GUDLAVALLERU ENINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

Seshadri Rao Knowledge Village, Gudlavalleru – 521 356.

Department of Computer Science and Engineering

HANDOUT

on

DIGITAL IMAGE PROCESSING

Vision

To be a Centre of Excellence in computer science and engineering education and training to meet the challenging needs of the industry and society.

Mission

- To impart quality education through well-designed curriculum in tune with the growing software needs of the industry.
- To serve our students by inculcating in them problem solving,leadership,teamwork skills and the value of commitment to quality,ethical behavior & respect for others.
- To foster industry-academia relationship for mutual benefit and growth.

Program Educational Objectives

- Identify, analyze, formulate and solve Computer Science and Engineering problems both independently and in a team environment by using the appropriate modern tools.

- Manage software projects with significant technical, legal, ethical, social, environmental and economic considerations

- Demonstrate commitment and progress in lifelong learning, professional development, leadership and Communicate effectively with professional clients and the public.

DIGITAL IMAGE PROCESSING

Course Objectives:

• To gain the knowledge in various image processing techniques.

Learning Outcomes:

Students will be able to

- understand the fundamentals of image processing.
- use appropriate image enhancement technique to improve the quality of an image.
- select an appropriate color model for an application.
- apply suitable image segmentation technique for an application.
- analyze various image compression techniques.
- apply morphological operations to modify the structure of an image.

Lecture Schedule:

Topic	No. of Periods	
	Theory	Tutorial
UNIT I : Introduction		
Digital image processing	**1**	
Fundamental steps in digital image processing, components of image processing system	**2**	
Examples of fields that use digital image processing	**2**	**1**
Image sensing and Acquisition	**1**	
Sampling and quantization	**2**	
Basic relationships between pixels	**1**	**1**
UNIT - II: Image enhancement in the spatial domain		
Introduction, Basic gray-level transformations,	**3**	
Histogram processing	**2**	
Enhancement using arithmetic and logic operators	**1**	**1**
Basics of spatial filtering	**1**	
smoothing and sharpening spatial filters	**4**	
Combining the spatial enhancement methods	**1**	**1**
UNIT - III: Color Image Processing		
Introduction, Color fundamentals	**2**	
Color models	**2**	

Pseudo color image processing	**2**	**1**
Basics of full color image processing	**1**	
Color transformations	**2**	
Color image smoothing and sharpening	**1**	
Color segmentation	**2**	**1**
UNIT - IV: Image Compression		
Fundamentals, image compression models	**2**	
Error-free compression	**2**	
Lossy predictive coding	**1**	**1**
UNIT - V: Morphological Image Processing		
Preliminaries, dilation, erosion	**1**	
Open and closing	**1**	
Hit or miss transformation	**1**	
Basic morphologic algorithms	**2**	**1**
UNIT - VI: Image Segmentation		
Detection of discontinuities	**2**	
Edge linking and boundary detection	**2**	
Thresholding	**2**	
Region–based segmentation	**1**	**1**

UNIT - I:

Introduction Digital image processing, Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system, Image sensing and Acquisition, sampling and quantization, basic relationships between pixels.

Digital image processing:

- An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the *intensity* or *gray level* of the image at that point.
- When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a *digital image*.
- The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value.
- These elements are referred to as picture elements, image elements, pels, and pixels.
- *Pixel* is the term most widely used to denote the elements of a digital image.
- Vision is the most advanced of our senses, so images play the single most important role in human perception.
- However, unlike humans, who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate on images generated by sources that humans are not accustomed to associating with images. These include ultra-sound, electron microscopy, and computer-generated images.

- Thus, digital image processing encompasses a wide and varied field of applications. There is no general agreement among authors regarding where image processing stops and other related areas, such as image analysis and computer vision, start.
- Sometimes a distinction is made by defining image processing as a discipline in which *both the input and output of a process are images*. We believe this to be a limiting and somewhat artificial boundary.
- For example, under this definition, even the trivial task of computing the average intensity of an image (which yields a single number) would not be considered an image processing operation. On the other hand, there are fields such as computer vision whose ultimate goal is to use computers to emulate human vision, including learning and being able to make inferences and take actions based on visual inputs. This area itself is a branch of artificial intelligence (AI) whose objective is to emulate human intelligence. The field of AI is in its earliest stages of infancy in terms of development, with progress having been much slower than originally anticipated.
- The area of image analysis (also called image understanding) is in between image processing and computer vision.
- There are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, and high-level processes.
- **Low level processes** involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images.

- **Mid-level processing** on images involves tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (e.g., edges, contours, and the identity of individual objects).
- **higher-level processing** involves "making sense" of an ensemble of recognized objects, as in image analysis, and, at the far end of the continuum, performing the cognitive functions normally associated with vision and, in addition, encompasses processes that extract attributes from images, up to and including the recognition of individual objects. As a simple illustration to clarify these concepts, consider the area of automated analysis of text. The processes of acquiring an image of the area containing the text, preprocessing that image, extracting (segmenting) the individual characters, describing the characters in a form suitable for computer processing, and recognizing those individual characters are in the scope of what we call digital image processing.

Fundamental Steps in Digital Image Processing:

- Image acquisition is the first process shown in Fig.2. 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.

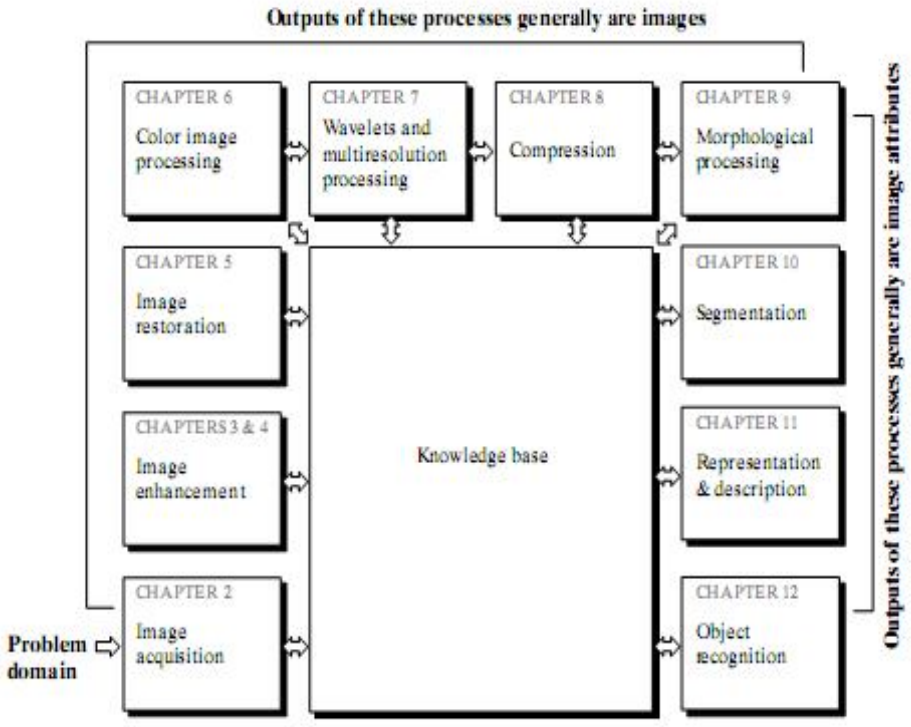


Fig.2. Fundamental steps in Digital Image Processing

- 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.

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.

- 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 frames) 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, sometimes 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.

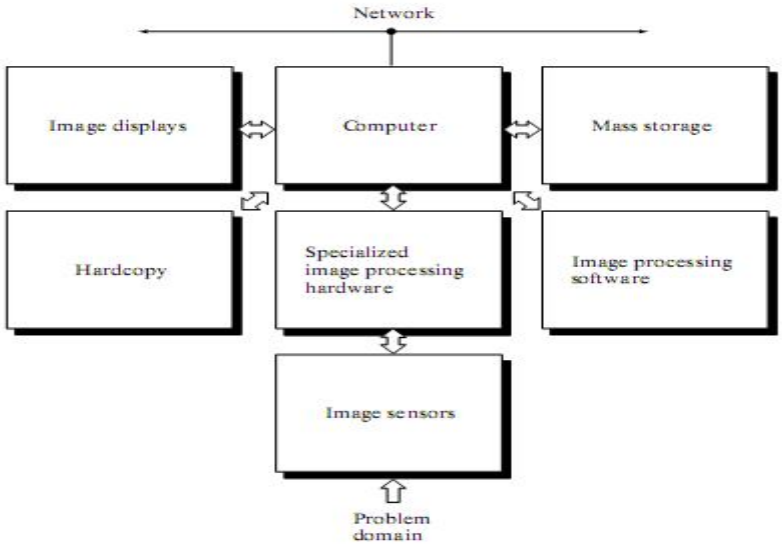


Fig.3. Components of a general purpose Image Processing System

- 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.

Process of image acquisition:

Image Sensing and Acquisition:

- The types of images in which we are interested are generated by the combination of an “illumination” source and the reflection or absorption of energy from that source by the elements of the “scene” being imaged. We enclose illumination and scene in quotes to emphasize the fact that they are considerably more general than the familiar situation in which a visible light source illuminates a common everyday 3-D (three-dimensional) scene. For example, the

illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern.

- Similarly, the scene elements could be familiar objects, but they can just as easily be molecules, buried rock formations, or a human brain. We could even image a source, such as acquiring images of the sun. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects. An example in the first category is light reflected from a planar surface. An example in the second category is when X-rays pass through a patient's body for the purpose of generating a diagnostic X-ray film. In some applications, the reflected or transmitted energy is focused onto a photo converter (e.g., a phosphor screen), which converts the energy into visible light. Electron microscopy and some applications of gamma imaging use this approach.

- Figure 5.1 shows the three principal sensor arrangements used to transform illumination energy into digital images. The idea is simple: 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.

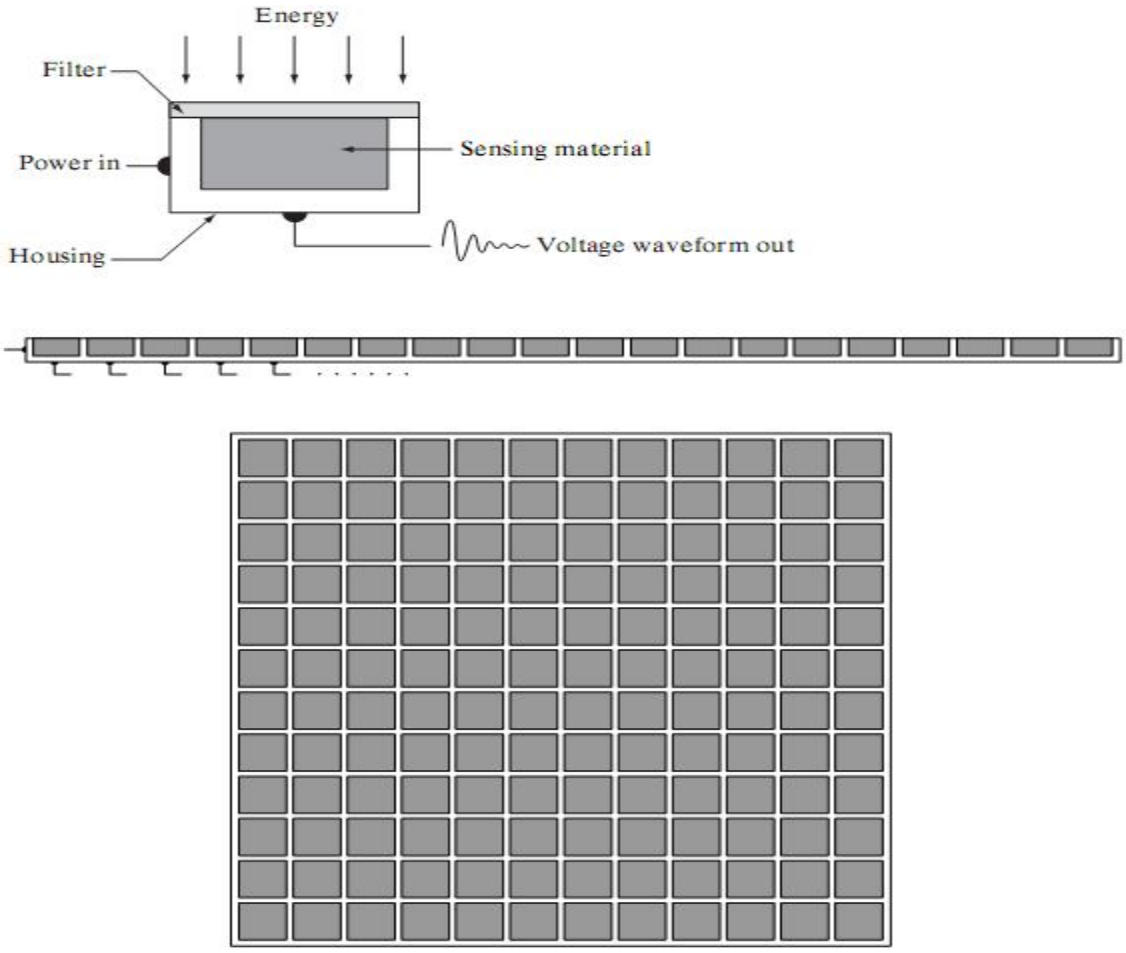


Fig.5.1 (a) Single imaging Sensor (b) Line sensor (c) Array sensor

(1)Image Acquisition Using a Single Sensor:

- Figure 5.1 (a) shows the components of a single sensor. Perhaps the most familiar sensor of this type is the photodiode, which is constructed of silicon materials and whose output voltage waveform is proportional to light. The use of a filter in front of a sensor improves selectivity. For example, a green (pass) filter in front of a light sensor favors light in the green band of the color spectrum. As a consequence, the sensor output will be stronger for green light than for other components in the visible spectrum.
- 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 5.2 shows an

arrangement used in high-precision scanning, where a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension. The single sensor is mounted on a lead screw that provides motion in the perpendicular direction. Since mechanical motion can be controlled with high precision, this method is an inexpensive (but slow) way to obtain high-resolution images. Other similar mechanical arrangements use a flat bed, with the sensor moving in two linear directions. These types of mechanical digitizers sometimes are referred to as microdensitometers.

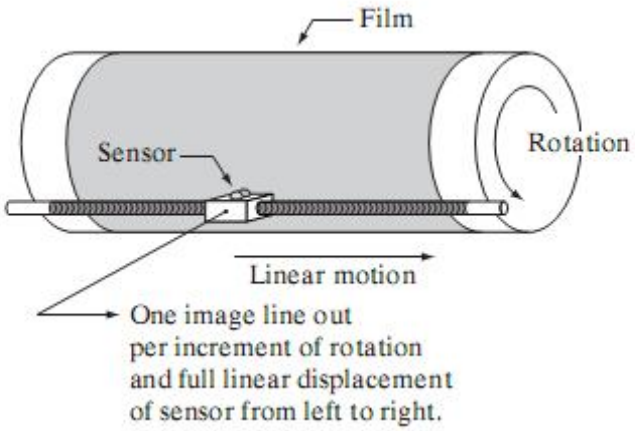


Fig.5.2. Combining a single sensor with motion to generate a 2-D image

(2) Image Acquisition Using Sensor Strips:

- A geometry that is used much more frequently than single sensors consists of an in-line arrangement of sensors in the form of a sensor strip, as Fig. 5.1 (b) shows. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction, as shown in Fig. 5.3 (a).This is the type of arrangement used in most flat bed scanners. Sensing devices with 4000 or more in-line sensors are possible. In-line sensors are used routinely in airborne imaging applications, in

which the imaging system is mounted on an aircraft that flies at a constant altitude and speed over the geographical area to be imaged. One-dimensional imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted perpendicular to the direction of flight. The imaging strip gives one line of an image at a time, and the motion of the strip completes the other dimension of a two-dimensional image. Lenses or other focusing schemes are used to project the area to be scanned onto the sensors.

- Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional ("slice") images of 3-D objects, as Fig. 5.3 (b) shows. A rotating X-ray source provides illumination and the portion of the sensors opposite the source collect the X-ray energy that pass through the object (the sensors obviously have to be sensitive to X-ray energy).This is the basis for medical and industrial computerized axial tomography (CAT). It is important to note that the output of the sensors must be processed by reconstruction algorithms whose objective is to transform the sensed data into meaningful cross-sectional images.

- In other words, images are not obtained directly from the sensors by motion alone; they require extensive processing. A 3-D digital volume consisting of stacked images is generated as the object is moved in a direction perpendicular to the sensor ring. Other modalities of imaging based on the CAT principle include magnetic resonance imaging (MRI) and positron emission tomography (PET).The illumination sources, sensors, and types of images are different, but conceptually they are very similar to the basic imaging approach shown in Fig. 5.3 (b).

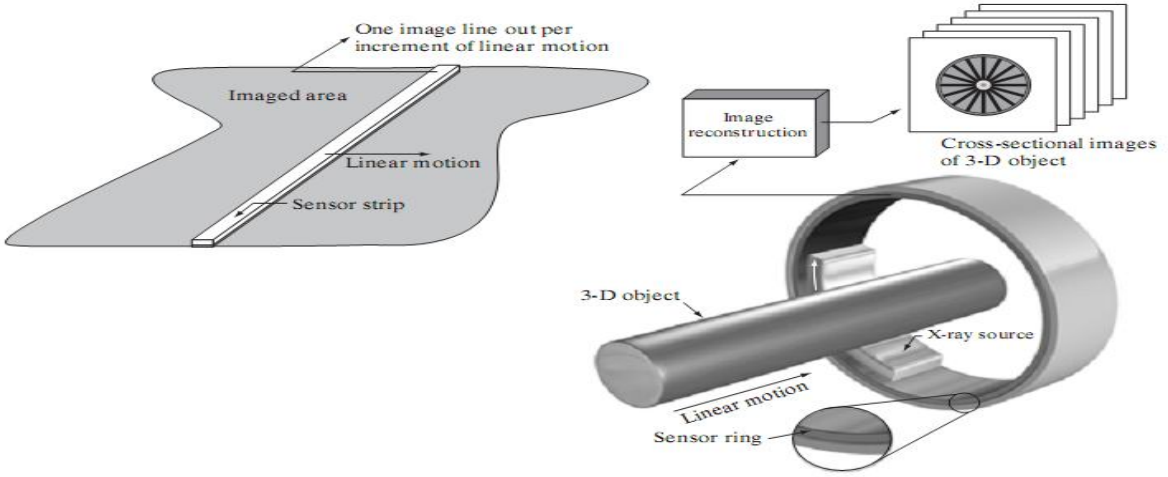


Fig.5.3 (a) Image acquisition using a linear sensor strip (b) Image acquisition using a circular sensor strip.

(3) Image Acquisition Using Sensor Arrays:

- Figure 5.1 (c) shows individual sensors arranged in the form of a 2-D array. Numerous electromagnetic and some ultrasonic sensing devices frequently are arranged in an array format. This is also the predominant arrangement found in digital cameras. A typical sensor for these cameras is a CCD array, which can be manufactured with a broad range of sensing properties and can be packaged in rugged arrays of 4000 * 4000 elements or more.
- CCD sensors are used widely in digital cameras and other light sensing instruments. The response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor, a property that is used in astronomical and other applications requiring low noise images. Noise reduction is achieved by letting the sensor integrate the input light signal over minutes or even hours. Since the sensor array shown in Fig. 5.4 (c) 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 Fig.5.4. This figure shows the energy from an illumination source being reflected from a scene element, but, as mentioned at the beginning of this section, the energy also could be transmitted through the scene elements. The first function performed by the imaging system shown in Fig.5.4 (c) is to collect the incoming energy and focus it onto an image plane.

- If the illumination is light, the front end of the imaging system is a lens, which projects the viewed scene onto the lens focal plane, as Fig. 2.15(d) shows. The sensor array, which is coincident with the focal plane, produces outputs proportional to the integral of the light received at each sensor. Digital and analog circuitry sweep these outputs and converts them to a video signal, which is then digitized by another section of the imaging system. The output is a digital image, as shown diagrammatically in Fig. 5.4 (e).

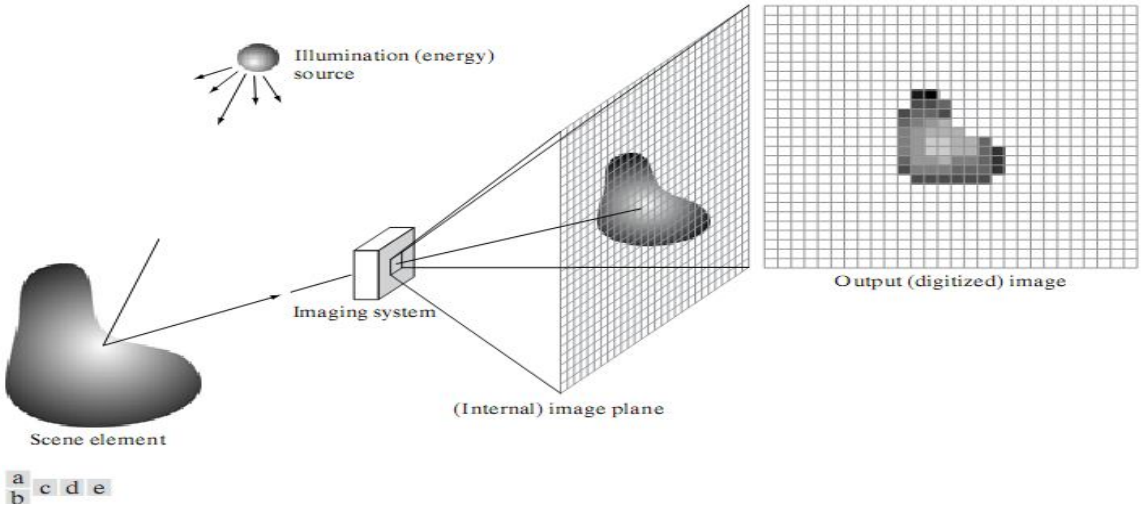


Fig.5.4 An example of the digital image acquisition process (a) Energy ("illumination") source (b) An element of a scene (c) Imaging system (d) Projection of the scene onto the image plane (e) Digitized image

Image Sampling and Quantization:

- The output of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed. 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:

- The basic idea behind sampling and quantization is illustrated in Fig.6.1. Figure 6.1(a) shows a continuous image, f(x, y), that we want to convert to digital form. An image may be continuous with respect to the x- and y-coordinates, and also in amplitude.
- To convert it 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*.
- The one-dimensional function shown in Fig.6.1 (b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in Fig. 6.1(a).The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in Fig.6.1 (c).The location of each sample is given by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives the sampled function. However, the values of the samples still span (vertically) a continuous range of gray-level values. In order to form a digital function, the gray-level values also must be converted (quantized) into discrete quantities. The right side of Fig. 6.1 (c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to

each of the eight gray levels. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in Fig.6.1 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.

- Sampling in the manner just described assumes that we have a continuous image in both coordinate directions as well as in amplitude. In practice, the method of sampling is determined by the sensor arrangement used to generate the image. When an image is generated by a single sensing element combined with mechanical motion, as in Fig. 2.13, the output of the sensor is quantized in the manner described above. However, sampling is accomplished by selecting the number of individual mechanical increments at which we activate the sensor to collect data. Mechanical motion can be made very exact so, in principle; there is almost no limit as to how fine we can sample an image. However, practical limits are established by imperfections in the optics used to focus on the sensor an illumination spot that is inconsistent with the fine resolution achievable with mechanical displacements. When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the sampling limitations in one image direction. Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors in the other. Quantization of the sensor outputs completes the process of generating a digital image.

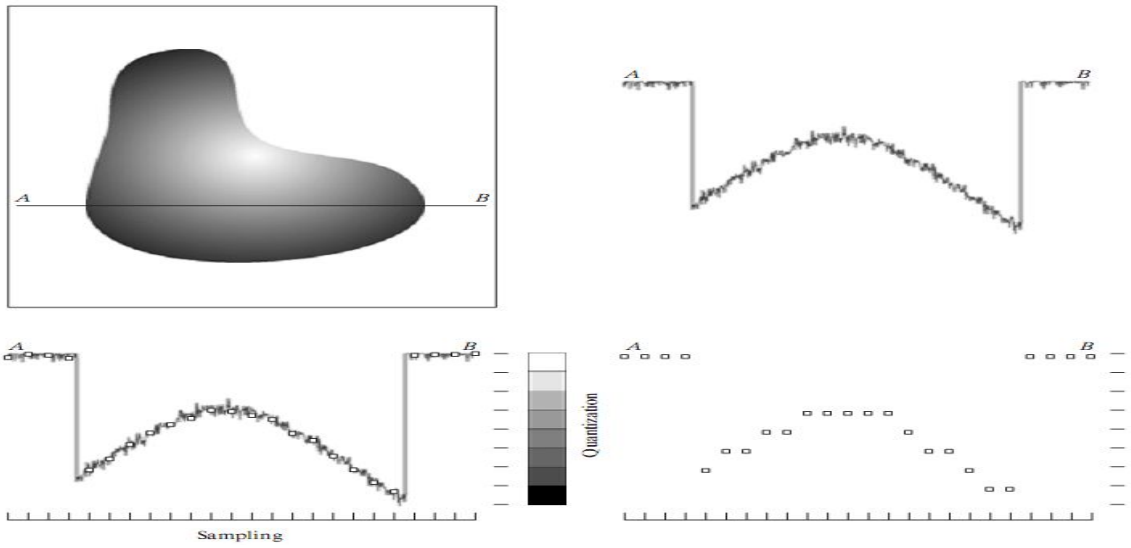


Fig.6.1. Generating a digital image (a) Continuous image (b) A scan line from A to Bin the continuous image, used to illustrate the concepts of sampling and quantization (c) Sampling and quantization. (d) Digital scan line

- 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. Figure 6.2 illustrates this concept. Figure 6.2 (a) shows a continuous image projected onto the plane of an array sensor. Figure 6.2 (b) shows the image after sampling and quantization. Clearly, the quality of a digital image is determined to a large degree by the number of samples and discrete gray levels used in sampling and quantization.

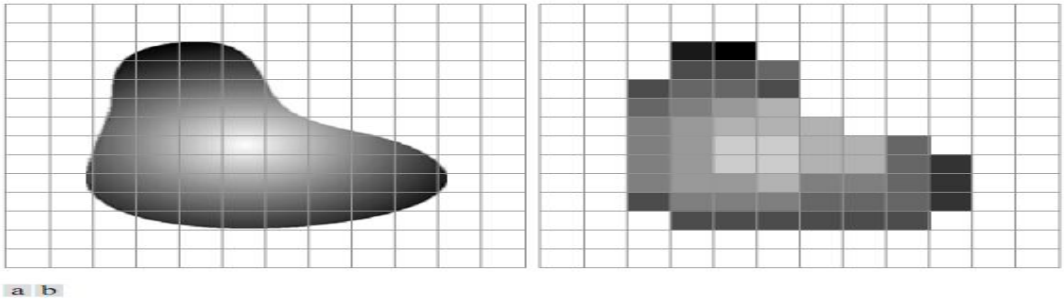


Fig.6.2. (a) Continuos image projected onto a sensor array (b) Result of image sampling and quantization.

Representing Digital Images:

- We will use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting digital image has M rows and N columns. The values of the coordinates (x, y) now become discrete quantities. For notational clarity and convenience, we shall use integer values for these discrete coordinates.
- Thus, the values of the coordinates at the origin are (x, y) = (0, 0). The next coordinate values along the first row of the image are represented as (x, y) = (0, 1). It is important to keep in mind that the notation (0, 1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Figure 1 shows the coordinate convention used.

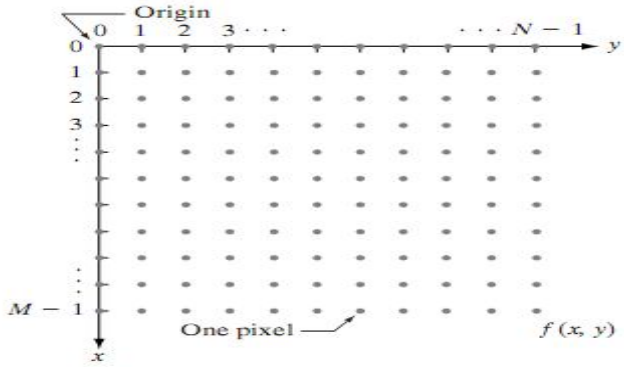


Fig 1 Coordinate convention used to represent digital images

- The notation introduced in the preceding paragraph allows us to write the complete M*N digital image in the following compact matrix form.

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

- The right side of this equation is by definition a digital image. Each element of this matrix array is called an image element, picture element, pixel, or pel.

Spatial and Gray-Level Resolution:

- Sampling is the principal factor determining the spatial resolution of an image. Basically, spatial resolution is the smallest discernible detail in an image. Suppose that we construct a chart with vertical lines of width W, with the space between the lines also having width W.A line pair consists of one such line and its adjacent space. Thus, the width of a line pair is 2W, and there are 1/2Wline pairs per unit distance. A widely used definition of resolution is simply the smallest number of discernible line pairs per unit distance; for example, 100 line pairs per millimeter. Gray-level resolution similarly refers to the smallest discernible change in gray level. We have considerable discretion regarding the number of samples used to generate a digital image, but this is not true for the number of gray levels. Due to hardware considerations, the number of gray levels is usually an integer power of 2.
- The most common number is 8 bits, with 16 bits being used in some applications where enhancement of specific gray-level ranges is necessary. Sometimes we find systems that can digitize the gray levels of an image with 10 or 12 bit of accuracy, but these are the exception rather than the rule. When an actual measure of physical resolution relating pixels and the level of detail they resolve in the original scene are not necessary, it is not uncommon to refer to an

Llevel digital image of size M*N as having a spatial resolution of M*N pixels and a gray-level resolution of L levels.

Fig.7.1. A 1024*1024, 8-bit image subsampled down to size 32*32 pixels The number of allowable gray levels was kept at 256

- The sub sampling was accomplished by deleting the appropriate number of rows and columns from the original image. For example, the 512*512 image was obtained by deleting every other row and column from the 1024*1024 image. The 256*256 image was generated by deleting every other row and column in the 512*512 image, and so on. The number of allowed gray levels was kept at 256. These images show the dimensional proportions between various sampling densities, but their size differences make it difficult to see the effects resulting from a reduction in the number of samples. The simplest way to compare these effects is to bring all the sub sampled images up to size 1024*1024 by row and column pixel replication. The results are shown in Figs. 7.2 (b) through (f). Figure7.2 (a) is the same 1024*1024, 256-level image shown in Fig.7.1; it is repeated to facilitate comparisons.

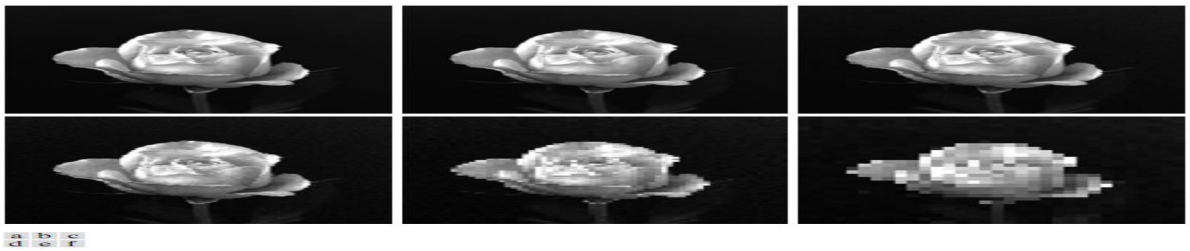


Fig. 7.2 (a) 1024*1024, 8-bit image (b) 512*512 image resampled into 1024*1024 pixels by
row and column duplication (c) through (f) 256*256, 128*128, 64*64, and 32*32 images resampled into 1024*1024 pixels

- Compare Fig. 7.2(a) with the 512*512 image in Fig. 7.2(b) and note that it is virtually impossible to tell these two images apart. The level of detail lost is simply too fine to be seen on the printed page at the scale in which these images are shown. Next, the 256*256 image in Fig. 7.2(c) shows a very slight fine checkerboard pattern in the borders between flower petals and the black background. A slightly more pronounced graininess throughout the image also is beginning to appear. These effects are much more visible in the 128*128 image in Fig. 7.2(d), and they become pronounced in the 64*64 and 32*32 images in Figs. 7.2 (e) and (f), respectively.
- In the next example, we keep the number of samples constant and reduce the number of gray levels from 256 to 2, in integer powers of 2.Figure 7.3(a) is a 452*374 CAT projection image, displayed with k=8 (256 gray levels). Images such as this are obtained by fixing the X-ray source in one position, thus producing a 2-D image in any desired direction. Projection images are used as guides to set up the parameters for a CAT scanner, including tilt, number of slices, and range. Figures 7.3(b) through (h) were obtained by reducing the number of bits from k=7 to k=1 while keeping the spatial resolution constant at 452*374 pixels. The 256-, 128-, and 64-level images are visually identical for all practical purposes. The 32-level image shown in Fig. 7.3 (d), however, has an almost imperceptible set of very fine ridge like structures in areas of smooth gray levels (particularly in the skull).This effect, caused by the use of an insufficient number of gray levels in smooth areas of a digital image, is called false contouring, so called because the ridges resemble topographic contours in a map. False contouring generally is quite visible in images displayed using 16 or less uniformly spaced gray levels, as the images in Figs. 7.3(e) through (h) show.

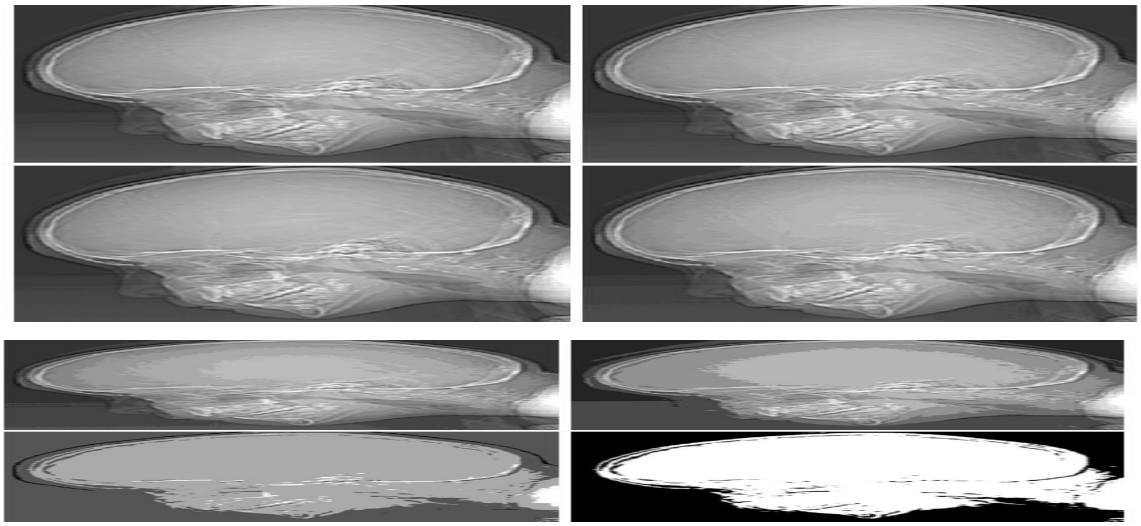


Fig. 7.3 (a) 452*374, 256-level image (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant (e)–(g) Image displayed in 16, 8, 4, and 2 gray levels.

- As a very rough rule of thumb, and assuming powers of 2 for convenience, images of size 256*256 pixels and 64 gray levels are about the smallest images that can be expected to be reasonably free of objectionable sampling checker-boards and false contouring.
- The results in Examples 7.2 and 7.3 illustrate the effects produced on image quality by varying N and k independently. However, these results only partially answer the question of how varying N and k affect images because we have not considered yet any relationships that might exist between these two parameters.

Aliasing and Moiré Patterns:

- Functions whose area under the curve is finite can be represented in terms of sines and cosines of various frequencies. The sine/cosine component with the highest frequency determines the highest "frequency content" of the function. Suppose that this highest frequency is finite and that the function is of unlimited duration (these functions are called band-limited functions).Then, the Shannon sampling theorem [Brace well (1995)] tells us that, 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. If the function is undersampled, then a phenomenon called aliasing corrupts the sampled image. The corruption is in the form of additional frequency components being introduced into the sampled function. These are called aliased frequencies. Note that the sampling rate in images is the number of samples taken (in both spatial directions) per unit distance.

- As it turns out, except for a special case discussed in the following paragraph, it is impossible to satisfy the sampling theorem in practice. We can only work with sampled data that are finite in duration. We can model the process of converting a function of unlimited duration into a function of finite duration simply by multiplying the unlimited function by a "gating function" that is valued 1 for some interval and 0 elsewhere. Unfortunately, this function itself has frequency components that extend to infinity. Thus, the very act of limiting the duration of a band-limited function causes it to cease being band limited, which causes it to violate the key condition of the sampling theorem. The principal approach for reducing the aliasing effects on an image is to reduce its high-frequency components by blurring the image prior to sampling. 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 so called Moiré patterns.
- There is one special case of significant importance in which a function of infinite duration can be sampled over a finite interval without violating the sampling theorem. When a function is periodic, it may be sampled at a rate equal to or exceeding twice its highest frequency and it is possible to recover the function from its samples provided that the sampling captures exactly an integer number of periods of the function. This special case allows us to

illustrate vividly the Moiré effect. Figure 8 shows two identical periodic patterns of equally spaced vertical bars, rotated in opposite directions and then superimposed on each other by multiplying the two images. A Moiré pattern, caused by a breakup of the periodicity, is seen in Fig.8 as a 2-D sinusoidal (aliased) waveform (which looks like a corrugated tin roof) running in a vertical direction. A similar pattern can appear when images are digitized (e.g., scanned) from a printed page, which consists of periodic ink dots.

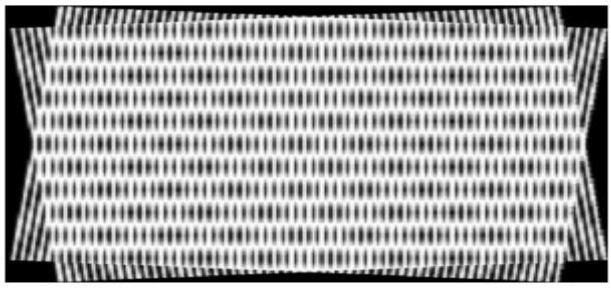


Fig.8. Illustration of the Moiré pattern effect

Basic relationships and distance measures between pixels in a digital image:

Neighbors of a Pixel:

- A pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by (x+1, y), (x-1, y), (x, y+1), (x, y-1). This set of pixels, called the 4-neighbors of p, is denoted by N4 (p). Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.
- The four diagonal neighbors of p have coordinates (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) and are denoted by ND (p). These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by N8 (p). As before, some of the points in ND (p) and N8 (p) fall outside the image if (x, y) is on the border of the image.

Connectivity:

- Connectivity between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as regions and boundaries. To establish if two pixels are connected, it must be determined if they are neighbors and if their gray levels satisfy a specified criterion of similarity (say, if their gray levels are equal). For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors, but they are said to be connected only if they have the same value.
- Let V be the set of gray-level values used to define adjacency. In a binary image, V={1} if we are referring to adjacency of pixels with value 1. In a grayscale image, the idea is the same, but set V typically contains more elements. For example, in the adjacency of pixels with a range of possible gray-level values 0 to 255, set V could be any subset of these 256 values. We consider three types of adjacency
  a. 4-adjacency. Two pixels p and q with values from V are 4-adjacent if q is in the set N4 (p).
  b. 8-adjacency. Two pixels p and q with values from V are 8-adjacent if q is in the set N8 (p).
  c. m-adjacency (mixed adjacency).Two pixels p and q with values from V are m-adjacent if
     i. q is in N4 (p), or
     ii. q is in ND (p) and the set has no pixels whose values are from V.
- Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used. For example, consider the pixel arrangement shown in Fig.9 (a) for V= {1}.The three pixels at the top of Fig.9 (b) show multiple (ambiguous) 8- adjacency, as indicated by the dashed lines. This

ambiguity is removed by using m-adjacency, as shown in Fig. 9 (c).Two image subsets S1 and S2 are adjacent if some pixel in S1 is adjacent to some pixel in S2. It is understood here and in the following definitions that adjacent means 4-, 8- , or m-adjacent. 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

$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

- Where $(x_0, y_0) = (x, y), (x_n, y_n) = (s, t)$,and Pixels$(x_i, y_i)$ and (x_{i-1}, y_{i-1}) are adjacent for 1<=i<=n In this case, n is the length of the path. If (xo, yo) = (xn, yn), the path is a closed path. We can define 4-, 8-, or m-paths depending on the type of adjacency specified. For example, the paths shown in Fig. 9 (b) between the northeast and southeast points are 8-paths, and the path in Fig. 9 (c) is an m-path. Note the absence of ambiguity in the m-path. 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 entirely of pixels in S. For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S. If it only has one connected component, then set S is called a connected set.

