DIGITAL IMAGE PROCESSING

UNIT – 1

Why do we need Image Processing?

- To improve the Pictorial information for human interpretation
    1) Noise Filtering
    2) Content Enhancement
        a) Contrast enhancement
        b) Deblurring
    3) Remote Sensing
- Processing of image data for storage, transmission and representation for autonomous machine perception

What is Image?

An image is 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 intensity or gray level of the image at that point. ***When x, y and the intensity values of 'f 'are all finite, discrete quantities then the image is called Digital Image***

Analog Image- An analog image is mathematically represented as a continuous range of values that give the position and intensity.

Digitization –it's the process of transforming images such as analog image into digital image or digital data.

PIXELS:

A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called Picture elements or Image elements, pels and **Pixels**.

What is Digital Image Processing?

Digital image processing is a method to perform some operations on an image. In order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics / features associated with that image.

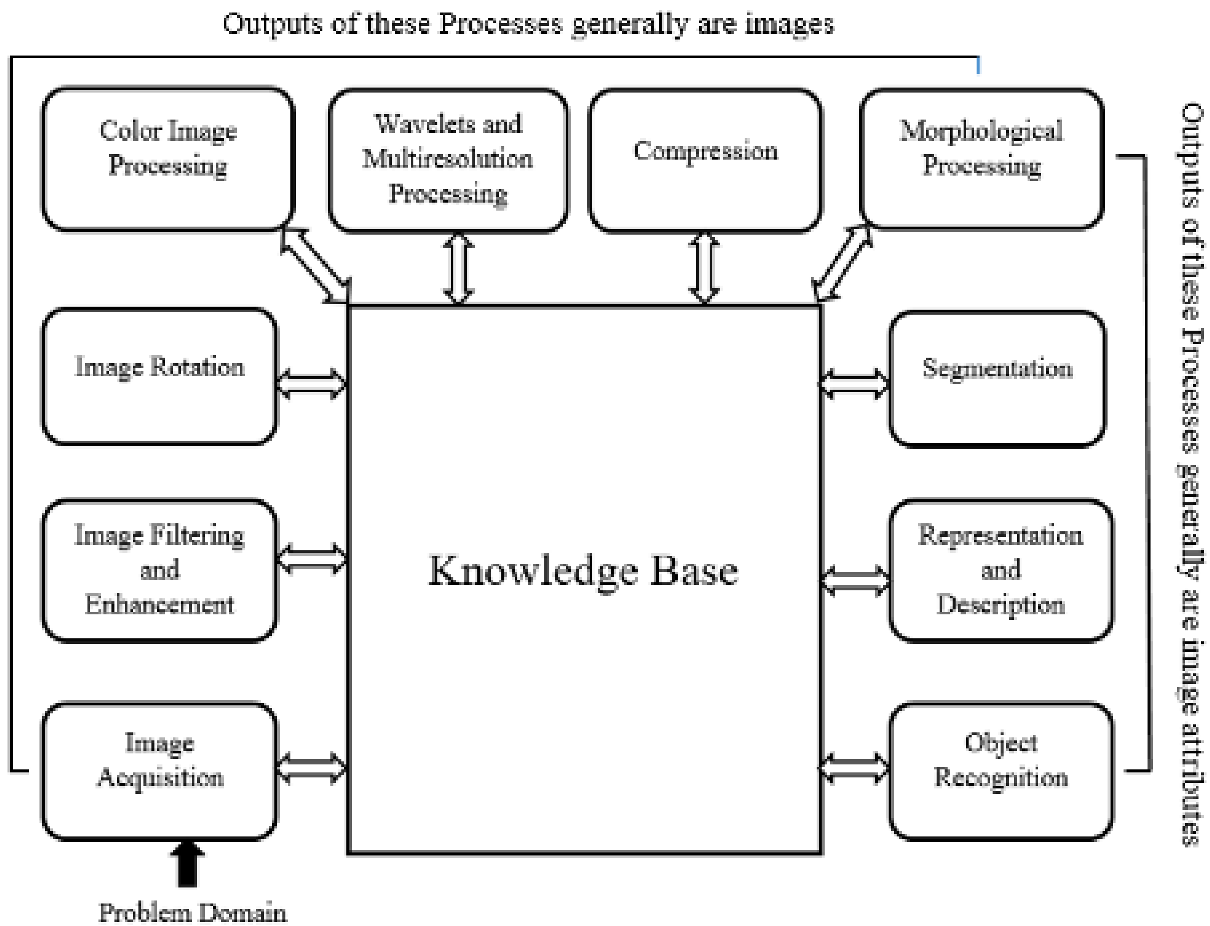
(or)

Digital image processing is defined as the process of analyzing and manipulating images using computer

The main advantage of DIP:

- It allows wide range of algorithms to be applied to the input data.
- It avoids noise and signals distortion problems.

1.1 Fundamentals of Digital Imaging:



1.1.1 Image Acquisition:

Image acquisition is the process of acquiring or getting an image. The entire processing has been done on images so that, the images are first needed to be loaded to the digital computer. Eg: Digital camera, Scanner…etc.,

1.1.2 Image Enhancement:

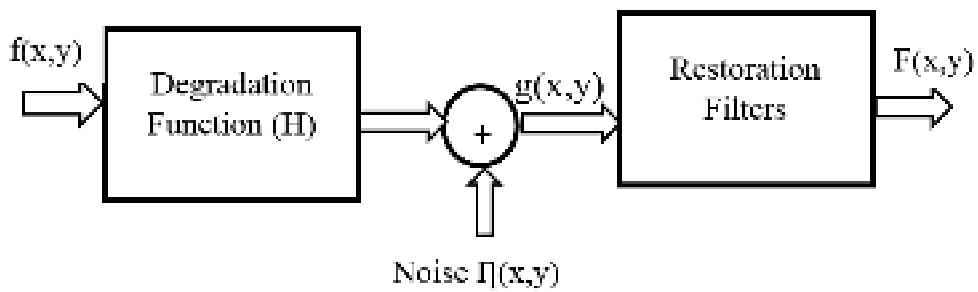
Image enhancement techniques have been widely used in many applications of image processing, where the subjective quality of image is important for human interpretation. *Image enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application.*

- It accentuates or sharpens image features such as edges boundaries or contrast to make a graphic display more helpful for display and analysis.

- The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the chosen features so that they can be detected easily.
- The greatest difficulty in image enhancement is quantifying the criterion for enhancement and therefore, a large number of image enhancement techniques are empirical and require interactive procedures to obtain satisfactory results
- Image enhancement method can be based on either spatial or frequency domain techniques, some examples of image enhancement techniques are,
    - Point operations
    - Spatial operations
    - Transform operations
    - Pseudo coloring

1.1.3 Image Restoration:

In many applications (e.g., satellite imaging, medical imaging, astronomical imaging, poor-quality family portraits) the imaging system introduces a slight distortion. Often images are slightly blurred and image restoration aims at deblurring the image.However, image 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.*



$$g(x,y) = H[f(x,y) + \eta(x,y) \qquad \ldots\ldots\ldots(1)$$

1.1.4 Color Image processing:

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. The use of color image processing is motivated by two principle factors

1) Color is a powerful descriptor that often simplifies object identification and extraction from a scene
2) Human can distinguish thousands of color shades and intensities compare to about only two dozen shades of gray

1.1.5 Wavelets:

Wavelets is a powerful tool in image processing, *It's a mathematical function used for representing images in various degrees of resolution*. It was very useful in Image compression and removal of noise.

1) The wavelet compressed image can be as small as about 25% the size of the similar quality image
2) The wavelets are used to remove the noises present in the image with greater efficiency when compared to other filtering techniques

Wavelets can be combined using a reverse, shift, multiply and integrate techniques called convolution with portions of a known signal to extract information from the unknown signal.

1.1.6 Image Compression:

*Image compression is a technique used for reducing the storage required to store/save an image or the bandwidth required to transmit an image***.** Image compression algorithms are basically classified into

1) Lossy compression –loss of information's present in the image during compression
2) Lossless compression –no loss of information's present in the image during compression

Image compression algorithms may take advantage of visual perception and the statistical properties of image data to provide superior results compared with generic compression methods

1.1.7 Morphological Processing:

Morphological processing is a tool for extracting image components that are useful in the representation and description of shape (extracting and describing image component regions)

1.1.8 Image Segmentation:

Segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

1) Threshold based segmentation
2) Edge based segmentation
3) Region based segmentation
4) Clustering techniques
5) Matching

1.1.9 Representation and Description:

Representation- *deals with compaction of segmented data into representation that facilitate the computation of descriptors*

Description- *deals with extracting attributes that result in some quantitative information of interest or basic for differentiating one class of objects from another*

