller lobes are mainly responsible for ringing.

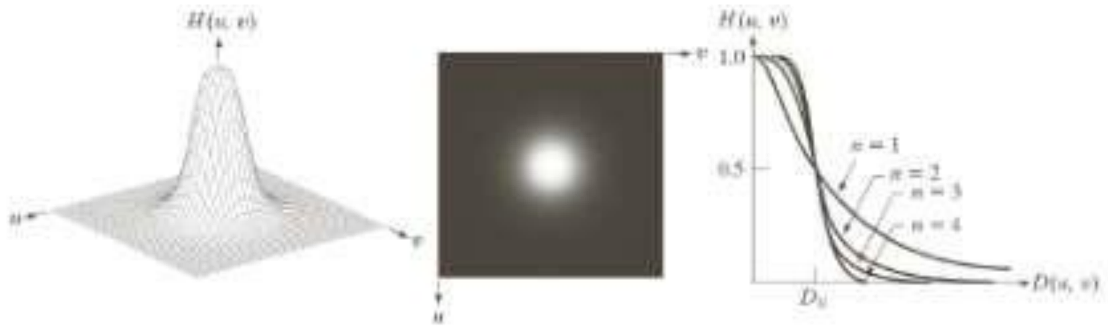
### 2.10.2 Butterworth Lowpass Filters

The transfer function of a Butterworth lowpass filter (BLPF) of order $n$, and with cutoff frequency at a distance $D_0$ from the origin, is defined as

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}},$$

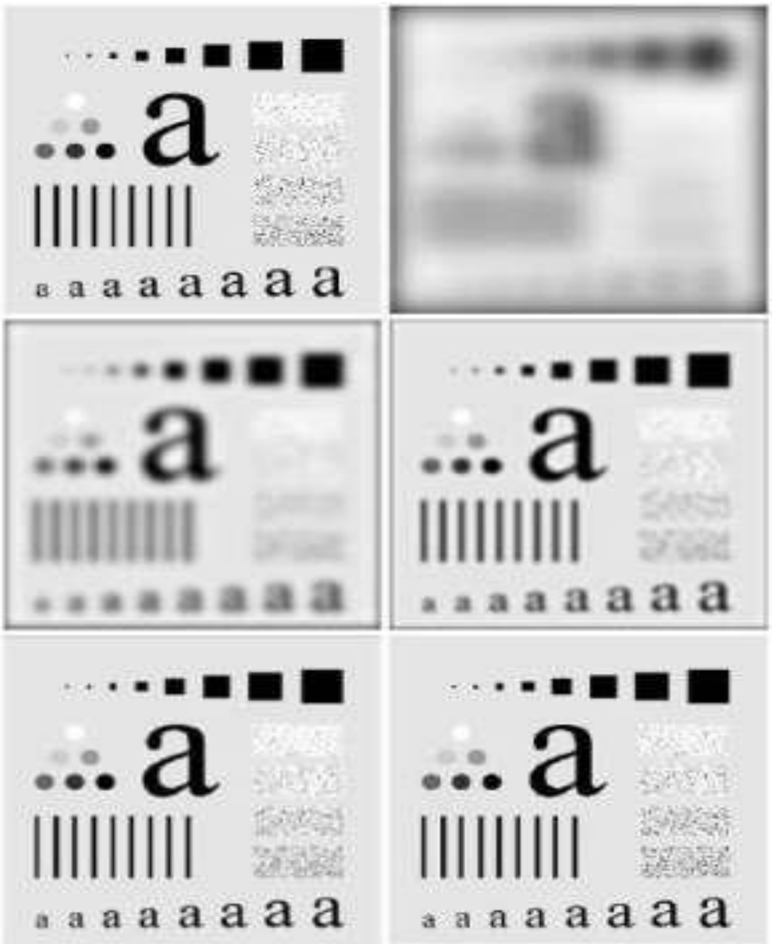
where $D(u,v)$ is given by

$$D(u,v) = \sqrt{(u - P/2)^2 + (v - Q/2)^2}.$$




a b c

**FIGURE 4.44** (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

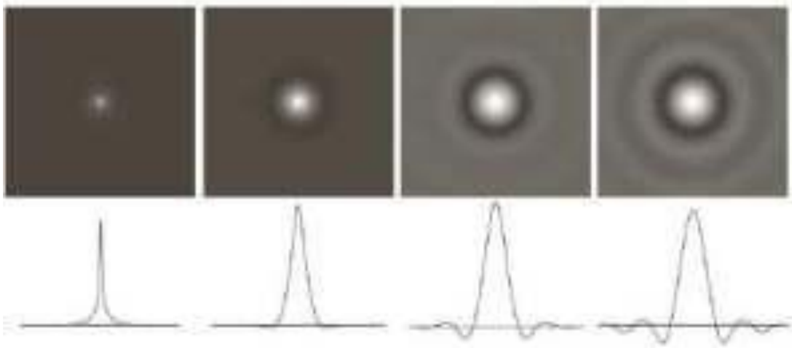


(b) – (f) results obtained with cut off frequencies at radii 10, 30, 60, 160 & 460

a b
c d
e f

**FIGURE 4.45** (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

A BLPF of order 1 has no ringing in the spatial domain. Ringing can become significant in filters of higher order.



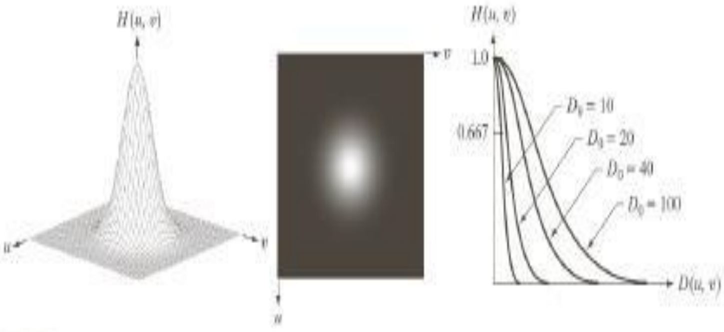
## 2.10.3 Gaussian Lowpass Filters

The form of Gaussian lowpass filters (GLPFs) in two dimensions is given by

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

Where σ = $D_0$ measure of spread about the center,

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




a b c

**FIGURE 4.47** (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of $D_0$.

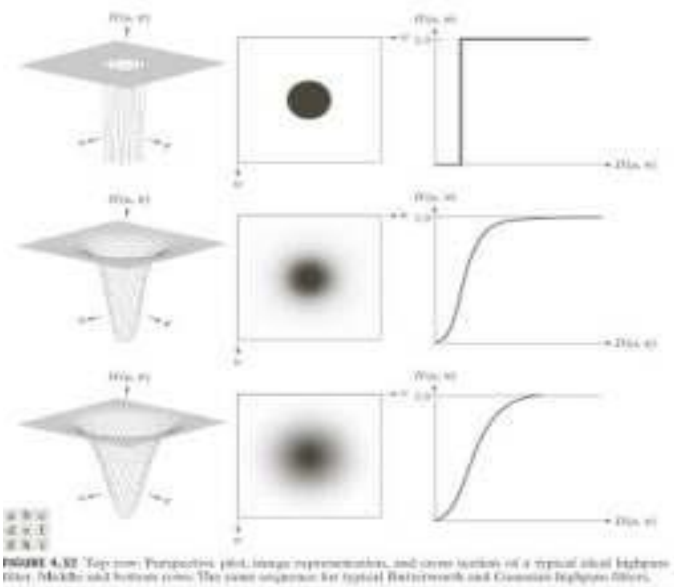
## 2.11 Image Sharpening Using Frequency Domain Filters

**Sharpening is achieved in the frequency domain by Highpass filtering.**

- **A highpass filter is obtained from a given lowpass filter using the equation**

$$H_{HP}(u,v) = 1 - H_{LP}(u,v),$$

**<u>Ideal, Butterworth, and Gaussian highpass filters</u>**



### Ideal Highpass Filters

A 2-D ideal highpass filter (IHPF) is defined as

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases} \qquad (4.9\text{-}2)$$

Because of the way they are related, we can expect IHPFs to have the same ringing properties as ILPFs.

Figure 4.54 shows the various IHPF results of using the original image in Figure 4.41 (a) with $D_0$ equal to 30, 60, and 160 pixels.

a b c

**FIGURE 4.54** Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with $D_0 = 30, 60,$ and 160.

### Butterworth Highpass Filters

A 2-D Butterworth highpass filter (BHPF) of order $n$ and cutoff frequency $D_0$ is defined as

$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2n}}, \qquad (4.9\text{-}3)$$

where $D(u,v)$ is given by

$$D(u,v) = \sqrt{(u - P/2)^2 + (v - Q/2)^2}. \qquad (4.8\text{-}2)$$

Figure 4.55 shows the performance of a BHPF.

a b c

**FIGURE 4.55** Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

### Gaussian Highpass Filters

The transfer function of the Gaussian highpass filter (GHPF) with cutoff frequency locus at a distance $D_0$ from the center of the frequency rectangle is defined as

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2} \qquad (4.9\text{-}4)$$

Figure 4.56 shows some comparable results from using GHPF s.

a b c

**FIGURE 4.56** Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

As expected, the results are more gradual than with the IHPFs and BHPFs.

## The Laplacian in the Frequency Domain

- The Laplacian was used for image enhancement in the spatial domain.
- Laplacian can yield equivalent results using frequency domain techniques.
- The Laplacian can be implemented in the frequency domain using the filter

$$H(u,v) = -4\pi^2(u^2+v^2),$$

$$H(u,v) = -4\pi^2\left[(u-P/2)^2+(v-Q/2)^2\right] = -4\pi^2 D^2(u,v)$$

Then, the Laplacian image is obtained as

$$\nabla^2 f(x,y) = -\mathscr{F}^{-1}\{H(u,v)F(u,v)\},$$

where $F(u,v)$ is the DFT of $f(x,y)$.

## Homomorphic Filtering:

- This technique uses illumination-reflectance model in its operation.
- 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).

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

- The intensity of $i(x, y)$ changes slower than $r(x, y)$.
- $i(x, y)$ is considered to have more low frequency components than $r(x, y)$
- Homomorphic filtering technique aims to reduce the significance of $i(x, y)$ by reducing the low frequency components of the image.

Implemented using five stages:

**STAGE 1**: Take a natural logarithm of both sides to decouple $i(x, y)$ and $r(x, y)$ components and apply transforms.

**STAGE 2**: Use the Fourier transform to transform the image into frequency domain:

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

or

$$Z(u, v) = F_i(u, v) + F_r(u, v)$$

Where $F_i(u, v)$ and $F_r(u, v)$ are the Fourier transforms of $\ln i(x, y)$ and $\ln r(x, y)$ 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:

**STAGE 4**: Take an inverse Fourier transform to get the filtered image in the spatial domain:

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

- In stage 1, natural log (ln) was considered, so filtered image is given as:

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

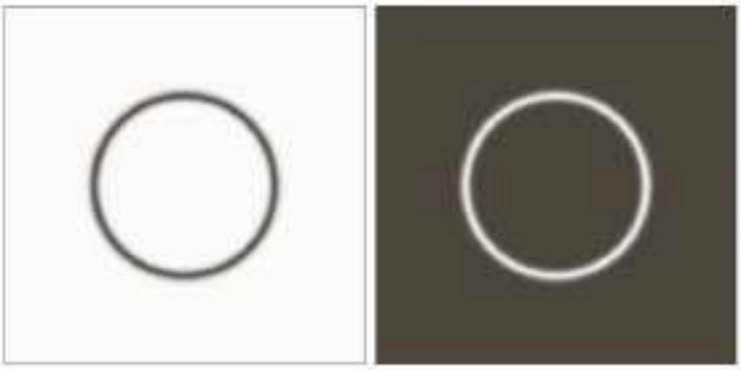
[Illumination and reflectance are not separable, but their approximate locations in the frequency domain may be located.]

## 2.12 Selective Filtering

Process specific bands of frequencies or small regions of the rectangle, which are called bandreject or bandpass filters.

**Bandreject and Bandpass Filters**

| Ideal | Butterworth | Gaussian |
|---|---|---|
| $H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$ | $H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$ | $H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$ |



a b

**FIGURE 4.63** (a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

A bandpass filter is obtained from a bandreject filter

$$H_{BP}(u,v) = 1 - H_{BR}(u,v). \qquad (4.10\text{-}1)$$

**Notch Filters**

A notch filter will reject (or pass) frequencies in a predefined neighbourhood.

Notch reject filters are constructed as products of highpass filters whose centers have been translated to the centers of the notches

$$H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v),$$

where $H_k(u,v)$ and $H_{-k}(u,v)$ are highpass filters whose centers are at $(u_k, v_k)$ and $(-u_k, -v_k)$. These "centers" are specified to the center of the frequency rectangle, $(M/2, N/2)$.

**A notch pass filter is obtained from a notch reject filter by**

$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$

**Module – 3**

**Restoration:** Noise models, Restoration in the Presence of Noise Only using Spatial Filtering and Frequency Domain Filtering, Linear, Position- Invariant Degradations, Estimating the Degradation Function, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Constrained Least Squares Filtering.

**[Text: Digital Image Processing**- Rafel C Gonzalez and Richard E. Woods

**Chapter 5: Sections 5.2, to 5.9]**

### 3.1 Restoration

The principal goal of restoration techniques is to improve an image in some predefined sense.

Although there are areas of overlap, image enhancement is largely a subjective process, while restoration is for the most part an objective process.

Restoration attempts to recover an image that has been degraded by using a priori knowledge of the degradation phenomenon. Thus, restoration techniques are oriented toward modelling the degradation and applying the inverse process in order to recover the original image.

The restoration approach usually involves formulating a criterion of goodness that will yield an optimal estimate of the desired result, while enhancement techniques are heuristic procedures to manipulate an image in order to take advantage of the human visual system.

Some restoration techniques are best formulated in the spatial domain, while others are better suited for the frequency domain.

**A Model of the Image Degradation/Restoration Process**

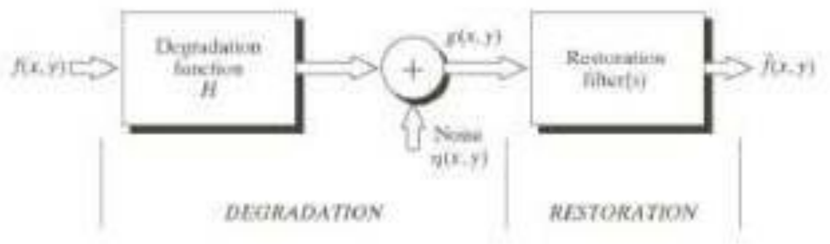



FIGURE 5.1 A model of the image degradation/restoration process.

Figure shows an image degradation/restoration process. The degraded image in the spatial domain is given by:

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

Where $h(x, y)$ is the spatial representation of the degradation function and “★” indicates convolution. Therefore, we can have the frequency domain representation by:

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

## 3.2 Noise models

The principal sources of noise in digital images arise during image acquisition and/or transmission.

### 3.2.1 Spatial and Frequency Properties of Noise

In the spatial domain, we are interested in the parameters that define the spatial characteristics of noise, and whether the noise is correlated with the image.

Frequency properties refer to the frequency content of noise in the Fourier sense.

Noise is independent of spatial coordinates and it is uncorrelated with respect to the image itself.

### 3.2.2 Some Important Noise Probability Density Functions

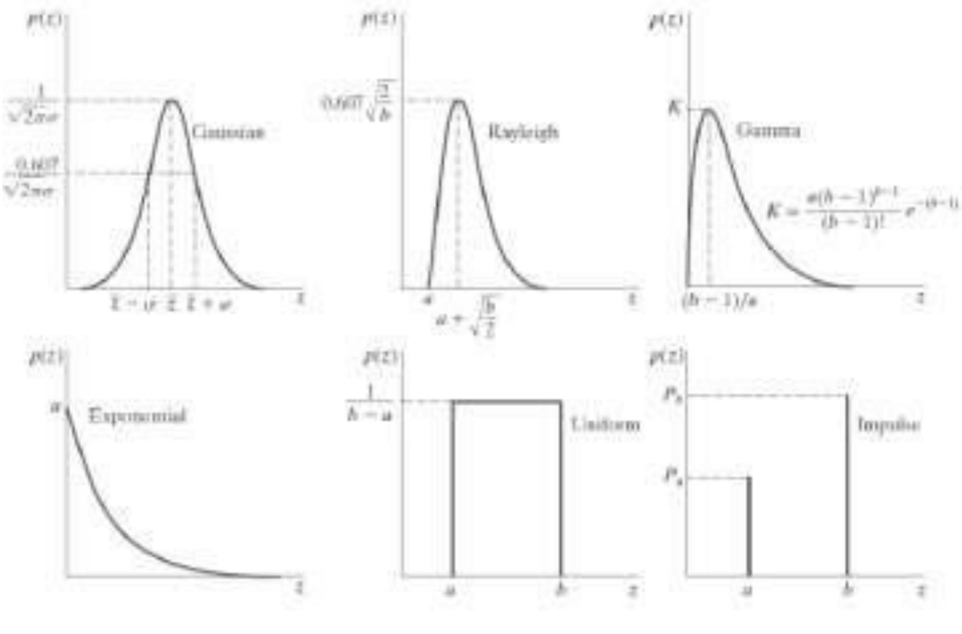
**Gaussian noise**

Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (normal) noise models are used frequently in practice.

The probability density function (PDF) of a Gaussian random variable, $z$, is given by

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

where $z$ represents intensity, is the mean (average) value of z, and $\sigma$ is its standard deviation.




**Rayleigh noise**

The probability density function of Rayleigh noise is given by

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

The mean and variance of this density are given by

$$\bar{z} = a + \sqrt{\pi b/4}$$

and

$$\sigma^2 = \frac{b(4-\pi)}{4}.$$

## Erlang (gamma) noise

The probability density function of Erlang noise is given by

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

where $a > 0$ and $b$ is a positive integer. The mean and variance of this density are given by

$$\bar{z} = \frac{b}{a}$$

and

$$\sigma^2 = \frac{b}{a^2}.$$

## Exponential noise

The PDF of exponential noise is given by

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

Where $a > 0$. The mean and variance of this density are given by

$$\bar{z} = \frac{1}{a}$$

and

$$\sigma^2 = \frac{1}{a^2}$$

## Uniform noise

The PDF of uniform noise is given by

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{for } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of this density function are given by

$$\bar{z} = \frac{a+b}{2}$$

and

$$\sigma^2 = \frac{(b-a)^2}{12}$$

## Impulse (salt-and-pepper) noise

The PDF of impulse noise is given by

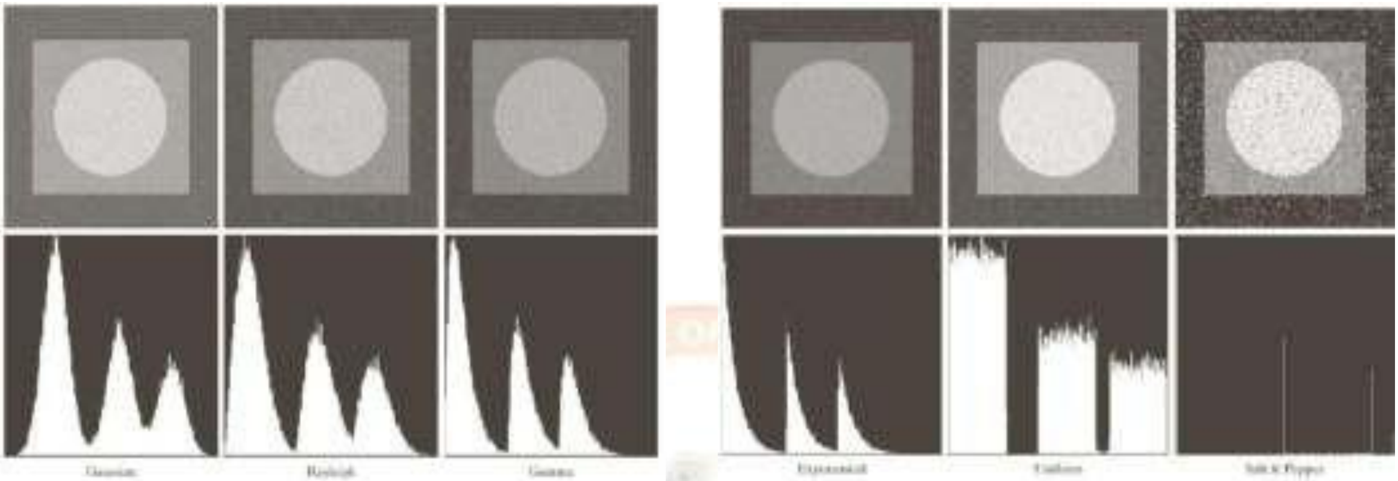
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

If $b > a$ , intensity $b$ appears as a light dot in the image. Conversely, intensity $a$ will appear like a dark dot.

If either $Pa$ or $Pb$ is zero, the impulse noise is called unipolar.

If neither $Pa$ nor $Pb$ is zero, and especially if they are approximately equal, the impulse noise values will resemble salt-and-pepper granules randomly distributed over the image.

**Example:** Noisy images and their histograms



**Periodic Noise**

Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition.

a
b
**FIGURE 5.5** (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

The periodic noise can be reduced significantly via frequency domain filtering,

## Estimation of Noise Parameters

The parameters of periodic noise can be estimated by inspection of the Fourier spectrum of the image.

Periodic noise tends to produce frequency spikes, which are detectable even by visual analysis.

In simplistic cases, it is also possible to infer the periodicity of noise components directly from the image.

Automated analysis is possible if the noise spikes are either exceptionally pronounced, or when knowledge is available about the general location of the frequency components of the interference.

It is often necessary to estimate the noise probability density functions for a particular imaging arrangement.

When images already generated by a sensor are available, it may be possible to estimate the parameters of the probability density functions from small patches of reasonably constant background intensity.

The simplest use of the data from the image strips is for calculating the mean and variance of intensity levels. Let $S$ denote a stripe and $p_s(z_i)$ , $i = 0,1,2,...,L-1$, denote the probability estimates of the intensities of the pixels in $S$ , then the mean and variance of the pixels in $S$ are

$$\bar{z} = \sum_{i=0}^{L-1} z_i p_S(z_i)$$

and

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p_S(z_i)$$

Note:

- The shape of the histogram identifies the closest probability density function match.
- The Gaussian probability density function is completely specified by these two parameters.
- For the other shapes discussed previously, we can use the mean and variance to solve the parameters $a$ and $b$.
- Impulse noise is handled differently because the estimate needed is of the actual probability of occurrence of the white and black pixels.

## 3.3 Restoration in the Presence of Noise Only using Spatial Filtering and Frequency Domain

When the only degradation present in an image is noise,

$$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y) \text{ and } G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Become

$$g(x,y) = f(x,y) + \eta(x,y) \text{ and } G(u,v) = F(u,v) + N(u,v)$$

Since the noise terms are unknown, subtracting them from $g(x,y)$ or $G(u,v)$ is not a realistic option.

In the case of periodic noise, it usually is possible to estimate $N(u, v)$ from the spectrum of $G(u, v)$ .

### 3.3.1 Mean Filters (Linear Filter)

**Arithmetic mean filter**

Let $S_{xy}$ represent the set of coordinates in a subimage window of size *mxn* , centered at $(x,y)$ . The arithmetic mean filter computes the average value of the corrupted image $g(x,y)$ in $S_{xy}$ .

The value of the restored image ˆ at point $(x,y)$ is the arithmetic mean computed in the region $S_{xy}$ :

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

**Geometric mean filter**

Using a geometric mean filter, an image is restored by

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

A geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process.

**Harmonic mean filter**

The harmonic mean filter is given by the expression

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

Works well for some types of noise like Gaussian noise and salt noise, but fails for pepper noise.

**Contraharmonic mean filter**

The contra-harmonic mean filter yields a restored image based on the expression

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

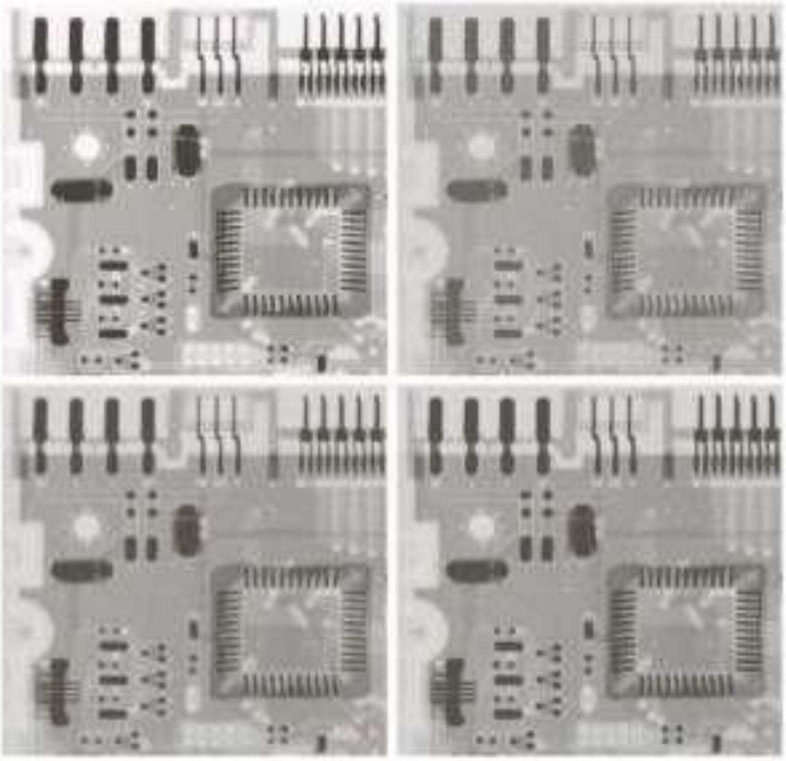
Where $Q$ is called the order of the filter.

The contraharmonic mean filter is well suited for reducing or eliminating the effects of salt-and-pepper noise.

- For positive values of $Q$, it eliminates pepper noise.
- For negative values of $Q$, it eliminates salt noise.
- When $Q = 0$, the contraharmonic mean filter reduces to the arithmetic mean filter.
- When $Q = -1$, the contraharmonic mean filter becomes the harmonic mean filter.

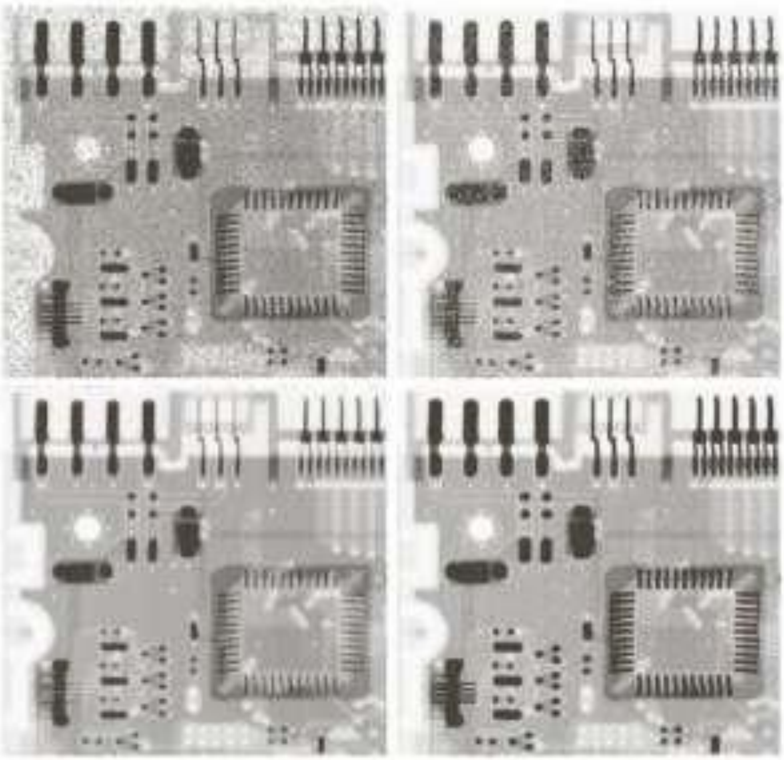
**Illustration of mean filters**

PTO



a b
c d

**FIGURE 5.7** (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size $3 \times 3$. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



a b
c d

**FIGURE 5.8** (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a $3 \times 3$ contra-harmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

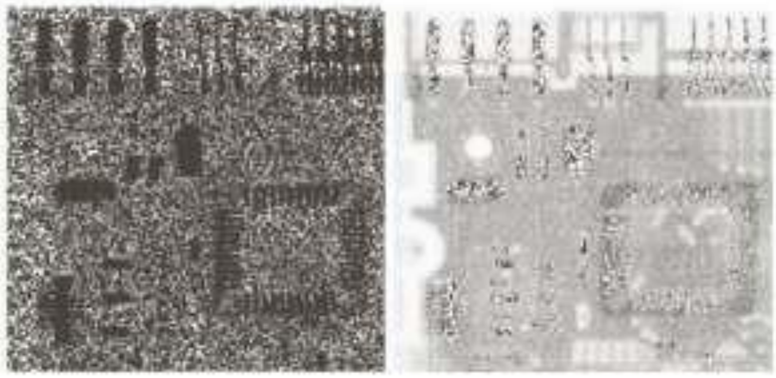
The positive-order filter did a better job of cleaning the background, at the expense of slightly thinning and blurring the dark areas.

The opposite was true of the negative-order filter.

In general, the arithmetic and geometric mean filters are suited for random noise like Gaussian or uniform noise.

The contraharmonic mean filter is well suited for impulse noise, with the disadvantage that it must be known whether the noise is dark or light in order to select *Q*.

Figure shows some results of choosing the wrong sign for *Q*



a b
**FIGURE 5.9** Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

## 3.3.2 Order-Statistic Filters (Non Linear Filter)

Order-statistic filters are spatial filters whose response is based on ordering (ranking) the values of the pixels contained in the image area encompassed by the filter.

**Median filter**

The best-known order-statistic filter is the median filter, which will replace the value of a pixel by the median of the intensity levels in the neighbourhood of that pixel:

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

The median filters are particularly effective in the presence of both bipolar and unipolar impulse noise.

**Max and min filters**

The max and min filters are defined as →

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}}\{g(s,t)\}$$

and

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\}$$

The max filter is useful for finding the brightest points in an image, while the min filter can be used for finding the darkest points in an image.

## Midpoint filter

The midpoint filter computes the midpoint between the maximum and minimum values in the area encompassed by the filter:

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

The midpoint filter works best for random distributed noise, like Gaussian or uniform noise.

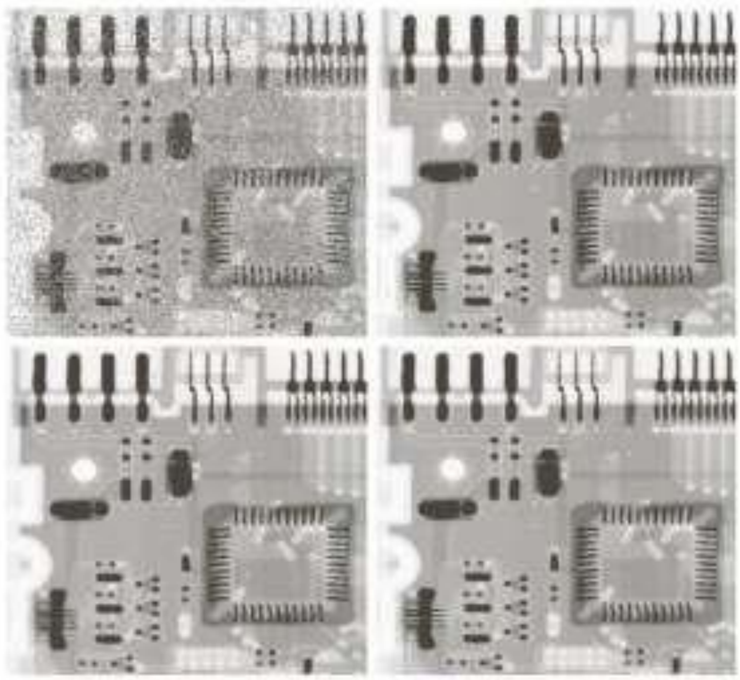
## Alpha-trimmed mean filter

Suppose that we delete the $d/2$ lowest and the $d/2$ highest intensity values of $g(s,t)$ in $S_{xy}$. Let $g_r(s,t)$ represent the remaining $mn-d$ pixels, an alpha-trimmed mean filter is given by

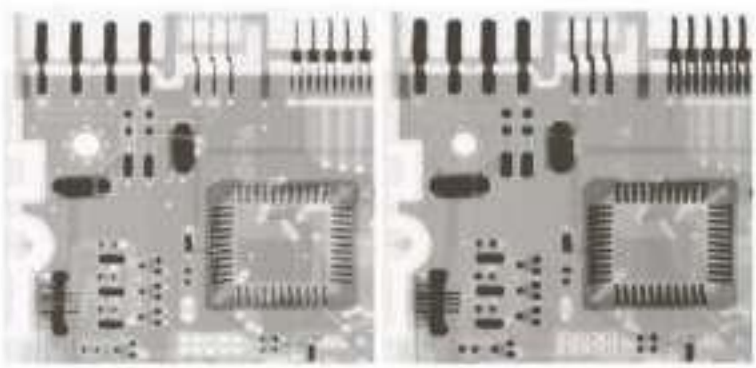
$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t)$$

- When $d = 0$, the alpha-trimmed mean filter is reduced to the arithmetic mean filter.
- If $d = mn - 1$, the alpha-trimmed mean filter becomes a median filter.