Fig.9 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency

- Let R be a subset of pixels in an image. We call R a region of the image if R is a connected set. 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. If R happens to be an entire image (which we recall is a rectangular set of pixels), then its boundary is defined as the set of pixels in the first and last rows and columns of the image. This extra definition is required because an image has no neighbors beyond its border. Normally, when we refer to a region, we are referring to a subset of an image, and any pixels in the boundary of the region that happen to coincide with the border of the image are included implicitly as part of the region boundary.

Distance Measures:

- For pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a distance function or metric if

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

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

- For this distance measure, the pixels having a distance less than or equal to some value r from(x, y) are the points contained in a disk of radius r centered at (x, y).
- The **D4** distance (**also called city-block distance**) between p and q is defined as

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

- In this case, the pixels having a D4 distance from (x, y) less than or equal to some value r form a diamond centered at (x, y). For example, the pixels with D4 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

- The pixels with D4 =1 are the 4-neighbors of (x, y).
- The **D8** distance (**also called chessboard distance**) between p and q is defined as

  $D_8(p, q) = \max(|x - s|, |y - t|).$
- In this case, the pixels with D8 distance from(x, y) less than or equal to some value r form a square centered at (x, y). For example, the pixels with D8 distance ≤ 2 from(x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

- The pixels with D8=1 are the 8-neighbors of (x, y). Note that the D4 and D8 distances between p and q are independent of any paths that might exist between the points because these distances involve only the coordinates of the points. If we elect to consider m-adjacency, however, the Dm distance between two points is defined as the shortest m-path between the points. In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors. For instance, consider the following arrangement of pixels and assume that p, p2 , and p4 have value 1 and that p1 and p3 can have a value of 0 or 1:

$$\begin{array}{cc} p_3 & p_4 \\ p_1 & p_2 \\ p & \end{array}$$

- Suppose that we consider adjacency of pixels valued 1 (i.e. = {1}). If p1 and p3 are 0, the length of the shortest m-path (the Dm

distance) between p and p4 is 2. If p1 is 1, then p2 and p will no longer be m-adjacent (see the definition of m-adjacency) and the length of the shortest m-path becomes 3 (the path goes through the points pp1p2p4). Similar comments apply if p3 is 1 (and p1is 0); in this case, the length of the shortest m-path also is 3. Finally, if both p1 and p3 are 1 the length of the shortest m-path between p and p4 is 4. In this case, the path goes through the sequence of points pp1p2p3p4.

Examples of Fields that Use Digital Image Processing:

- 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 (e.g., visual, X-ray, and so on).
- The principal energy source for images in use today is the electromagnetic energy spectrum.
- Other important sources of energy include acoustic, ultrasonic, and electronic (in the form of electron beams used in electron microscopy).
- Synthetic images, used for modeling and visualization, are generated by computer.
- Electromagnetic waves can be conceptualized as propagating sinusoidal waves of varying wavelengths, or they can be thought of as a stream of mass less particles, each traveling in a wavelike pattern and moving at the speed of light. Each mass less particle contains a certain amount (or bundle) of energy. Each bundle of energy is called a *photon*.
- If spectral bands are grouped according to energy per photon, we obtain the spectrum shown in Fig. 1.5, ranging from gamma rays (highest energy) at one end to radio waves (lowest energy) at the other.

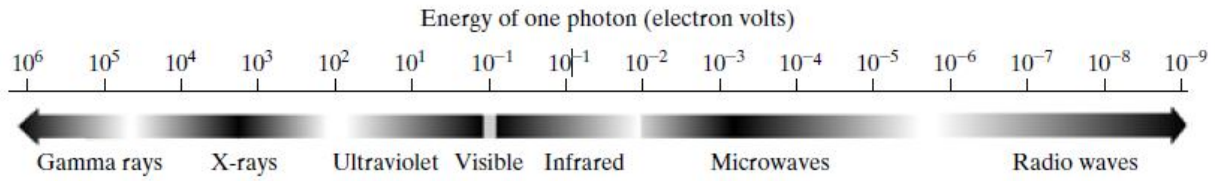
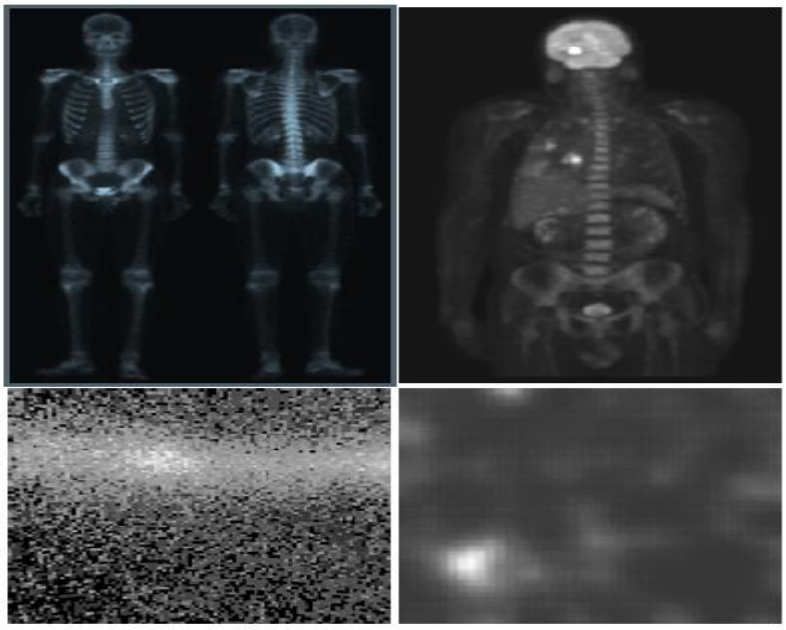


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

Gamma-Ray Imaging:

- Major uses of imaging based on gamma rays include nuclear medicine and astronomical observations.
- In nuclear medicine, the approach is to inject a patient with a radioactive isotope that emits gamma rays as it decays. Images are produced from the emissions collected by gamma ray detectors. Figure 1.6(a) shows an image of a complete bone scan obtained by using gamma-ray imaging. Images of this sort are used to locate sites of bone pathology, such as infections or tumors.
- Figure 1.6(b) shows another major modality of nuclear imaging called positron emission tomography (PET).The principle is the same as with X-ray tomography. However, instead of using an external source of X-ray energy, the patient is given a radioactive isotope that emits positrons as it decays. When a positron meets an electron, both are annihilated and two gamma rays are given off. These are detected and a tomographic image is created using the basic principles of tomography. The image shown in Fig. 1.6(b) is

one sample of a sequence that constitutes a 3-D rendition of the patient. This image shows a tumor in the brain and one in the lung, easily visible as small white masses.



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.)

- A star in the constellation of Cygnus exploded about 15,000 years ago, generating a superheated stationary gas cloud (known as the Cygnus Loop) that glows in a spectacular array of colors. Figure 1.6(c) shows the Cygnus Loop imaged in the gamma-ray band. Unlike the two examples shown in Figs. 1.6(a) and (b), this image was obtained using the natural radiation of the object being imaged.
- Fig. 1.6(d) shows an image of gamma radiation from a valve in a nuclear reactor. An area of strong radiation is seen in the lower, left side of the image.

X-ray Imaging:

- X-rays are among the oldest sources of EM radiation used for imaging. The best known use of X-rays is medical diagnostics, but they also are used extensively in industry and other areas, like astronomy.

- X-rays for medical and industrial imaging are generated using an X-ray tube, which is a vacuum tube with a cathode and anode. The cathode is heated, causing free electrons to be released. These electrons flow at high speed to the positively charged anode. When the electrons strike a nucleus, energy is released in the form of X-ray radiation.

- The energy (penetrating power) of the X-rays is controlled by a voltage applied across the anode, and the number of X-rays is controlled by a current applied to the filament in the cathode.
- Figure 1.7(a) shows a familiar chest X-ray generated simply by placing the patient between an X-ray source and a film sensitive to X-ray energy. The intensity of the X-rays is modified by absorption as they pass through the patient, and the resulting energy falling on the film develops it, much in the same way that light develops photographic film.

- In digital radiography, digital images are obtained by one of two methods: (1) by digitizing X-ray films; or (2) by having the X-rays that pass through the patient fall directly onto devices (such as a phosphor screen) that convert X-rays to light. The light signal in turn is captured by a light-sensitive digitizing system.

- Angiography is another major application in an area called contrast enhancement radiography. This procedure is used to obtain images (called angiograms) of blood vessels.

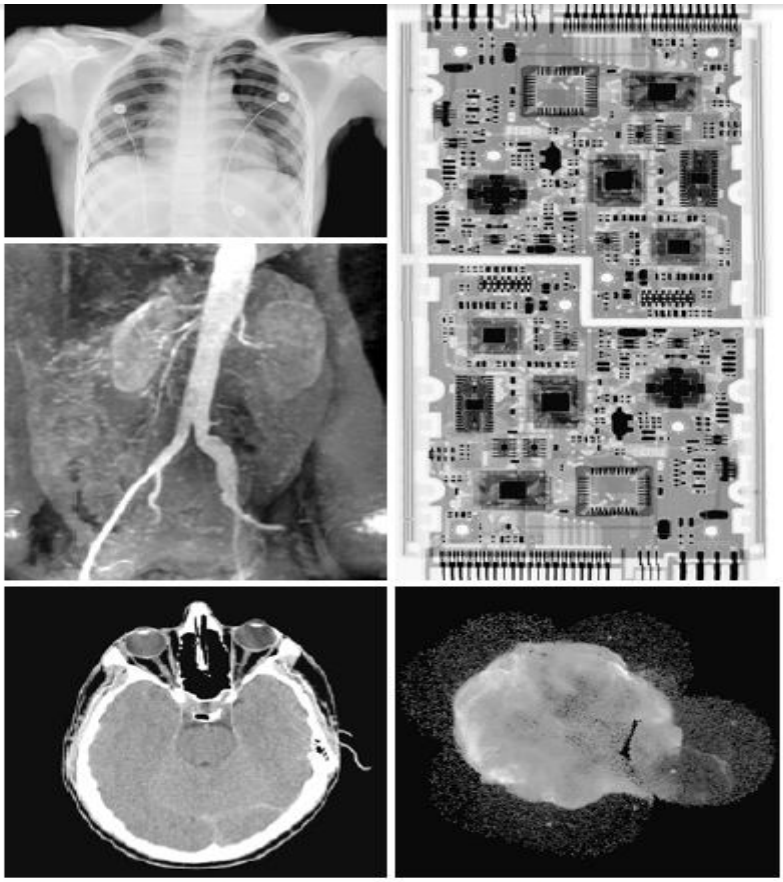
- A catheter (a small, flexible, hollow tube) is inserted, for example, into an artery or vein in the groin. The catheter is threaded into the blood vessel and guided to the area to be studied. When the catheter reaches the site under investigation, an X-ray contrast medium is injected through the catheter. This enhances contrast of the blood vessels and enables the radiologist to see any irregularities or blockages.

- Figure 1.7(b) shows an example of an aortic angiogram. The catheter can be seen being inserted into the large blood vessel on the lower left of the picture. Note the high contrast of the large vessel as the contrast medium flows up in the direction of the kidneys, which are also visible in the image.

- CAT (computerized axial tomography) image is a “slice” taken perpendicularly through the patient. Numerous slices are generated as the patient is moved in a longitudinal direction. The ensemble of such images constitutes a 3-D rendition of the inside of the patient, with the longitudinal resolution being proportional to the number of slice images taken. Figure 1.7(c) shows a typical head CAT slice image.

- Techniques similar to the ones just discussed, but generally involving higher energy X-rays, are applicable in industrial processes. Figure 1.7(d) shows an X-ray image of an electronic circuit board. Such images, representative of literally hundreds of industrial applications of X-rays, are used to examine circuit

boards for flaws in manufacturing, such as missing components or broken traces. Industrial CAT scans are useful when the parts can be penetrated by X-rays, such as in plastic assemblies, and even large bodies, like solid-propellant rocket motors. Figure 1.7(e) shows an example of X-ray imaging in astronomy. This image is the Cygnus Loop of Fig. 1.6(c), but imaged this time in the X-ray band.



a b c d e

FIGURE 1.7 Examples of X-ray imaging. (a) Chest X-ray. (b) Aortic angiogram. (c) Head CT. (d) Circuit boards. (e) Cygnus Loop. (Images courtesy of (a) and (c) Dr. David R. Pickens, Dept. of Radiology & Radiological Sciences, Vanderbilt University Medical Center, (b) Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School, (d) Mr. Joseph E. Pascente, Lixi, Inc., and (e) NASA.)

Imaging in the Ultraviolet Band:

- Applications of ultraviolet "light" are varied. They include lithography, industrial inspection, microscopy, lasers, biological imaging, and astronomical observations.
- Ultraviolet light is used in fluorescence microscopy. Fluorescence is a phenomenon discovered in the middle of the nineteenth century, when it was first observed that the mineral fluorspar fluoresces when ultraviolet light is directed upon it.

- The ultraviolet light itself is not visible, but when a photon of ultraviolet radiation collides with an electron in an atom of a fluorescent material, it elevates the electron to a higher energy level. Subsequently, the excited electron relaxes to a lower level and emits light in the form of a lower-energy photon in the visible (red) light region. The basic task of the fluorescence microscope is to use an excitation light to irradiate a prepared specimen and then to separate the much weaker radiating fluorescent light from the brighter excitation light. Thus, only the emission light reaches the eye or other detector. The resulting fluorescing areas shine against a dark background with sufficient contrast to permit detection.
- The darker the background of the non fluorescing material, the more efficient the instrument.

Imaging in the Visible and Infrared Bands:

TABLE 1.1 Thematic bands in NASA's LANDSAT satellite.

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

- Major area of visual processing is remote sensing, which usually 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 on the planet.

- In order to develop a basic appreciation for the power of this type of multispectral imaging, consider Fig. 1.10, which shows one image for each of the spectral bands in Table 1.1.The area imaged is Washington D.C., which includes features such as buildings, roads, vegetation, and a major river (the Potomac) going though the city. Images of population centers are used routinely (over time) to assess population growth and shift patterns, pollution, and other factors harmful to the environment. The differences between visual and infrared image features are quite noticeable in these images. Observe, for example, how well defined the river is from its surroundings in Bands 4 and 5.

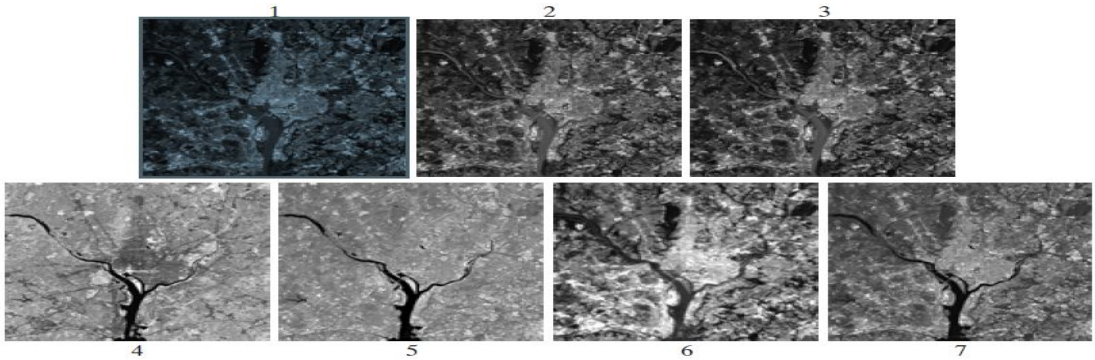


FIGURE 1.10 LANDSAT satellite images of the Washington, D.C. area. The numbers refer to the thematic bands in Table 1.1. (Images courtesy of NASA.)

Imaging in the Microwave Band:

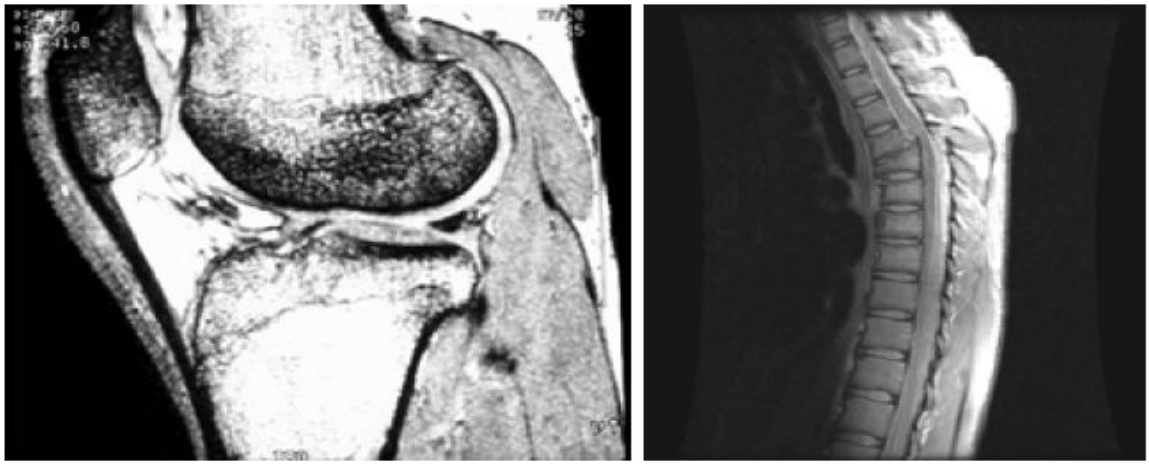
- The dominant application of imaging in the microwave band is radar. The unique feature of imaging radar is its ability to collect data over virtually any region at any time, regardless of weather or ambient lighting conditions.
- Some radar waves can penetrate clouds, and under certain conditions can also see through vegetation, ice, and extremely dry sand. In many cases, radar is the only way to explore inaccessible regions of the Earth's surface. Imaging radar works like a flash camera in that it provides its own illumination (microwave pulses) to illuminate an area on the ground and take a snapshot image.

Instead of a camera lens, radar uses an antenna and digital computer processing to record its images.

- In a radar image, one can see only the microwave energy that was reflected back toward the radar antenna.

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). This technique places a patient in a powerful magnet and passes radio waves through his or her body in short pulses. Each pulse causes a responding pulse of radio waves to be emitted by the patient's tissues. The location from which these signals originate and their strength are determined by a computer, which produces a two-dimensional picture of a section of the patient. MRI can produce pictures in any plane. Figure 1.17 shows MRI images of a human knee and spine.



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.)

UNIT-I

Assignment-Cum-Tutorial Questions

SECTION-A

I. Objective Questions

1. _____ process is characterized by the fact that both its inputs and outputs are images.
2. _____ process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images
3. Digitizing the _____ values is called *sampling*.
4. Digitizing the _____ values is called *quantization*
5. D8 distance also called _____
6. D4 distance also called _____
7. Major uses of imaging based on _____ include nuclear medicine and astronomical observations
8. _____ light is used in fluorescence microscopy
9. The major applications of imaging in the _____ band are in medicine and astronomy.
10. A pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by _____called the 4-neighbors of p [ ]

A. (x+1, y) B. (x-1, y) C. (x, y+1) D. (x, y-1) E. ALL

11. A pixel p at coordinates (x, y) has called the 8-neighbors of p if it has_____ [ ]

A. Horizontal and vertical neighbors(x+1, y), (x-1, y), (x, y+1), (x, y-1)

B. Diagonal neighbors(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)

C. Both A and B D. None of the above

12. In a binary image with values 0 and 1, two pixels may be 4-neighbors, but they are said to be connected only if they have the ____value. [ ]

A. Same B. Different C. Both A and B D. none

13. 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._____

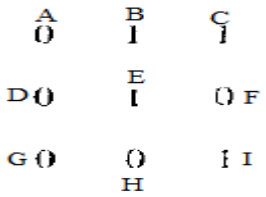
14. Intensity levels in 8-bit image are [ ]

A. 128 B. 255 C. 256 D. 512

15. In bit plane slicing 8 bit image will have_____ number of planes. [ ]

A. 6 B. 7 C. 8 D. 9

16. The number of intensity levels in RGB image are _____[ ]



17. Consider the above pixel format for V={1}, pixel E is connected to pixel D **(True/False).**

18. Consider the above pixel format for V={1,0} Pixel H is connected to _____ pixels.

19. Consider the above pixel format for V={1} find out which pixels fall under a path.

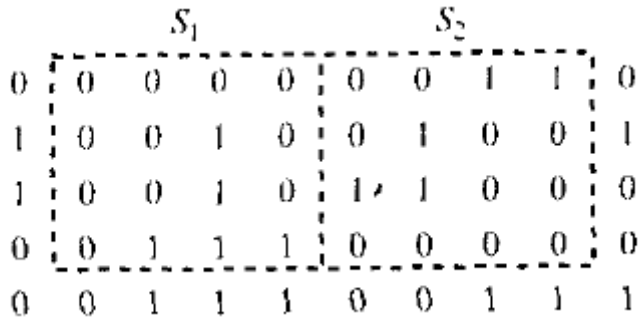
20. Consider the above pixel format for V={0}. There exists a closed path. **(True/False).**

SECTION-B

II. Descriptive Questions

1. What is meant by Digital Image Processing? Explain how digital images can be represented?
2. Explain of Fields that Use Digital Image Processing.
3. What are the fundamental steps in Digital Image Processing?
4. What are the components of an Image Processing System?
5. Explain the process of image acquisition.
6. Explain about image sampling and quantization process.

7. Define spatial and gray level resolution. Explain about isopreference curves.
8. Explain about Aliasing and Moire patterns.
9. Explain about the basic relationships and distance measures between pixels in a digital image.
10. Calculate the 4 neighbors of a pixel at coordinates p(2,3).
11. Calculate the resolution of an 1024X1024 image.
12. Find out 8 neighbors of a pixel at coordinates p(8,8).
13. Calculate the number of bits required to store an 128X128 image with 64 gray levels.
14. Consider the two image subsets, S1 and S2, shown in the following figure. For V={1}, determine whether these two subsets are (a) 4-connect (b)8-connect or (c) m-adjacent.



15. Consider the image segment shown below, 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

16. Compute for V = {1,2} with the same data in problem 6.
17. Calculate the (Euclidean) distance between points (2, -1) and (-2, 2).

UNIT - II

UNIT - II: Image enhancement in the spatial domain

Introduction, Basic gray-level transformations, histogram processing, enhancement using arithmetic and logic operators, Basics of spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

Introduction:

Spatial Domain and Point Processing:

- The term spatial domain refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will 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 processed image, and T is an operator on f, defined over some neighborhood of (x, y). In addition, T can operate on a set of input images, such as performing the pixel-by-pixel sum of K images for noise reduction.
- The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular sub image area centered at (x, y), as Fig.2.1 shows. The center of the sub image is moved from pixel to pixel starting, say, at the top left corner. The operator T is applied at each location (x, y) to yield the output, g, at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood.

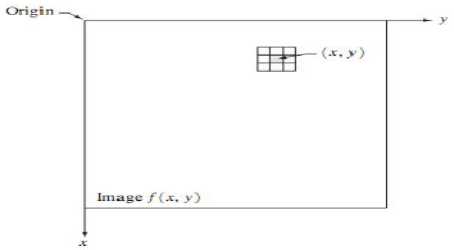


Fig.2.1 A 3*3 neighborhood about a point (x, y) in an image.

- The simplest form of T is when the neighborhood is of size 1*1 (that is, a single pixel). In this case, g depends only on the value of f at (x, y), and T becomes a gray-level (also called an intensity or mapping) transformation function of the form **s=T(r)** where, for simplicity in notation, r and s are variables denoting, , the gray level of f(x, y) and g(x, y) respectively at any point (x, y).
- For example, if T(r) has the form shown in Fig. 2.2(a), the effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below m and brightening the levels above m in the original image. In this technique, known as contrast stretching, the values of r below m are compressed by the transformation function into a narrow range of s, toward black. The opposite effect takes place for values of

r above m. In the limiting case shown in Fig. 2.2(b), T(r) produces a two-level (binary) image. A mapping of this form is called a **thresholding function**.

- Some fairly simple, yet powerful, processing approaches can be formulated with gray-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as **point processing**.

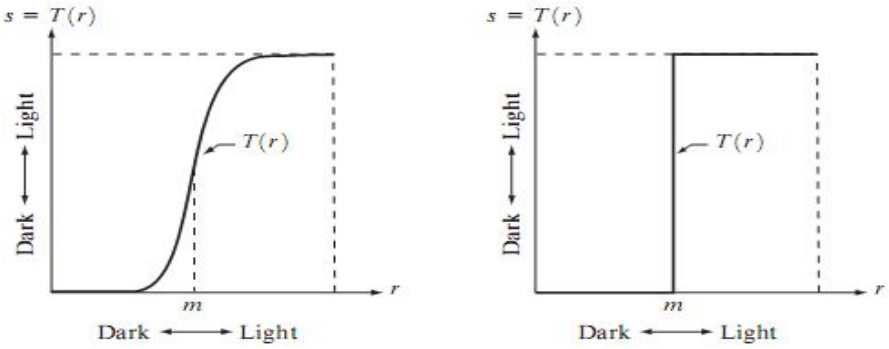


Fig.2.2 Graylevel transformation functions for contrast enhancement.

- Larger neighborhoods allow considerably more flexibility. The general approach is to use a function of the values of f in a predefined neighborhood of (x, y) to determine the value of g at (x, y).One of the principal approaches in this formulation is based on the use of so-called masks (also referred to as filters, kernels, templates, or windows). Basically, a mask is a small (say, 3*3) 2-D array, such as the one shown in Fig. 2.1, in which the values of the mask coefficients determine the nature of the process, such as image sharpening.

Image enhancement by point processing:

Basic Gray Level Transformations:

- The image enhancement technique is done by gray-level transformation functions. The values of pixels, before and after processing, will be denoted by **r** and **s**, respectively. These values are related by an expression of the form **s=T(r)**, where **T** is a transformation that maps a pixel value **r** into a pixel value **s**.
- Since we are dealing with digital quantities, values of the transformation function typically are stored in a one-dimensional array and the mappings from r to s are implemented via table lookups. For an 8-bit environment, a lookup table containing the values of T will have 256 entries.
- As an introduction to gray-level transformations, consider Fig. 1.1, which shows **three** basic types of functions used frequently for image enhancement:
    1. linear (negative and identity transformations),
    2. logarithmic (log and inverse-log transformations), and
    3. power-law (nth power and nth root transformations).

- The identity function is the trivial case in which output intensities are identical to input intensities.

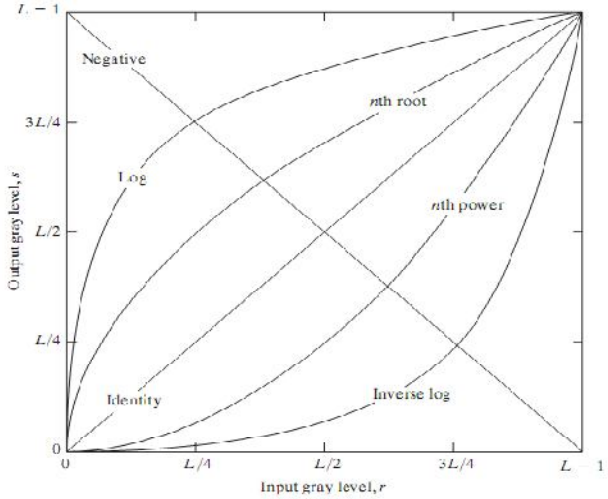


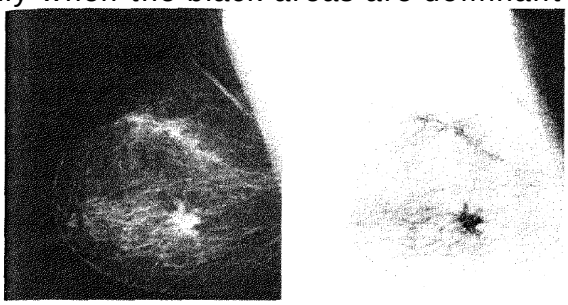
Fig.1.1 Some basic gray-level transformation functions used for image enhancement

Image Negatives:

- The negative of an image with gray levels in the range [0, L-1] is obtained by using the negative transformation shown in Fig.1.1, which is given by the expression

$$s = L - 1 - r.$$

- Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is particularly suited for enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.



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

Log Transformations:

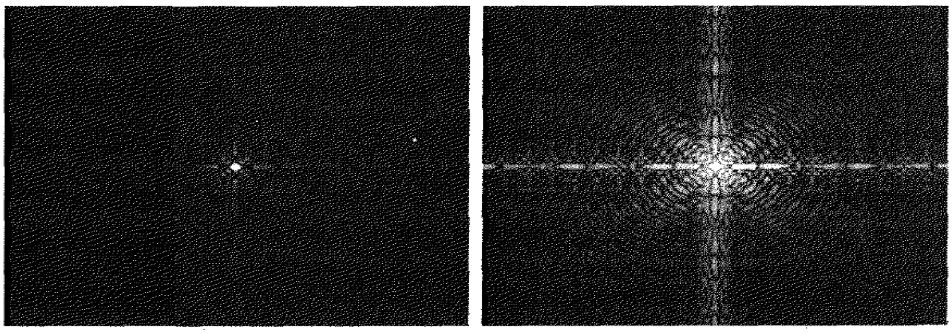
- The general form of the log transformation shown in Fig.1.1 is

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

- Where c is a constant, and it is assumed that r ≥ 0.
- The shape of the log curve in Fig. 1.1 shows that this transformation maps a narrow range of low gray-level values in the input image into a wider range of output levels. The opposite is true of higher values of input levels.

- We would use a transformation of this type to expand the values of dark pixels in an image while compressing 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 with $c = 1$.



- In the above figure 3.5(a) shows a Fourier spectrum with values in the range 0 to 1.5 X 10^6. When these values are scales linearly for display in an 8 bit system, the bright pixels will dominate the display. But if we apply the log transformation with C=1 to the spectrum values then the range of values of the spectrum becomes 0 to 6.2, then the result scaling image will be as shown in the figure 3.5(b).

Power-Law Transformations:

- Power-law transformations have the basic form $s = cr^{\gamma}$ where c and γ are positive constants.
- Sometimes Eq. is written as $s = c(r + \varepsilon)^{\gamma}$ to account for an offset (that is, a measurable output when the input is zero) However, offsets typically are an issue of display calibration and as a result they are normally ignored in Eq. Plots of s versus r for various values of g are shown in Fig. 1.2.

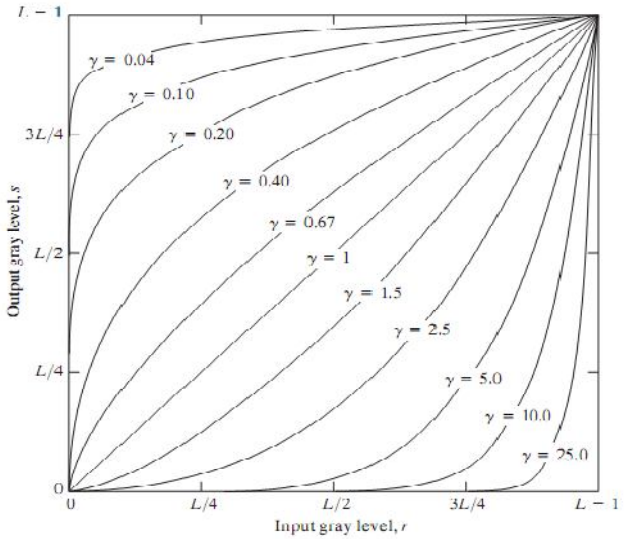


Fig.1.2 Plots of the equation $s = cr^{\gamma}$ for various γ values of (c=1 in all cases).

- We see in Fig.1.2 that curves generated with values of $\gamma > 1$ have exactly the opposite effect as those generated with values of $\gamma < 1$. Finally, we note that Eq. reduces to the identity transformation when $c = \gamma = 1$. A variety of devices used for image capture, printing, and display respond according to a power law. By convention, the exponent in the power-law equation is referred to as gamma. The process used to correct this power-law response phenomenon is called gamma correction.
- For example, cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with exponents varying from approximately 1.8 to 2.5.With reference to the curve for g=2.5 in Fig.1.2, we see that such display systems would tend to produce images that are darker than intended.

Piecewise-Linear Transformation Functions:

- The principal advantage of piecewise linear functions over the types of functions we have discussed above is that the form of piecewise functions can be arbitrarily complex. The principal disadvantage of piecewise functions is that their specification requires considerably more user input.

Contrast stretching:

- Low contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition. The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.
- Figure 1.3 (a) shows a typical transformation used for contrast stretching. The locations of points (r1,s1) and (r2,s2) control the shape of the transformation function. If r1=s1 and r2=s2, the transformation is a linear function that produces no changes in gray levels. If r1=r2,s1=0 and s2=L-1, the transformation becomes a thresholding function that creates a binary image, as illustrated in Fig. 1.3 (b). Intermediate values of (r1 , s1) and (r2 , s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast. In general, $r1 \leq r2$ and $s1 \leq s2$ is assumed so that the function is single valued and monotonically increasing. This condition preserves the order of gray levels, thus preventing the creation of intensity artifacts in the processed image.

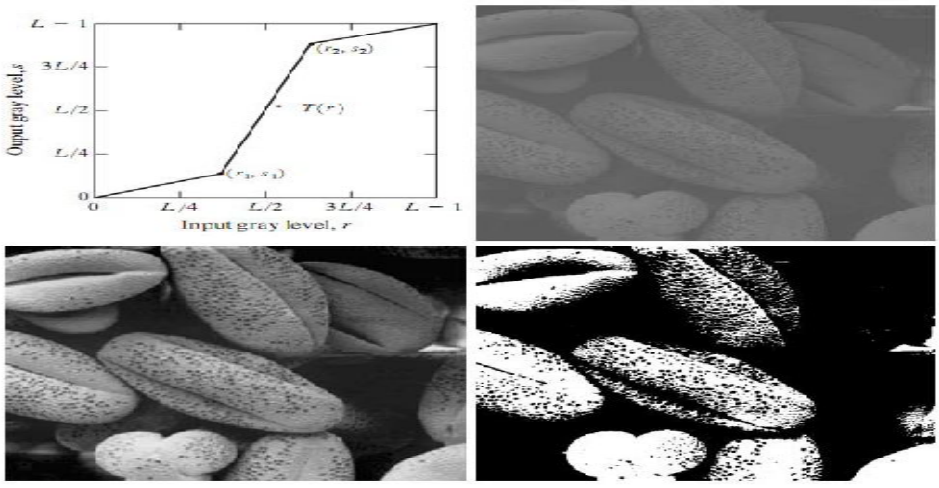


Fig.1.3 Contrast Stretching (a) Form of Transformation function (b) A low-contrast image (c) Result of contrast stretching (d) Result of thresholding.

- Figure 1.3 (b) shows an 8-bit image with low contrast. Fig. 1.3(c) shows the result of contrast stretching, obtained by setting (r1 , s1) = (rmin , 0) and (r2 , s2) = (rmax , L-1) where rmin and rmax denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0,L-1]. Finally, Fig. 1.3 (d) shows the result of using the thresholding function defined previously, with r1 = r2 = m, the mean gray level in the image. The original image on which these results are based is a scanning electron microscope image of pollen, magnified approximately 700 times.

Gray-level slicing:

- There are several ways of doing level slicing, but most of them are variations of two basic themes. One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This transformation, shown in Fig. 1.4 (a), produces a binary image. The second approach, based on the transformation shown in Fig. 1.4 (b), brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image. Figure 1.4(c) shows a gray-scale image, and Fig. 1.4 (d) shows the result of using the transformation in Fig. 1.4 (a).Variations of the two transformations shown in Fig. 1.4 are easy to formulate.

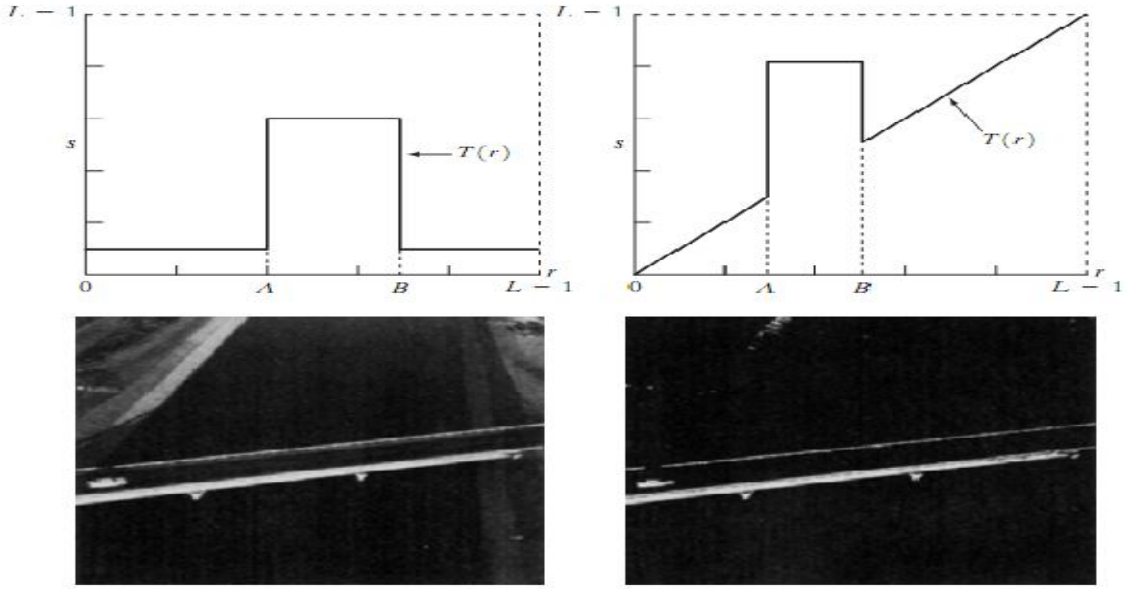


Fig.1.4 (a) This transformation highlights range [A, B] of gray levels and reduce all others to a constant level (b) This transformation highlights range [A, B] but preserves all other levels (c) An image (d) Result of using the transformation in (a).

Bit-plane slicing:

- Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits.

- Imagine that the image is composed of eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bit plane 7 for the most significant bit.
- In terms of 8-bit bytes, plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits.
- Figure 1.5 illustrates these ideas, and Fig. 1.7 shows the various bit planes for the image shown in Fig.1.6. Note that the higher-order bits (especially the top four) contain the majority of the visually significant data. The other bit planes contribute to more subtle details in the image. Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel.

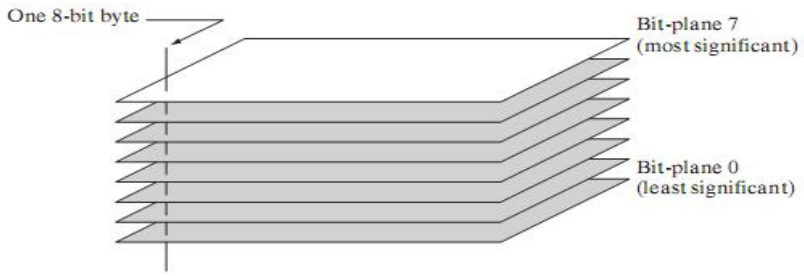


Fig.1.5 Bit-plane representation of an 8-bit image.

- In terms of bit-plane extraction for an 8-bit image, it is not difficult to show that the (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray level transformation function that (1) maps all levels in the image between 0 and 127 to one level (for example, 0); and (2) maps all levels between 129 and 255 to another (for example, 255).