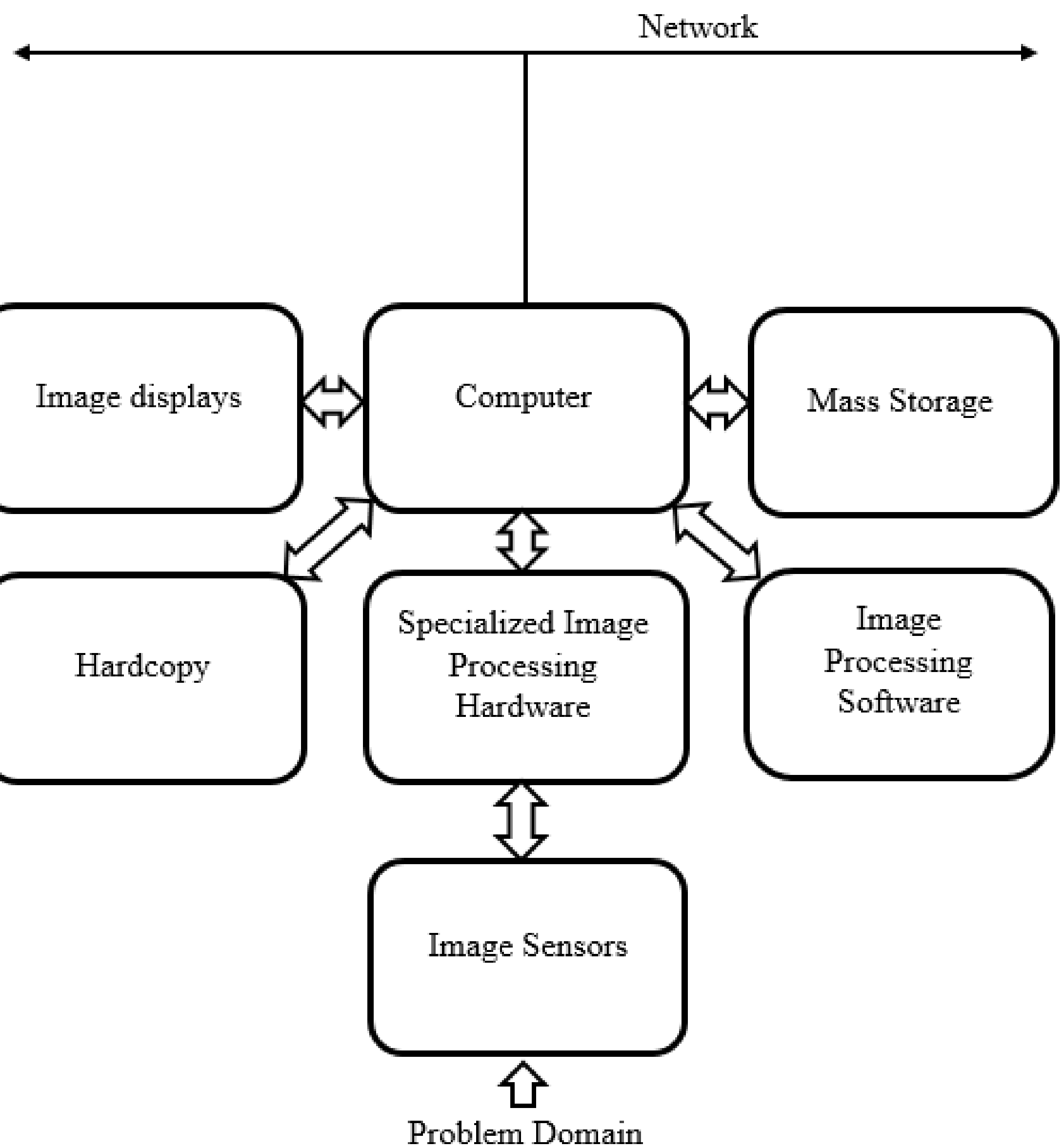
1.1.10 Object Recognition:

Object recognition is the process that assigns a label to an object based on its descriptors

1.1.11 Knowledge Base:

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge data base. This knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information.

1.2 COMPONENTS OF IMAGE PROCESSING SYSTEM:



1.2.1 Image sensors:

Image sensors are used to acquire a digital image, two elements are required to acquire a digital image

1) Physical device - It's sensitive to the energy radiated by the object we wish to image
2) Digitizer – A device for converting output of physical sensing device into digital form.

1.2.2 Image processing software:

The software for image processing has specialized modules which perform specific tasks. Some software packages have the facility for the user to write ode using the specialized modules Eg: MATLAB Software

1.2.3 Specialized image processing hardware:

Image processing hardware performs mostly primitive operations such as an arithmetic logic unit(ALU), that performs arithmetic and logical operations in parallel on entire images. For example, ALU is used as 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.

1.2.4 Computer:

The computer is an image processing system is a general purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes custom computers are used to achieve a required level of performance. In these systems, almost any well-equipped PC-type machine is suitable for off-line image processing tasks.

1.2.5 Software:

Software for image processing consists of specialized modules that perform specific tasks. A well designed package also induces the capability for the user to write a 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.

1.2.6 Mass Storage:

Mass storage capability is a must in image processing applications. For example, an image size of 1024X1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space. When dealing with thousands or even millions of images, providing an adequate storage for image processing can be a challenge. Digital storage for image processing applications falls into three principal categories

1) Short term storage – during processing (Computer memory or buffers)
2) On-line storage – for relatively fast recall (magnetic discs or optical-media storage)
3) Archival storage- infrequent access (magnetic discs or optical disks housed in "jukeboxes")

1.2.7 Image Displays:

Image displays are used for displaying images (eg; color TV monitors). Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. For image display applications, display cards are required and it's a part of the computer system.

1.2.8 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. 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.

1.2.9 Networking:

Networking is a default function in image processing application, because of the large amount of data inherent in image processing applications the key consideration in image transmission is bandwidth. In dedicated networks the bandwidth is not a problem, but communication with remote sites via the internet are not always as efficient. With the help of optical fibers and broadband technologies improving the results.

1.3ELEMENTS OF VISUAL PERCEPTION:

Vision is the most advances human sense. So images play the most important role in visual perception and also the human visual perception is very important because the selection of image processing techniques is based only on visual judgements.

STRUCTURE OF HUMAN EYE:

The human eye is nearly in the shape of a sphere. Its average diameter is approximately 20mm. The eye, called the ***optic globe***is enclosed by three membranes known as,

1) The Cornea and Sclera outer cover
2) The Choroid and
3) The Retina

1.3.1 The Cornea and Sclera outer cover:

- The **Cornea** is a tough, transparent tissue that covers the anterior (Front surface of the eye)
- The **Sclera** is an opaque (not Transparent) membrane that is continuous with the cornea and encloses the remaining portion of the eye.

1.3.2 The Choroid:

- The choroid is located directly below the sclera.
- It has a network of blood vessels which are the major nutrition source to eye. Slight injury to choroid can lead to severe eye damage as it causes restriction of blood flow.
- The outer cover of the choroid is heavily pigmented (colored). This reduces the amount of light entering the eye from outside and backscatter within the optical globe.
- The choroid is divided into two at its anterior extreme as,
    1) The Ciliary body
    2) The Iris Diaphragm

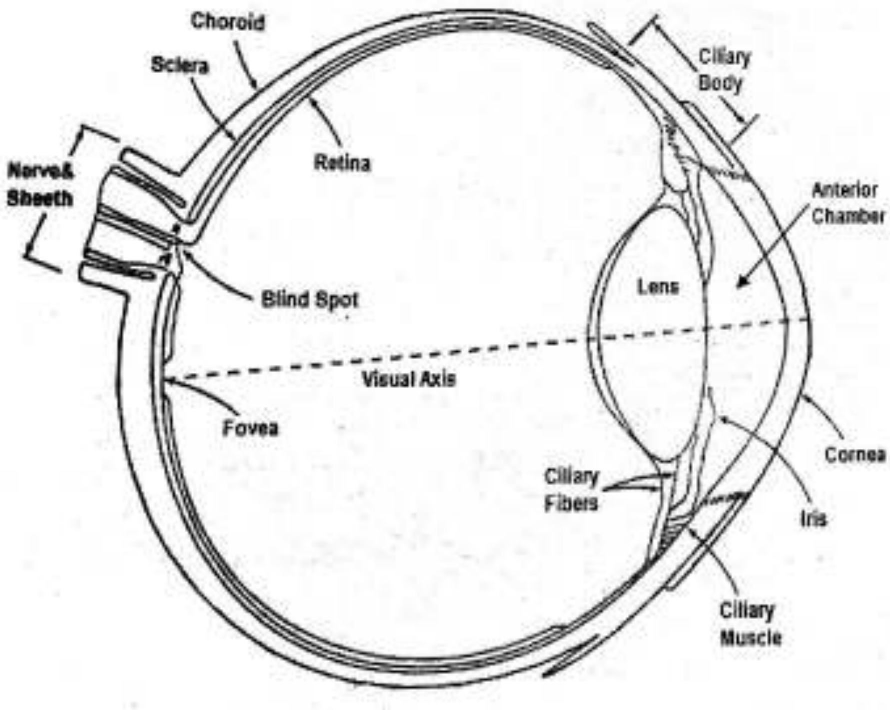


FIG:*Human eye – cross section*

1.3.2.1 The Iris Diaphragm

- It contracts and expands to control the amount of light enters the eye. The central opening of iris is known as ***pupil,*** whose diameter varies from ***2 to 8 mm***.
- The front of the iris contains the visible pigment of the eye and the back has a black pigment.