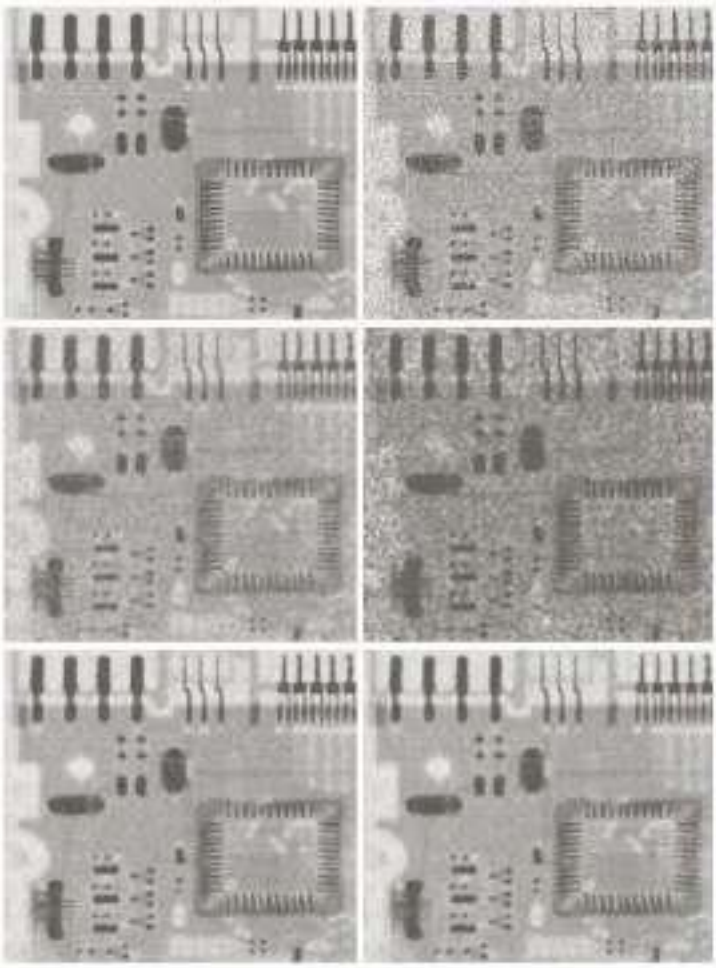
## Illustration of order-statistic filters



a b
c d
FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3 × 3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.



a b
FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.



a b
c d
e f

**FIGURE 5.12** (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image (b) filtered with a $5 \times 5$: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

### 3.3.3 Adaptive Filters

**Adaptive, local noise reduction filter**

The simplest statistical measures of a random variable are its mean and variance, which are reasonable parameters for an adaptive filter.

The mean gives a measure of average intensity in the region over which the mean is computed, and the variance gives a measure of contrast in that region.

The response of a filter, which operates on a local region $S_{xy}$, at any point $(x,y)$ is to be based on four quantities:

(a) $g(x,y)$ , the value of the noisy image at $(x,y)$ ;
(b) $\sigma_n^2$, the variance of the noise corrupting $f(x,y)$ to form $g(x,y)$ ;
(c) $m_L$, the local mean of the pixels in $S_{xy}$ ;
(d) $\sigma_L^2$ , the local variance of the pixels in $S_{xy}$ .

The following behaviours for the filter are required:

1. $\sigma_n^2$ is zero, the filter should just return the value of $g(x,y)$ . This is the zero-noise case in which $g(x,y)$ is equal to $f(x,y)$ .
2. If the local variance, $\sigma_L^2$, is high relative to $\sigma_n^2$, the filter should return a value close to $g(x,y)$ .

A high local variance typically is associated with edges, which should be preserved.

3. If the two variances are equal, we want the filter to return the arithmetic mean value of the pixels in $S_{xy}$ .

This condition occurs when the local area has the same properties as the overall image, and local noise is to be reduced simply by averaging.

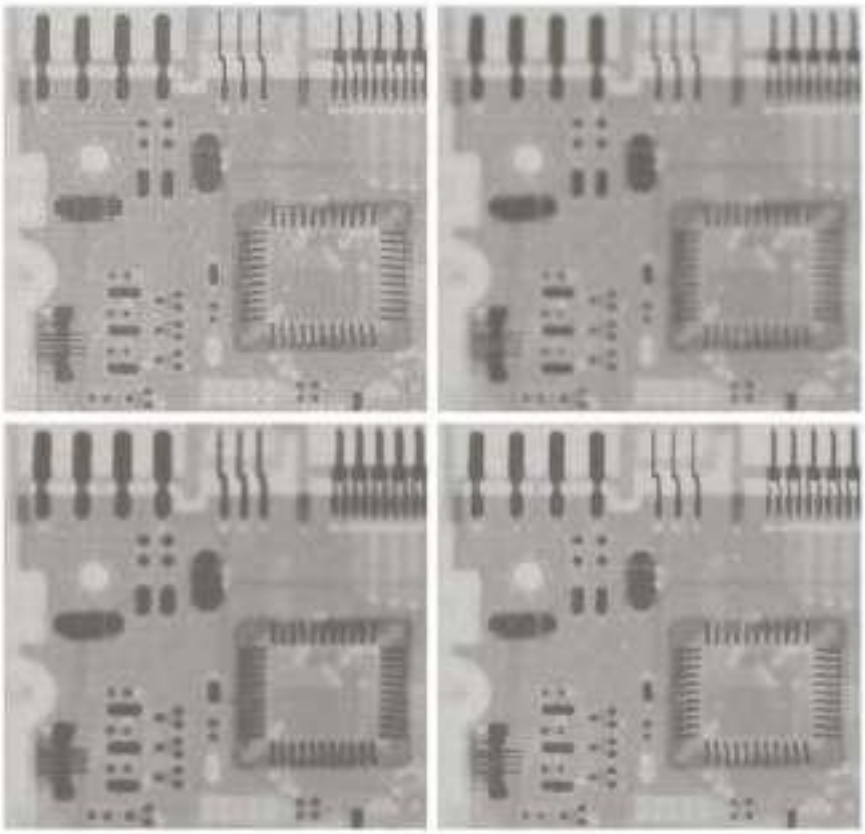
Based on these assumptions, an adaptive expression for obtaining ˆ (x, y) may be written as

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

The only quantity needed to be estimated is the variance of the overall noise, $\sigma_n^2$, and other parameters can be computed from the pixels in $S_{xy}$ .

a b
c d

**FIGURE 5.13** (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.



## Adaptive median filter

The median filter discussed previously performs well if the spatial density of the impulse noise is not large ($Pa$ and $Pb$ are less than 0.2 ).

The adaptive median filtering can handle impulse noise with probabilities larger than these. Unlike other filters, the adaptive median filter changes the size of $S_{xy}$ during operation, depending on certain conditions.

Let:

$z_{min}$ = minimum intensity value in $S_{xy}$
$z_{max}$ = maximum intensity value in $S_{xy}$
$z_{med}$ = median of intensity values in $S_{xy}$
$z_{xy}$ = intensity value at coordinates $(x,y)$
$S_{max}$ = maximum allowed size of $S_{xy}$

The adaptive median filtering algorithm works in two stages:

Stage A: $A1 = z_{med} - z_{min}$

$A2 = z_{med} - z_{max}$

If $A1 > 0$ AND $A2 < 0$, go to stage B

Else increase the window size

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

Else output $z_{med}$

Stage B: $B1 = z_{xy} - z_{min}$

$B2 = z_{xy} - z_{max}$

If $B1 > 0$ AND $B2 < 0$, output $z_{xy}$

Else output $z_{med}$

Purpose:

- To remove salt-and-pepper (impulse) noise;
- To provide smoothing of other noise that may not be impulsive; and
- To reduce the distortion of object boundaries.

a b c

**FIGURE 5.14** (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a $7 \times 7$ median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

## 3.4 Periodic Noise Reduction by Frequency Domain Filtering

Periodic noise can be analyzed and filtered effectively by using frequency domain techniques.

**Bandreject Filters**

Rejects (attenuates) band of frequencies and allows the rest

Figure shows perspective plots of ideal, Butterworth, and Gaussian bandreject filters,

a b c

**FIGURE 5.15** From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Bandreject filters. $W$ is the width of the band, $D$ is the distance $D(u, v)$ from the center of the filter, $D_0$ is the cutoff frequency, and $n$ is the order of the Butterworth filter. We show $D$ instead of $D(u, v)$ to simplify the notation in the table.

| Ideal | Butterworth | Gaussian |
|---|---|---|
| $H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$ | $H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$ | $H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$ |

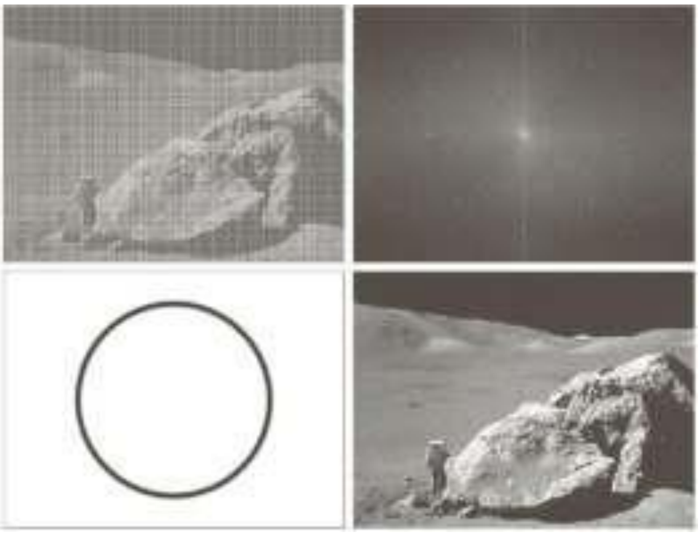
One of the principal applications of bandreject filtering is for noise removal in applications where the general location of the noise component(s) in the frequency domain is approximately known.

## Illustration:

The noise components can be seen as symmetric pairs of bright dots in the Fourier spectrum shown in Figure (b)

a b
c d

**FIGURE 5.16** (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)



Since the component lie on an approximate circle about the origin of the transform, so a circularly symmetric bandreject filter is a good choice.

### Bandpass Filters

A bandpass filter performs the opposite operation of a bandreject filter.
The transfer function $H_{BP}(u,v)$ of a bandpass filter is obtained from a corresponding bandreject filter transfer function $H_{BR}(u,v)$ by using the equation.

$$H_{BP}(u,v) = 1 - H_{BR}(u,v).$$

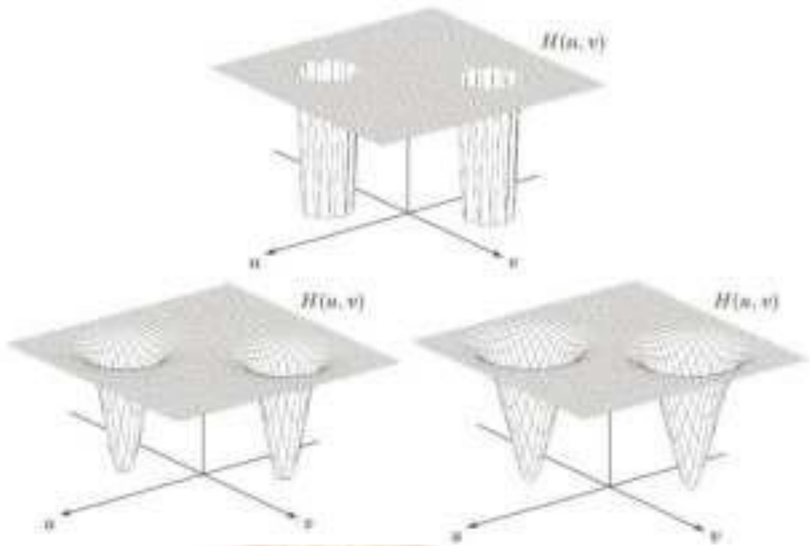
Performing straight bandpass filtering on an image is not a common procedure because it generally removes too much image detail. However, bandpass filtering is useful in isolating the effects on an image caused by selected frequency bands.

## Notch Filters

A notch filter rejects/passes frequencies in predefined neighbourhoods about a center frequency. Figure shows plots of ideal, Butterworth, and Gaussian notch (reject) filters.

Restoration is done by Placing the notch filter at the location of spike (noise).

a
b c
**FIGURE 5.18** Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

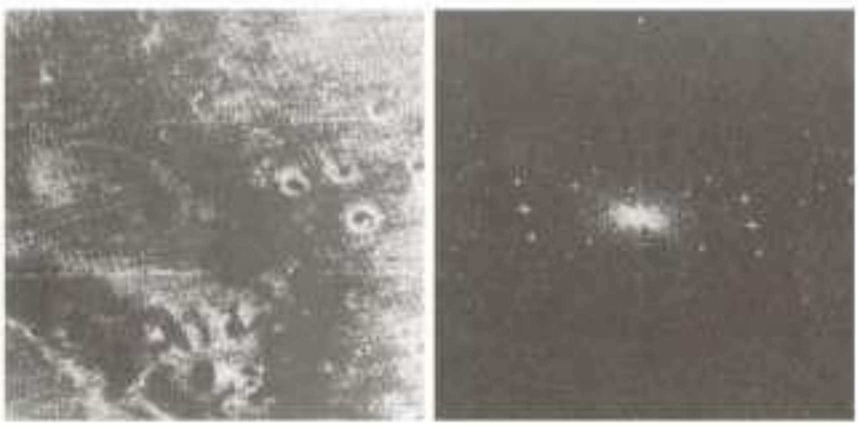



The transfer function $H_{NP}(u,v)$ of a notch pass filter is obtained from a corresponding notch reject filter transfer function, $H_{NR}(u,v)$ , by using the equation

$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$

## Optimum Notch Filtering

a b
**FIGURE 5.20** (a) Image of the Martian terrain taken by *Mariner 6*. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)



When several interference components are present, the methods discussed previously are not always acceptable because they may remove too much image information in the filtering process.

Optimum notch filters minimizes local variances of the restored estimate ˆ (x, y)

**Step 1:** The first step in Optimum notch filtering is to *find* the *principal frequency* components and ***placing notch pass filters* at the location of *each spike in* G(u,v),** yielding ***H(u, v).*** The Fourier transform of the interference pattern is thus given by

$$N(u, v) = H_{NP}(u, v)\, G(u, v)$$

where G(u, v)is the Fourier transform of the corrupted image.

**Step 2:** The corresponding interference pattern in the spatial domain is obtained with the inverse Fourier transform

$$\eta(x, y) = \mathfrak{F}^{-1}\{H_{NP}(u, v) G(u, v)\}$$

**Step3:** f(x, y) can be obtained $f(x,y) = g(x,y) - \eta(x,y)$

Where, g(x, y) is corrupted image

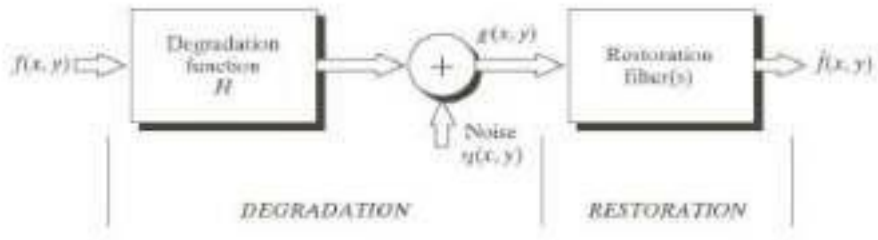
**Step 4:** The effect of components not present in the estimate of $\eta(x,y)$ can be minimized by subtracting from $g(x,y)$ a weighted portion of $\eta(x,y)$ to obtain an estimate of $f(x,y)$ :

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y)$$

**Note:** Weighting function can be chosen according to need; one approach minimizes the local variance

## 3.5 Linear, Position- Invariant Degradations. [Important]

FIGURE 5.1 A model of the image degradation/ restoration process.




The input-output relationship shown in Figure before the restoration can be expressed as

$$\mathbf{g(x, y) = H[f(x, y)] + \eta(x, y)} \qquad \mathbf{(1)}$$

First, we assume that $\eta(x, y) = 0$ so that,

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

**Linearity Property:**
***H*** is Linear if:

$$\mathbf{H[af_1(x, y) + bf_2(x, y)] = a\,H[f_1(x, y)] + b\,H[f_2(x, y)]} \qquad \mathbf{(2)}$$

Where $a$ and $b$ are scalars and $f_1(x, y)$ and $f_2(x, y)$ are any two input images. If $a = b = 1$, then equation (2) becomes

$$\mathbf{H[af_1(x, y) + bf_2(x, y)] = H[f_1(x, y)] + H[f_2(x, y)]} \qquad \mathbf{(3)}$$

Which is called the property of additivity.

If $f_2(x, y) = 0$ , equation (2) becomes

$$\mathbf{H}\,[a f_1(x, y) + b f_2(x, y)] = a\,\mathbf{H}\,[f_1(x, y)] \quad \textbf{(4)}$$

This is called the property of homogeneity. It says that the response to a constant multiple of any input is equal to the response to that input multiplied by the same constant.

**Position- Invariant:**

An operator having the input-output relationship

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

is said to be position (space) invariant if

$$H\,[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta) \quad \textbf{(5)}$$

For any $f(x,y)$ and any $\alpha$ & $\beta$ . Eq (5) indicates that the response at any point in the image depends only on the value of the input at that point, not on its position.

With a slight change in notation in the definition of the impulse in -

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t-t_0, z-z_0)\,dt\,dz = f(t_0, z_0)$$

$f(x, y)$ can be expressed as

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta \quad \textbf{(6)}$$

Assuming $\eta(x, y) = 0$ , then substituting (6) into (1) we have:

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

$$= H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta\right] \quad \textbf{(7)}$$

If $H$ is a linear operator, then

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H\left[f(\alpha,\beta)\delta(x-\alpha, y-\beta)\right]d\alpha\,d\beta \quad \textbf{(8)}$$

Since $f(\alpha, \beta)$ is independent of $x$ and $y$, using the homogeneity property, it follows that

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta) H\left[\delta(x-\alpha, y-\beta)\right]d\alpha\,d\beta$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta) h(x,\alpha,y,\beta)\,d\alpha\,d\beta$$

Where the term → $h(x, \alpha, y, \beta) = H\,[\delta(x-\alpha, y-\beta)]$ is called the impulse response of $H$.

In other words, if $\eta(x, y) = 0$, then $h(x, \alpha, y, \beta)$ is the response of $H$ to an impulse at $(x, y)$.

Equation

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta)d\alpha d\beta \qquad \textbf{(9)}$$

is called the superposition (or Fredholm) integral of the first kind, and is a fundamental result at the core of linear system theory.

If $H$ is position invariant, from

$$H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta)$$

$$\downarrow$$

We have

$$H[\delta(x - \alpha, y - \beta)] = h(x - \alpha, y - \beta)$$

Equation (9) reduces to

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta)d\alpha d\beta$$

Above Equation tells us that knowing the impulse of a linear system allows us to compute its response, g , to any input $f$ . The result is simply the convolution of the impulse response and the input function.

In the presence of additive noise, Equation (9) becomes,

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta)d\alpha d\beta + \eta(x, y)$$

If $H$ is position invariant, it becomes

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta)d\alpha d\beta + \eta(x, y)$$

Assuming that the values of the random noise $\eta\ (x,y)$ are independent of position, we have

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

Based on the convolution theorem, we can express above equation in the frequency domain as

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

A linear, spatially invariant degradation system with additive noise can be modelled in the spatial domain as the convolution of the degradation function with an image, followed by the additive of noise. The same process can be expressed in the frequency domain.

## 3.6 Estimating the Degradation Function

There are three principal ways to estimate the degradation function used in image restoration.

- Estimation by Image Observation
- Estimation by Experimentation
- Estimation by Modeling

### 3.6.1 Estimation by Image Observation

Suppose that we are given a degraded image without any knowledge about the degradation function *H*.

Based on the assumption that the image was degraded by a linear, position-invariant process, one way to estimate *H* is to gather information from the image itself.

In order to reduce the effect of noise, we would look for an area in which the signal content is strong.

Let the observed sub-image be denoted by $g_s(x, y)$, and the processed sub-image be denoted by ˆ ············Assuming that the effect of noise is negligible, it follows from

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

That

$$H_s(u, v) = \frac{G_s(u, v)}{\hat{F}_s(u, v)}$$

Then, we can have $H(u, v)$ based on our assumption of position invariant.

**For example,** suppose that a radial plot of $H_s(u, v)$ has the approximate shape of a Gaussian curve. Then we can construct a function $H(u, v)$ on a large scale, but having the same basic shape. This estimation is a **laborious process** used in very specific circumstances.

### 3.6.2 Estimation by Experimentation

If equipment similar to the equipment used to acquire the degraded image is available, it is possible in principle to obtain an accurate estimate of the degradation.

Images similar to the degraded image can be acquired with various system settings until they are degraded as closely as possible to the image we wish to restore.

Then the idea is to obtain the impulse response on the degradation by imaging an impulse (small dot of light) using the same system settings.

An impulse is simulated by a bright dot of light, as bright as possible to reduce the effect of noise to negligible values. Since the Fourier transform of an impulse is a constant, it follows

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

a b
FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

### 3.6.3 Estimation by Modeling

Degradation modeling has been used for years. In some cases, the model can even take into account environmental conditions that cause degradations.

For example, a degradation model proposed by Hufnagel and Stanley is based on the physical characteristics of atmospheric turbulence

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

Where $k$ is a constant that depends on the nature of the turbulence. Figure shows examples of using Equation with different values of $k$.

a b
c d

**FIGURE 5.25** Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

A major approach in modeling is to derive a mathematical model starting from basic principles. We show this procedure by a case in which an image has been blurred by uniform linear motion between the image and the sensor during image acquisition. Suppose that an image $f(x, y)$ undergoes planar motion and that $x_0(t)$ and $y_0(t)$ are the time-varying components of motion in the $x$ – and $y$- directions.

The total exposure at any point of the recording medium is obtained by integrating the instantaneous exposure over the time interval when the imaging system shutter is open. If the $T$ is the duration of the exposure, the blurred image $g(x, y)$ is

$$g(x,y) = \int_0^T f[x - x_0(t), y - y_0(t)]\,dt$$

The Fourier transform of g(x, y) is given by:

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f[x - x_0(t), y - y_0(t)]\,dt\right] e^{-j2\pi(ux+vy)}\,dx\,dy$$

By reversing the order of integration,

$$G(u,v) = \int_0^T \left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)]\, e^{-j2\pi(ux+vy)}\,dx\,dy\right] dt$$

Since the term inside the outer brackets is the Fourier transform of the displaced function $f[x - x_0(t), y - y_0(t)]$, we have

$$G(u,v) = \int_0^T F(u,v) e^{-j2\pi[ux_0(t)+vy_0(t)]}\,dt$$
$$= F(u,v)\int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\,dt$$

By defining

$$H(u,v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\,dt$$

We can rewrite

$$G(u, v) = H(u, v)\, F(u, v)$$

## 3.7 Inverse Filtering

The simplest approach to restoration is direct inverse filtering, where we compute an estimate, ˆ , of the transform of the original image by:

W.K.T $G(u, v) = H(u, v)\, F(u, v)$

Therefore,

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \qquad \textbf{(1)}$$

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

Equation (1) becomes

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \qquad \textbf{(2)}$$

**Case 1**: We cannot recover the undegraded image exactly because $N(u,v)$ is not known.
**Case 2**: If the degradation function $H(u,v)$ has zero or very small values, so the second term of Eq (2) could easily dominate the estimate of ˆ .

One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin. As we know that $H(0,0)$ is usually the highest value of $H(u, v)$ in the frequency domain.

## 3.8 Minimum Mean Square Error (Wiener) Filtering

- It is Better than the inverse filter
- It incorporates both the degradation function and statistical characteristics of noise into the restoration process.
- Consider image and noise as random variables
- The objective is to find an estimate $\hat{}$ of the uncorrupted image $f$ such that the mean square error between them is minimized.

The error measure is given by

$$e^2 = E\left\{(f - \hat{f})^2\right\} \quad \textbf{(1)}$$

Where $E\{\ .\ \}$ is the expected value of the argument.

Assumptions:
1. The noise and the image are uncorrelated.
2. One or the other has zero mean.
3. The intensity levels in the estimate are a linear function of the levels in the degraded image.

Based on the above assumptions and with expectation of the minimum of the error function as in equation (1), can be obtained in the frequency domain by the expression

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

$\hat{F}(u,v)$ is the frequency domain estimate

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

$H(u,v)$ is the transform of the degradation function

$H^*(u,v)$ is complex conjugate of $H(u,v)$

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

$S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise

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

Simplifying above equation

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

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

Where ***K = $S_n$(u, v)/$S_f$(u, v)***
Value of ―K‖ usually not known and will be added to H(u, v) term on trial and error

Note: The value of *K* was chosen interactively to yield the best visual result.

Note: If the noise or K value is zero, then the Wiener filter reduces to the inverse filter.

**Measure of Restored image:**
Important measures is the signal-to-noise ratio, approximated using frequency domain quantities such as

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

The mean square error given in statistical form

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

If one considers the restored image to be signal and the difference between this image and the original to be noise, we can define a signal-to-noise ratio in the spatial domain as

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

## 3.9 Constrained Least Squares Filtering.

Bases optimality of restoration on a measure of smoothness. Seek minimum of a criterion function

$$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\nabla^2 f(x, y)]^2 \quad \text{(second derivative)}$$

From the degradation system we get

$$\begin{aligned} g(x, y) &= h(x, y) * f(x, y) + \eta(x, y) \\ \mathbf{g} &= \mathbf{Hf} + \eta \quad \text{(vector-matrix form)} \end{aligned}$$

The criterion function is subject to the constraint

$$||\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}||^2 = ||\eta||$$

where $\hat{\mathbf{f}}$ is an estimate of the undegraded image. The frequency domain solution thus becomes

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

where $\gamma$ is the parameter to be adjusted ($\gamma = 0 \Rightarrow$ inverse filtering), and $P(u, v)$ is the fourier transform of the function

$$p(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{(laplacian)}$$

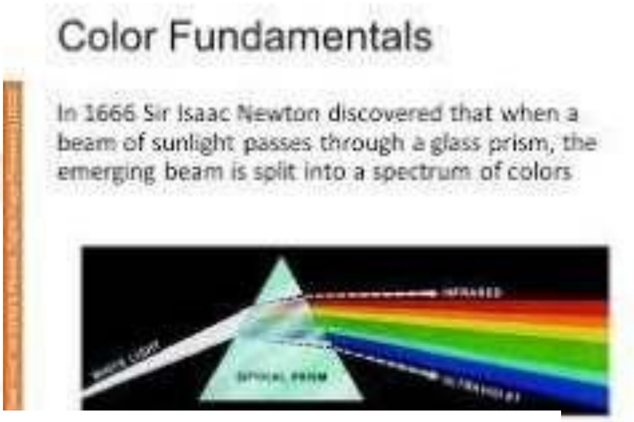
Either adjust $\gamma$ interactively for acceptable results, or use mean and variance of noise to iteratively adjust $\gamma$ (to satisfy criterion function constraint).

| Module – 4 | |
|---|---|
| **Color Image Processing:** Color Fundamentals, Color Models, Pseudocolor Image Processing.<br>**Wavelets:** Background, Multi-resolution Expansions.<br>**Morphological Image Processing:** Preliminaries, Erosion and Dilation, Opening and Closing, The Hit-or-Miss Transforms, Some Basic Morphological Algorithms.<br>**[Text: Chapter 6: Sections 6.1 to 6.3, Chapter 7: Sections 7.1 and 7.2, Chapter 9: Sections 9.1 to 9.5]** | L1, L2, L3 |

### 4.1 Color Image Processing:

#### 4.1.1 Color Fundamentals

- The color that humans perceived in an object are determined by the nature of the light reflected from the object.
- Light is electromagnetic spectrum.



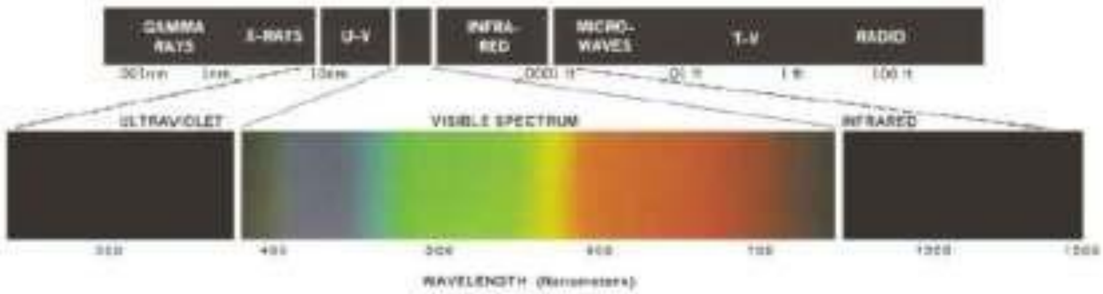




**FIGURE 6.2** Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

**Visible light and Color:**

- Visible light is composed of a relatively narrow band of frequencies in the ES.
- Human color perceive is a composition of different wavelength spectrum
- The color of an object is a body that favours reflectance in a limited range of visible spectrum exhibits some shade of colors
- Example
  - White: a body that reflects light that balanced in all visible wavelengths
  - E.g. green objects reflect light with wavelength primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelengths.