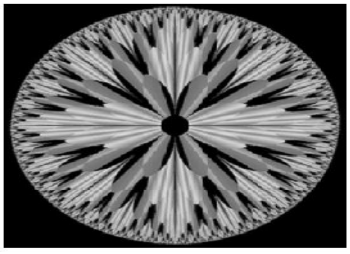


Fig.1.6 An 8-bit fractal image

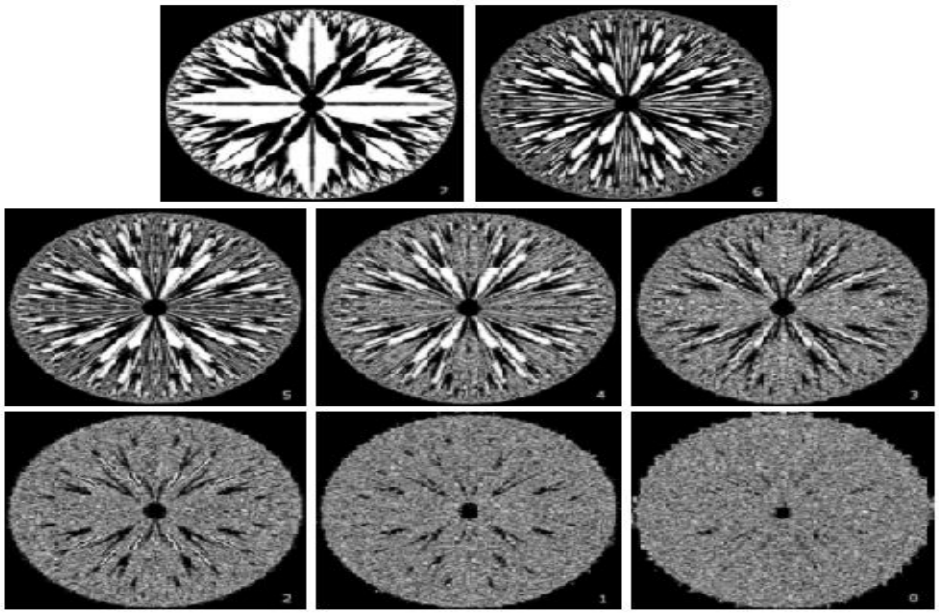


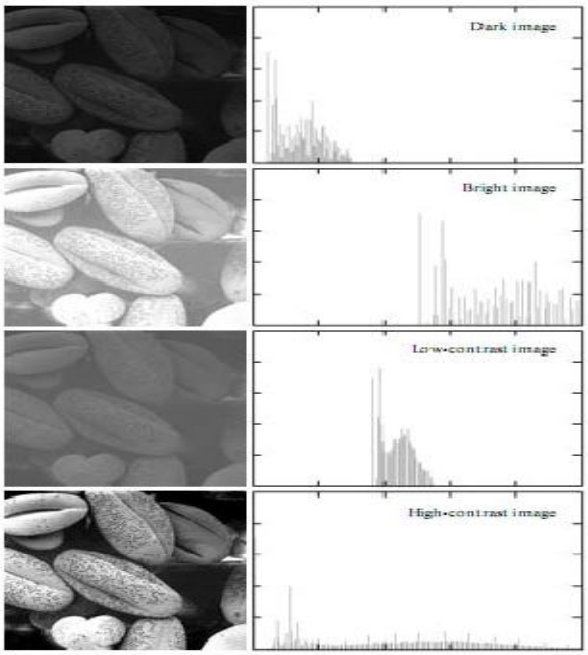
Fig.1.7 The eight bit planes of the image in Fig.1.6. The number at the bottom, right of each image identifies the bit plane.

Histogram Processing:

- The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $h(r_k) = (n_k)$, where r_k is the kth gray level and n_k is the number of pixels in the image having gray level r_k. It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by

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

  for k=0,1,..... .,L-1. Loosely speaking, $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k. Note that the sum of all components of a normalized histogram is equal to 1.
- Histogram manipulation can be used effectively for image enhancement for real-time image processing.
- Consider Fig. 3, which is the pollen image shown in four basic gray-level characteristics: dark, light, low contrast, and high contrast. The right side of the figure shows the histograms corresponding to these images. The horizontal axis of each histogram plot corresponds to gray level values, r_k. The vertical axis corresponds to values of $h(r_k) = n_k$ or $p(r_k) = n_k/n$ if the values are normalized. Thus, as indicated previously, these histogram plots are simply plots of $h(r_k) = n_k$ versus r_k or $p(r_k) = n_k/n$ versus r_k.



Questions: **Fig.3 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.**

- We note in the dark image that the components of the histogram are concentrated on the low (dark) side of the gray scale. Similarly, the components of the histogram of the bright image are biased toward the high side of the gray scale. An image with low contrast has a histogram that will be narrow and will be centered toward the middle of the gray scale. For a monochrome image this implies a dull, washed- out gray look.
- Finally, we see that the components of the histogram in the high-contrast image cover a broad range of the gray scale and, further, that the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others. Intuitively, it is reasonable to conclude that an image, whose pixels tend to occupy the entire range of possible gray levels and, in addition, tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones.

Histogram Equalization:

- Consider for a moment continuous functions, and let the variable r represent the gray levels of the image to be enhanced. We assume that r has been normalized to the interval [0, 1], with r=0 representing black and r=1 representing white. Later, we consider a discrete formulation and allow pixel values to be in the interval [0, L-1]. For any r satisfying the aforementioned conditions, we focus attention on transformations of the form

$$s = T(r) \qquad 0 \le r \le 1$$

that produce a level s for every pixel value r in the original image. For reasons that will become obvious shortly, we assume that the transformation function T(r) satisfies the following conditions:

a. T(r) is single-valued and monotonically increasing in the interval $0 \le r \le 1$; and

b. $0 \le T(r) \le 1$ for $0 \le r \le 1$.

- The requirement in (a) that T(r) be single valued is needed to guarantee that the inverse transformation will exist, and the monotonicity condition preserves the increasing order from black to white in the output image. A transformation function that is not monotonically increasing could result in at least a section of the intensity range being inverted, thus producing some inverted gray levels in the output image. Finally, condition (b) guarantees that the output gray levels will be in the same range as the input levels.
- Figure 4.1 gives an example of a transformation function that satisfies these two conditions. The inverse transformation from s back to r is denoted

$$r = T^{-1}(s) \qquad 0 \le s \le 1.$$

- It can be shown by example that even if T(r) satisfies conditions (a) and (b), it is possible that the corresponding inverse T-1 (s) may fail to be single valued.

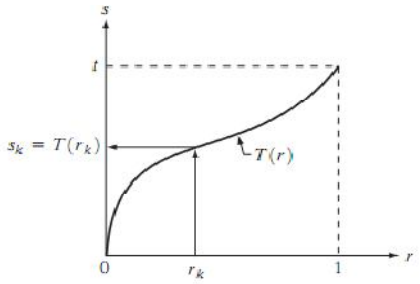


Fig.4.1 A gray-level transformation function that is both single valued and monotonically increasing.

- The gray levels in an image may be viewed as random variables in the interval [0, 1].One of the most fundamental descriptors of a random variable is its probability density function (PDF).Let $p_r(r)$ and $p_s(s)$ denote the probability density functions of random variables r and s, respectively, where the subscripts on p are used to denote that pr and ps are different functions. A basic result from an elementary probability theory is that, if $p_r(r)$ and T(r) are known and T^{-1} (s) satisfies condition (a), then the probability density function $p_s(s)$ of the transformed variable s can be obtained using a rather simple formula:

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

- Thus, the probability density function of the transformed variable, s, is determined by the gray level PDF of the input image and by the chosen transformation function. A transformation function of particular importance in image processing has the form

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

- Where w is a dummy variable of integration. The right side of Eq. above is recognized as the cumulative distribution function (CDF) of random variable r. Since probability density functions are always positive, and recalling that

the integral of a function is the area under the function, it follows that this transformation function is single valued and monotonically increasing, and, therefore, satisfies condition (a). Similarly, the integral of a probability density function for variables in the range [0, 1] also is in the range [0, 1], so condition (b) is satisfied as well. Substituting this result for dr/ds, and keeping in mind that all probability values are positive, yields

$$\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \\ &= p_r(r)\left|\frac{1}{p_r(r)}\right| \\ &= 1 \qquad 0 \le s \le 1. \end{aligned}$$

- Because ps(s) is a probability density function, it follows that it must be zero outside the interval [0, 1] in this case because its integral over all values of s must equal 1.We recognize the form of ps(s) as a uniform probability density function. Simply stated, we have demonstrated that performing the transformation function yields a random variable s characterized by a uniform probability density function. It is important to note from Eq. discussed above that T(r) depends on pr(r), but, as indicated by Eq. after it, the resulting ps(s) always is uniform, independent of the form of pr(r). For discrete values we deal with probabilities and summations instead of probability density functions and integrals. The probability of occurrence of gray level r in an image is approximated by

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots, L-1$$

- Where, n is the total number of pixels in the image, nk is the number of pixels that have gray level rk, and L is the total number of possible gray levels in the image. The discrete version of the transformation function given in Eq. is

$$\begin{aligned} s_k = T(r_k) &= \sum_{j=0}^{k} p_r(r_j) \\ &= \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0, 1, 2, \ldots, L-1. \end{aligned}$$

- Thus, a processed (output) image is obtained by mapping each pixel with level rk in the input image into a corresponding pixel with level sk in the output image. As indicated earlier, a plot of pr (rk) versus rk is called a histogram. The transformation (mapping) is called histogram equalization or histogram linearization.

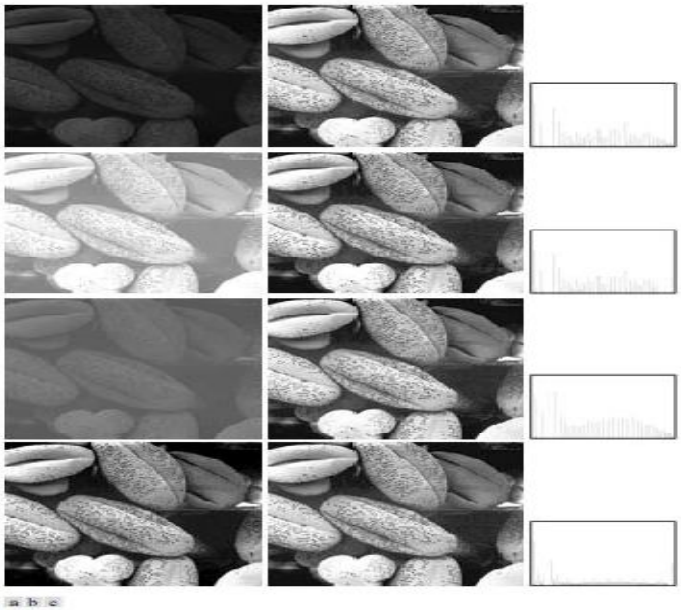


Fig.4.2 (a) Images from Fig.3 (b) Results of histogram equalization. (c) Corresponding histograms.

- The inverse transformation from s back to r is denoted by

$$r_k = T^{-1}(s_k) \qquad k = 0, 1, 2, \ldots, L-1$$

Histogram Matching (Specification):

- In particular, it is useful sometimes to be able to specify the shape of the histogram that we wish the processed image to have. The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

Development of the method:

- In this notation, r and z denote the gray levels of the input and output (processed) images, respectively. We can estimate $p_r(r)$ from the given input image, while $p_z(z)$ is the specified probability density function that we wish the output image to have. Let s be a random variable with the property

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

- Where w is a dummy variable of integration. We recognize this expression as the continuo's version of histogram equalization. Suppose next that we define a random variable z with the property

$$G(z) = \int_0^z p_z(t)\, dt = s$$

- Where t is a dummy variable of integration. It then follows from these two equations that G(z)=T(r) and, therefore, that z must satisfy the condition

$$z = G^{-1}(s) = G^{-1}[T(r)].$$

- The transformation T(r) can be obtained once pr(r) has been estimated from the input image. Similarly, the transformation function G(z) can be obtained

because $p_z(z)$ is given. Assuming that G-1 exists and that it satisfies conditions (a) and (b) in the histogram equalization process, the above three equations show that an image with a specified probability density function can be obtained from an input image by using the following procedure:

1. Obtain the transformation function T(r).
2. To obtain the transformation function G(z).
3. Obtain the inverse transformation function G-1
4. Obtain the output image by applying above Eq. to all the pixels in the input image.

- The result of this procedure will be an image whose gray levels, z, have the specified probability density function $p_z(z)$. Although the procedure described is straightforward in principle, it is possible in practice to obtain analytical expressions for T(r) and for G^{-1}.

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

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

- Where n is the total number of pixels in the image, nj is the number of pixels with gray level r_j, and L is the number of discrete gray levels. Similarly, the discrete formulation is obtained from the given histogram $p_z(zi)$, i=0, 1, 2,……, L-1, and has the form

$$v_k = G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k \qquad k = 0, 1, 2, \ldots, L-1.$$

- As in the continuo's case, we are seeking values of z that satisfy this equation. The variable vk was added here for clarity in the discussion that follows. Finally, the discrete version of the above Eqn. is given by

$$z_k = G^{-1}[T(r_k)] \qquad k = 0, 1, 2, \ldots, L-1$$

Or

$$z_k = G^{-1}(s_k) \qquad k = 0, 1, 2, \ldots, L-1.$$

Enhancement using arithmetic and logic operators:

Image Subtraction:

- The difference between two images f(x, y) and h(x, y), expressed as

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

is obtained by computing the difference between all pairs of corresponding pixels from f and h.

- The use of subtraction is the enhancement of differences between images. The higher order bit planes of an image carry a significant amount of visually

relevant detail, while the lower planes contribute more to fine (often imperceptible) detail.

- Figure 7(a) shows the fractal image used earlier to illustrate the concept of bit planes. Figure 7(b) shows the result of discarding (setting to zero) the four least significant bit planes of the original image. The images are nearly identical visually, with the exception of a very slight drop in overall contrast due to less variability of the gray level values in the image of Fig. 7(b).The pixel-by-pixel difference between these two images is shown in Fig. 7(c).The differences in pixel values are so small that the difference image appears nearly black when displayed on an 8-bit display. In order to bring out more detail, we can perform a contrast stretching transformation. We chose histogram equalization, but an appropriate power-law transformation would have done the job also. The result is shown in Fig. 7(d). This is a very useful image for evaluating the effect of setting to zero the lower-order planes.

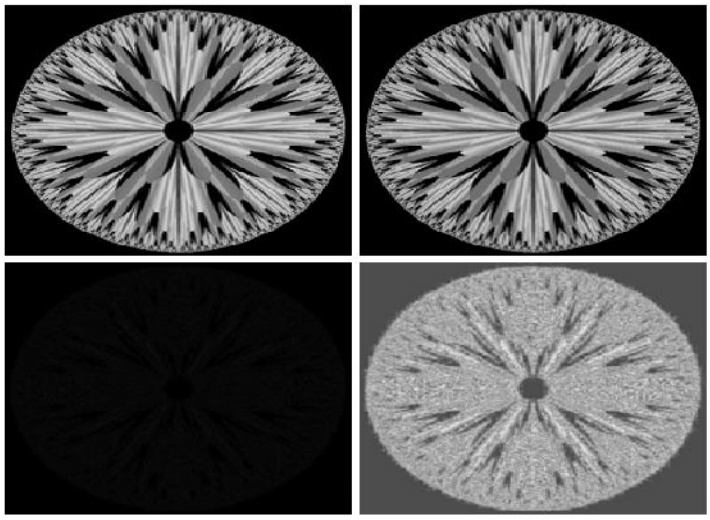


Fig.7 (a) Original fractal image (b) Result of setting the four lower-order bit planes to zero (c) Difference between (a) and (b) (d) Histogram equalized difference image.

- The use of image subtraction is in the area of medical imaging called mask mode radiography. In this case h(x, y), the mask, is an X-ray image of a region of a patient's body captured by an intensified TV camera (instead of traditional X-ray film) located opposite an X-ray source. The procedure consists of injecting a contrast medium into the patient's bloodstream, taking a series of images of the same anatomical region as h(x, y), and subtracting this mask from the series of incoming images after injection of the contrast medium. The net effect of subtracting the mask from each sample in the incoming stream of TV images is that the areas that are different between f(x, y) and h(x, y) appear in the output image as enhanced detail. Because images can be captured at TV rates, this procedure in essence gives a movie showing how the contrast medium propagates through the various arteries in the area being observed.

Image averaging:

- Consider a noisy image g(x, y) formed by the addition of noise h(x, y) to an original image f(x, y); that is,

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

- Where the assumption is that at every pair of coordinates (x, y) the noise is uncorrelated and has zero average value. The objective of the following procedure is to reduce the noise content by adding a set of noisy images, $\{g_i(x, y)\}$. If the noise satisfies the constraints just stated, it can be shown that if an image $\bar{g}(x, y)$ is formed by averaging K different noisy images,

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

Then it follows that

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

And

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

- Where $E\{\bar{g}(x, y)\}$ is the expected value of $\bar{g}$ and $\sigma^2_{\bar{g}(x,y)}$ and $\sigma^2_{\bar{\eta}(x,y)}$ are the variances of $\bar{g}$ and η, all at coordinates (x, y). The standard deviation at any point in the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \sigma_{\eta(x,y)}.$$

- As K increases, the above equations indicate that the variability (noise) of the pixel values at each location (x, y) decreases. Because $E\{\bar{g}(x, y)\} = f(x, y),$ this means that $\bar{g}(x, y)$ approaches f(x, y) as the number of noisy images used in the averaging process increases. In practice, the images $g_i(x, y)$ must be registered (aligned) in order to avoid the introduction of blurring and other artifacts in the output image.

Basics of Spatial Filtering:

- Some neighborhood operations work with the values of the image pixels in the neighborhood and the corresponding values of a sub image that has the same dimensions as the neighborhood. The sub image is called a filter, mask, kernel, template, or window.
- The values in a filter sub image are referred to as coefficients, rather than pixels. The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain. We use the term spatial filtering to differentiate this type of process from the more traditional frequency domain filtering.
- The mechanics of spatial filtering are illustrated in Fig.9.1. The process consists simply of moving the filter mask from point to point in an image. At each point (x, y), the response of the filter at that point is calculated using a

predefined relationship. The response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. For the 3 x 3 mask shown in Fig. 9.1, the result (or response), R, of linear filtering with the filter mask at a point (x, y) in the image is

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

- Which we see is the sum of products of the mask coefficients with the corresponding pixels directly under the mask. Note in particular that the coefficient w(0, 0) coincides with image value f(x, y), indicating that the mask is centered at (x, y) when the computation of the sum of products takes place. For a mask of size m x n, we assume that m=2a+1 and n=2b+1,where a and b are nonnegative integers.

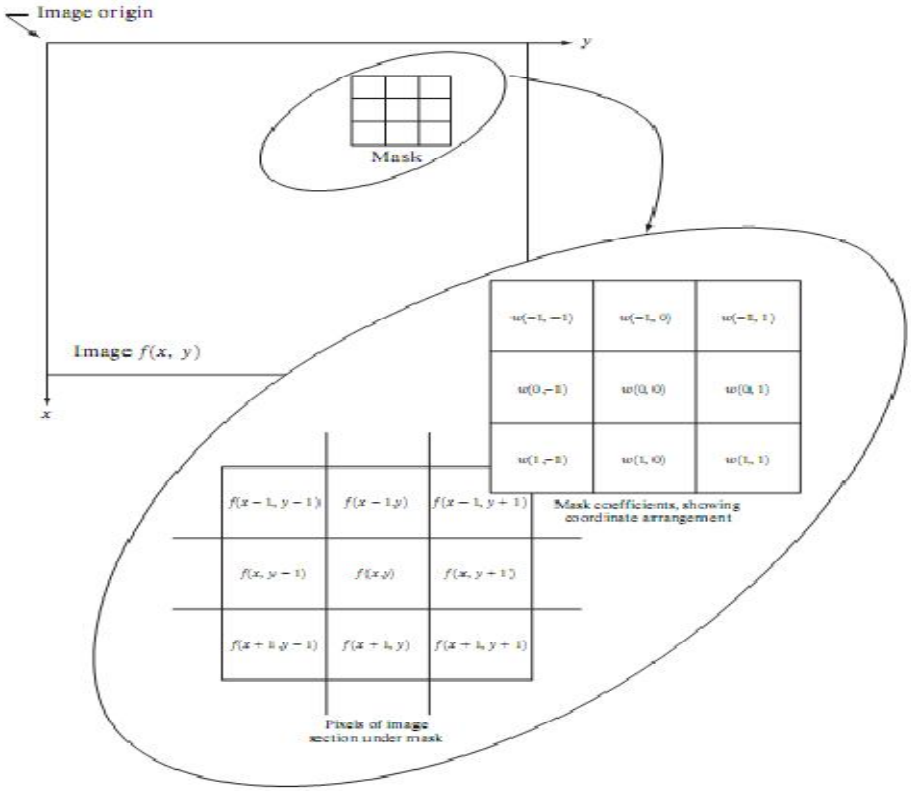


Fig.9.1 The mechanics of spatial filtering. The magnified drawing shows a 3X3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

- In general, linear filtering of an image f of size M x N with a filter mask of size m x n is given by the expression:

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

- Where, from the previous paragraph, a=(m-1)/2 and b=(n-1)/2. To generate a complete filtered image this equation must be applied for x=0,1,2,……, M-1

and y=0,1,2,……, N-1. In this way, we are assured that the mask processes all pixels in the image. It is easily verified when m=n=3 that this expression reduces to the example given in the previous paragraph.

- The process of linear filtering is similar to a frequency domain concept called convolution. For this reason, linear spatial filtering often is referred to as "convolving a mask with an image." Similarly, filter masks are sometimes called convolution masks. The term convolution kernel also is in common use. When interest lies on the response, R, of an m x n mask at any point (x, y), and not on the mechanics of implementing mask convolution, it is common practice to simplify the notation by using the following expression:

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

- where the w's are mask coefficients, the z's are the values of the image gray levels corresponding to those coefficients, and mn is the total number of coefficients in the mask. For the 3 x 3 general mask shown in Fig.9.2 the response at any point (x, y) in the image is given by

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

- An important consideration in implementing neighborhood operations for spatial filtering is the issue of what happens when the center of the filter approaches the border of the image. Consider for simplicity a square mask of size n x n. At least one edge of such a mask will coincide with the border of the image when the center of the mask is at a distance of (n-1)/2 pixels away from the border of the image. If the center of the mask moves any closer to the border, one or more rows or columns of the mask will be located outside the image plane. There are several ways to handle this situation. The simplest is to limit the excursions of the center of the mask to be at a distance no less than (n-1)/2 pixels from the border. The resulting filtered image will be smaller than the original, but all the pixels in the filtered imaged will have been processed with the full mask. If the result is required to be the same size as the original, then the approach typically employed is to filter all pixels only with the section of the mask that is fully contained in the image. With this approach, there will be bands of pixels near the border that will have been processed with a partial filter mask. Other

approaches include “padding” the image by adding rows and columns of 0’s (or other constant gray level), or padding by replicating rows or columns. The padding is then stripped off at the end of the process.

- This keeps the size of the filtered image the same as the original, but the values of the padding will have an effect near the edges that becomes more prevalent as the size of the mask increases. The only way to obtain a perfectly filtered result is to accept a somewhat smaller filtered image by limiting the excursions of the center of the filter mask to a distance no less than (n-1)/2 pixels from the border of the original image.

Smoothing Spatial Filters:

- Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and also by non-linear filtering.

(1) Smoothing Linear Filters:

- The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called averaging filters. The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced “sharp” transitions in gray levels. Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels, so averaging filters have the undesirable side effect that they blur edges. Another application of this type of process includes the smoothing of false contours that result from using an insufficient number of gray levels.

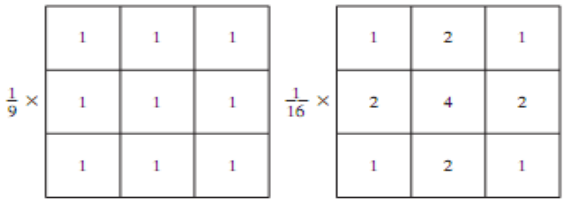


Fig.10.1 Two 3 x 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

- A major use of averaging filters is in the reduction of “irrelevant” detail in an image. By “irrelevant” we mean pixel regions that are small with respect to the size of the filter mask.
- Figure 10.1 shows two 3 x 3 smoothing filters. Use of the first filter yields the standard average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask in

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

- which is the average of the gray levels of the pixels in the 3 x 3 neighborhood defined by the mask. Note that, instead of being 1/9, the coefficients of the filter are all 1's.The idea here is that it is computationally more efficient to have coefficients valued 1. At the end of the filtering process the entire image is divided by 9. An m x n mask would have a normalizing constant equal to 1/mn.
- A spatial averaging filter in which all coefficients are equal is sometimes called a box filter.
- The second mask shown in Fig.10.1 yields weighted average, terminology used to indicate that pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others.
- In the mask shown in Fig. 10.1(b) the pixel at the center of the mask is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation of the average. The other pixels are inversely weighted as a function of their distance from the center of the mask. The diagonal terms are further away from the center than the orthogonal neighbors (by a factor of $\sqrt{2}$) and, thus, are weighed less than these immediate neighbors of the center pixel.
- The basic strategy behind weighing the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process. We could have picked other weights to accomplish the same general objective. However, the sum of all the coefficients in the mask of Fig. 10.1(b) is equal to 16, an attractive feature for computer implementation because it has an integer power of 2. In practice, it is difficult in general to see differences between images smoothed by using either of the masks in Fig. 10.1, or similar arrangements, because the area these masks span at any one location in an image is so small.
- The general implementation for filtering an M x N image with a weighted averaging filter of size m x n (m and n odd) is given by the expression

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

(2) Order-Statistics Filters:

- Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result. The best-known example in this category is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median). Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of impulse noise, also called salt-and-

pepper noise because of its appearance as white and black dots superimposed on an image.

- The median, ε, of a set of values is such that half the values in the set are less than or equal to ε, and half are greater than or equal to ε. In order to perform median filtering at a point in an image, we first sort the values of the pixel in question and its neighbors, determine their median, and assign this value to that pixel. For example, in a 3 x 3 neighborhood the median is the 5th largest value, in a 5 x 5 neighborhood the 13th largest value, and so on. When several values in a neighborhood are the same, all equal values are grouped. For example, suppose that a 3 x 3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). These values are sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20. Thus, the principal function of median filters is to force points with distinct gray levels to be more like their neighbors. In fact, isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2 / 2$ (one-half the filter area), are eliminated by an n x n median filter. In this case "eliminated" means forced to the median intensity of the neighbors. Larger clusters are affected considerably less.

Use of Second Derivatives for Enhancement–The Laplacian:

- The approach basically consists of defining a discrete formulation of the second-order derivative and then constructing a filter mask based on that formulation. We are interested in isotropic filters, whose response is independent of the direction of the discontinuities in the image to which the filter is applied.
- In other words, isotropic filters are rotation invariant; in the sense that rotating the image and then applying the filter gives the same result as applying the filter to the image first and then rotating the result.

Development of the method:

- It can be shown (Rosenfeld and Kak [1982]) that the simplest isotropic derivative operator is the Laplacian, which, for a function (image) f(x, y) of two variables, is defined as

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

- Because derivatives of any order are linear operations, the Laplacian is a linear operator. In order to be useful for digital image processing, this equation needs to be expressed in discrete form. There are several ways to define a digital Laplacian using neighborhoods. digital second Taking into account that we now have two variables, we use the following notation for the partial second-order derivative in the x-direction:

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

and, similarly in the y-direction, as

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

- The digital implementation of the two-dimensional Laplacian in Eq. is obtained by summing these two components

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

- This equation can be implemented using the mask shown in Fig.11.1(a), which gives an isotropic result for rotations in increments of 90°.
- The diagonal directions can be incorporated in the definition of the digital Laplacian by adding two more terms to Eq., one for each of the two diagonal directions. The form of each new term is the same as either Eq.

0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1
0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−1	−1	−1

Fig.11.1. (a) Filter mask used to implement the digital Laplacian (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

- but the coordinates are along the diagonals. Since each diagonal term also contains a –2f(x, y) term, the total subtracted from the difference terms now would be –8f(x, y). The mask used to implement this new definition is shown in Fig.11.1(b). This mask yields isotropic results for increments of 45°. The other two masks shown in Fig. 11 also are used frequently in practice.
- They are based on a definition of the Laplacian that is the negative of the one we used here. As such, they yield equivalent results, but the difference in sign must be kept in mind when combining (by addition or subtraction) a Laplacian-filtered image with another image.
- Because the Laplacian is a derivative operator, its use highlights gray-level discontinuities in an image and deemphasizes regions with slowly varying gray levels. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background. Background features can be “recovered” while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images. As noted in the previous paragraph, it is important to keep in mind which definition of the Laplacian is used. If the definition used has a negative center coefficient, then we subtract, rather than add, the Laplacian image to obtain a sharpened result. Thus, the basic way in which we use the Laplacian for image enhancement is as follows:

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

Use of First Derivatives for Enhancement—The Gradient:

- First derivatives in image processing are implemented using the magnitude of the gradient. For a function f(x, y), the gradient of f at coordinates (x, y) is defined as the two-dimensional column vector

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

- The magnitude of this vector is given by

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

- The components of the gradient vector itself are linear operators, but the magnitude of this vector obviously is not because of the squaring and square root operations. On the other hand, the partial derivatives are not rotation invariant (isotropic), but the magnitude of the gradient vector is. Although it is not strictly correct, the magnitude of the gradient vector often is referred to as the gradient.
- The computational burden of implementing over an entire image is not trivial, and it is common practice to approximate the magnitude of the gradient by using absolute values instead of squares and square roots:

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

- This equation is simpler to compute and it still preserves relative changes in gray levels, but the isotropic feature property is lost in general. However, as in the case of the Laplacian, the isotropic properties of the digital gradient defined in the following paragraph are preserved only for a limited number of rotational increments that depend on the masks used to approximate the derivatives. As it turns out, the most popular masks used to approximate the gradient give the same result only for vertical and horizontal edges and thus the isotropic properties of the gradient are preserved only for multiples of 90°.
- As in the case of the Laplacian, we now define digital approximations to the preceding equations, and from there formulate the appropriate filter masks. In order to simplify the discussion that follows, we will use the notation in Fig. 11.2 (a) to denote image points in a 3 x 3 region. For example, the center point, z5 , denotes f(x, y), z1 denotes f(x-1, y-1), and so on. The simplest approximations to a first-order derivative that satisfy the conditions

stated in that section are $G_x = (z_8 - z_5)$ and $G_y = (z_6 - z_5)$. Two other definitions proposed by Roberts [1965] in the early development of digital image processing use cross differences:

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

- we compute the gradient as

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

- If we use absolute values, then substituting the quantities in the equations gives us the following approximation to the gradient:

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

- This equation can be implemented with the two masks shown in Figs. 11.2 (b) and(c). These masks are referred to as the Roberts cross-gradient operators. Masks of even size are awkward to implement. The smallest filter mask in which we are interested is of size 3 x 3.An approximation using absolute values, still at point z5 , but using a 3*3 mask, is

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|.$$

- The difference between the third and first rows of the 3 x 3 image region approximates the derivative in the x-direction, and the difference between the third and first columns approximates the derivative in the y-direction. The masks shown in Figs. 11.2 (d) and (e), called the Sobel operators. The idea behind using a weight value of 2 is to achieve some smoothing by giving more importance to the center point. Note that the coefficients in all the masks shown in Fig. 11.2 sum to 0, indicating that they would give a response of 0 in an area of constant gray level, as expected of a derivative operator.

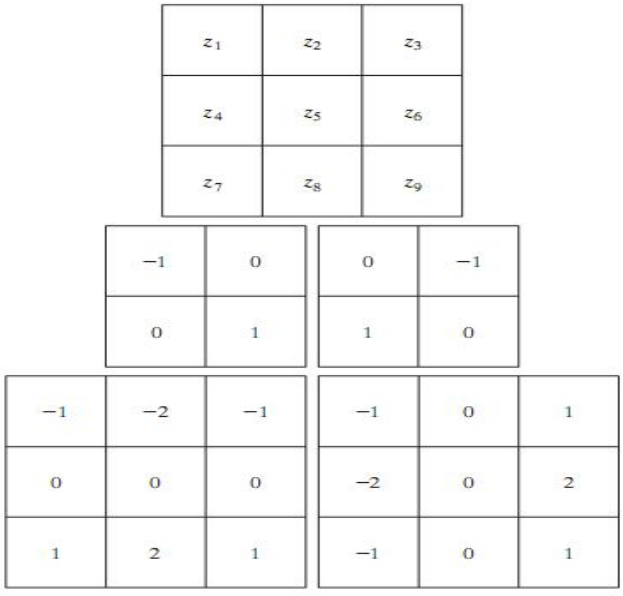


Fig.11.2 A 3 x 3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

UNIT-II
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. Spatial domain processes will be denoted by the expression_____
2. Power-law transformations have the basic form _____
3. The general form of the log transformation is _____
4. The negative of an image with gray levels in the range [0, L-1] is obtained by using the transformation function _____
5. The use of image subtraction is in the area of medical imaging called _____.
6. The sub image is called a _____.
7. The values in a filter sub image are referred to as _____, rather than pixels.
8. A spatial averaging filter in which all coefficients are equal is sometimes called a_____.
9. Smoothing filters are mostly used in _____.
10. In spatial domain, which of the following operation is done on the pixels in sharpening the image? [ ]

A, Integration B, Average C, Median D, Differentiation

11. Sum of all components in normalized histogram is equal to

A, 100 B,2 C, 0 d, 1 [ ]

12. If r be the gray-level of image before processing and s after processing then which expression defines the negative transformation, for the gray-level in the range [0, L-1]? [ ]

A, s = L – 1 – r

B, s = cr□, c and □ are positive constants

C, s = c log (1 + r), c is a constant and r ≥ 0

D, none of the mentioned

13. The power-law transformation is given as: s = cr□, c and □ are positive constants, and r is the gray-level of image before processing and s after processing. Then, for what value of c and □ does power-law transformation becomes identity transformation? [ ]

A, c = 1 and □ < 1 B, c = 1 and □ > 1 C, c = -1 and □ = 0 D, c = □ = 1

14. The power-law transformation is given as: s = cr□, c and □ are positive constants, and r is the gray-level of image before processing and s after processing. What happens if we increase the gamma value from 0.3 to 0.7? [ ]

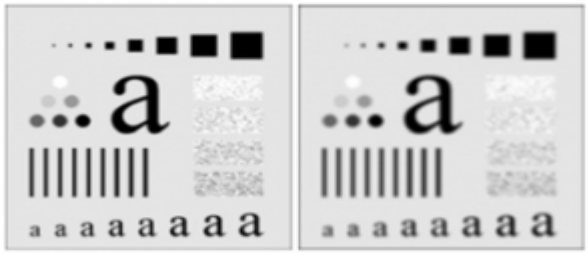
A, The contrast increases and the detail increases

B, The contrast decreases and the detail decreases

C, The contrast increases and the detail decreases

D, The contrast decreases and the detail increases

15. If the size of the averaging filter used to smooth the original image to first image is 9, then what would be the size of the averaging filter used in smoothing the same original picture to second in second image? [ ]



A,3 B,5 C,9 D,15

16. In power-law transformation what happens if we change the gamma value from 0.9 to 0.2? [ ]

A, The contrast increases and the detail increases
B, The contrast decreases and the detail decreases
C, The contrast increases and the detail decreases
D, The contrast decreases and the detail increases

17. What is the maximum area of the cluster that can be eliminated by using an n×n median filter? [ ]

A,n^2 B, $n^2/2$ C, $2*n^2$ D, n

Explanation: Isolated clusters of pixels that are light or dark with respect to their neighbours, and whose area is less than n2/2, i.e., half the area of the filter, can be eliminated by using an n×n median filter

18. In contrast stretching, if r1=s1 and r2=s2 then which of the following is true? [ ]

A, The transformation is not a linear function that produces no changes in gray levels
B, The transformation is a linear function that produces no changes in gray levels
C, The transformation is a linear function that produces changes in gray levels
D, The transformation is not a linear function that produces changes in gray levels

19. If f(x,y) is an image function of two variables, then the first order derivative of a one dimensional function, f(x) is: [ ]

A, f(x+1)-f(x) B, f(x)-f(x+1) C, f(x-1)-f(x+1) D, f(x)+f(x-1)

20. The derivative of digital function is defined in terms of difference. Then, which of the following defines the second order derivative $\partial 2\ f/\partial x2$ = _____ of a one-dimensional function f(x)? [ ]

A, f(x+1)-f(x) B, f(x+1)+f(x-1)-2f(x)
C, All of the mentioned depending upon the time when partial derivative will be dealt along two spatial axes
D, None of the mentioned

SECTION-B

SUBJECTIVE QUESTIONS

1. What is the objective of image enhancement? Define spatial domain. Define point processing.
2. What is meant by image enhancement by point processing? Discuss any two methods in it.
3. Define histogram of a digital image. Explain how histogram is useful in image enhancement?
4. Write about histogram equalization.
5. Write about histogram specification.
6. What is meant by image subtraction? Discuss various areas of application of image subtraction.
7. Explain about image averaging process.
8. Discuss about the mechanics of filtering in spatial domain. Mention the points to be considered in implementation neighborhood operations for spatial filtering.
9. Write about Smoothing Spatial filters.
10. What is meant by the Gradiant and the Laplacian? Discuss their role in image enhancement.
11. Implement the two-dimensional Laplacian equation for a pixel (x,y)
12. Implement first derivative enhancement function
13. What effect would setting to zero the lower-order bit planes have on the histogram of an image in general?
14. What would be the effect on the histogram if we set to zero the higher order bit planes instead?
15. Develop a procedure for computing the median of an n*n neighborhood.
16. Propose a technique for updating the median as the center of the neighborhood is moved from pixel to pixel.
17. Give a 3*3 mask for performing unsharp masking in a single pass through an image.

UNIT - III

UNIT - III: Color Image Processing

Introduction, Color fundamentals, color models, pseudo color image processing, basics of full color image processing, color transformations, color image smoothing and sharpening, color segmentation.

Introduction:

- Color of an object is determined by the nature of the light reflected from it.
- When a beam of sunlight passes through a glass prism, the emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other.
- As Fig.1 shows, the color spectrum may be divided into six broad regions: violet, blue, green, yellow, orange, and red.
- When viewed in full color (Fig.2), no color in the spectrum ends abruptly, but rather each color blends smoothly into the next.

Fig. 1 Color spectrum seen by passing white light through a prism

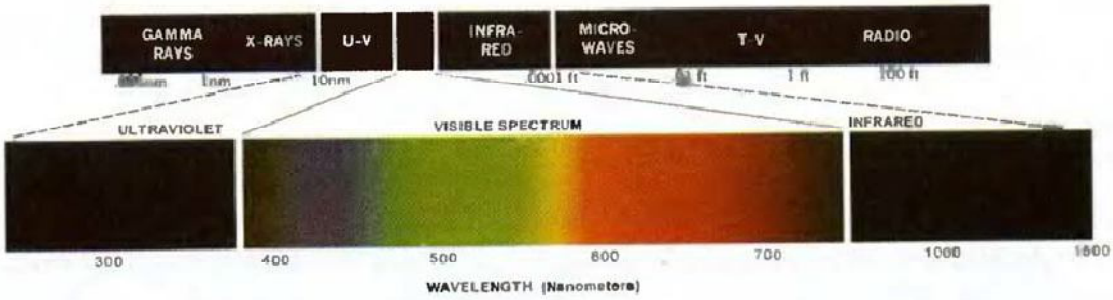


Fig. 2 Wavelengths comprising the visible range of the electromagnetic spectrum.

- As shown in Fig.2, visible light is composed narrow band of frequencies in the electromagnetic spectrum. A body that reflects light that is balanced in all visible wavelengths appears white to the observer.
- However, a body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color. For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelengths.
- Characterization of light is central to the science of color. If the light is achromatic (void of color), its only attribute is its **intensity**, or amount. **Achromatic light** is what viewers see on a **black and white television** set.
- Three basic quantities are used to describe the quality of a chromatic light source: radiance, luminance, and brightness.
- **Radiance:** Radiance is the total amount of energy that flows from the light source, and it is usually measured in watts (W).
- **Luminance:** Luminance, measured in lumens (lm), gives a measure of the amount of energy an observer perceives from a light source. For example, light emitted from a source operating in the far infrared region of the spectrum could have significant energy (radiance), but an observer would hardly perceive it; its luminance would be almost zero.
- **Brightness:** Brightness is a subjective descriptor that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.