1.3.2.2 Lens:

- The lens is made up of many layers of fibrous cells. Its suspended (hang up by the fibers attached to the ciliary body) and also it contains 60 to 70% water, 6% fat and more protein.

1.3.2.3Cataracts:

- The lens is colored by a slightly yellow pigmentation. This coloring increases with age, which leads to clouding of lens. ***Excessive clouding*** of lens happens in extreme cases which is known as “Cataracts”.
- This leads to poor color discrimination and loss of clear vision

1.3.3 The Retina:

- The retina is a innermost layer or membrane of the eye. It covers the inside of the walls entire posterior (back portion).
- The central part of the retina is called ***fovea,*** it’s a circular indentation with a diameter of 1.5mm.
- **Light Receptors:** When the eye is properly focused, light from an object outside the eye is imaged on the retina. Light receptors provide this “pattern vision” to the eye. These receptors are distributed over the retina and these receptors are classified into two classes, known as
  - **a)** Cones
  - **b)** Rods

1.3.3.1 Cones:

- In each eye there are ***6 to 7 million cones*** are present. They are highly sensitive to color and are located in the fovea
- Each cone is connected with its own nerve end. Therefore, humans can resolve fine details with the use of cones. Cone vision is called ***photopic or bright-light vision***

1.3.3.2 Rods:

- The number of rods in each eye rages from ***75 to 159 million***. They are sensitive to low level illumination (lightings and are not involved in color vision).
- Many number of rods are connected to a common, single nerve. Thus the amount of detail recognizable is less. Therefore, the rods provide only a general, overall picture of the field of view.
- Rods vision are called scotopic or dim-light vision (Due to stimulation of rods, the objects that appear with bright color in daylight, will appear colorless in moonlight. This phenomenon is called as “scotopic or dim-light vision”)

1.4 DIGITAL CAMERA:

A digital camera that produces digital images that can be stored in a computer, displayed on a screen and printed. The functioning of digital camera is very simple; it allows to take unlimited photographs.

1.4.1 Working Principle: The basic mechanism of digital camera is the technology of converting analog information to digital information. As the smallest unit of an image called a pixel consists of 1's and 0's a digital image is composed of such a sting of 1's and 0's

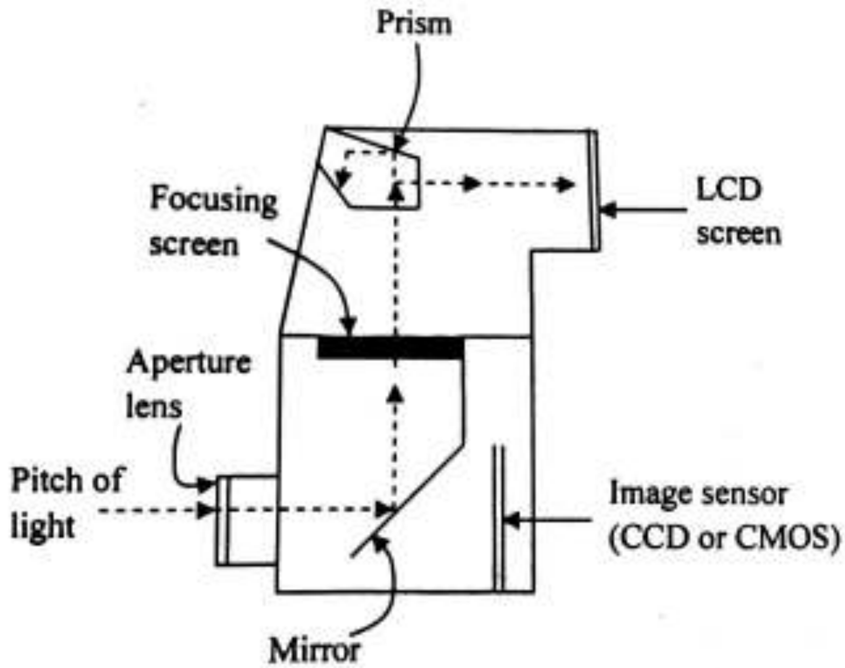


FIG:*Working of Digital Camera*

In a digital camera there are some silicon chips containing ***light sensitive sensors***. These sensors gather light that comes into the camera through the aperture and then convert the data into electrical impulses. These impulses are actually the information about the images. Thus the light is converted into electrons by these sensors and each light sensitive spot on the sensor determines the brightness of the image. But digital cameras have three separate sensors: **Red, Green and Blue.** These three colors are combined in different ratios to form a full color space

1.4.2 Exposure to Light:

Exposure is the duration for which the shutter in a digital camera remains open to allow the light to enter through the aperture. Exposure of the aperture determines how much light will reach the sensor. Also shutter speed or exposure can be controlled manually or can be automatically. The higher the shutter speed, the lesser the light will reach the sensor and vice versa. In order to take picture in ***bright light***, exposure should be less as more light will blur the image and if in ***dark***, exposure should be more in order to allow more light to reach sensors

1.4.3 Focus:

In digital camera focusing helps an image to have better clarity. The focus is based the quality of the lens, because the lens of the camera controls the way of the light is directed towards the sensors. By using a combination of lenses, distance image can be magnified for a better picture.

1.4.4 Photo Storage – Memory:

Digital camera has an internal memory chip that is used to store images that are captured. Internal chips can be supplemented by a removable memory chip for extended storage pace. The memory chip stores the digital information about an image that has been collected within the camera. The storage space required is directly proportional to the size of the image.

1.4.3 Resolutions:

Resolution is defined as the ***amount of detail present in an image.*** In digital camera resolutions determines the amount of detail it can capture. Each digital camera has its own particular resolution. If the resolution of the camera is high, the depth, the clarity and minute details of the picture will be better. eg: 256 x 256 or 4064 x 2704 pixels

1.5 IMAGE THROUGH SCANNER:

A scanner is a device that is used for producing an exact digital image replica of a photo, text written in paper, or even an object. This digital image can be saved as a file to your computer and can be used to alter/enhance the image or apply it to the web.

1.5.1 Types of Scanners:

- **Drum Scanners -** This scanner is mainly used in the publishing industry. The technology used behind the scanning is called a photomultiplier tube (PMT).
- **Flatbed scanners -** Flatbed scanner is the most commonly used scanning machine nowadays. They are also called desktop scanners. They use Charge-coupled device (CCD) to scan the object
- **Hand-Held Scanners -** to scan documents by dragging the scanner across the surface of the document. This scanning will be effective only if with a steady hand technique, or else the image may seem distorted
- **Film Scanners** – to scan positive and negative photographic images. The film will be inserted into the carrier. It will be moved with a stepper motor and the scanning process will be done with a CCD sensor

1.5.2 Working of Flatbed Scanner

Charge-coupled device [CCD] is used in flat bed scanner. A CCD sensor is used to capture the light from the scanner and then convert it into the proportional electrons. The charge developed will be more if the intensity of light that hits on the sensor is more

***Any flatbed scanner will have the following devices*.**

- Charge-coupled device (CCD) array
- Scan head
- Stepper motor
- Lens
- Power supply
- Control circuitry
- Interface ports
- Mirrors
- Glass plate
- Lamp
- Filters
- Stabilizer bar
- Belt
- Cover

Glass plate, Cover:

A scanner consists of a flat transparent glass bed under which the CCD sensors, lamp, lenses, filters and also mirrors are fixed. The document has to be placed on the glass bed. There will also be a cover to close the scanner. This cover may either be white or black in color. This color helps in providing uniformity in the background. This uniformity will help the scanner software to determine the size of the document to be scanned.

Lamp:

The lamp brightens up the text to be scanned. Most scanners use a cold cathode fluorescent lamp (CCFL).

Stepper Motor:

A stepper motor under the scanner moves the scanner head from one end to the other. The movement will be slow and is controlled by a belt.

Scan Head, CCD, Lens, Stabilize bar:

The scanner head consists of the mirrors, lens, CCD sensors and also the filter. The scan head moves parallel to the glass bed and that too in a constant path. As deviation may occur in its motion, a stabilizer bar will be provided to compromise it. The scan head moves from one end of the machine to the other. When it has reached the other end the scanning of the document has been completed. For some scanners, a two-way scan is used in which the scan head has to reach its original position to ensure a complete scan.

As the scan head moves under the glass bed, the light from the lamp hits the document and is reflected back with the help of mirrors angled to one another. According to the design of the device there may be either 2-way mirrors or 3-way mirrors. The mirrors will be angled in such a way that the reflected image will be hitting a smaller surface. In the end, the image will reach a lens which passes it through a filter and causes the image to be focussed on CCD sensors. The CCD sensors convert the light to electrical signals according to its intensity.