**Characterization of light:**

- If the light achromatic (void of color), if its only attribute is intensity.
- Gray level refers to a scalar measure of intensity that ranges from black, to grays, and finally to white
- Chromatic light spans the ES from about 400 to 700 nm
- Three basic quantities are used to describe the quality of a chromatic light source
    - Radiance: total amount of energy flows from the light source
    - Luminance: amount of energy perceive from light source
    - Brightness: a subjective descriptor that is practically impossible to measure

**Color sensors of eyes: cones**

- Primary colors for standardization
    - blue : 435.8 nm, green : 546.1 nm, red : 700 nm
- Not all visible colors can be produced by mixing these three primaries in various intensity proportions
- Cones in human eyes are divided into three sensing categories
    - 65% are sensitive to red light, 33% sensitive to green light, 2% sensitive to blue (but most sensitive)
    - The R, G, and B colors perceived by human eyes cover a range of spectrum

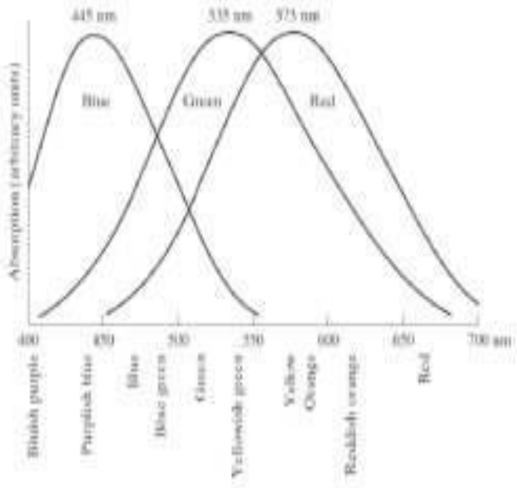
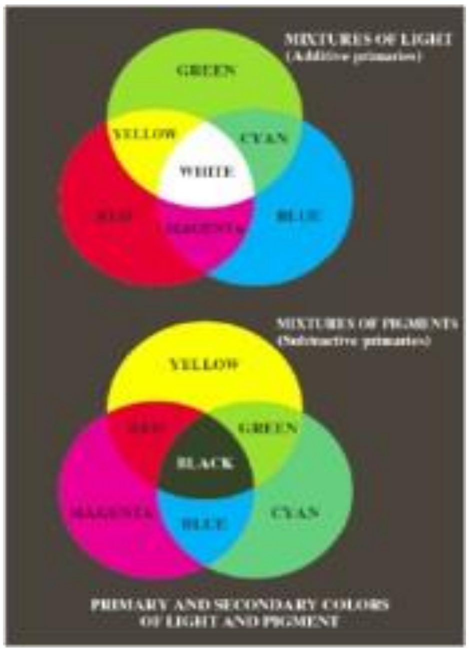



**FIGURE 6.3** Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

**Primary colors and secondary colors:**

- CIE (Commission Internationale de l'Eclariage) standard for primary color
    - Red: 700 nm
    - Green: 546.1 nm
    - Blue: 435.8 nm
- Primary color can be added to produce secondary colors
    - Primary colors can not produce all colors
- Pigments (colorants)
    - Define the primary colors to be the absorbing one and reflect other two

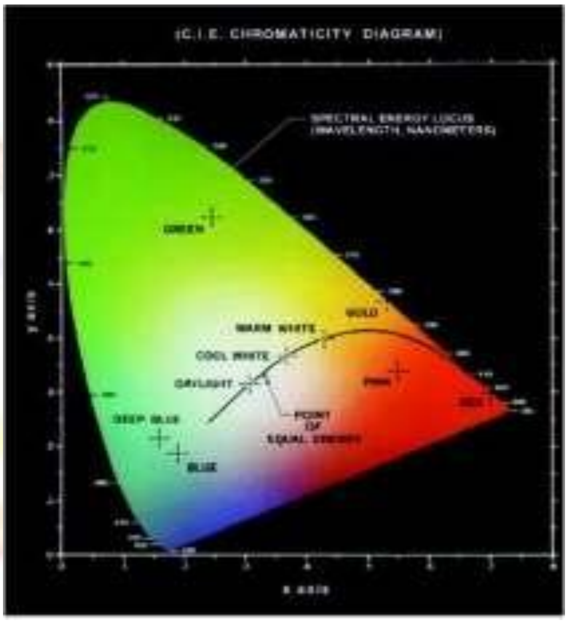


**Chromaticity:**

- Hue + saturation = chromaticity
  - Hue: an attribute associated with the dominant wavelength or dominant colors perceived by an observer
  - Saturation: relative purity or the amount of white light mixed with a hue (the degree of saturation is inversely proportional to the amount of added white light)
- Color = brightness + chromaticity
- Tristimulus values (the amount of R, G, and B needed to form any particular color : X, Y, Z
  - Trichromatic coefficients :

$$x = X/(X+Y+Z) \qquad y = Y/(X+Y+Z) \qquad z = Z/(X+Y+Z)$$

**Chromaticity diagram**




- Show color composition as a function of *x*, *y*, and *z*
- Spectrum colors are indicated around the boundary of the tongue-shaped chromaticity diagram
- Point of equal energy : equal fractions of three primary colors →CIE defined white light
- Points located on the boundary of chromaticity diagram are fully saturated -- the saturation at the center point is zero

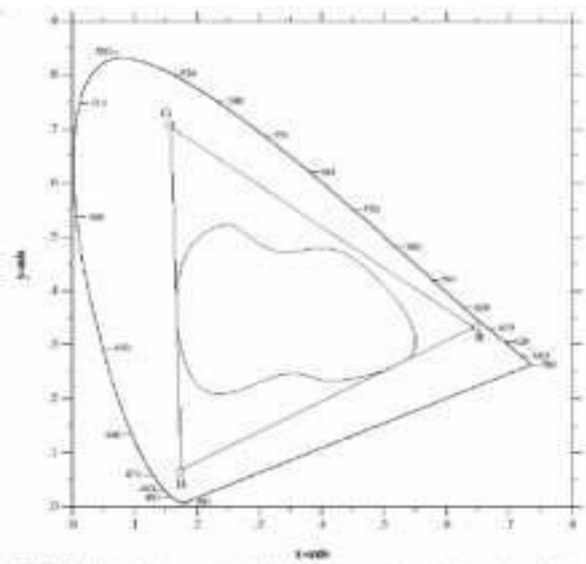


FIGURE 6.6 Typical color gamut of color monitors (triangle) and color printing devices (irregular region).

- A straight line segment joining any two points defines all color variations of the combination of them
- No three colors in the diagram can span the whole color space --not all colors can be obtained with three single and fixed primaries
- The color gamut produced by RGB monitors ⇒ Triangle
- The color printing gamut is irregularly-shaped ⇒ Irregular Region

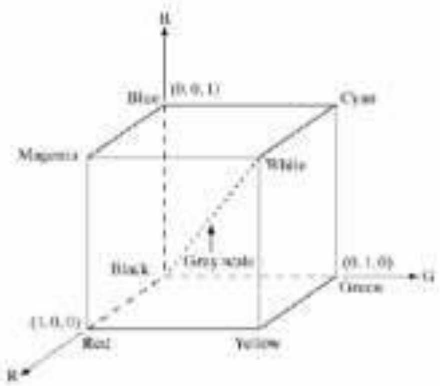
### 4.1.2 Color Models

- A color model is a specification of a coordinate system within which each color is represented by a single point
- Hardware-oriented color models,
    - e.g., color monitors and printers
    - RGB, CMY (cyan, magenta, yellow), CMYK (+black)
- Application-oriented color model
    - HSI (hue, saturation, intensity)

**RGB Color Model:**

- Each color appears in its primary spectral components of R, G, and B
- Based on a Cartesian coordinate system (cube)
- (R, G, B): all values of R, G, B are between 0 and 1.
- With digital representation, for a fixed length of bits each color element. The total number of bits is called color depth, or pixel depth.

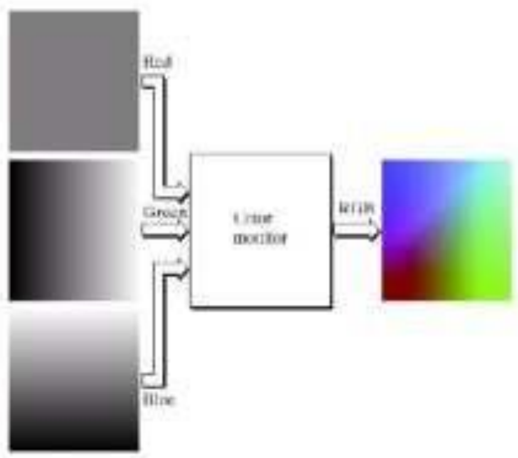
**FIGURE 6.7** Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).



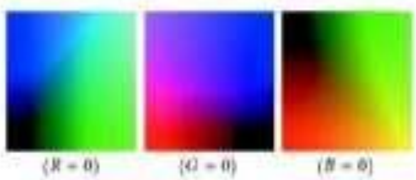

- For example, 24-bit RGB color (r, g, b), 8-bits for each color. The 8-bit binary number r represents the value of r/256 in [0,1]

a
b

**FIGURE 6.9** (a) Generating the RGB image of the cross-sectional color plane (127, G, B). (b) The three hidden surface planes in the color cube of Fig. 6.8.

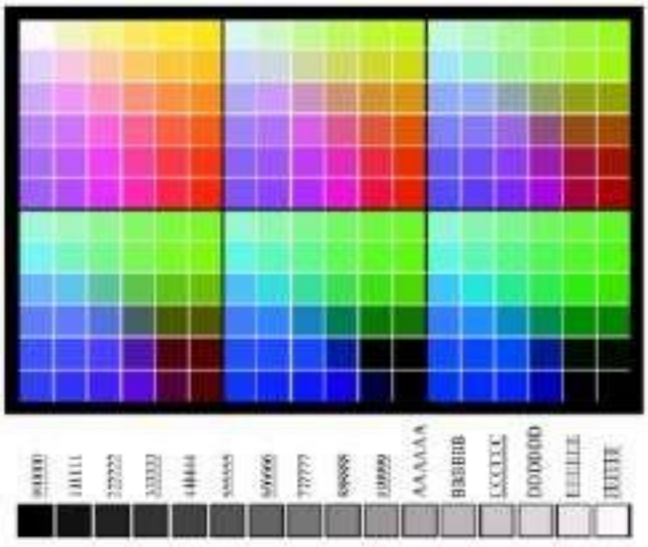



Displaying Colors in RGB model




| Number System | Color Equivalents | | | | | |
|---|---|---|---|---|---|---|
| Hex | 00 | 33 | 66 | 99 | CC | FF |
| Decimal | 0 | 51 | 102 | 153 | 204 | 255 |

**TABLE 6.1** Valid values of each RGB component in a safe color.



a b
**FIGURE 6.10** (a) The 216 safe RGB colors. (b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined).

**CMY and CMYK color Models:**

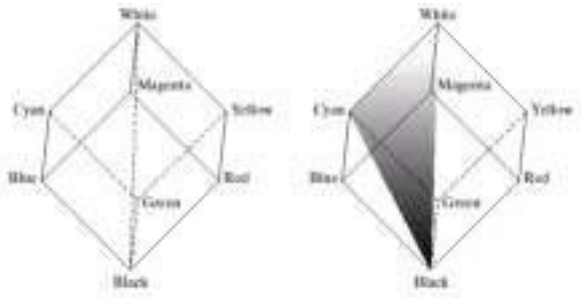
- Useful in color printers and copiers
- Conversion between RGB and CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

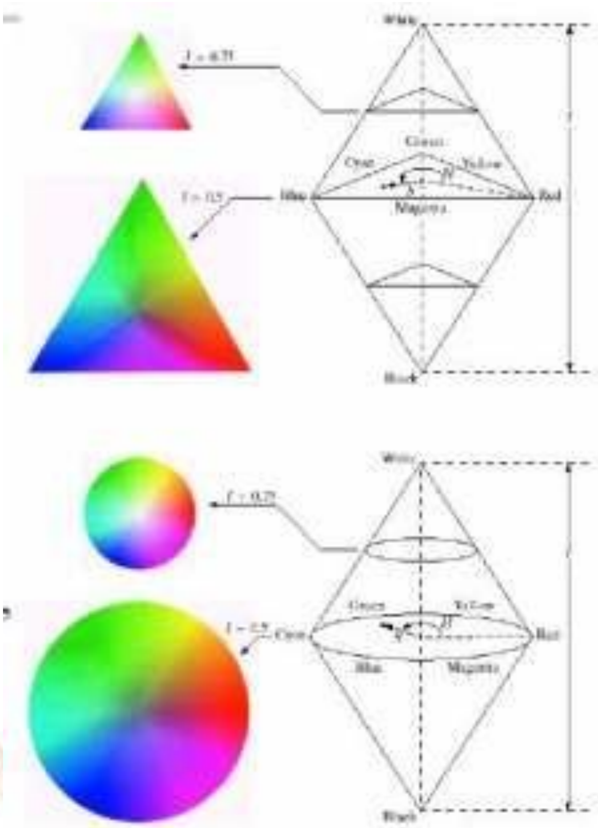
- In practice, combining CMY colors produces a muddy-looking black. To produce true black, a forth color, black, is added ⇒ CMYK color model
- CMYK: (C, M, Y, B), where B is a fixed black color. This basically for printing purpose, where black is usually the dominating color. When printing black, using B rather than using (C, M, B) = (1, 1, 1)

**HSI color model:**

- RGB, CMY, and similar others are not practical for human interpretation
- Hue : a color attribute that describes a pure color
- Saturation : a measure of the degree to which a pure color is diluted by white light
- Derivation of HSI from RGB color cube
    - o All points contained in the plane segment defined by the intensity axis (i.e., from black to white) and one color point on the boundaries of the cube have the same hue




a b
**FIGURE 6.12** Conceptual relationships between the RGB and HSI color models.




- The HSI space is represented by a vertical intensity axis, the length (saturation) of a vector from the axis to a color point, and the angle (hue) this vector makes with the red axis
- The power of HSI color model is to allow independent control over hue, saturation, and intensity

**Conversion between RGB and HSI:**

- From RGB to HSI

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases} \qquad \theta = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\}$$

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

- From HSI to RGB
  - RG sector ($0^\circ$<H<$120^\circ$)

$$B = I(1-S) \qquad R = I[1 + \frac{S\cos H}{\cos(60^\circ - H)}]$$

$$G = 3I - (R+B)$$

  - GB sector ($120^\circ$<H<$240^\circ$)

$$H = H - 120^\circ \qquad R = I(1-S)$$

$$G = I[1 + \frac{S\cos H}{\cos(60^\circ - H)}] \quad B = 3I - (R+G)$$

  - BR sector ($240^\circ$<H<$360^\circ$)

$$H = H - 240^\circ \qquad G = I(1-S)$$

$$B = I[1 + \frac{S\cos H}{\cos(60^\circ - H)}] \quad R = 3I - (G+B)$$

**4.1.3** Pseudocolor Image Processing.

Pseudo color – also called False Color

Pseudocolor image processing is to assign colors to gray values based on a specified criterion.

Purpose: human visualization, and interpretation for gray-scale events

Two types:

1. Intensity Slicing
2. Intensity to color transformation

**Intensity Slicing:**

If an image is interpreted as a 3D function, the method can be viewed as one of placing planes parallel to the coordinate plane of the image. Each plane then slices the function in the area of intersection.

Fig. shows a plane at $f(x, y)=l_i$ to slice the image function into two levels. If a different color is assigned to each side of the plane shown in Fig., 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.

The result is a two color image whose relative appearance can be controlled by moving the slicing plane up and down the gray level axis. The idea of planes is useful primarily for a geometric interpretation of the intensity-slicing technique. When more levels are used, the mapping function takes on a staircase form.

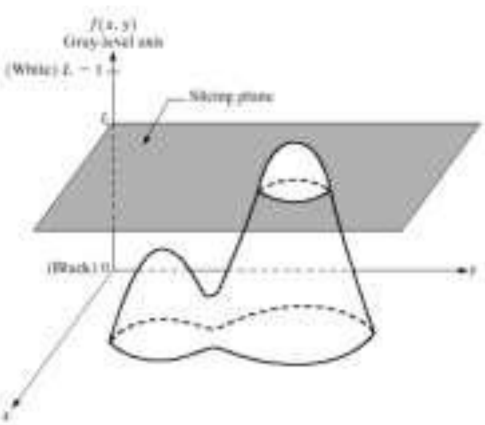



FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

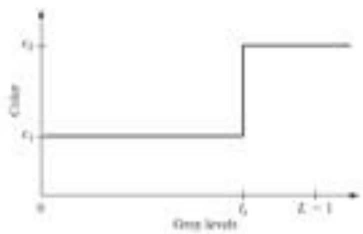
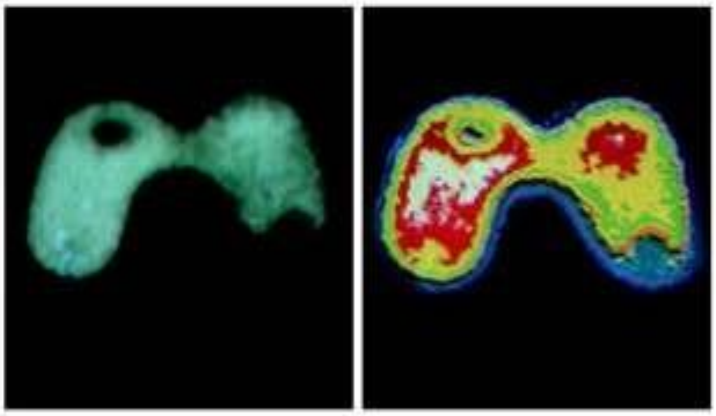


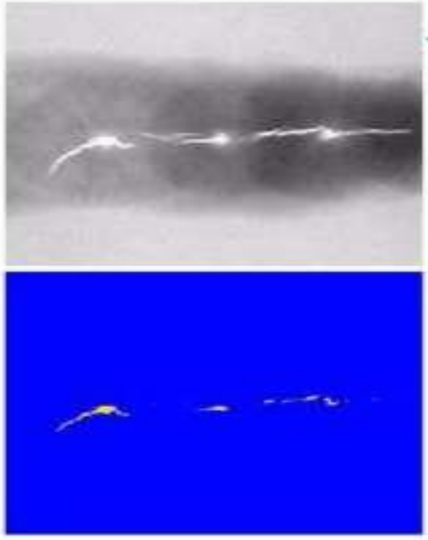

FIGURE 6.19 An alternative representation of the intensity-slicing technique.

Intensity slicing: partition the gray-scale into P+1 intervals, $V_1, V_2, \ldots, V_{p+1}$ . Let $f(x, y) = c_k$ , if $f(x,y)$ is in $V_k$ where $c_k$ is the color assigned to level $k$



a b

**FIGURE 6.20** (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)



Technique summary

- Gray scale representation – $[0, L-1]$
- Black represented by $l_0$, $[f(x,y) = 0]$
- White represented by $[l_{L-1}]$, $[f(x,y) = L-1]$
- Define $P$ planes perpendicular to intensity axis at levels $l_1, l_2, \ldots, l_P$
- $0 < P < L-1$
- $P$ planes partition the gray scale into $P+1$ intervals as $V_1, V_2, \ldots, V_{P+1}$
- Make gray-level to color assignment as

$$f(x,y) = c_k \qquad \text{if } f(x,y) \in V_k$$

where $c_k$ is the color associated with $k$th intensity interval $V_k$ defined by partitioning planes at $l = k-1$ and $l = k$

## Intensity to color transformation

Transform intensity function f(x, y) into three color component

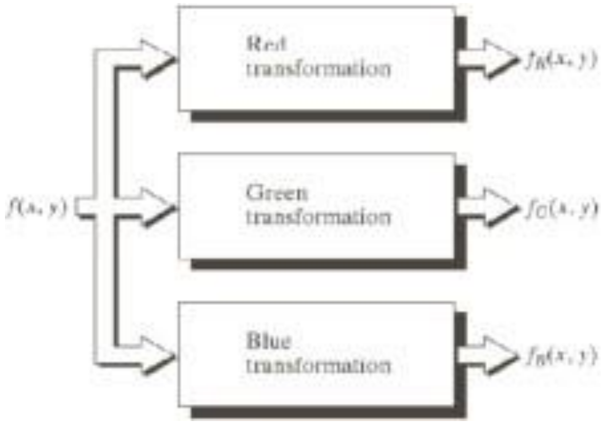



**FIGURE 6.23** Functional block diagram for pseudocolor image processing. $f_R$, $f_G$, and $f_B$ are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Basically the idea underlying this approach is 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 method 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 functions of position.

Ex.

Let f(x, y) be an given gray scale image

$f_r(x, y) = f(x, y)$

$f_g(x, y) = 0.33*f(x, y)$

$f_b(x, y) = 0.12*f(x, y)$

**Transformed color image = $f_r+f_g+f_b$**

# Morphological Image Processing

## 1.1 Preliminaries:

- ***"Morphology "*** – a branch in biology that deals with the form and structure of animals and plants.
- **"*Mathematical Morphology*"** – as a tool for extracting image components, that are useful in the representation and description of region shape.
- The language of mathematical morphology is – **Set theory**.
- Unified and powerful approach to numerous image processing problems.
- In binary images , the set elements are members of the 2-D integer space – $Z^2$. where each element *(x, y)* is a coordinate of a black (or white) pixel in the image.

Used to extract image components that are useful in the representation and description of region shape, such as

- Boundaries extraction
- Skeletons
- Convex hull
- Morphological filtering
- Thinning
- Pruning

**Basic Concepts in Set Theory**

- Subset: $A \subseteq B$

  Means every element in set A is also in set B.
- Union: $A \cup B$
- Intersection: $A \cap B$
  - disjoint / mutually exclusive: $A \cap B = \phi$
- Complement:

$$A^c \equiv \{ w \mid w \notin A \}$$

- Difference:

$$A - B \equiv \{ w \mid w \in A, w \notin B \} = A \cap B^c$$

- Reflection:

$$\mathcal{B} \equiv \{ w \mid w = -b, \quad \forall b \in B \}$$

- Translation

$$(A)z \equiv \{ c \mid c = a + z, \quad \forall a \in A \}$$

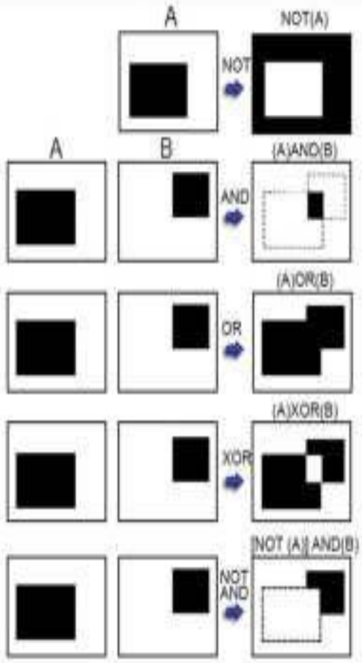
## Logic Operations Involving Binary Pixels and Images

- The principal logic operations used in image processing are: **AND, OR, NOT** (**COMPLEMENT**).

  **AND (Intersection):** *A* and *B* is the set that contains all elements of *A* that also belong to *B* (or equivalently, all elements of *B* that also belong to *A*), but no other elements.

  **OR** (**Union**): **A** is a subset of **B**. **set** A is included in **set B**. A is a subset of **B**, but A is not equal to **B**. A is a superset of **B**, but **B** is not equal to A.

  **XOR**: "one or the other, but not both", and set difference, **A−B**.

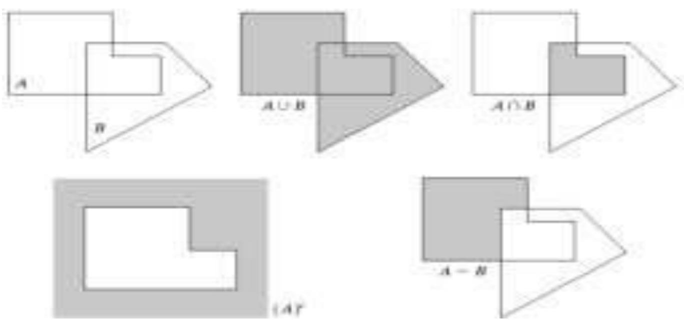



- A **subset** is denoted by A ⊆ Band means every element in set A is also in set B.
- A **superset** is denoted by A ⊇ B and means every element in set B is also in set A.
- A **proper subset** is denoted by A ⊂ B and means every element in set A is also in set B and the two sets are not identical.
- A **proper superset** is denoted by A ⊂ B and means every element in set B is also in set A and the two sets are not identical.
- These operations are *functionally complete*.

- Logic operations are performed on a pixel by pixel basis between corresponding pixels (bitwise).

- Other important logic operations : **XOR (exclusive OR), NAND (NOT-AND)**

- Logic operations are just a private case for a **binary set operations**, such: AND – Intersection, OR – Union, NOT-Complement.

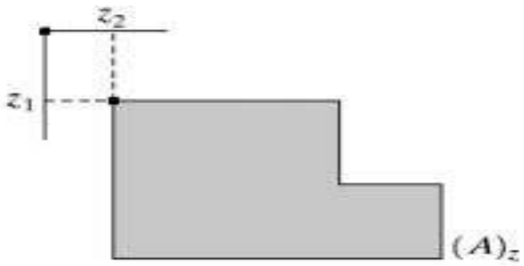


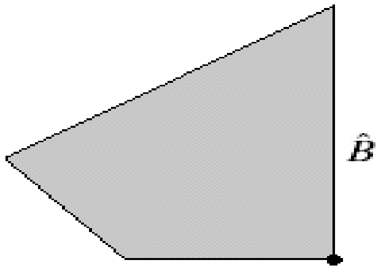

a b c
d e
FIGURE 9.1 (a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

## Reflection and Translation

The **translation** of A by $x = (x_1, x_2)$ is $(A)_z = \{c \mid c = a + z, \text{ for } a \in A\}$

Where, $c = (c_1, c_2) = (a_1+x_1, a_2+x_2)$




The **reflection** of B is $\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$

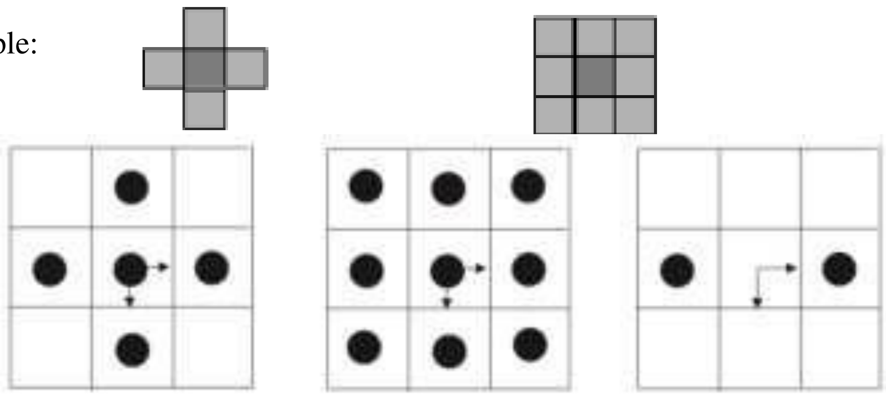
Here the reflection is with respect to a specific origin, such as a point center in the shape, e.g., the center of the shape.

## 1.2 Erosion and Dilation

### Structuring element (SE):

- Small set to probe the image under study.
- For each SE, define origin.
- Shape and size must be adapted to geometric properties for the objects.

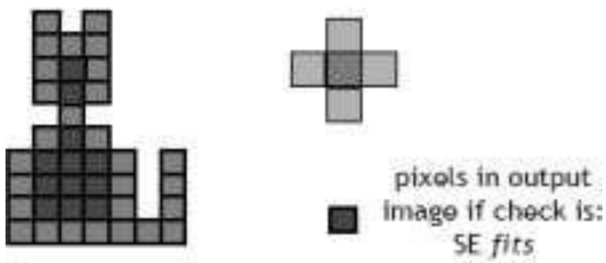
Example:



In parallel for each pixel in binary image:

– Check if SE is ‖satisfied‖

– Output pixel is set to 0 or 1 depending on used operation



***Five Binary Morphological Operations***

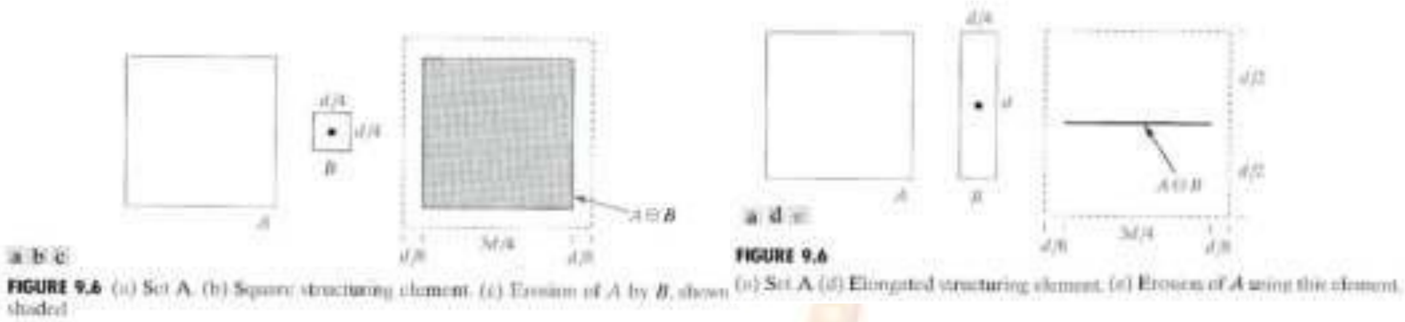
- Erosion $\ominus$
- Dilation $\oplus$
- Opening $\circ$
- Closing $\bullet$
- Hit-or-Miss transform $\otimes$

***Erosion:*** Erosion is used for shrinking of element A by using element B

- Erosion for Sets ***A*** and ***B*** in ***Z***$^2$, is defined by the following equation:

$$A \ominus B = \{z | [(B)z \subseteq A\} \quad (9.2-3)$$

- This equation indicates that the erosion of ***A*** by ***B*** is the set of all points *z* such that ***B***, translated by *z*, is contained in ***A***.



***Dilation:*** Dilation is used for expanding an element A by using structuring element B

Dilation of A by B and is defined by the following equation:

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (9.2-1)$$

This equation is based on obtaining the reflection of ***B*** about its origin and shifting this reflection by z.

The dilation of ***A*** by ***B*** is the set of all displacements z, such that ˆ  and ***A*** overlap by at least one element. Based on this interpretation the above equation of can be rewritten as:

$$A \oplus B = \{z | [(\hat{B})z \cap A] \subset A\} \quad (9.2-2)$$

Example – Document image analysis

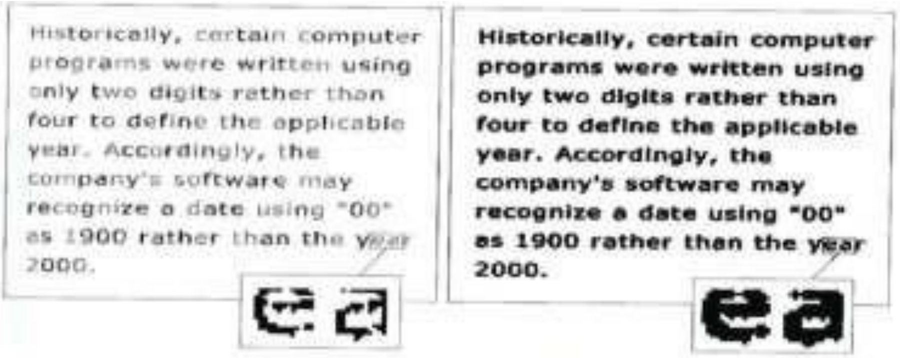



FIGURE 9.5 (a) Sample text of poor resolution with broken characters (magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

Usefulness:

Erosion: Removal of structures of certain shape and size, given by SE
Dilation: Filling of holes of certain shape and size, given by SE

***Duality:***

***Dilation and erosion are duals of each other*** with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

or

$$(A \oplus B)^c = A^c \ominus \hat{B}$$

In other words, dilating the –foreground‖ is the same as eroding the –background‖, but the structuring element reflects between the two. Likewise, eroding the foreground is the same as dilating the background.

So, strictly speaking we don‘t really need *both* dilate and erode: with one or the other, and with set complement and reflection of the structuring element, we can achieve the same functionality. Hence, dilation and erosion are duals.

***Proof:***

We know by erosion definition: $(A \Theta B)^c = \{ z \mid (B)_z \subseteq A\}^c$

If set $(B)_z$ is contained in A, then $(B)_z \cap A^c = \phi$, therefore

$$(A \Theta B)^c = \{ z \mid (B)_z \cap A^c = \Phi \}^c$$

But the complement of z‘s satisfies $\{ z \mid (B)_z \cap A^c = \Phi \}^c = \{ z \mid (B)_z \cap A^c \neq \Phi \}$

Therefore, $(A \Theta B)^c = \{ z \mid (B)_z \cap A^c \neq \Phi \}$

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

Hence Proved….

## 1.3 Opening and Closing

***Opening:***

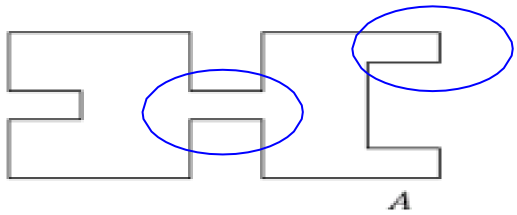
An *opening* is an erosion followed by a dilation with the same structuring element:

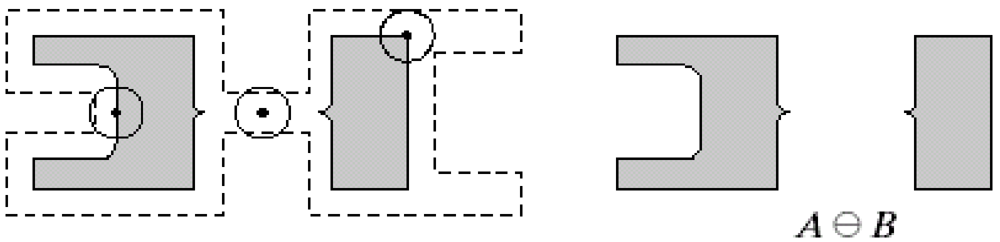
$$A \circ B = (A \ominus B) \oplus B$$

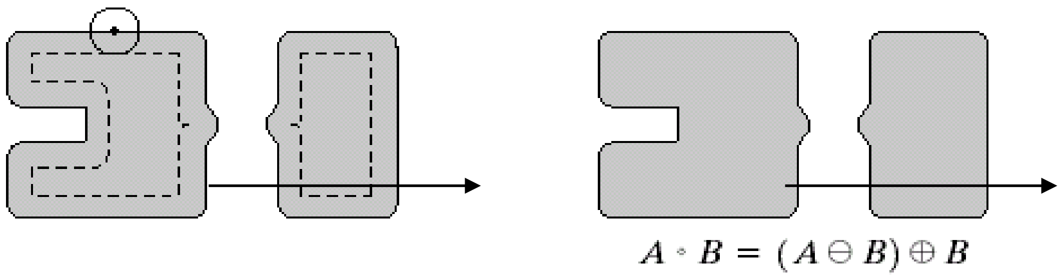
Remember that erosion finds all the places where the structuring element fits inside the image, but it only marks these positions at the origin of the element.