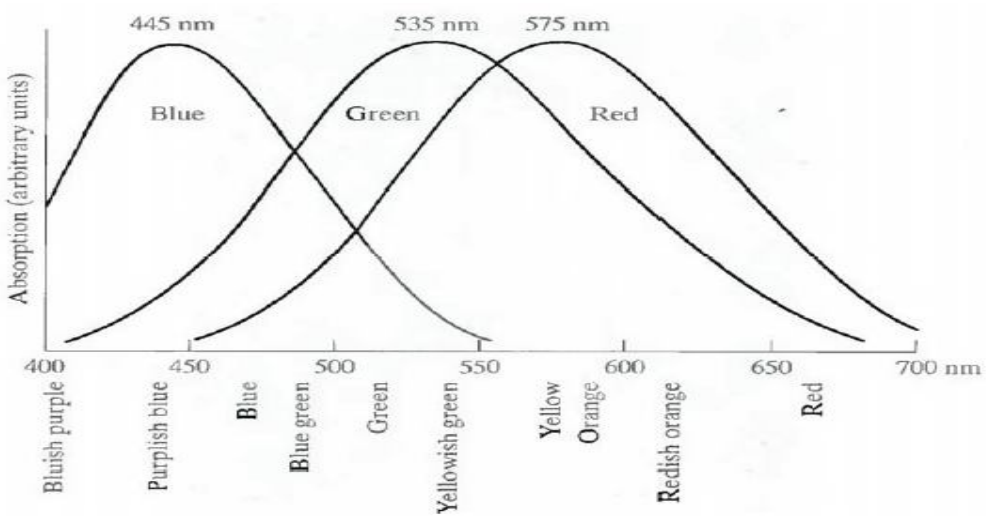


Fig.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

- Cones are the sensors in the eye responsible for color vision. Detailed experimental evidence has established that the 6 to 7 million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue. Approximately 65% of all cones are sensitive to red light, 33% are sensitive to green light, and only about 2% are sensitive to blue (but the blue cones are the most sensitive). Figure3 shows average experimental curves detailing the absorption of light by the red, green, and blue cones in the eye. Due to these absorption characteristics of the human eye, colors arc seen as variable combinations of the so- called primary colors red (R), green (G), and blue (B).
- The primary colors can be added to produce the secondary colors of light --magenta (red plus blue), cyan (green plus blue), and yellow (red plus green). Mixing the three primaries, or a secondary with its opposite primary color, in the right intensities produces white light.
- The characteristics generally used to distinguish one color from another are brightness, hue, and saturation. **Brightness** embodies the chromatic notion of intensity. **Hue** is an attribute associated with the dominant wavelength in a mixture of light waves. Hue represents dominant color as perceived by an observer. **Saturation** refers to the relative purity or the amount of white light mixed with a hue. The pure spectrum colors are fully saturated. Colors such as **pink** (red and white) and **lavender** (violet and white) are less saturated, with the degree of saturation being inversely proportional to the amount of white light-added.
- Hue and saturation taken together are called **chromaticity**, and. therefore, a color may be characterized by its brightness and chromaticity.
- The amount of red, green or blue in a color is known as **Tristimulus** Values and they are denoted as X, Y and Z respectively. A color is then specified by its **Trichromatic coefficients** denoted as:

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

and

$$z = \frac{Z}{X + Y + Z}.$$

- It is noted from above equations that $x+y+z=1$
- For any value of x(red) and y(green) the corresponding value of z(blue) is obtained by above equation as **z=1-(x+y)**

RGB color model:

- The purpose of a color model (also called **color space** or **color system**) is to facilitate the specification of colors in some standard.
- In essence, a color model is a specification of a coordinate system and a subspace within that system where each color is represented by a single point.
- RGB Model for color monitors and color video cameras.
- CMY and CMYK models for color printing.
- HIS model used by humans to describe and interpret colors of image and also used to decouple the color and grayscale information in an image.

The RGB Color Model:

- In the RGB model, each color appears in its primary spectral components of red, green, and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube shown in Fig.4, in which RGB values are at three corners; cyan, magenta, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. In this model, the gray scale (points of equal RGB values) extends from black to white along the line joining these two points. The different colors in this model arc points on or inside the cube, and are defined by vectors extending from the origin. For convenience, the assumption is that all color values have been normalized so that the cube shown in Fig.4 is the unit cube. That is, all values of R, G. and B are assumed to be in the range [0, 1].

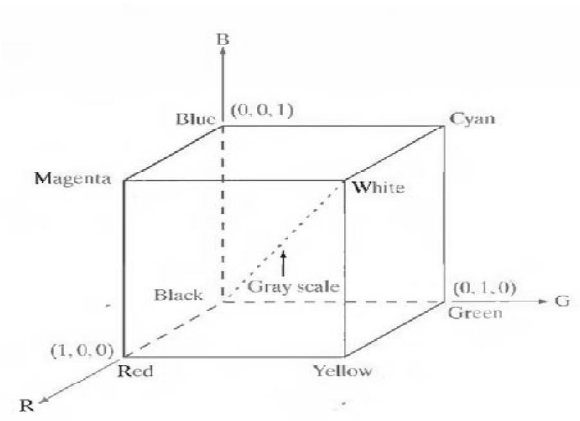


Fig.4 Schematic of the RGB color cube

- Images represented in the RGB color model consist of three component images, one for each primary color. When fed into an RGB monitor, these three images combine on the phosphor screen to produce a composite color image. The number of bits used to represent each pixel in RGB space is called the pixel depth.
- Consider an RGB image in which each of the red, green, and blue images is an 8-bit image. Under these conditions each RGB color pixel [that is, a triplet of values (R, G, B)] is said to have a depth of 24 bits C image planes times the number of bits per plane). The term full-color image is used often to denote a 24-bit RGB color image. The total number of colors in a 24-bit RGB image is $(2^8)^3 = 16,777,216$.
- RGB is ideal for image color generation (as in image capture by a color camera or image display in a monitor screen), but its use for color description is much more limited.
- Even though each color is represented with 256 different intensities depending on viewer hardware capabilities there are subset (216) of intensities which are viewed. These subsets of intensities are called **safe RGB colors** or ***all-system-safe colors***.
- Now the RGB triplets of these safe RGB colors is given as $(2^6)^3$

CMY color model:

- Cyan, magenta, and yellow are the secondary colors of light or, alternatively, the primary colors of pigments. For example, when a surface coated with cyan pigment is illuminated with white light, no red light is reflected from the surface. That is, cyan subtracts red light from reflected white light, which itself is composed of equal amounts of red, green, and blue light

- Most devices that deposit colored pigments on paper, such as color printers and copiers, require CMY data input or perform an RGB to CMY conversion internally. This conversion is performed using the simple operation (1) where, again, the assumption is that all color values have been normalized to the range [0, 1]. Equation (1) demonstrates that light reflected from a surface coated with pure cyan does not contain red (that is, C = 1 — R in the equation).

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{--------}\rightarrow 1$$

- Similarly, pure magenta does not reflect green, and pure yellow does not reflect blue. Equation (1) also reveals that RGB values can be obtained easily from a set of CMY values by subtracting the individual CMY values from 1. As indicated earlier, in image processing this color model is used in connection with generating hardcopy output, so the inverse operation from CMY to RGB generally is of little practical interest.
- Equal amounts of the pigment primaries, cyan, magenta, and yellow should produce black. In practice, combining these colors for printing produces a **muddy-looking black**.
- To produce true black (predominant colour in printing) a 4th colour, black, is added => CMYK model (CMY + Black)

HSI color model:

- When humans view a color object, we describe it by its hue, saturation, and brightness. Hue is a color attribute that describes a pure color (pure yellow, orange, or red), whereas saturation gives a measure of the degree to which a pure color is diluted by white light. Brightness is a subjective descriptor that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.
- Intensity (gray level) is a most useful descriptor of monochromatic images. This quantity definitely is measurable and easily interpretable. The HSI (hue, saturation, intensity) color model, decouples the intensity component from the color-carrying information (hue and saturation) in a color image. As a result, the HSI model is an ideal tool for developing image processing algorithms based on color descriptions that are natural and intuitive to humans.

- In Fig 5 the primary colors are separated by 120°. The secondary colors are 60° from the primaries, which means that the angle between secondaries is also 120°. Figure 5(b) shows the same hexagonal shape and an arbitrary color point (shown as a dot).The hue of the point is determined by an angle from some reference point. Usually (but not always) an angle of 0° from the red axis designates 0 hue, and the hue increases counterclockwise from there. The saturation (distance from the vertical axis) is the length of the vector from the origin to the point. Note that the origin is defined by the intersection of the color plane with the vertical intensity axis. The important components of the HSI color space are the vertical intensity axis, the length of the vector to a color point, and the angle this vector makes with the red axis.

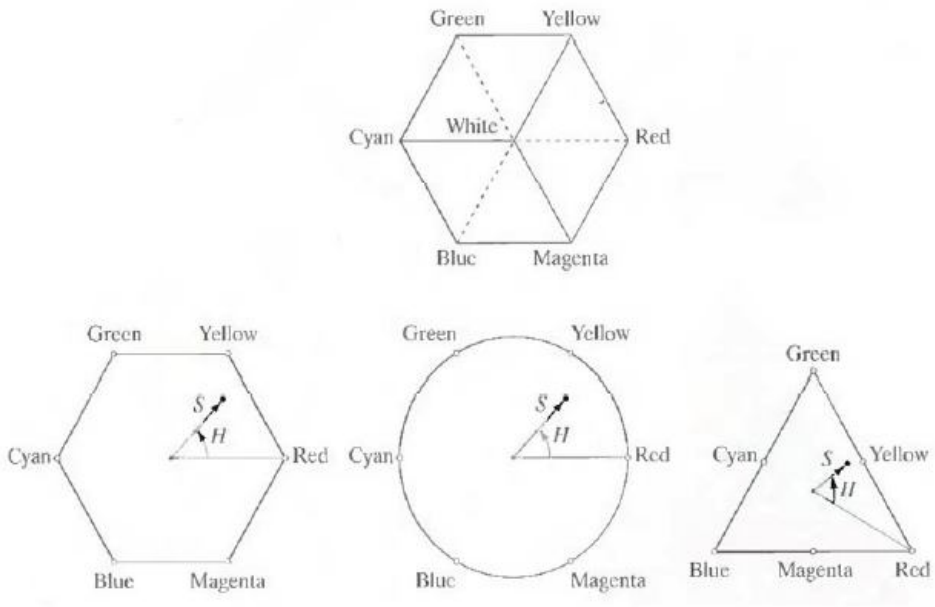


Fig 5 Hue and saturation in the HSI color model.

Conversion from RGB color model to HSI color model:

- Given an image in RGB color format, the H component of each RGB pixel is obtained using the equation

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases} \text{ -------}\rightarrow 1$$

- With

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

- The saturation component is given by

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)]. \quad \text{----------} \rightarrow 3$$

- Finally, the intensity component is given by

$$I = \frac{1}{3}(R + G + B). \quad \text{----------} \rightarrow 4$$

- It is assumed that the RGB values have been normalized to the range [0, 1] and that angle θ is measured with respect to the red axis of the HST space. Hue can be normalized to the range [0, 1] by dividing by 360° all values resulting from Eq. (1). The other two HSI components already are in this range if the given RGB values are in the interval [0, 1].

Conversion from HSI color model to RGB color model:

- Given values of HSI in the interval [0, 1], one can find the corresponding RGB values in the same range. The applicable equations depend on the values of H. There are three sectors of interest, corresponding to the 120° intervals in the separation of primaries.

RG sector ($0^o \leq H < 120°$):

- When H is in this sector, the RGB components are given by the equations

$$\mathbf{B = I (1 – S)}$$

$$\mathbf{G = 3 I – (R + B)}$$

$$\mathbf{R = I [1 + (S * \cos H / \cos(60^o – H)]}$$

GB sector ($120^o \leq H < 240^o$):

If the given value of H is in this sector, first subtract 120° from it.

$$\mathbf{H = H - 120^0}$$

Then the RGB components are

$$\mathbf{R = I (1 – S)}$$

$$\mathbf{B = 3 I – (R + G)}$$

$$\mathbf{G = I [1 + (S * \cos H / \cos(60^o – H)]}$$

BR sector ($240^o \leq H \leq 360^o$):

If H is in this range, subtract 240^o from it

$$\mathbf{H = H - 240^0}$$

Then the RGB components are

$$\mathbf{G = I (1 – S)}$$

$$R = 3I - (B + G)$$

$$B = I [1 + (S * \cos H / \cos(60^o - H)]$$

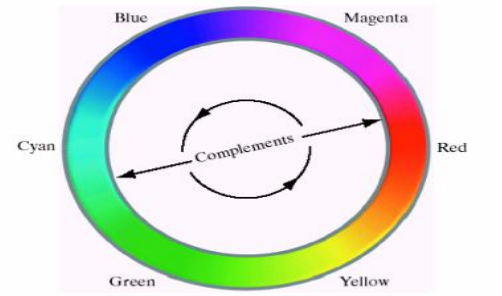
Color transformations:

Formulation:

- As with the gray-level transformation techniques we model color transformations using the expression g(x,y)=T[f(x,y)].
- Where f(x,y) is a color input image. G(x,y) is transformed image. T is an operation on f over a spatial neighborhood of (x,y).
- The principle difference between color image trans formation and gray-level image trans formation is that in color image transformation the pixel value is **triplets or quartets**
- $\{T_1,T_2,T_3,\ldots,T_n\}$ is a set of transformation or color mapping functions that operate on r_i to produce S_i .
- If the RGB color space is selected, for example, n=3 and r_1,r_2, and r_3 denote the red, green, and blue components of the input image.
- Suppose, for example, that we wish to modify the intensity of the image using $g(x, y) = kf(x, y)$
- In the HIS color space, this can be done with the simple transformation
  $s_3 = kr_3$ where $s_1 = r_1$ and $s_2 = r_2$. Only HSI intensity component r_3 is modified
- In the RGB color space, three components must be transformed:
  $s_i = kr_i$ for i =1,2,3
- The CMY space requires a similar set of linear transformations:
  $s_i = kr_i + (1- k)$ for i =1,2,3

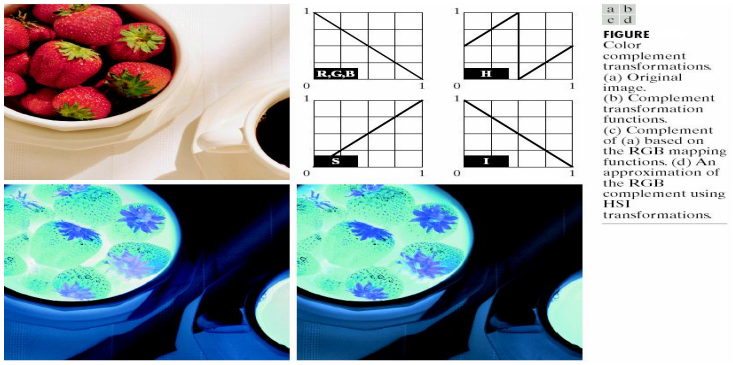
Color complements

- The Hues directly opposite one another on the color circle are called the complements as shown in the following figure



Our interest in complements stems from the fact that they are Analogous to gray scale negatives.

As the gray scale case color complements are used to enhance details buried in dark regions of a color image



a b
c d
FIGURE Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.

- The RGB compliment transformation function used in this may not have a straight forward HIS space equivalent.
- Saturation component of the compliment cannot be computed from the saturation component of the input image alone.
- In the above Figures C and D the saturation component is unaltered

Color slicing

- Color slicing is used to highlight a specific range of colors in an image to separate objects from surroundings.
- The basic idea to this is:
  1. Display the colors of interest so that they stand out from the background. or
  2. use the regions defined by specified colors as a mask for further processing
- – since a color pixel is n-dimensional quantity color transformation functions are more complex than gray-level slicing.
- One simple way to slice the color image is to map the color outside some range of interest to a non prominent natural color.
- Using a cube of width W to enclose the reference color with components (a1,a2,……..,an) the transformation is given by

$$s_i = \begin{cases} 0.5 & \text{if } \left[|r_j - a_j| > \frac{W}{2}\right]_{\text{any } 1 \le j \le n} \\ r_i & \text{otherwise} \end{cases} \quad i = 1, 2, \ldots, n$$

- These transformations highlight the colors around the prototype by forcing all other colors to the midpoint of the reference color space.
- For RGB color space, for example, a suitable neutral point is middle gray or color (0.5, 0.5, 0.5).
- If the color of interest is specified by a sphere of radius R0, the ransformation is

$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^{n}(r_j - a_j)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases} \quad i = 1, 2, \ldots, n$$

- Here R_0 is the radious of the enclosing sphere and (a_1,a_2,a_3,…..a_n) are components of its center.

Tone and color corrections

- This is used for photo enhancement and color reproduction
- This is Device independent color model from CIE relating the color gamuts
- It use a color profile to map each device to color model
- CIE L*a*b* system
  - CIE systm is most common model for color management systems
  - For CIE system Components given by the following equations

$$\begin{aligned} L* &= 116 \cdot h\left(\frac{Y}{Y_W}\right) - 16 \\ a* &= 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right] \\ b* &= 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right] \end{aligned}$$

Where

$$h(q) = \begin{cases} q^{\frac{1}{3}} & \text{if } q > 0.008856 \\ 7.787q + \frac{16}{116} & \text{otherwise} \end{cases}$$

- X_W, Y_W, and Z_W are values for refence white, called D65 which is defined by x = 0:3127 and y = 0:3290 in the CIE chromaticity diagram
- X, Y, Z are computed from rgb values as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix}$$

- Rec. 709 RGB corresponds to D65 white point

- L*a*b* is calorimetric (colors perceived as matching are encoded identically), perceptually uniform (color differences among various hues are perceived uniformly), and device independent
- Not directly displayable on any device but its gamut covers the entire visible spectrum
- L*a*b* decouples intensity from color, making it useful for image manipulation (hue and contrast editing) and image compression applications
    - L* represents lightness or intensity
    - a* gives red minus green
    - b* gives green minus blue
- Allows tonal and color imbalances to be corrected interactively and independently
    - Tonal range refers to general distribution of key intensities in an image
        - Adjust image brightness and contrast to provide maximum detail over a range of intensities
        - The colors themselves are not changed

Histogram processing

- This provides an automated way to perform enhancement
- This is simmilar to the grayscale image Histogram equalization
- Adapt the grayscale technique to multiple components
- Applying grayscale techniques to different colors independently yields erroneous colors
- Spread the intensities uniformly leaving the hues unchanged.

Smoothing and sharpening:

Color image smoothing

- Extend spatial filtering mask to color smoothing, dealing with component vectors
- Let S_{xy} be the neighborhood centered at (x, y)
- Average of RGB components in the neighborhood is given by

$$\bar{\mathbf{c}}(x,y) = \frac{1}{K} \sum_{(s,t) \in S_{xy}} \mathbf{c}(s,t)$$

Which is the same as

$$\bar{\mathbf{c}}(x,y) = \begin{bmatrix} \frac{1}{K}\sum_{(s,t)\in S_{xy}} R(s,t) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} G(s,t) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} B(s,t) \end{bmatrix}$$

Color image sharpening:

- Image sharpening is done using the Laplacian. For vector analysis we know that the Laplacian of a vector is defined as a vector whose components are equal to the laplacian of the individual scalar components of the input vector.
- In RGB color system the Laplation of vector e is as follows

$$\nabla^2[\mathbf{c}(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix}$$

- It tells us that we can compute the Laplation of full color image by computing the Laplacian ofeach component image separately.

Pseudo color image processing:

- Pseudo color (also called false color) image processing consists of assigning colors to gray values based on a specified criterion.
- The term pseudo or false color is used to differentiate the process of assigning colors to monochrome images from the processes associated with true color images.
- The process of gray level to color transformations is known as pseudo color image processing.
- The two techniques used for pseudo color image processing are,
  1. Intensity Slicing
  2. Gray Level to Color Transformation

Intensity Slicing:

- The technique of intensity (sometimes called density) slicing and color coding is one of the simplest examples of pseudo color image processing. If an image is interpreted as a 3-D function (intensity versus spatial coordinates), 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. Figure 6 shows an example of using 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.6, any pixel whose gray level is above the plane will be coded with one color, and any pixel below the plane will be coded with the other. Levels that lie on the plane itself may be arbitrarily assigned one of the two colors. 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.
- In general, the technique may be summarized as follows. Let [0, L - 1] represent the gray scale, let level lo represent black [f(x, y) = 0], and level l_{L-1} represent white [f(x, y) = L - 1]. Suppose that P planes perpendicular to the intensity axis are defined at levels l_1, l_2,…,l_p.. Then, assuming that 0 < P < L – 1, the P planes partition the gray scale into P + 1 intervals, V_1, V_2,...,V_p+ 1. Gray-level to color assignments are made according to the relation $f(x, y) = c_k$ if $f(x, y) \in V_k$

  Where c_k is the color associated with the kth intensity interval V_k defined by the partitioning planes at l = k - 1 and l = k.

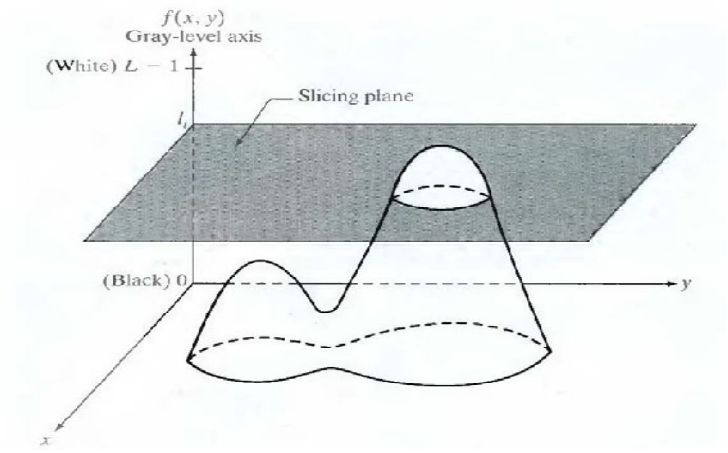


Fig.6 Geometric interpretation of the intensity-slicing technique.

- The idea of planes is useful primarily for a geometric interpretation of the intensity slicing technique. Figure7 shows an alternative representation that defines the same mapping as in Fig.6. According to the mapping function shown in Fig.7, any input gray level is assigned one of two colors, depending on whether it is above or below the value of l_i when more levels are used, the mapping function takes on a staircase form.

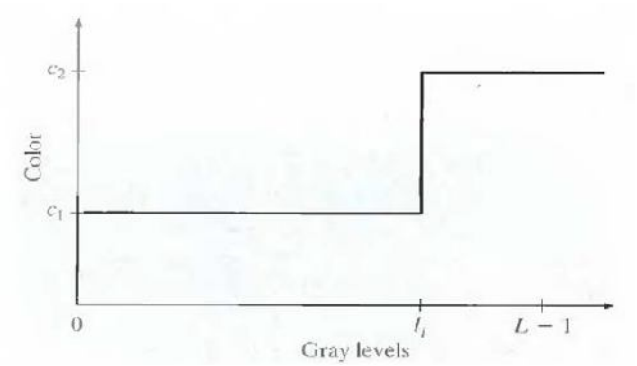


Fig.7 An alternative representation of the intensity-slicing technique.

Gray Level to Color Transformation:

- 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 transformation functions. Note that these are transformations on the gray-level values of an image and are not functions of position.
- In intensity slicing, piecewise linear functions of the gray levels are used to generate colors. On the other hand, this method can be based on smooth, nonlinear functions, which, as might be expected, gives the technique considerable flexibility.

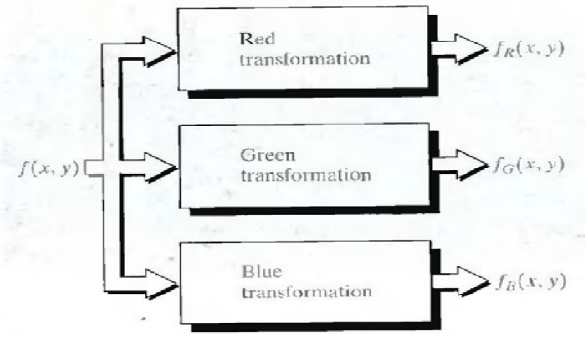


Fig. 8 Functional block diagram for pseudocolor image processing.

- The output of each transformation is a composite image.

Basics of full color image processing:

- Full-color image processing approaches fall into two major categories. In the first category, each component image is processed individually and then form a composite processed color image from the individually processed components. In the second

category, one works with color pixels directly. Because full-color images have at least three components, color pixels really are vectors. For example, in the RGB system, each color point can be interpreted as a vector extending from the origin to that point in the RGB coordinate system.

- Let c represent an arbitrary vector in RGB color space:

$$\mathbf{c} = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad \text{------}\rightarrow 1$$

- This equation indicates that the components of c are simply the RGB components of a color image at a point. If the color components are a function of coordinates (x, y) by using the notation

$$\mathbf{c}(x, y) = \begin{bmatrix} c_R(x, y) \\ c_G(x, y) \\ c_B(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix}. \quad \text{----}\rightarrow 2$$

- For an image of size M X N, there are MN such vectors, c(x, y), for x = 0,1, 2,...,M- l; y = 0,1,2,...,N- 1.
- It is important to keep clearly in mind that Eq. (2) depicts a vector whose components are spatial variables in x and y.
- In order for per-color-component and vector-based processing to be equivalent, two conditions have to be satisfied: First, the process has to be applicable to both vectors and scalars. Second, the operation on each component of a vector must be independent of the other components.

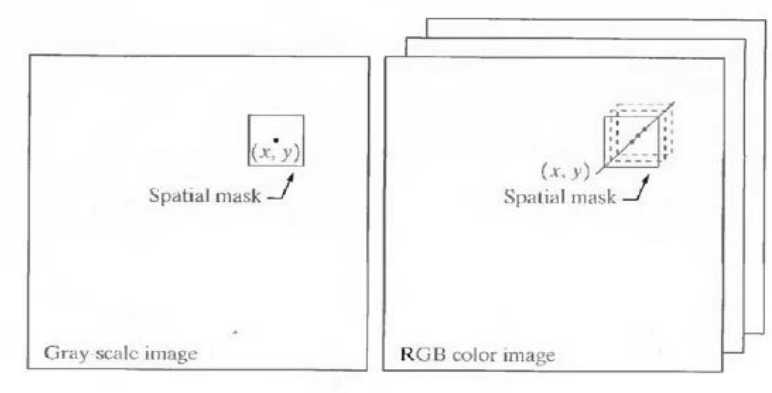


Fig 9 Spatial masks for gray-scale and RGB color images.

- Fig 9 shows neighborhood spatial processing of gray-scale and full-color images. Suppose that the process is neighborhood averaging. In Fig. 9(a), averaging would be

accomplished by summing the gray levels of all the pixels in the neighborhood and dividing by the total number of pixels in the neighborhood. In Fig. 9(b), averaging would be done by summing all the vectors in the neighborhood and dividing each component by the total number of vectors in the neighborhood. But each component of the average vector is the sum of the pixels in the image corresponding to that component, which is the same as the result that would be obtained if the averaging were done on a per-color-component basis and then the vector was formed.

Color segmentation process:

- Segmentation is a process that partitions an image into regions.
- Partitioning an image into regions based on color is known as color segmentation.

Segmentation in HSI Color Space:

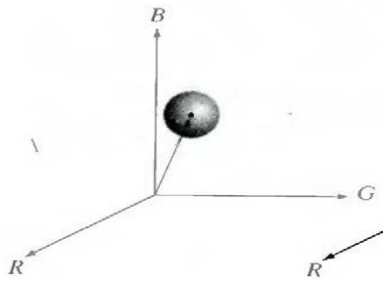
- If anybody want to segment an image based on color, and in addition, to carry out the process on individual planes, it is natural to think first of the HSI space because color is conveniently represented in the hue image.
- Typically, saturation is used as a masking image in order to isolate further regions of interest in the hue image.
- The intensity image is used less frequently for segmentation of color images because it carries no color information.

Segmentation in RGB Vector Space:

- Although, working in HSI space is more intuitive, segmentation is one area in which better results generally are obtained by using RGB color vectors.
- Suppose that the objective is to segment objects of a specified color range in an RGB image. Given a set of sample color points representative of the colors of interest, we obtain an estimate of the "average" color that we wish to segment. Let this average color be denoted by the RGB vector a.
- The objective of segmentation is to classify each RGB pixel in a given image as having a color in the specified range or not. In order to perform this comparison, it is necessary to have a measure of similarity. One of the simplest measures is the Euclidean distance. Let z denote an arbitrary point in RGB space. z is similar to a if the distance between them is less than a specified threshold, D_o. The Euclidean distance between z and a is given by

$$D(\mathbf{z}, \mathbf{a}) = \|\mathbf{z} - \mathbf{a}\|$$
$$= [(\mathbf{z} - \mathbf{a})^T(\mathbf{z} - \mathbf{a})]^{\frac{1}{2}}$$
$$= [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{\frac{1}{2}}$$

- where the subscripts R, G, and B, denote the RGB components of vectors a and z .The locus of points such that $D(z, a) \leq D_o$ is a solid sphere of radius D_o.



- Points contained within or on the surface of the sphere satisfy the specified color criterion; points outside the sphere do not. Coding these two sets of points in the image with, say, black and white, produces a binary segmented image.
- A useful generalization of previous equation is a distance measure of the form $D(z, a) = [(z-a)^T C^{-1} (z-a)]^{½}$
- Where C is the covariance matrix1 of the samples representative of the color to be segmented.
- The above equation represents an ellipse with color points such that $D(z, a) \leq D_o$.

UNIT-III
Assignment-Cum-Tutorial Questions

SECTION-A

Objective Questions

1. color spectrum is divided into _____broad regions [ ]
A,3 B,6 C,9 D,12
2. If the light is achromatic , its only attribute is its intensity and it is used in Black and white TV light [True/False]
3. _____ is the total amount of energy that flows from the light source, and it is usually measured in _____.
4. The amount of energy an observer perceives from a light source is called [ ]
A, Radiance B, Luminance C, Brightness D, Both A and B
5. Primary colors are [ ]
A, RGB B, CMY C, CMYK D, None
6. Secondary colors are **[ ]**
A, RGB B, CMY C, Both A and B D, None
7. The characteristics generally used to distinguish one color from another are [ ]
A, Brightness B, Hue C, Saturation D, All
8. Hue and saturation taken together are called _____
9. Equal amounts of the pigment primaries cyan, magenta, and yellow produce muddy-black [True/False]
10. Black level is represented by formula [ ]
A, $[f(x) = 0]$ B,$[f(y) = 0]$ C,$[f(x,y) = 0]$ D,$[f(x,y) = 1]$
11. White color in a Cartesian coordinate system can be represented as
A, (0,1,1) B, (0,1,0) C,(0,0,1) D,(1,1,1) [ ]

12. The total number of colors in a 24-bit RGB image is [ ]
A, 2^8 B, 2^3 C, $(2^8)^3$ D, None
13. _____ is the normalized representation of intensity 255, for intensity levels 0 to 255 [ ]
A, 1 B, 0 C, 0.5 D, none
14. If normalized RGB image intensities are R=G=B=1, after converting the RGB image to HSI format what is I value. [ ]
A, 1 B, 0 C,0.5 D, not determined
15. Grouping of two hex numbers forms an 8-bit byte. [True/False]
16. The bright read in decimal notation has R=255 and G=B=1 [True/ False]
17. Represent Bright green in Hex number system [ ]
A, (00FF00) B, (11FF11) C,(11FF00) D, (00FF11)
18. If normalized RGB image intensities are R=G=B=0, after converting the RGB image to HSI format what is S value. [ ]
A, 1 B, 0 C,0.5 D, 2
19. $(FFFFFF)_{16}$ is represent _____color in RGB format [ ]
A, White B, Black C, red D, both A and B

20. For any value of x(red)=.3 and y(green)=.6 the corresponding value of z(blue) is _____ [ ]

A, 0.9 B, 0 C, 1 D, 0.1

21. If H Component of RGB Pixel is obtained as θ By_____ [ ]

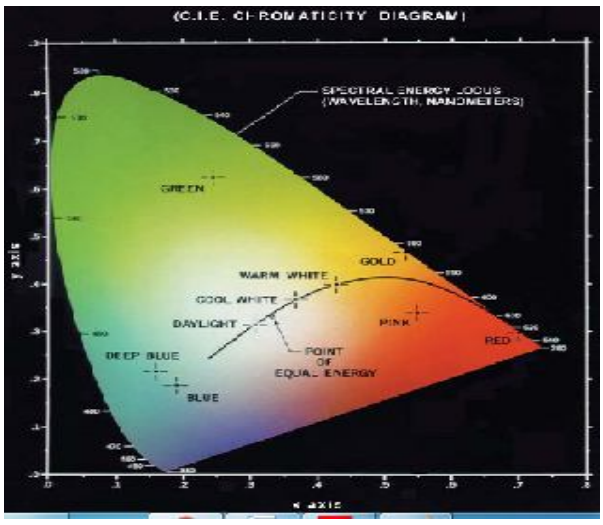
A, H=θ if B<=G B, H=360-θ if B>G

C, H=θ if B>=G D, Both A and B

SECTION-B

SUBJECTIVE QUESTIONS

1. Explain about color fundamentals.
2. Illustrate RGB color model.
3. Explain CMY color model.
4. Explain HSI color model.
5. Explain about pseudocolor image processing.
6. Summarize color segmentation process.
7. Outline the basics of full color image processing.
8. Demonstrate image smoothing and image sharpening Summarize color transformations
9. Develop procedure for conversion from HSI color model to RGB color model.
10. Construct procedure for conversion from RGB color model to HSI color model.
11. Identify the percentage of red(X), green (G), and blue (Z) light required to generate the point labeled “Warm white” in below figure.



UNIT - IV

Image Compression

Fundamentals, image compression models, error-free compression, lossy predictive coding.

Image Compression Fundamentals:

- The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information.
- A clear distinction must be made between data and information. They are not synonymous. In fact, data are the means by which information is conveyed. Various amounts of data may be used to represent the same amount of information.
- Such might be the case, for example, if a long-winded individual and someone who is short and to the point were to relate the same story. Here, the information of interest is the story; words are the data used to relate the information. If the two individuals use a different number of words to tell the same basic story, two different versions of the story are created, and at least one includes nonessential data. That is, it contains data (or words) that either provide no relevant information or simply restate that which is already known. It is thus said to contain data redundancy.
- Data redundancy is a central issue in digital image compression. It is not an abstract concept but a mathematically quantifiable entity. If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy R_D of the first data set (the one characterized by n_1) can be defined as

$$R_D = 1 - \frac{1}{C_R}$$

- Where C_R , commonly called the compression ratio, is

$$C_R = \frac{n_1}{n_2}.$$

- For the case $n_2 = n_1$, $C_R = 1$ and $R_D = 0$, indicating that (relative to the second data set) the first representation of the information contains no redundant data.
- When $n_2 << n_1$, $C_R \rightarrow \infty$ and $R_D \rightarrow 1$, implying significant compression and highly redundant data.
- Finally, when $n_2 >> n_1$, $C_R \rightarrow 0$ and $R_D \rightarrow \infty$, indicating that the second data set contains much more data than the original representation. This, of course, is the normally undesirable case of data expansion. In general, C_R and R_D lie in the open intervals (0, ∞) and (-∞, 1), respectively. A practical compression ratio, such as 10 (or 10:1), means that the first data set has 10 information carrying units (say, bits) for every 1 unit in the second or compressed data set. The corresponding redundancy of 0.9 implies that 90% of the data in the first data set is redundant.

- In digital image compression, three basic data redundancies can be identified and exploited: **coding redundancy**, **interpixel redundancy**, and **psychovisual redundancy**. Data compression is achieved when one or more of these redundancies are reduced or eliminated.

Coding Redundancy:

- In this, we utilize formulation to show how the gray-level histogram of an image also can provide a great deal of insight into the construction of codes to reduce the amount of data used to represent it.
- Let us assume, once again, that a discrete random variable r_k in the interval [0, 1] represents the gray levels of an image and that each r_k occurs with probability p_r (r_k).

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots, L-1$$

- Where L is the number of gray levels, n_k is the number of times that the k^{th} gray level appears in the image, and n is the total number of pixels in the image. If the number of bits used to represent each value of r_k is l (r_k), then the average number of bits required to represent each pixel is

$$L_{\text{avg}} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k).$$

- That is, the average length of the code words assigned to the various gray-level values is found by summing the product of the number of bits used to represent each gray level and the probability that the gray level occurs. Thus the total number of bits required to code an M X N image is MNL_{avg}.

Interpixel Redundancy:

- Consider the images shown in Figs. 1.1(a) and (b). As Figs. 1.1(c) and (d) show, these images have virtually identical histograms. Note also that both histograms are trimodal, indicating the presence of three dominant ranges of gray-level values. Because the gray levels in these images are not equally probable, variable-length coding can be used to reduce the coding redundancy that would result from a straight or natural binary encoding of their pixels.
- The coding process, however, would not alter the level of correlation between the pixels within the images. In other words, the codes used to represent the gray levels of each image have nothing to do with the correlation between pixels. These correlations result from the structural or geometric relationships between the objects in the image.

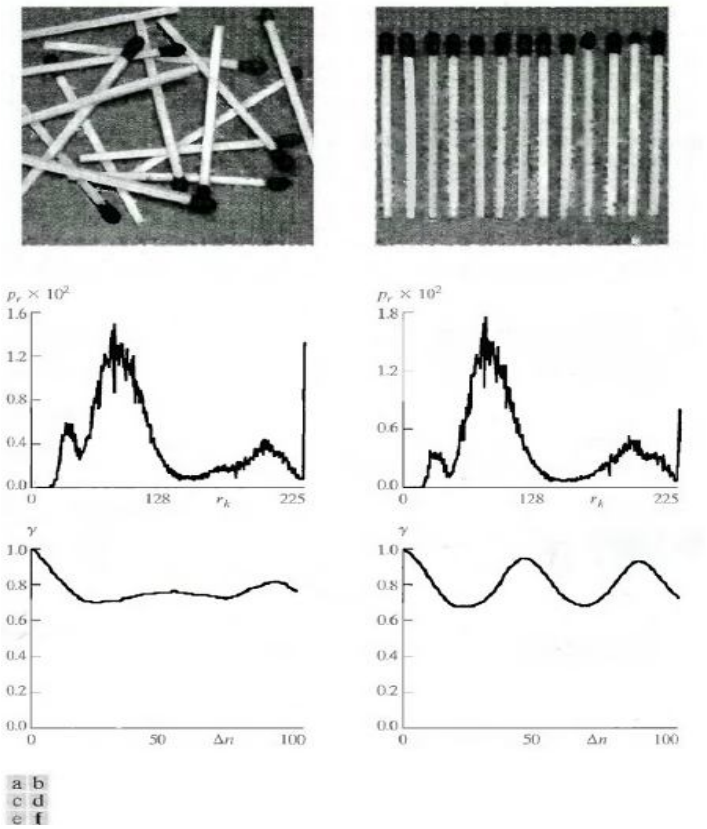


Fig.1.1 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

- Figures 1.1(e) and (f) show the respective autocorrelation coefficients computed along one line of each image.

$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)}$$

Where

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) f(x, y + \Delta n).$$

- The scaling factor in Eq. above accounts for the varying number of sum terms that arise for each integer value of Δn. Of course, Δn must be strictly less than N, the number of pixels on a line. The variable x is the coordinate of the line used in the computation. Note the dramatic difference between the shape of the functions shown in Figs. 1.1(e) and (f). Their shapes can be qualitatively related to the structure in the images in Figs. 1.1(a) and (b).This relationship is particularly noticeable in Fig. 1.1 (f), where the high correlation between pixels separated by 45 and 90 samples can be directly related to the spacing between the vertically oriented matches of Fig. 1.1(b). In addition, the adjacent pixels of both images are highly correlated. When Δn is 1, γ is 0.9922 and 0.9928 for the images of Figs. 1.1 (a) and (b), respectively. These values are typical of most properly sampled television images.
- These illustrations reflect another important form of data redundancy—one directly related to the interpixel correlations within an image. Because the value of any given pixel can be

reasonably predicted from the value of its neighbors, the information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the basis of the values of its neighbors. A variety of names, including spatial redundancy, geometric redundancy, and interframe redundancy, have been coined to refer to these interpixel dependencies. We use the term interpixel redundancy to encompass them all.

- In order to reduce the interpixel redundancies in an image, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient (but usually "nonvisual") format. For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type (that is, those that remove interpixel redundancy) are referred to as mappings. They are called reversible mappings if the original image elements can be reconstructed from the transformed data set.

