UNIT-1:

Introduction to Digital Image processing , Example fields and Applications, Fundamental steps in DIP, Components of DIP , Image Sensing and Acquisition, Image Modelling - Sampling, Quantization, Digital Image representation , Basic relationships between pixels, Mathematical tools/ operations applied on images , Imaging geometry.

Introduction to Digital Image processing:

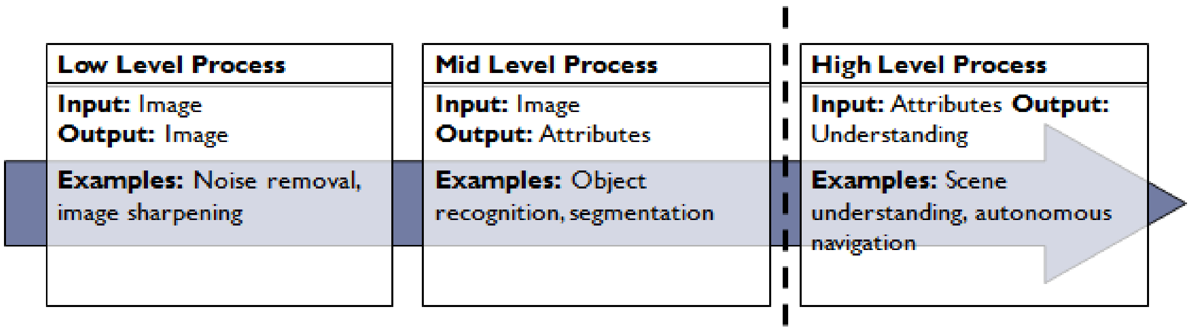
- Digital image processing methods are interested by <u>two major applications:</u>

i. Improvement of pictorial information for human interpretation and processing of image data for storage, transmission.
   - Noise filtering
   - Content Enhancement
     - Contrast enhancement
     - Deblurring
   - Remote sensing

ii. Representation for autonomous machine perception.

Definition:

- The process of receiving and analyzing visual information by the human species is referred to a *sight*, *perception* or *understanding*. Similarly, the process of receiving and analyzing visual information by digital computer is called *digital image processing* and *scene analysis*.
- 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. 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.
- The continuum from image processing to computer vision can be broken up into low-, mid- and high-level processes.



Applications

The use of digital image processing techniques has exploded and they are now used for all kinds of tasks in all kinds of areas

1. Image enhancement/restoration
2. Medical visualization
3. Artistic effect
4. Industrial inspection
5. Law enforcement
6. Human computer interfaces
7. Remote sensing via satellites and other spacecrafts
8. Image transmission and storage for business applications
9. Radar ,SONAR, Acoustic image processing
10. Robotics

Fundamental Steps in Digital Image Processing:

Image acquisition is the first process shown in Fig.1. 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. ***Image enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application***.

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. Image restoration is the process of restoring the approximation of the original image from the degraded image by using knowledge about degradation function and noise characteristics.

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. Color is used as the basis for extracting features of interest in an image.

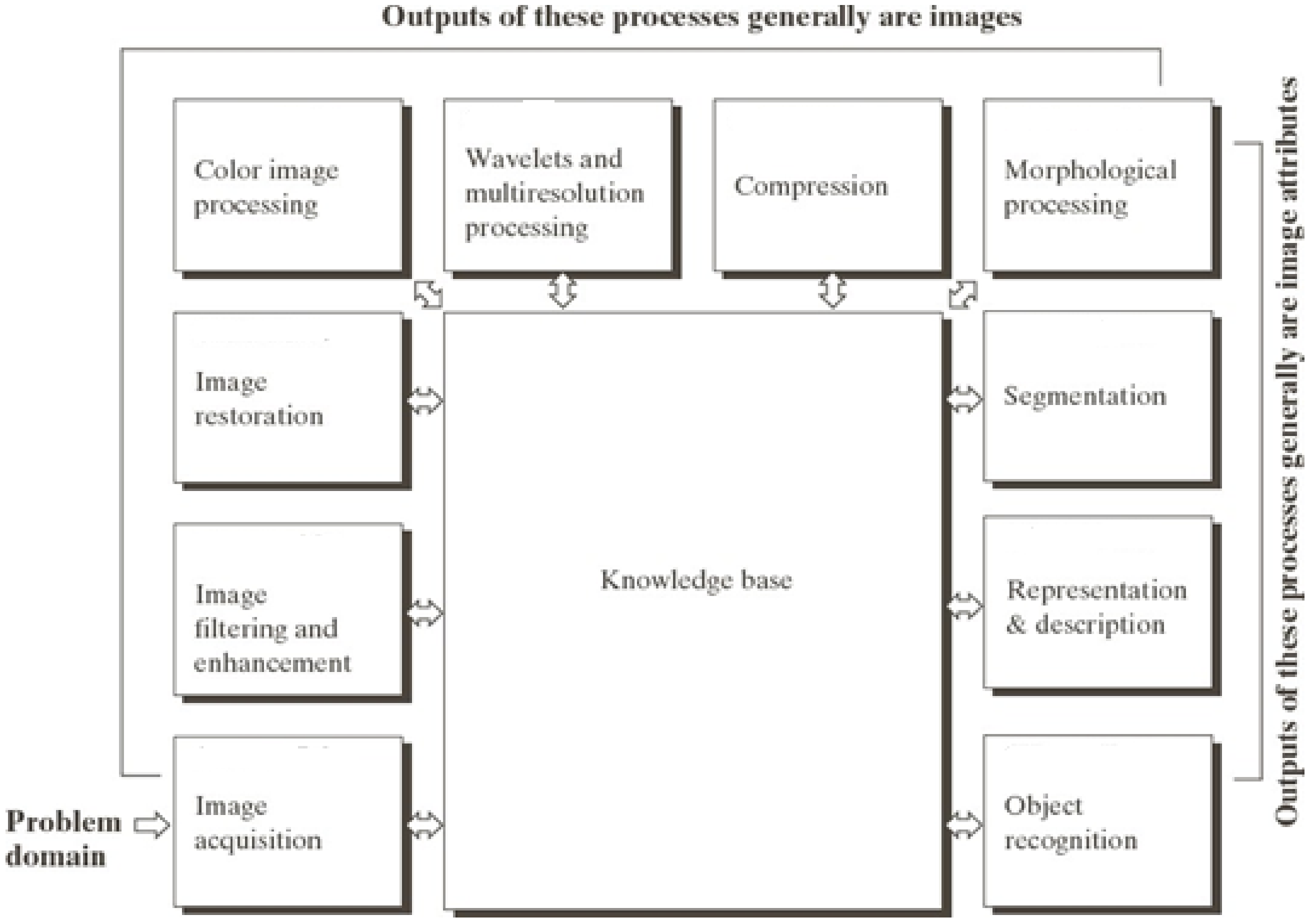


Fig.1. Fundamental steps in Digital Image Processing

Wavelets are the foundation for representing images in various degrees of resolution. This is used for image data compression and for pyramidal representation, in which images are subdivided successively into smaller regions.

Compression deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

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

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

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

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

The interaction between the ***knowledge base*** and the processing modules are shown in fig (1). Knowledge about a problem domain is coded into an image processing system in the form of knowledge database. The knowledge base may be either simple or complex .It guides the operation of each processing module and also controls the interaction between the modules.

Components of an Image Processing System:

Figure 2 shows the basic components comprising a typical *general-purpose system* used for digital image processing.

Image sensors*: T*wo 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. 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: It consists of the digitizer and 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.

Computer: 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.

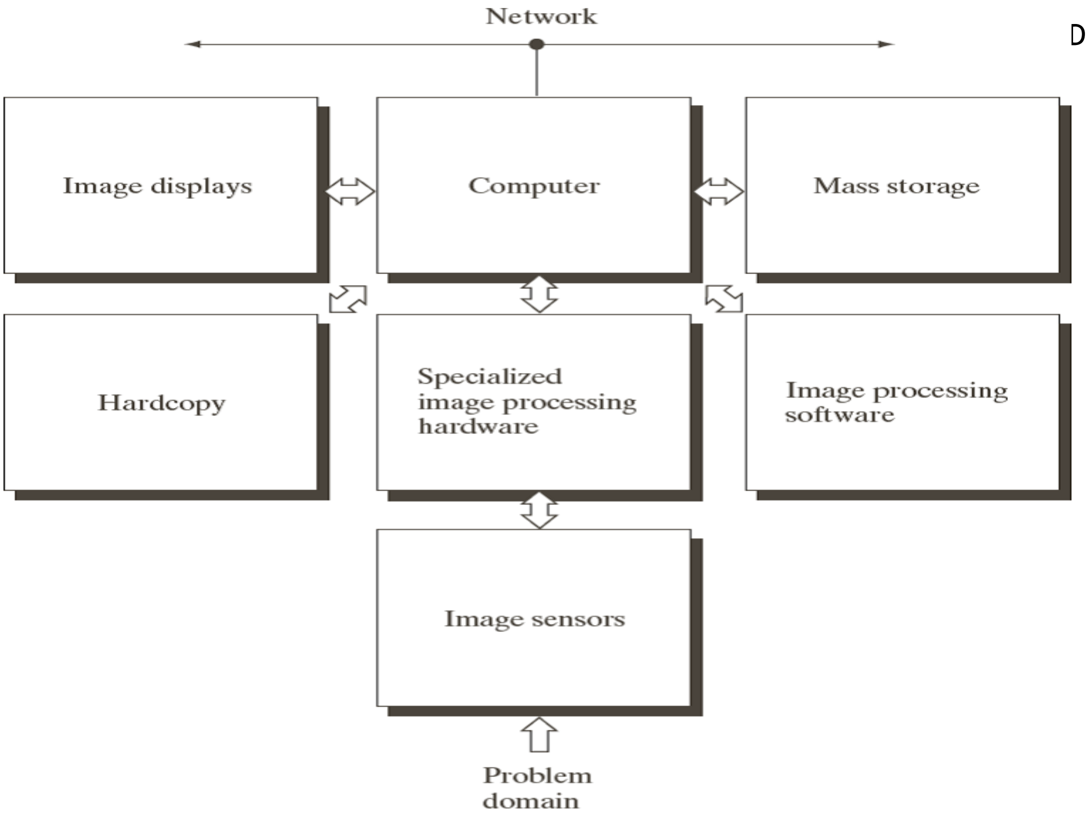


Fig.2. Components of a general purpose Image Processing System

Image processing Software: *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 utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

Mass storage: 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.

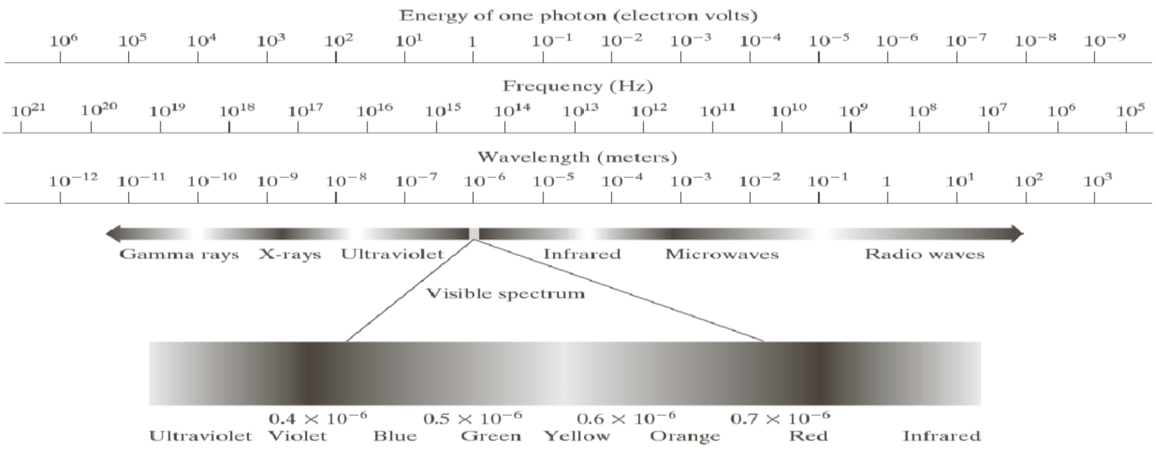
Image displays*:* Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system.

Hardcopy: 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.

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.

Light and Electromagnetic Spectrum: Electromagnetic waves can be visualized as propagating sinusoidal waves with wavelength λ, or they can be thought of as a stream of mass less particles, each travelling in a wave like pattern and moving at the speed of light. Each mass less particle contains a certain amount of energy, called a *photon*.

This energy is proportional to frequency, so the higher-frequency (shorter wavelength) electromagnetic phenomena carry more energy per photon. Light is a particular type of electromagnetic radiation that can be sensed by the human eye. The colors that humans perceive in an object are determined by the nature of the light *reflected* from the object.