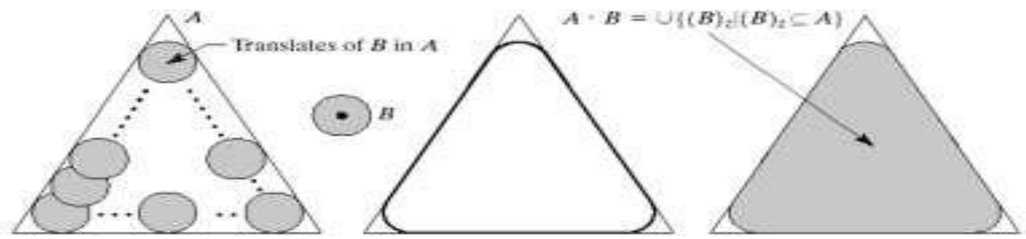
By following an erosion by a dilation, we −fill back in‖ the full structuring element at places where the element fits inside the object.

So, an opening can be considered to be the union of all translated copies of the structuring element that can fit inside the object. Openings can be used to remove small objects, protrusions from objects, and connections between objects.




Find contour→ Fill in Contour →

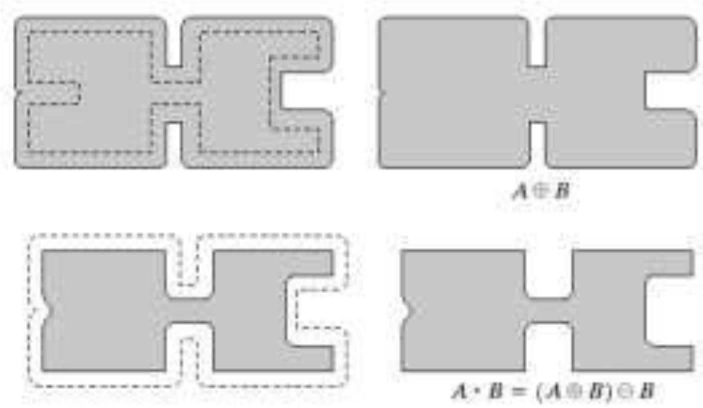
Smooth the contour of an image, breaks narrow isthmuses, eliminates thin protrusions

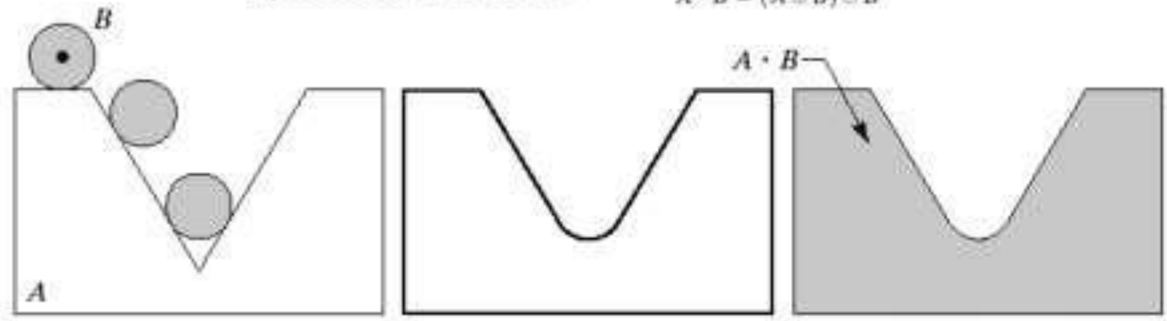
## *Closing:*

Closing works in an opposite fashion from opening:

$$A \bullet B = (A \oplus B) \ominus B$$

Whereas opening removes all pixels where the structuring element won't fit inside the image foreground, closing fills *in* all places where the structuring element will not fit in the image *background*.




Find contour→ Fill in Contour →

Smooth the object contour, fuse narrow breaks and long thin gulfs, eliminate small holes, and fill in gaps.

**Properties**

**Opening**

(i) A°B is a subset (subimage) of A

(ii) If C is a subset of D, then C °B is a subset of D °B

(iii) (A °B) °B = A °B

**Closing**

(i) A is a subset (subimage) of A•B

(ii) If C is a subset of D, then C •B is a subset of D •B

(iii) (A •B) •B = A •B

**Note:** repeated openings/closings have no effect!

## 1.4 The Hit-or-Miss Transforms

- A basic morphological tool for **shape detection**.
- Let the origin of each shape be located at its center of gravity.
- If we want to find the location of a shape , say – **X** , at (larger) image, say – **A** :
    - Let X be enclosed by a small window, say – **W**.
    - The ***local background*** of X with respect to W is defined as the *set difference* (**W - X**).
    - Apply *erosion* operator of A by X, will get us the set of locations of the origin of X, such that X is completely contained in A.
    - It may be also view geometrically as the set of all locations of the origin of X at which X found a match (**hit**) in A.

        - Apply *erosion* operator on the *complement of A* by the *local background* set (W – X).
        - Notice, that the set of locations for which X **exactly** fits inside A is the ***intersection*** of these two last operators above.

This intersection is precisely the location sought.

Formally:

If B denotes the set composed of X and it‘s background –

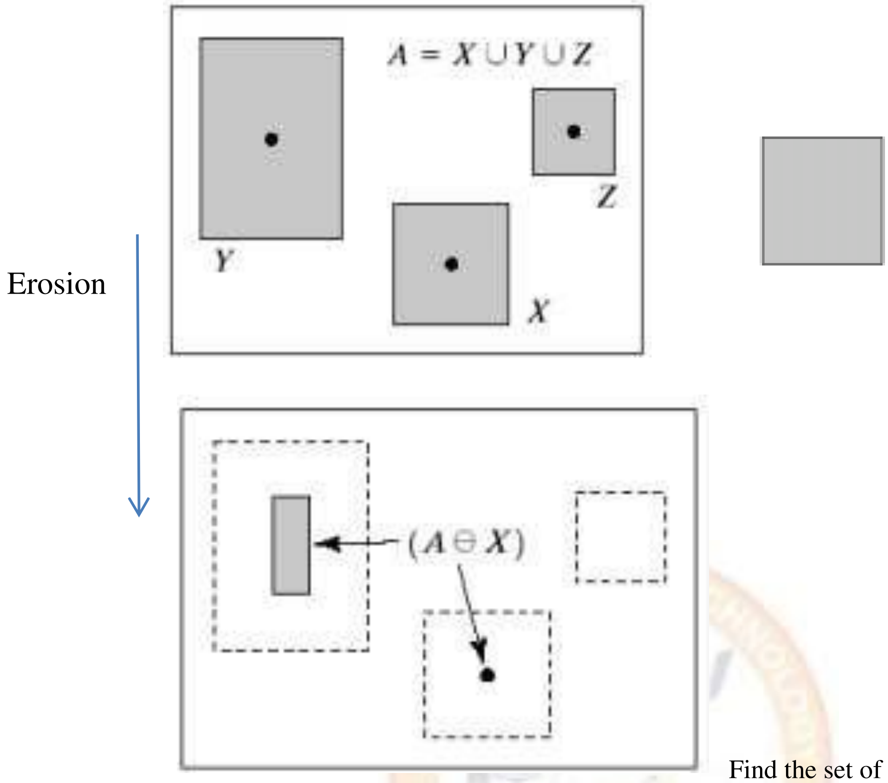
B = (B1, B2); B1 = X, B2 = (W – X).

The match (or set of matches) of B in A, denoted is:

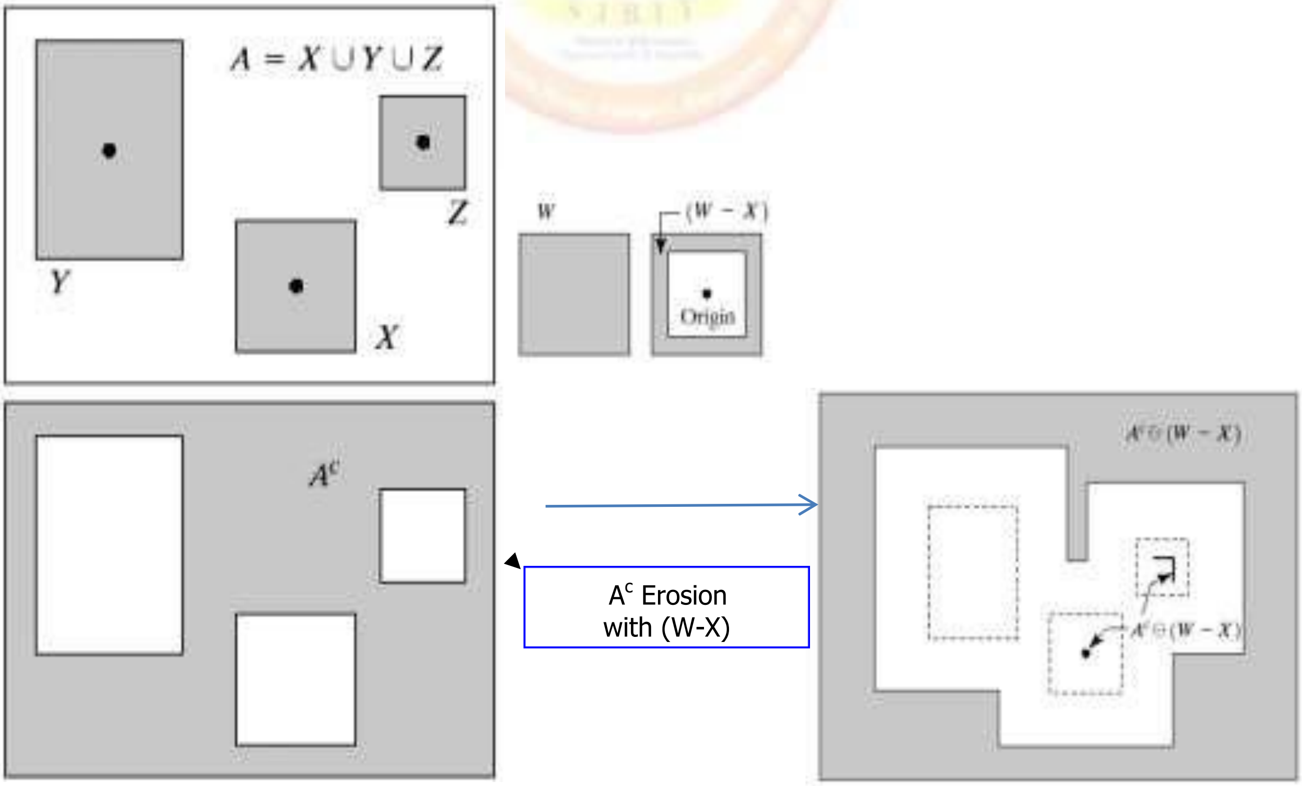
$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

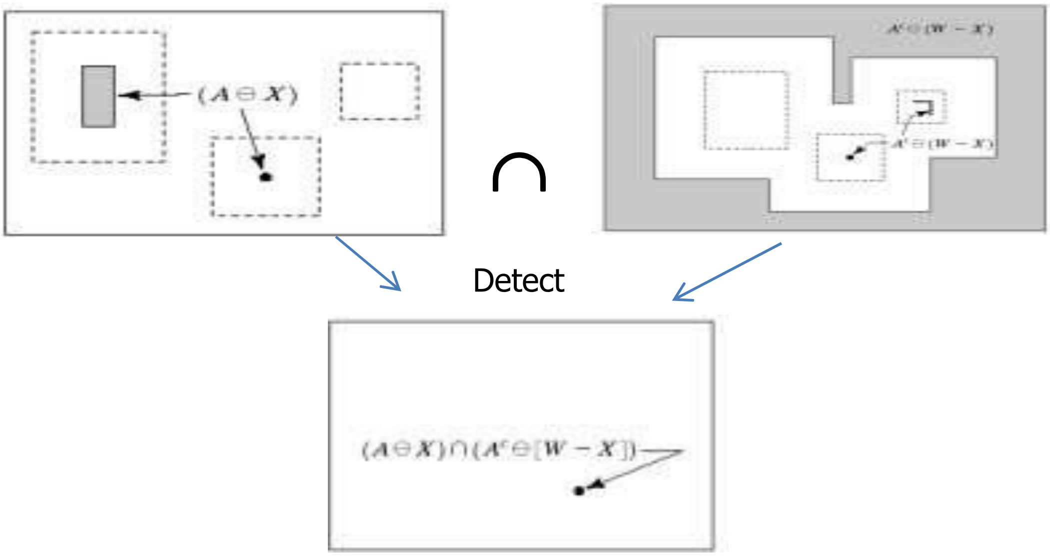
Steps:

- Find the location of certain shape, ex. ‗X'




Find the set of pixels that contain shape *X*






## 1.5 Some Basic Morphological Algorithms

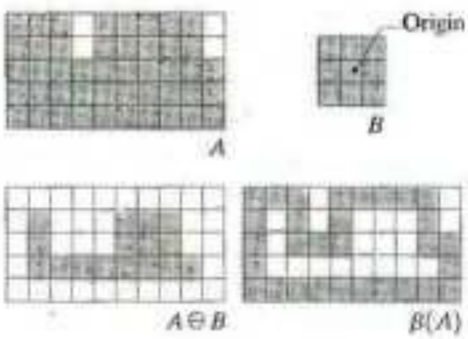
Extract image components that are useful in the representation and description of shape:

- Boundary extraction
- Region filling
- Extract of connected components
- Convex hull
- Thinning
- Thickening
- Skeleton
- Pruning

***Boundary Extraction:***

- First, erode A by B, then make set difference between A and the erosion
- The thickness of the contour depends on the size of constructing object – B

$$\beta(A) = A - (A \ominus B)$$

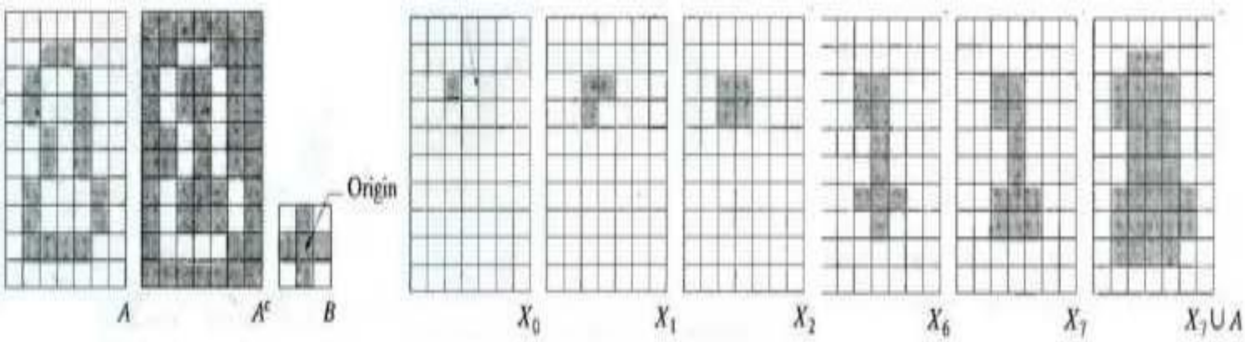


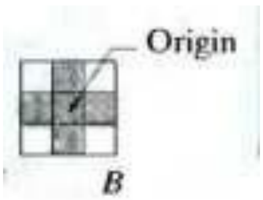
***Region Filling***

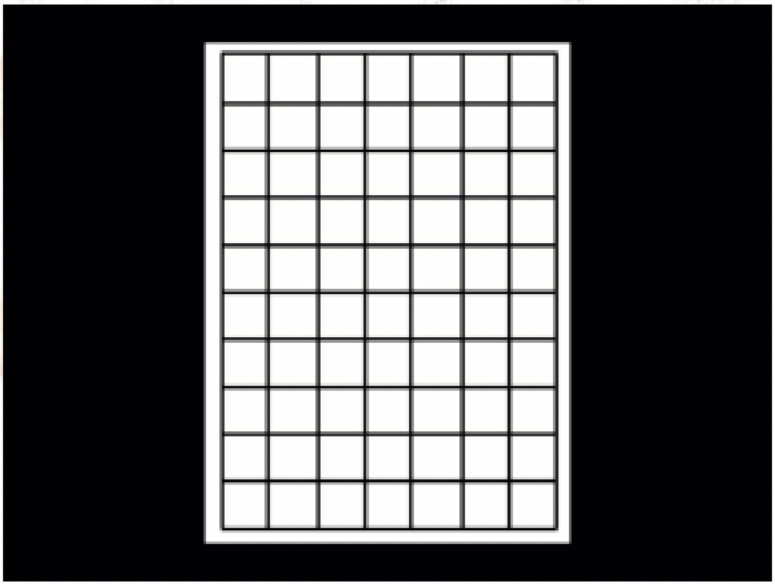
- This algorithm is based on a set of dilations, complementation and intersections
- p is the point inside the boundary, with the value of 1
- X(k) = (X(k-1) xor B) conjunction with complemented A

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

- The process stops when X(k) = X(k-1)
- The result that given by union of A and X(k), is a set contains the filled set and the boundary



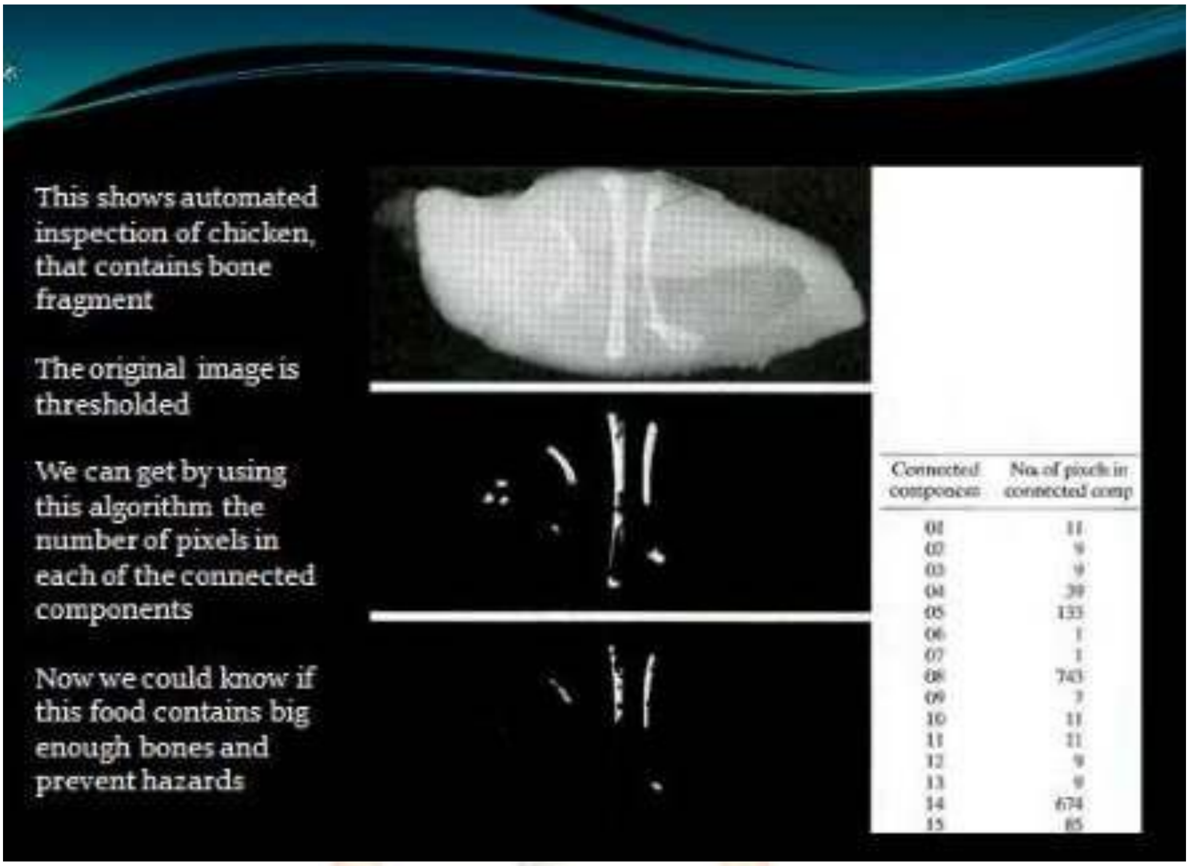



$$X_k = (X_{k-1} \oplus B) \cap A^c$$

***Extraction of Connected Components***

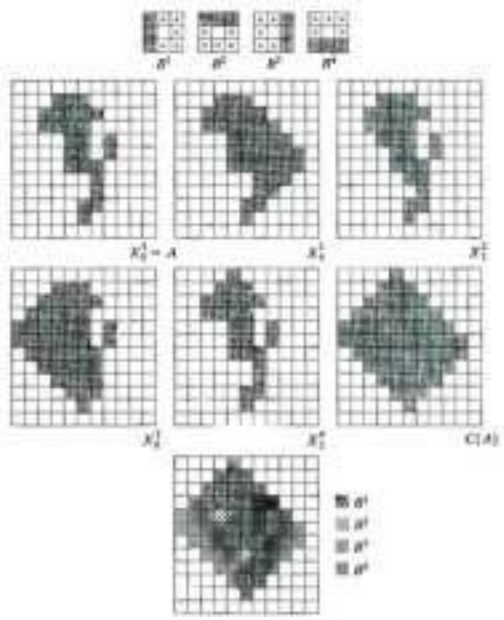
- This algorithm extracts a component by selecting a point on a binary object A
- Works similar to region filling, but this time we use in the conjunction the object A, instead of it‘s complement

$$X_k = (X_{k-1} \oplus B) \cap A$$




***Convex Hull***

- A is said to be convex if a straight line segment joining any two points in A lies entirely within A
- The convex hull H of set S is the smallest convex set containing S
- Convex deficiency is the set difference H-S
- Useful for object description
- This algorithm iteratively applying the hit-or-miss transform to A with the first of B element, unions it with A, and repeated with second element of B



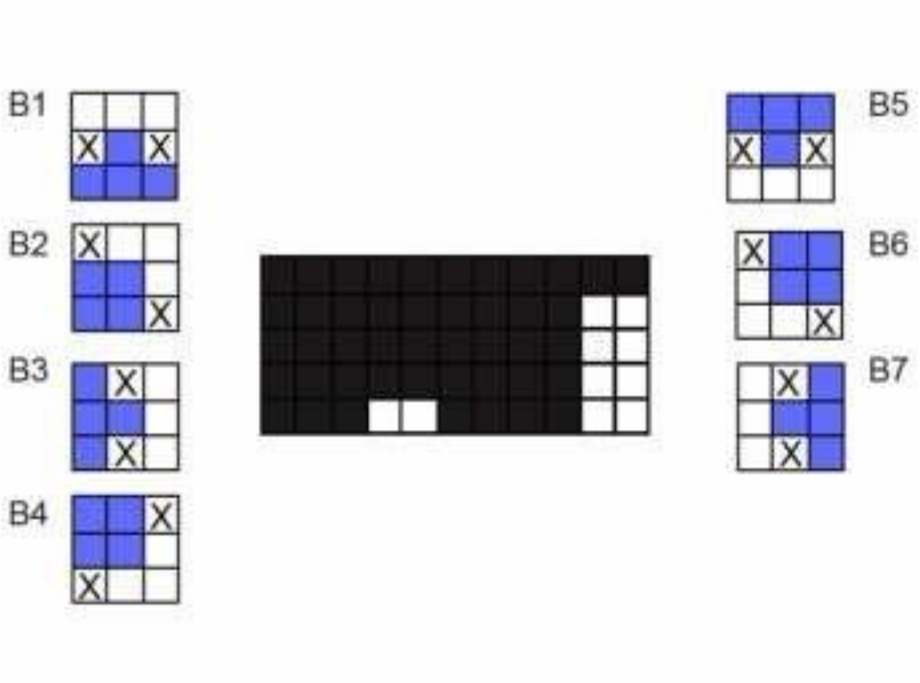
$$X_k^i = (X_{k-1} \circledast B^i) \cup A$$

## ***Thinning***

- The thinning of a set A by a structuring element B, can be defined by terms of the hit-and-miss transform:

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

- A more useful expression for thinning A symmetrically is based on a sequence of structuring elements:
  $\{B\}=\{B^1, B^2, B^3, \ldots, B^n\}$
- Where $B^i$ is a rotated version of $B^{i-1}$. Using this concept we define thinning by a sequence of structuring elements:
- The process is to thin by one pass with $B^1$, then thin the result with one pass with $B^2$, and so on until A is thinned with one pass with $B^n$.
- The entire process is repeated until no further changes occur.
- Each pass is performed using the equation: $A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$




## ***Thickening***

- Thickening is a morphological dual of thinning.
- Definition of thickening .
- As in thinning, thickening can be defined as a sequential operation:

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n)$$

- The structuring elements used for thickening have the same form as in thinning, but with all 1's and 0's interchanged.
- A separate algorithm for thickening is often used in practice, Instead the usual procedure is to thin the background of the set in question and then complement the result.
- In other words, to thicken a set A, we form $C=A^c$, thin C and then form $C^c$.

- Depending on the nature of A, this procedure may result in some disconnected points. Therefore thickening by this procedure usually require a simple post-processing step to remove disconnected points.
- We will notice in the next example that the thinned background forms a boundary for the thickening process, this feature does not occur in the direct implementation of thickening
- This is one of the reasons for using background thinning to accomplish thickening.

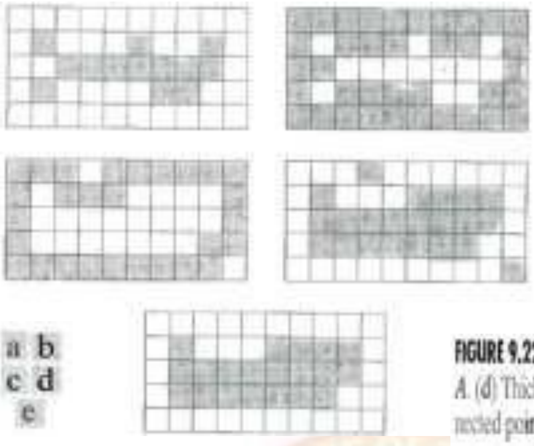



**FIGURE 9.22** (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

***Skeleton***

- The notion of a skeleton S(A) of a set A is intuitively defined, we deduce from this figure that:
    a) If z is a point of S(A) and (D)z is the largest disk centered in z and contained in A (one cannot find a larger disk that fulfils this terms) – this disk is called –maximum disk‖.
    b) The disk (D)z touches the boundary of A at two or more different places.

- The skeleton of A is defined by terms of erosions and openings:

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

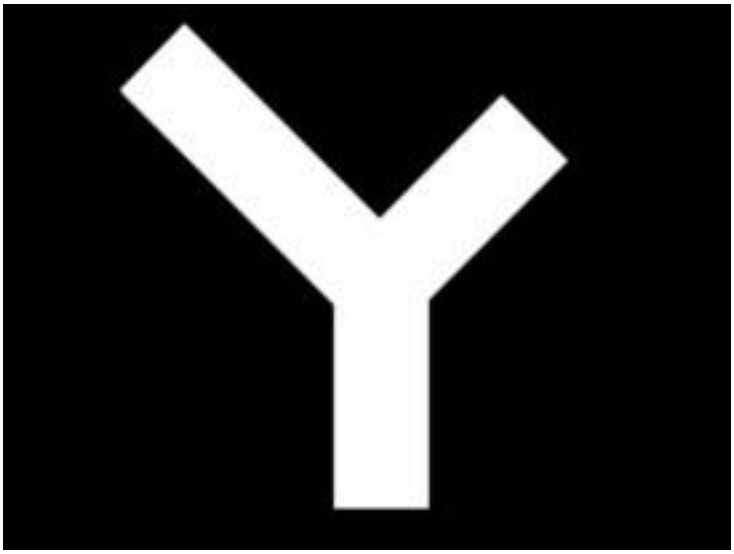
- With $$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

- Where B is the structuring element and $(A \ominus kB)$ indicates k successive erosions of A:

$$(A \ominus kB) = (\ldots((A \ominus B) \ominus B) \ominus \ldots) \ominus B$$

- k times, and K is the last iterative step before A erodes to an empty set, in other words: $K = \max\{k \mid (A \ominus kB) \neq \emptyset\}$

- In conclusion S(A) can be obtained as the union of skeleton subsets Sk(A).



- A can be also reconstructed from subsets Sk(A) by using the equation:

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$$

- Where $(S_k(A) \oplus kB)$ denotes k successive dilations of $S_k(A)$ that is:

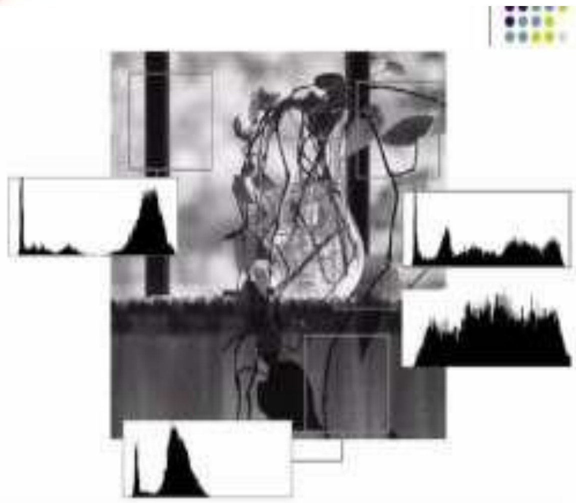
$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B$$

### 4.2 Wavelets:

#### 4.2.1 Background

Unlike the Fourier transform, which decomposes a signal to a sum of sinusoids, the wavelet transform decomposes a signal (image) to small waves of *varying frequency* and *limited duration*. The advantage is that we also know when (where) the frequency appears.

FIGURE 7.1 A natural image and its local histogram variations.

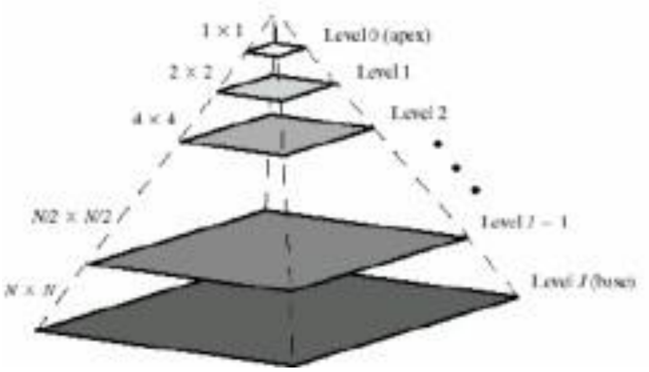


Many applications in image compression, transmission, and analysis. We will examine wavelets from a multi-resolution point of view and begin with an overview of imaging techniques involved in multi-resolution theory.

Small objects are viewed at high resolutions. Large objects require only a coarse resolution. Images have locally varying statistics resulting in combinations of edges, abrupt features and homogeneous regions.

**Image Pyramids:**

Originally devised for machine vision and image compression.
It is a collection of images at decreasing resolution levels.
Base level is of size $2^J$x$2^J$ or $N$x$N$. Level $j$ is of size $2^j$x$2^j$

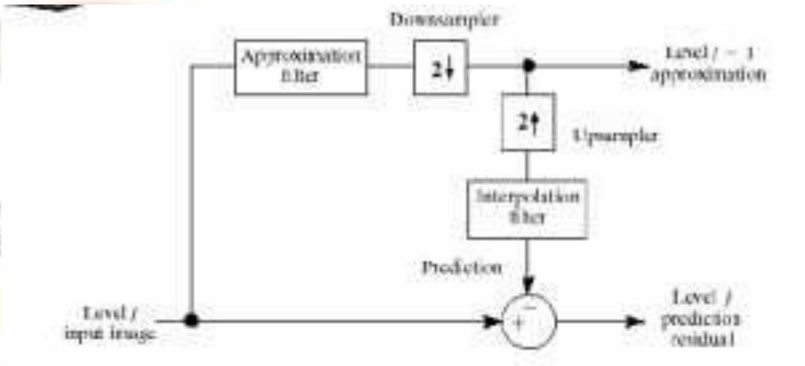



Approximation pyramid:

At each reduced resolution level we have a filtered and downsampled image.
$f_{\downarrow 2}(n) = f(2n)$

Prediction pyramid:
A prediction of each high resolution level is obtained by up-sampling (inserting zeros) the previous low resolution level (prediction pyramid) and interpolation (filtering).