Psychovisual Redundancy:

- The brightness of a region, as perceived by the eye, depends on factors other than simply the light reflected by the region. For example, intensity variations (Mach bands) can be perceived in an area of constant intensity. Such phenomena result from the fact that the eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processing. This information is said to be psychovisually redundant. It can be eliminated without significantly impairing the quality of image perception.
- That psychovisual redundancies exist should not come as a surprise, because human perception of the information in an image normally does not involve quantitative analysis of every pixel value in the image. In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings. The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process. Psychovisual redundancy is fundamentally different from the redundancies discussed earlier. Unlike coding and interpixel redundancy, psychovisual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psychovisually redundant data results in a loss of quantitative information, it is commonly referred to as quantization.
- This terminology is consistent with normal usage of the word, which generally means the mapping of a broad range of input values to a limited number of output values. As it is an irreversible operation (visual information is lost), quantization results in lossy data compression.

Fidelity criterion:

- The removal of psychovisually redundant data results in a loss of real or quantitative visual information. Because information of interest may be lost, a repeatable or reproducible means of quantifying the nature and extent of information loss is highly desirable. Two general classes of criteria are used as the basis for such an assessment:

    A. Objective fidelity criteria and

B. Subjective fidelity criteria.

- When the level of information loss can be expressed as a function of the original or input image and the compressed and subsequently decompressed output image, it is said to be based on an objective fidelity criterion. A good example is the root-mean-square (rms) error between an input and output image. Let f(x, y) represent an input image and let f(x, y) denote an estimate or approximation of f(x, y) that results from compressing and subsequently decompressing the input. For any value of x and y, the error e(x, y) between f (x, y) and f^ (x, y) can be defined as

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

so that the total error between the two images is

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

- Where the images are of size M X N. The root-mean-square error, e_{rms}, between f(x, y) and f^(x,y) then is the square root of the squared error averaged over the M X N array, or

$$e_{rms} = \left[\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x, y) - f(x, y)]^2\right]^{1/2}$$

- A closely related objective fidelity criterion is the mean-square signal-to-noise ratio of the compressed-decompressed image. If f^ (x, y) is considered to be the sum of the original image f(x, y) and a noise signal e(x, y), the mean-square signal-to-noise ratio of the output image, denoted SNR_{rms}, is

$$SNR_{rms} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x, y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x, y) - f(x, y)]^2}.$$

- The rms value of the signal-to-noise ratio, denoted SNR_{rms}, is obtained by taking the square root of Eq. above.
- Although objective fidelity criteria offer a simple and convenient mechanism for evaluating information loss, most decompressed images ultimately are viewed by humans. Consequently, measuring image quality by the subjective evaluations of a human observer often is more appropriate. This can be accomplished by showing a "typical" decompressed image to an appropriate cross section of viewers and averaging their evaluations. The evaluations may be made using an absolute rating scale or by means of side-by-side comparisons of f(x, y) and f^(x,y).

Image compression models:

- Fig. 3.1 shows, a compression system consists of two distinct structural blocks: an encoder and a decoder. An input image f(x, y) is fed into the encoder, which creates a set of symbols from the input data. After transmission over the channel, the encoded representation is fed to the decoder, where a reconstructed output image f^(x, y) is generated. In general, f^(x, y) may or may not be an exact replica of f(x, y). If it is, the system is error free or information

preserving; if not, some level of distortion is present in the reconstructed image. Both the encoder and decoder shown in Fig. 3.1 consist of two relatively independent functions or sub blocks. The encoder is made up of a source encoder, which removes input redundancies, and a channel encoder, which increases the noise immunity of the source encoder's output. As would be expected, the decoder includes a channel decoder followed by a source decoder. If the channel between the encoder and decoder is noise free (not prone to error), the channel encoder and decoder are omitted, and the general encoder and decoder become the source encoder and decoder, respectively.

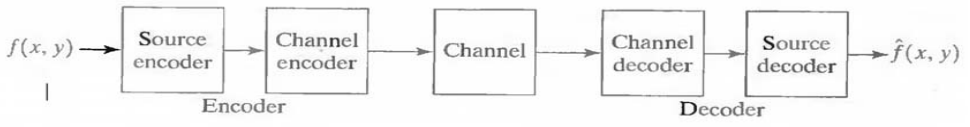


Fig.3.1 A general compression system model

The Source Encoder and Decoder:

- The source encoder is responsible for reducing or eliminating any coding, interpixel, or psychovisual redundancies in the input image. The specific application and associated fidelity requirements dictate the best encoding approach to use in any given situation. Normally, the approach can be modeled by a series of three independent operations. As Fig. 3.2 (a) shows, each operation is designed to reduce one of the three redundancies. Figure 3.2 (b) depicts the corresponding source decoder. In the first stage of the source encoding process, the mapper transforms the input data into a (usually nonvisual) format designed to reduce interpixel redundancies in the input image. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image.

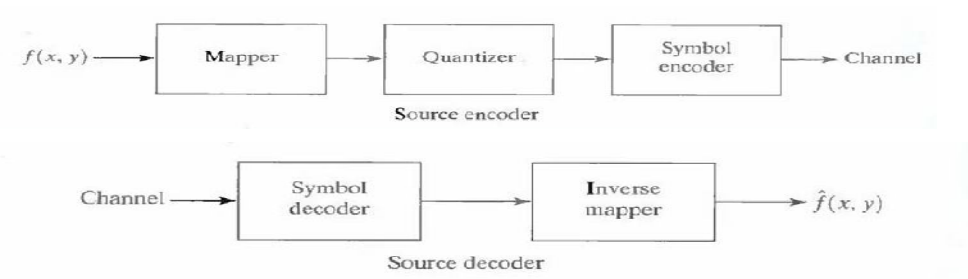


Fig.3.2 (a) Source encoder and (b) source decoder model

- Run-length coding is an example of a mapping that directly results in data compression in this initial stage of the overall source encoding process. The representation of an image by a set of transform coefficients is an example of the opposite case. Here, the mapper transforms the image into an array of coefficients, making its interpixel redundancies more accessible for compression in later stages of the encoding process.
- The second stage, or quantizer block in Fig. 3.2 (a), reduces the accuracy of the mapper's output in accordance with some preestablished fidelity criterion. This stage reduces the psychovisual redundancies of the input image. This operation is irreversible. Thus it must be omitted when error-free compression is desired.
- In the third and final stage of the source encoding process, the symbol coder creates a fixed- or variable-length code to represent the quantizer output and maps the output in accordance

with the code. The term symbol coder distinguishes this coding operation from the overall source encoding process. In most cases, a variable-length code is used to represent the mapped and quantized data set. It assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancy. The operation, of course, is reversible. Upon completion of the symbol coding step, the input image has been processed to remove each of the three redundancies.

- Figure 3.2(a) shows the source encoding process as three successive operations, but all three operations are not necessarily included in every compression system. Recall, for example, that the quantizer must be omitted when error-free compression is desired. In addition, some compression techniques normally are modeled by merging blocks that are physically separate in Fig. 3.2(a). In the predictive compression systems, for instance, the mapper and quantizer are often represented by a single block, which simultaneously performs both operations.
- The source decoder shown in Fig. 3.2(b) contains only two components: a symbol decoder and an inverse mapper. These blocks perform, in reverse order, the inverse operations of the source encoder's symbol encoder and mapper blocks. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general source decoder model shown in Fig. 3.2(b).

The Channel Encoder and Decoder:

- The channel encoder and decoder play an important role in the overall encoding-decoding process when the channel of Fig. 3.1 is noisy or prone to error. They are designed to reduce the impact of channel noise by inserting a controlled form of redundancy into the source encoded data. As the output of the source encoder contains little redundancy, it would be highly sensitive to transmission noise without the addition of this "controlled redundancy." One of the most useful channel encoding techniques was devised by R. W. Hamming (Hamming [1950]). It is based on appending enough bits to the data being encoded to ensure that some minimum number of bits must change between valid code words. Hamming showed, for example, that if 3 bits of redundancy are added to a 4-bit word, so that the distance between any two valid code words is 3, all single-bit errors can be detected and corrected. (By appending additional bits of redundancy, multiple-bit errors can be detected and corrected.) The 7-bit Hamming (7, 4) code word $h_1, h_2, h_3 \ldots, h_6, h_7$ associated with a 4-bit binary number $b_3b_2b_1b_0$ is

$$
\begin{aligned}
h_1 &= b_3 \oplus b_2 \oplus b_0 \qquad & h_3 &= b_3 \\
h_2 &= b_3 \oplus b_1 \oplus b_0 \qquad & h_5 &= b_2 \\
h_4 &= b_2 \oplus b_1 \oplus b_0 \qquad & h_6 &= b_1 \\
& & h_7 &= b_0
\end{aligned}
$$

- Where $\oplus$ denotes the exclusive OR operation. Note that bits h_1, h_2, and h_4 are even-parity bits for the bit fields $b_3\ b_2\ b_0$, $b_3b_1b_0$, and $b_2b_1b_0$, respectively. (Recall that a string of binary bits has even parity if the number of bits with a value of 1 is even.) To decode a Hamming encoded result, the channel decoder must check the encoded value for odd

parity over the bit fields in which even parity was previously established. A single-bit error is indicated by a nonzero parity word $c_4c_2c_1$, where

$$c_1 = h_1 \oplus h_3 \oplus h_5 \oplus h_7$$
$$c_2 = h_2 \oplus h_3 \oplus h_6 \oplus h_7$$
$$c_4 = h_4 \oplus h_5 \oplus h_6 \oplus h_7.$$

- If a nonzero value is found, the decoder simply complements the code word bit position indicated by the parity word. The decoded binary value is then extracted from the corrected code word as $h_3h_5h_6h_7$.

Error Free Compression:

Variable-Length Coding:

- The simplest approach to error-free image compression is to reduce only coding redundancy. Coding redundancy normally is present in any natural binary encoding of the gray levels in an image. It can be eliminated by coding the gray levels. To do so requires construction of a variable-length code that assigns the shortest possible code words to the most probable gray levels. Here, we examine several optimal and near optimal techniques for constructing such a code. These techniques are formulated in the language of information theory. In practice, the source symbols may be either the gray levels of an image or the output of a gray-level mapping operation (pixel differences, run lengths, and so on).

Huffman coding:

- The most popular technique for removing coding redundancy is due to Huffman (Huffman [1952]). When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol. In terms of the noiseless coding theorem, the resulting code is optimal for a fixed value of n, subject to the constraint that the source symbols be coded one at a time.
- The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction. Figure 4.1 illustrates this process for binary coding (K-ary Huffman codes can also be constructed). At the far left, a hypothetical set of source symbols and their probabilities are ordered from top to bottom in terms of decreasing probability values. To form the first source reduction, the bottom two probabilities, 0.06 and 0.04, are combined to form a "compound symbol" with probability 0.1. This compound symbol and its associated probability are placed in the first source reduction column so that the probabilities of the reduced source are also ordered from the most to the least probable. This process is then repeated until a reduced source with two symbols (at the far right) is reached.
- The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. The minimal length binary code for a two-symbol source, of course, is the symbols 0 and 1. As Fig. 4.2 shows, these

symbols are assigned to the two symbols on the right (the assignment is arbitrary; reversing the order of the 0 and 1 would work just as well). As the reduced source symbol with probability 0.6 was generated by combining two symbols in the reduced source to its left, the 0 used to code it is now assigned to both of these symbols, and a 0 and 1 are arbitrarily appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original source is reached. The final code appears at the far left in Fig.4.2. The average length of this code is

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)$$
$$= 2.2 \text{ bits/symbol}$$

and the entropy of the source is 2.14 bits/symbol. The resulting Huffman code efficiency is 0.973.

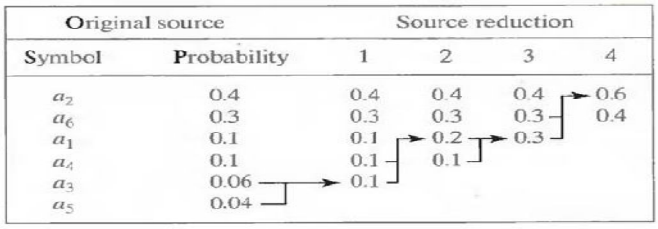


Fig.4.1 Huffman source reductions.

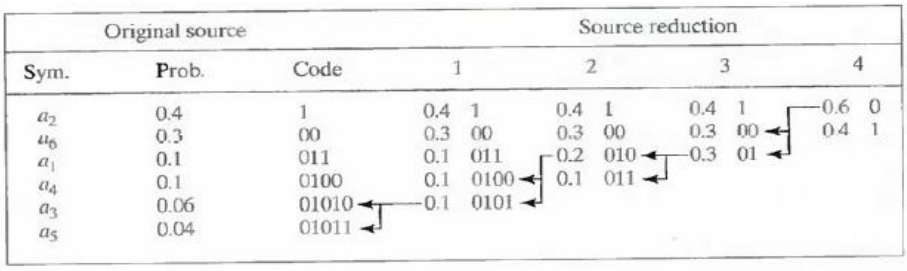


Fig.4.2 Huffman code assignment procedure.

- Huffman's procedure creates the optimal code for a set of symbols and probabilities subject to the constraint that the symbols be coded one at a time. After the code has been created, coding and/or decoding is accomplished in a simple lookup table manner. The code itself is an instantaneous uniquely decodable block code. It is called a block code because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner. For the binary code of Fig. 4.2, a left-to-right scan of the encoded string 010100111100 reveals that the first valid code word is 01010, which is the code for symbol a3 .The next valid code is 011, which corresponds to symbol a1. Continuing in this manner reveals the completely decoded message to be a3a1a2a2a6.

LZW Coding:

- Variable length sequences of source symbols but requires no a priori knowledge of the probability of occurrence of the symbols to be encoded. LZW compression has been

integrated into a variety of mainstream imaging file formats, including the graphic interchange format (GIF), tagged image file format (TIFF), and the portable document format (PDF).

- LZW coding is conceptually very simple (Welch [1984]). At the onset of the coding process, a codebook or "dictionary" containing the source symbols to be coded is constructed. For 8-bit monochrome images, the first 256 words of the dictionary are assigned to the gray values 0, 1, 2..., and 255. As the encoder sequentially examines the image's pixels, gray level sequences that are not in the dictionary are placed in algorithmically determined (e.g., the next unused) locations. If the first two pixels of the image are white, for instance, sequence “255- 255” might be assigned to location 256, the address following the locations reserved for gray levels 0 through 255. The next time that two consecutive white pixels are encountered, code word 256, the address of the location containing sequence 255-255, is used to represent them. If a 9-bit, 512-word dictionary is employed in the coding process, the original (8 + 8) bits that were used to represent the two pixels are replaced by a single 9-bit code word. Cleary, the size of the dictionary is an important system parameter. If it is too small, the detection of matching gray level sequences will be less likely; if it is too large, the size of the code words will adversely affect compression performance.
- Consider the following 4 x 4, 8-bit image of a vertical edge:

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

- Table 6.1 details the steps involved in coding its 16 pixels. A 512-word dictionary with the following starting content is assumed:

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

- Locations 256 through 511 are initially unused. The image is encoded by processing its pixels in a left-to-right, top-to-bottom manner. Each successive gray-level value is concatenated with a variable—column 1 of Table 6.1 —called the "currently recognized sequence." As can be seen, this variable is initially null or empty. The dictionary is searched for each concatenated sequence and if found, as was the case in the first row of the table, is replaced by the newly concatenated and recognized (i.e., located in the dictionary) sequence. This was done in column 1 of row 2.

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

Table 6.1 LZW coding example

- No output codes are generated, nor are the dictionary altered. If the concatenated sequence is not found, however, the address of the currently recognized sequence is output as the next encoded value, the concatenated but unrecognized sequence is added to the dictionary, and the currently recognized sequence is initialized to the current pixel value. This occurred in row 2 of the table. The last two columns detail the gray-level sequences that are added to the dictionary when scanning the entire 4 x 4 image. Nine additional code words are defined. At the conclusion of coding, the dictionary contains 265 code words and the LZW algorithm has successfully identified several repeating gray-level sequences—leveraging them to reduce the original 128-bit image lo 90 bits (i.e., 10 9-bit codes). The encoded output is obtained by reading the third column from top to bottom. The resulting compression ratio is 1.42:1.
- A unique feature of the LZW coding just demonstrated is that the coding dictionary or code book is created while the data are being encoded. Remarkably, an LZW decoder builds an identical decompression dictionary as it decodes simultaneously the encoded data stream. . Although not needed in this example, most practical applications require a strategy for handling dictionary overflow. A simple solution is to flush or reinitialize the dictionary when it becomes full and continue coding with a new initialized dictionary. A more complex option is to monitor compression performance and flush the dictionary when it becomes poor or unacceptable. Alternately, the least used dictionary entries can be tracked and replaced when necessary.

Bit-Plane Coding:

- An effective technique for reducing an image's interpixel redundancies is to process the image's bit planes individually. The technique, called bit-plane coding, is based on the concept of decomposing a multilevel (monochrome or color) image into a series of binary images and compressing each binary image via one of several well-known binary compression methods.

Bit-plane decomposition:

- The gray levels of an m-bit gray-scale image can be represented in the form of the base 2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_1 2^1 + a_0 2^0.$$

- Based on this property, a simple method of decomposing the image into a collection of binary images is to separate the m coefficients of the polynomial into m 1-bit bit planes. The zeroth order bit plane is generated by collecting the a0 bits of each pixel, while the (m - 1) st-order bit plane contains the am-1, bits or coefficients. In general, each bit plane is numbered from 0 to m-1 and is constructed by setting its pixels equal to the values of the appropriate bits or polynomial coefficients from each pixel in the original image. The inherent disadvantage of this approach is that small changes in gray level can have a significant impact on the complexity of the bit planes. If a pixel of intensity 127 (01111111) is adjacent to a pixel of intensity 128 (10000000), for instance, every bit plane will contain a corresponding 0 to 1 (or 1 to 0) transition.
- For example, as the most significant bits of the two binary codes for 127 and 128 are different, bit plane 7 will contain a zero-valued pixel next to a pixel of value 1, creating a 0 to 1 (or 1 to 0) transition at that point.
- An alternative decomposition approach (which reduces the effect of small gray-level variations) is to first represent the image by an m-bit Gray code. The m-bit Gray code gm-1... g2g1g0 that corresponds to the polynomial in Eq. above can be computed from

$$g_i = a_i \oplus a_{i+1} \qquad 0 \le i \le m-2$$
$$g_{m-1} = a_{m-1}.$$

- Here, $\oplus$ denotes the exclusive OR operation. This code has the unique property that successive code words differ in only one bit position. Thus, small changes in gray level are less likely to affect all m bit planes. For instance, when gray levels 127 and 128 are adjacent, only the 7th bit plane will contain a 0 to 1 transition, because the Gray codes that correspond to 127 and 128 are 11000000 and 01000000, respectively.

Lossless Predictive Coding:

- The error-free compression approach does not require decomposition of an image into a collection of bit planes. The approach, commonly referred to as lossless predictive coding, is based on eliminating the interpixel redundancies of closely spaced pixels by extracting and coding only the new information in each pixel. The new information of a pixel is defined as the difference between the actual and predicted value of that pixel.
- Figure 8.1 shows the basic components of a lossless predictive coding system. The system consists of an encoder and a decoder, each containing an identical predictor. As each successive pixel of the input image, denoted f_n, is introduced to the encoder, the predictor generates the anticipated value of that pixel based on some number of past inputs. The output of the predictor is then rounded to the nearest integer, denoted $\hat{f}_n$ and used to form the difference or prediction error which is coded using a variable-length code (by the symbol encoder) to generate the next element of the compressed data stream.

$$e_n = f_n - \hat{f}_n,$$

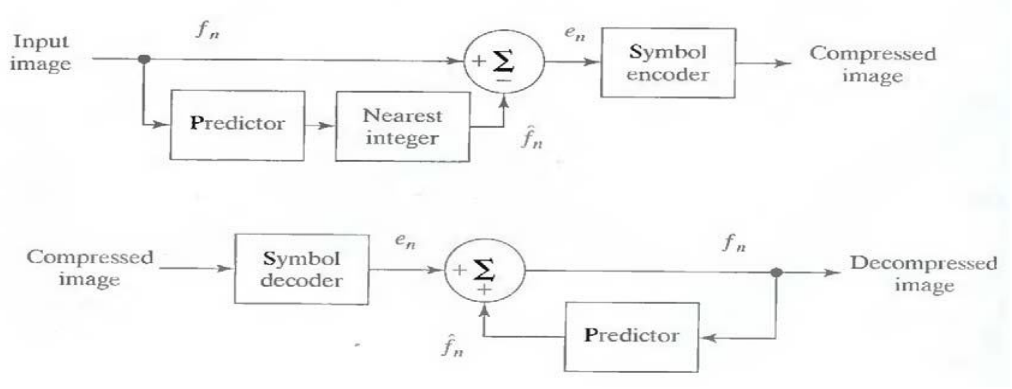


Fig.8.1 A lossless predictive coding model: (a) encoder; (b) decoder

- The decoder of Fig. 8.1 (b) reconstructs en from the received variable-length code words and performs the inverse operation

$$f_n = e_n + \hat{f}_n.$$

- Various local, global, and adaptive methods can be used to generate $\hat{f}_n$. In most cases, however, the prediction is formed by a linear combination of m previous pixels. That is,

$$\hat{f}_n = \text{round}\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

- Where m is the order of the linear predictor, round is a function used to denote the rounding or nearest integer operation, and the αi, for i = 1,2,..., m are prediction coefficients. In raster scan applications, the subscript n indexes the predictor outputs in accordance with their time of occurrence. That is, f_n, $\hat{f}_n$ and en in Eqns. above could be replaced with the more explicit notation f (t), $\hat{f}$(t), and e (t), where t represents time. In other cases, n is used as an index on the spatial coordinates and/or frame number (in a time sequence of images) of an image. In 1-D linear predictive coding, for example, Eq. above can be written as

$$\hat{f}_n(x, y) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(x, y - i)\right]$$

- where each subscripted variable is now expressed explicitly as a function of spatial coordinates x and y. The Eq. indicates that the 1-D linear prediction f(x, y) is a function of the previous pixels on the current line alone. In 2-D predictive coding, the prediction is a function of the previous pixels in a left-to-right, top-to-bottom scan of an image. In the 3-D case, it is based on these pixels and the previous pixels of preceding frames. Equation above cannot be evaluated for the first m pixels of each line, so these pixels must be coded by using other means (such as a Huffman code) and considered as an overhead of the predictive coding process. A similar comment applies to the higher-dimensional cases.

Lossy Predictive Coding:

- In this type of coding, we add a quantizer to the lossless predictive model and examine the resulting trade-off between reconstruction accuracy and compression performance. As Fig.9 shows, the quantizer, which absorbs the nearest integer function of the error-free encoder, is inserted between the symbol encoder and the point at which the prediction error is formed. It maps the prediction error into a limited range of outputs, denoted $\hat{e}_n$ which establish the amount of compression and distortion associated with lossy predictive coding.

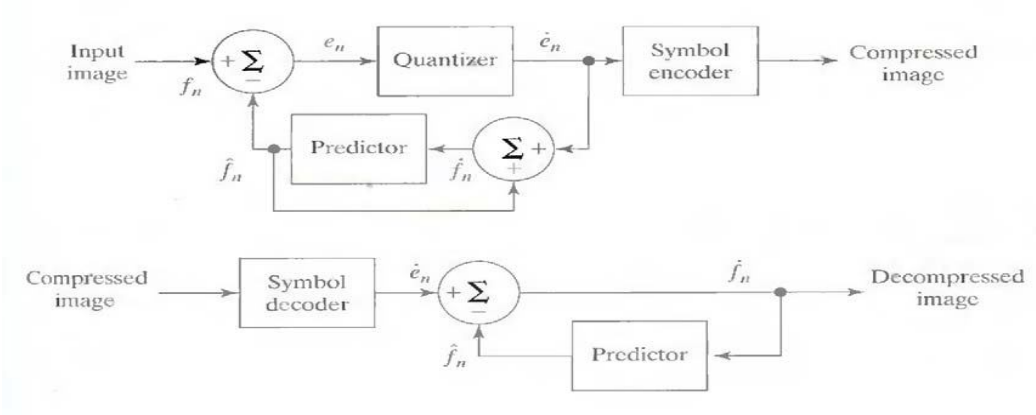


Fig. 9 A lossy predictive coding model: (a) encoder and (b) decoder.

- In order to accommodate the insertion of the quantization step, the error-free encoder of figure must be altered so that the predictions generated by the encoder and decoder are equivalent. As Fig.9 (a) shows, this is accomplished by placing the lossy encoder's predictor within a feedback loop, where its input, denoted f˙n, is generated as a function of past predictions and the corresponding quantized errors. That is,

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

- This closed loop configuration prevents error buildup at the decoder's output. Note from Fig. 9 (b) that the output of the decoder also is given by the above Eqn.

Optimal predictors:

- The optimal predictor used in most predictive coding applications minimizes the encoder's meansquare prediction error

$$E\{e_n^2\} = E\{[f_n - \hat{f}_n]^2\}$$

subject to the constraint that

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f_n$$

And

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i f_{n-i}.$$

- That is, the optimization criterion is chosen to minimize the mean-square prediction error, the quantization error is assumed to be negligible (e˙n ≈ en), and the prediction is constrained to a linear combination of m previous pixels.1 These restrictions are not essential, but they simplify the analysis considerably and, at the same time, decrease the computational complexity of the predictor. The resulting predictive coding approach is referred to as differential pulse code modulation (DPCM).

Transform Coding:

- All the predictive coding techniques operate directly on the pixels of an image and thus are spatial domain methods. In this coding, we consider compression techniques that are based on modifying the transform of an image. In transform coding, a reversible, linear transform (such as the Fourier transform) is used to map the image into a set of transform coefficients, which are then quantized and coded. For most natural images, a significant number of the coefficients have small magnitudes and can be coarsely quantized (or discarded entirely) with little image distortion. A variety of transformations, including the discrete Fourier transform (DFT), can be used to transform the image data.

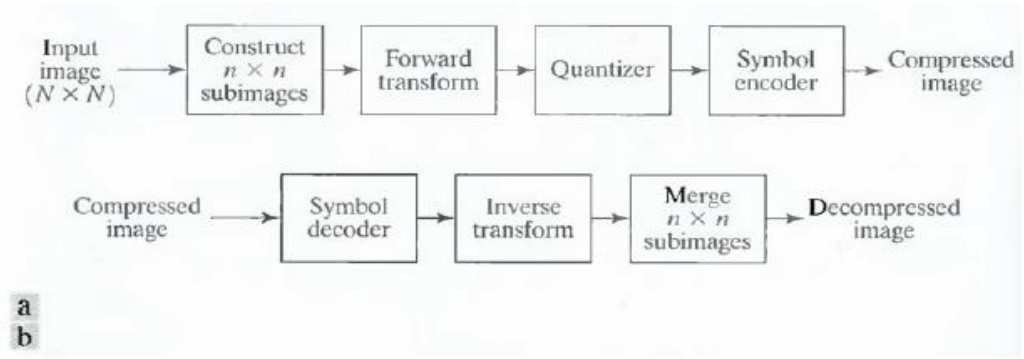


Fig. 10 A transform coding system: (a) encoder; (b) decoder.

- Figure 10 shows a typical transform coding system. The decoder implements the inverse sequence of steps (with the exception of the quantization function) of the encoder, which performs four relatively straightforward operations: sub image decomposition, transformation, quantization, and coding. An N X N input image first is subdivided into sub images of size n X n, which are then transformed to generate (N/n) 2 sub image transform arrays, each of size n X n. The goal of the transformation process is to decorrelate the pixels of each sub image, or to pack as much information as possible into the smallest number of transform coefficients. The quantization stage then selectively eliminates or more coarsely quantizes the coefficients that carry the least information. These coefficients have the smallest impact on reconstructed sub image quality. The encoding process terminates by coding (normally using a variable-length code) the quantized coefficients. Any or all of the transform encoding steps can be adapted to local image content, called adaptive transform coding, or fixed for all sub images, called non adaptive transform coding.

Wavelet Coding:

- The wavelet coding is based on the idea that the coefficients of a transform that decorrelates the pixels of an image can be coded more efficiently than the original pixels themselves. If the transform's basis functions—in this case wavelets—pack most of the important visual information into a small number of coefficients, the remaining coefficients can be quantized coarsely or truncated to zero with little image distortion.
- Figure 11 shows a typical wavelet coding system. To encode a 2^J X 2^J image, an analyzing wavelet, Ψ, and minimum decomposition level, J - P, are selected and used to compute the image's discrete wavelet transform. If the wavelet has a complimentary scaling function φ, the fast wavelet transform can be used. In either case, the computed transform converts a large portion of the original image to horizontal, vertical, and diagonal decomposition coefficients with zero mean and Laplacian-like distributions.

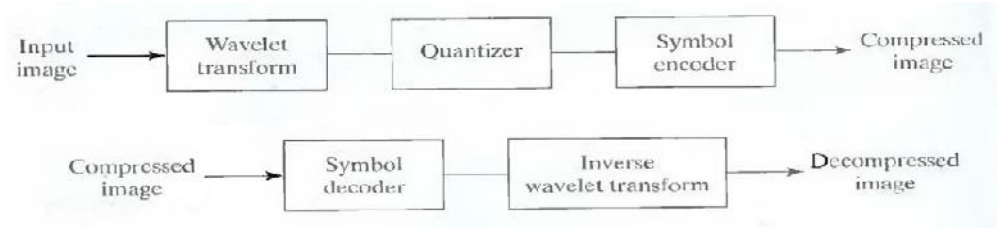


Fig.11 A wavelet coding system: (a) encoder; (b) decoder.

- Since many of the computed coefficients carry little visual information, they can be quantized and coded to minimize inter coefficient and coding redundancy. Moreover, the quantization can be adapted to exploit any positional correlation across the P decomposition levels. One or more of the lossless coding methods, including run-length, Huffman, arithmetic, and bit-plane coding, can be incorporated into the final symbol coding step. Decoding is accomplished by inverting the encoding operations—with the exception of quantization, which cannot be reversed exactly.
- The principal difference between the wavelet-based system and the transform coding system is the omission of the transform coder's sub image processing stages. Because wavelet transforms are both computationally efficient and inherently local (i.e., their basis functions are limited in duration), subdivision of the original image is unnecessary.

UNIT-IV
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information then compression ratio C_R of the first data set (the one characterized by n_1) is calculated as_____ [ ]

A) n_1/n_2 B) n_1*n_2 C) n_2/n_1 D) none of the above

2. If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy R_D of the first data set (the one characterized by n_1) can be defined as_____

3. In coding redundancy, If the number of bits used to represent each value of r_k is l (r_k), then the average number of bits required to represent each pixel is_____

4. In coding redundancy, the total number of bits required to code an M X N image is MNL_{avg}. [True/False]

5. _____ is responsible for reducing or eliminating any coding, interpixel, or psychovisual redundancies in the input image. [ ]

A)source encoder B)source decoder C)channel encoder D)channel decoder

6. Quantizer stage reduces the psychovisual redundancies of the input image. This operation is irreversible. [True/False]

7. ___it must be omitted when error-free compression is desired.[ ]

A) Quantizer B) Mapper C) Symbol Encoder D) All

8. All the predictive coding techniques operate directly on the pixels of an image and thus are spatial domain methods [True/False]

9. In Transform coding An N X N input image first is subdivided into subimages of size n X n, which are then transformed to generate $(N/n)^2$ subimage transform arrays, each of size n X n. [True**/**False]

10. For any value of x and y, the error e(x, y) between f (x, y) and f^ (x, y) can be defined as_____

11. If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information then compression ratio C_R of the first data set (the one characterized by n_1) is_____for the case $n_1>>n_2$

[ ]

A) 0 B) 1 C) ∞ D) none.

12. If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information, the relative data

redundancy R_D of the first data set (the one characterized by n_1) can _____for the case $n_1=n_2$ [ ]

A) 0 B) 1 C) Both D) none

13. Thus the total number of bits required to code an (M X N) 256X256 image with $L_{avg}=1.18$ is_____ MNL_{avg}.

14. Gray code that correspond to $(127)_{10}$ is_____ [ ]

A)11000000 B) 01000000 C) 01111111 D) 10000000

15. Binary code that correspond to $(127)_{10}$ is_____ [ ]

A)11000000 B) 01000000 C) 01111111 D) 10000000

16. An alphabet consists of the letters A, B, C and D. The probability of occurrence is P(A) = 0.4, P(B) = 0.1, P(C) = 0.2 and P(D) = 0.3. The Huffman code is [ ]

A) A = 0 B = 11 C = 10 D = 111

B) A = 0 B = 111 C = 11 D = 101

C) A = 0 B = 111 C = 110 D = 10

D) A = 01 B = 111 C = 110 D = 10

17. A Huffman code: A = 1, B = 000, C = 001, D = 01 P(A) = 0.4, P(B) = 0.1, P(C) = 0.2, P(D) = 0.3 The average number of bits per letter is [ ]

A) 2.1 bit B) 1.9 bit C) 8.0 bit D) 2.0 bit

18. Given a Gray Code: $g_3\ g_2\ g_1\ g_0$ = 1 0 0 1 then Binary Code: $b_3\ b_2\ b_1\ b_0$ is: [ ]

A) 1 1 1 0 B)1 1 1 1 C)1 0 1 1 D)0 1 0 1

19. If f^ (x, y) is considered to be the sum of the original image f(x, y) and a noise signal e(x, y), the mean-square signal-to-noise ratio of the output image SNRrms, is calculated as_____

20. If the images are of size M X N is given. The root-mean-square error, e_{rms}, between f(x, y) and f^(x,y) then is _____

SECTION-B

SUBJECTIVE QUESTIONS

1. Define image compression. Explain about the redundancies in a digital image.
2. Explain about fidelity criterion.
3. Explain about image compression models.
4. Explain a method of generating variable length codes with an example.
5. Explain arithmetic encoding process with an example.
6. Explain LZW coding with an example
7. Explain the concept of bit plane coding method.
8. Explain with a block diagram about transform coding system.
9. Explain about wavelet coding
10. Calculate the Channel Encoder and Decoder for the bits 1000 using Hamming (7,4) code
11. Explain a method of generating variable length codes with an example
12. Apply Huffman's approach source reductions for the given probabilities.

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4				
a_6	0.3				
a_1	0.1				
a_4	0.1				
a_3	0.06				
a_5	0.04				

13. Apply Huffman's approach assignment procedure or working back to the original source for the given data. And calculate average length of the code.

Original source			Source reduction			
Sym.	Prob.	Code	1	2	3	4
a_2	0.4					0.6 0
a_6	0.3					0.4 1
a_1	0.1					
a_4	0.1					
a_3	0.06					
a_5	0.04					

14. Convert the $(127)_{10}$ and $(128)_{10}$ numbers in to binary and gray code binary and relate this process to Bit-Plane Coding.
15. Derive lossless predictive coding.
16. Derive lossy predictive coding.

UNIT – V

UNIT - V: Morphological Image Processing Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms.

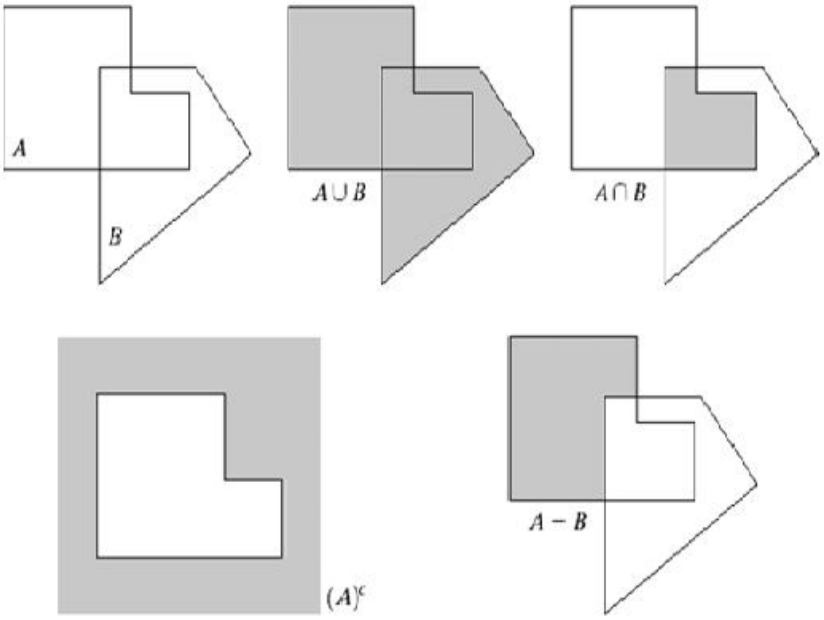
- The word morphology refers to the scientific branch that deals the forms and structures of animals/plants.
- Morphology in image processing is a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries and skeletons.
- The morphological operations can be used for filtering, thinning and pruning.
- The language of the Morphology comes from the set theory, where image objects can be represented by sets. For example an image object containing black pixels can be considered a set of black pixels in 2D space of Z^2.where each element of a set is a tuple (2-D vector) whose coordinates are the (x,y) coordinates of a black (or white depending on convention) pixel in the image.
- Gray scale digital image can be represented as sets whose components are in Z^3. In this case two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete gray level value.

Preliminaries:

Some Basic Concepts form Set Theory:

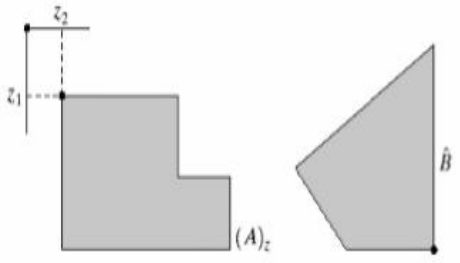
- Let A be a set in Z^2, If a= (a_1, a_2) is an element of A, then we write $a \in A$
- Similarly, if a is not an element of A, we write $a \notin A$
- The set with no elements is called the null or empty set and is denoted by the symbol $\emptyset$.
- When we write an expression of the form C={w|w = -d, for d∈D} we mean that set C is the set of elements w, such that w is formed by multiplying each of the two coordinates of all the elements of set D by -1.
- If every element of a set A is also an element of another set B, Then A is said to be a subset of B.
- If every element of a set A is also an element of another set B, then A is said to be a subset of B, denoted as $A \subseteq B$
- The union of two sets A and B, denoted by $C = A \cup B$
- The intersection of two sets A and B, denote by $D = A \cap B$ {p|p ∈ A and p ∈ B}

- The sets A and B are said to be Disjoint or mutually exclusive if they have no common elements, denoted by $A \cap B = \varnothing$
- The complement of a set A is the set of elements not contained in A $A^c = \{\omega | \omega \notin A\}$
- The difference of two sets A and B, denoted A -B, is defined as $A - B = \{\omega | \omega \in A, \omega \notin B\} = A \cap B^c$
  We see that this is the set of elements that belongs to A, but not to B



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*.

- The reflection of set B, denoted B^ ,is defined as
$$\hat{B} = \{\omega | \omega = -b, for \quad b \in B\}$$
- The translation of set A by point z = (z_1,z_2), denoted $(A)_z$ is defined as
$$(A)_z = \{c | c = a + z, for \quad a \in A\}$$



a b
FIGURE 9.2
(a) Translation of *A* by *z*. (b) Reflection of *B*. The sets *A* and *B* are from Fig. 9.1.

- Set reflection and set translation are used to formulate operations based on so-called structuring elements.

Logic operators involving Binary images:

- The main logic operators used in image processing are AND, OR and NOT (COMPLIMENT). These properties are summarized in the following table9.1. These operations are functionally complete in the sense that they can be combined to form another logic operation.
- Logic operations are performed on a pixel by pixel basis between corresponding pixels of two or more images (except NOT, which operates on the pixels of a single image)
- Basic AND operation of two binary variables is 1 only when both variables are 1, the result at any location in a resulting AND image is 1 only if the corresponding pixels in the two input images are 1.

TABLE 9.1
The three basic logical operations.

p	q	p AND q (also $p \cdot q$)	p OR q (also $p + q$)	NOT (p) (also $\bar{p}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

- The following figure shows the various examples of logical operations involving images, where black indicates a binary 1 and white indicates a 0.
- The XOR (exclusive OR) operation yields a 1 when one or the other pixel (but not both) is 1, and it yields a 0 otherwise. This operation is unlike the OR operation, which is 1 when one or the other pixel is 1, or when both pixels are 1.
- NOT-AND operation selects the black pixels that simultaneously are in B, and not in A.