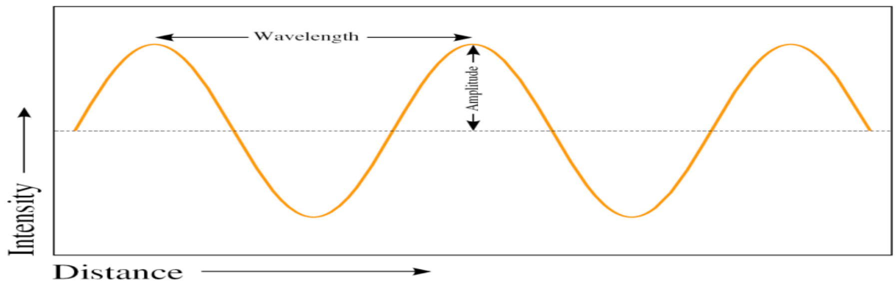


Fig: light wave.

- Light is a particular type of electromagnetic radiation that can be sensed by the human eye.
- The visible band of the EM spectrum spans the range from ~0.43 micrometers (violet) to about 0.79 micrometers (red).
- The colors that humans perceive in an object are determined by the nature of the light *reflected* from the object.
- A body that reflects light relatively balanced in all visible wavelengths appears white to the observer.
- Light that is void of color is called monochromatic or achromatic light.
- The only attribute is intensity or amount. Because the intensity of monochromatic light is perceived to vary from black to grays and finally to white.
- The range of measured values of monochromatic light from black to white is usually called as gray scale, and monochromatic images are referred to as gray-scale images.
- Chromatic (color) light spans the EM energy spectrum from ~0.43 micrometers (violet) to about 0.79 micrometers (red).
- Basic quantities are frequency, radiance, luminance and brightness.
- Radiance(watts) is the total amount of energy that flows from the light source
- Luminance(lumens(lm)) gives a measure of the amount of energy an observer perceives from a light source
- Brightness: It is a subjective descriptor of light perception that is practically impossible to measure.

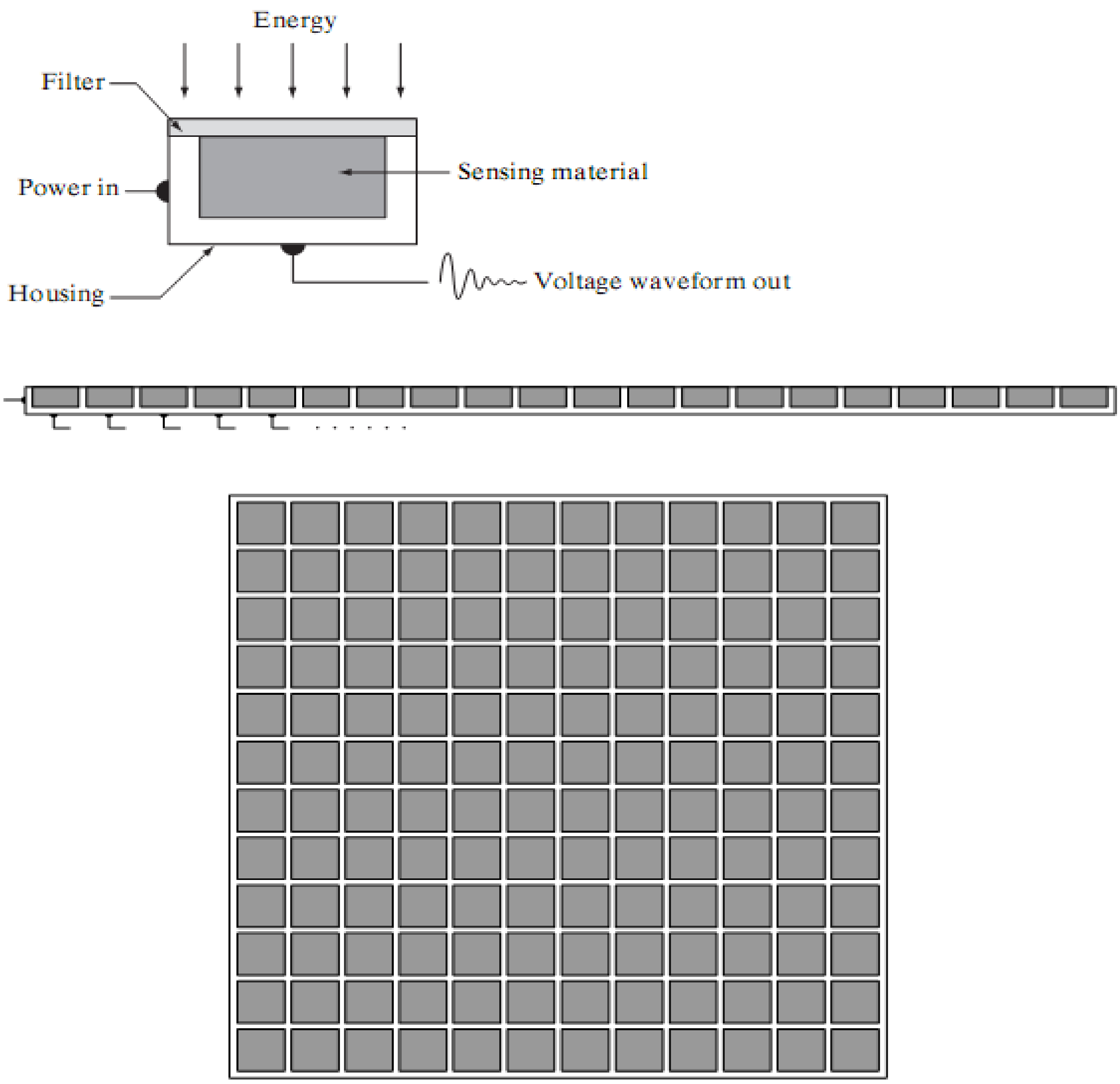
Image Sensing and Acquisition:

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.

For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. 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 1.4 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.



a b c

Fig.1.4 (a) Single imaging Sensor (b) Line sensor (c) Array sensor

(1) Image Acquisition Using a Single Sensor:

Figure 1.4 (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 1.5 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*.

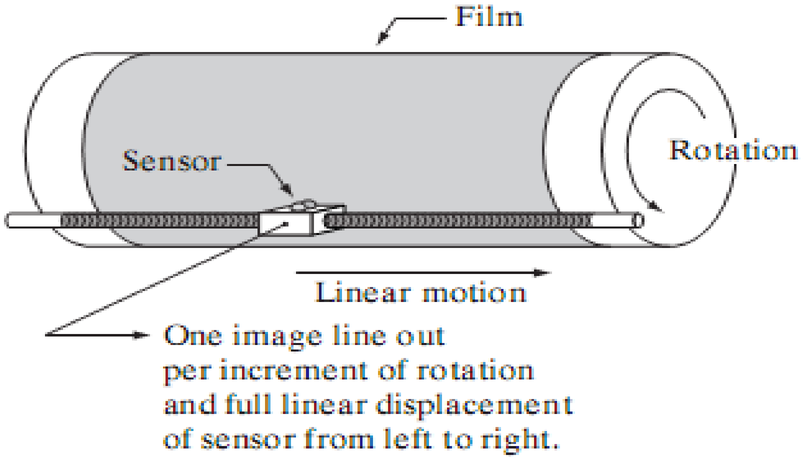
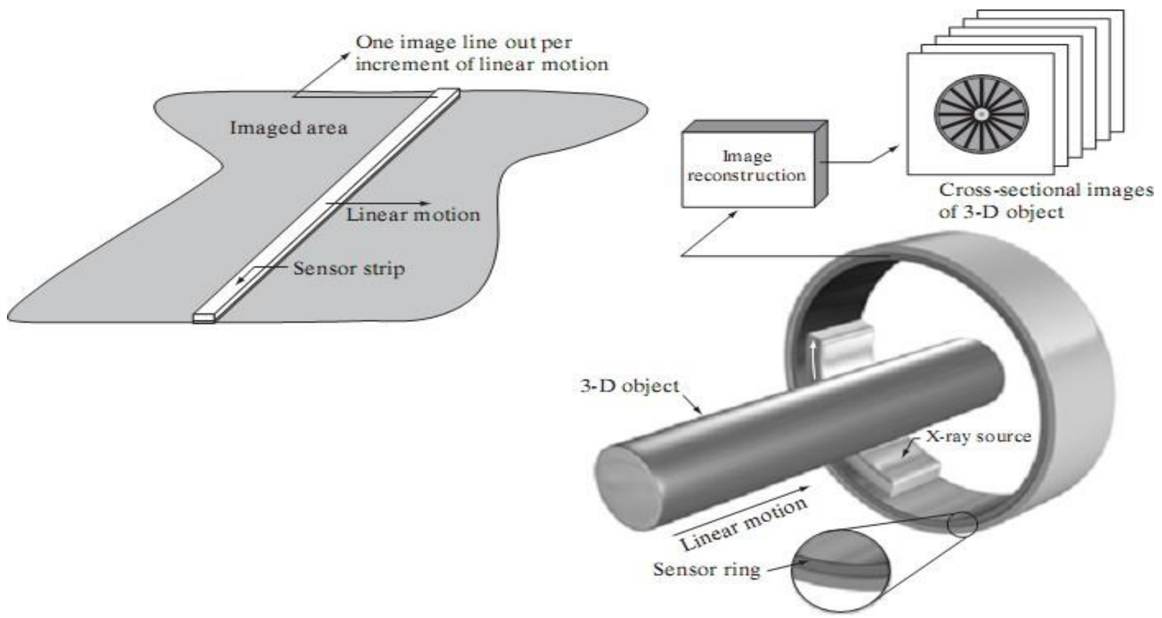


Fig.1.5. Combining a single sensor with motion to generate a 2-D image

(2) Image Acquisition Using Sensor Strips:



a b

Fig.1.6 (a) Image acquisition using a linear sensor strip (b) Image acquisition using a circular sensor strip.

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. 1.4 (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. 1.6 (a). 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. 1.6 (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).

(3)Image Acquisition Using Sensor Arrays:

Figure 1.4 (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. Because the sensor array shown in Fig. 1.4 (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.1.7. This figure shows the energy from an illumination source being reflected from a scene element and the energy also could be transmitted through the scene elements. The first function performed by the imaging system shown in Fig.1.7 (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. 1.7(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 sweeps 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. 1.7 (e).

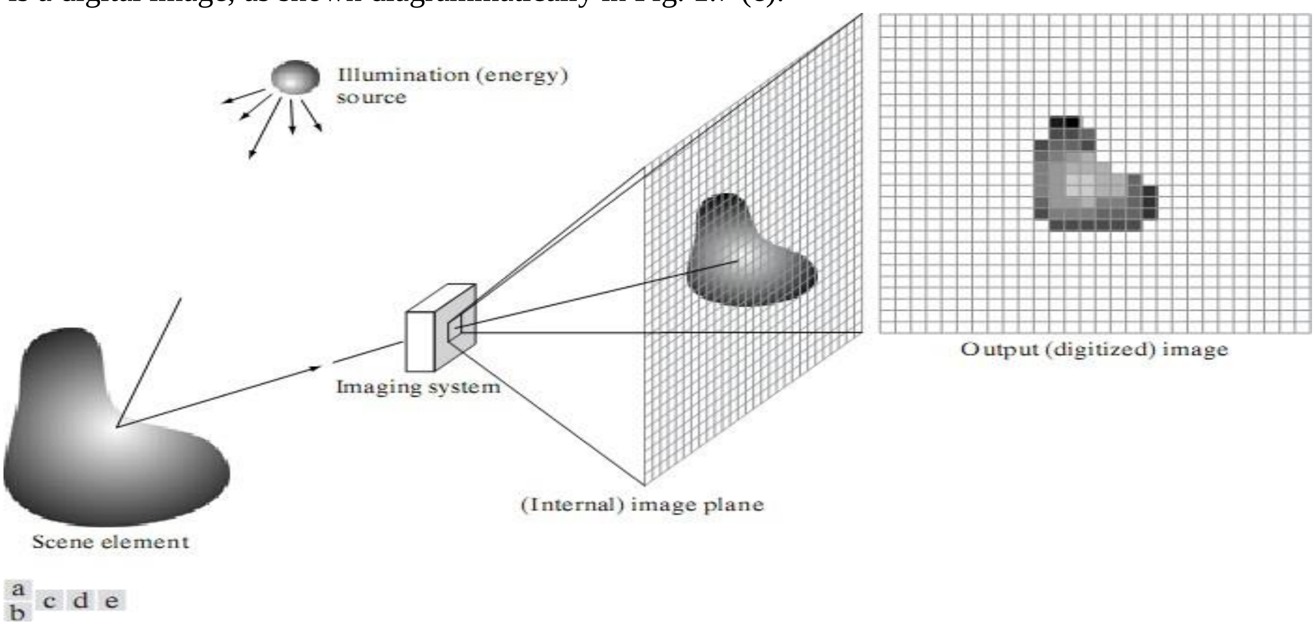


Fig.1.7 An example of the digital image acquisition process (a) Energy ("illumination") source (b) An element of a scene (c) Imaging system (d) Projection of the scene onto the image plane (e) Digitized image.