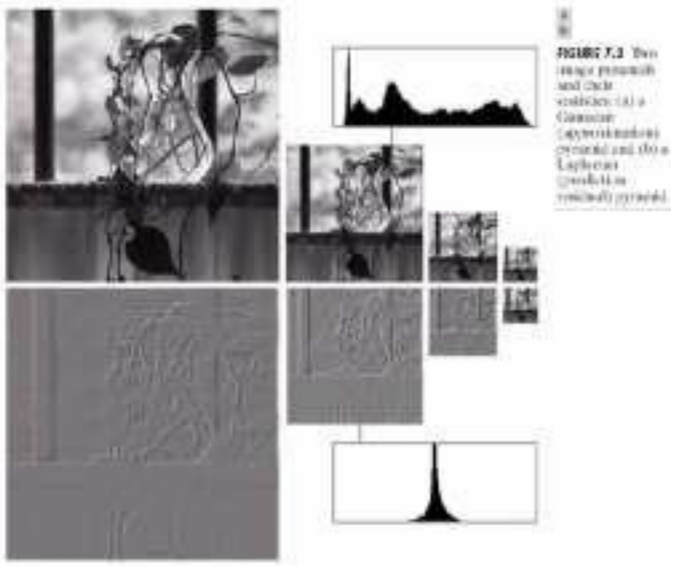
$$f_{2\uparrow}(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$




Prediction residual pyramid:
At each resolution level, the prediction error is retained along with the lowest resolution level image. The original image may be reconstructed from this information.

Approximation pyramid

Prediction residual pyramid



**Subband Coding:**

{Review - Digital signal filtering}

An image is decomposed to a set of band limited components (sub-bands). The decomposition is carried by filtering and down-sampling. If the filters are properly selected the image may be reconstructed without error by filtering and up-sampling.

Consider the two-band subband coding and decoding system as shown in figure (a). The system is composed of two filter banks, each containing two FIR filters ($h_0(n)$, $h_1(n)$ & $g_0(n)$, $g_1(n)$).

Figure a

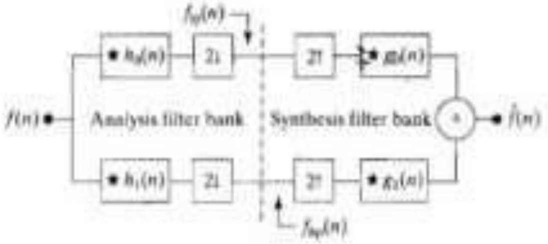
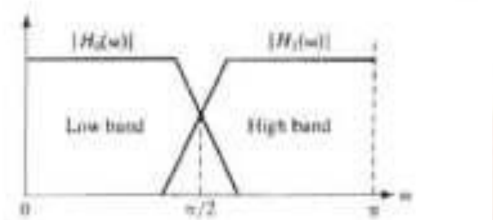



Figure b




**Analysis filter** bank includes $h_0(n)$ & $h_1(n)$ to break f(n) into two half-length sequences $f_{lp}(n)$ & $f_{hp}(n)$. Filters $h_0(n)$ & $h_1(n)$ are half-band filters whose idealized characteristics are $H_0(w)$ and $H_1(w)$ are as shown in figure (b).

$h_0(n)$ → low pass filter, output $f_{lp}(n)$ is called approximation of f(n).

$h_1(n)$ → high pass filter, output $f_{hp}(n)$ is called detail part of f(n).

**Synthesis filter** bank includes $g_0(n)$ & $g_1(n)$ combines $f_{lp}(n)$ & $f_{hp}(n)$ to produce ˆ .

The goal in subband coding is to select $h_0(n)$, $h_1(n)$, $g_0(n)$ & $g_1(n)$ filters so that ˆ .

The resulting system is said to be ***Perfect Reconstruction Filters***.

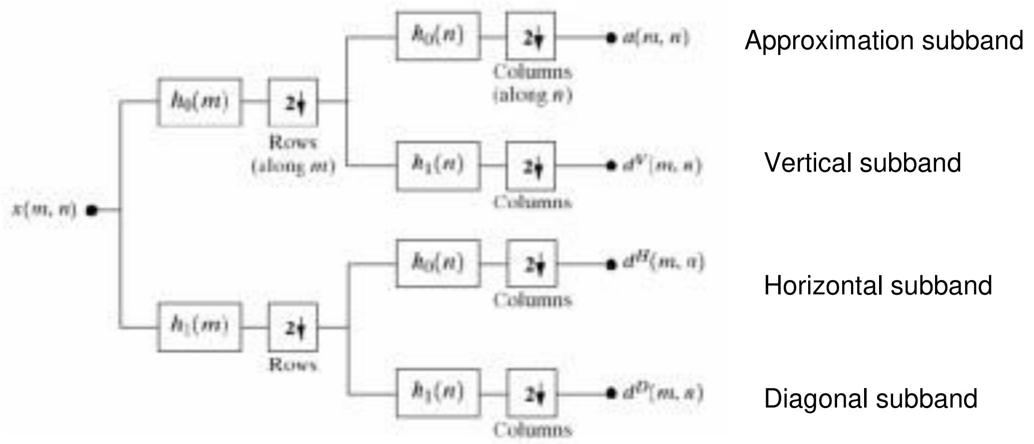
To obtain ***Perfect Reconstruction Filters,*** The analysis and synthesis filters should be related in one of the two ways:

$$g_0(n) = (-1)^n h_1(n)$$
$$g_1(n) = (-1)^{n+1} h_0(n)$$

or

$$g_0(n) = (-1)^{n+1} h_1(n)$$
$$g_1(n) = (-1)^n h_0(n)$$

These filters are called *cross-modulated*.




Also, the filters should be *biorthogonal*:

$$\langle h_i(2n-k), g_j(k)\rangle = \delta(i-j)\delta(n), \quad i, j = \{0,1\}$$

Of special interest in subband coding are filters that move beyond biorthogonality and require being *orthonormal*:

$$\langle g_i(n), g_j(n+2m)\rangle = \delta(i-j)\delta(m), \quad i, j = \{0,1\}$$

In addition, orthonormal filters satisfy the following conditions:

$$g_i(n) = (-1)^n g_0(K_{even} - 1 - n)$$
$$h_i(n) = g_i(K_{even} - 1 - n), \; i = \{0,1\}$$

Where the subscript means that the size of the filter should be even.

$$g_i(n) = (-1)^n g_0(K_{even} - 1 - n)$$
$$h_i(n) = g_i(K_{even} - 1 - n), \; i = \{0,1\}$$

Synthesis filters are related by order reversal and modulation.
Analysis filters are both order reversed versions of the synthesis filters.

An orthonormal filter bank may be constructed around the impulse response of $g_0$ which is called the prototype.

1-D orthonormal filters may be used as 2-D separable filters for subband image coding.

The wavy lines are due to aliasing of the barely discernable window screen. Despite the aliasing, the image may be perfectly reconstructed.

**The Haar Transform:**

It is due to Alfred Haar [1910]. Its basis functions are the simplest known orthonormal wavelets. The Haar transform is both separable and Symmetric. The Haar transform can be expressed in matrix form

$$T = HFH^T$$

Where F is an N*N image matrix, H is an N*N transformation matrix, T is the resulting N*N transform.

- For the Haar transform, transformation matrix H contains the Haar basis functions, h(z).
- They are defined over the continuous, closed interval for $z \in [0,1]$ for k=0,1,2,…,N-1, where $N = 2^n$.
- To generate H, we define the integer k such that
  $k = 2^P + q - 1$ where, $0 \le p \le n - 1$ $q = 0$ or $1$ for $p = 0$
  $1 \le q \le 2^p$ for $p \ne 0$
- For the above pairs of p and q, a value for k is determined and the Haar basis functions are computed.

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \; z \in [0,1]$$

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \cdot \begin{cases} 2^{p/2} & (q-1)/2^p \le z \le (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \le z \le q/2^p \\ 0 & \text{otherwise}, \; z \in [0,1] \end{cases}$$

The $i^{th}$ row of a $N$x$N$ Haar transformation matrix contains the elements of $h_k(z)$ for $z$=0/$N$, 1/$N$, 2/$N$,…, ($N$-1)/$N$.

e.g. For instance, for N=4, p, q and k have the following values:

| k | p | q |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | 2 |

And the 4x4 transformation matrix is:

$$\mathbf{H}_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Similarly, for N=2, the 2x2 transformation matrix is:

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The rows of $\mathbf{H}_2$ are the simplest filters of length 2 that may be used as analysis filters $h_0(n)$ and $h_1(n)$ of a perfect reconstruction filter bank.

Moreover, they can be used as scaling and wavelet vectors (defined in what follows) of the simplest and oldest wavelet transform.

**4.2.2** Multi-resolution Expansions.

Series Expansions

Scaling Functions

Wavelet Functions

Assignment

| Module – 5 | |
|---|---|
| **Segmentation**: Point, Line, and Edge Detection, Thresholding, Region-Based Segmentation, Segmentation Using Morphological Watersheds.<br>**Representation and Description:** Representation, Boundary descriptors.<br>**[Text: Chapter 10: Sections 10.2, to 10.5 and Chapter 11: Sections 11.1 and 11.2]** | L1, L2, L3 |

**5.1 Segmentation**:

**5.1.1 Point, Line, and Edge Detection**

**5.1.2 Thresholding, Region-Based Segmentation,**

**5.1.3 Segmentation Using Morphological Watersheds.**

**5.2 Representation and Description:**

**5.2.1 Representation**

**5.2.2 Boundary descriptors.**

# Module 5

## Segmentation, Representation & Description

By:
Dr. Supreeth H S G
Associate Professor
Dept. of ECE, SJBIT

# Outline:

**Segmentation**: Point, Line, and Edge Detection, Thresholding, Region-Based Segmentation, Segmentation Using Morphological Watersheds.

**Representation and Description:** Representation, Boundary descriptors.

**[Text: Chapter 10: Sections 10.2, to 10.5 and Chapter 11: Sections 11.1 and 11.2] – Gonzalez & Woods**

# Preview

- Segmentation subdivides an image to regions or objects

The goal is usually to find individual objects in an image.

Two basic properties of intensity values

- Discontinuity
  - Edge detection
- Similarity
  - Thresholding
  - Region growing/splitting/merging

# Detection of discontinuities – **Point Detection**

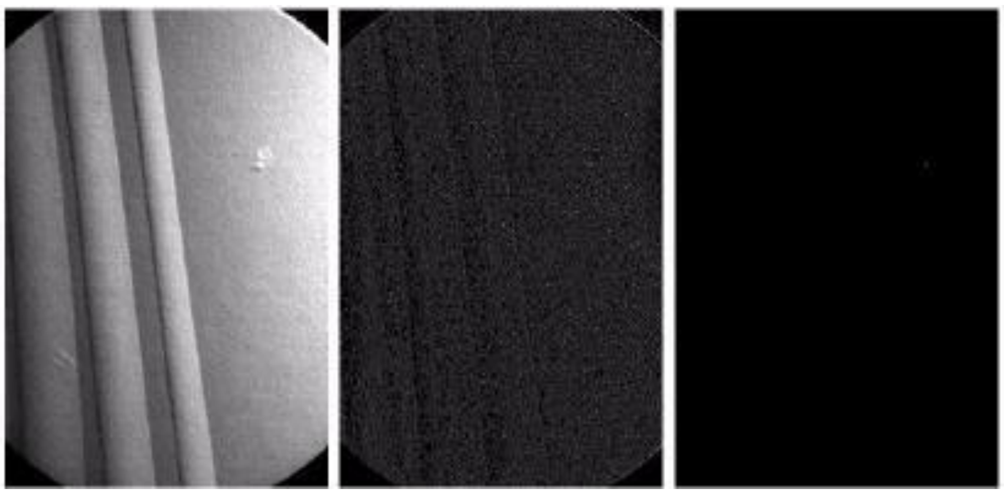
- Mask operation

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

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

# Point detection

- Isolated point $|R| \geq T$
  - whose gray value is significantly different from its background



| | | |
|---|---|---|
| −1 | −1 | −1 |
| −1 | 8 | −1 |
| −1 | −1 | −1 |

# Line Detection

**Mask operation**

– Preferred direction is weighted by with a larger coefficient

– The coefficients in each mask sum to zero response of constant gray level areas

– Compare values of individual masks (run all masks) or run only the mask of specified direction

| | | |
|---|---|---|
| −1 | −1 | −1 |
| 2 | 2 | 2 |
| −1 | −1 | −1 |

Horizontal

| | | |
|---|---|---|
| −1 | −1 | 2 |
| −1 | 2 | −1 |
| 2 | −1 | −1 |

+45°

| | | |
|---|---|---|
| −1 | 2 | −1 |
| −1 | 2 | −1 |
| −1 | 2 | −1 |

Vertical

| | | |
|---|---|---|
| 2 | −1 | −1 |
| −1 | 2 | −1 |
| −1 | −1 | 2 |

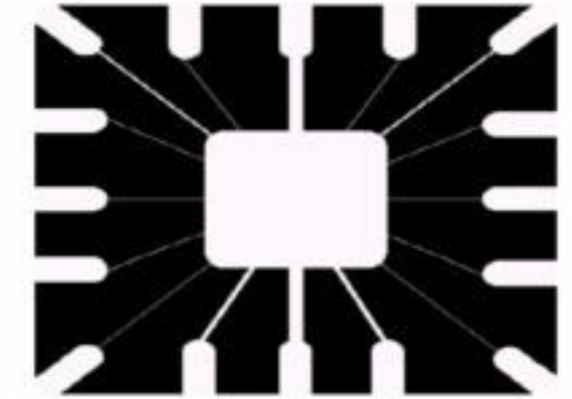
−45°

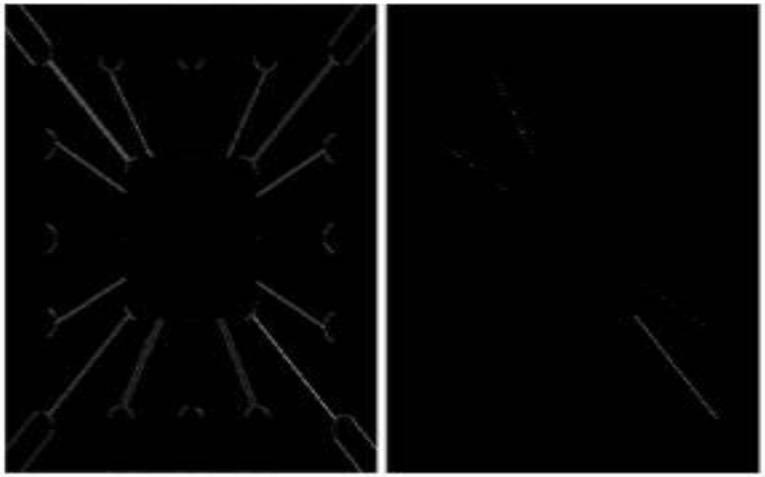
Example:

– Interested in lines of -45 degree

– Run the corresponding mask

– All other lines are eliminated

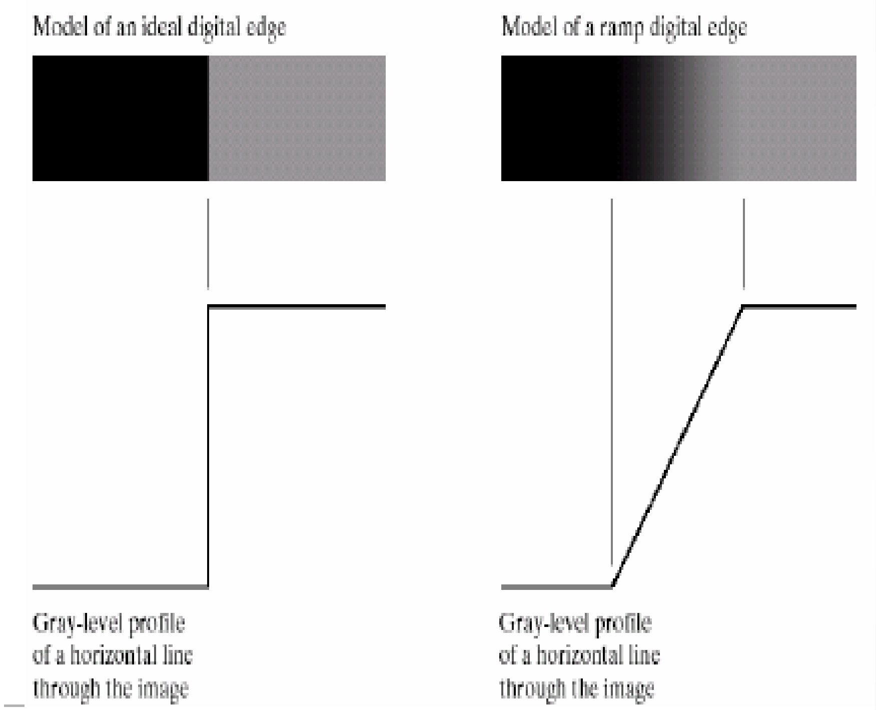




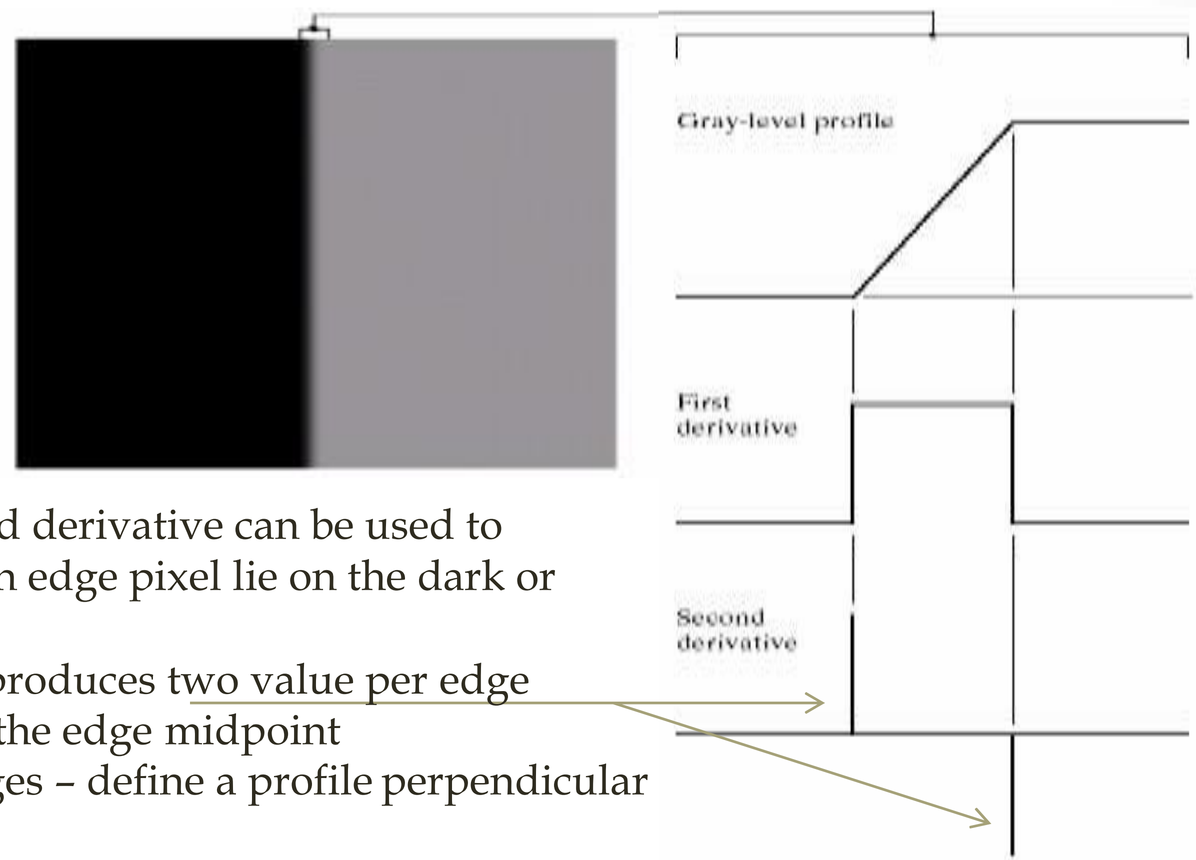
# Edge Detection

Edge: a set of connected pixels that lie on the boundary between two regions

- 'Local' concept is contrast to 'more global' boundary concept
- To be measured by grey-level transitions
- Ideal and blurred edges



First derivative can be used to detect the presence of an edge (if a point is on a ramp)




The sign of the second derivative can be used to determine whether an edge pixel lie on the dark or light side of an edge

– Second derivative produces two value per edge

– Zero crossing near the edge midpoint

– Non-horizontal edges – define a profile perpendicular the edge direction

- Edges in the presence of noise

– Derivatives are sensitive to (even fairly little) noise
– Consider image smoothing prior to the use of derivatives

- Edge definition again

– Edge point – whose first derivative is above a pre-specified threshold
– Edge – connected edge points
– Derivatives are computed through gradients (1st) and Laplacians (2nd)

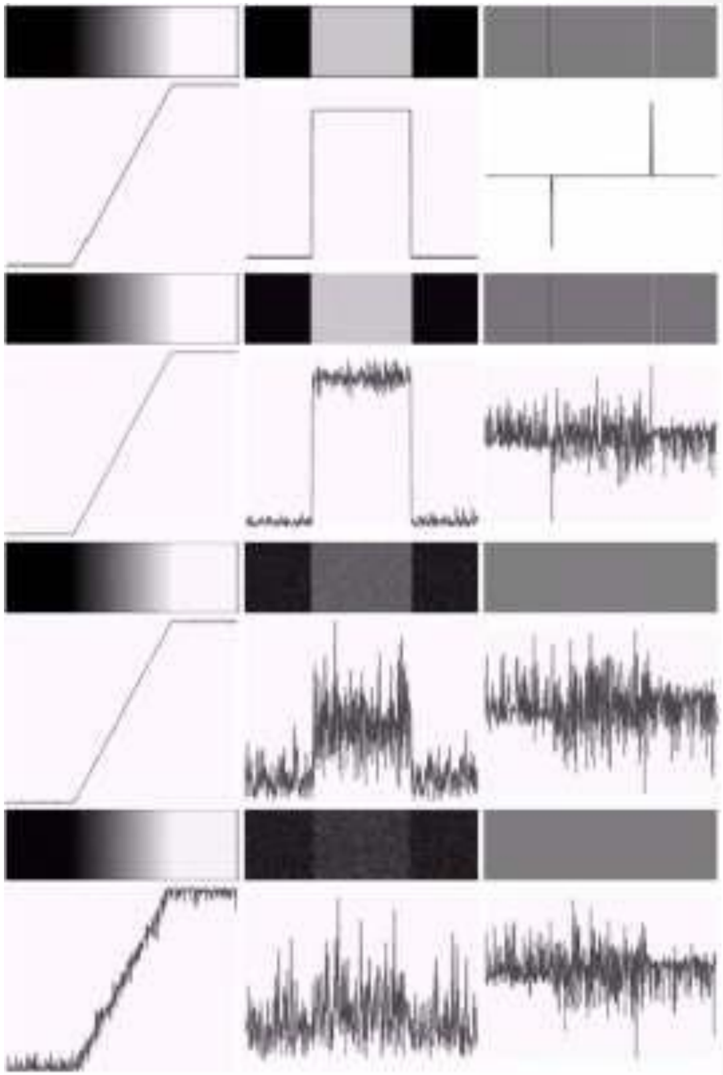


**FIGURE 10.7** First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma$ = 0.0, 0.1, 1.0, and 10.0, respectively. Second column: first-derivative images and gray-level profiles. Third column: second-derivative images and gray-level profiles.

- Gradient
  - **Vector** pointing to the direction of maximum rate of change of $f$ at coordinates $(x,y)$
  $$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$
  - **Magnitude**: gives the quantity of the increase (some times referred to as *gradient* too) $\nabla f = mag(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2}$ $\quad \nabla f \approx |G_x| + |G_y|$
  - **Direction**: perpendicular to the direction of the edge at $(x,y)$
  $$\alpha(x,y) = \tan^{-1}\left(\frac{G_x}{G_y}\right)$$
  - Partial derivatives computed through 2x2 or 3x3 masks
    - Sobel operators introduce some smoothing and give more importance to the center point

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

| | | | | |
|---|---|---|---|---|
| −1 | 0 | | 0 | −1 |
| 0 | 1 | | 1 | 0 |

Roberts

| | | | | | | |
|---|---|---|---|---|---|---|
| −1 | −1 | −1 | | −1 | 0 | 1 |
| 0 | 0 | 0 | | −1 | 0 | 1 |
| 1 | 1 | 1 | | −1 | 0 | 1 |

Prewitt

| | | | | | | |
|---|---|---|---|---|---|---|
| −1 | −2 | −1 | | −1 | 0 | 1 |
| 0 | 0 | 0 | | −2 | 0 | 2 |
| 1 | 2 | 1 | | −1 | 0 | 1 |

Sobel

## Detecting diagonal edges

| | | |
|---|---|---|
| 0 | 1 | 1 |
| −1 | 0 | 1 |
| −1 | −1 | 0 |

| | | |
|---|---|---|
| −1 | −1 | 0 |
| −1 | 0 | 1 |
| 0 | 1 | 1 |

Prewitt

| | | |
|---|---|---|
| 0 | 1 | 2 |
| −1 | 0 | 1 |
| −2 | −1 | 0 |

| | | |
|---|---|---|
| −2 | −1 | 0 |
| −1 | 0 | 1 |
| 0 | 1 | 2 |

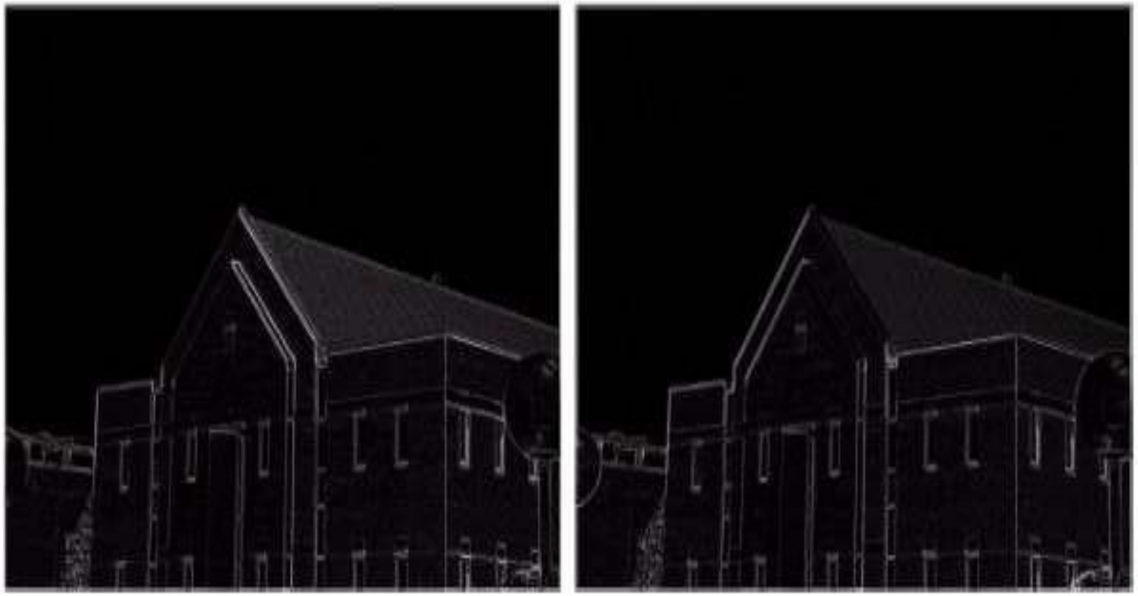
Sobel

a b
c d

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

a b
c d

**FIGURE 10.11** Same sequence as in Fig. 10.10, but with the original image smoothed with a $5 \times 5$ averaging filter.



a b

**FIGURE 10.12** Diagonal edge detection. (a) Result of using the mask in Fig. 10.9(c). (b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).

## Laplacian

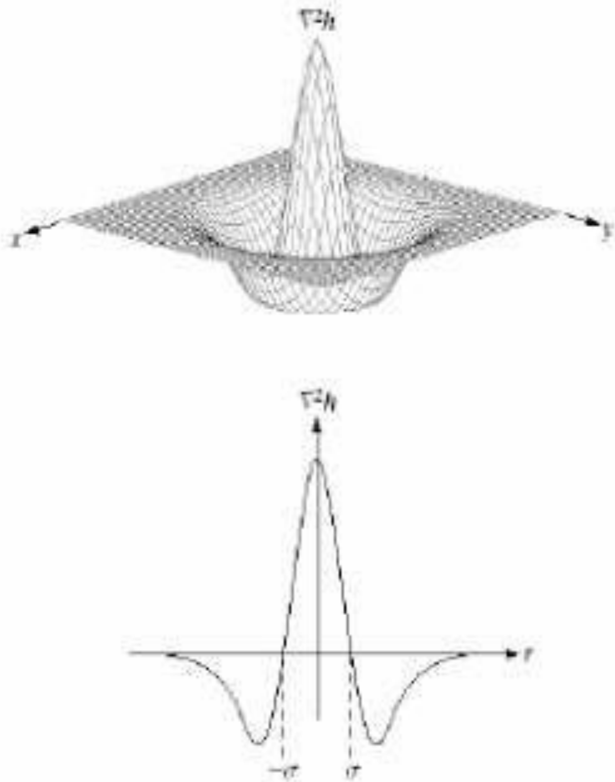
- Second-order derivative of a 2-D function $\nabla^2 f = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2}$
- Digital approximations by proper masks
- Complementary use for edge detection
  - Cons: Laplacian is very sensible to noise; double edges
  - Pros: Dark or light side of the edge; zero crossings are of better use
  - Laplacian of Gaussian (LoG): preliminary smoothing to find edges through zero crossings

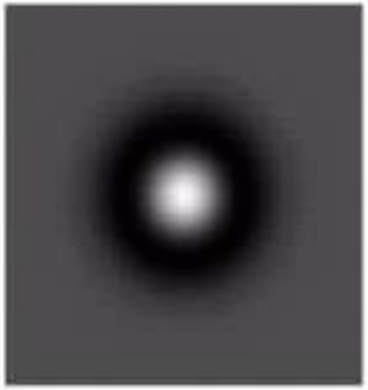
| | | |
|---|---|---|
| 0 | −1 | 0 |
| −1 | 4 | −1 |
| 0 | −1 | 0 |

| | | |
|---|---|---|
| −1 | −1 | −1 |
| −1 | 8 | −1 |
| −1 | −1 | −1 |

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

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

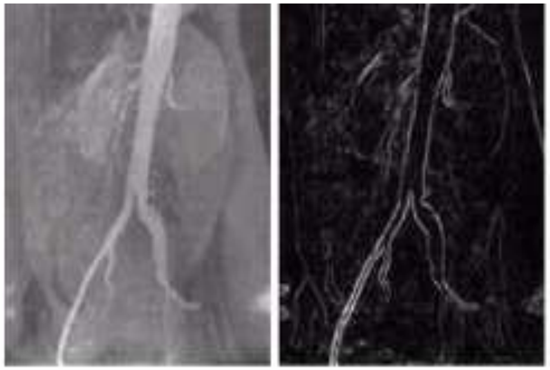


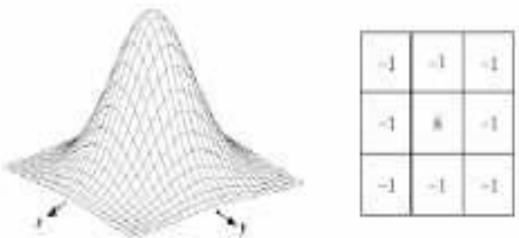



| | | | | |
|---|---|---|---|---|
| 0 | 0 | −1 | 0 | 0 |
| 0 | −1 | −2 | −1 | 0 |
| −1 | −2 | 16 | −2 | −1 |
| 0 | −1 | −2 | −1 | 0 |
| 0 | 0 | −1 | 0 | 0 |

a b
c d

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





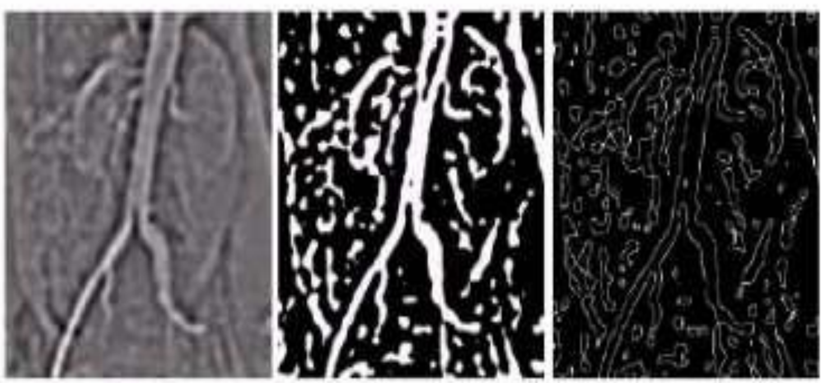


FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

- Example: Edges through LoG zero-crossings on an angiogram image
  - 27x27 Gaussian mask, 3x3 Laplacian
  - Thinner than the gradient edges
  - Closeed loops (spagetti effect)
  - Zero-crossing calculation is not straighforward