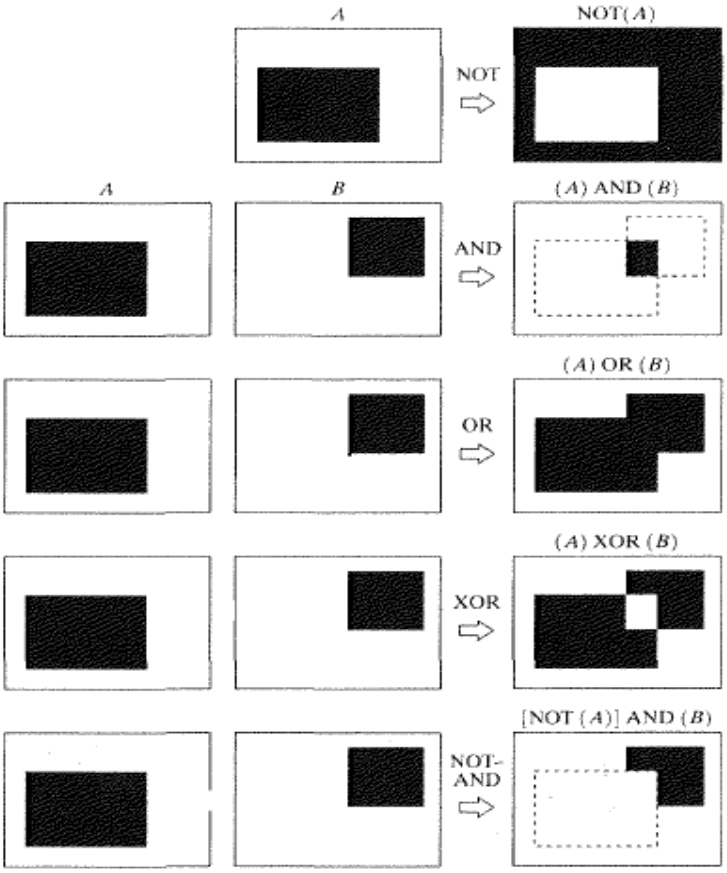


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Dilation and erosion:

- Dilation and erosion are basic morphological processing operations. They are defined in terms of more elementary set operations, but are employed as the basic elements of many algorithms.
- Both dilation and erosion are produced by the interaction of a set called a structuring element with a set of pixels of interest in the image. The structuring element has both a shape and an origin.

Dilation:

- With A and B as set in Z2, the dilation of A by B, denoted A B, is defined as

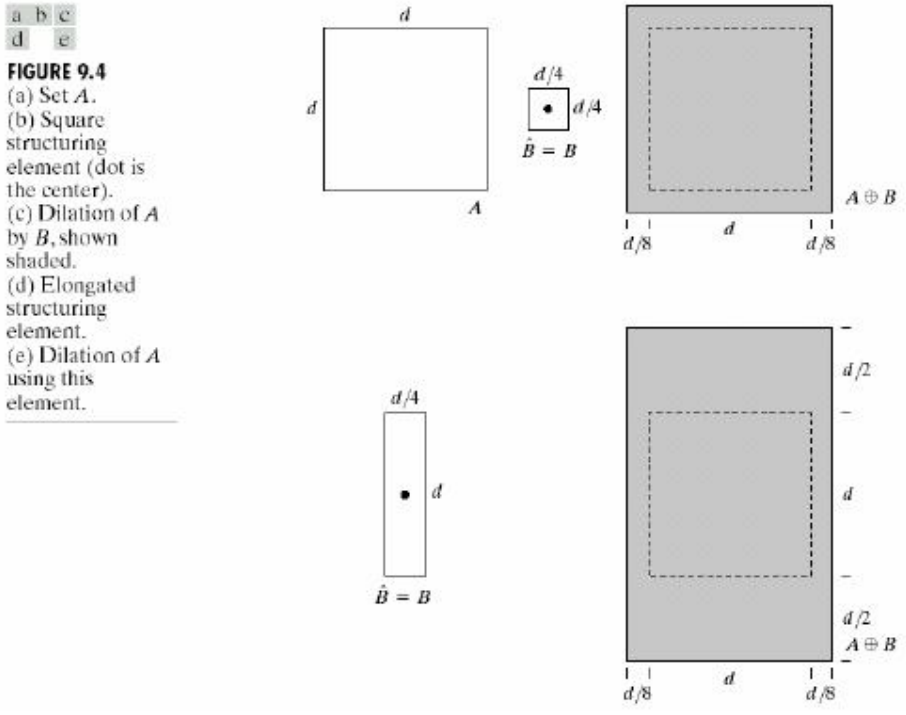
$$A \oplus B = \left\{ z \middle| \left(\hat{B} \right)_z \cap A \neq \varnothing \right\}$$

- This equation is based on Reflection of B about the origin, and shifts the reflection by z.

The dilation of A by B then is the set of all displacements, z, such that B^ and A overlap by at least one element.

$$A \oplus B = \left\{ z \middle| \left[(\hat{B})_z \cap A \right] \subseteq A \right\}$$

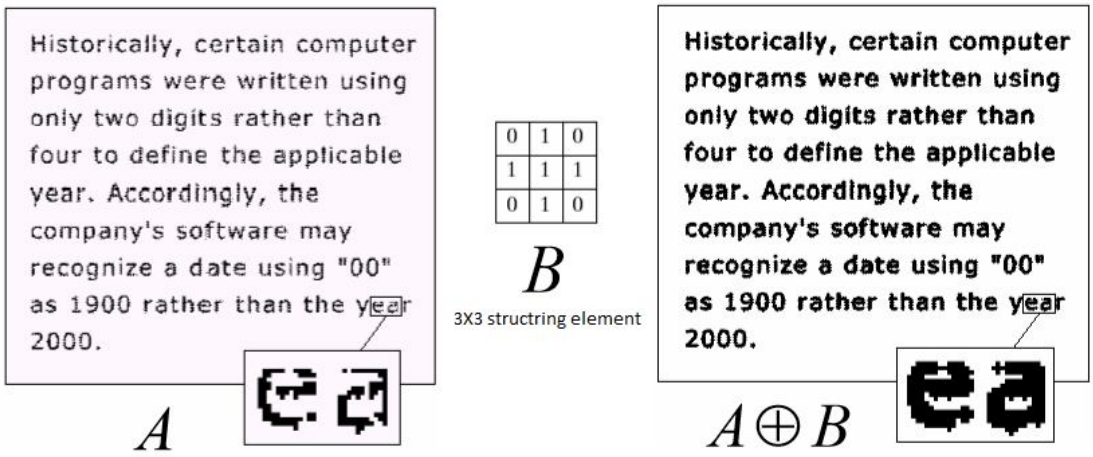
- Set B is commonly referred to as the structuring element in dilation.
- Although dilation is based on set operations, where as convolution is based on arithmetic operations, the basic process of "flipping" B about its origin and then successively displacing it so that it slides over set A is analogues to the convolution process.
- EX: In the following figure 9.4(a) shows a simple set 9.4 (b) shows a structuring element and its reflection(The dark dot denotes the origin of the element). In this case the structuring element and its reflection are equal because B is symmetric with respect to its origin. The dashed line in figure 9.4(c) shows the original set for reference, and the solid line shows the limit beyond which any further displacements of the origin of B^ by Z would cause the intersection of B^ and A to be empty. Therefore, all points inside this boundary constitute the dilation of A by B.
- Figure 9.4(d) shows a structuring element designed to achieve more dilation vertically than horizontally figure 9.4(e) shows the dilation achieved with this element.



a b c
d e

FIGURE 9.4
(a) Set A.
(b) Square structuring element (dot is the center).
(c) Dilation of A by B, shown shaded.
(d) Elongated structuring element.
(e) Dilation of A using this element.

- One of the simplest applications of dilation is for bridging the gaps.
- Given the following distorted text image where the maximum length of the broken characters is 2 pixels. The image can be enhanced by bridging the gaps by using the structuring element given below:

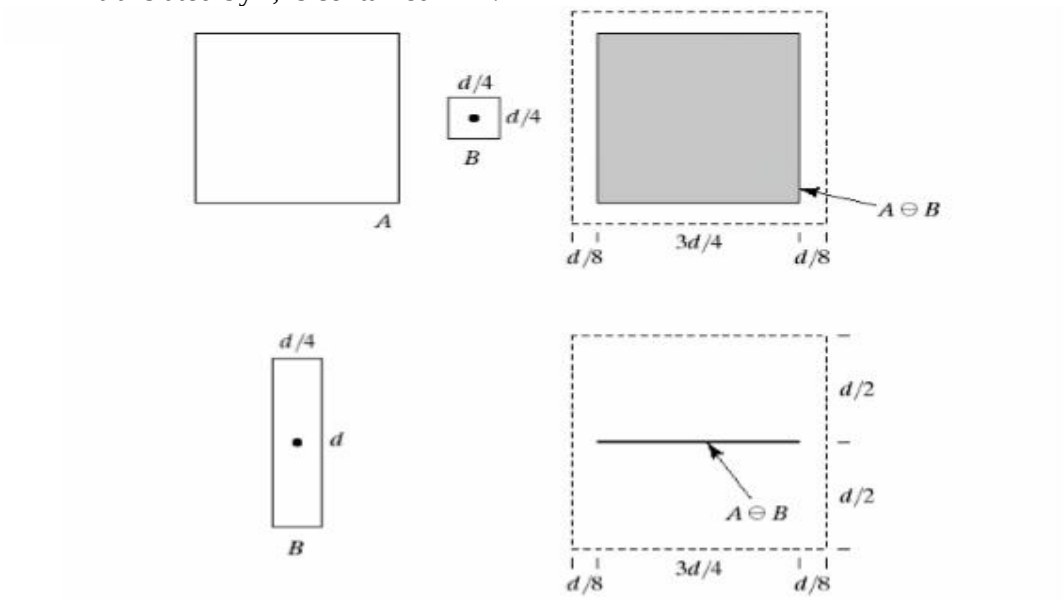


Erosion:

- Given A and B sets in Z^2, the erosion of A by structuring element B, is defined by:

$$A \ominus B = \{z | (B)_z \subseteq A\}$$

- The erosion of A by structuring element B is the set of all points z, such that B, translated by z, is contained in A.



a b c
d e

FIGURE 9.6 (a) Set A. (b) Square structuring element. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A using this element.

- Set A is shown as a dashed line for reference in Fig9.6(c). The boundary of the shaded region shows the limit beyond which further displacement of the origin of B would cause this set to cause being completely contained in A. Thus, the locus of points within this boundary constitutes the erosion of A by B.
- Erosion and dilation are duals of each other with respect to set complementation and reflection

$$(A \ominus B)^c = A^c \oplus \hat{B}$$
$$(A \oplus B)^c = A^c \ominus \hat{B}$$

- Duality property is especially useful when SE is symmetric with respect to its origin so that B^ = B
- Allows for erosion of an image by dilating its background (A^c) using the same SE and complementing the results

Proving duality:

- Erosion can be written as

$$(A \ominus B)^c = \{z \mid (B)_z \subseteq A\}^c$$

$$(B)_z \subseteq A \Rightarrow (B)_z \cap A^c = \emptyset$$

- So, the previous expression yields

$$(A \ominus B)^c = \{z \mid (B)_z \cap A^c = \emptyset\}^c$$

- The complement of the set of z's that satisfy (B)z $\cap A^c = \emptyset$ is the set of z's such that (B)z $\cap A^c$!= $\emptyset$
- This leads to

$$(A \ominus B)^c = \{z \mid (B)_z \cap A^c \neq \emptyset\}$$
$$= A^c \oplus \hat{B}$$

Opening and Closing:

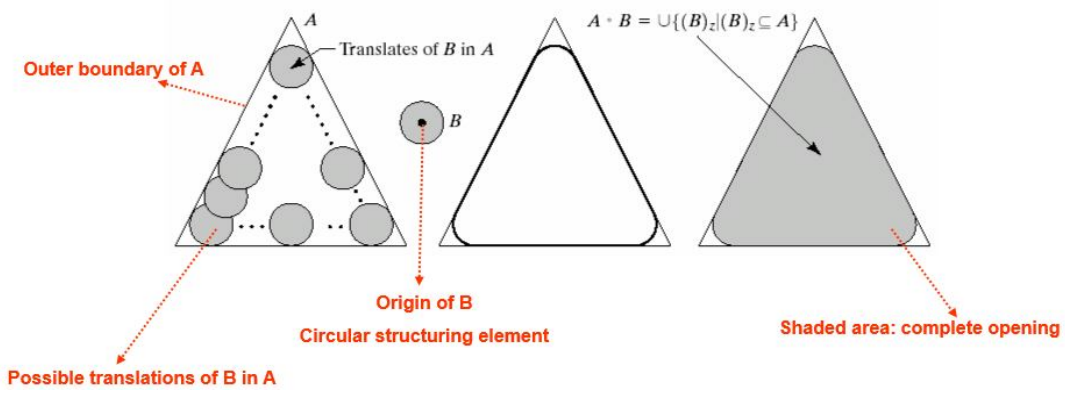
- As we have seen, dilation expands an image and erosion shrinks it.
- Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions.
- Closing also tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.
- The process of erosion followed by dilation is called **opening**. It has the effect of eliminating small and thin objects, breaking the objects at thin points and smoothing the boundaries/contours of the objects.
- Opening of a set A by se B, denoted by A◦B, is defined by

$$A \circ B = (A \ominus B) \oplus B$$

- Erosion of A by B, followed by a dilation of the result by B is **Opening A by B.**

- The opening of A by the structuring element B is obtained by taking the union of all translates of B that fit into A.
- The opening operation can also be expressed by the following formula:

$$A \circ B = \cup\{B_z | (B_z) \subseteq A\}$$

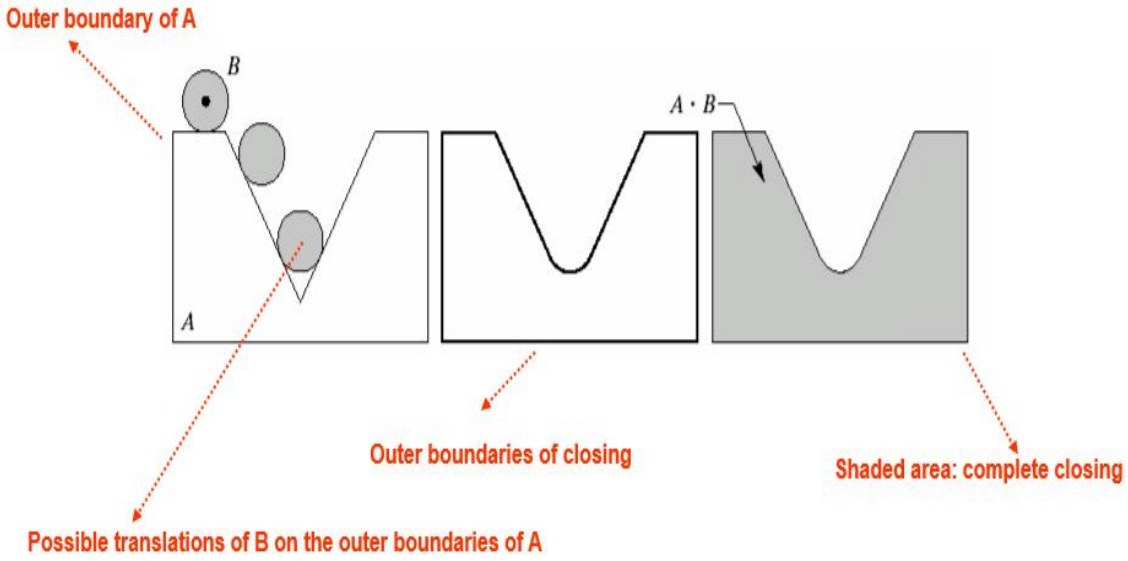


- The process of dilation followed by erosion is called **closing**. It has the effect of filling small and thin holes, connecting nearby objects and smoothing the boundaries/contours of the objects.
- Closing of a set A by se B, denoted by A•B, is defined by

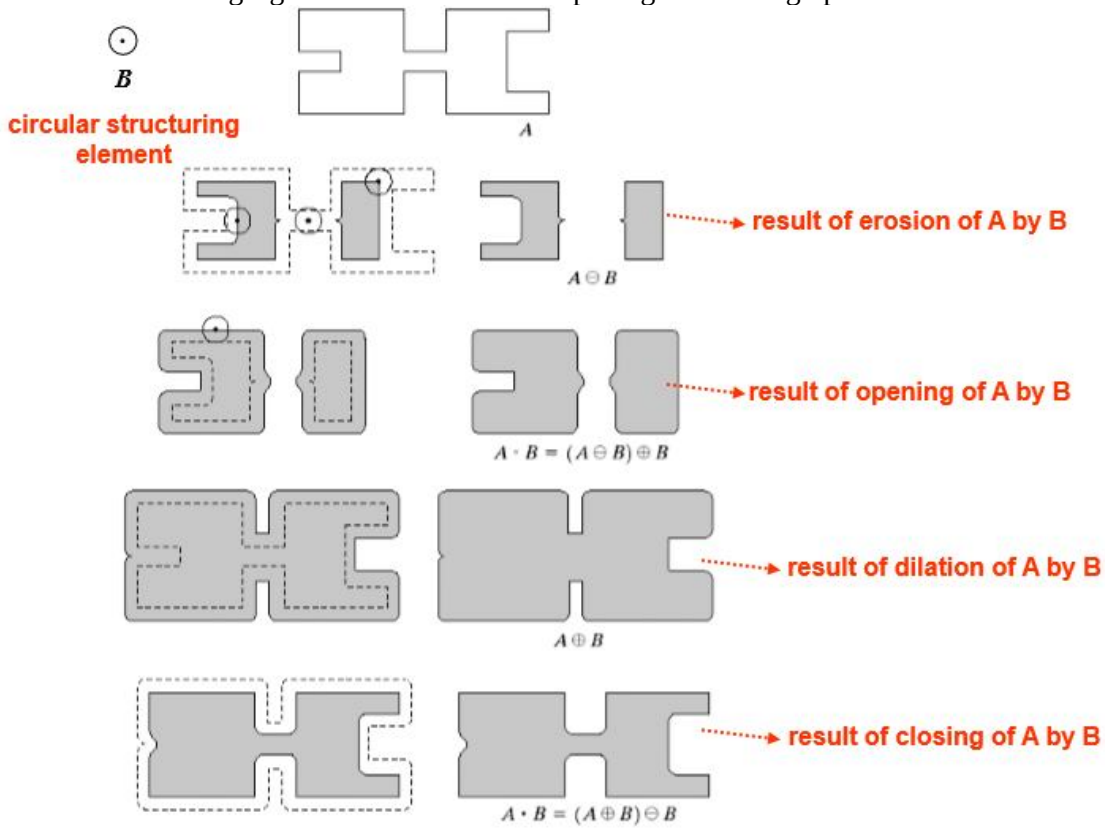
$$A \bullet B = (A \oplus B) \ominus B$$

- Dilation of A by B, followed by the erosion of the result by B is **Closing A by B.**
- The closing has a similar geometric interpretation except that we roll B on the outside of the boundary.
- The opening operation can also be expressed by the following formula:

$$A \bullet B = \cup\{(B_z) | (B_z) \cap A \neq \varnothing\}$$



- The following figure 9.10 illustrates the opening and closing operations.



- As in the case of dilation and erosion, opening and closing are duels of each other with respect to set complementation and reflection. That is,

$$(A \bullet B)^c = (A^c \circ \hat{B})$$
$$(A \circ B)^c = (A^c \bullet \hat{B})$$

- Opening operation satisfies the following properties:

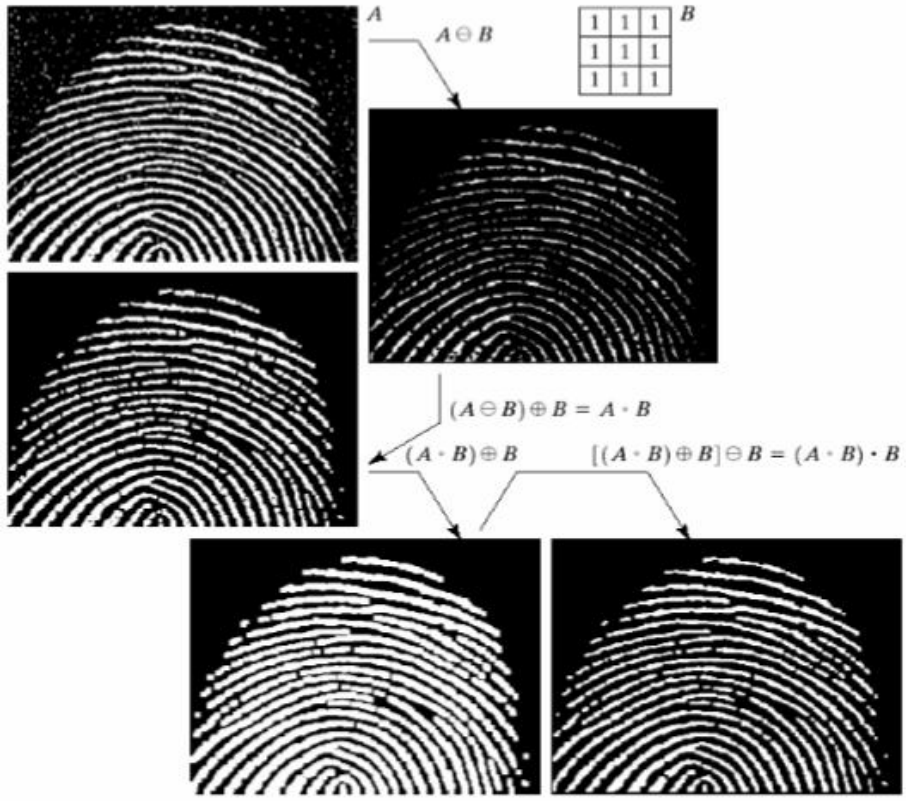
1. $A \circ B \subseteq A$
2. $C \subseteq D \Rightarrow C \circ B \subseteq D \circ B$
3. $(A \circ B) \circ B = A \circ B$

- Similarly, closing operation satisfies:

1. $A \subseteq A \bullet B$
2. $C \subseteq D \Rightarrow C \bullet B \subseteq D \bullet B$
3. $(A \bullet B) \bullet B = A \bullet B$

- In both the above cases, multiple applications of opening and closing have no effect after the first application.
- Example: Removing noise from finger prints
- The background noise was completely eliminated in the erosion stage of opening because in the case all noise components are physically smaller than the structuring element.

- We note in Fig 9.11(d) that the net effect of opening was to eliminate virtually all noise components in both the background and the fingerprint itself.
- However, new gaps between the finger print ridges were created. To counter this undesirable effect, we perform dilation on the opening.



a b
d c
e f

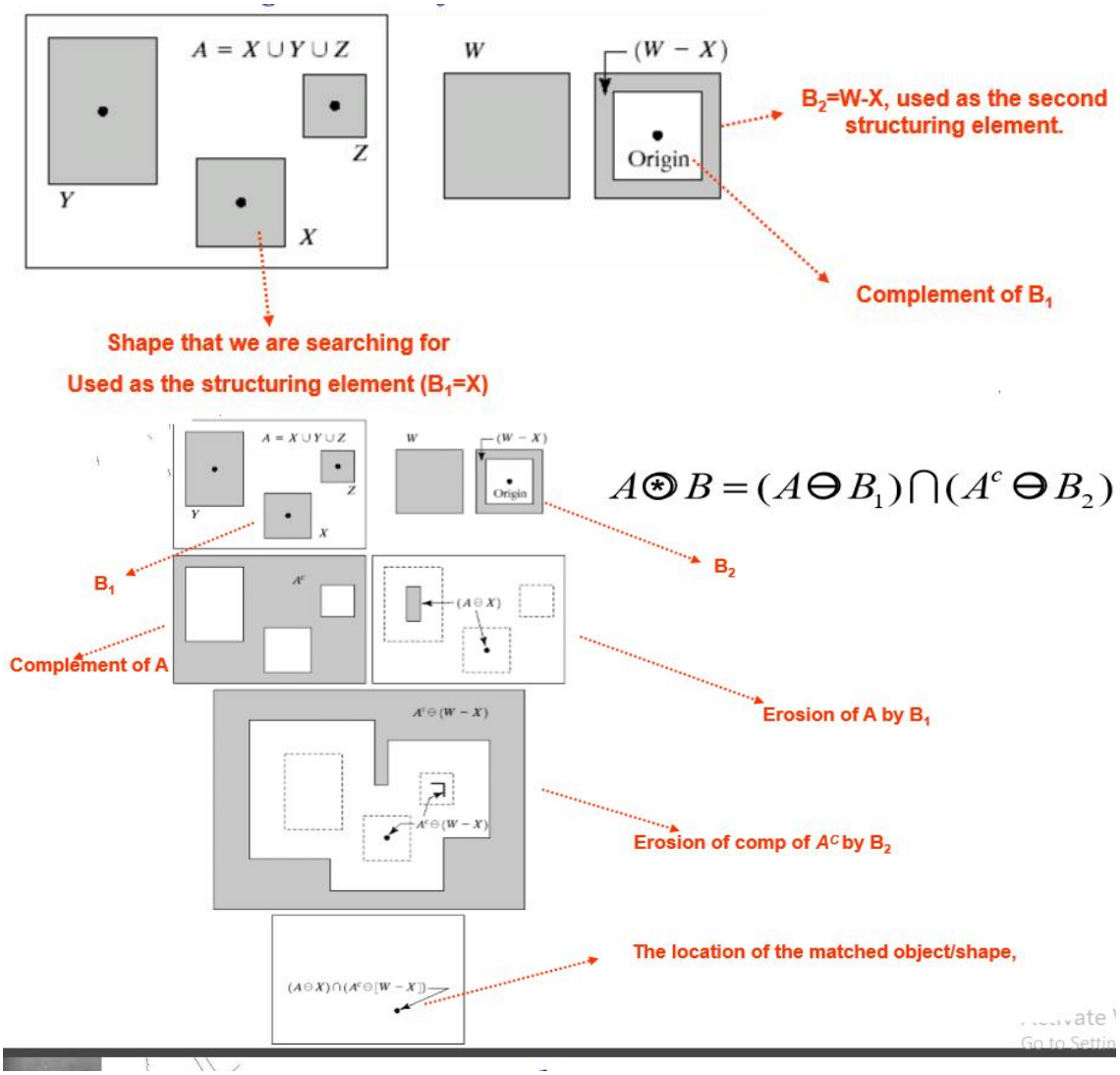
FIGURE 9.11
(a) Noisy image.
(c) Eroded image.
(d) Opening of A.
(d) Dilation of the opening.
(e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

Hit-or-Miss Transform (Template Matching):

- Hit-or-miss transform can be used for shape detection/ Template matching.
- Uses the morphological erosion operator and a pair of disjoint SEs
- First SE fits in the foreground of input image.
- Second SE misses it completely.
- The pair of two SEs is called *composite structuring* element.
- Given the shape as the structuring element B_1 the Hit-or-miss transform is defined by:

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

- Where B_2 =W-X and B_1=X. W is the window enclosing B_1. Windowing is used to isolate the structuring element/object.



- Three disjoint shapes denoted C, D, and E

$$A = C \cup D \cup E$$

- Objective: To find the location/origin of one of the shapes, say D.
- Origin/location of each shape given by its center of gravity.
- Let D be enclosed by a small window W.
- Local background of D defined by the set di□erence (W −D). Note that D and W −D provide us with the two disjoint ses D∩(W −D) = ∅
  1. Compute A^c
  2. Compute $A \ominus D$
  3. Compute $A^c \ominus (W - D)$

- Set of locations where D exactly fits inside A is $(A \ominus D) \cap (A^c \ominus (W - D))$

  The exact location of D

- If B is the set composed of D and its background, the match of B in A is given by

$$A \circledast B = (A \ominus D) \cap [A^c \ominus (W - D)]$$

- The above can be generalized to the composite se being defined by B = (B_1,B_2) leading to

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

- B_1 is the set formed from elements of B associated with the object; B_1 = D and B_2 = (W −D)
- A point z in universe A belongs to the output if (B_1)z fits in A (hit) and (B_2)z misses A

Some basic morphological algorithms:

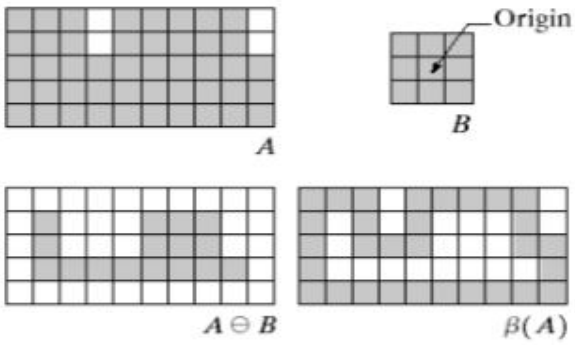
- These algorithms are useful in extracting image components for representation and description of shape.

Boundary extraction:

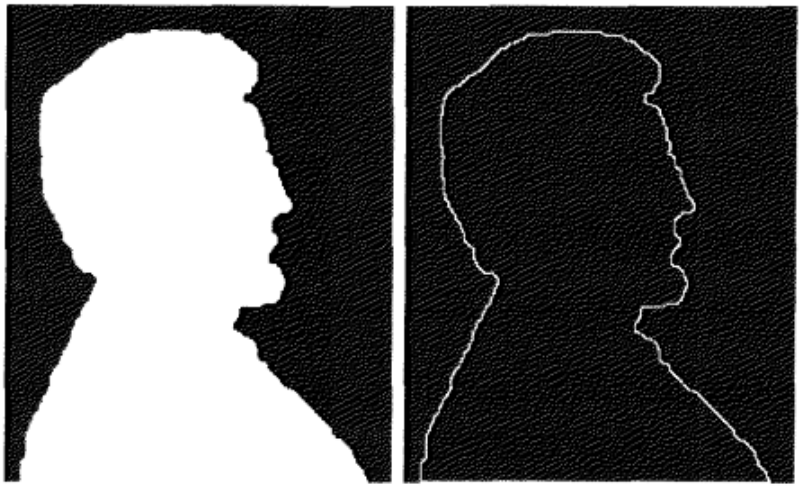
- Boundary of a set A is Denoted by β(A), can be obtained by first eroding A by a suitable structuring element B and computing set difference between A and its erosion.

$$\beta(A) = A - (A \ominus B)$$

Ex1: 3*3 structuring element is used for boundary extraction



Ex 2: same 3*3 structuring element is used in this example also



a b

FIGURE 9.14
(a) A simple binary image, with 1's represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

- **Note**: That thicker boundary can be obtained by increasing the size of structuring element.

Region Filling/Hole filling:

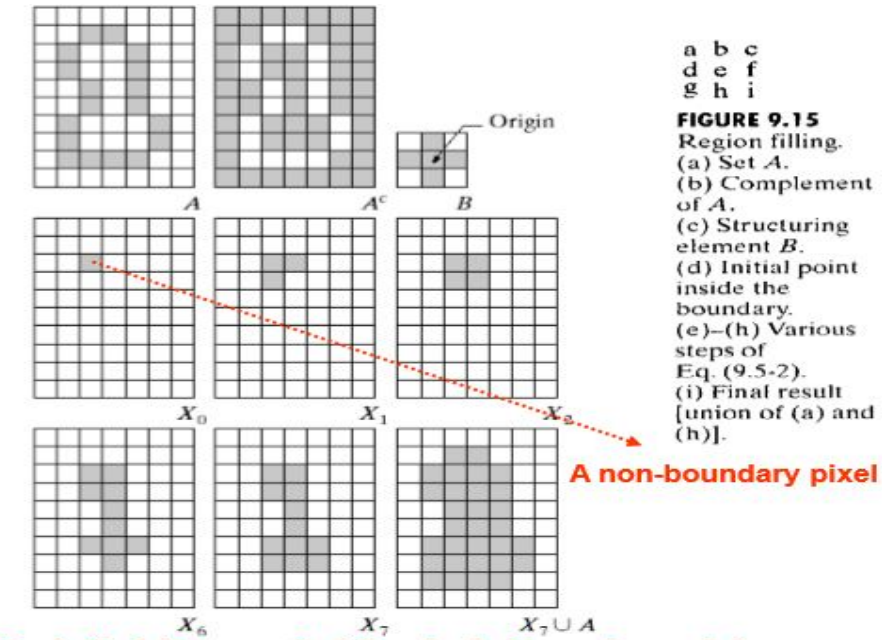
- We develop a simple algorithm for region filling based on set dilations, complementation and intersection.
- Region filling can be performed by using the following definition. Given a symmetric structuring element B, one of the non-boundary pixels (Xk) is consecutively diluted and its intersection with the complement of A is taken as follows:

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3,\ldots$$

$$terminates\ when\ X_k = X_{k-1}$$

$$X_0 = 1\ (inner\ pixel)$$

- Let A be a set whose elements are 8-connected boundaries, each boundary enclosing a background (hole).Given a point in each hole, we want to fill all holes.
- Start by forming an array X_0 of 0s of the same size as A. The locations in X_0 corresponding to the given point in each hole are set to 1.Let B be a symmetric SE with 4-connected neighbours to the origin as shown in the following figure.
- Where X_0=P, and B is the symmetric structuring element. The algorithm terminates at iteration step k if X_k=X_{k-1}. The set union of X_k and A contains the filled set and its boundary.
- The dilation process would fill the entire area if left unchecked. However the intersection at each step with A^c limits the result to inside the region of interest it also called as *conditioned dilation*.

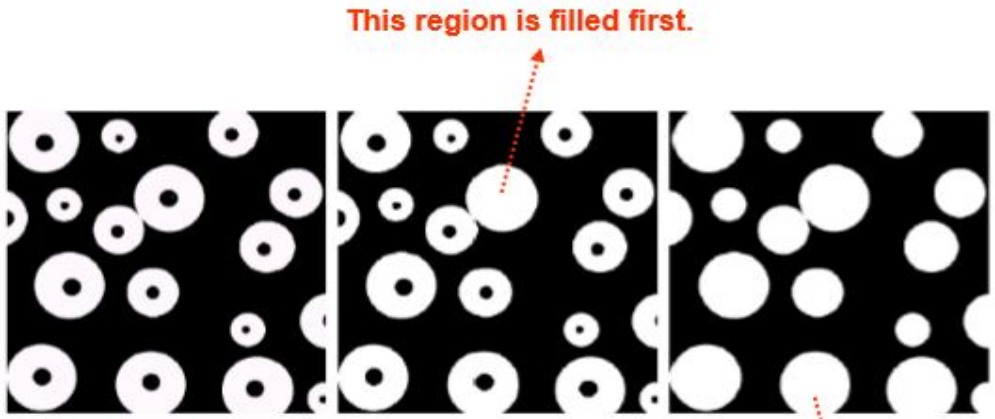


a b c
d e f
g h i

FIGURE 9.15 Region filling. (a) Set A. (b) Complement of A. (c) Structuring element B. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

Ex 1: X_0=1 (Assume that the shaded boundary points are 1 and the white pixels are 0)

EX2:



a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the region-filling algorithm). (b) Result of filling that region (c) Result of filling all regions.

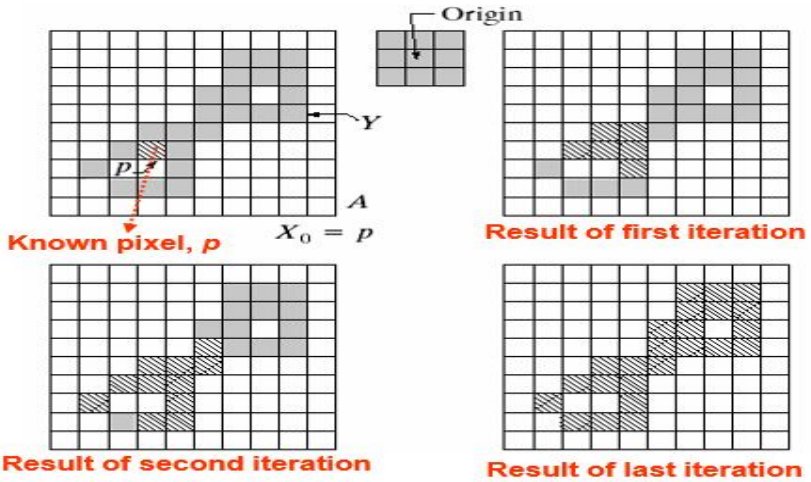
Extraction of connected components:

- Let Y represent a connected component contained in set A and assume that a point p of Y is known. Then the following iterative expression yields all the elements of Y:

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots$$

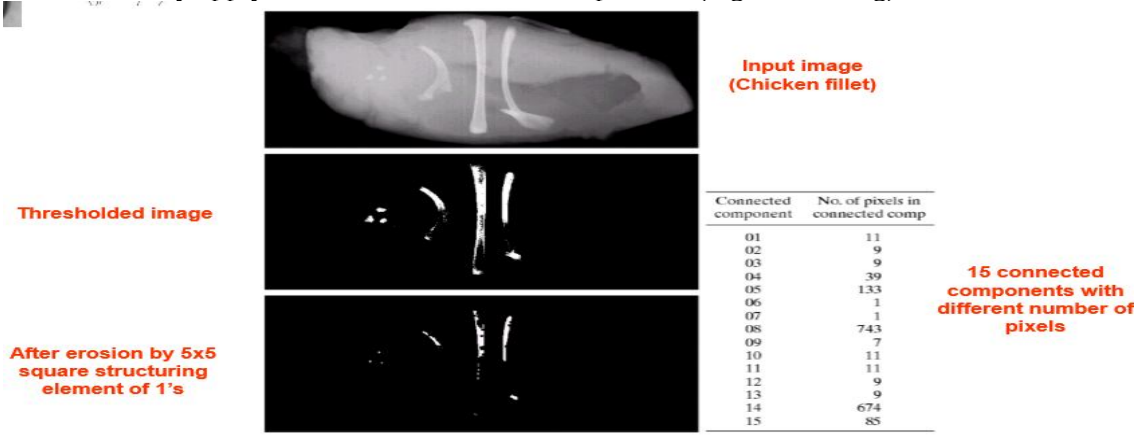
- Where X_0=p, and B is a suitable structuring element, as shown in the following figure. If X_k=X_{k+1} the algorithm has converged and we let Y=X_k.

- The connected components finding equation and region filling equation both are similar. The only difference is the use of A instead of its compliment (A^c). This difference arise because all the elements sought (that is, the element of the connected component) are labelled 1.
- The intersection with A at each iterative step eliminates dilations centered on elements labelled 0.



Example:

- X-ray image of chicken breast with bone fragments
- Objects of "significant size" can be selected by applying erosion to the thresholded image
- We may apply labels to the extracted components (region labeling)



Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

Convex hull:

- A set A is said to be convex set a Straight line segment joining any two points in A lies entirely within A.
- Convex hull H of an arbitrary set of points S is the smallest convex set containing S – Set di□erence H −S is called the convex deficiency of S.