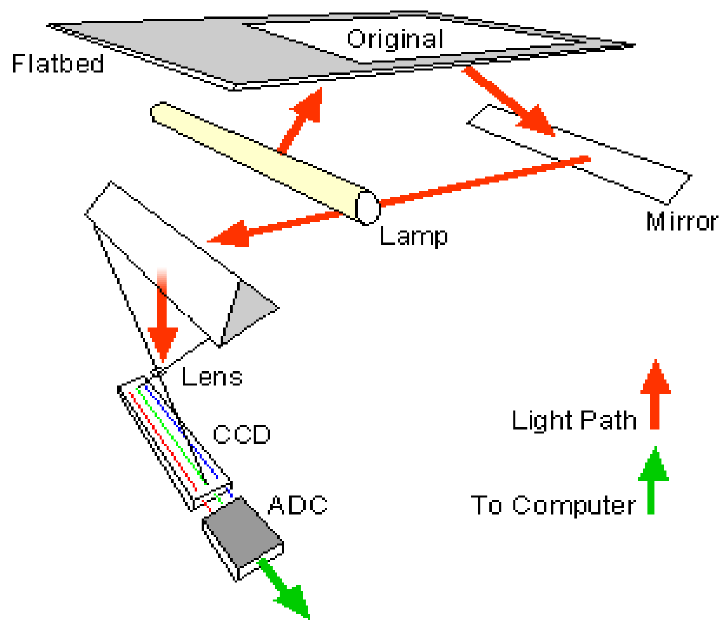


FIG: *Working of Scanner*

The electrical signals will be converted into image format inside a computer. This reception may also differ according to the variation in the lens and filter design. A method called three pass scanning is commonly used in which each movement of the scan head from one end to another uses each composite color to be passed between the lens and the CCD sensors. After the three composite colors are scanned, the scanner software assembles the three filtered images into one single-color image.

There is also a single pass scanning method in which the image captured by the lens will be split into three pieces. These pieces will pass through any of the color composite filters. The output will then be given to the CCD sensors. Thus, the single-color image will be combined by the scanner

1.5 IMAGE SAMPLING AND QUANTIZATION:

In order to become suitable for digital processing, an image function f(x,y) must be digitized both spatially and in amplitude. Typically, a frame grabber or digitizer is used to sample and quantize the analogue video signal. Hence in order to create an image which is digital, we need to covert continuous data into digital form. There are two steps in which it is done:

- Sampling
- Quantization

The sampling rate determines the spatial resolution of the digitized image, while the quantization level determines the number of grey levels in the digitized image. A magnitude of the sampled image is expressed as a digital value in image processing. The transition between continuous values of the image function and its digital equivalent is called quantization.

The number of quantization levels should be high enough for human perception of fine shading details in the image. The occurrence of false contours is the main problem in image which has been quantized with insufficient brightness levels.

- ***Sampling*: Process of digitizing the coordinate values is called sampling**
- ***Quantization*: Process of digitizing the amplitude values is called quantization**

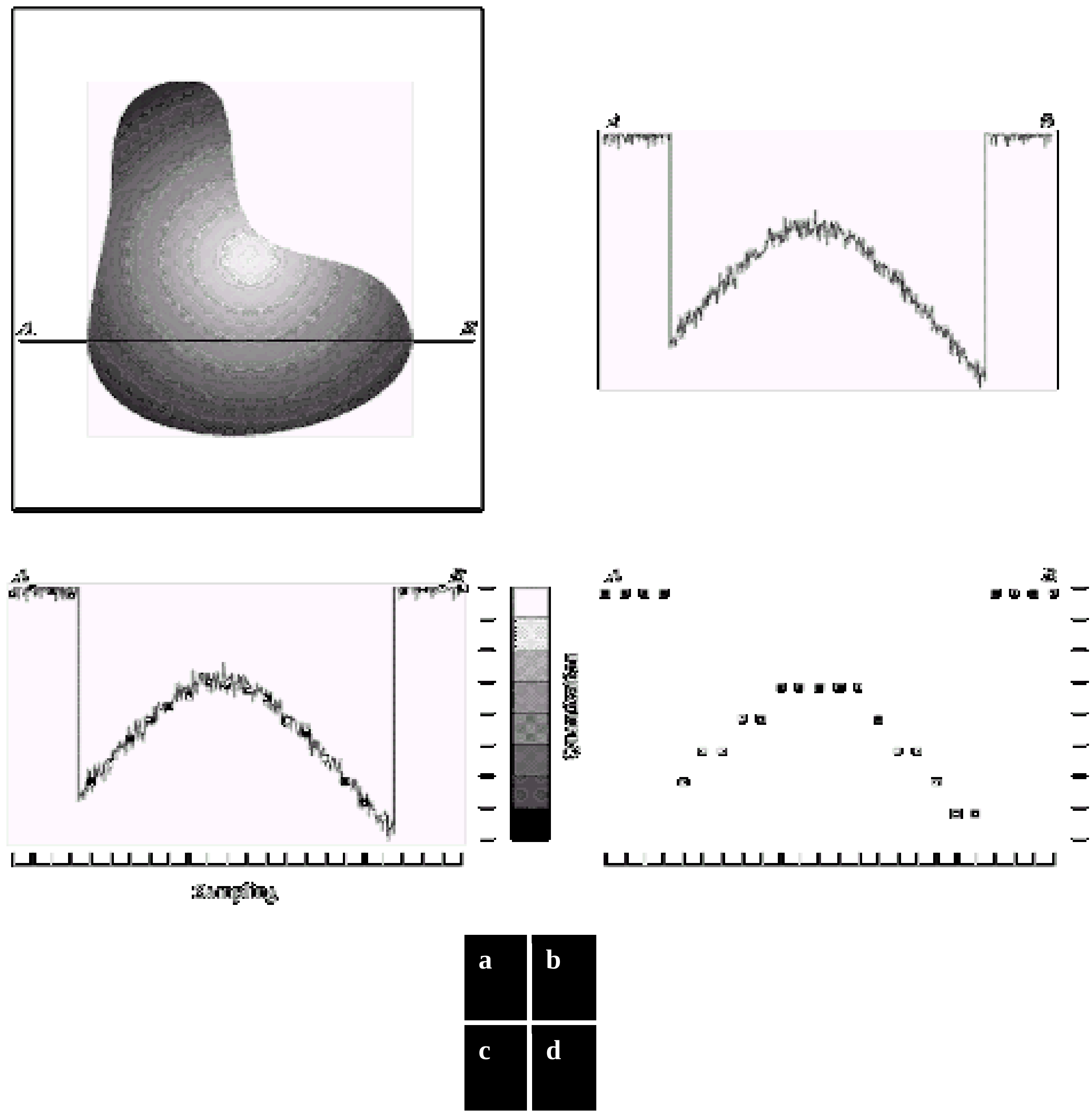
The Basic concepts of image sampling and quantization can be explained with the example given below.

Example:

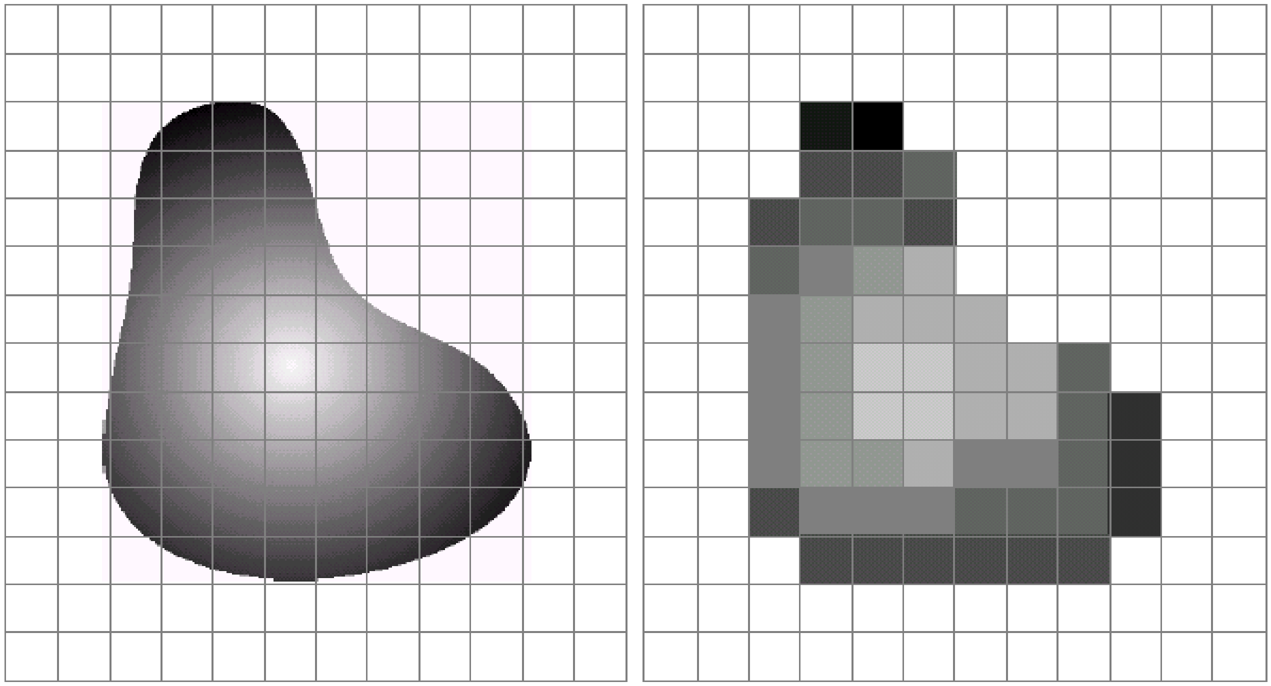
Consider a continuous image f(x,y) shown in figure (a) which is needed to be converted into digital form. Its gray level plot along line AB is given in figure (b). This image is continuous with respect to the x and y coordinates as well as in amplitude. i.e. gray level values. Therefore, to convert into digital form, both the coordinates and amplitude values should be sampled.