# Edge Linking & Boundary Detection

1. ***Local processing***
   - Analyze pixels in a small neighborhood following predefined criteria.
   - Two properties of edge points are useful for edge linking:
     - the strength (or magnitude) of the detected edge points
     - their directions (determined from gradient directions)

- Adjacent edge points with similar magnitude and direction are linked.
- For example, an edge pixel with coordinates $(x_0, y_0)$ in a predefined neighborhood of $(x, y)$ is similar to the pixel at $(x, y)$ if

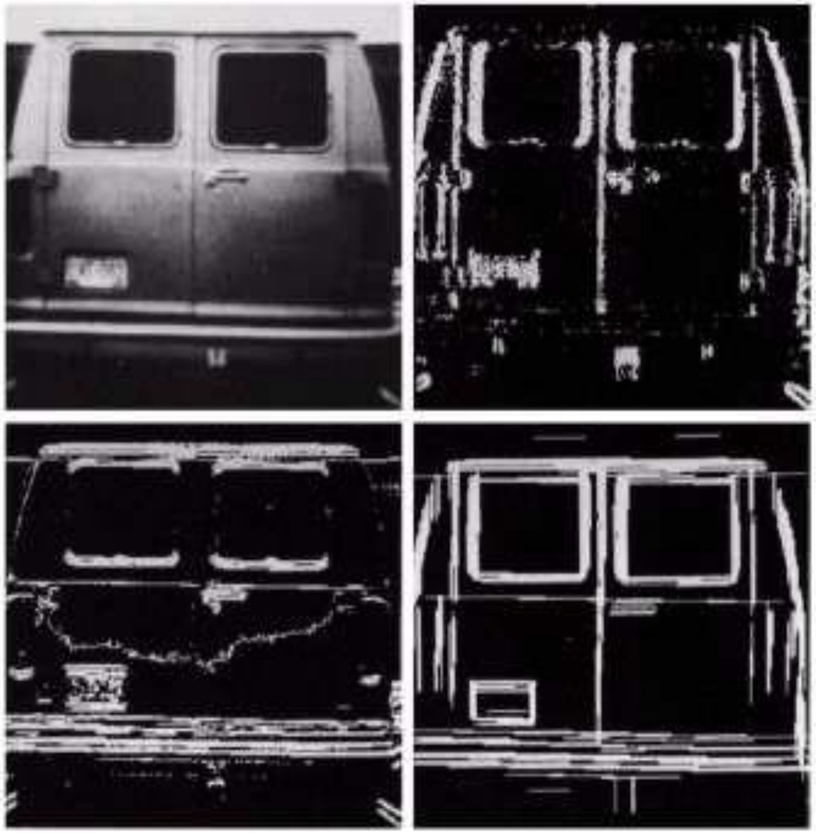
$$\left|\nabla f(x,y) - \nabla(x_0, y_0)\right| \le E, \quad E: \text{a nonnegative threshold}$$

$$\left|\alpha(x,y) - \alpha(x_0, y_0)\right| < A, \quad A: \text{a nonegative angle threshold}$$

Both magnitude and angle criteria should be satisfied

Example: find rectangular shapes similar to license plate

- Find gradients
- Connect edge points
- Check horizontal-vertical proportion



## 2. *Global Processing – the Hough Transform*

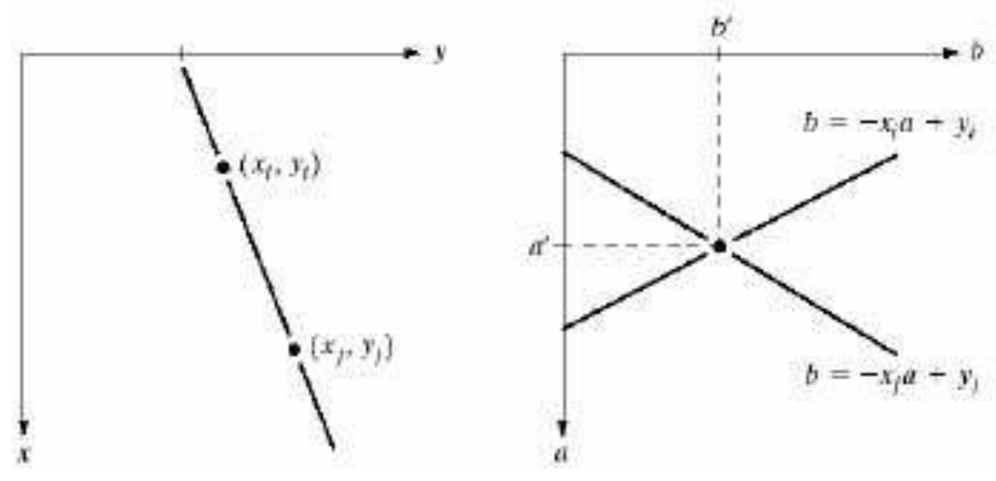
- Determine if points lie on a curve of specified shape
- Consider a point (*xi*, *yi*) and the general line equation

$$y_i = ax_i + b$$

- Write the equation with respect to *ab*-plane (*parametric space*)

$$b = -x_i a + y_i$$

- Write the equation for a second point (*xj*, *yj*) and find the intersection point (a′, b′) on the parametric space
- All points on a line intersect at the same parametric point



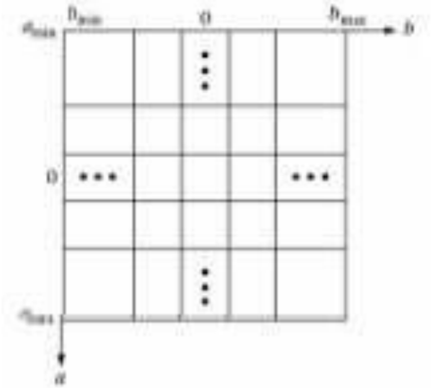

a b

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

## Computational aspects of the Hough transform

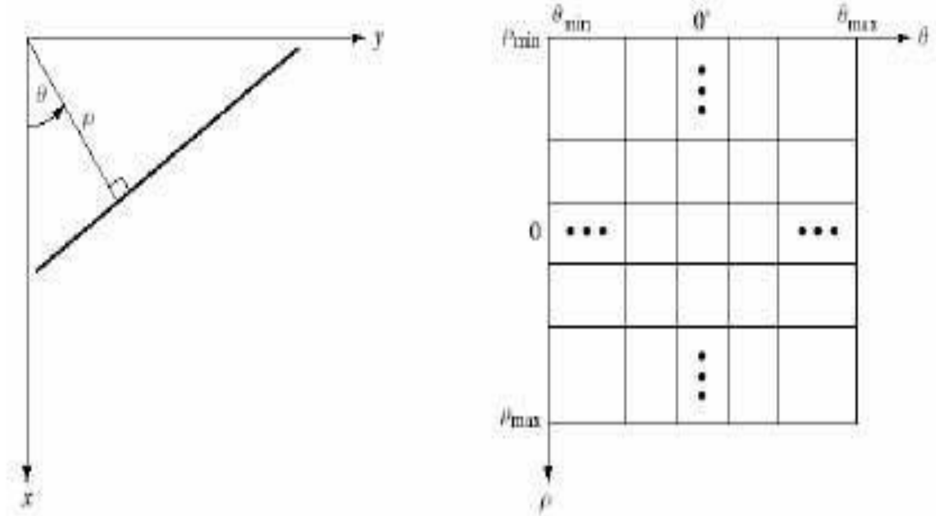
–

- Subdivision of the parametric space into *accumulator cells*
- The cell at $(i,j)$ with accumulator values $A(i,j)$ corresponds to $(ai,bj)$
- For every point $(xk,yk)$ vary $a$ from cell to cell and solve for $b$:
- If $ap$ generates $bq$, then increment the accumulator $A(p,q)=A(p,q)+1$
- At the end of the procedure, a value of $Q$ in $A(i,j)$ corresponds to $Q$ points in the $xy$-plane lying on the line
- $K$ different incremens of $a$ generate $K$ different values of $b$; for $n$ different image point, the method involves $nK$ computations (linear complexity)



## Hough transform: handling the vertical lines

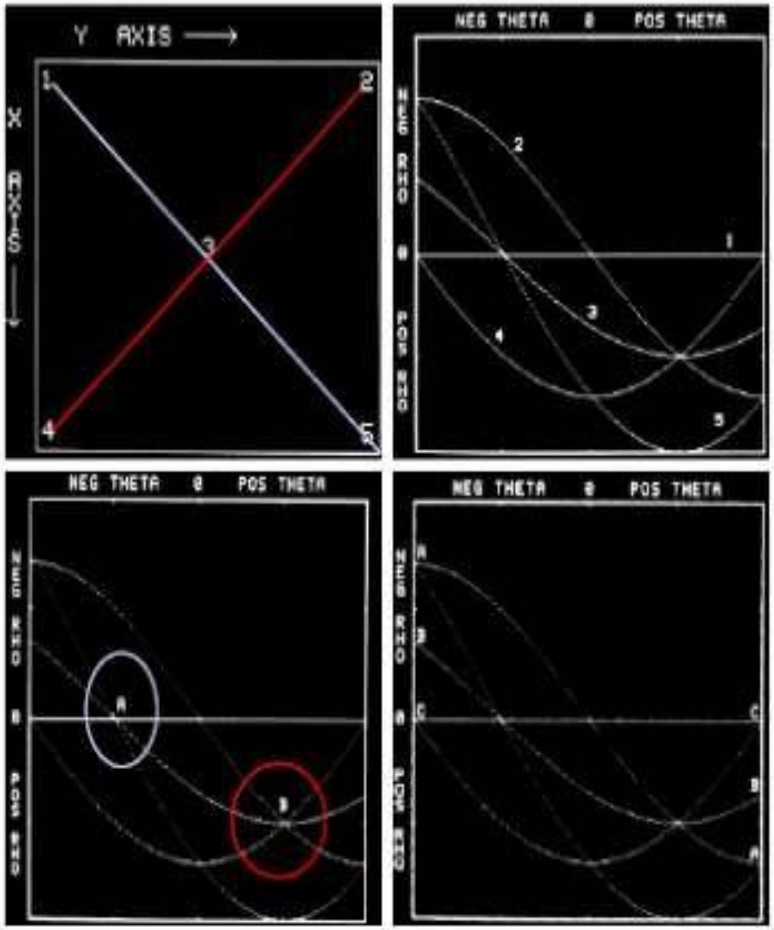
- Through normal representation $x\cos\theta + y\sin\theta = \rho$
- Instead of straight lines, there are sinusoidal curves in the parameter space
- The number or intesecting sinusoids is accumulated and then the value Q in the accumulator $A(i,j)$ shows the number of colinear points lying on a line $x\cos\theta_j + y\sin\theta j = \rho_i$




a b

**FIGURE 10.19** (a) Normal representation of a line. (b) Subdivision of the $\rho\theta$-plane into cells.

- Example: two lines connecting three points each
  - Fig (d) indicates that the Hough transform exhibits a reflective adjacency relationship
- Summary of Hough transform for edge linking
  - Compute the gradient
  - Specify subdivisions in the parametric plane
  - Examine the counts of the accumulator cells
  - Examine the continuity relationship between pixels in a chosen cell



Example: Hough transform removing gaps (no longer than 5 pixels) between edge pixels

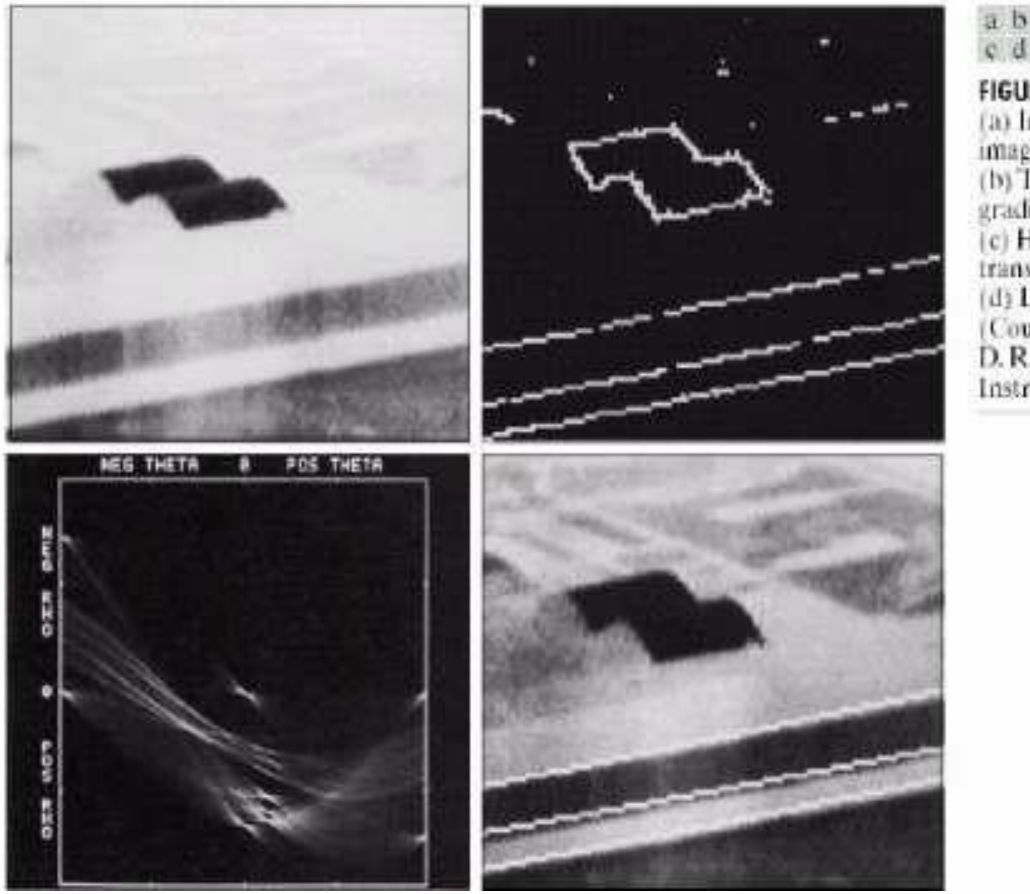


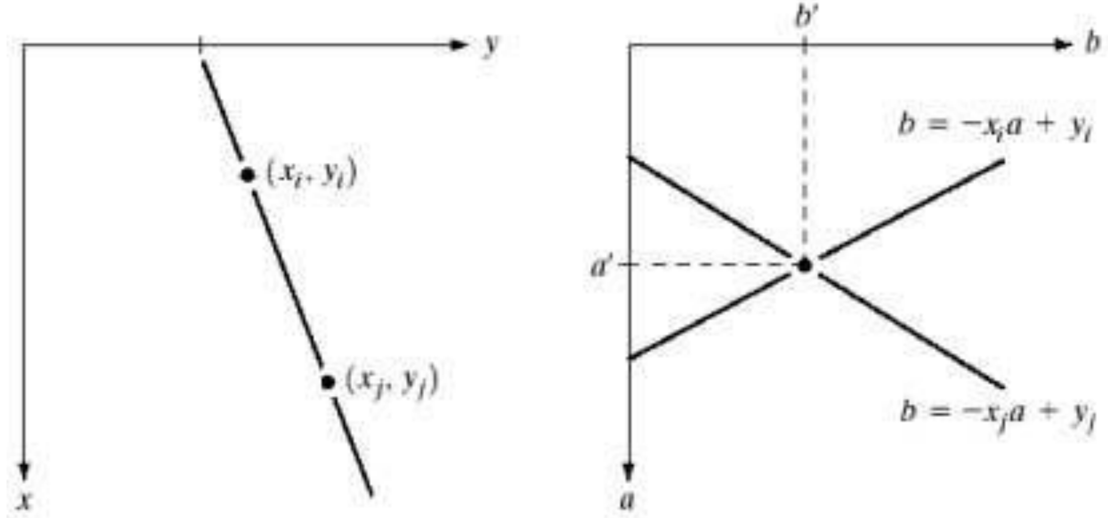

**FIGURE 10.21** (a) Infrared image. (b) Thresholded gradient image. (c) Hough transform. (d) Linked pixels. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

E. Woods

# Summary – Hough Transform

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

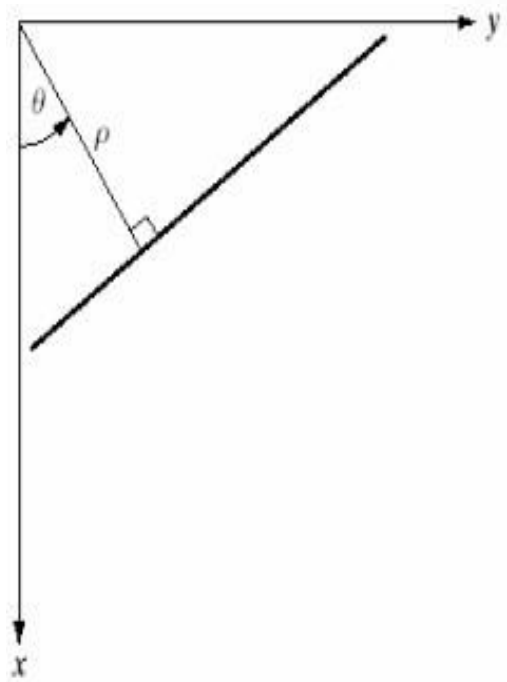
$$y_i = ax_i + b$$

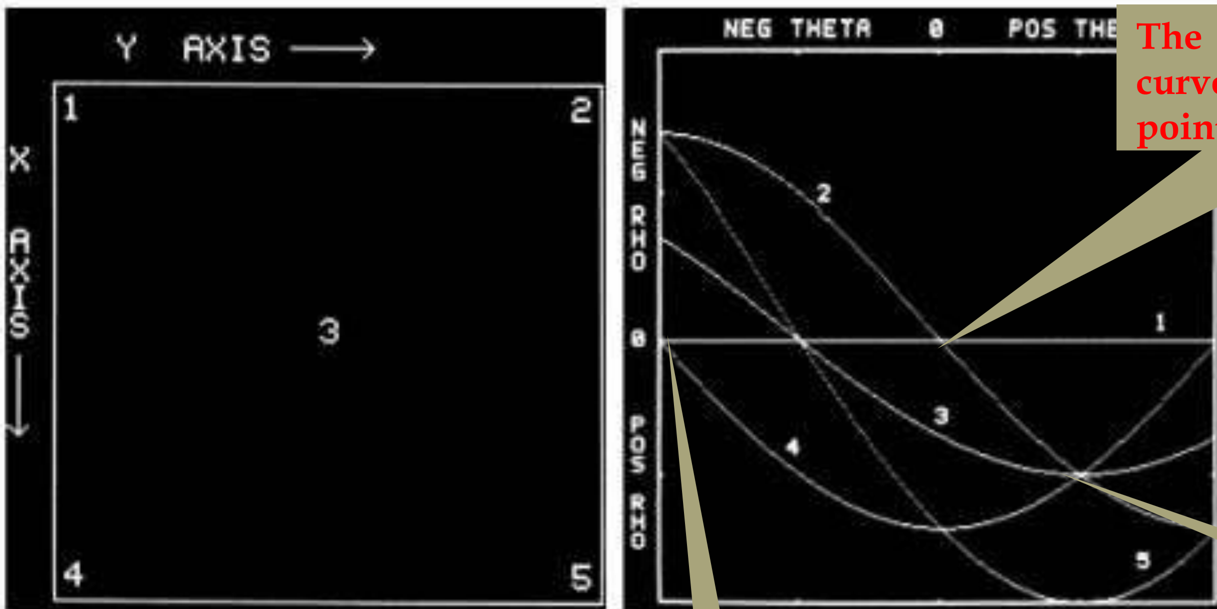



a b

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

- The Hough transform consists of finding all pairs of values of $\theta$ and $\rho$ which satisfy the equations that pass through $(x,y)$. $x\cos\theta + y\sin\theta = \rho$
- These are accumulated in what is basically a 2-dimensional histogram.
- When plotted these pairs of $\theta$ and $\rho$ will look like a sine wave. The process is repeated for all appropriate $(x,y)$ locations.

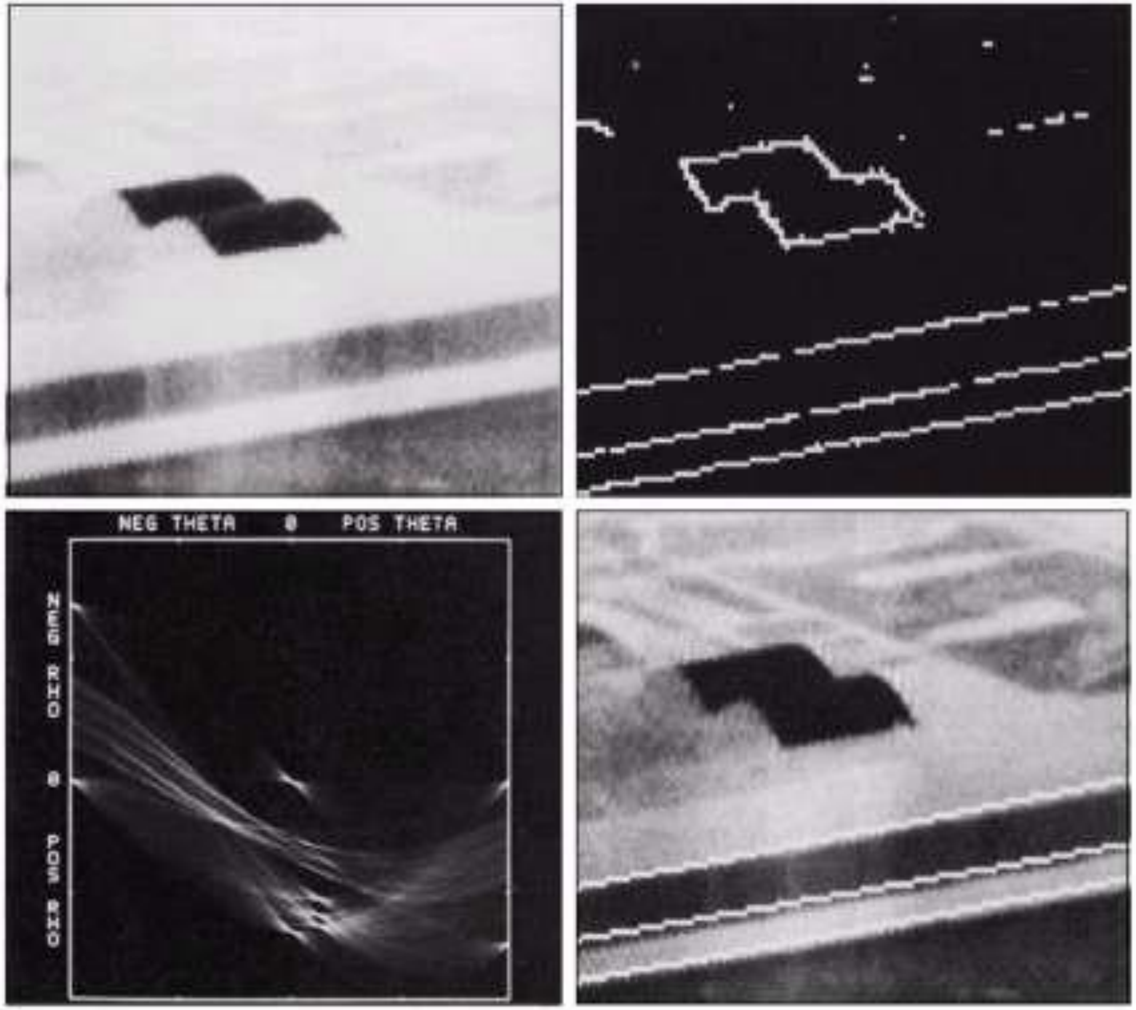





**The intersection of the curves corresponding to points 1,3,5**

2,3,4

1,4




a b
c d

**FIGURE 10.21**
(a) Infrared image.
(b) Thresholded gradient image.
(c) Hough transform.
(d) Linked pixels.
(Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

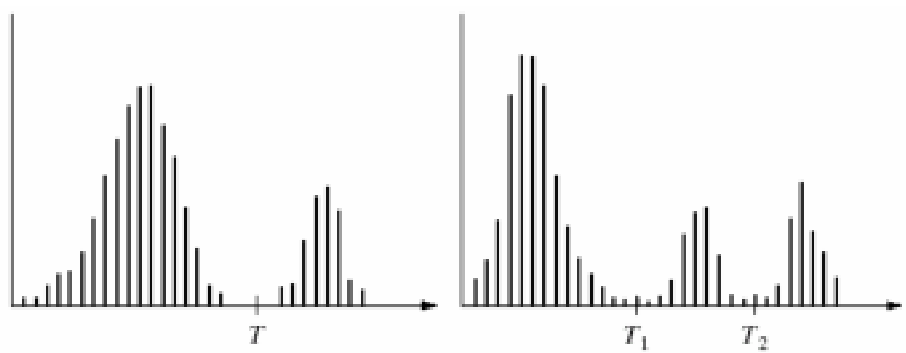
# Thresholding

Foundation

– Histogram dominant modes: two or more

– Threshold and thresholding operation

$$T = T\left[x, y, p(x,y), f(x,y)\right]; \qquad g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \end{cases}$$



**Illumination**

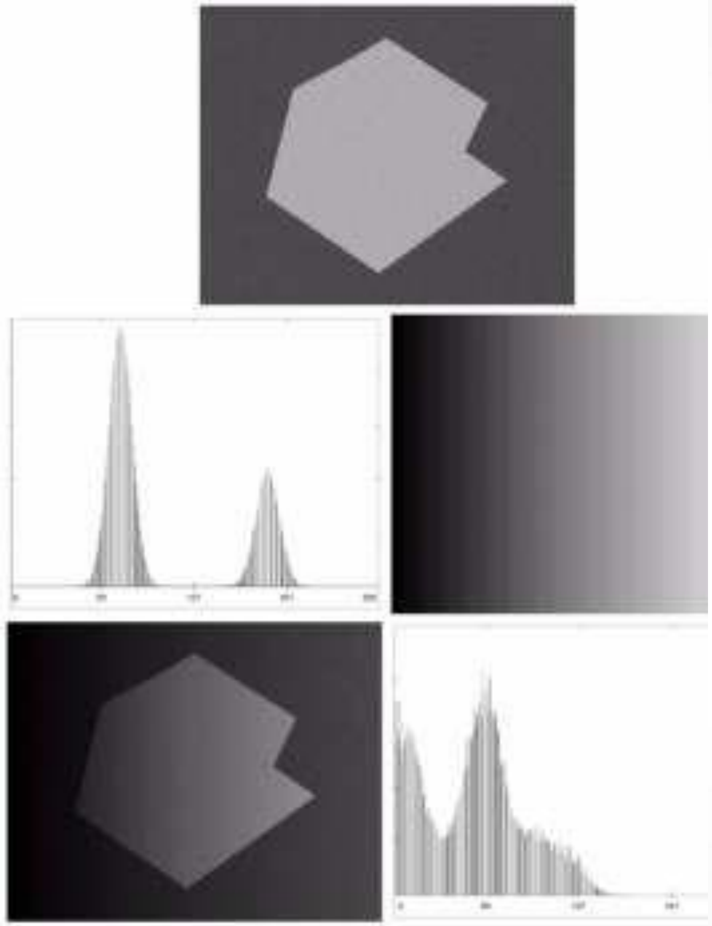
– Image is a product of reflectance and illuminance

– Reflection nature of objects and background

– Poor (nonlinear) illumination could impede the segmentation

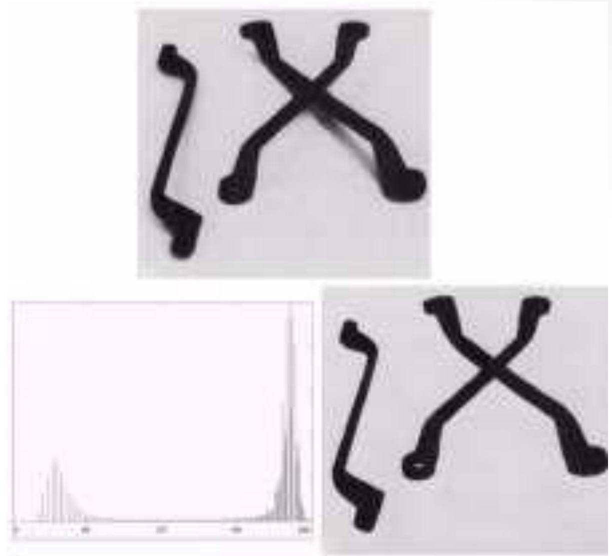
– The final histogram is a result of convolution of the histogram of the log reflectance and log illuminance functions

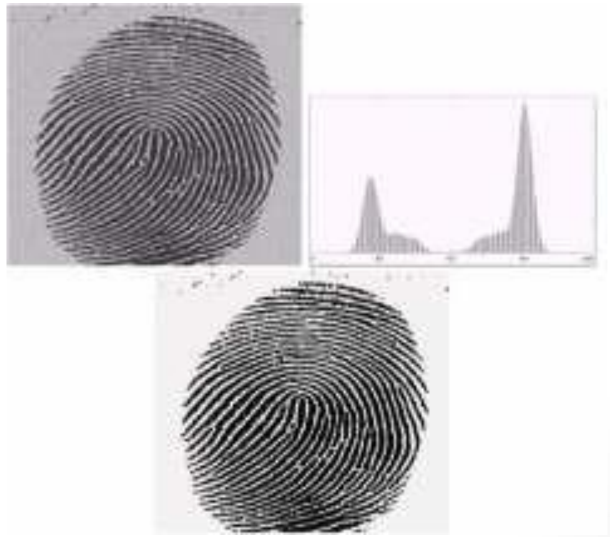
– Normalization if the illuminance function is known



# Basic Global Thresholding

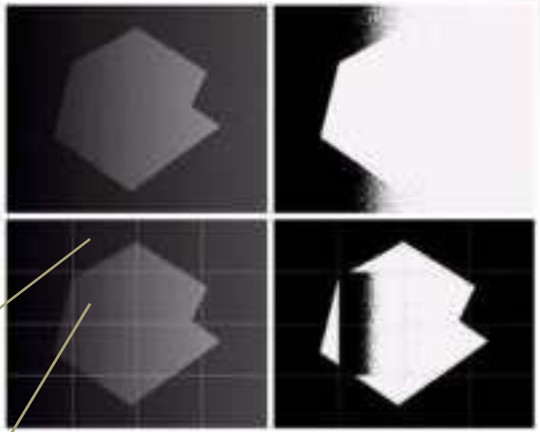
- Threshold midway between maximum and minimum gray levels
- Appropriate for industrial inspection applications with controllable illumination
- Automatic algorithm
  - Segment with initial $T$ into regions $G1$ and $G2$
  - Compute the average gray level $m1$ and $m2$
  - Compute new $T$=0.5($m1$+$m2$)
  - Repeat until reach an acceptably small change of $T$ in successive iterations

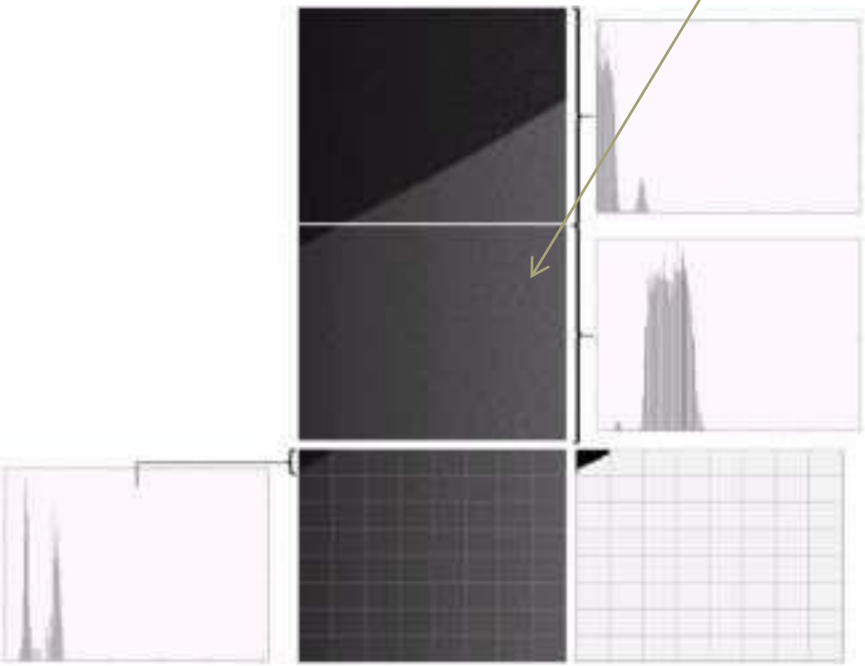




## Basic Adaptive Thresholding

- Divide the image into sub-images and use local thresholds





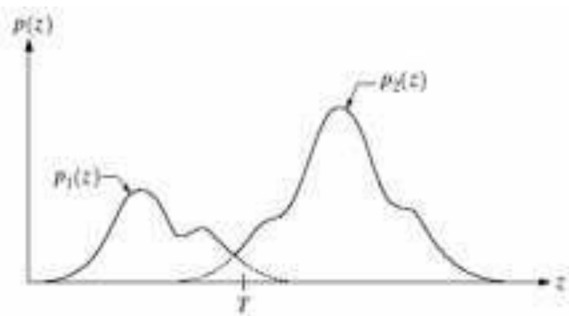
# Optimal Global and Adaptive Thresholding