- Convex hull and convex deficiency are useful to describe objects.
- Algorithm to compute convex hull C(A) of a set A:
- Let B^i, i = 1,2,3,4 represent the four structuring elements in the figure·
- B^i is a clockwise rotation of B^{i-1} by 90◦.
- Implement the equation $X_k^i = (X_{k-1} \circledast B^i) \cup A$ i = 1,2,3,4 and k = 1,2,3,... with $X^i{}_0$ = A.
- Apply hit-or-miss with B^1 till $X_k == X_{k-1}$, then, with B^2 over original A, B^3, and B^4.
- Procedure converges when $X^i{}_k = X^i{}_{k-1}$ and we let $D^i = X^i{}_k$.
- Convex hull of A is given by

$$C(A) = \bigcup_{i=1}^{4} D^i$$

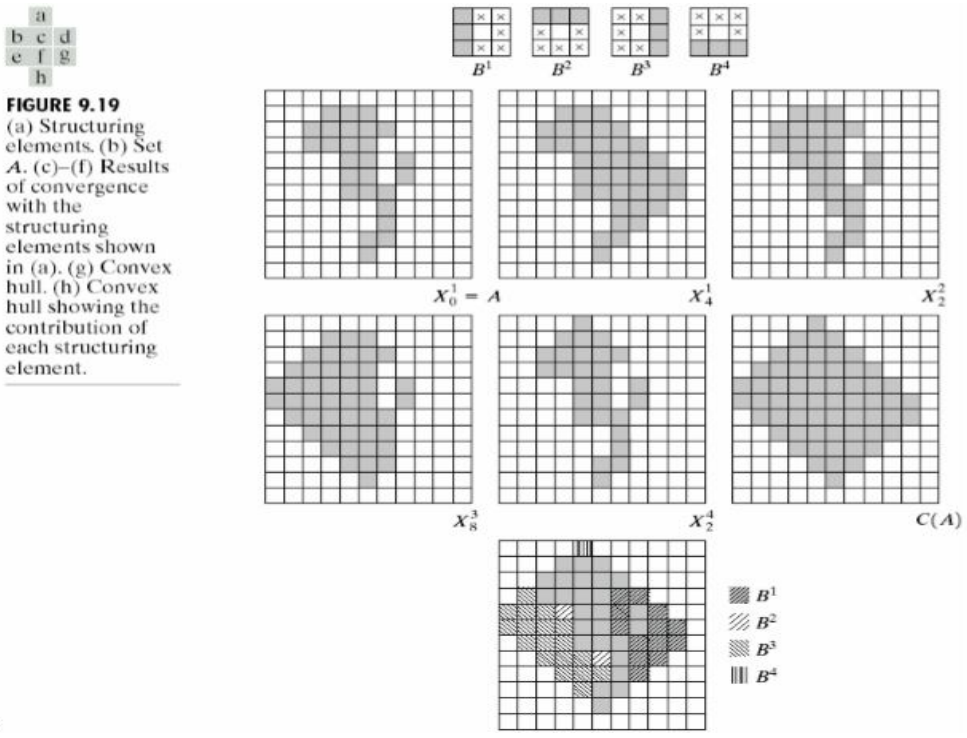


FIGURE 9.19 (a) Structuring elements. (b) Set *A*. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.

c

Thinning:

- Thinning of a set A by structuring element B is denoted by A⊗B, and can be find in terms of hit-or-miss transform:

$$\begin{aligned} A \otimes B &= A - (A \circledast B) \\ &= A \cap (A \circledast B)^c \end{aligned}$$

- We are interested only in pattern matching with the structuring elements, so no background operation required in hit-or-miss transform. A more useful expression for

thinning A symmetrically based on a sequence of structuring elements: {B} = $\{B^1, B^2, ..., B^n\}$ where B^i is a rotated version of B^{i-1}.using this concept, we now define thinning by a sequence of structuring elements as

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$$

- The process is to thin A by one pass with B^1, then the result with one pass of B^2, and so on, until A is thinned with one pass of B^n. The entire process is repeated until no further change s occur.

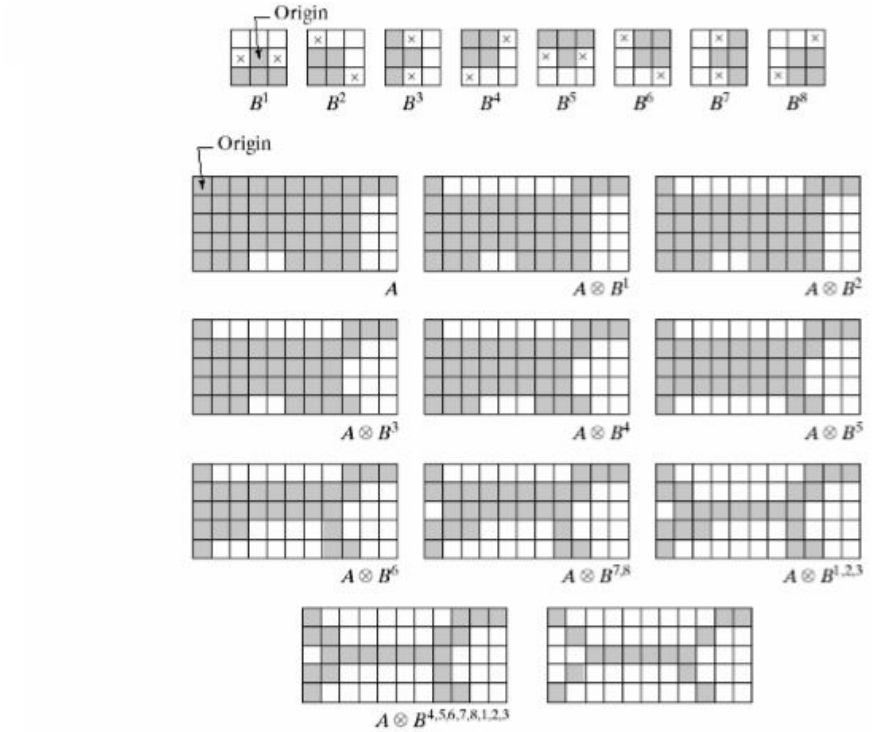


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set *A*. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to *m*-connectivity.

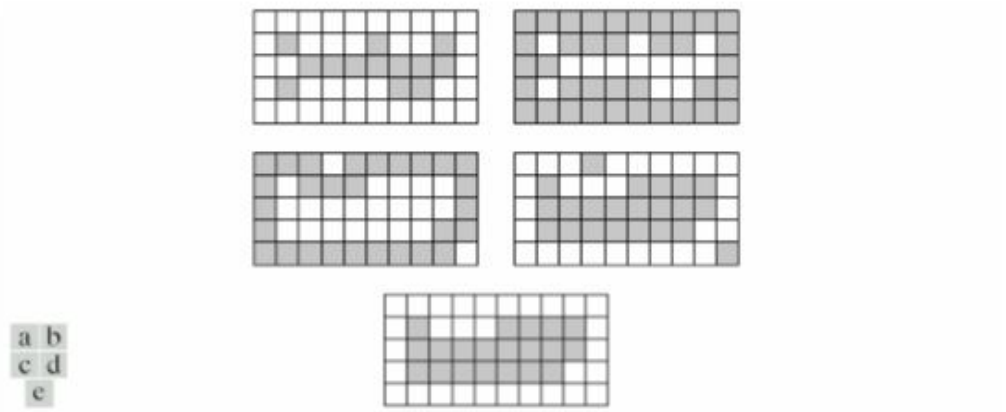
Thickening:

- Thickening is morphological dual of thickening and is defined by the expression:

$$A \odot B = A \cup (A \circledast B)$$

- Where B is the structuring element suitable for thickening. As in thinning thickening can be defined as sequential operations.

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n)$$



a b
c d
e

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.

- 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 as illustrated in the above figure 9.22.

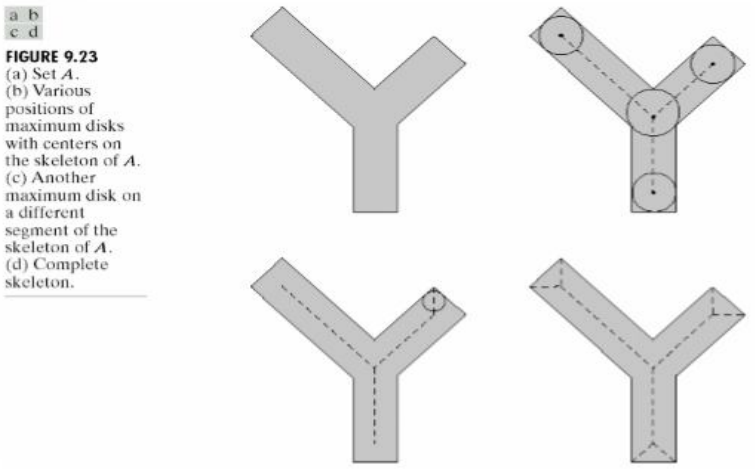
Skeletons:

- As shown in the following figure 9.23,the notation of a Skeleton, S(A), of a set A is intuitively simple. We deduce from this figure that
  1. If z is a point of S(A) and (D)z is the largest disk centered at z and contained in A, one cannot find a larger disk (not necessarily centered at z) containing (D)z and included in A. The disk (D)z is called a maximum disk.
  2. Disk (D)z touches the boundary of A at two or more di□erent places.
- Skeleton can be expressed in terms of erosions and openings. That is

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

where

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$



a b
c d

FIGURE 9.23
(a) Set A.
(b) Various positions of maximum disks with centers on the skeleton of A.
(c) Another maximum disk on a different segment of the skeleton of A.
(d) Complete skeleton.

- Where B is a 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

$$K = \max\{k \mid (A \ominus kB) \neq \emptyset\}$$

- S(A) can be obtained as the union of skeleton subsets $S_k(A)$. Also it can be shown that A can be reconstructed from the subsets using the equation

$$\bigcup_{k=0}^{K} (S_k(A) \oplus kB)$$

- Where $(S_k(A) \oplus_k B)$ denotes k successive dilations of $S_k(A)$. That is

$$(S_k(A) \oplus_k B) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B)$$

Example:

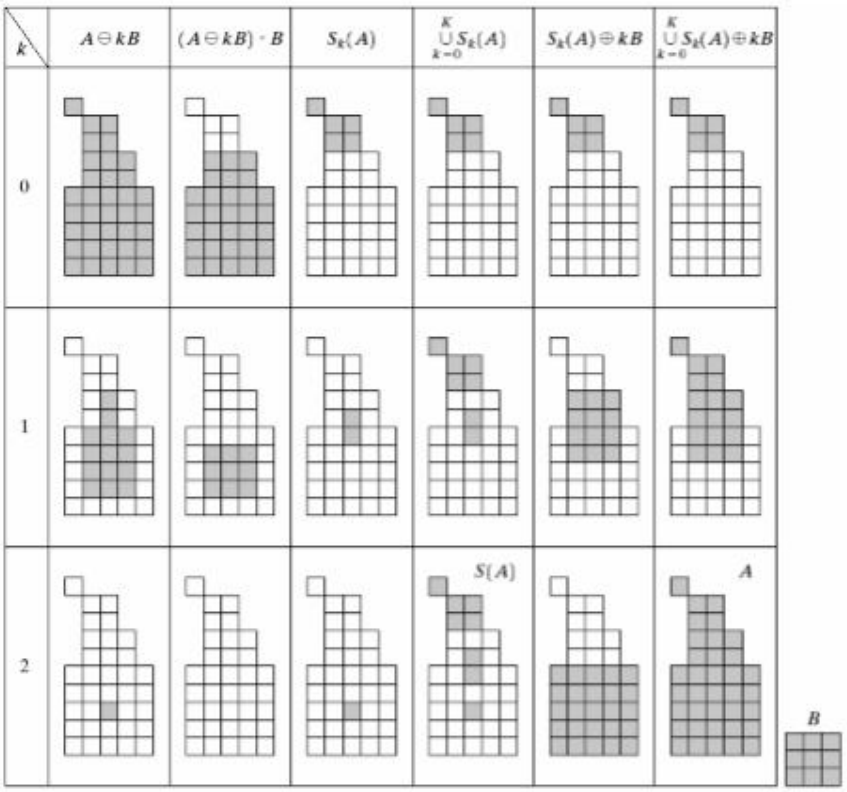


FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

Pruning:

- Pruning methods are an essential complement to thinning and skeletonising algorithms because these procedures tend to leave parasitic components that need to be "cleaned up" by post processing.

A common approach in the automated reorganization of hand-printed characters is as follows.

- Analyze the shape of the skeleton of each character.
- Skeletons characterized by "spurs" or parasitic components.
- Spurs caused during erosion by non-uniformities in the strokes.
- Assume that the length of a spur or parasitic component does not exceed a specific number of pixels.

Figure 9.25 – Skeleton of hand-printed "a"

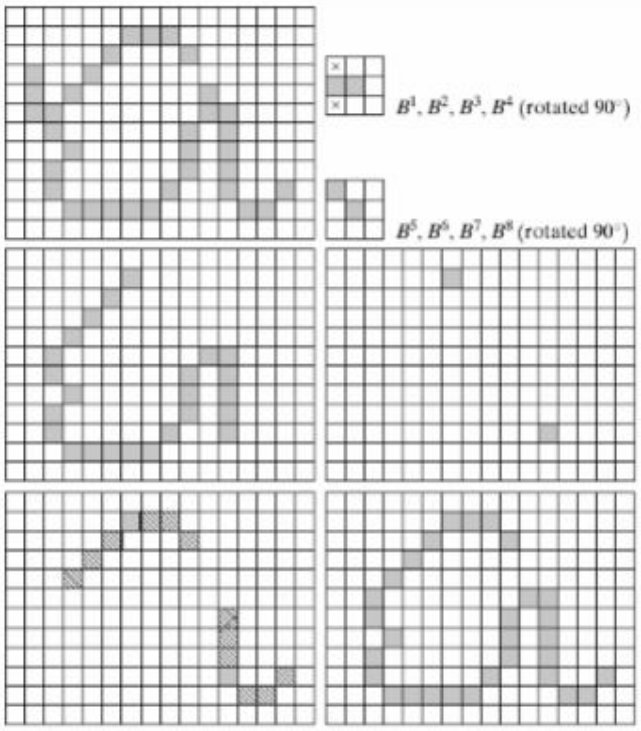
- Suppress a parasitic branch by successively eliminating its end point.
- Assumption: Any branch with ≤ 3 pixels will be removed.
- Achieved with thinning of an input set A with a sequence of SEs designed to detect only end points $X1 = A \otimes \{B\}$

Figure 9.25d – Result of applying the above thinning three times.

- Restore the character to its original form with the parasitic branches removed.
- Form a set X2 containing all end points in X1

$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$$

Dilate end points three times using set A as delimiter $X_3 = (X_2 \oplus H) \cap A$ where H is a 3×3 se of 1s and intersection with A is applied after each step.
The final result comes from $X_4 = X_1 \cup X_3$



a b c d e f g

FIGURE 9.25
(a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilation of end points conditioned on (a). (g) Pruned image.

UNIT-V
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. _____is erosion followed by dilation.
2. _____ is dilation followed by erosion.
3. The opening of set A by structuring element B is_____
4. _____ Smoothes the contour of an object, breaks narrow isthmuses, and eliminate thin protrusions. [ ]

A) Opening B) Closing C) Dilation D) Erosion

5. Closing of set A by structuring element B is $A \cdot B = (A \oplus B) \ominus B$.

[True/False]

6. _____ eliminates small holes and gaps in the contour.

A) Opening B) Closing C) Dilation D) Erosion

7. The morphological hit-or-miss transform is a basic tool for shape detection. [True/False]
8. The boundary of a set A with the suitable structuring element B is given as_____
9. We use morphological algorithms for_____

A) Extracting boundaries C) Connected components
B) Convex hull, skeleton of the region D) All the above

10. The convex hull H of an arbitrary set S is the smallest convex set containing S. The set difference H-S is called convex deficiency of S.

[True/False]

11. Region filling is based on_____

A) Only the Set dilation C) Only the Set complementation
B) Only the Set intersection D) All the three

12. Skeleton of A can be formulated in terms of erosion and opening

[True/false]

13. Erosion of A by B, followed by a dilation of the result by B is_____

A) Opening A by B C. Closing A by B [ ]
B) Opening B by A D. Closing B by A

14. Dilation of A by B, followed by the erosion of the result by B is _____

A) Opening A by B C. Closing A by B [ ]
B) Opening B by A D. Closing B by A

15. First eroding A by suitable structuring element B and then performing the set difference between A and its erosion is_____ operation

A) Boundary Extraction C. Region Filling [ ]
B) Both D. None

16. Consider the following expressionand analyse the use of that expression. .

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$$

Where A is one set, B is a structuring element {B}={B^1,B^2,B^3,.....,B^n) and

$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c.$$

A) Thinning B) Thickening C) Skeleton D) none[ ]

17. Consider the following expression and analyse the use of that expression.

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n).$$ [ ]

Where A is one set, B is a structuring element {B}={B^1,B^2,B^3,.....,B^n) and

$$A \odot B = A \cup (A \circledast B)$$

A) Thinning B) Thickening C) Skeleton D) none

18. If z is point of skeleton S(A) and $(D)_z$ is the largest disk centred at z and contained in A, one cannot find largest disk containing $(D)_z$ and include in A. The disk $(D)_z$ is called_____

[ ]

A) mask disk B) large disk C) maximum disk D) grater disk

19 The maximum disk $(D)_z$ can touches the boundaries of at how many places? [ ]

A) 1 B) 0 C) 2 D) More than 2

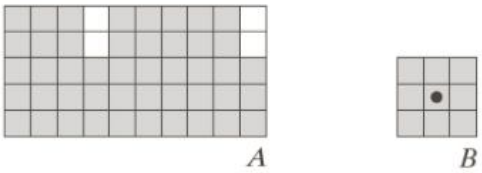
20 Set A is said to be _____ if the straight line segment joining any two points in A lies entirely within A. [ ]

A) Convex B) Concave C) Both D) None

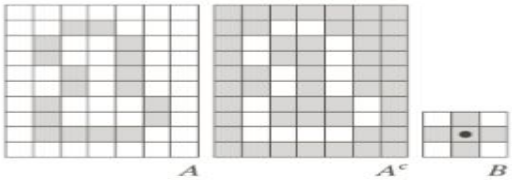
SECTION-B

SUBJECTIVE QUESTIONS

1. Explain Basic concepts from set theory on binary images in morphological image processing.
2. Explain Basic concepts from logical operations involving binary images in morphological image processing.
3. Describe Dilation and Erosion morphological transformations on a binary image.
4. Explain the opening operation in image morphology with examples?
5. Explain the closing operation in image morphology with examples?
6. Write about the importance of Hit-or-Miss morphological transformation operation on a digital binary image.
7. Explain boundary extraction and region filling process
8. Write the procedure for extraction of connected components
9. Explain convex hull
10. Prove that Dilation and erosion are duels of each other with respect to set complementation and reflection, that is $(A \ominus B)^c = A^c \oplus \hat{B}$.
11. Prove that multiple openings or closings of a set have no effect after the operator has been applied once.
12. Extract Boundary of set A with the structuring element B given below by using morphological boundary extraction process.



13. Fill the region of set A with the structuring element B given below by using morphological region filling process.



14. Compose Thinning in terms of the hit-or-miss transform
15. Prove that Thickening is morphological dual of thinning.
16. Formulate the skeleton of A in terms of erosion and opening.
17. Develop a morphological solution for Pruning.

UNIT - VI
Image Segmentation

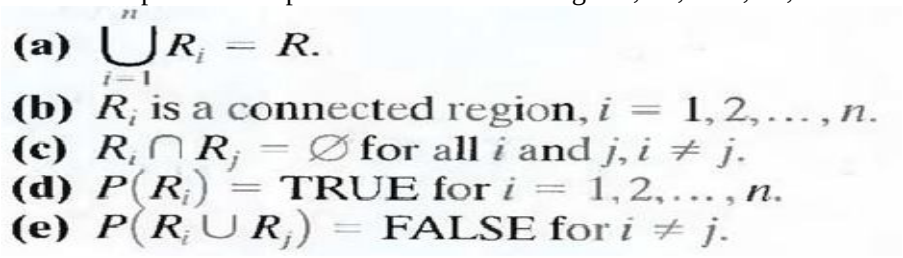
Detection of discontinuous, edge linking and boundary detection, thresholding, region–based seg

Image Segmentation:

- Segmentation is to subdivide an image into its component regions or objects.
- Segmentation should stop when the objects of interest in an application have been isolated.

- Segmentation algorithms generally are based on one of 2 basis properties of intensity values: **Discontinuity**: to partition an image based on sharp changes in intensity

  Similarity: to partition an image into regions that are similar according to a set of predefined criteria.
- A more formal definition – Let R represent the entire image region. We may view segmentation as a process that partitions R into n sub regions, R_1, R_2..., R_n, such that



- Here, P (R_i) is a logical predicate defined over the points in set R_i and *Ø`* is the null set. Condition
- (a) Indicates that the segmentation must be complete; that is, every pixel must be in a region. Condition
- (b) Requires that points in a region must be connected in some predefined sense. Condition
- (c) Indicates that the regions must be disjoint. Condition
- (d) Deals with the properties that must be satisfied by the pixels in a segmented region—for example P (R_i) = TRUE if all pixels in R_i, have the same gray level.
- Finally, condition (e) indicates that regions R_i and R_j are different in the sense of predicate P.

Detection of discontinuities:

- In this we present several techniques for detecting the three basic types of gray level discontinuities in a digital image: points, lines and edges.

- The most common way to look for discontinuities is to run a mask through the image. The 3X3 mask shown in following figure.

- This procedure involves computing the sum of products of the coefficients with the gray levels contained in the region encompassed by the mask.

- The response of the mask at any point in the image is given by

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

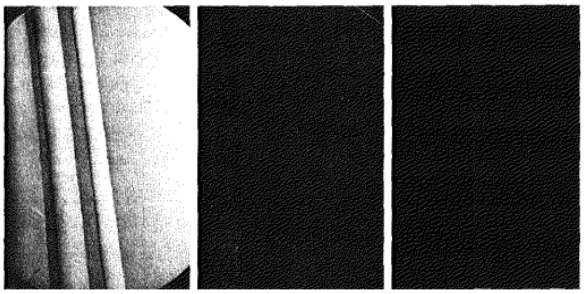
- Where Z_i is the gray level of the pixel associated with mask coefficient W_i.
- The response of the mask is defined with respect to its centre location

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

Point Detection:

- The detection of isolated points in an image is straight forward in principle. Using the mask shown in figure:6.2(a),we say that a point has been detected at the location on which the mask is cantered if |R|>=T
- Where T is a non negative threshold and R is given by equation 6.1-1.Basically this formulation measures the weighted differences between the centre point and its neighbours .The idea is an isolated point will be quite different from its surroundings and thus be wisely detectable by this type of mask.0

−1	−1	−1
−1	8	−1
−1	−1	−1



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

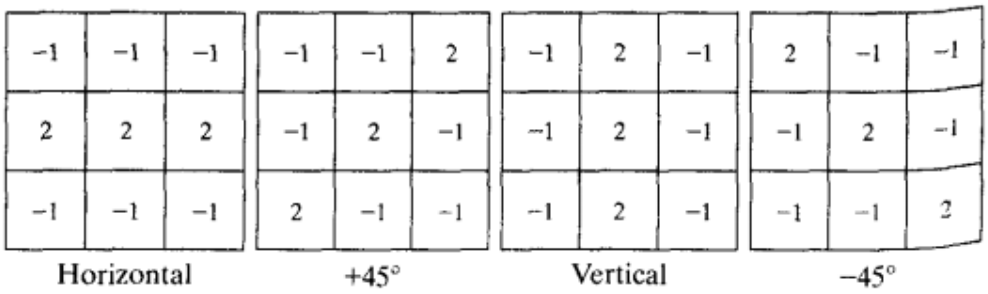
Line Detection:

- The next level of complexity is line detection. consider the masks shown in figure 6.3.If the first mask were moved around an image ,it would respond more strongly to lines(1 pixel thick)oriented

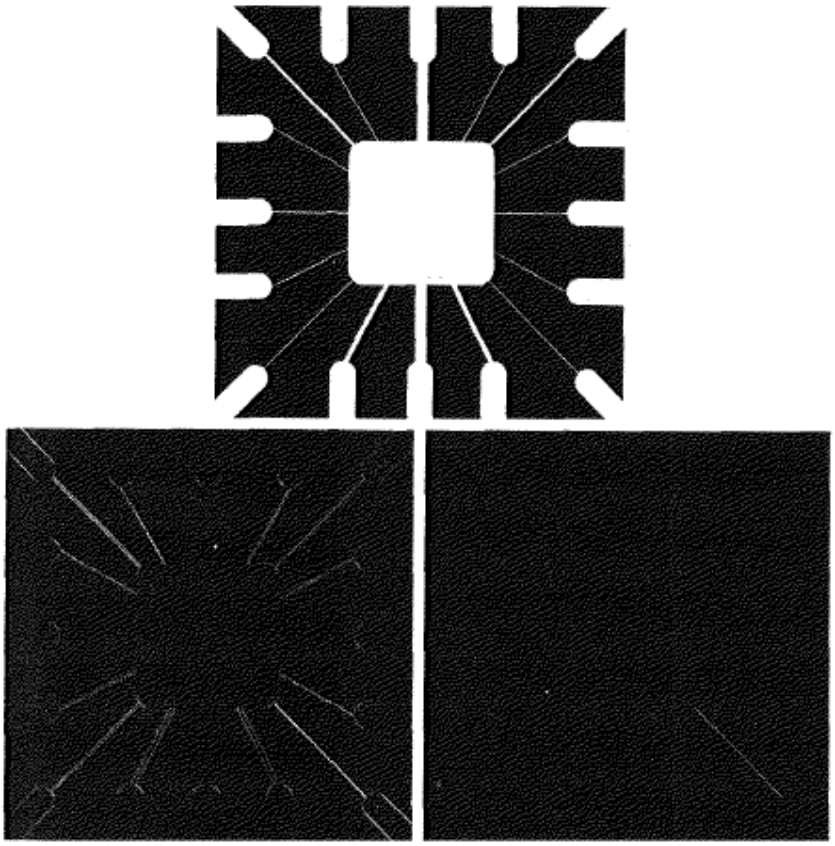
horizontally .With a constant background ,the maximum response would result when he line passed through the middle row of the mask.

- This is easily verified by sketching a simple array of 1's with a line of a different gray level (say 5s)running horizontally through the array .A similar experiment would reveal that the second mask in following figure response best to lines oriented at +45 degrees.
- The third mask to the vertical lines, and the fourth mask to lines in the -45 degrees direction. These directions can be establish also by noting that the preferred direction of each mask is waited with a larger co-efficient (i.e. 2) than other possible directions.
- Note that the coefficients in each mask sum to 0, indicating a zero response from the masks in areas of constant gray level.

e



- Let R1,R2,R3, and R4 denote the responses of the masks in the above figure, from left to right, suppose that 4 masks are run individually through an image. If at a certain point in image, $|R_i| > |R_j|$, for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i.
- For example, if at a point in the image $|R_1| > |R_j|$ for j=2,3,4 that particular point is said to be more likely associated with a horizontal line.
- Alternatively, we may be interested in detecting lines in a specified direction. In this case, we would use the mask associated with that direction and threshold its output.
- In the following figure 10.4(a) shows a digitized (binary) portion of a wire – bond mask for an electronic circuit. Suppose that we are interested in finding all the lines that are one pixel thick and are oriented at -45^0. For this purpose we use the last mask shown in the above figure. The absolute value of the result is shown in the figure 10.4(b).
- Note that all vertical and horizontal components of the image were eliminated, and that the components of the original image that tend towards a -45^0 direction produced the strongest response. In order to determine which lines best fit the mask, we simply threshold this image. The result of using a threshold equal to the maximum value in the image is shown in figure 10.4(c).



a
b c

FIGURE 10.4
Illustration of line detection.
(a) Binary wire-bond mask.
(b) Absolute value of result after processing with −45° line detector.
(c) Result of thresholding image (b).

Edge detection:

- Intuitively, an edge is a set of connected pixels that lie on the boundary between two regions. Fundamentally, an edge is a "local" concept whereas a region boundary, owing to the way it is defined, is a more global idea. A reasonable definition of "edge" requires the ability to measure gray-level transitions in a meaningful way. We start by modeling an edge intuitively. This will lead us to formalism in which "meaningful" transitions in gray levels can be measured.
- Intuitively, an ideal edge has the properties of the model shown in Fig. 2.1(a). An ideal edge according to this model is a set of connected pixels (in the vertical direction here), each of which is located at an orthogonal step transition in gray level (as shown by the horizontal profile in the figure).
- In practice, optics, sampling, and other image acquisition imperfections yield edges that are blurred, with the degree of blurring being determined by factors such as the quality of the image acquisition system, the sampling rate, and illumination conditions under which the image is acquired. As a result, edges are more closely modeled as having a "ramp like" profile, such as the one shown in Fig.2.1 (b).

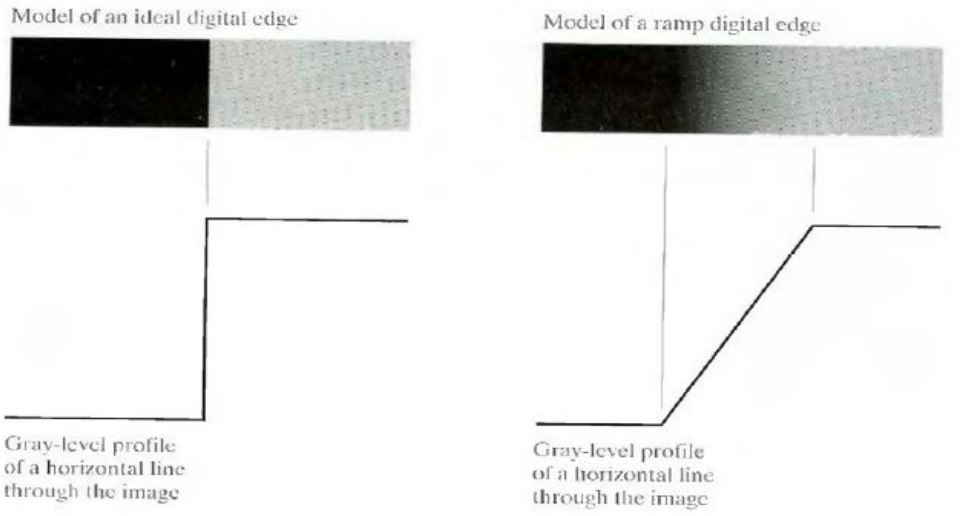


Fig.2.1 (a) Model of an ideal digital edge (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

- The slope of the ramp is inversely proportional to the degree of blurring in the edge. In this model, we no longer have a thin (one pixel thick) path. Instead, an edge point now is any point contained in the ramp, and an edge would then be a set of such points that are connected. The "thickness" of the edge is determined by the length of the ramp, as it transitions from an initial to a final gray level. This length is determined by the slope, which, in turn, is determined by the degree of blurring. This makes sense: Blurred edges lend to be thick and sharp edges tend to be thin. Figure 2.2(a) shows the image from which the close-up in Fig. 2.1(b) was extracted. Figure 2.2(b) shows a horizontal gray-level profile of the edge between the two regions. This figure also shows the first and second derivatives of the gray-level profile. The first derivative is positive at the points of transition into and out of the ramp as we move from left to right along the profile; it is constant for points in the ramp; and is zero in areas of constant gray level. The second derivative is positive at the transition associated with the dark side of the edge, negative at the transition associated with the light side of the edge, and zero along the ramp and in areas of constant gray level. The signs of the derivatives in Fig. 2.2(b) would be reversed for an edge that transitions from light to dark.
- We conclude from these observations that the magnitude of the first derivative can be used to detect the presence of an edge at a point in an image (i.e. to determine if a point is on a ramp). Similarly, the sign of the second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge. We note two additional properties of the second derivative around an edge: A) It produces two values for every edge in an image (an undesirable feature); and B) an imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge. This zero-crossing property of the second derivative is quite useful for locating the centers of thick edges.

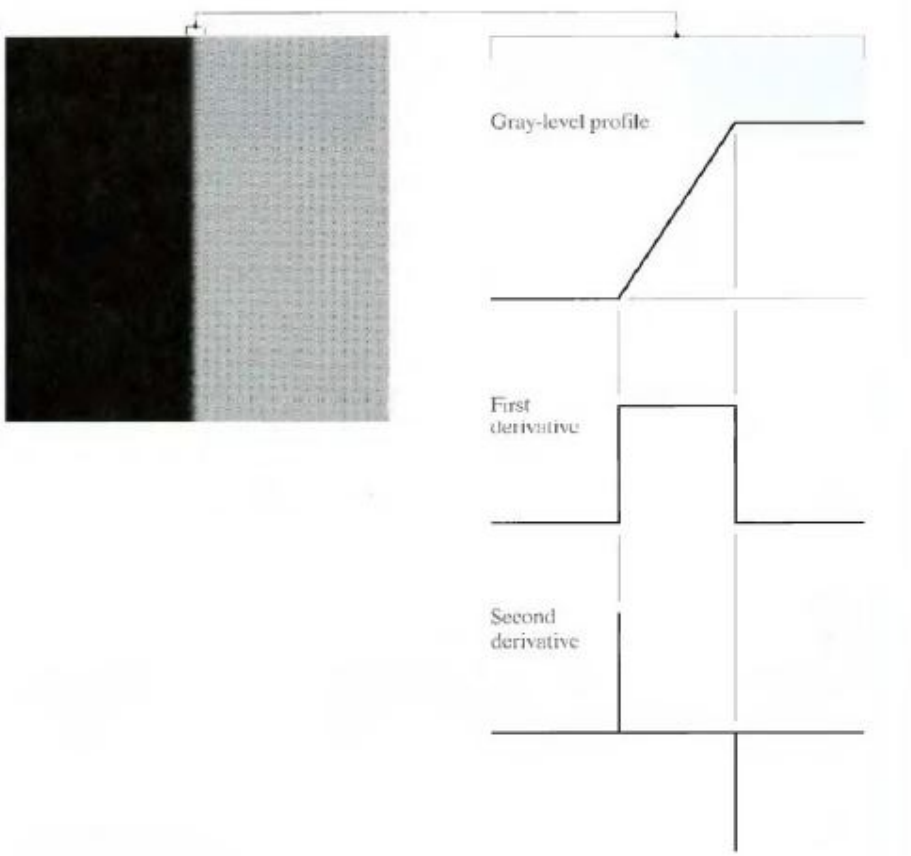


Fig.2.2 (a) Two regions separated by a vertical edge (b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.

Gradient operators:

- First-order derivatives of a digital image are based on various approximations of the 2-D gradient. The gradient of an image f (x, y) at location (x, y) is defined as the vector

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

- It is well known from vector analysis that the gradient vector points in the direction of maximum rate of change of f at coordinates (x, y). An important quantity in edge detection is the magnitude of this vector, denoted by $\nabla \mathbf{f}$, where

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

- This quantity gives the maximum rate of increase of f (x, y) per unit distance in the direction of $\nabla \mathbf{f}$. It is a common (although not strictly correct) practice to refer to $\nabla \mathbf{f}$ also as the gradient. The direction of the gradient vector also is an important quantity. Let α (x, y) represent the direction angle of the vector $\nabla \mathbf{f}$ at (x, y). Then, from vector analysis,

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

- Where the angle is measured with respect to the x-axis. The direction of an edge at (x, y) is perpendicular to the direction of the gradient vector at that point. Computation of the gradient

of an image is based on obtaining the partial derivatives ∂f/∂x and ∂f/∂y at every pixel location. Let the 3x3 area shown in Fig. 1.1 (a) represent the gray levels in a neighborhood of an image.One of the simplest ways to implement a first-order partial derivative at point z_5 is to use the following Roberts cross-gradient operators:

$$G_x = (z_9 - z_5)$$

and

$$G_y = (z_8 - z_6).$$

- These derivatives can be implemented for an entire image by using the masks shown in Fig. 1.1(b). Masks of size 2 X 2 are awkward to implement because they do not have a clear center. An approach using masks of size 3 X 3 is given by

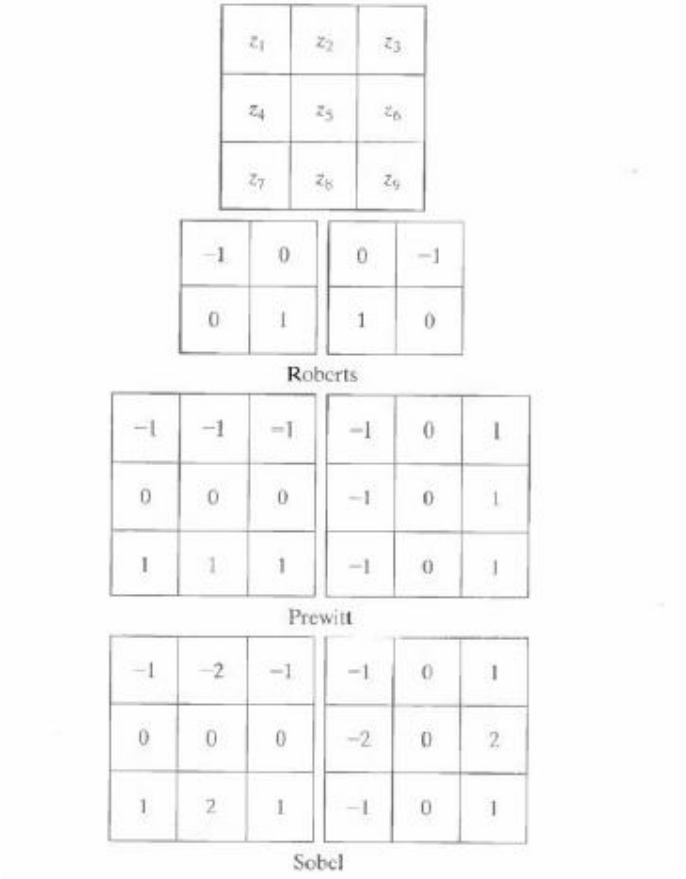


Fig.1.1 A 3 X 3 region of an image (the z's are gray-level values) and various masks used to compute the gradient at point labeled z_5.

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

and

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

- A weight value of 2 is used to achieve some smoothing by giving more importance to the center point. Figures 1.1(f) and (g), called the Sobel operators, and are used to implement these two equations. The Prewitt and Sobel operators are among the most used in practice for

computing digital gradients. The Prewitt masks are simpler to implement than the Sobel masks, but the latter have slightly superior noise-suppression characteristics, an important issue when dealing with derivatives. Note that the coefficients in all the masks shown in Fig. 1.1 sum to 0, indicating that they give a response of 0 in areas of constant gray level, as expected of a derivative operator

- The masks just discussed are used to obtain the gradient components G_x and G_y. Computation of the gradient requires that these two components be combined. However, this implementation is not always desirable because of the computational burden required by squares and square roots. An approach used frequently is to approximate the gradient by absolute values:

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

- This equation is much more attractive computationally, and it still preserves relative changes in gray levels. However, this is not an issue when masks such as the Prewitt and Sobel masks are used to compute Gx and Gy.
- It is possible to modify the 3 X 3 masks in Fig. 1.1 so that they have their strongest responses along the diagonal directions. The two additional Prewitt and Sobel masks for detecting discontinuities in the diagonal directions are shown in Fig. 1.2.

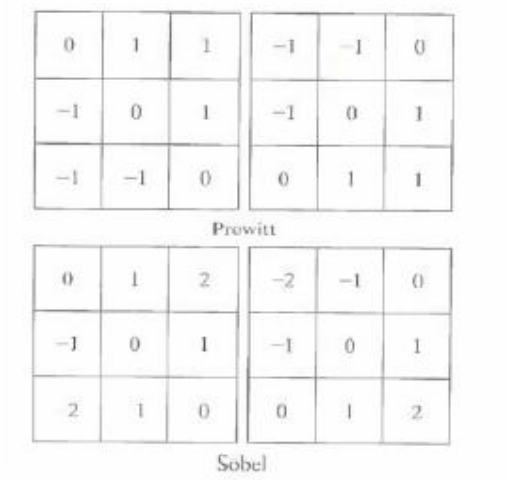


Fig.1.2 Prewitt and Sobel masks for detecting diagonal edges

The Laplacian:

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

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

For a 3 X 3 region, one of the two forms encountered most frequently in practice is

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

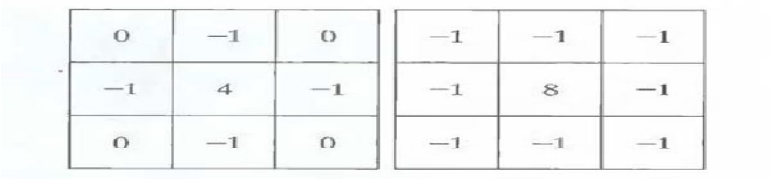


Fig.1.3 Laplacian masks used to implement Eqns. above.

- Where the z's are defined in Fig. 1.1(a). A digital approximation including the diagonal neighbors is given by

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9).$$

- Masks for implementing these two equations are shown in Fig. 1.3. We note from these masks that the implementations of Eqns. are isotropic for rotation increments of 90° and 45°, respectively.