To sample this function, equally spaced samples are taken along the line AB. The samples are shown as small squares in figure (c). The set of these discrete locations give the sampled functions.

Even after sampling, the gray level values of the samples have a continuous range. Therfore, to make it discrete, the samples are needed to be quantized. For this purpose, a gray level scale shown at the figure (c) right side is used. It is divided into eight discrete levels, ranging from black to white. Now, by assigning one of the eight discrete gray levels to each sample, the continuous gray levels are quantized



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



a b

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

1.7 RELATIONSHIP BETWEEN PIXELS:

A relation of pixels plays an important role in digital image processing. Where the pixels relations are used for finding the differences of images and also in its sub images.

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

1.7.1 Neighbors of a Pixel:

A pixel, p can have three types of neighbors known as,

1. 4 – Neighbors, $N_4(p)$
2. Diagonal Neighbors, $N_D(p)$
3. 8 – Neighbors, $N_8(p)$

a. 4 - Neighbors, $N_4(p)$

The neighbors of a pixel 'p' at coordinates (x, y) induces two horizontal and two vertical neighbors. The coordinate of these neighbors is given by,

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

Here, each pixel is at unit distance from (x,y) as shown in figure. If (x,y) is on the border of the image, some of the neighbors of pixel ‘p’ lie outside the digital image.

b. Diagonal Neighbors, $N_D(p)$

The coordinates of the four diagonal neighbors of ’p’ are given by

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

Here also, some of the neighbors lie outside the image if (x,y) is on the border of the image

c. 8 - Neighbors, $N_8(p)$

The diagonal neighbors together with the 4-neighbors are called the 8-neighbors of the pixel ‘p’. It’s denoted by $N_8(p)$.

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

1.7.2 Adjacency:

Let {V} be the set of intensity values used to define adjacency. In a binary image V={1} if we are referring to adjacency of pixels with value 1. In a gray-scale 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 intensity values 0 to 255, set V could be any subset of these 256 values. The adjacency has been classified into three types,

1. 4-Adjacency
2. 8-Adjacency
3. m-Adjacency (or) Mixed-Adjacency

Let {V} be the set of gray levels used to define adjacency

a. 4-Adjacency

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

b. 8-Adjacency

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

c. m-Adjacency

Mixed adjacency is a modification of 8-adjacency. It is used to remove the ambiguities present in 8-adjacency.

Two pixels p and q with values from {V} are m-adjacent if the following conditions are satisfied.

- q is in $N_4(p)$.
- q is in $N_D(p)$ and the set $[N_4(p) \cap N_4(q)]$ is empty (has no pixels whose values are from V).

1.7.3 Connectivity:

Two pixels p and q are said to be connected if

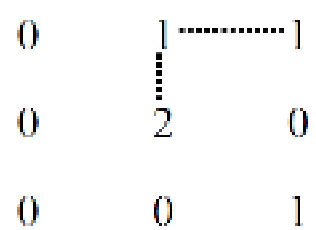
- they are neighbors and
- their gray levels satisfy a specified similarity criterion (E.g: if their gray levels are equal)

The connectivity has been classified into three types,

1. 4-Connectivity
2. 8-Connectivity
3. m-Connectivity (or) Mixed-Connectivity

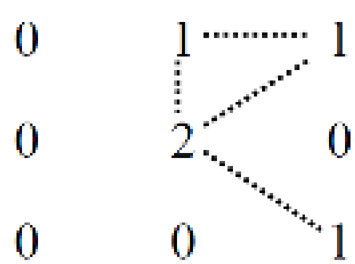
a. 4-Connectivity

Two pixels *p* and *q*, both having values from a set *V* are *4*-connected if *q* is from the set N4(p).



b. 8-Connectivity

Two pixels *p* and *q*, both having values from a set *V* are *4*-connected if *q* is from the set N8(p).



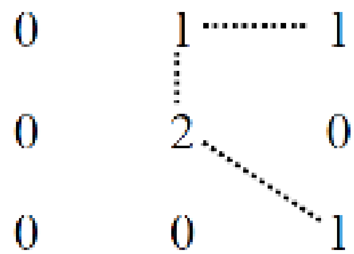
c. m-Connectivity

Mixed connectivity is a modification of 8-adjacency. It is used to remove the ambiguities present in 8-connectivity.

Two pixels p and q with values from {V} are m-connectivityif the following conditions are satisfied.

- q is in $N_4(p)$.

- q is in $N_D(p)$ and the set $[N_4(p) \cap N_4(q)]$ is empty (has no pixels whose values are from V).



1.7.4 Paths and Path length:

A path is also known as digital path or curve. A path from pixel, p with coordinates (x,y) to pixel q with coordinates (s,t) is defined as the sequence of different pixels with coordinates.

$$(X_0, Y_0), (X_1, Y_1), \dots\dots\dots (X_n, Y_n)$$

Where, $(X_0, Y_0) = (x, y)$ and $(X_n, Y_n) = (s, t)$; (X_i, Y_i) and (X_{i-1}, Y_{i-1}) are adjacent for $1 \leq i \leq n$

- **Path Length:**

  Path length is the number of pixels present in a path. It's is given by the value of 'n' here.

- **Closed Path:**

  In a path, if $(X_0, Y_0) = (X_n, Y_n)$ i.e. the first and last pixel are the same, it's known as a closed path

According to the adjacency present, paths can be classified as:

1. 4 – path
2. 8 – path
3. m – path

1.7.5 Region, Boundary and Edges:

- In an image I of pixels, a subset R of pixels in an image I is called a Region of the image if R is a connected set.
- Boundary is also known as border or contour. The boundary of the region R is the set of pixels in the region that have one or more neighbors that are notin R. if R is an entire image, its boundary is defined as the set of pixels in the first and last rows and columns of the image.
- An edge can be defined as a set of contiguous pixel positions where an abrupt change of intensity (gray or color) values occur

1.7.6 Distance Measure:

Distance measures are used to determine the distance between two different pixels in a same image. Various distance measures are used to determine the distance between different pixels.

Conditions: Consider three pixels p, q and z, p has coordinates (x, y), q has coordinates (s, t) and z has coordinates (v, w). For these three pixels **D** is a ***Distance function or metric*** if

- D(p, q) ≥ 0, [D(p, q)=0 if p = q]
- D(p, q) = D(q, p) and
- D(p, z) ≤ D(p, q) + D(q, z)

Types:

- Euclidean Distance
- City – Block (or) D4 Distance
- Chessboard (or) D8 Distance
- Quasi-Euclidean Distance
- Dm Distance

a. **Euclidean Distance:** The Euclidean distance is the straight-line distance between two pixels.

$$\mathbf{D_e(p, q)} = \sqrt{(x-s)^2 + (y-t)^2}$$

0	0	0
0	1	0
0	0	0

Image

1.41	1.0	1.41
1.0	0.0	1.0
1.41	1.0	1.41

Distance Transform

b. **City – Distance:** The city block distance metric measures the path between the pixels based on a 4-connected neighborhood. Pixels whose edges touch are 1 unit apart; pixels diagonally touching are 2 units apart.

$$\mathbf{D_4(p, q)} = |x-s| + |y-t|$$

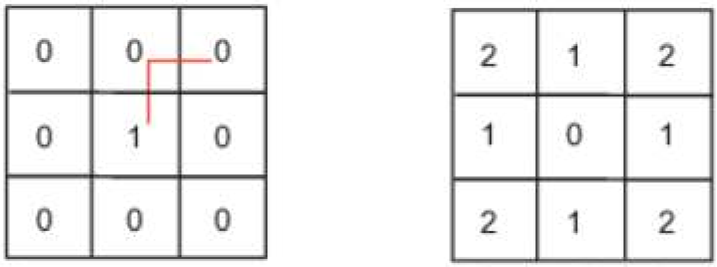


Image Distance Transform

c. **Chessboard Distance:** The chessboard distance metric measures the path between the pixels based on an 8-connected neighborhood. Pixels whose edges or corners touch are 1 unit apart

$$\mathbf{D_8(p, q)} = \boldsymbol{max}(|\boldsymbol{x}-\boldsymbol{s}|, |\boldsymbol{y}-\boldsymbol{t}|)$$

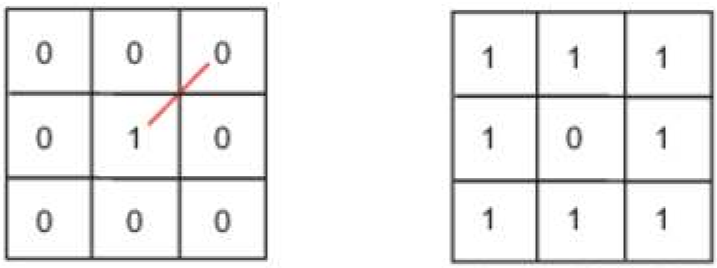


Image Distance Transform

d. **Quasi – Euclidean Distance:** The quasi-Euclidean metric measures the total Euclidean distance along a set of horizontal, vertical, and diagonal line segments.