- Histograms considered as estimates of probability density functions
- Mixure probability $p(z) = P_1 p_1(z) + P_2 p_2(z);\ P_1 + P_2 = 1$
- Select the value of T that minmizes the average error in making the decision that a given pixel belongs to an object of to the background

$$E(T) = P_2 \int_{-\infty}^{T} p_2(z)dz + P_1 \int_{T}^{\infty} p_1(z)dz$$

  - Minimizing the probability of erroneous classification
    - Differentiate the error equation and solve for T $P_2 p_2(T) = P_1 p_1(T)$

- Estimating the densities using simple models, e.g. Gaussian



Boundary Characteristics for Histogram Thresholding

- Consider only pixels lying on and near edges
- Use gradient or Laplacian to preliminary process the image

$$s(x,y) = \begin{cases} 0 & \text{if } \nabla f < T \\ + & \text{if } \nabla f \geq T \quad \text{and } \nabla^2 f \geq 0 \\ - & \text{if } \nabla f \geq T \quad \text{and } \nabla^2 f < 0 \end{cases}$$

- Transition from light background to dark object is characterized (-,+), object interior is coded by either 0 or +, transition from object to background (+,-)

Thresholds based on several variables

- Color or multispectral histograms
- Thresholding is based on finding clusters in multi-dimensional space
- Example: face detection
- Different color models
  - Hue and saturation instead of RGB

# Region based Segmentation

## Basic formulation

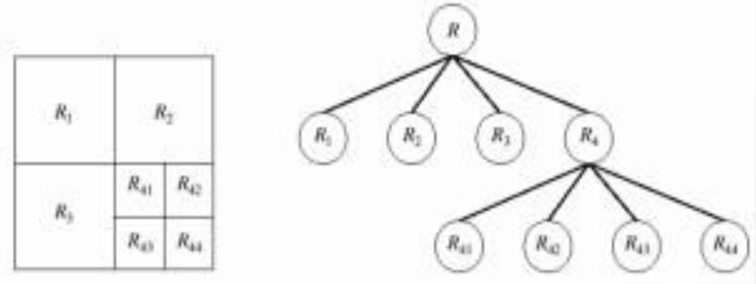
- Every pixel must be in a region
- Points in a region must be connected
- Regions must be disjoint
- Logical predicate for one region and for distinguishing between regions

## Region growing

- Group pixels from sub-regions to larger regions
- Start from a set of 'seed' pixels and append pixels with similar properties
  - Selection of similarity criteria: color, descriptotors (gray level + moments)
  - Stopping rule

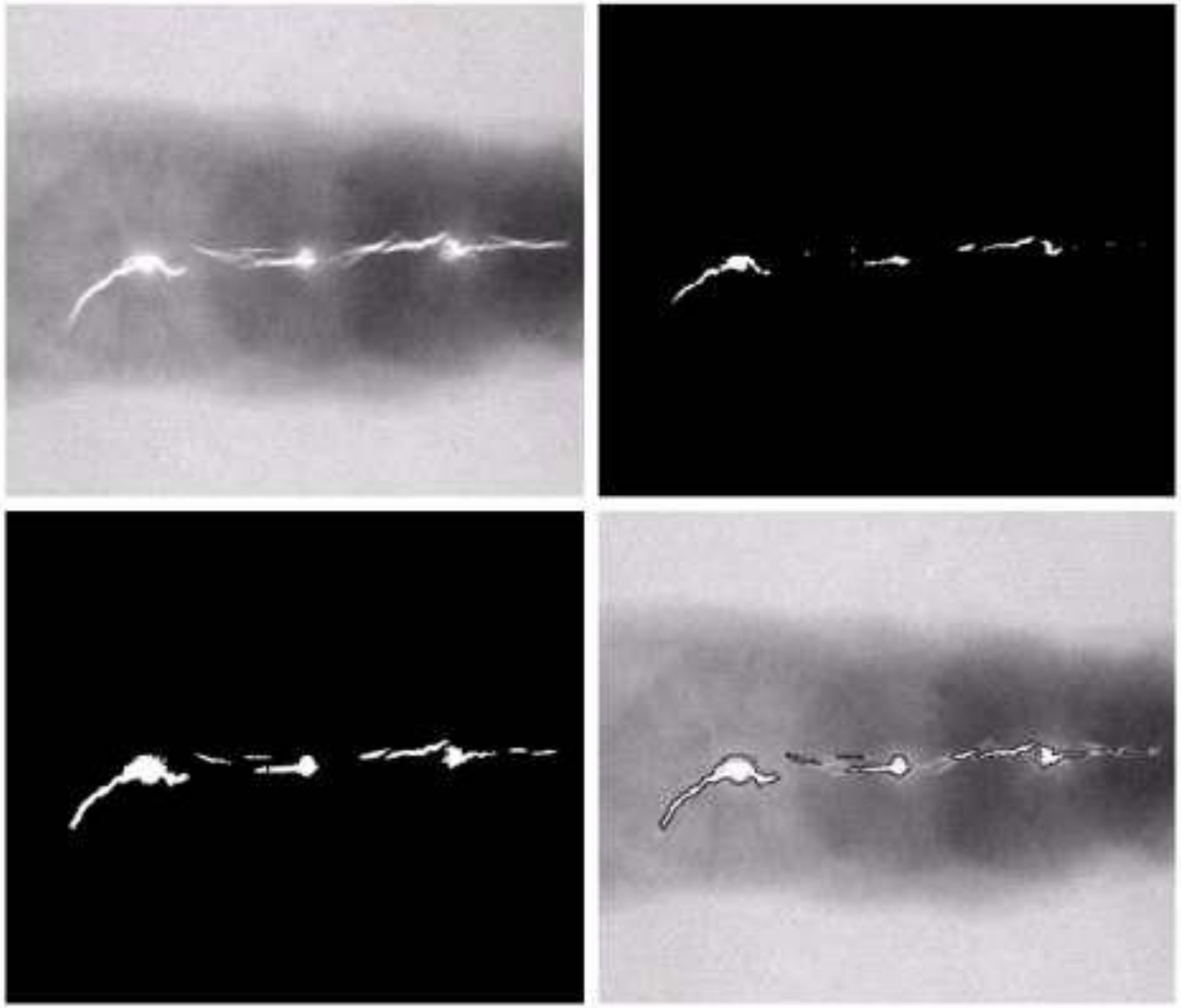
## Region splitting and merging

- Quadtree decomposition



a b
c d

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



- Fig. 10.41 shows the histogram of Fig. 10.40 (a).
- It is difficult to segment the defects by thresholding methods. (Applying region growing methods are better in this case.)

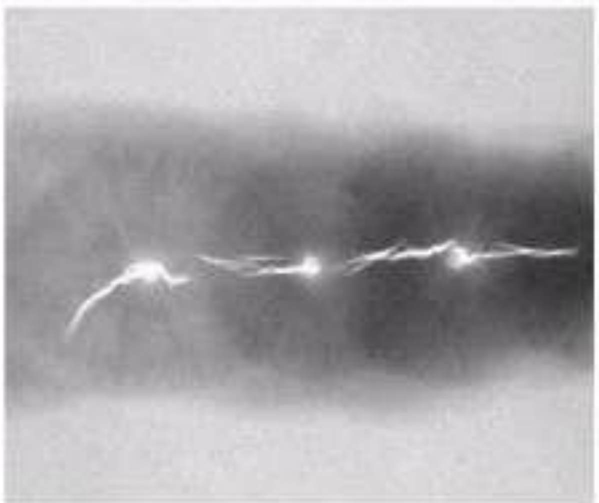


Figure 10.40 (a)

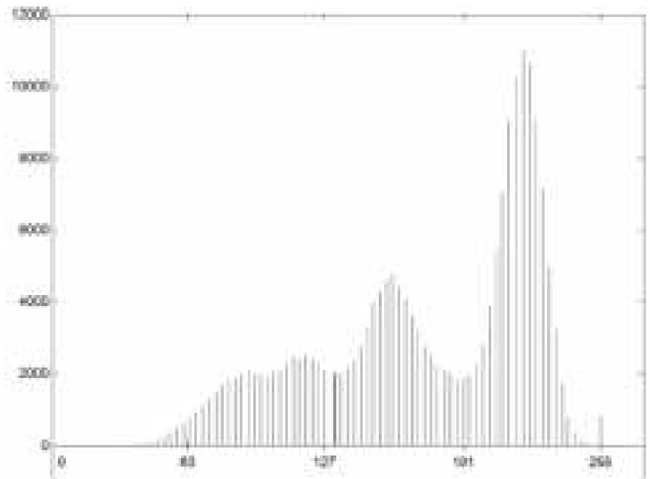



Figure 10.41

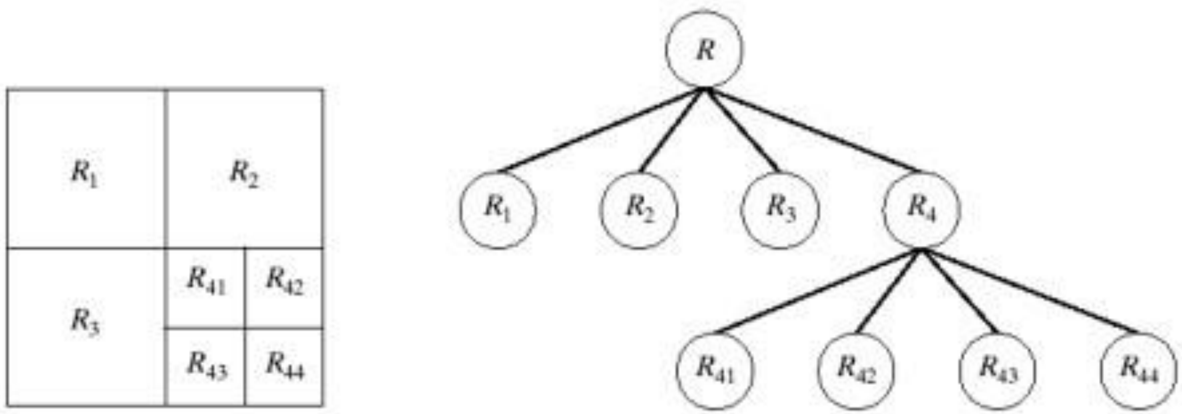
## **Region splitting** is the opposite of region growing

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

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

a b

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



The split and merge procedure:

- Split into four disjoint quadrants any region $R_i$ for which

$$P(R_i) = \text{FALSE}.$$

- Merge any adjacent regions $R_j$ and $R_k$ for which

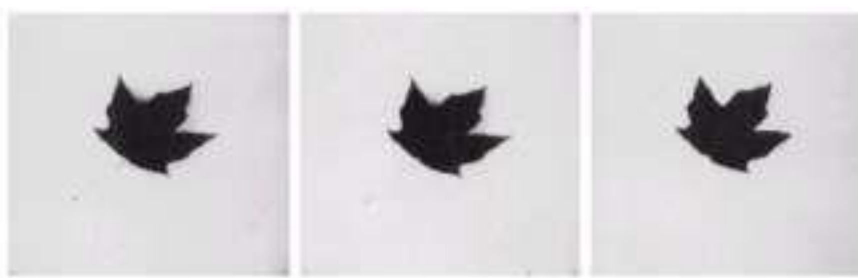
$$P(R_j \cup R_k) = \text{TRUE}.$$

(the quadtree structure may not be preserved)

- Stop when no further merging or splitting is possible.

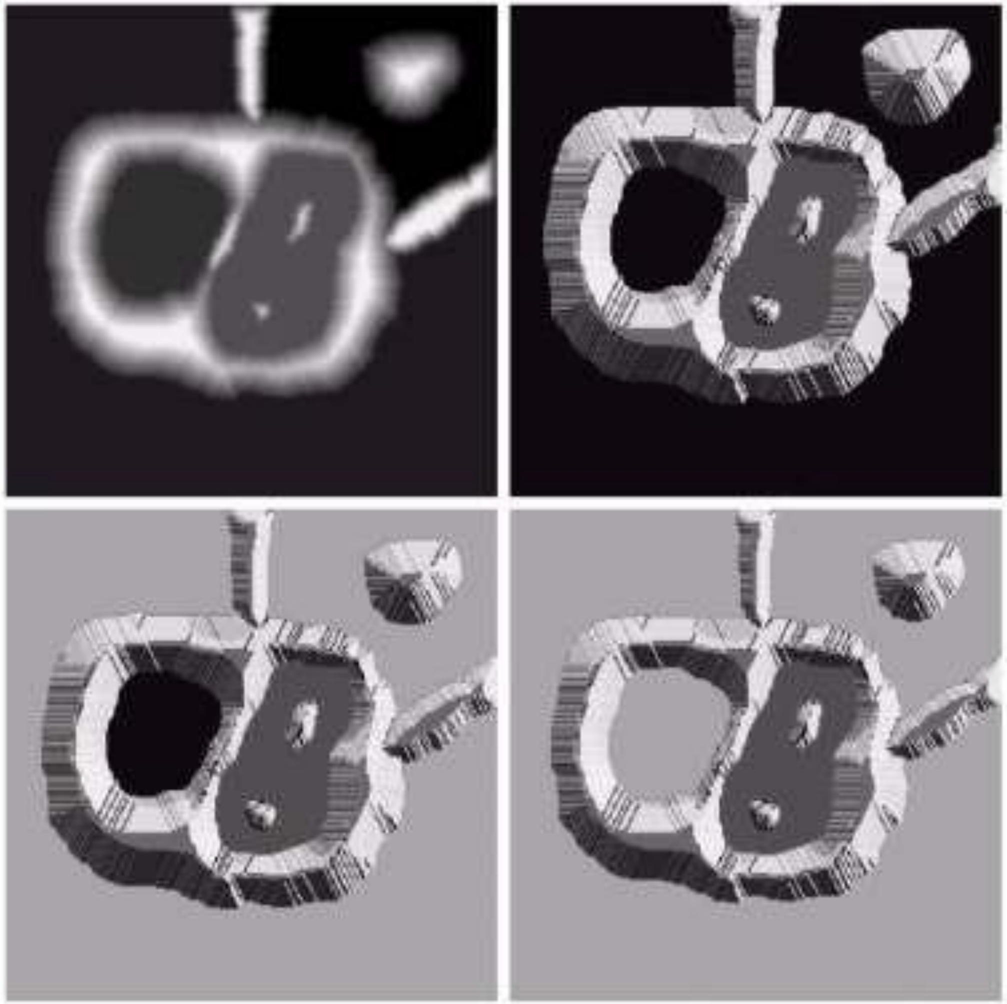
a b c

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



# Segmentation by Morphological Watersheds

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



a b
c d

**FIGURE 10.44**
(a) Original image.
(b) Topographic view. (c)–(d) Two stages of flooding.

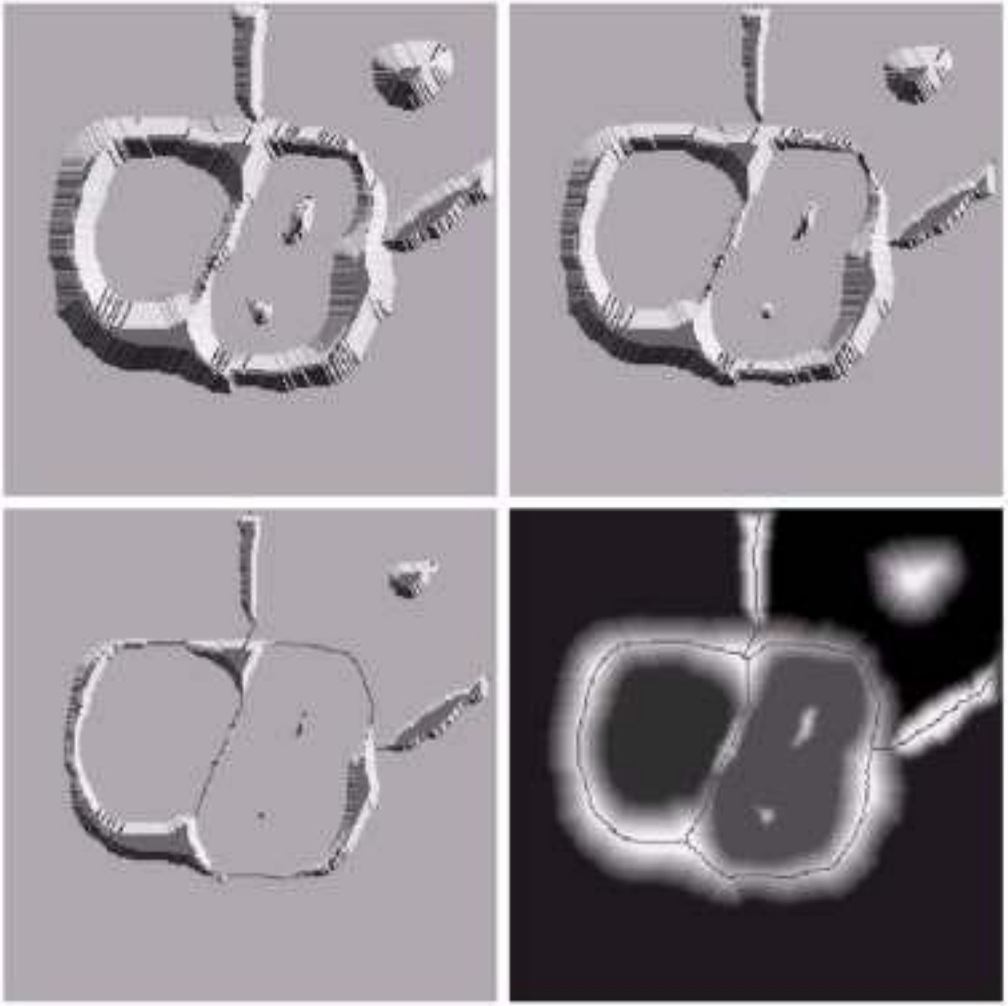



**FIGURE 10.44** *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

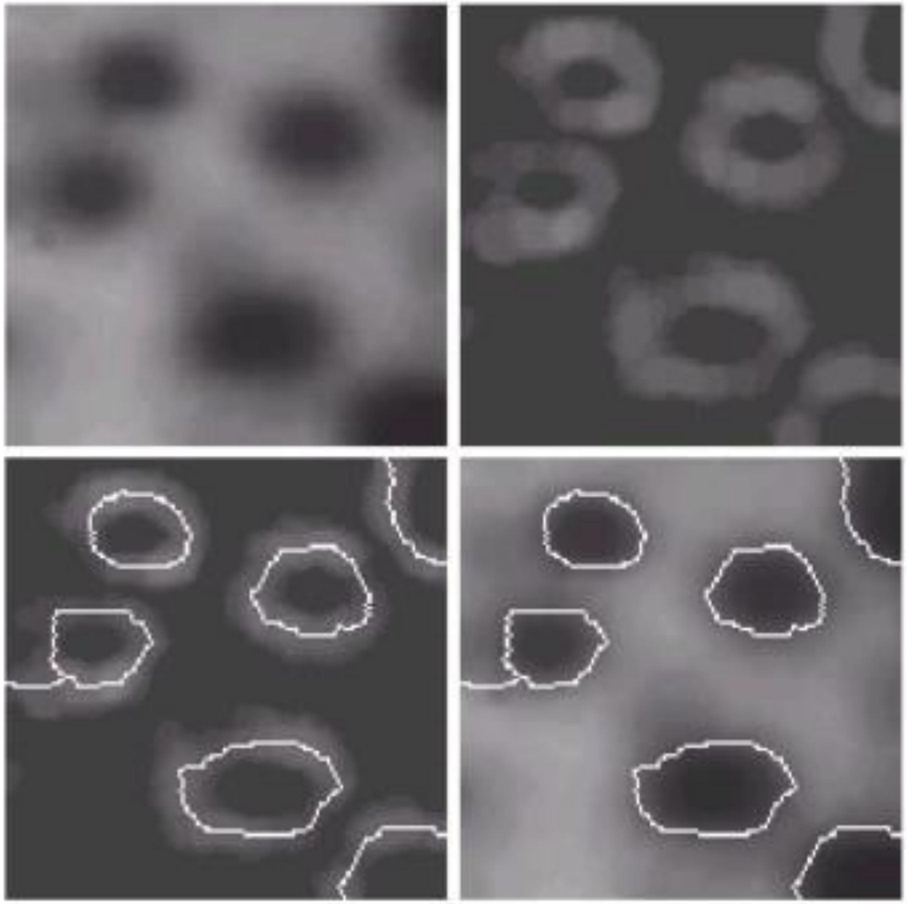



**FIGURE 10.46** (a) Image of blobs. (b) Image gradient. (c) Watershed lines. (d) Watershed lines superimposed on original image. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

# MODULE 5

## Image Segmentation

Algorithms are based on 2 basic categories dealing with properties of intensity values -

1) Discontinuity (Edge based)
2) Similarity (Region based)

Approaches.

### Discontinuity Based

1) Point line & Edge Detection

* Detecting sharp local changes in intensity.
* 3 types of features → Isolated points
  → lines
  → Edges

* Edge Pixels are pixels @ which intensity of an image function changes abruptly.
  Edges → sets of connected edge points.
* Abrupt local changes in intensity can be detected using "Derivatives".
* Derivatives of a digital function are defined in terms of differences.

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

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

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

Conclusion-

1) $1^{st}$ order derivative produces thicker edges in an image.
2) $2^{nd}$ order derivative → strong response to fine detail, such as thin lines, isolated points and noise.
3) $2^{nd}$ order derivative → produces a double-edge response @ ramp and step transitions.
4) Sign of $2^{nd}$ order derivatives → used to detect edge transition. (light to dark or dark to light).

To implement $1^{st}$ & $2^{nd}$ order derivatives -

Use spatial filters of eqn. 3×3 Mask / filter

Response $R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$

$$R = \sum_{k=1}^{9} w_k z_k$$

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

P.T.O.

## Detection of Isolated Points-

Use of second derivative

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

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

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

∴ Laplacian is then:

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

Laplacian Mask

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

or

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

Apply laplacian mask on $f(x,y)$ and obtain the resultant image $g(x,y)$ is obtained using

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise.} \end{cases}$$

↳ * detection of isolated points.

3) Line Detection -

Laplacian detector is isotropic and the response is independent of ~~distor~~ dimension. To detect dimensions -

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 2 | 2 | 2 |
| -1 | -1 | -1 |

Horizontal

| | | |
|---|---|---|
| 2 | -1 | -1 |
| -1 | 2 | -1 |
| -1 | -1 | 2 |

45°

| | | |
|---|---|---|
| -1 | 2 | -1 |
| -1 | 2 | -1 |
| -1 | 2 | -1 |

Vertical

| | | |
|---|---|---|
| -1 | -1 | 2 |
| -1 | 2 | -1 |
| 2 | -1 | -1 |

45°

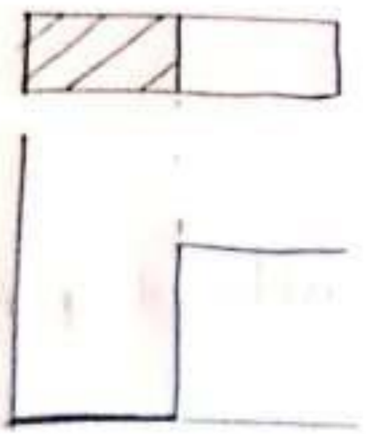
* Preferred direction of each mask is weighted with larger coefficient than other possible dimensions
* Coefficients in each mask sum to zero, indicating a zero response in areas of constant intensity.

4) Edge Models -

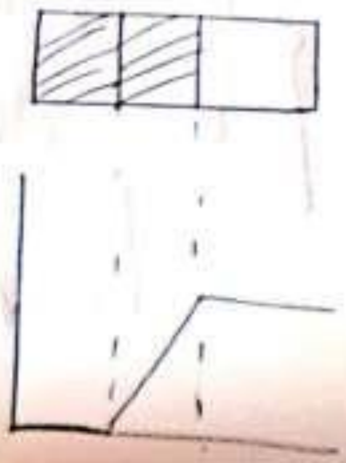
Segmenting images based on abrupt (local) changes in intensity

3 types

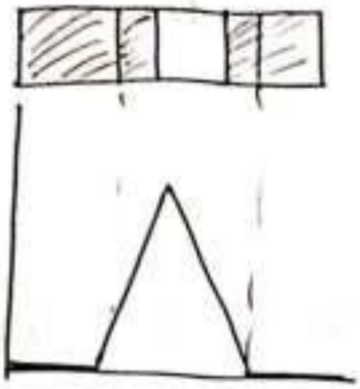
1) Step



2) Ramp



.oof-



Roof edge

<u>Step Edge</u> - transition b/w two intensity levels occuring ideally over the distance of one pixel.

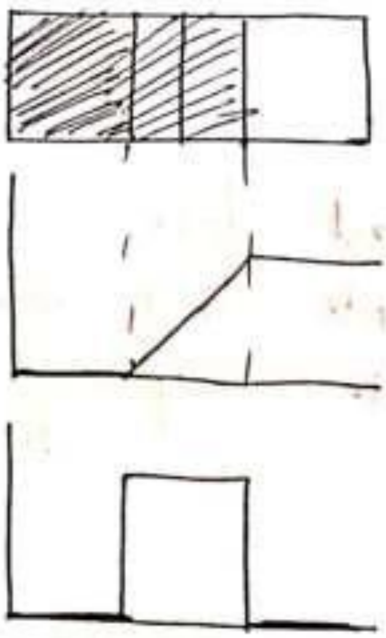
<u>Ramp Edge</u> - In practice, digital images have edges that are blurred and noisy. Edges are more closely modeled as having an intensity Ramp. Slope of Ramp is inversely proportional to degree of blurring the edge

<u>Roof Edge</u> - Models of lines through a region, with the base (width) of a roof edge being determined by the thickness and sharpness of the line.

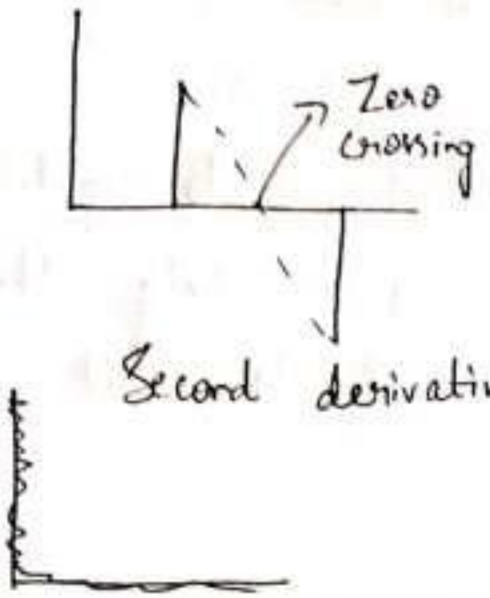
Image strip with Ramp edge 1)

2) Edge Ramp

3) First derivative



4)




Second derivative

* Interaction b/w zero intensity axis and a line extending b/w extreme of $2^{nd}$ derivative points is called "Zero Crossing."

Image

Conclusion -

1) Magnitude of $1^{st}$ derivative used to detect the presence of an edge @ a point.
2) Sign of $2^{nd}$ derivative → used to detect whether edge pixel lies on the dark or light side of an edge.
3) $2^{nd}$ derivative produces two values for every edge.
4) Zero crossings can be used for locating centres of thick edges

Note - $2^{nd}$ derivatives → more sensitive to noise.

* Fundamental Steps in Edge Detection -
  1) Image Smoothing - Noise Reduction
  2) Detection of Edge Points - extracts potential candidates to become edge points.
  3) Edge localization - Select from the candidate edge points only the points that are true members of the set of points comprising an edge.

## Basic Edge Detection -

### Image gradient and its properties -

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}$$

Edge strength and direction at location $(x,y)$ of an image "$f$".

Magnitude (length) of vector $\nabla f$ denoted as $M(x,y)$ -

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

Direction of gradient vector is given by angle $\alpha(x,y)$

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

* Gradient operators -

Computing potential derivatives $\frac{\delta f}{\delta x}$ & $\frac{\delta f}{\delta y}$ at every pixel location in the image.

w.k.t. $g_x = \frac{\delta f(x,y)}{\delta x} = f(x+1,y) - f(x,y)$

$$g_y = \frac{\delta f(x,y)}{\delta y} = f(x,y+1) - f(x,y)$$

i.e.

| | |
|---|---|
| -1 | |
| 1 | |

| | |
|---|---|
| -1 | 1 |

When diagonal edge direction is required.

**2D - Mask**

$$g_x = \frac{\delta f}{\delta x} = z_9 - z_5$$

$$g_y = \frac{\delta f}{\delta y} = z_8 - z_6$$

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

*

| | |
|---|---|
| -1 | 0 |
| 0 | 1 |

| | |
|---|---|
| 0 | -1 |
| 1 | 0 |

Roberts

**3D - Mask**

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

"Prewitt"

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

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

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

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

"Sobel" { have better noise supression

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

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

Prewitt & Sobel for detecting diagonal edges.

| | | |
|---|---|---|
| 0 | 1 | 1 |
| -1 | 0 | 1 |
| -1 | -1 | 0 |

| | | |
|---|---|---|
| -1 | -1 | 0 |
| -1 | 0 | 1 |
| 0 | 1 | 1 |

<u>Prewitt</u>

* emphasizes edges along the diagonal directions

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

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

<u>Sobel</u>

$$M(x,y) = |g_x| + |g_y|$$

⟹ Preserves relative changes in intensity levels

* <u>Combining the Gradient with Thresholding</u> -
* Objective of highlighting the principal edges & maintaining connectivity, smoothing & thresholding is used.
* Threshold the Gradient Image.

6) Advanced techniques for Edge detection -

Taking into account factors such as -

1) Image noise  2) Nature of edges.

The Marr-Hildreth Edge Detector -

→ Previously Edge detection methods were based on using small operators (Sobel masks).

Marr-Hildreth argues that -

1) Intensity changes are not independent of image scales. So, their detection requires the use of operators of different signs.

2) Sudden intensity change will give rise to peak or trough in $1^{st}$ derivative or zero-crossing in $2^{nd}$ derivative.

The operator fulfilling these conditions is the filter "$\nabla^2 G$"

$\nabla^2$ is Laplacian operator, $\left(\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2}\right)$

$G$ is 2D Gaussian function.

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

$\sigma$ → Standard deviation

Space constant.

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

$$\therefore \nabla^2 G(x,y) = \frac{\delta^2 G(x,y)}{\delta x^2} + \frac{\delta^2 G(x,y)}{\delta y^2}$$

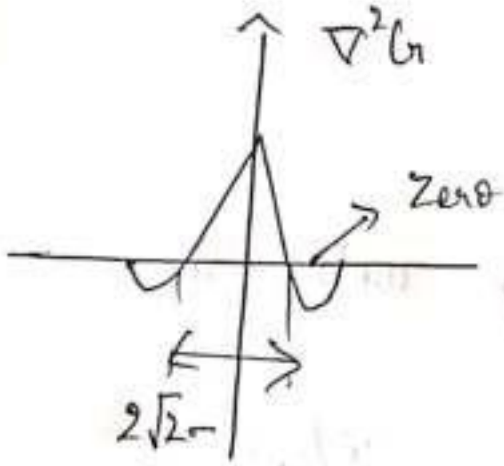
$$= \frac{\delta}{\delta x}\left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] + \frac{\delta}{\delta y}\left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right]$$

$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\therefore \nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Also called as <u>Laplacian of a Gaussian (LoG)</u>

Sometimes called as '<u>Mexican hat operator</u>'




| | | | | |
|---|---|---|---|---|
| 0 | 0 | -1 | 0 | 0 |
| 0 | -1 | -2 | -1 | 0 |
| -1 | -2 | 16 | -2 | -1 |
| 0 | -1 | -2 | -1 | 0 |
| 0 | 0 | -1 | 0 | 0 |

<u>5X5 LoG filter</u>

Algorithm consists of -

1) Convolving LoG filter with an i/P image $f(x,y)$

i.e. $g(x,y) = \nabla^2 G(x,y) * f(x,y)$

$$g(x,y) = \nabla^2 [G(x,y) * f(x,y)]$$

2) Find zero crossings to determine the locations of edges in $f(x,y)$.

Zero crossings are the key features of Marr - Hildreth Edge Detection Method.

* <u>The Canny Edge Detection</u> -

<u>3 basic objectives</u>

→ low error rate

→ Edge points should be well localized.