Image Formation model:

we denote images by two-dimensional functions of the form f(x, y).

The value or amplitude of f at spatial coordinates is a positive scalar quantity whose physical meaning is determined by the source of the image.

When an image is generated from a physical process, its intensity values are proportional to energy radiated by a physical source (e.g., electromagnetic waves).

As a consequence, f(x, y) must be nonzero and finite

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

The function may be characterized by two components:

(1) the amount of source illumination incident on the scene being viewed, and

(2) the amount of illumination reflected by the objects in the scene.

Appropriately, these are called the illumination and reflectance components and are denoted by and i(x, y) ,and r(x, y) respectively. The two functions combine as a product to form :

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

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

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

Let the intensity of a monochrome image at any pair of coordinates(x_0,y_0) be denoted by

$l=f(x_0,y_0)$

L lies in the range Lmin<=l<=Lmax

In theory Lmin to be minimimum ,Lmax to be finite

Lmin=imin rmin

Lmax=imax rmax

The interval [Lmin, Lmax] is called gray(intensity) scale

Common practice is to shift the interval [0 L-1]

where L=0 is black

L-1is white

All intermediate values are shades of gray.

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 Figure 1.8(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*.

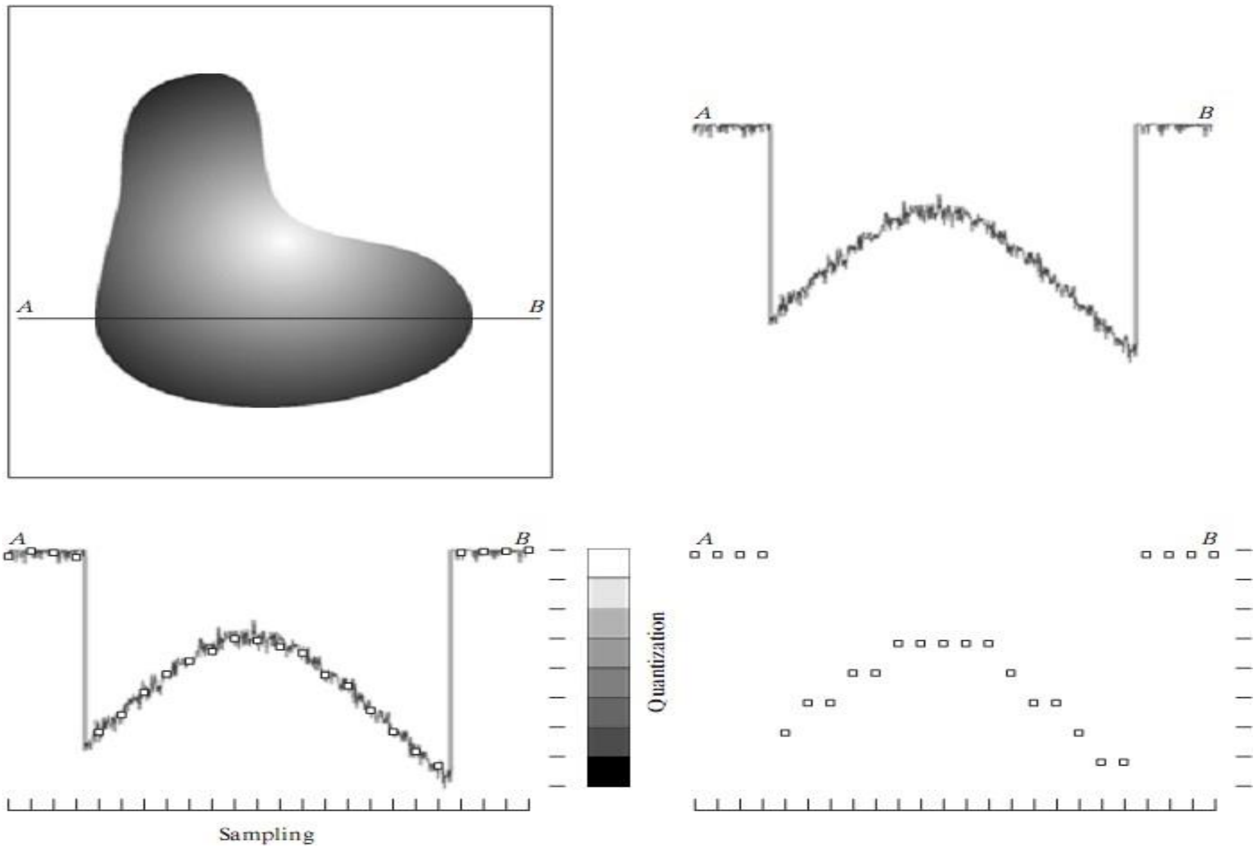
The one-dimensional function shown in Fig.1.8 (b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in Fig. 1.8(a).The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in Fig.1.8 (c).

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.

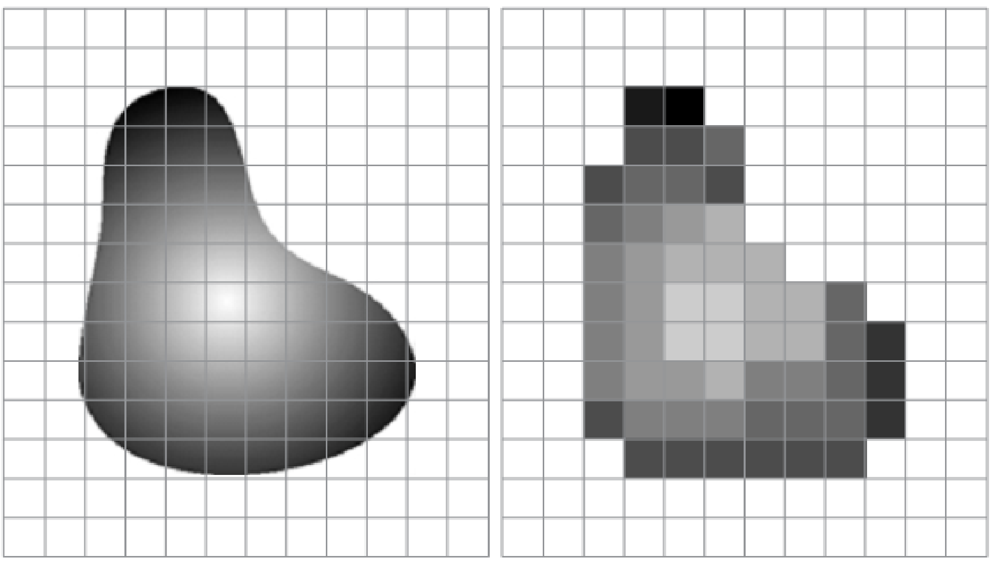
The right side of Fig. 1.8 (c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight gray levels. The continuous gray levels are quantized simply by assigning one of the eight discrete 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.1.8 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.

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.

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 1.9 illustrates this concept. Figure1.9 (a) shows a continuous image projected onto the plane of an array sensor. Figure 1.9 (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.



a b
c d
Fig.1.8. 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



a b

Fig.1.9. (a) Continuous image projected onto a sensor array (b) Result of image sampling and quantization.

UNIFORM SAMPLING:
Digitizing the coordinate values with equal spacing is called uniform *sampling*.
NON UNIFORM SAMPLING:
Digitizing the coordinate values with non uniform spacing is called non uniform *sampling*.
There are two types:
Fine sampling: it is required in the neighborhood of sharp gray level transitions.
Coarse sampling: it is utilized in relatively smooth regions.
Eg: consider a simple image consisting of face superimposed on a uniform background. Clearly the background carries little detailed information and it can be represented by coarse sampling. The face contains considerably more details and it is represented by fine sampling.

<u>Representing Digital Images</u>

There are 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). Figure 1.10 shows the coordinate convention used.

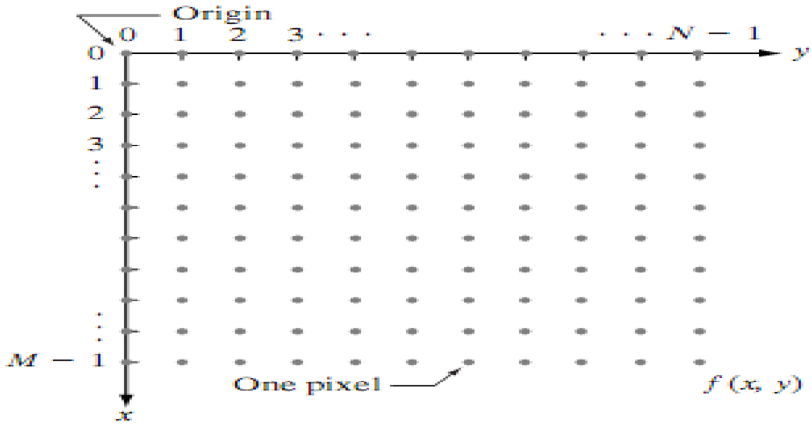


Fig 1.10 Coordinate convention used to represent digital images

The complete M*N digital image 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.

- The number of intensity levels (L) typically is an integer power of 2:

$$L=2^k$$

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

- The number, b, of bits required to store a digitized image is

$$b=M \times N \times K$$

When M=N, this equation becomes

$$b=N^2k$$

- When an image can have 2^k intensity levels, then the image is called as **k-bit image.**

For example, an image with 256 possible discrete intensity values is called an 8-bit image.

- The range of values spanned by the gray scale is referred to as the **dynamic range**.
- In other words, The ***dynamic range*** of an imaging system is defined as the ratio of the maximum measurable intensity to minimum detectable intensity level in the system.
- The upper limit is determined by ***saturation*** and the lower limit by ***noise*.**
- Basically dynamic range establishes the lowest and highest intensity levels that image can have.
- **Image contrast** is the difference in intensity between the highest and lowest intensity levels in an image.
- High dynamic range: image has high contrast;
- Low dynamic range: image has dull, washed-out gray look.

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 1/2W line 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.
- Dots per unit distance are a measure of image resolution used commonly in the printing and publishing industry. Newspapers are printed with a resolution of 75 dpi, magazines at 133 dpi, and glossy brochures at175 dpi, book page at 2400 dpi.
- ***Gray-level (intensity) resolution*** **refers to the smallest discernible change in gray level.**
- Due to hardware considerations, the number of gray levels is usually an integer power of 2. The common practice is the number of bits used to quantize intensity as the intensity or gray-level resolution. The most common number is 8 bits, with 16 bits being used in some applications where enhancement of specific gray-level ranges is necessary.
- **BIT DEPTH** is determined by the number of bits used to define each pixel. The greater the bit depth, the greater the number of tones (grayscale or color) that can be represented.

Digital Image Types:

Digital images may be produced in

1. black and white (binary or monochrome or bitonal),
2. gray scale (intensity),
3. color (RGB).

A ***binary image*** is represented by pixels consisting of 1 bit each, which can represent two tones (typically black and white), using the values 0 for black and 1 for white or vice versa.

Eg:
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

A ***grayscale image*** is composed of pixels represented by multiple bits of information, typically ranging from 2 to 8 bits or more.

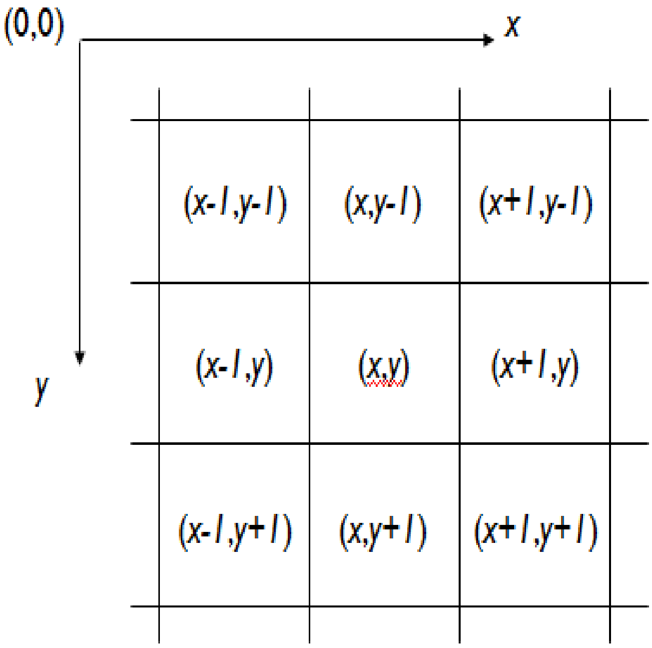
Eg:
$$\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 6 & 26 & 37 \\ 15 & 25 & 13 & 22 \\ 32 & 15 & 87 & 39 \end{bmatrix}$$

Example: In a 2-bit image, there are four possible combinations: 00, 01, 10, and 11. If "00" represents black, and "11" represents white, then "01" equals dark gray and "10" equals light gray. The bit depth is two, but the number of tones that can be represented is 2^2 or 4. At 8 bits, 256 (2^8) different tones can be assigned to each pixel.