$$\mathbf{D_{qe}(p, q)} = \sqrt{\mathbf{2}}\textbf{-1}\{(x\text{-}s) + (y\text{-}t)\}$$

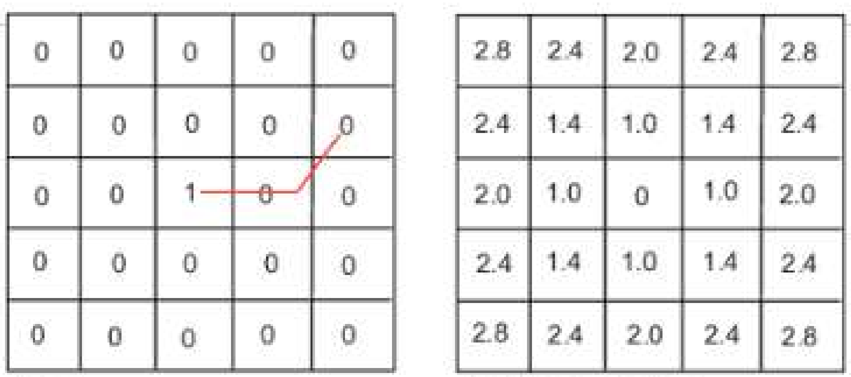


Image Distance Transform

1.8 CONCEPTS OF GRAYLEVELS:

Gray level resolution refers to the predictable or deterministic change in the shades or levels of gray in an image. In short gray level resolution is equal to the number of bits per pixel**.** The number of different colors in an image is depends on the depth of color or bits per pixel.

The mathematical relation that can be established between gray level resolution and bits per pixel can be given as.

$$L = 2^k$$

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

1.8.1 Gray level to binary conversion:

THRESHOLD METHOD

The threshold method uses a threshold value which converts the grayscale image into binary image. The output image replaces all pixels in the input image with luminance greater than the threshold value with the value 1 (white) and replaces all other pixels with the value 0 (black).

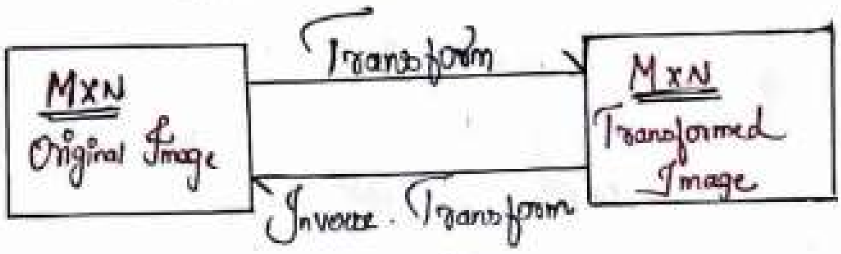
Unit – 2

Image Transform:-

Image transforms can be simple arithmetic operations on images or complex mathematical operations which convert images from one representation to another.

(Or)

A function or operator that takes an image as its input and produces an image as its output. Depending on the transform chosen, the input and output images may appear entirely different and have different interpretations.



Need of Image Transforms:

* Viewing an image in different domains enables the identification of features that may not be as easily detected in spatial domain.

* Image transforms are used majorly in feature extraction, image compression and image enhancement.

1. Two Dimensional Fast Fourier Transform:

A fourier transform is like a mathematical prism that separates a given function into various components based on their frequency components or frequency content. An image is a spatially varying function, One way to analyze spatial variations is to decompose an image into a set of orthogonal functions, one such set being Fourier functions. A fourier transform is used to transform an intensity image into the domain of spatial frequency.

One dimensional DFT for a function $f(x)$ is given by,

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cdot e^{\frac{-j 2\pi u x}{N}} \quad \text{——— (1)}$$

From eq (1) $e^{\frac{-j2\pi ux}{N}}$ is the transformation function where it can be written as,

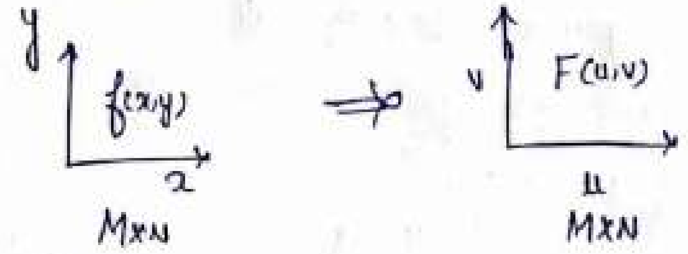
$$e^{\frac{-j2\pi}{N}ux} = \cos\frac{2\pi}{N}ux - j\sin\frac{2\pi}{N}ux$$

* Fourier transform is a powerful tool for performing linear filtering in frequency Domain.

One dimensional IDFT for a function $F(u)$ is given by

$$f(x) = \sum_{u=0}^{N-1} F(u)\, e^{\frac{j2\pi ux}{N}} \quad \text{——— (2)}$$

An image is a 2D-functional of $f(x,y)$, when we take a 2D-DFT for the sequence $f(x,y)$ can be obtained as



2D-DFT of a function $f(x,y)$ is given as.

$$F(u,v) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-j\frac{2\pi}{M}ux} \cdot e^{-j\frac{2\pi}{N}vy} \quad \text{--- (3)}$$

Eq (3) can be Re-written as

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad \text{--- (4)}$$

And its inverse transform is given by,

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j\frac{2\pi}{M}ux} \cdot e^{j\frac{2\pi}{N}vy} \quad \text{--- (5)}$$

Eq (5) can be Re-written as

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad \text{--- (6)}$$

The No of Calculations required to compute DFT by the application of FFT is considerably less and the computational efficiency is very high.

PROPERTIES OF FFT:

i) Seperable Property:

The Seperable property allows a 2D transform to be computed in two steps by successive 1D operations on rows and columns of an image

$$F(u,v) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-j\frac{2\pi}{M}ux} \cdot e^{-j\frac{2\pi}{N}vy}$$

$$\Rightarrow \frac{1}{M} \left\{ \frac{1}{N} \sum_{y=0}^{N-1} f(x,y) \; e^{-j\frac{2\pi}{N}vy} \right\} \sum_{x=0}^{M-1} e^{-j\frac{2\pi}{4}ux}$$

$$\Rightarrow \left\{ \frac{1}{M} \sum_{x=0}^{M-1} f(x,v) \cdot e^{-j\frac{2\pi}{4}ux} \right\}$$

$$\Rightarrow F(u,v)$$

→ Thus performing a 2D fourier transform is equivalent to performing two 1D transform as

* Performing a 1D transform on each row of image $f(x,y)$ to get $F(x,v)$
* Performing a 1D transform on each column of image $F(x,v)$ to get $F(u,v)$

ii) Spatial Shift Property:

The 2D DFT of a shifted version of the image $f(x,y)$, i.e., $f(x-x_0, y)$ is given by

$$f(x-x_0, y) \xrightarrow{DFT} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x-x_0, y)\, e^{\frac{-j2\pi}{M} u(x-x_0)} \cdot e^{\frac{-j2\pi}{N} vy} \quad \text{(7)}$$

Where x_0 represents the number of times that the function $f(x,y)$ is shifted.

The Above equation Can be Re-written as

$$f(x-x_0, y) \xrightarrow{DFT} e^{\frac{-j2\pi}{M} ux_0} \cdot F(u,v)$$

This theorem Proves that the DFT of a shifted function is unaltered except for a linearly varying phase factor.

iii) Periodicity Property:

If a signal $f(x,y)$ is Continuous / Periodic along $M \times N$ samples, then the DFT / FFT of $F(u,v)$ is also periodic with periodicity of $M \times N$ samples

if $\quad f(x+M, y+N) \Leftrightarrow f(x,y)$

then,

$$F(u+M, v+N) \Leftrightarrow F(u,v)$$

iv) Convolution Property:

The FFT of Circular Convolution of two signal sequences $f(x,y)$ and $g(x,y)$ in spatial domain is equivalent to Product / Multiplication of their individual in frequency domain

DB

$$DFT\left\{ f(x,y) * g(x,y) \right\} \Longleftrightarrow F(u,v) \cdot G(u,v)$$

$|||^{ly}$

$$DFT\left\{ f(x,y) \cdot g(x,y) \right\} \Longleftrightarrow F(u,v) * G(u,v).$$

The Convolution theorem tells us that the Convolution of two functions in the spatial domain Corresponds to multiplication in the frequency domain and vice versa.

v) Correlation Property:

It States that the Correlation of two functions $f(x,y)$ and $g(x,y)$ in the spatial domain is equivalent to the multiplication of the Complex Conjugate of the first function with second one in the frequency domain and vice versa

$$DFT\left\{ f(x,y) \circ g(x,y) \right\} \Longleftrightarrow F^{*}(u,v) \cdot G(u,v)$$

$$DFT\left\{ f^{*}(x,y) \cdot g(x,y) \right\} \Longleftrightarrow F(u,v) \circ G(u,v)$$

vi) Scaling Property:

It States that the expansion of an Image in One domain is equal to Compression of an image in another domain

$$DFT\{f(ax, by)\} = \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

The scaling theorem tells us that compression in one domain produces a corresponding expansion in the Fourier domain and vice versa

vii) Conjugate Symmetry:

If the DFT of $f(x,y)$ is $F(u,v)$ then the

$$DFT \text{ of } [f^*(m,n)] = F^*(-u,-v)$$

$$\therefore F(u,v) = F^*(-u,-v)$$

viii) Multiplication by Exponential:

If the $DFT[f(x,y)] = F(u,v)$ then

$$DFT\left[e^{j\frac{2\pi}{N}u_0 x} \cdot e^{j\frac{2\pi}{N}v_0 y} \cdot f(x,y)\right] = F(u-u_0, v-v_0)$$

This theorem proves that multiplication of a function $f(x,y)$ with an exponential in the spatial domain leads to a frequency shift.

ix) Rotation Property:

The rotation property states that if a function is rotated by the angle, its Fourier transform also rotates by an equal amount.

$$f(x,y) = f(r\cos\theta, r\sin\theta)$$

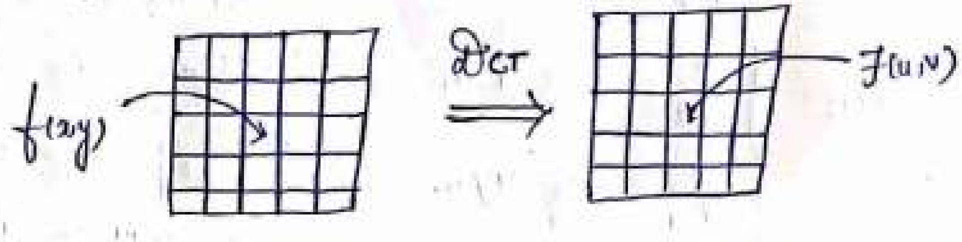
$$DFT[f(r\cos\theta, r\sin\theta)] \rightarrow F[R\cos\phi, R\sin\phi]$$

$$DFT[f(r\cos(\theta+\theta_0), r\sin(\theta+\theta_0))] \rightarrow F[R\cos(\phi+\phi_0), R\sin(\phi+\phi_0)]$$

2D- Discrete Cosine transform:

→ DCT transforms a Signal or image from Spatial domain to frequency domain. Which is similar to DFT

→ The Major difference of DCT is it refers to the even part of the fourier series.



DCT transforms an image brings out a set of Numbers called Co-efficients. These Co-efficients of the transformed image are all real values Whereas the DFT Contains both real and imaginary values it gives us the Computational Complexity.

In an image most of the energy will be Concentrated in the lower frequencies, so if we transform an image into its frequency Components and throw away the higher frequency Component or Co-efficients, We Can reduce the amount of data needed to describe the image Without sacrificing toomuch image quality.

The DCT has been Mainly used in image Compression techniques and it's a lossy Compression.

2D- DCT is given by,

$$F(u,v) = \alpha(u)\,\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)\,u\pi}{2M}\right] \cdot \cos\left[\frac{(2y+1)\,v\pi}{2N}\right] \quad (1)$$

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{M}}, & \text{if } u=0 \\ \sqrt{\frac{2}{M}}, & \text{if } u\neq 0 \end{cases}$$

$$\alpha(v) = \begin{cases} \frac{1}{\sqrt{N}}, & \text{if } v=0 \\ \sqrt{\frac{2}{N}}, & \text{if } v\neq 0 \end{cases}$$

M and N are row and column size of $f(x,y)$. If we apply DCT to real data, the result is also real

2D-IDCT is given by.

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \alpha(u)\cdot\alpha(v)\cdot F(u,v) \cos\left[\frac{(2x+1)\,u\pi}{2M}\right] \cos\left[\frac{(2y+1)\,v\pi}{2N}\right] \quad (2)$$

Properties of DCT:

i) The Cosine transform is real and orthogonal

$$C = C^{*} \Rightarrow C^{T} = C^{-1}$$

ii) The Cosine transform is a fast transform

iii) The Computational complexity of direct and FFT based DCT are the same

iv) The DCT has excellent energy Compaction for highly Correlated images or data

v) The DCT of an N- element Vector can be calculated in $O(N \log_2 N)$ Operations using an N-point FFT as an intermediate step

THE WALSH TRANSFORM

Fourier Analysis is basically the representation of a signal by a set of orthogonal sinusoidal waveforms. The co-eff of this representation is called frequency components and the waveforms are ordered by frequency. The walsh transform is a complete set of orthonormal square-wave function to represent these function. The computational simplicity of the walsh function is due to the fact that walsh functions are real and they take only two values which are either +1 or -1.

For a function $f(x)$, where $N = 2^n$ the 1-Dimensional Discrete walsh transform is given by

$$W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \prod_{i=0}^{n-1} (-1)^{b_i(x)\, b_{n-i-1}(u)}$$

$$\therefore\ g(x,k) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x)\, b_{n-i-1}(u)}$$

where, $g(x,k)$ is the walsh kernel, here 'k' represents the frequency index and also 'n' represents the number of bits to represent a number and $b_i(x)$ represents the i^{th} (from LSB) bit of the binary value, of n decimal number represented in binary. $\therefore \boxed{n = \log_2 N}$

One Dimensional Inverse Discrete walsh transform is given by,

$$f(x) = \sum_{u=0}^{N-1} W(u) \prod_{i=0}^{n-1} (-1)^{b_i(x)\, b_{n-1-i}(u)}$$

The two dimensional Discrete walsh transform is given by

$$W(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \prod_{i=0}^{n-1} (-1)^{[b_i(x)\, b_{n-1-i}(u) + b_i(y)\, b_{n-1-i}(v)]}$$

The two dimensional Inverse DWT is given by

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} W(u,v) \prod_{i=0}^{n-1} (-1)^{[b_i(x)\, b_{n-1-i}(u) + b_i(y)\, b_{n-1-i}(v)]}$$

PROPERTIES:

i) The walsh transform 'W' is real, Symmetric and Orthogonal

$$W = W^{*} = W^{T} = W^{-1}$$

ii) The walsh transform is a fast transform

iii) It reduces the number of addition and Subtractions from N^2 to $N \log_2 N$

iv) The walsh transform Contains only ± 1 values. Therefore no multiplications are required in the transform Calculations

The Hadamard Transform

The Hadamard transform is basically the same as the Walsh transform except the rows of the transform matrix are re-ordered. The elements of the mutually orthogonal basis vectors of a hadamard transform are either +1 or -1, which results in very low computational complexity in the calculation of the transform coefficients. Hadamard matrices are easily constructed for $N = 2^n$ by the following procedure

The order $N \times N$ hadamard matrix can be obtained by the following equation

The 2-D hadamard Matrix / transform is given by

$$H(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) (-1)^{\sum_{i=0}^{n-1} [b_i(x) b_i(u) + b_i(y) b_i(v)]} \quad (1)$$

where,

$$g(x,y,u,v) = \frac{1}{MN} (-1)^{\sum_{i=0}^{n-1} [b_i(x) b_i(u) + b_i(y) b_i(v)]}$$

↳ 2D forward Hadamard Kernel.

Also, the 2D inverse Hadamard transform is expressed as

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} H(u,v) (-1)^{\sum_{i=0}^{n-1} [b_i(x) b_i(u) + b_i(y) b_i(v)]} \quad (2)$$

Where,

$$h(x,y,u,v) = (-1)^{\sum_{i=0}^{n-1} [b_i(x) b_i(u) + b_i(y) b_i(v)]}$$

↳ 2D. Inverse Hadamard Kernel.

Hadamard Matrix:

The forward and inverse Hadamard transforms can also be implemented with the help of hadamard transformation Matrices.

The Hadamard matrix of order 2N is given by the expression

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

Where, $N = 2^n$

$H_N \Rightarrow$ Matrix of order N

From this, the lower order Hadamard Matrix i.e, N=2

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Property:

i) The Hadamard transform 'H' is real, symmetric & orthogonal

$$H = H^* = H^T = H^{-1}$$

ii) The Hadamard transform is a fast transform

iii) It reduces the Number of Add and Sub from N^2 to $N\log_2 N$

iv) The Hadamard transform contains only ±1 values. Therefore no multiplications are required in the transform calculations

v) The Natural order of the H.T Co-eff are equal to the bit reversed gray code representation of its sequency

vi) The Hadamard transform has good to very good energy compaction for highly correlated images

The Haar Transform.