Edge linking and Boundary detection:

- The different methods for edge linking are as follows
  (i) Local processing
  (ii) Global processing via the Hough Transform
  (iii) Global processing via graph-theoretic techniques.

i) Local Processing:

- One of the simplest approaches for linking edge points is to analyze the characteristics of pixels in a small neighborhood (say, 3 X 3 or 5 X 5) about every point (x, y) in an image that has been labeled an edge point. All points that are similar according to a set of predefined criteria are linked, forming an edge of pixels that share those criteria.
- The two principal properties used for establishing similarity of edge pixels in this kind of analysis are (1) the strength of the response of the gradient operator used to produce the edge pixel; and (2) the direction of the gradient vector. The first property is given by the value of $\nabla \mathbf{f}$.
- Thus an edge pixel with coordinates (x_o, y_o) in a predefined neighborhood of (x, y), is similar in magnitude to the pixel at (x, y) if

$$|\nabla f(x, y) - \nabla f(x_0, y_0)| \leq E$$

- The direction (angle) of the gradient vector is given by Eq. An edge pixel at (x_o, y_o) in the predefined neighborhood of (x, y) has an angle similar to the pixel at (x, y) if

$$|\alpha(x, y) - \alpha(x_0, y_0)| < A$$

- Where A is a nonnegative angle threshold. The direction of the edge at (x, y) is perpendicular to the direction of the gradient vector at that point. A point in the predefined neighborhood of (x, y) is linked to the pixel at (x, y) if both magnitude and direction criteria are satisfied. This process is repeated at every location in the image. A record must be kept of linked points as the center of the neighborhood is moved from pixel to pixel. A simple bookkeeping procedure is to assign a different gray level to each set of linked edge pixels.

ii) Global processing via the Hough Transform:

- In this process, points are linked by determining first if they lie on a curve of specified shape. We now consider global relationships between pixels. Given n points in an image, suppose that we want to find subsets of these points that lie on straight lines. One possible solution is to first find all lines determined by every pair of points and then find all subsets of points that are close to particular lines. The problem with this procedure is that it involves finding n(n - 1)/2 ~ n^2 lines and then performing (n)(n(n - l))/2 ~ n^3 comparisons of every point to all lines. This approach is computationally prohibitive in all but the most trivial applications.
- Hough [1962] proposed an alternative approach, commonly referred to as the Hough transform. Consider a point (x_i, y_i) and the general equation of a straight line in slope-intercept form, $y_i = a.x_i + b$. Infinitely many lines pass through (x_i, y_i) but they all satisfy the equation $y_i = a.x_i + b$ for varying values of a and b. However, writing this equation as $b = -a.x_i$

+ y_i, and considering the ab-plane (also called parameter space) yields the equation of a single line for a fixed pair (x_i, y_i). Furthermore, a second point (x_j, y_j) also has a line in parameter space associated with it, and this line intersects the line associated with (x_i, y_i) at (a', b'), where a' is the slope and b' the intercept of the line containing both (x_i, y_i) and (x_j, y_j) in the xy-plane. In fact, all points contained on this line have lines in parameter space that intersect at (a', b'). Figure 3.1 illustrates these concepts.

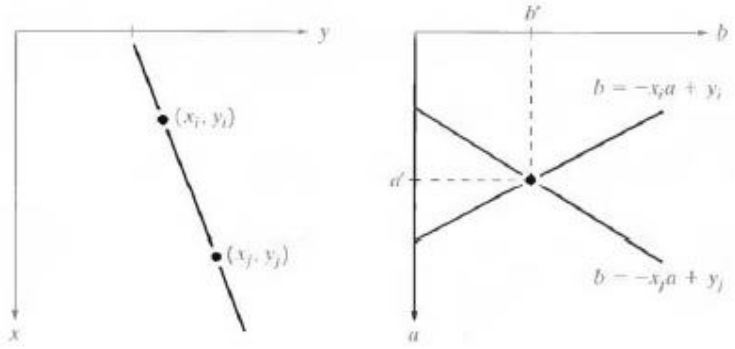


Fig.3.1 (a) xy-plane (b) Parameter space

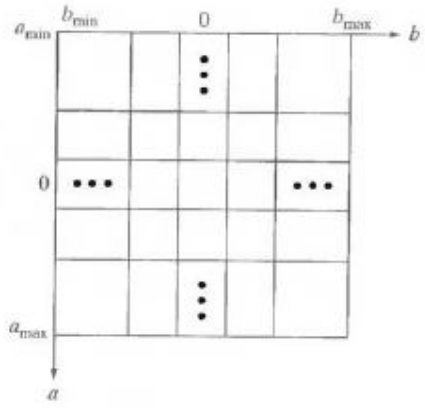


Fig.3.2 Subdivision of the parameter plane for use in the Hough transform

- The computational attractiveness of the Hough transform arises from subdividing the parameter space into so-called accumulator cells, as illustrated in Fig. 3.2, where (a_{max} , a_{min}) and (b_{max} , b_{min}), are the expected ranges of slope and intercept values. The cell at coordinates (i, j), with accumulator value A(i, j), corresponds to the square associated with parameter space coordinates (a_i , b_i).
- Initially, these cells are set to zero. Then, for every point (x_k, y_k) in the image plane, we let the parameter a equal each of the allowed subdivision values on the fl-axis and solve for the corresponding b using the equation b = - x_k a + y_k .The resulting b's are then rounded off to the nearest allowed value in the b-axis. If a choice of a_p results in solution b_q, we let A (p, q) = A (p, q) + 1. At the end of this procedure, a value of Q in A (i, j) corresponds to Q points in the xyplane lying on the line y = a_i x + b_j. The number of subdivisions in the ab-plane determines the accuracy of the co linearity of these points. Note that subdividing the a axis into K increments gives, for every point (x_k, y_k), K values of b corresponding to the K possible values of a. With n image points, this method involves nK computations. Thus the procedure just discussed is linear in n, and the product nK does not approach the number of computations discussed at the beginning unless K approaches or exceeds n.

- A problem with using the equation y = ax + b to represent a line is that the slope approaches infinity as the line approaches the vertical. One way around this difficulty is to use the normal representation of a line: $x\cos\theta + y\sin\theta = \rho$
- Figure 3.3(a) illustrates the geometrical interpretation of the parameters used. The use of this representation in constructing a table of accumulators is identical to the method discussed for the slope-intercept representation. Instead of straight lines, however, the loci are sinusoidal curves in the $\rho\theta$ -plane. As before, Q collinear points lying on a line $x\cos\theta_j + y\sin\theta_j = \rho$, yield Q sinusoidal curves that intersect at (p_i, θ_j) in the parameter space. Incrementing θ and solving for the corresponding p gives Q entries in accumulator A (i, j) associated with the cell determined by (p_i, θ_j). Figure 3.3 (b) illustrates the subdivision of the parameter space

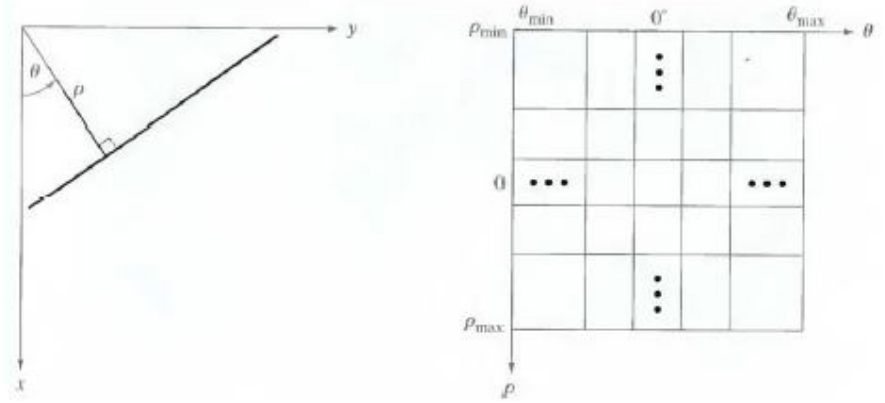


Fig.3.3 (a) Normal representation of a line (b) Subdivision of the ρθ-plane into cells

- The range of angle θ is $\pm 90°$, measured with respect to the x-axis. Thus with reference to Fig. 3.3 (a), a horizontal line has $\theta = 0°$, with ρ being equal to the positive x-intercept. Similarly, a vertical line has $\theta = 90°$, with p being equal to the positive y-intercept, or $\theta = -90°$, with ρ being equal to the negative y-intercept.

iii) Global processing via graph-theoretic techniques:

- In this process we have a global approach for edge detection and linking based on representing edge segments in the form of a graph and searching the graph for low-cost paths that correspond to significant edges. This representation provides a rugged approach that performs well in the presence of noise.

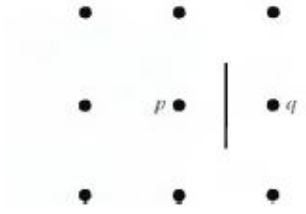


Fig.3.4 Edge element between pixels p and q

- We begin the development with some basic definitions. A graph G = (N,U) is a finite, nonempty set of nodes N, together with a set U of unordered pairs of distinct elements of N. Each pair (n_i, n_j) of U is called an arc. A graph in which the arcs are directed is called a directed graph. If an arc is directed from node n_i to node n_j, then n_j is said to be a successor of the parent node n_i. The process of identifying the successors of a node is called expansion of the node. In each graph we define levels, such that level 0 consists of a single node, called the start or root node, and the nodes in the last level are called goal nodes. A cost $c(n_i, n_j)$ can be associated with every arc (n_i, n_j). A sequence of nodes $n_1, n_2 \ldots n_k$, with each node n_i being a successor of node n_{i-1} is called a path from n_1 to n_k. The cost of the entire path is

$$c = \sum_{i=2}^{k} c(n_{i-1}, n_i).$$

- The following discussion is simplified if we define an edge element as the boundary between two pixels p and q, such that p and q are 4-neighbors, as Fig.3.4 illustrates. Edge elements are identified by the xy-coordinates of points p and q. In other words, the edge element in Fig. 3.4 is defined by the pairs (x_p, y_p) (x_q, y_q). Consistent with the definition an edge is a sequence of connected edge elements.
- We can illustrate how the concepts just discussed apply to edge detection using the 3 X 3 image shown in Fig. 3.5 (a). The outer numbers are pixel coordinates and the numbers in brackets represent gray-level values. Each edge element, defined by pixels p and q, has an associated cost, defined as

$$c(p, q) = H - [f(p) - f(q)]$$

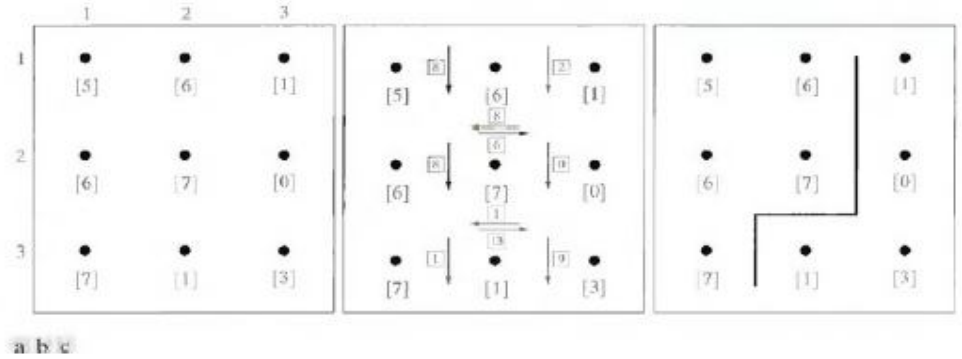


Fig.3.5 (a) A 3 X 3 image region, (b) Edge segments and their costs, (c) Edge corresponding to the lowest-cost path in the graph shown in Fig. 3.6

- Where H is the highest gray-level value in the image (7 in this case), and f(p) and f(q) are the gray-level values of p and q, respectively. By convention, the point p is on the right-hand side of the direction of travel along edge elements. For example, the edge segment (1, 2) (2, 2) is between points (1, 2) and (2, 2) in Fig. 3.5 (b). If the direction of travel is to the right, then p is the point with coordinates (2, 2) and q is point with coordinates (1, 2); therefore, c (p, q) = 7 - [7 - 6] = 6. This cost is shown in the box below the edge segment. If, on the other hand, we are traveling to the left between the same two points, then p is point (1, 2) and q is (2, 2). In this case the cost is 8, as shown above the edge segment in Fig. 3.5(b). To simplify the discussion, we assume that edges start in the top row and terminate in the last row, so that the first element of an edge can be only between points (1, 1), (1, 2) or (1, 2), (1, 3). Similarly, the last edge element has to be between points (3, 1), (3, 2) or (3, 2), (3, 3). Keep in mind that p and q are 4-neighbors, as noted earlier. Figure 3.6 shows the graph for this problem. Each node (rectangle) in the graph corresponds to an edge element from Fig. 3.5. An arc exists between two nodes if the two corresponding edge elements taken in succession can be part of an edge.
- As in Fig. 3.5 (b), the cost of each edge segment, is shown in a box on the side of the arc leading into the corresponding node. Goal nodes are shown shaded. The minimum cost path is shown dashed, and the edge corresponding to this path is shown in Fig. 3.5 (c).

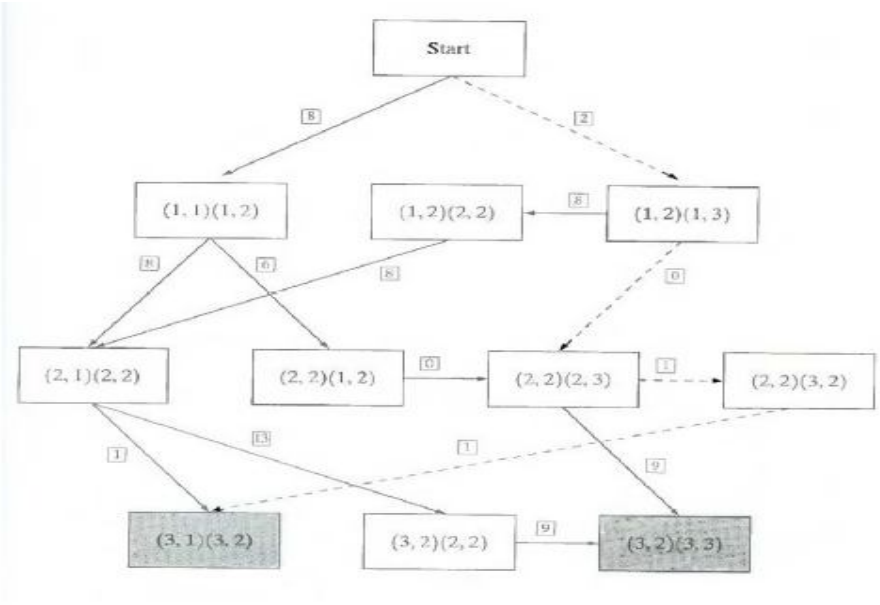


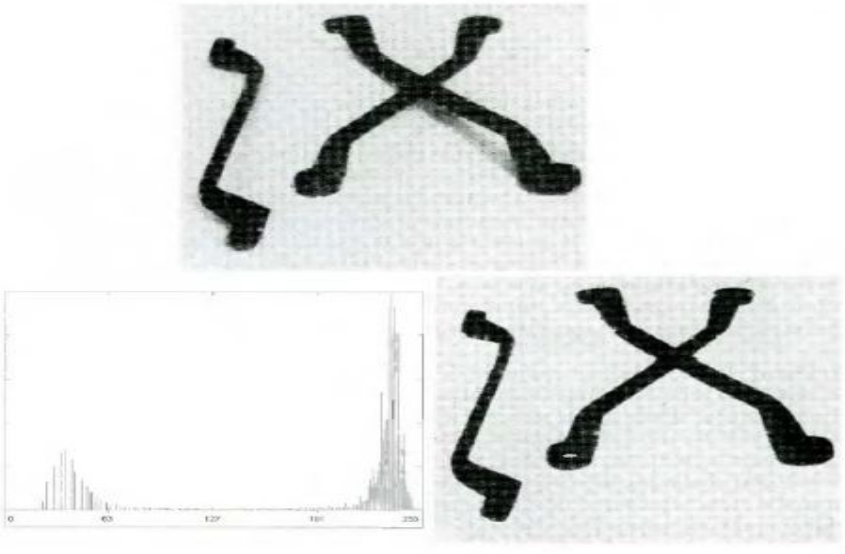
Fig. 3.6 Graph for the image in Fig.3.5 (a). The lowest-cost path is shown dashed.

Thresholding:

- Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation.

Global Thresholding:

- The simplest of all thresholding techniques is to partition the image histogram by using a single global threshold, T. Segmentation is then accomplished by scanning the image pixel by pixel and labeling each pixel as object or back-ground, depending on whether the gray level of that pixel is greater or less than the value of T. As indicated earlier, the success of this method depends entirely on how well the histogram can be partitioned.



a
b c **Fig.4.1 FIGURE 10.28 (a) Original image, (b) Image histogram, (c) Result of global thresholding with T midway between the maximum and minimum gray levels.**

- Figure 4.1(a) shows a simple image, and Fig. 4.1(b) shows its histogram. Figure 4.1(c) shows the result of segmenting Fig. 4.1(a) by using a threshold T midway between the maximum

and minimum gray levels. This threshold achieved a "clean" segmentation by eliminating the shadows and leaving only the objects themselves. The objects of interest in this case are darker than the background, so any pixel with a gray level ≤ T was labeled black (0), and any pixel with a gray level ≥ T was labeled white (255).The key objective is merely to generate a binary image, so the black-white relationship could be reversed. The type of global thresholding just described can be expected to be successful in highly controlled environments. One of the areas in which this often is possible is in industrial inspection applications, where control of the illumination usually is feasible.

- The threshold in the preceding example was specified by using a heuristic approach, based on visual inspection of the histogram. The following algorithm can be used to obtain T automatically:
    1. Select an initial estimate for T.
    2. Segment the image using T. This will produce two groups of pixels: G_1 consisting of all pixels with gray level values >T and G_2 consisting of pixels with values < T.
    3. Compute the average gray level values μ_1 and μ_2 for the pixels in regions G_1 and G_2.
    4. Compute a new threshold value:

$$T = \frac{1}{2}(\mu_1 + \mu_2).$$

    5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_o.
- When there is reason to believe that the background and object occupy comparable areas in the image, a good initial value for T is the average gray level of the image. When objects are small compared to the area occupied by the background (or vice versa), then one group of pixels will dominate the histogram and the average gray level is not as good an initial choice. A more appropriate initial value for T in cases such as this is a value midway between the maximum and minimum gray levels. The parameter T_o is used to stop the algorithm after changes become small in terms of this parameter. This is used when speed of iteration is an important issue.

Basic Adaptive Thresholding:

- Imaging factors such as uneven illumination can transform a perfectly segmentable histogram into a histogram that cannot be partitioned effectively by a single global threshold. An approach for handling such a situation is to divide the original image into subimages and then utilize a different threshold to segment each subimage. The key issues in this approach are how to subdivide the image and how to estimate the threshold for each resulting subimage. Since the threshold used for each pixel depends on the location of the pixel in terms of the subimages, this type of thresholding is adaptive.

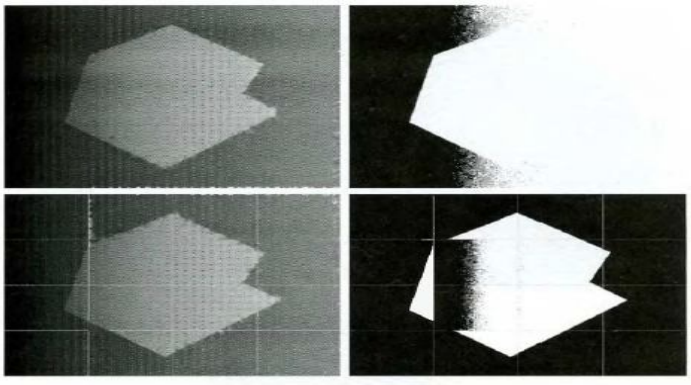


Fig.5 (a) Original image, (b) Result of global thresholding. (c) Image subdivided into individual subimages (d) Result of adaptive thresholding.

- We illustrate adaptive thresholding with a example. Figure 5(a) shows the image, which we concluded could not be thresholded effectively with a single global threshold. In fact, Fig.

5(b) shows the result of thresholding the image with a global threshold manually placed in the valley of its histogram. One approach to reduce the effect of non uniform illumination is to subdivide the image into smaller sub images, such that the illumination of each subimage is approximately uniform. Figure 5(c) shows such a partition, obtained by subdividing the image into four equal parts, and then subdividing each part by four again. All the sub images that did not contain a boundary between object and back-ground had variances of less than 75. All sub images containing boundaries had variances in excess of 100. Each sub image with variance greater than 100 was segmented with a threshold computed for that sub image using the algorithm. The initial value for T in each case was selected as the point midway between the minimum and maximum gray levels in the sub image. All sub images with variance less than 100 were treated as one composite image, which was segmented using a single threshold estimated using the same algorithm. The result of segmentation using this procedure is shown in Fig. 5(d).

- With the exception of two sub images, the improvement over Fig. 5(b) is evident. The boundary between object and background in each of the improperly segmented sub images was small and dark, and the resulting histogram was almost unimodal.

Use of Boundary Characteristics for Histogram Improvement and Local Thresholding:

- It is intuitively evident that the chances of selecting a "good" threshold are enhanced considerably if the histogram peaks are tall, narrow, symmetric, and separated by deep valleys. One approach for improving the shape of histograms is to consider only those pixels that lie on or near the edges between objects and the background. An immediate and obvious improvement is that histograms would be less dependent on the relative sizes of objects and the background.
- For instance, the histogram of an image composed of a small object on a large background area (or vice versa) would be dominated by a large peak because of the high concentration of one type of pixels.
- If only the pixels on or near the edge between object and the background were used, the resulting histogram would have peaks of approximately the same height. In addition, the probability that any of those given pixels lies on an object would be approximately equal to the probability that it lies on the back-ground, thus improving the symmetry of the histogram peaks. Finally, as indicated in the following paragraph, using pixels that satisfy some simple measures based on gradient and Laplacian operators has a tendency to deepen the valley between histogram peaks.
- The principal problem with the approach just discussed is the implicit assumption that the edges between objects and background arc known. This information clearly is not available during segmentation, as finding a division between objects and background is precisely what segmentation is all about. However, an indication of whether a pixel is on an edge may be obtained by computing its gradient. In addition, use of the Laplacian can yield information regarding whether a given pixel lies on the dark or light side of an edge. The average value of the Laplacian is 0 at the transition of an edge, so in practice the valleys of histograms formed from the pixels selected by a gradient/Laplacian criterion can be expected to be sparsely populated. This property produces the highly desirable deep valleys.
- The gradient ∇f at any point (x, y) in an image can be found. Similarly, the Laplacian $\nabla^2 f$ can also be found. These two quantities may be used to form a three-level image, as follows:

$$s(x,y) = \begin{cases} 0 & \text{if } \nabla f < T \\ + & \text{if } \nabla f \geq T \quad \text{and} \quad \nabla^2 f \geq 0 \\ - & \text{if } \nabla f \geq T \quad \text{and} \quad \nabla^2 f < 0 \end{cases}$$

- Where the symbols 0, +, and - represent any three distinct gray levels, T is a threshold, and the gradient and Laplacian are computed at every point (x, y). For a dark object on a light background, the use of the Eqn. produces an image s(x, y) in which (1) all pixels that are not on an edge (as determined by ∇f being less than T) are labeled 0; (2) all pixels on the dark side of an edge are labeled +; and (3) all pixels on the light side of an edge are labeled -. The symbols + and - in Eq. above are reversed for a light object on a dark background. Figure 6.1 shows the labeling produced by Eq. for an image of a dark, underlined stroke written on a light background.
- The information obtained with this procedure can be used to generate a segmented, binary image in which l's correspond to objects of interest and 0's correspond to the background. The transition (along a horizontal or vertical scan line) from a light background to a dark object must be characterized by the occurrence of a - followed by a + in s (x, y). The interior of the object is composed of pixels that are labeled either 0 or +. Finally, the transition from the object back to the background is characterized by the occurrence of a + followed by a -. Thus a horizontal or vertical scan line containing a section of an object has the following structure:

(· · ·)(−, +)(0 or +)(+, −)(· · ·)

Fig.6.1 Image of a handwritten stroke coded by using Eq. discussed above

- Where (…) represents any combination of +, -, and 0. The innermost parentheses contain object points and are labeled 1. All other pixels along the same scan line are labeled 0, with the exception of any other sequence of (- or +) bounded by (-, +) and (+, -).

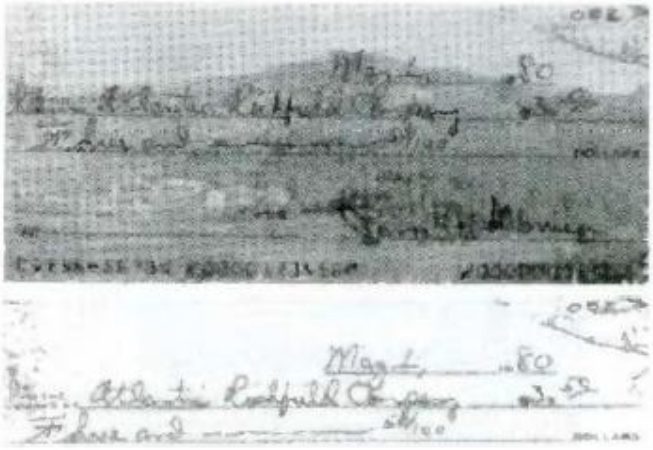


Fig.6.2 (a) Original image, (b) Image segmented by local thresholding.

- Figure 6.2 (a) shows an image of an ordinary scenic bank check. Figure 6.3 shows the histogram as a function of gradient values for pixels with gradients greater than 5. Note that this histogram has two dominant modes that are symmetric, nearly of the same height, and arc separated by a distinct valley. Finally, Fig. 6.2(b) shows the segmented image obtained by with T at or near the midpoint of the valley. Note that this example is an illustration of local thresholding, because the value of T was determined from a histogram of the gradient and Laplacian, which are local properties.

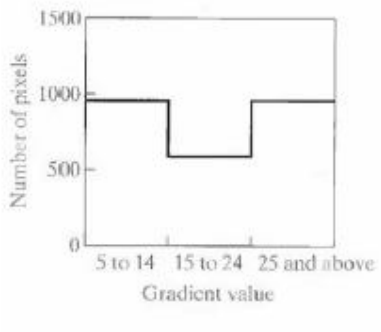


Fig.6.3 Histogram of pixels with gradients greater than 5

Region-Based Segmentation:

- The objective of segmentation is to partition an image into regions. We approached this problem by finding boundaries between regions based on discontinuities in gray levels, whereas segmentation was accomplished via thresholds based on the distribution of pixel properties, such as gray-level values or color.

Basic Formulation:

- Let R represent the entire image region. We may view segmentation as a process that partitions R into n sub regions, R_1, R_2..., R_n, such that

$$\text{(a)} \bigcup_{i=1}^{n} R_i = R.$$
$$\text{(b)} \ R_i \text{ is a connected region}, i = 1, 2, \ldots, n.$$
$$\text{(c)} \ R_i \cap R_j = \varnothing \text{ for all } i \text{ and } j, i \neq j.$$
$$\text{(d)} \ P(R_i) = \text{TRUE for } i = 1, 2, \ldots, n.$$
$$\text{(e)} \ P(R_i \cup R_j) = \text{FALSE for } i \neq j.$$

- Here, P (R_i) is a logical predicate defined over the points in set R_i and Ø` is the null set. Condition
- (a) Indicates that the segmentation must be complete; that is, every pixel must be in a region. Condition
- (b) Requires that points in a region must be connected in some predefined sense. Condition
- (c) Indicates that the regions must be disjoint. Condition
- (d) Deals with the properties that must be satisfied by the pixels in a segmented region—for example P (R_i) = TRUE if all pixels in R_i, have the same gray level.
- Finally, condition (e) indicates that regions R_i and R_j are different in the sense of predicate P.

Region Growing:

- As its name implies, **region growing** is a procedure that groups pixels or sub regions into larger regions based on predefined criteria. The basic approach is to start with a set of "seed" points and from these grow regions by appending to each seed those neighbouring pixels that

have roperties similar to the seed (such as specific ranges of gray level or color). When a priori information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations shows clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds.

- The selection of similarity criteria depends not only on the problem under consideration, but also on the type of image data available. For example, the analysis of land-use satellite imagery depends heavily on the use of color. This problem would be significantly more difficult, or even impossible, to handle without the inherent information available in color images. When the images are monochrome, region analysis must be carried out with a set of descriptors based on gray levels and spatial properties (such as moments or texture).
- Basically, growing a region should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as gray level, texture, and color, are local in nature and do not take into account the "history" of region growth. Additional criteria that increase the power of a region growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far (such as a comparison of the gray level of a candidate and the average gray level of the grown region), and the shape of the region being grown. The use of these types of descriptors is based on the assumption that a model of expected results is at least partially available.
- In following Figure (a) shows an X-ray image of a weld (the horizontal dark region) containing several cracks and porosities (the bright, white streaks running horizontally through the middle of the image). We wish to use region growing to segment the regions of the weld failures. These segmented features could be used for inspection, for inclusion in a database of historical studies, for controlling an automated welding system, and for other numerous applications.

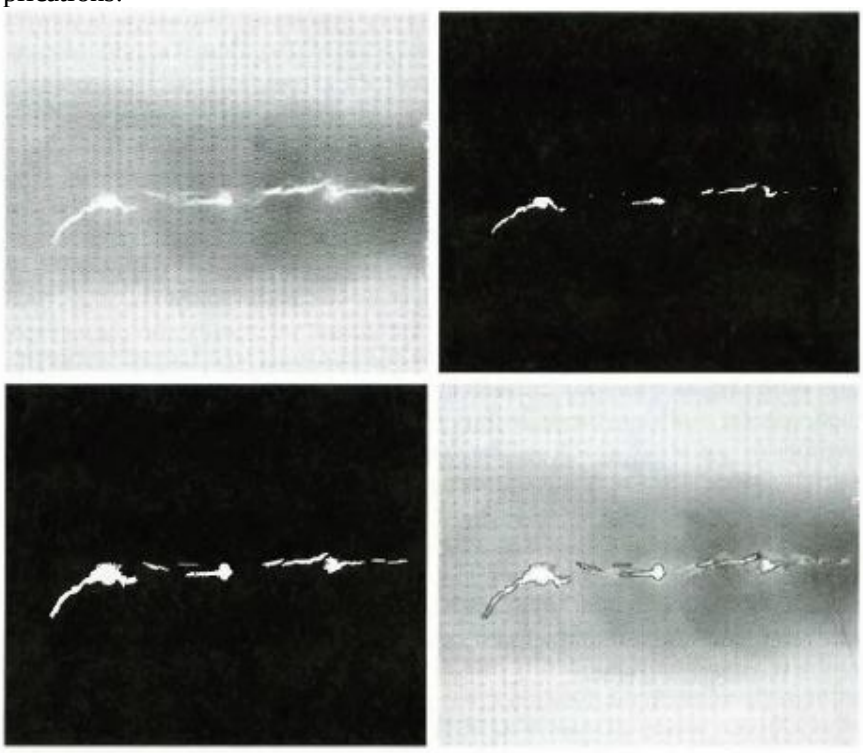


Fig.7.1 (a) Image showing defective welds, (b) Seed points, (c) Result of region growing, (d) Boundaries of segmented ; defective welds (in black).

- The first order of business is to determine the initial seed points. In this application, it is known that pixels of defective welds tend to have the maximum allowable digital value 255 in this case. Based on this information, we selected as starting points all pixels having values of 255. Note that many of the points are clustered into **seed regions**.
- The next step is to choose criteria for region growing. In this particular example we chose two criteria for a pixel to be annexed to a region: (1) the absolute gray-level difference between any pixel and the seed had to be less than 65. This number is based on the histogram shown in Fig. 7.2 and represents the difference between 255 and the location of the first major valley to the left, which is representative of the highest gray level value in the dark weld region. (2) To be included in one of the regions, the pixel had to be 8-connected to at least one pixel in that region.
- If a pixel was found to be connected to more than one region, the regions were merged. Figure 7.1 (c) shows the regions that resulted by starting with the seeds in Fig. 7.2 (b) and utilizing the criteria defined in the previous paragraph. Superimposing the boundaries of these regions on the original image [Fig. 7.1(d)] reveals that the region-growing procedure did indeed segment the defective welds with an acceptable degree of accuracy. It is of interest to note that it was not necessary to specify any stopping rules in this case because the criteria for region growing were sufficient to isolate the features of interest.

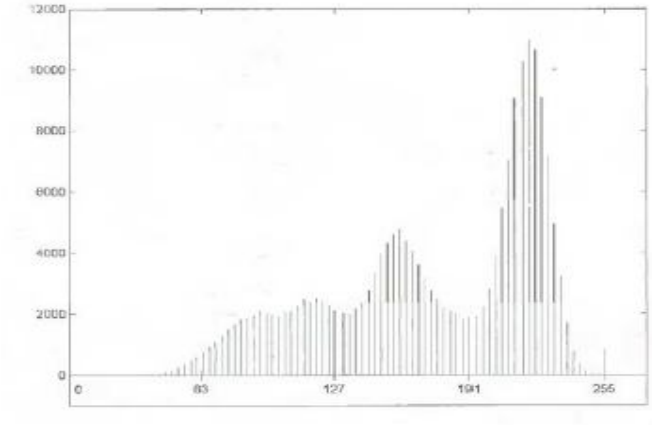


Fig.7.2 Histogram of Fig. 7.1 (a)

Region Splitting and Merging:

- The procedure just discussed grows regions from a set of seed points. An alternative is to subdivide an image initially into a set of arbitrary, disjointed regions and then merge and/or split the regions in an attempt to satisfy the conditions. A split and merge algorithm that iteratively works toward satisfying these constraints is developed.
- Let R represent the entire image region and select a predicate P. One approach for segmenting R is to subdivide it successively into smaller and smaller quadrant regions so that, for any region R_i, $P(R_i)$ = TRUE. We start with the entire region. If P(R) = FALSE, we divide the image into quadrants. If P is FALSE for any quadrant, we subdivide that quadrant into sub quadrants, and so on. This particular splitting technique has a convenient representation in the form of a so-called quadtree (that is, a tree in which nodes have exactly four descendants), as illustrated in Fig. 7.3. Note that the root of the tree corresponds to the entire image and that each node corresponds to a subdivision. In this case, only R_4 was subdivided further.

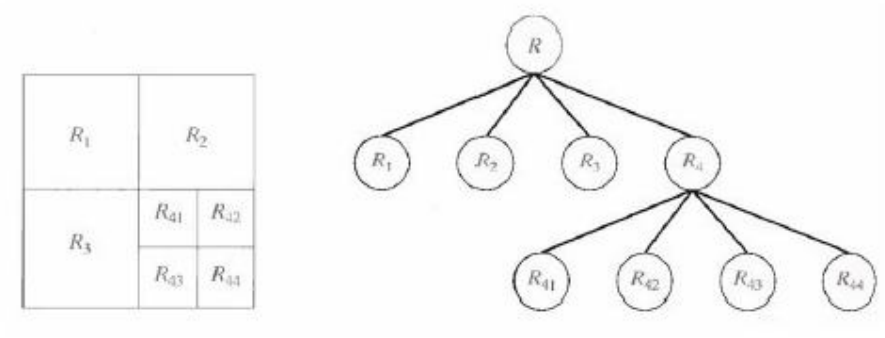


Fig. 7.3 (a) Partitioned image (b) Corresponding quadtree.

- If only splitting were used, the final partition likely would contain adjacent regions with identical properties. This drawback may be remedied by allowing merging, as well as splitting. Satisfying the constraints, requires merging only adjacent regions whose combined pixels satisfy the predicate P. That is, two adjacent regions R_j and R_k are merged only if P (R_j U R_k) = TRUE.
- The preceding discussion may be summarized by the following procedure, in which, at any step we
    1. Split into four disjoint quadrants any region R_i, for which P (R_i) = FALSE.
    2. Merge any adjacent regions R_j and R_k for which P (R_j U R_k) = TRUE.
    3. Stop when no further merging or splitting is possible.
- Several variations of the preceding basic theme are possible. For example, one possibility is to split the image initially into a set of blocks. Further splitting is carried out as described previously, but merging is initially limited to groups of four blocks that are descendants in the quadtree representation and that satisfy the predicate P. When no further mergings of this type are possible, the procedure is terminated by one final merging of regions satisfying step 2. At this point, the merged regions may be of different sizes. The principal advantage of this approach is that it uses the same quadtree for splitting and merging, until the final merging step.

UNIT-VI
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. $R_i \cap R_j = \varnothing$ for all i and j, $i \neq j$. means regions must be disjoint.
[True/False]
2. _____ are basic types of discontinuities in digital images. [ ]
A) Points B) Lines c) Edges D)all
3_____ is a set of connected pixels that lie on the boundary between two regions.
A)Edge B) border C) point D)none [ ]
4How many thresholds permit to detect strong and weak edges?[ ]
A)1 B) 2 C) 3 D) 4
5Weak edges are accepted only if they are _____to strong edges. [ ]
A)connected B) Not connected c) both D) none
6_____ is a procedure that groups pixels or sub regions into larger regions based on predefined criteria.
7What is the choose criteria for region growing for a pixel to be annexed to a region. [ ]
A) The absolute gray-level difference between any pixel and the seed had to be less than some threshold
B) To be included in one of the regions, the pixel had to be 8-connected to at least one pixel in that region.
C)Both
D)none
8A cost c (ni, nj) can be associated with every arc (ni, nj). A sequence of nodes n1, n2... nk, with each node ni being a successor of node ni-1 is called a path from n1 to nk. The cost of the entire path is_____

9In Region Growing, If a pixel was found to be connected to more than one region, the regions were merged
[True/False]
10In region splitting and merging we stop the process when no further merging or splitting is possible. [True/False]

11.In Region Splitting and Merging, Split into four disjoint quadrants any region Ri, for which P (Ri) = _____ []
A) FALSE B)TRUE C)NULL D)None
12.In Region Splitting and Merging Merge any adjacent regions Rj and Rk for which P (Rj U Rk) = _____ [ ]
A) FALSE B) TRUE C)NULL D)None
13. If f(x,y) is an image function of two variables, then the first order derivative of a one dimensional function, f(x) is: [ ]
A, f(x+1)-f(x) B, f(x)-f(x+1) C, f(x-1)-f(x+1) D, f(x)+f(x-1)

14.The derivative of digital function is defined in terms of difference. Then, which of the following defines the second order derivative $\partial 2\ f/\partial x2$ = _____ of a one-dimensional function f(x)? [ ]

A, f(x+1)-f(x) B, f(x+1)+f(x-1)-2f(x)

C, All of the mentioned depending upon the time when partial derivative will be dealt along two spatial axes

D, None of the mentioned

15.Using the equation y = ax + b to represent a line is that the slope approaches _____as the line approaches the vertical.

A) Infinity B) 1 C) 0 D) none [ ]

16._____is the parametric representation of a line.

17.In this line equation $x\cos\theta + y\sin\theta = \rho$ if $\theta = 0°$, with ρ being equal to the positive x-intercept. Then the line is _____

A) Vertical line B) Horizontal line C)angle with 45^0 D) none[ ]

18.In this line equation $x\cos\theta + y\sin\theta = \rho$ if $\theta = 90°$, with ρ being equal to the positive y-intercept. Then the line is _____ [ ]

AVertical line B) Horizontal line C)angle with 45^0 D) none

19.In this line equation $x\cos\theta + y\sin\theta = \rho$ if $\theta = -90°$, with ρ being equal to the negative y-intercept. Then the line is _____ [ ]

AVertical line B) Horizontal line C)angle with 45^0 D) none

20.The highest gray-level value in image is 7. Each edge element, defined by pixels p and q and the gray level of p is 7 and q is 6 then it has an associated cost_____ [ ]

A) 6 B) 7 C) 0 D) 1

SECTION-B

SUBJECTIVE QUESTIONS

1Explain the method for point detection.
2Explain the method for line detection.
3Explain the method for edge detection.
4What are the derivative operators useful in image segmentation? Explain their role in segmentation.
5Explain about the edge linking procedures.
6What is thresholding? Explain about global thresholding.
7Explain about basic adaptive thresholding process used in image segmentation.
8Explain in detail the threshold selection based on boundary characteristics.
9Explain Region Growing in region based segmentation.
10Explain about Region splitting and merging in region based segmentation.

11Develop a procedure for edge detection by using gradient
12Derive and represent various masks used to compute the gradient at point labeled z5 in a 3 X 3 region of an image (the z's are gray-level values).

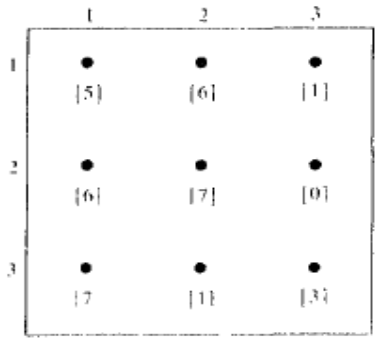
13Represent Prewitt and Sobel masks for detecting diagonal edges
14Develop a procedure for edge detection by using laplacian.
15Develop Laplacian masks used to implement the equation

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

16Develop Laplacian masks used to implement the equation

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9).$$

17Find the edge corresponding to the minimum cost path in the sub image shown below



18Derive Global processing via the Hough Transform.
19Develop region splitting and merging in terms of basic formulation.