A ***color image*** is typically represented by a bit depth ranging from 8 to 24 or higher. With a 24-bit image, the bits are often divided into three groupings: 8 for red, 8 for green, and 8 for blue. Combinations of those bits are used to represent other colors. A 24-bit image offers 16.7 million (2^{24}) color values. Increasingly scanners are capturing 10 bits or more per color channel and often outputting 8 bits to compensate for "noise" in the scanner and to present an image that more closely mimics human perception.

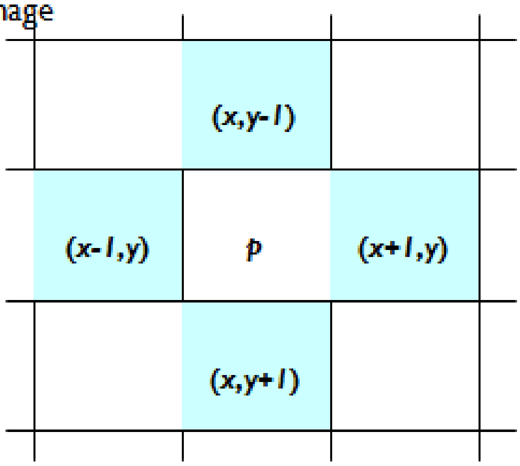
Eg:
$$R=\begin{bmatrix} 99 & 70 & 56 & 78 \\ 60 & 90 & 96 & 67 \\ 85 & 85 & 43 & 92 \\ 32 & 65 & 87 & 99 \end{bmatrix} \quad G=\begin{bmatrix} 65 & 70 & 56 & 43 \\ 32 & 54 & 96 & 67 \\ 21 & 54 & 47 & 42 \\ 54 & 65 & 65 & 39 \end{bmatrix} \quad B=\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 6 & 26 & 37 \\ 15 & 25 & 13 & 22 \\ 32 & 15 & 87 & 39 \end{bmatrix}$$

Some basic relationships between pixels:



Conventional indexing method

4-neighbors : A pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by (x+1, y), (x-1, y), (x, y+1), (x, y-1). This set of pixels, called the *4-neighbors of p*, is denoted by N_4 (p). Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image



4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{matrix} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{matrix} \right\}$$

Neighborhood relation is used to tell adjacent pixels. It is useful for analyzing regions.

4-neighborhood relation considers only vertical and horizontal neighbors.

Note: $q \in N_4(p)$ implies $p \in N_4(q)$

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 N_D (p).

(x-1,y-1)		**(x+1,y-1)**
	p	
(x-1,y+1)		**(x+1,y+1)**

Diagonal neighbors of *p*:

$$N_D(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x+1,y-1) \\ (x-1,y+1) \\ (x+1,y+1) \end{array} \right\}$$

Diagonal -neighborhood relation considers only diagonal neighbor pixels.

8- neighbors The four diagonal neighbors together with the 4-neighbors, are called the *8- neighbors of p*, denoted by N_8 (p).

(x-1,y-1)	**(x,y-1)**	**(x+1,y-1)**
(x-1,y)	**p**	**(x+1,y)**
(x-1,y+1)	**(x,y+1)**	**(x+1,y+1)**

8-neighbors of *p*:

$$N_8(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x,y-1) \\ (x+1,y-1) \\ (x-1,y) \\ (x+1,y) \\ (x-1,y+1) \\ (x,y+1) \\ (x+1,y+1) \end{array} \right\}$$

8-neighborhood relation considers all neighbor pixels.

Adjacency, Connectivity, Region, and Boundaries

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 N_D (p) and the set N4 (p) ∩ N4 (q) 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.1.16 (a) for V= {1}.The three pixels at the top of Fig.1.16 (b) show multiple (ambiguous) 8- adjacency, as indicated by the dashed lines. This ambiguity is removed by using m-adjacency, as shown in Fig. 1.16 (c).Two image subsets S1 and S2 are adjacent if some pixel in S1 is adjacent to some pixel in S2.

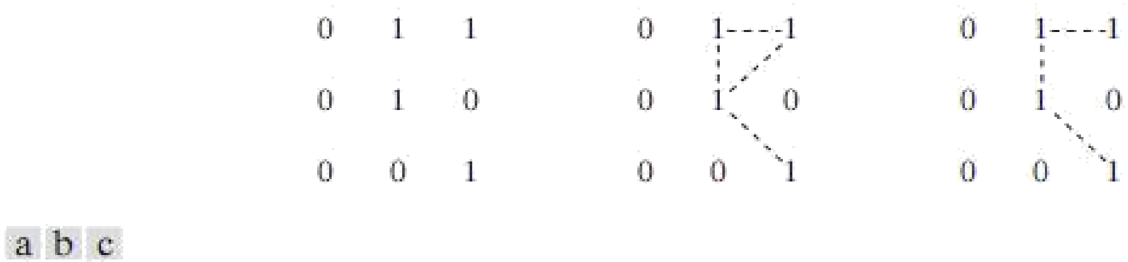


Fig.1.16 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency

Path:

A ***path*** from pixel p at (x,y) to pixel q at (s,t) is a sequence of distinct pixels:

$$(x_0,y_0), (x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)$$

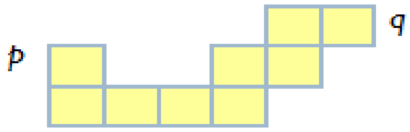
such that

$$(x_0,y_0) = (x,y) \text{ and } (x_n, y_n) = (s,t)$$

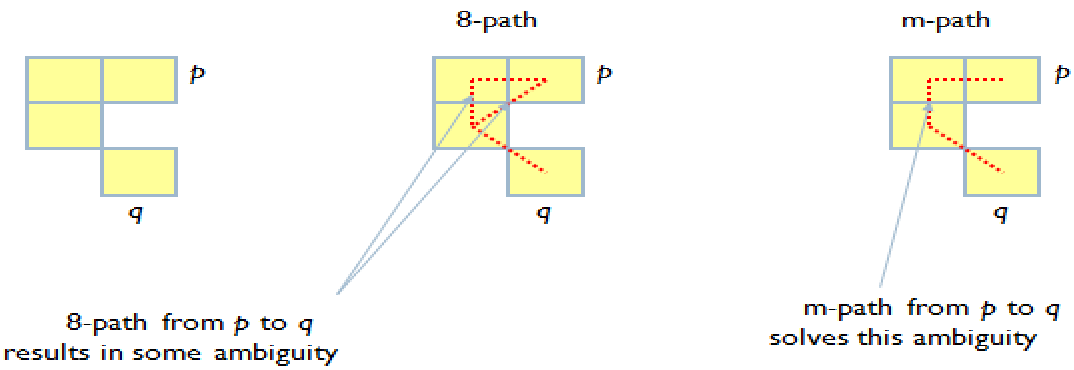
and

$$(x_i, y_i) \text{ is adjacent to } (x_{i-1},y_{i-1}), \quad i = 1,\ldots,n \; (n \text{ is length of the path})$$

If $(x_0,y_0)= (x_n,y_n)$ ➔the path is a closed path



We can define types of path: 4-path, 8-path or m-path depending on the type of adjacency specified.



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***.

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.

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) $D(p, q) \geq 0$ $\quad$ $(D(p, q) = 0$ if $p = q)$

(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
$De(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$
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

$$D4(p, q) = |x - s| + |y - t|$$

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

$$D8(p, q) = \max(|x - s|, |y - t|)$$

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).

Mathematical tools/operations applied on images:

Array versus Matrix Operations

An array operation involving one or more images is carried out on a *pixel-by-pixel* basis.

- Consider two 2 x 2 images
  $$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$
- Array Product is:
  $$\begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$
- Matrix Product is:
  $$\begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Ex for array operation: i) raising an image to a power:Individual pixel is raised to that power.
ii) Dividing an image by another: Division is between corresponding pixel pairs.

Linear versus Nonlinear Operations:

Consider a general operator, H that produces an output image, g (x, y), for a given input image, f(x, y):

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

H is said to be a *linear operator* if

$$H [a_i f_i(x, y) + a_j f_j(x, y)] = a_i H [f_i(x, y)] + a_j H [f_j(x, y)]$$

$$= a_i g_i(x, y) + a_j g_j(x, y)$$

i.e. it satisfies *additive* and *homogeneity* properties, otherwise it is said to be *non-linear operator.*

Example 1:

- Suppose H is the sum operator,
- $\sum[a_i f_i(x, y) + a_j f_j(x,y)] = \sum a_i f_i(x,y) + \sum a_j f_j(x,y)$

$$= a_i \sum f_i(x,y) + a_j \sum f_j(x,y)$$

$$= a_i g_i(x,y) + a_j g_j(x,y)$$

Thus, operator is linear.

Example 2:

- Consider ***max*** operation,
- Let f1 = $\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}$, f2= $\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$, a1 = 1, a2 =-1.
- To Test Linearity,
- LHS of eq(i): $max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = -2$
- RHS of eq(i): $(1)max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\}$ = -4
- LHS ≠ RHS
- So, ***max*** is non-linear operation.

Arithmetic Operations:

- Arithmetic operations are array operations that are carried out between corresponding pixel pairs.

Four arithmetic operations:

- s(x, y) = f(x, y) + g(x, y): reduces noise by averaging set of noisy images.
- d(x, y) = f(x, y) - g(x, y):Enhancement of difference between images, mask mode radiography
- p(x, y) = f(x, y) * g(x, y):Shading correction, region of interest operations
- v(x, y) = f(x, y) / g(x, y)

Where, x = 0,1,2,…,M-1, y = 0,1,2,….N-1.

All images are of size M (rows) x N (columns).

Set and Logical Operations:

Basic Set operations

1. Set: Composed of ordered elements. Ex: A = {coordinates of pixels representing regions in an image}
2. *Null* or *Empty* set
3. *Intersection*
4. *Union*
5. *Disjoint* or *Mutually exclusive*
6. *Set Universe*
7. *Complement of a Set*
8. *Difference of two sets*

Let A - set composed of ordered pairs of real numbers.

If pixel a = (x,y), is an element of A

$$a \in A$$

If a is not an element of A

$$a \notin A$$

Set with no elements is called the null or empty set

$$\varnothing.$$

- If every element of a set *A* is also an element of a set *B*, then *A* is said to be a *subset* of *B*

$$A \subseteq B$$

- *Union* of two sets *A* and *B*

$$C = A \cup B$$

- *Intersection* of two sets *A* and *B*

$$D = A \cap B$$

- Two sets *A* and *B* are *disjoint* or *mutually exclusive* if they have no common elements

$$A \cap B = \varnothing$$

- The *set universe*, *U*, is the set of all elements in a given application.
- *complement* of a set *A* is the set of elements that are not in *A*

$$A^c = \{w \mid w \notin A\}$$

- *difference* of two sets *A* and *B*,

$$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$$

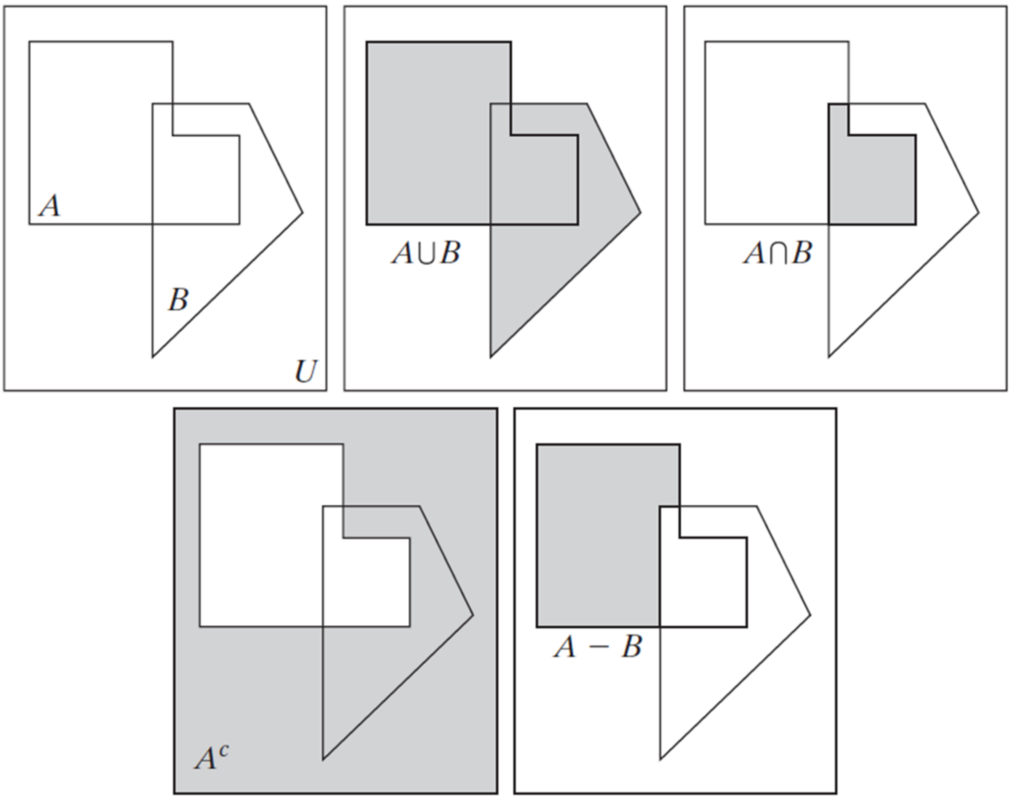


Fig: Illustration of Set Concept

Logical Operations:

1. *NOT*
2. *AND*
3. *OR*
4. *XOR*

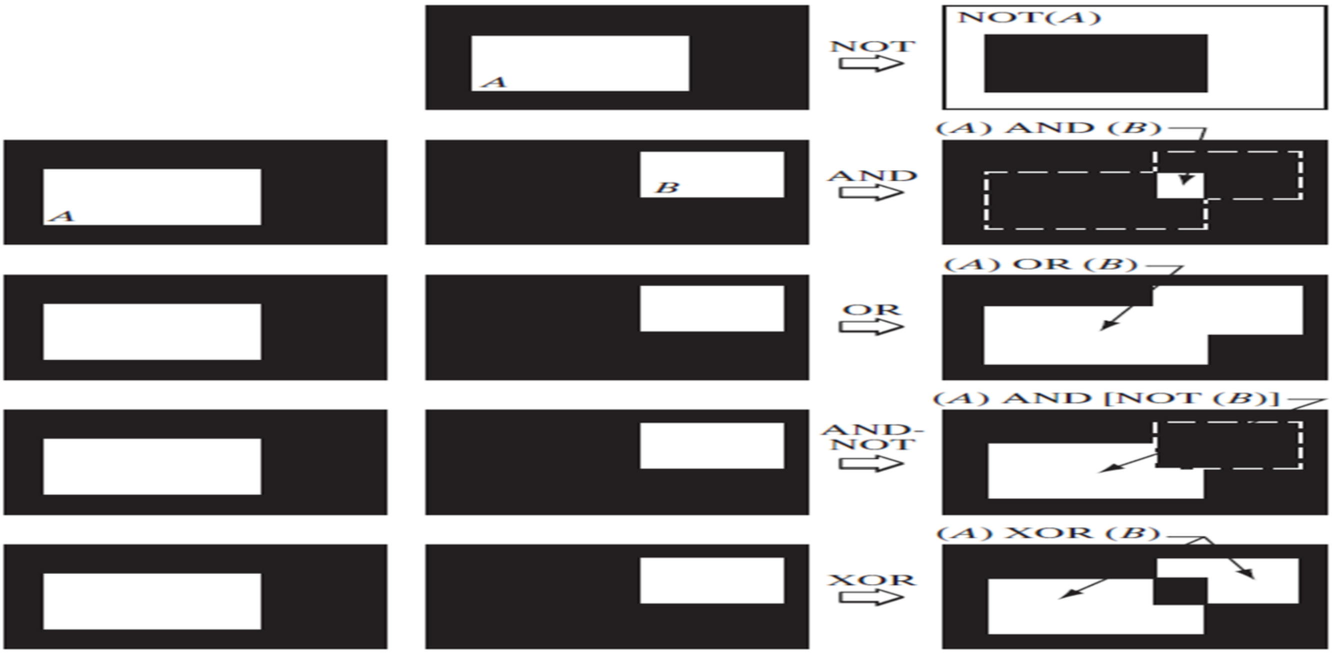


Fig: Illustration of Logical Operators

Spatial Operations:

- Spatial operations are performed directly on the pixels of a given image.
  - (1) single-pixel operations,
  - (2) neighborhood operations, &
  - (3) geometric spatial transformations.

Single-pixel operations: to alter the values of its individual pixels based on their intensity

$$s = T(z)$$

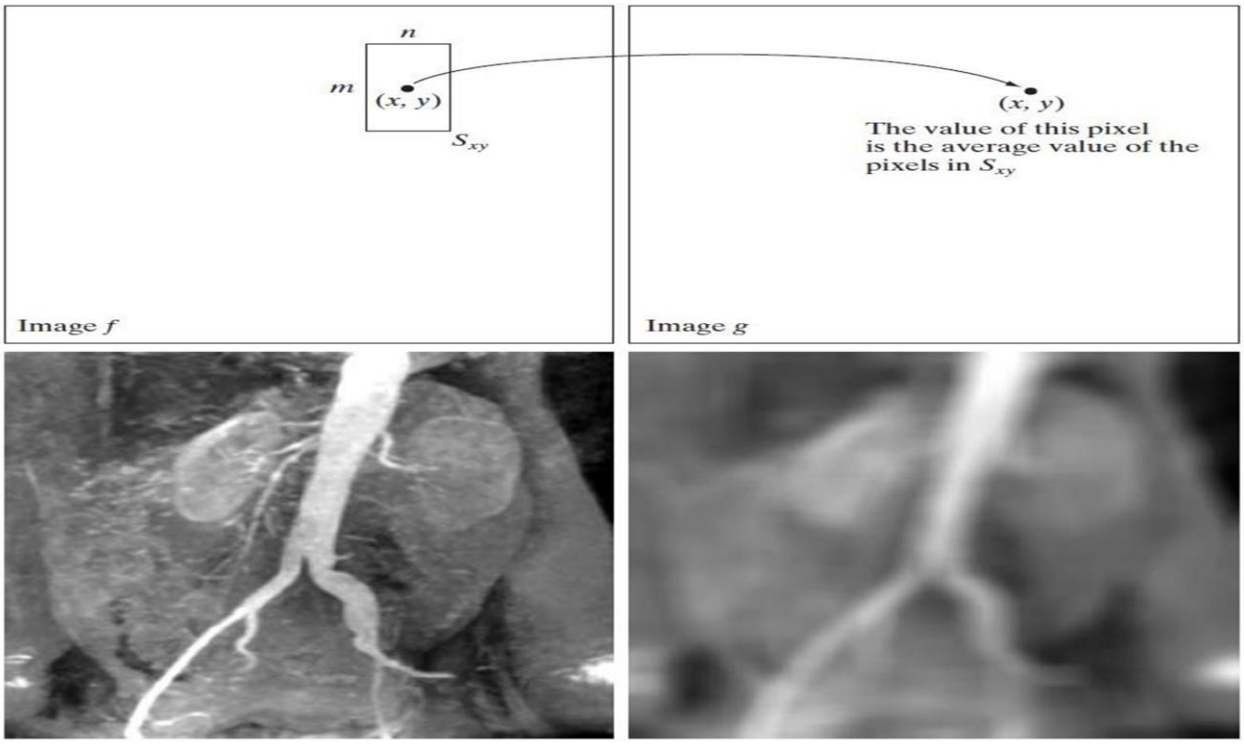
z - intensity of a pixel in the original image

s - (mapped) intensity of the corresponding pixel in the processed image.

Neighborhood operations:

Let S_{xy} denotes set of coordinates of a neighborhood centered on an arbitrary point (x,y) in an image, f. Neighborhood processing generates a corresponding pixel at the same coordinates in an output(processed) image, g, such that the value of the pixel is determined by a specified operation involving the pixels in the input image with coordinates in S_{xy}.

For example, suppose that the specified operation is to compute the average value of the pixels in a rectangular neighborhood of size m x n centered on (x, y). The locations of pixels in this region constitute the set S_{xy}. Fig (a)&(b) illustrate the process.



We can express the operation in equation form as

$$g(x, y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} f(r, c)$$

where *r* and *c* are the row and column coordinates of the pixels whose coordinates are members of the set S_{xy}.

Geometric spatial transformations:

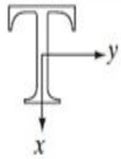
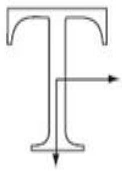
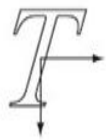
They modify the spatial relationship between pixels in an image.
It is also known as *rubber-sheet* transformations.
They consists of two basic operations:
(1) spatial transformation of coordinates and
(2) intensity interpolation that assigns intensity values to the spatially transformed pixels.
The transformation of coordinates may be expressed as

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

(v, w) - pixel coordinates in the original input image
(x, y) - the corresponding pixel coordinates in the transformed output image.
One of the most commonly used spatial coordinate transformations is the ***affine transform***
Its General Form

$$[x \;\; y \;\; 1] = [v \;\; w \;\; 1]\,\mathbf{T} = [v \;\; w \;\; 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

This transformation can scale, rotate, translate, or shear a set of coordinate points, depending on the value chosen for the elements of matrix **T**.

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = w\cos\theta - v\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Derive transformation matrices for i)Translation ii)Rotation iii)Scaling :

a)Translation:-

Translation is a process of changing the position of an object in a straight-line path from one coordinate location to another. We can translate a two dimensional Point by adding translation distances tx and ty, to the original coordinate position (v, w) to move the point to a new position (x, y) as shown in the figure (a).

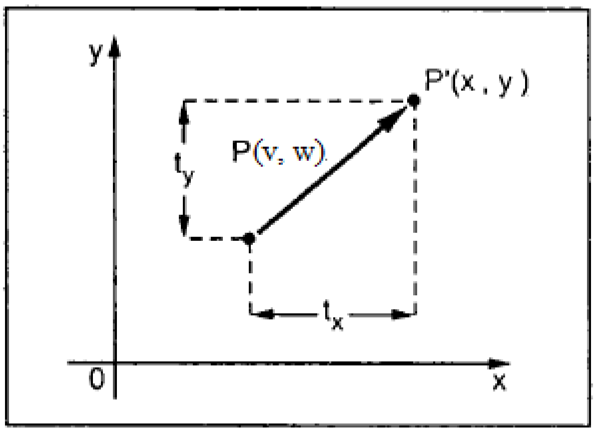


Fig. (a)

x = v + t_x .... (1)

y = w + t_y ... (2)

The translation distance pair (tx , ty) is called a **translation vector or shift vector.**
One of the most commonly used spatial coordinate transformations is the ***affine transform***.
Its General Form

$$[x \;\; y \;\; 1] = [v \;\; w \;\; 1]\,\mathbf{T} = [v \;\; w \;\; 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

$$[x \;\; y \;\; 1] = [v \;\; w \;\; 1]\,\mathbf{T} = [v \;\; w \;\; 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Transformation matrix is:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Rotation:-

A two dimensional rotation is applied to an object by repositioning it along a circular path in the xy plane. To generate a rotation we specify a rotation angle θ and the position of the rotation point about which the object is to be rotated.

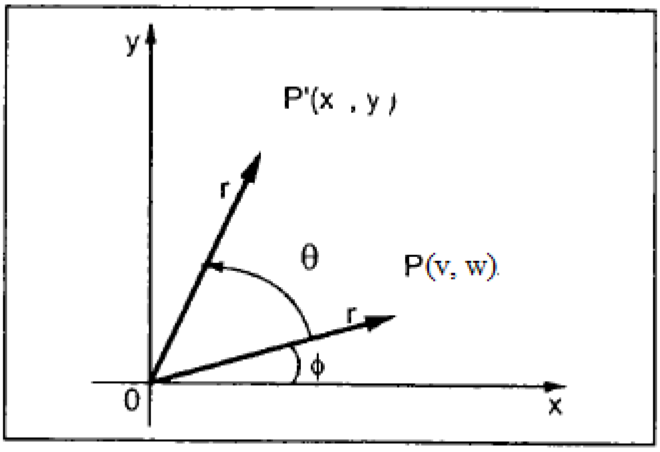


Fig. (b)

Let us consider the rotation of the object about the origin, as shown in the figure (b). Here, r is the constant distance of the point from the origin, angle Ø is the original angular position of the point from the horizontal, and θ is the rotation angle. Using standard trigonometric equations, we can express the transformed coordinates in terms of angles θ and Ø as

$$\left.\begin{aligned} x &= r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta \\ y &= r\sin(\phi+\theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \end{aligned}\right\} \quad \text{...... (4)}$$

The original coordinates of the point in polar coordinates are given as

$$\left.\begin{aligned} v &= r\cos\phi \\ w &= r\sin\phi \end{aligned}\right\} \quad \text{.... (5)}$$

Substituting equations (5) into (4), we get the transformation equations for rotating a point (v,w) through an angle θ about the origin as :

$$\left.\begin{aligned} x &= v\cos\theta - w\sin\theta \\ y &= v\sin\theta + w\cos\theta \end{aligned}\right\} \quad \text{.... (6)}$$

$$[x \ \ y \ \ 1] = [v \ \ w \ \ 1]\,\mathbf{T} = [v \ \ w \ \ 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

$$[x \ \ y \ \ 1] = [v \ \ w \ \ 1]\,\mathbf{T} = [v \ \ w \ \ 1]\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling :-

A scaling transformation changes the size of an object. This operation can be carried out for polygons by multiplying the coordinate values (v,w) of each vertex by scaling factors Cx and Cy, to Produce the transformed coordinates (x, y)

$$\begin{aligned} x &= c_x v \\ y &= c_y w \end{aligned} \quad \textbf{..... (10)}$$