→ Single edge point response.

<u>Steps</u> -

<u>Step 1</u> - Smoothen the i/P image with a Gaussian filter.

1st derivative of Gaussian $\left\{ \frac{\delta}{\delta x} e^{-x^2/2\sigma^2} = \frac{-x^2}{\sigma^2} e^{-x^2/2\sigma^2} \right.$

Let $f(x,y) \rightarrow$ i/P image

$G(x,y) \rightarrow$ Gaussian function

∴ Smoothened image

$$f_s(x,y) = G(x,y) * f(x,y)$$

Step 2 - Compute gradient magnitude & ~~ang~~ angle to images

$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{g_x}{g_y}\right)$$

$$g_x = \frac{\delta f_s}{\delta x}, \quad g_y = \frac{\delta f_s}{\delta y}$$

*

$M(x,y) \Rightarrow$ generates wide ridges around local Maxima which needs to be supressed to find thin edges

Step 3 - "Apply non-maxima Supression to gradient magnitude image" (~~Thin edges~~) (Step 3 is iterative for all directions)

1) Find the direction $d_k$ that is closest to $\alpha(x,y)$.

2) If the values of $M(x,y)$ is less than atleast one of its two neighbors along $d_k$, let $g_N(x,y) = 0$ (Suppression)

otherwise let $\boxed{g_N(x,y) = M(x,y)}$

Step 4 - Use double Thresholding & Connectivity analysis to detect & link Edges

we use - Hysteresis Thresholding : uses 2 thresholds

$T_L \rightarrow$ low threshold $T_H \rightarrow$ high threshold

We can visualize thresholding operation creating 2 additional images.

$$g_{NH}(x,y) = g_N(x,y) \geq T_H$$

&

$$g_{NL}(x,y) = g_N(x,y) \geq T_L$$

* Initially $g_{NH}(x,y)$ & $g_{NL}(x,y)$ are set to 0.

1) Locate unvisited edge pixel P in $g_{NH}(x,y)$.
2) Mark as valid edge pixels in $g_{NH}$ & all the weak pixels in $g_{NL}(x,y)$ that are connected to "p" using 8 connectivity.
3) If all non-zero pixels in $g_{NH}(x,y)$ have been visited, go to step 4 else return to step 1
4) Set to zero all pixels in $g_{NL}(x,y)$ that were not marked as valid edge pixels.

4) <u>Edge Linking & Boundary Detection</u> -

Ideally $\rightarrow$ Edge detection should yield sets of pixels lying only on edges.

In practice $\rightarrow$ edges will contain noise, breaks & other discontinuities in intensity values.

$\therefore$ Edge detection is typically followed by linking algorithms producing meaningful edges &/or

~~Region~~ region boundaries There are 3 types -
→ Local Processing → Global Processing
→ Regional processing.

7.1 * Local Processing -

* Analyse the characteristics of pixels in a small neighbourhood about ~~to~~ every point $(x,y)$.
* Points that are similar ~~to~~ according to pre-defined criteria are linked.

Two Properties used for establishing Similarities of edge pixels -
1) Strength (Magnitude) 2) The direction.

Let $S_{x,y}$ → Set of co-ordinates of neighborhood centered at $(x,y)$.

An edge pixel with co-ordinate $(s,t)$ in $S_{xy}$ is identified based on

$|M(s,t) - M(x,y)| \leq E$ Positive threshold.

$|\alpha(s,t) - \alpha(x,y)| \leq A$ Angle threshold.

Steps -

1) Compute gradient magnitude & angle arrays $M(x,y)$ & $\alpha(x,y)$ of image $f(x,y)$
2) Form a binary image 'g' where,

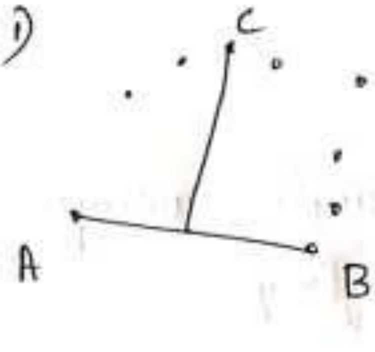
$$g(x,y) = \begin{cases} 1 & \text{if } M(x,y) > T_M \ \& \ \alpha(x,y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

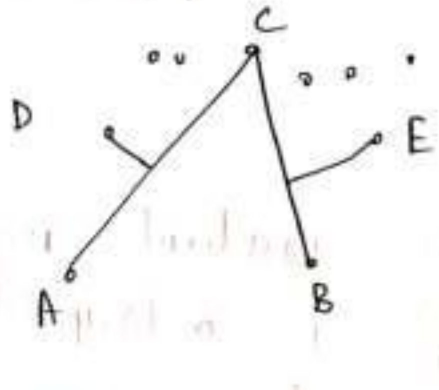
Tm → Threshold of magnitude.
A → Specified angle direction & $\pm T_A$ ] Acceptable band of directions.

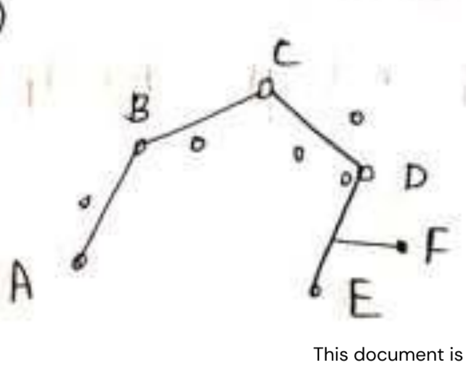
3) Scan the rows of "g" & fill (set to 1) all gaps (set to 0) in each row that do not exceed a specified length k.

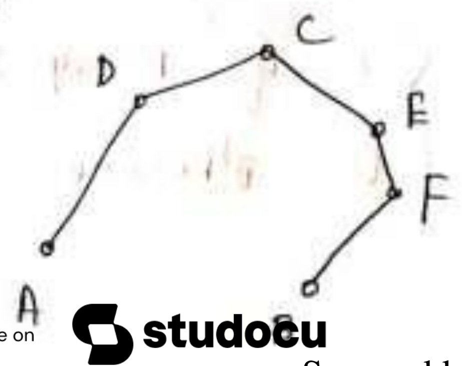
4) To detect gaps in any other direction θ, rotate "g" by this angle(θ) & apply the horizontal scanning in Step 3.
Rotate the result by "−θ".

* <u>Regional</u> <u>Processing</u> -
→ Linking pixels on Regional basis.
→ fit a 2D curve to known points.
→ Polygonal fit to open & closed curves may be stated as follows -

1)



2)



3)



4)


1). Let A & B represents points of open curve.

2) Compute parameters of a straight line passing through A & B.

3) Compute ⊥$^{r}$ distance from all other points to this line.

4) Select the point that yielded the max distance.

5) If distance exceeds specific threshold 'T'

6) Point is labeled 'C' is declared a vertex.

7) Lines from $C \to A$ & $C \to B$ are established.

8) Distance from all points b/w AC, max-distance is declared as vertex D.

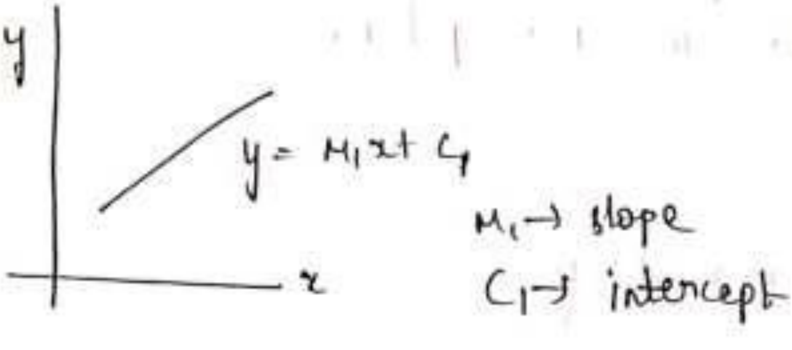
* Repeat for all other points & obtained

$A \to D \to C \to E \to F \to B$.

7.2 * If edge points are outside local neighbors

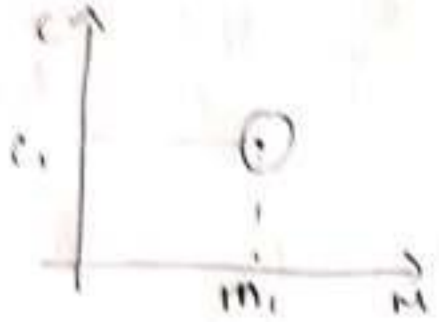
Global Processing using "Hough Transform"

It is mapping from spatial domain to a parameter space.



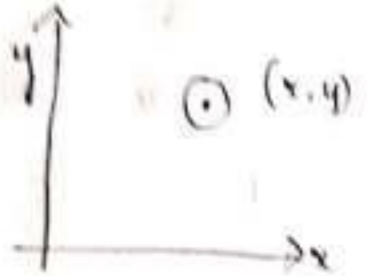
Let $y = m_1 x + c_1$ be line in $xy$ plane

Try to map straight line in parameter space is $m_1$ & $c_1$.



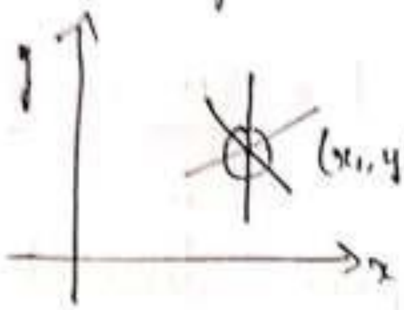

* Straight line looks like single point in parameter space (cm plane).

* Suppose we have given point (x,y) in xy-plane




* If $y = mx + c$ is straight line passing through the point (x,y) then it should satisfy
$y_1 = mx_1 + c$.

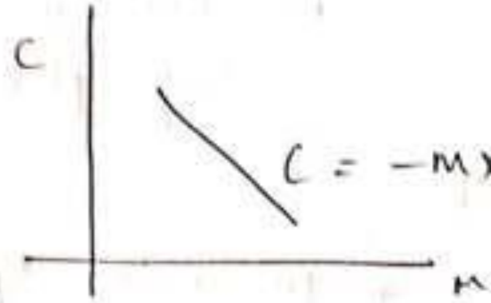
* Ideally we can have infinite distance of straight line passing through given point $(x_1, y_1)$.




→ but m & c will have different values.
→ $x_1$ & $y_1$ are constants.
m & c are variables.

$\therefore \quad c = -mx_1 + y_1$

straight line in mc plane.

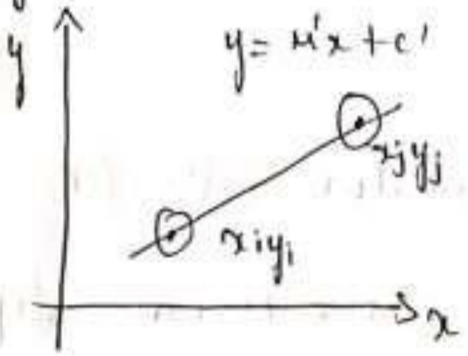



## 2nd Observation -

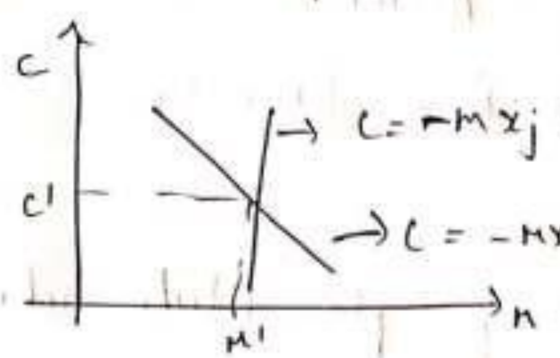
i) Straight line in xy plane is mapped to a point in mc plane.

2) Point in xy plane is mapped into straight line in mc plane.

* If we have 2 points $(x_i, y_i)$ & $(x_j, y_j)$ in xy plane. & straight line through these pts $y = m'x + c'$ should satisfy $y = m'x + c'$.

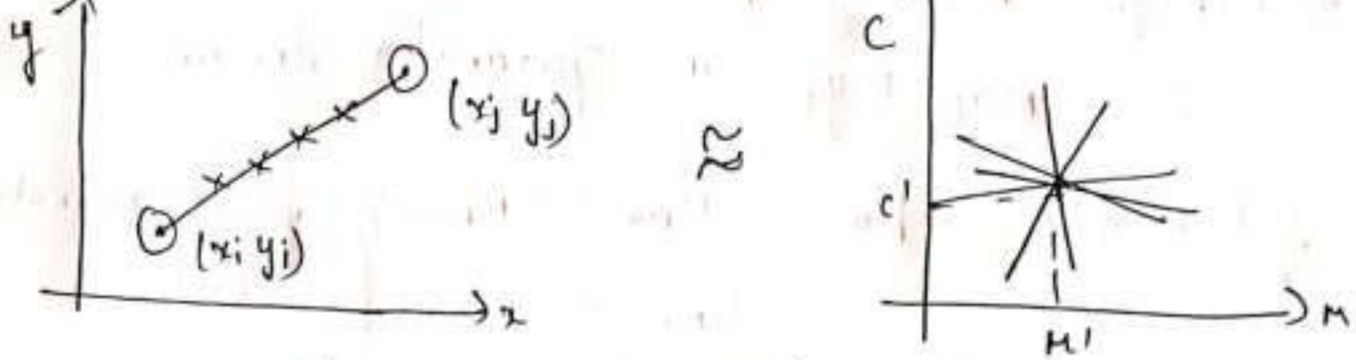



w.r.t. points in xy-plane are mapped into straight line in mc-plane as shown.

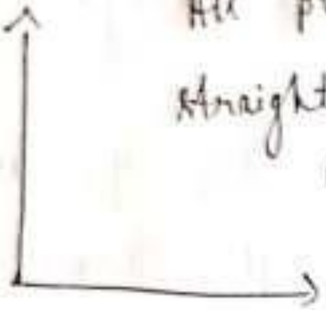



⇒ we find m' & c' are nothing parameters in xy plane.

* If large no. of pts in xy plane -



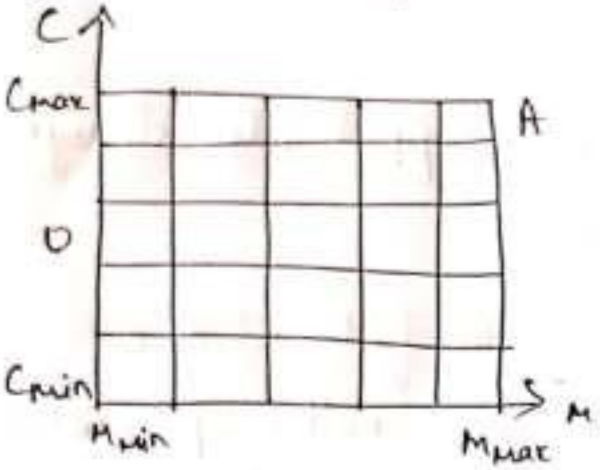

Note - If we have more no. of collinear pts in xy plane will be mapped to a single straight line in ~~parallel~~ parameter space (mc plane) at single pt $(m', c')$



All pts in xy plane mapped into straight lines passing through $(m', c')$ in mc-plane

<u>To</u> <u>Edge</u> <u>linking</u> –

1) Divide MC plane into Accumulator cells. (A)




$\Rightarrow M_{min} < m < M(max)$ slope

$\Rightarrow C_{min} < C < C_{max}$ intercept

A → Accumulator cell

$A(i,j) \Rightarrow (M_i, C_j)$.

2) Initialize $A(i,j) \leftarrow 0$.

3) We have set of boundary pts (tube linked) in spatial domain.

4) $(x_k\ y_k)$ boundary points in <u>spatial domain</u>

$\therefore \ C = -Mx_k + y_k$ in <u>parameter domain</u>

<u>find m & c</u> b/w $\left.\begin{array}{l} M_{min} \to M_{max} \\ C_{min} \to C_{max} \end{array}\right\}$ from accumulator cell

Suppose we have $M_p$ & equivalent $C_q$.
Corresponding increment Accumulator cell is

$A(p,q) = A(p,q) + 1$.

Repeat for all Boundary points.

c) <u>At the end</u>, if $A(i,j) = k$ (no of points)

We get $m_i$ & $c_j$ : $y = m_i x + c_j$

<u>Eg</u> -


5 pts → St line
2 pts
5 pts
4 pts

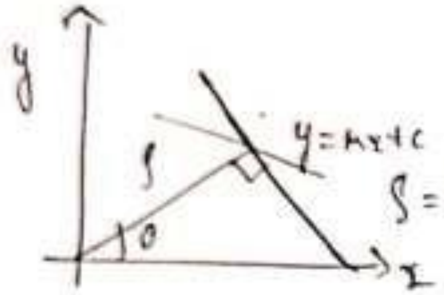
<u>Image & Boundary Pts</u>

* Depending upon Accumulator value → <u>St lines</u> are detected.

<u>Special</u> <u>Case</u> - $y = mx + c$ (If $m$ is very large $\simeq \infty$)

use $\boxed{\rho = x\cos\theta + y\sin\theta}$

Ref of St line in normal form.

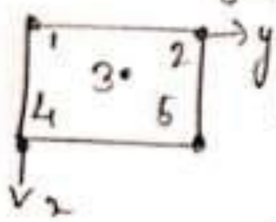



$\rho = x\cos\theta + y\sin\theta$

Now parameters are <u>$\rho$ & $\theta$</u>.

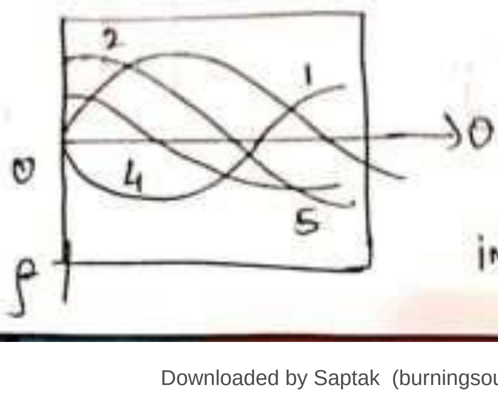
where $\theta = \pm 90°$

$\rho = \sqrt{M^2 + N^2}$




$M \times N$ → Image Size

Any pt in xy plane mapped onto sinusoidal lines in $\rho\ \theta$ plane.




Find out how many lines intersect at a pt.

Step 5 - Hough transform

1) Obtain Binary edge image using any of the techniques.

2) Specify subdivisions in $\rho\theta$ plane
   * Map points of xy plane onto Sinusoidal lines in parametric $(\rho\theta)$ plane

3) Examine the counts of the accumulator cells for high pixel concentrations.

4) Examine the relationship b/w pixels in a chosen cell.

[3] Thresholding -

Advantages - 1) Simplicity of Implementation
2) Computational Speed

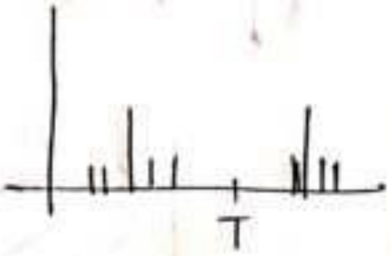
3.1) Foundation -

A) Intensity thresholding -

$$\text{Segmented image} \rightarrow g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) < T. \end{cases}$$

* T is Threshold.

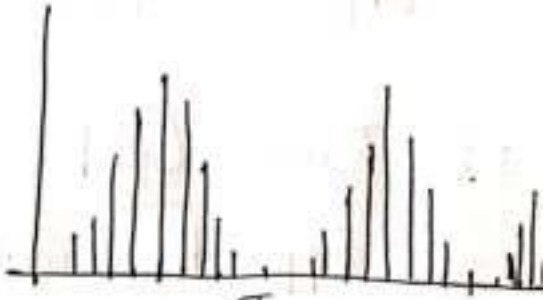
* $f(x,y) \rightarrow$ i/p image



if "T" varies → variable Thresholding
→ local or regional thresholding
→ Dynamic thresholding.

Multiple Thresholding -

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) > T_2 \\ b & \text{if } T_1 < f(x,y) \leq T_2 \\ c & \text{if } f(x,y) \leq T_1 \end{cases}$$




Note- Key factors affecting Intensity Thresholding -

1) Seperation b/w peaks.
2) noise content
3) Relative sizes of objects & background.
4) Uniformity of illumination source.
5) Reflectance properties of image.

B) Role of Noise -

Placing Threshold value in Histogram is a trivial task.

Noise increases the width of Histogram, sometimes Histograms become more broaden that it becomes more difficult to find the depth of valley & to locate/place Threshold point to differentiate two modes.

c) Role of illumination & Reflectance -

* Non-uniform illumination generates shading Deep valleys b/w peaks will be corrupted in Histogram
* Reflectance induces variations in the Surface of objects & / or background.

3.2) Basic Global thresholding -

→ Applicable when intensity distributions of objects & background pixels are sufficiently distinct.

→ Single (global) threshold applicable over entire image

Algorithm -

1) Select an initial estimate of global threshold T.
2) Segment image using T. This will produce 2 group of pixels $G_1$ $[f(x,y) > T]$
$G_2$ $[f(x,y) \leq T]$
3) Compute average (mean) values $m_1$ & $m_2$ for pixels in $G_1$ & $G_2$.
4) Compute a new threshold $T = \frac{1}{2}(m_1 + m_2)$
5) Repeat step 2 through 4, until difference b/w values of T in successive iterations is smaller than predefined "$\Delta T$".

3.3) Optimum Global Thresholding using Otsu's Method.-

Steps- 1) Compute the normalized Histogram of i/p image "$P_i$" where $i = 0, 1, 2, \ldots, L-1$.

2) Compute cumulative sums $P_1(k)$ for $k = 0, 1, 2, \ldots, L-1$ using

$$\boxed{P_1(k) = \sum_{i=0}^{k} P_i}$$

3) Compute cumulative means $m(k)$ for $k = 0, 1, 2, \ldots, L-1$ using

$$\boxed{m(k) = \sum_{i=0}^{k} i P_i}$$

4) Compute Global intensity mean, $m_G$ using

$$\boxed{m_G = \sum_{i=0}^{L-1} i P_i}$$

5) Compute b/w class variance $\sigma_B^2(k)$ for $k = 0, 1, 2, \ldots, L-1$.

***

$$\boxed{\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)\,[1 - P_1(k)]}}$$

6) Obtain Otsu threshold, $k^*$ as value of $k$ for which $\sigma_B^2(k)$ is maximum.

[Maximize b/w class variance]

7) Obtain seperability measure, $\eta^*$ using

$$\boxed{\eta(k) = \frac{P\,\sigma_B^2(k)}{\sigma_G^2}}$$ where $k = k^*$

3.4] Using Image Smoothing to Improve Global Thresholding -

When noise cannot be reduced at the source & thresholding is segmentation method of choice, use image smoothing prior to Thresholding.

Since smoothed image was caused by blurring of boundary " ~~Poor~~ & "Poor Segmented result"

3.5] Using Edges to Improve Global thresholding -

Steps - 1) Compute an Edge image [using any methods]

2) Specify a threshold value 'T'.

3) Threshold the image from Step 1. using threshold from step 2. to produce binary image $g_T(x,y)$. Image $g_T(x,y)$ is used as mask image in Step 4 to select pixel from $f(x,y)$. corresponding to "strong" edge pixels.

4) Compute Histogram using only pixels in $f(x,y)$ correspond to locations of $g_T(x,y)$.

5) Use Histogram from Step 4 to segment $f(x,y)$ globally Eg using, Otsu's Method.

3.6]. Multiple Thresholds -

In case of $K$ classes, $C_1, C_2, \ldots C_K$ b/w the class variance can be generalized as -

$$\sigma_B^2 = \sum_{K=1}^{K} P_K (m_K - m_G)^2$$

where $$P_K = \sum_{i \in C_K} P_i$$

$$m_K = \frac{1}{P_K} \sum_{i \in C_K} i P_i$$

K classes are seperated by K-1 thresholds whose values $k_1^*, k_2^*, \ldots k_{K-1}^*$ that maximizes

$$\sigma_B^2 (k_1^*, k_2^*, k_3^*, \ldots, k_{K-1}^*) = \max \sigma_B^2 (k_1, k_2, \ldots k_{K-1})$$

$$0 < k_1 < k_2 \ldots k_{n-1} < L-1$$

w.k.t.

$$P_1 = \sum_{i=0}^{k_1} P_i \qquad m_1 = \frac{1}{P_1} \sum_{i=0}^{k_1} i P_i$$

$$P_2 = \sum_{i=k_1+1}^{k_2} P_i \qquad m_2 = \frac{1}{P_2} \sum_{i=k_1+1}^{k_2} i P_i$$

$$P_3 = \sum_{i=k_2+1}^{L-1} P_i \qquad m_3 = \frac{1}{P_3} \sum_{i=k_2+1}^{L-1} i P_i$$

Find the optimum thresholds by finding

$$\sigma_B^2 (k_1^*, k_2^*) = \max_{0 < k_1 < k_2 < L-1} \sigma_B^2 (k_1, k_2)$$

Threshold image is given by,

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) \leq K_1^* \\ b & \text{if } K_1^* < f(x,y) \leq K_2^* \\ c & \text{if } f(x,y) > K_2^* \end{cases}$$

& seperability measure-

$$\eta(K_1^*, K_2^*) = \frac{\sigma_B^2 (K_1^*, K_2^*)}{\sigma_G^2}$$

## 3.7] Variable Thresholding -

Sometimes it is impractical or simply ineffective using smoothing & edge information. Next level of thresholding complexity involves variable thresholding.

1) Image partitioning
2) Based on local image properties.
3) Using moving Averages.

### 1) Image Partitioning -

* Subdivide an image into non-overlapping rectangles.
* It is used to compensate for non-uniformities in illumination and/or reflectance.
* Rectangles choosen are small enough.
* Works well when objects of interest & background occupy regions of reasonably comparable size.

2) Based on local image properties -

Let $\sigma_{xy}$ & $m_{xy}$ ⇒ Std. deviation & mean value of set of pixels contained @ $S_{xy}$ in neighborhood "$S_{xy}$".

Following are common forms of variable local Threshold.

$$T_{xy} = a\sigma_{xy} + b\mu_{xy}$$

& $$T_{xy} = a\sigma_{xy} + b\mu_{G}$$

a & b are non-negative constants

$\mu_G$ → global image mean.

Segmented image is computed as-

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T_{xy} \\ 0 & \text{if } f(x,y) \leq T_{xy}. \end{cases}$$

$f(x,y)$ is the i/p image. This eqn. is evaluated for all pixel locations in the image. & different threshold is computed at each location $(x,y)$ using pixels in neighbourhood $S_{xy}$.

3) Using Moving Averages -

* Based on computing a moving average along scan lines of an image.
→ Scanning line in a zigzag pattern to reduce illumination bias.

* Let $Z_{k+1}$ denote the intensity of the point in scanning sequence at step $k+1$.
* The moving average (mean intensity) at this new point is given by -

$$M(k+1) = \frac{1}{n} \sum_{i=k+2}^{k+1} Z_i$$

$$= M(k) + \frac{1}{n} \left( Z_{k-1} + \cdots Z_{k-n} \right)$$

where $n$ denotes no. of points used in computing average and $M(1) = Z_1/n$.

* Moving average is computed for every point in image.

3.8] <u>Multivariable Thresholding</u> -

⇒ So far, thresholding was based on a single variable: gray scale intensity.

⇒ In color imaging, we have R, G & B components.

⇒ Each pixel is characterized by three values & represents as a 3D vector $Z = (Z_1, Z_2, Z_3)^T$ ~~(RGB color at point)~~ (RGB color at point).

⇒ 3D points often referred to as voxels denotes volumetric elements.

⇒ Multivariable Thresholding is viewed as distance computation

$$\therefore \quad g = \begin{cases} 1 & \text{if } D(z, a) < T \\ 0 & \text{otherwise} \end{cases}$$

$D(z, a) \rightarrow$ Distance b/w an arbitrary color point $z$ & average color $a$.

<u>Eg</u> - Let "a" be average reddish color

<u>n-Dimensional Euclidean distance</u> -

$$D(z, a) = \|z - a\|$$

$$D(z, a) = \left[(z-a)^T (z-a)\right]^{1/2}$$

$$D(z, a) = \left[(z-a)^T C^{-1} (z-a)\right]^{1/2}$$

$C \rightarrow$ covariance matrix.

4] <u>Region Based Segmentation</u> -

Find regions based on discontinuities in intensity levels at boundarys.

1) Region growing 2) Region Splitting & Merging.

4.1] <u>Region Growing</u> -

It is procedure that groups pixels on sub-regions into larger regions based on predefined criteria for growth.

<u>Approach</u> - Start with a set of seed points & from these grow regions by appending ~~to~~

each seed those neighboring pixels that have predefined properties similar to seed.

Steps - based on 8-connectivity

1) Let $f(x,y) \rightarrow$ i/p image array

$s(x,y) \rightarrow$ seed array

f & S are assumed to be at same size.

$s(x,y)$ contains 1's at locations of seed points & 0's elsewhere

2) Let Q denote a predicate to be applied to each location $(x,y)$.

3) Find all connected components in $S(x,y)$ & erode each cc to one pixel.

Label all such pixels found as $\rightarrow 1$

& other pixel in S as $\rightarrow 0$.

4) Form an image to such that

let $f_Q(x,y) = 1$ if i/p image satisfies 'Q'

otherwise $f_Q(x,y) = 0$.

5) Let "g" be an image formed by appending each seed point in "S" all 1-valued points in $f_Q$. that are 8-connected to that seed point

6) label each connected component in "g" with a different region label. {Eg 1, 2, 3, .. }.

7). $\boxed{g(x,y) \text{ is Segmented image}}$ obtained by Region Growing.

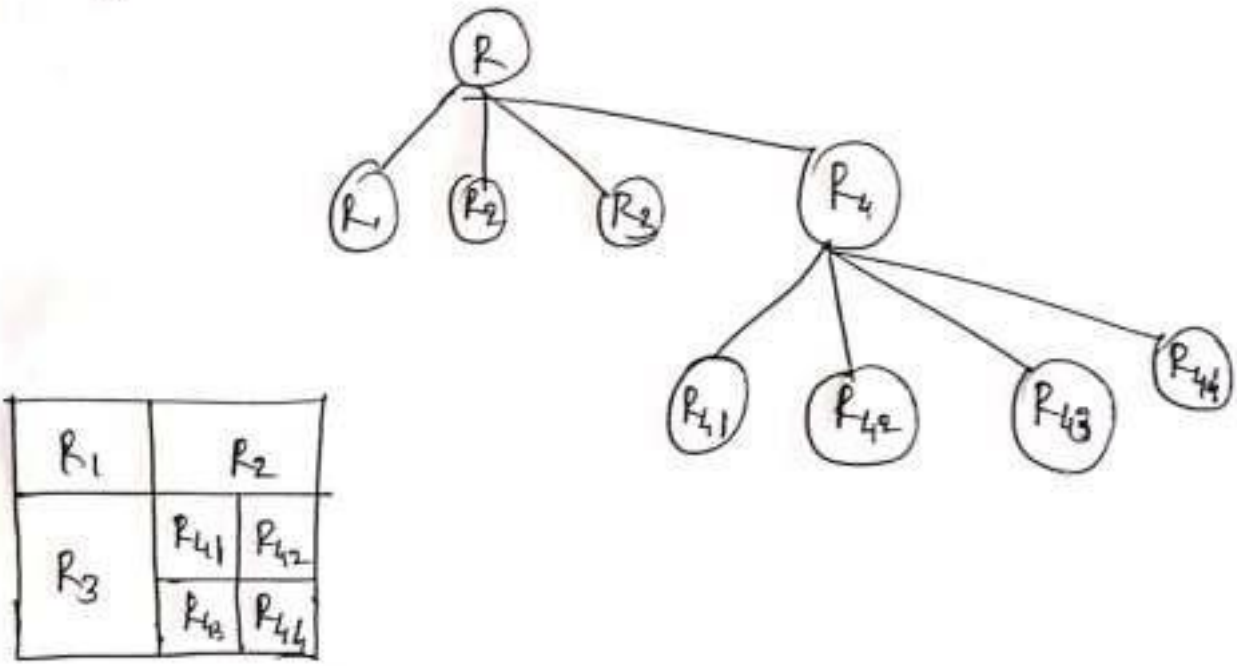
## 4.2] Region Splitting & Merging -

**Principle -**

1) Subdivide an image initially into a set of arbitrary disjoint regions.
2) Merge & /or split the regions to satisfy predefined conditions of segmentation.

**Steps -**

1) Split/subdivide image into four disjoint quadrant regions "$R_i$" for which $Q(R_i)$ = False




2) When no further splitting is possible, merge any adjacent regions "$R_j$" & "$R_k$"

if $Q(R_j \cup R_k) = \text{TRUE}$.

3) Stop when no further merging is possible.

## 5] Segmentation using Morphological WaterSheds –