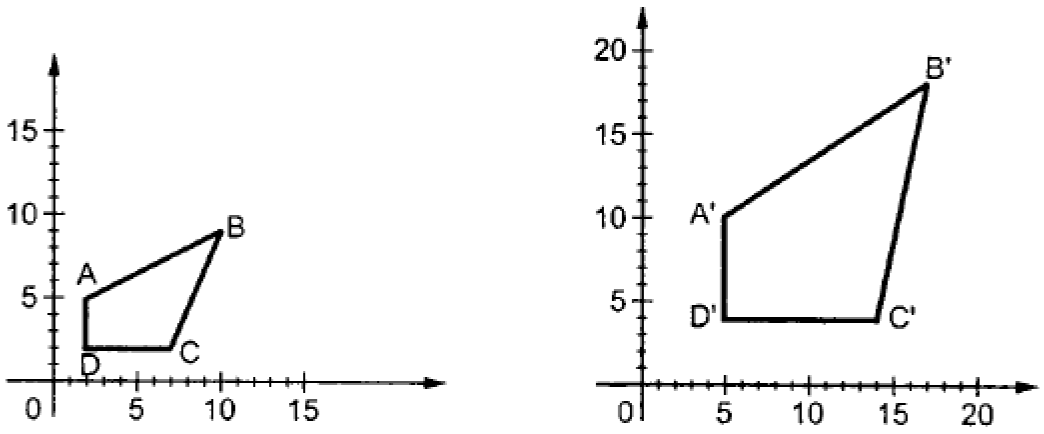


Fig. (c)

Scaling factor Cx scales object in the x direction and scaling factor Cy scales object in the y direction. The equations (10) can be written in the matrix form as given below:

$$[x \ \ y \ \ 1] = [v \ \ w \ \ 1] \ \mathbf{T} = [v \ \ w \ \ 1] \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Any Positive numeric values are valid for scaling factors Cx and Cy. Values less than 1 reduce the size of the objects and values greater than 1 produce an enlarged object. For both Cx and Cy, values equal to 1, the size of object does not change. To get uniform scaling it is necessary to assign same value for Cx and Cy. Unequal values for Cx and Cy, result in a differential scaling.

UNIT – III

IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

INTRODUCTION

The aim of image enhancement is to process an image so that result is more suitable than original image for specific application.

- It is a first step in digital image processing.
- It basically improves the subjective quality of the images by working with the existing data.
- Image enhancement includes gray level and contrast manipulation, noise reduction, edge crispening, and sharpening, filtering, interpolation and magnification , pseudo coloring and so on

Image enhancement techniques can be divided into two broad categories:

1. Spatial domain methods
2. Frequency domain methods

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

In a transform domain, first transforming an image into the transform domain and doing the processing there, and obtaining the inverse transform to bring the results back into the spatial domain.

Spatial Domain Methods

- **types**
- **Intensity transformations (or Point operations)**
- **Spatial filtering (or Neighborhood operations)**

Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression

1.

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

2.

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).

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.3.1 shows.

The center of the sub image is moved from pixel to pixel starting, say, at the top left corner.

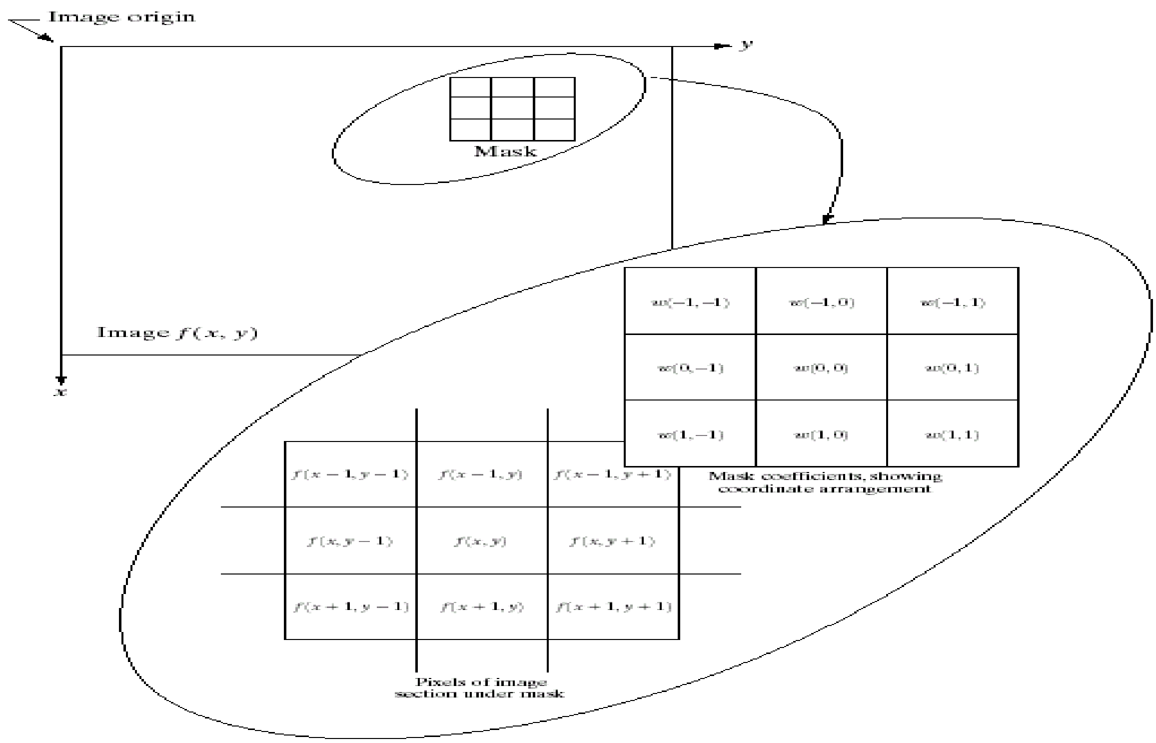


Fig.3.1 A 3*3 neighborhood about a point (x, y) in an image

- The type of operation performed in the neighboring input pixel values is called as Spatial filter or spatial mask or kernel or template or window or neighborhood operation.
- It consists of moving the origin of the neighborhood from pixel to pixel and applying the operator T to the pixels in the neighborhood to yield the output at that location.
- Thus ,for any specific location (x, y), the value of the output image g at those coordinates is equal to the result of applying T to the neighborhood with origin at (x, y) in f.
- The smallest possible neighborhood is of size 1×1. In this ,the g depends only on the value of f at a single point (x, y) and T becomes an intensity (or gray level or mapping) transformation function of the form

$$s=T(r)$$

Where s ->intensity of g at (x, y)
r ->intensity of f at (x, y)

- Enhancement approaches whose results depend only on the intensity at a point are called *point processing techniques.*

Some basic intensity transformation functions (or) point operations:

- Image negatives
- Log transformations
- Power law (Gamma)transformations
- Piecewise-linear transformations
    - --contrast stretching
    - --intensity-level slicing
    - --Bit-plane slicing

3.

Image Negatives

The negative of an image with gray levels in the range [0, L-1] is obtained by using the image negative transformation shown in Fig.3.2, which is given by the expression

$$s = L - 1 - r$$

It reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative.

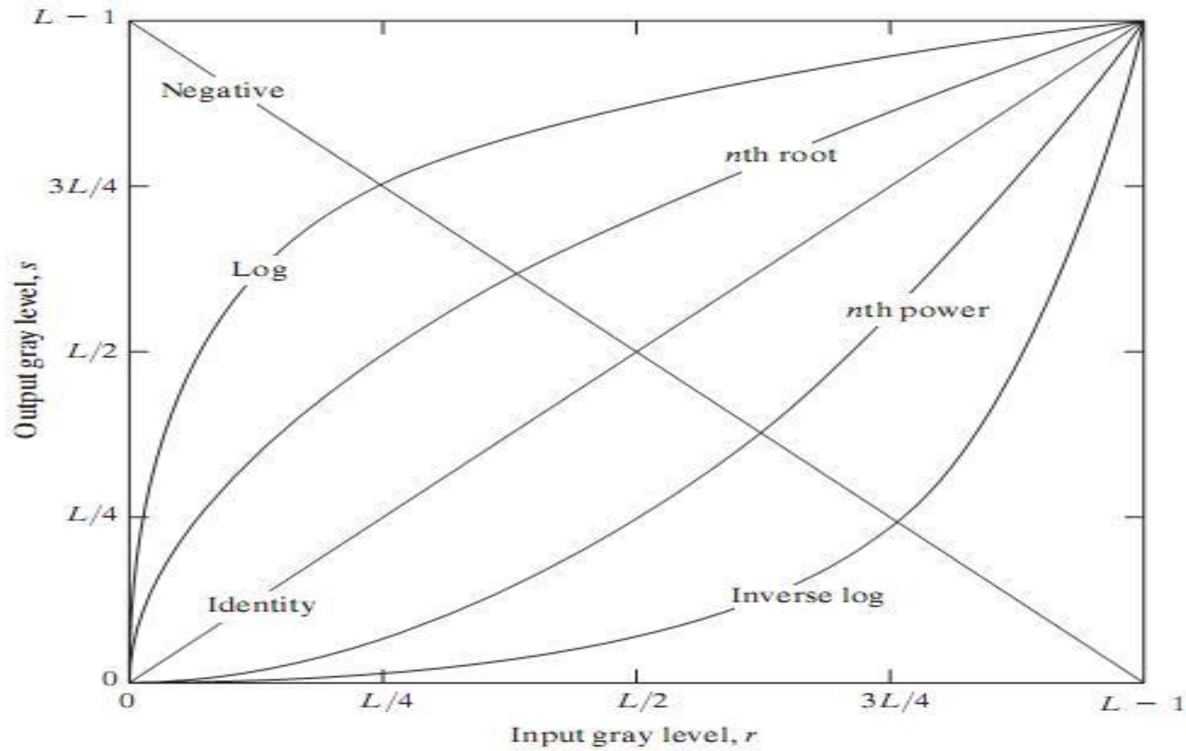


Fig 3.2 Basic Intensity Transformations

Log Transformations

The general form of the log transformation shown in Fig.3.2 is

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

where c is a constant, and it is assumed that $r \geq 0$.The shape of the log curve in 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.

Power-Law Transformations

Power-law transformations have the basic form

$$s = c\, r^{\gamma}$$

where s is the ouput pixel value
r is the input pixel value
c and γ are real numbers

For various values of γ different levels of enhancements can be obtained. This technique is commonly called as gamma correction and used in monitor displays. Curves generated with values of $\gamma>1$ have exactly the opposite effect as those generated with values of $\gamma<1$. We can get identity transformation when $c = \gamma = 1$.

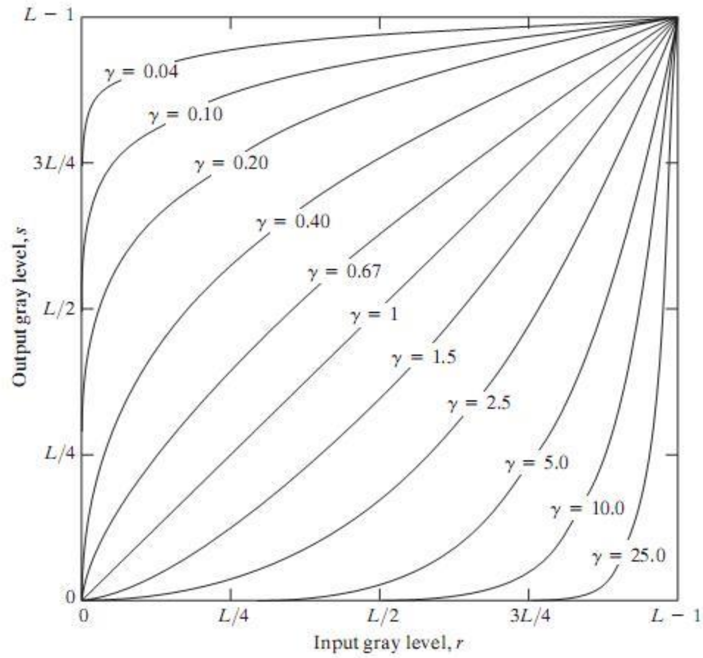


Fig.3.3 Plots of the equation $s = c\, r^{\gamma}$ for various values of γ (c=1 in all cases)

Piecewise-Linear Transformation Functions

Contrast stretching:

One of the simplest piecewise linear functions is a contrast-stretching transformation.

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.

Contrast stretching is a process that expands range of intensity levels in an image.

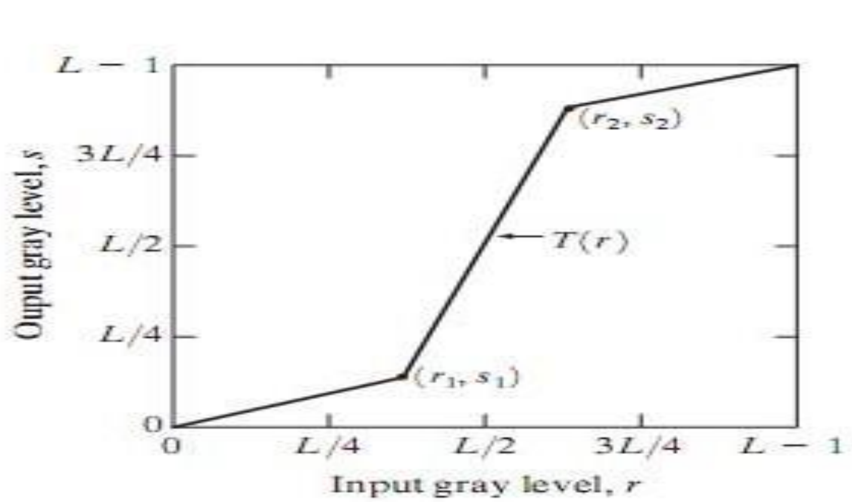


Fig 3.4 :contrast stretching transformation

The transformation used for contrast stretching is shown in fig.(a) .The values (r_1,s_1) and (r_2,s_2) determines shape of transformation.

If $r_1=s_1$ and $r_2=s_2$ we can get linear transformation.

If $r_1= r_2$ and $s_1=0, s_2=L-1$ we can get Thresholding function.

$$S=T(r) = \begin{cases} a_1 r & : 0 \le r < r_1 \quad S_1=T(r_1) \\ a_2 (r-r_1) + S_1 & : r_1 \le r < r_2 \quad S_2=T(r_2) \\ a_3(r-r_2) + S_2 & : r2 \le r \le L-1 \end{cases}$$

Where $a_1, a_2, \& a_3$ control the result of contrast stretching.

If $a_1=a_2=a_3=1$ no change in gray levels.

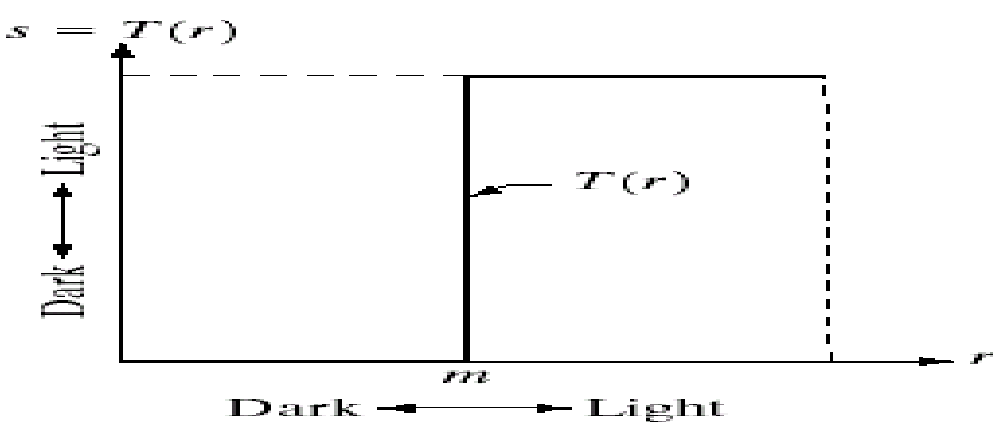
If a1=a2=0 and r1=r2 function is thresholding function then the result is a binary image.

Thresholding function:

Thresholding is required to extract a part of an image which contains all the information. Thresholding is a part of a more general segmentation problem.

In thresholding, pixels having intensity lower than the threshold **T** are set to zero and the pixels having intensity greater than the threshold are set to 255

It generates a binary image



$$\mathbf{S=T(r) = 0 \;\; ;r<m}$$
$$= 1 \;\; ;r>m$$

Gray-level slicing:

Highlighting a specific range of gray levels in an image often is desired.

There are several ways of doing gray 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. The second approach brightens the desired range of gray levels and remaining gray levels are unchanged.

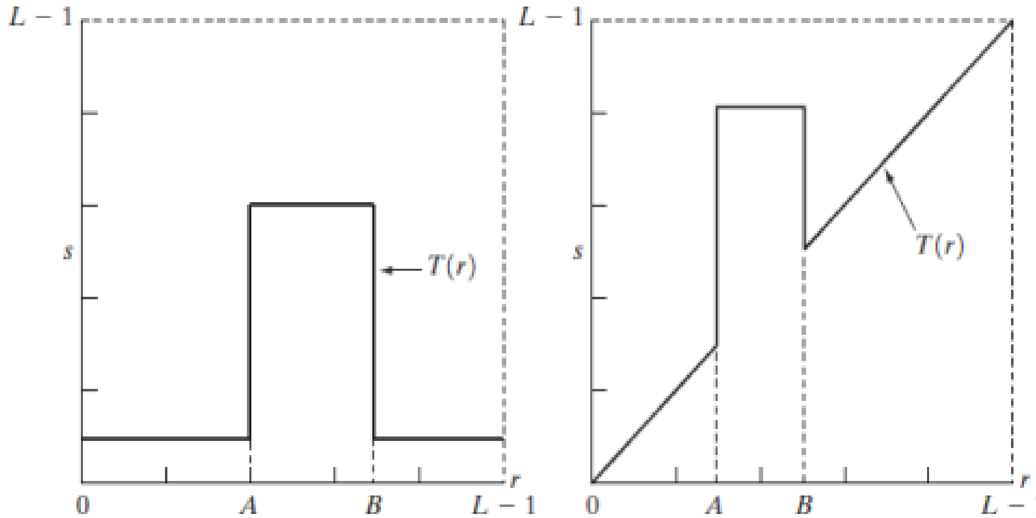


Fig 3.5 (a) This transformation highlights range [A, B] of gray levels and reduce all others to a constant level (b) This transformation highlights range [A, B] but preserves all other levels.

Bit-Plane Slicing:

Suppose that each pixel in an image is represented by 8 bits. The image is composed of eight 1-bit planes, ranging from bit-plane 0 to bit plane 7.

Bit-plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits.

Note that the higher-order bit planes contain the majority of the visually significant data. The other bit planes contains less 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.

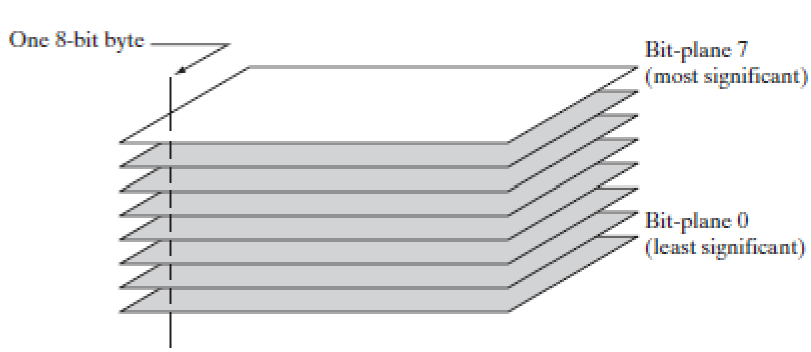


FIGURE
Bit-plane representation of an 8-bit image.

Fig 3.6 bit plane representation of 8-bit image

Histogram Processing:

- The histogram of a digital image with intensity levels in the range [0,L-1]is a discrete function

$$h(r_k)=n_k$$

Where r_k is the Kth intensity value and

n_k is the number of pixels in the image with the intensity r_k

- **Normalized histogram is**

$$P(r_k)= n_k /MN$$

where k=0,1,2,……L-1

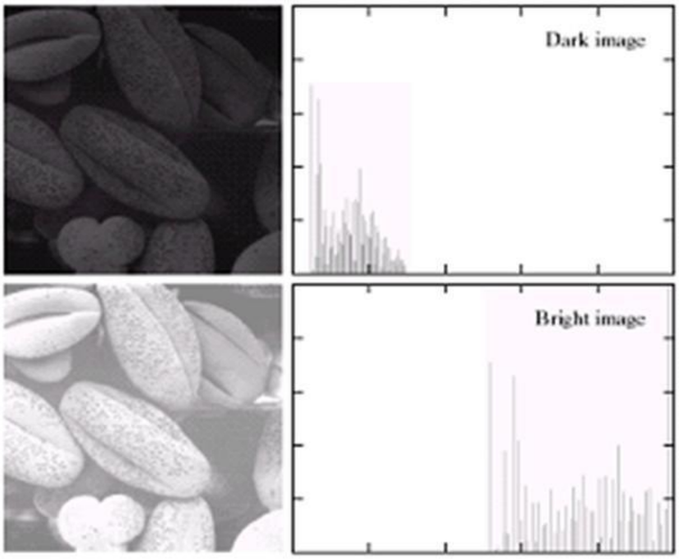
M &N are the row and column dimensions of the image.

- $P(r_k)$ is an estimate of the probability of occurrence of intensity level r_k in an image.

The sum of all components of a normalized histogram is equal to 1.

The histogram plots are simple plots of $h(r_k)=n_k$ versus r_k.

In the dark image the components of the histogram are concentrated on the low (dark) side of the gray scale. Bright image the histogram components are biased towards the high side of the gray scale. The histogram of a low contrast image will be narrow and will be centered towards the middle of the gray scale. The components of the histogram in the high contrast image cover a broad range of the gray scale. The net effect of this will be an image that shows a great deal of gray levels details and has high dynamic range.



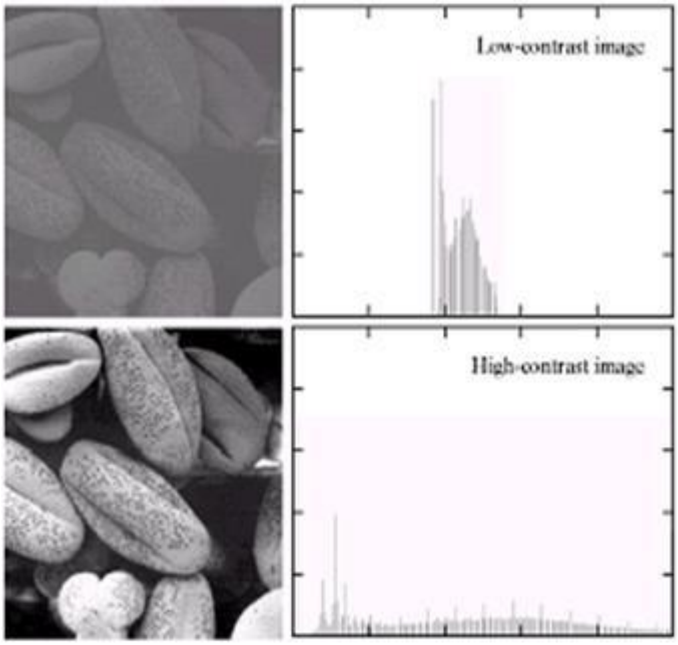


Fig 3.7 Four basic image types : dark, light, low contrast, high contrast images and their corresponding histograms

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 to the interval [0, L-1], with r=0 representing black and r=L-1 representing white. we focus attention on transformations of the form

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

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 $0 \le r \le L-1$; and

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

The inverse transformation from s back to r is denoted

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

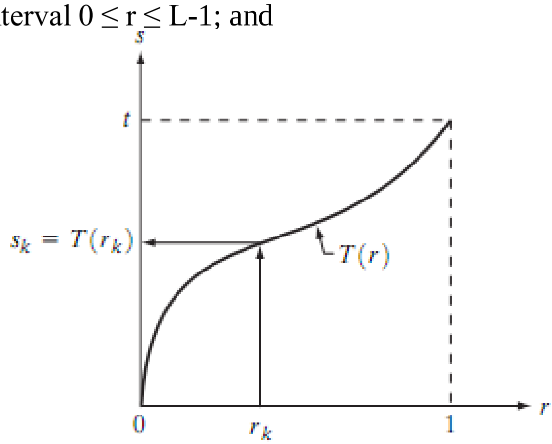


Fig.3.8 A gray-level transformation function that is both increasing.

In discrete version:

- The probability of occurrence of gray level r_k in an image is

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

n : the total number of pixels in the image

n_k : the number of pixels that have gray level r_k

L : the total number of possible gray levels in the image

The transformation function is (called as histogram equalization or linearization transformation)

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

Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k.

Spatial Filtering:

The mechanics of spatial filtering are illustrated in below fig

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

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

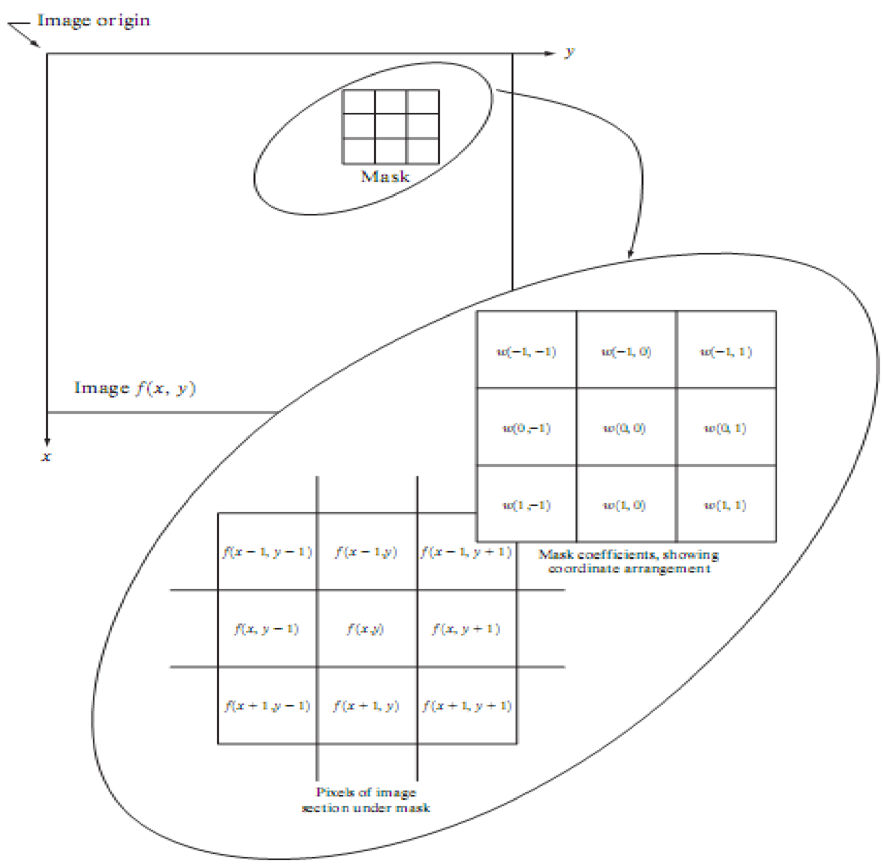


Fig 3.10 the mechanics of linear spatial filtering using 3*3 filter mask

The response R at particular location (x,y) is given by

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

where the w's are filter coefficients, the z's are the values of the image graylevels 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

Fig 3.11. representation of a general 3 x 3 spatial filter mask

- Spatial filtering term is the filtering operations that are performed directly on the pixels of an image. The process consists simply of moving the filter mask from point to point in an image.
  - – Smoothing spatial filters
  - – Sharpening spatial filters

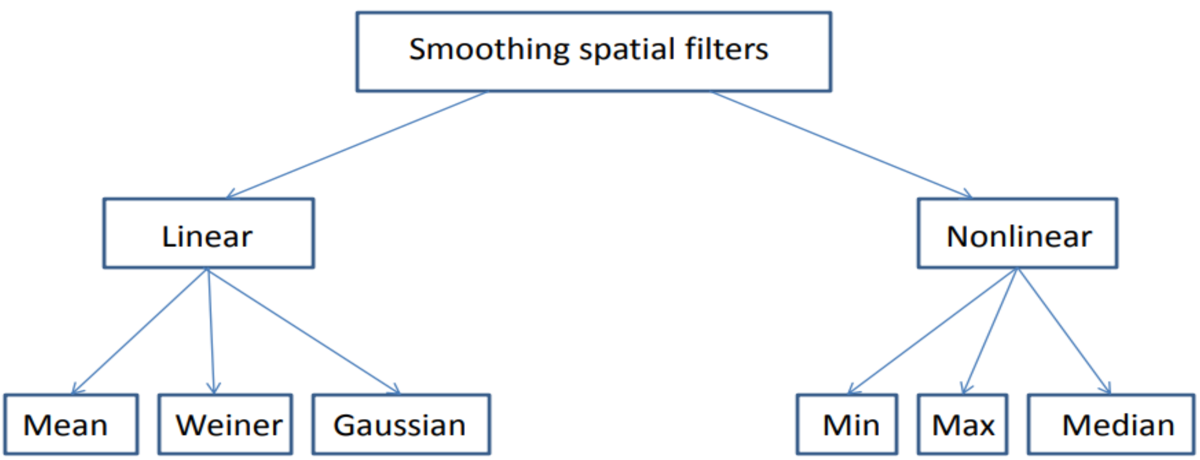
Smoothing Spatial Filters:

(1) Smoothing is often used to reduce noise within an image.

(2) Image smoothing is a key technology of image enhancement, which can remove noise in images. So, it is a necessary functional module in various image-processing software.

(3) Image smoothing is a method of improving the quality of images.

(4) Smoothing is performed by spatial and frequency filters



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.

- Also called averaging filters or Low pass filter.

• By replacing the value of every pixel in an image by the average of the intensity levels in the neighborhood defined by the filter mask.

• Reduced "sharp" transition in intensities.

• Random noise typically consists of sharp transition.

• Edges also characterized by sharp intensity transitions, so averaging filters have the undesirable side effect that they blur edges.

• If all coefficients are equal in filter then it is also called a box filter.

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Fig 3.12 averaging filter

In this filter all weights are equal. This is also called as box filter

Use of the first filter gives average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask

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

Weighted average filter

In this filter, pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others.

In the mask, 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.

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Fig 3.13 weighted averaging filter

It attempt to reduce blurring in the smoothing process.

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

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

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 center pixel by the median of the gray levels in the neighborhood of that pixel.
Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring.
Median filters are particularly effective in the presence of salt-and-pepper noise.
Median represents the 50th percentile of a ranked set of numbers while 100th or 0th percentile results in the so called max filter or min filter respectively

Sharpening Spatial Filters

• Objective of sharpening is to highlight transitions in intensity.
• Uses in printing and medical imaging to industrial inspection and autonomous guidance in military systems.
• Averaging is analogous to integration, so sharpening is analogous to spatial differentiation.
• Thus, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying intensities.

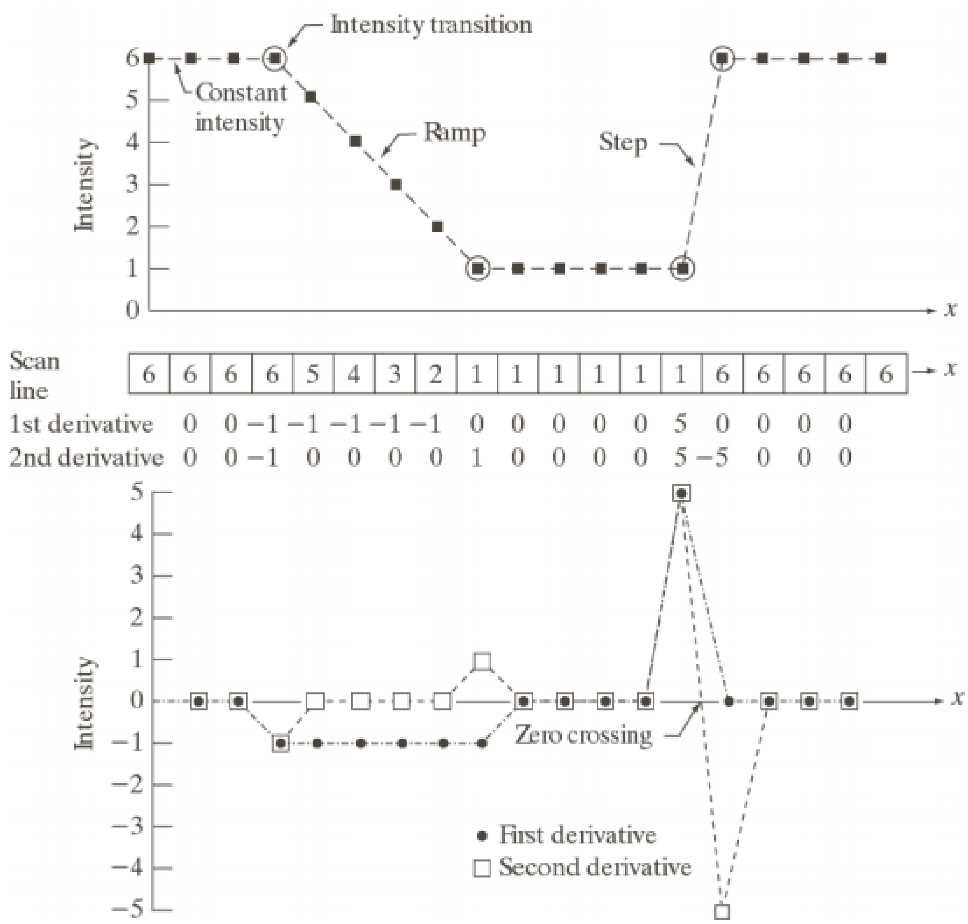
Foundation:

• Definition for a first order derivative
  (1) must be zero in areas of constant intensity
  (2) must be nonzero at the onset of an intensity step or ramp and
  (3) must be nonzero along ramps.

• For a second order derivatives
  (1) must be zero in constant areas
  (2) must be nonzero at the onset and
  (3) must be zero along ramps of constant slope.

• First order derivative of a one dimensional function f(x) is the difference of f(x+1) – f(x).

• Second order = f(x+1) + f(x-1) -2f(x)



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Use of Second Derivatives for image sharpening–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.

Development of the method:

the simplest isotropic derivative operator is the Laplacian, which, for a 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. 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

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

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

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Fig.3.14. (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.

In Laplacian filter sum of all filter coefficients is zero. We can take center coefficient as positive or negative. Laplacian filter highlights edges in an image.

Laplacian is a derivative operator, its use highlights gray-level discontinuities in an image and deemphasizes regions with slowly varying gray levels.

The resultant image g(x,y) by using the Laplacian operator as follows:

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

Unsharp Masking and High boost Filtering

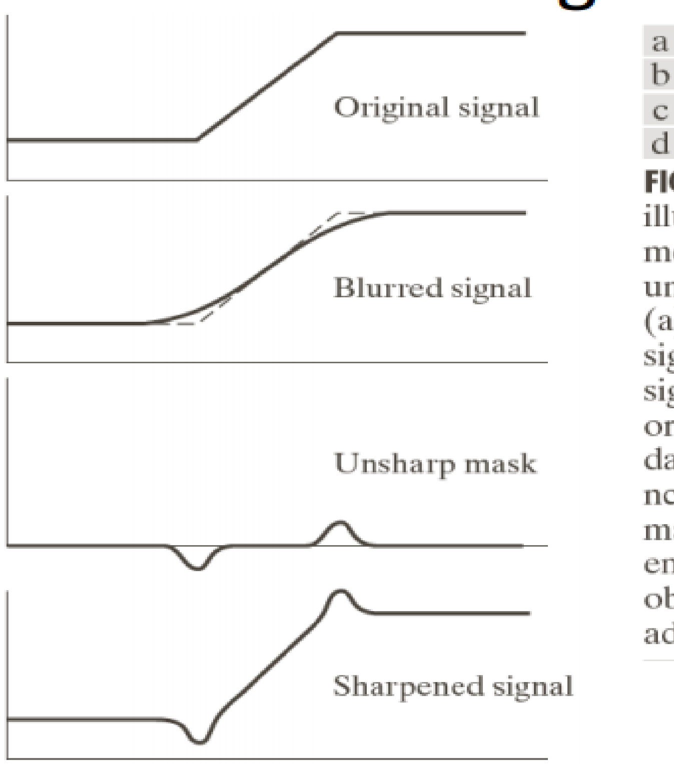


FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for refere-nce. (c) Unsharp mask. (d) Sharp-ened signal, obtained by adding (c) to (a).

• Un sharp Masking
– Read Original Image f(x,y)
– Blurred original image f'(x,y)
– Mask = f(x,y) – f'(x,y)
– g(x,y) = f(x,y) + Mask

• High Boost Filtering
– Read Original Image f(x,y)
– Blurred original image f'(x,y)
– Mask = f(x,y) – f'(x,y)
– g(x,y) = f(x,y) + k*Mask, where k>1

Use of First Derivatives for image sharpening -The Gradient:

Image sharpening can be done by using first order derivatives
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

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

The magnitude of this vector is given by

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

It is common practice to approximate the magnitude of the gradient by using absolute values instead of squares and square roots:

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

Two other definitions proposed by Roberts [1965] in the early development of digital image processing use cross differences:

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

G_x

-1	0
0	1

G_y

0	-1
1	0

Figure 4: Convolution masks for Robert operator.

Fig 3.15 Roberts cross gradient operator

we compute the magnitude of gradient as

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

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

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

Sobel operator

Sobel operator highlights edges in image.
It highlights discontinuities in an image
In sobel operator also sum of all filter coefficients is zero
Idea behind using a weight value of 2 in the center coefficient is to achieve some smoothing

$$G_x = (Z_7 + 2 * Z_8 + Z_9) - (Z_1 + 2 * Z_2 + Z_3)$$

$$G_y = (Z_3 + 2 * Z_6 + Z_9) - (Z_1 + 2 * Z_4 + Z_7)$$

−1	−2	−1
0	0	0
1	2	1

GX

−1	0	1
−2	0	2
−1	0	1

Gy

Fig 3.16 filter masks of sobel operator