The Haar functions $h_k(x)$ are defined on a continuous interval $x \in [0, 1)$. The Haar transform is based on a class of orthogonal matrices whose elements are either 1, -1 or 0 multiplied by powers of $\sqrt{2}$. The Haar transform is a computationally efficient transform as the transform of an N-point vector requires only $2(N-1)$ additions and N Multiplications

Algorithm to generate Haar Basis

i) Determine the Order of N of the Haar basis

ii) Determine n where $n = \log_2 N$

iii) Determine p and q

a) $0 \leq p < n-1$

b) if $p=0$, then $q=0$ (or) $q=1$

c) if $p \neq 0$, then $1 \leq q \leq 2^p$

iv) Determine 'k'

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

For $N=4$

$\therefore \eta = \log_2 4 = 2$

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

v) Determine x.

$$x \rightarrow [0, 1) \Rightarrow \left\{ \frac{0}{N}, \frac{1}{N} \cdots\cdots \frac{N-1}{N} \right\}$$

vi)

If $k=0$ then $h_k(x) = h_0(x) = \frac{1}{\sqrt{N}}$

Otherwise,

$$h_k(x) = h_{p,q}(x) = \frac{1}{\sqrt{N}} \begin{cases} +2^{p/2} & \frac{(q-1)}{2^p} \leq x < \frac{(q-1/2)}{2^p} \\ -2^{p/2} & \frac{(q-1/2)}{2^p} \leq x < q/2^p \\ 0 & \text{Otherwise} \end{cases}$$

For $N=2$, the Haar transform is given by,

$$h_k(x) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

As Haar transform takes differences of local averages of the samples of input vector, it can be used for edge extraction.

Properties of Haar transform:

i) The Haar transform is real and orthogonal

$$H_k = H_k^*$$

$$H_k^{-1} = H_k^T$$

ii) It is a very fast transform

iii) The basis vectors of the Haar transform are sequency ordered

iv) The Haar transform has poor energy compaction of images

The Slant Transform.

The slant transform is based on a class of Orthogonal transform containing sawtooth waveforms or slant basis vectors. A slant basis vector that is monotonically decreasing in constant steps from maximum to minimum has the sequency property and has a fast computational algorithm.

Let S_n denote an $N \times N$ slant matrix $N = 2^n$

$$S_n = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|c|cc|c} 1 & 0 & \multirow{2}{*}{0} & 1 & 0 & \multirow{2}{*}{0} \\ a_n & b_n & & -a_n & b_n & \\ \hline \multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & I_{(N/2)-2} \\ \hline 0 & 1 & \multirow{2}{*}{0} & 0 & -1 & \multirow{2}{*}{0} \\ -b_n & a_n & & b_n & a_n & \\ \hline \multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & -I_{(N/2)-2} \end{array}\right] \left[\begin{array}{c|c} S_{n-1} & 0 \\ \hline 0 & S_{n-1} \end{array}\right]$$

Where, $N = 2^n$, I_M denotes an $M \times M$ identity Matrix,

for $N = 2$,

$$S_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The parameters a_N and b_N values are given by the following recursive formulae;

$$b_n = \left(1 + 4a_{n-1}^2\right)^{-1/2}, \quad a = 1$$

$$a_n = 2 b_n a_{n-1}$$

these solve to give

$$a_{n+1} = \left(\frac{3N^2}{4N^2-1}\right)^{1/2}, \quad N = 2^n$$

$$b_{n+1} = \left(\frac{N^2-1}{4N^2-1}\right)^{1/2}, \quad N = 2^n$$

using these formulas, the 4×4 slant transformation matrix is obtained as

$$S_2 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3/\sqrt{5} & 1/\sqrt{5} & -1/\sqrt{5} & -3/\sqrt{5} \\ 1 & -1 & -1 & 1 \\ 1/\sqrt{5} & -3/\sqrt{5} & 3/\sqrt{5} & -1/\sqrt{5} \end{bmatrix}$$

Properties of Slant transform:

i) the Slant transform is real and orthogonal

$$S = S^{*} \text{ and } S^{T} = S^{-1}$$

ii) It is a fast transform

iii) the slant transform requires $O(N \log_2 N)$ operations for an $N \times 1$ vector

iv) It has very good to excellent energy compaction for images

v) The basis vectors of the slant transform matrix S_N are not sequency ordered.

1. IMAGE ENHANCEMENT TECHNIQUES:

Image enhancement techniques are designed to improve the quality of an image as perceived by a human being. Image enhancement can be performed both in spatial and as well as in the frequency domain.

1.1 Image Enhancement in Spatial Domain:

The Spatial domain technique deals with the manipulation of pixel values. The spatial domain technique is used to improve the interpretability of the information present in images for human viewers.

1.1.2 Point Operation:

In point operation, each pixel is modified by an equation that is not dependent on other pixel value. The point operation can be represented by,

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

In point operation, 'T' operates on one pixel or there exists a one to one mapping between the input image f(x,y) and the output image g(x,y).

Types of Point Operation:

A. Brightness Modification:

The brightness refers to the overall lightness or darkness of the image. The brightness of the image depends on the value associated with the pixel of the image. When changing the brightness of an image, a constant is added or subtracted from the luminance of all sample values.

a) **Increasing the brightness of the image:** Adding a constant value 'k' to each and every pixel of the image. This is represented by,

$$g(x,y) = f(x,y) + k$$

b) **Decreasing the brightness of the image:** Subtracting a constant value 'k' to each and every pixel of the image. This is represented by,

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

B. Contrast Adjustment:

The contrast is the difference in brightness in objects or regions. Contrast adjustment is done by scaling all the pixels of the image by a constant value 'k'. Changing the brightness of the image, changes the range of luminance values present in the image.

a) **Increasing the contrast of the image:** Multiplying a constant value 'k' to each and every pixel of the image. This is represented by,

$$g(x,y) = f(x,y) * k$$

b) **Decreasing the contrast of the image:** Dividing a constant value 'k' to each and every pixel of the image. This is represented by,

$$g(x,y) = f(x,y) \div k$$

1.1.3 Histogram Processing:

Histogram

The histogram of an image is a plot or graph drawn between gray level values (0 -255) in the X-axis and the number of pixels having the corresponding gray levels in the Y- axis.

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

$$h(r_k) = n_k$$

where, r_k – k^{th} gray level

n_k – Number of pixels in the image having gray level r_k

Normalized Histogram

A normalized histogram is obtained by dividing each value of the histogram by the total number of pixels in the image. It's given by

$$p(r_k) = \frac{n_k}{n} \quad \text{for } k = 0, 1\ldots L-1$$

Where, n – Total number of pixels in the image

$p(r_k)$ gives the probability of occurrence of gray level r_k. Also, the sum of all components of the normalized histogram is equal to 1.

Features of Histogram:

- For a dark image, the components of a histogram will be concentrated on the low (dark) side of the gray scale.
- For a bright image, the components of a histogram will be biased toward the high (bright) side of the gray scale.
- A low contrast image will have a histogram that is narrow and centered toward the middle of the gray scale.
- A high contrast image histogram covers a broad range of gray scale and the distribution of pixels will be almost uniform.

Therefore, all the histogram processing techniques try to increase the dynamic range of an image to achieve high contrast.

1.1.3.1 Histogram Equalization (or) Histogram Linearization:

Histogram Equalization is a process of automatically determining a transformation function which produces an output image with a uniform histogram. Histogram equalization reassigns the brightness values of pixels based on the image histogram. Histogram equalization provides more visually pleasing results across a wider range of images.

Let, the pixel values are in the interval [0,L-1]. The discrete form of the transformation function is given by

$$S_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$$

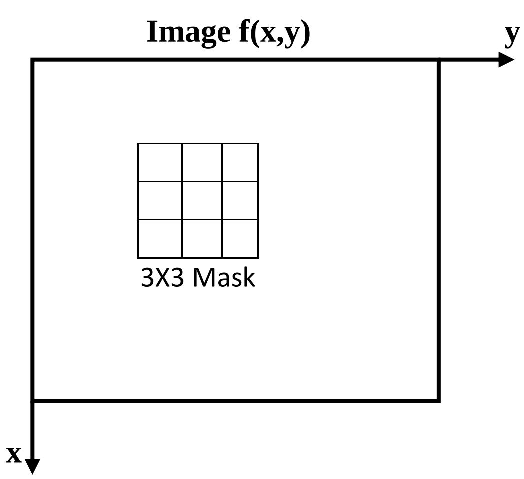
Thus, mapping each pixel with level r_k in the input image into corresponding pixel with level S_k in the output image using the above equation will produce an enhanced image. A plot of $p_r(r_k)$ versus r_k is known as histogram and the transformation or mapping using above equation is called histogram equalization.

2. SPATIAL FILTERING:

Filtering operation that is performed directly on the pixels of an image is known as spatial filtering. A filter is simply a subimage which is also known as a mask, kernel, template or window. The values in a filter subimage are called coefficients.

Filtering Technique:

- In the filtering process, the filter mask is moved from point to point in an image starting at one end.
- Then, using a predefined relationship, the response of the filter at each point is calculated.



Mask Coefficients

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

2.1 Smoothing Spatial Filters:

Smoothing is the process of ***reducing sharp transitions*** in the gray levels of an image.

The important uses of smoothing are,

i. **Blurring:** Removal of small details from an image and also bridging of small gaps in lines or curves.
ii. **Noise Reduction:**

2.1.1 Smoothing by Linear Filters:

In linear filtering, each and every pixel in an image is replaced by the average value of the gray levels in the neighborhood of the filter mask. Thus, an image with reduced sharp transitions is obtained.

a) **Box Filter / Mean Filter / Averaging Filter:**

A spatial averaging filter in which all ***coefficients are equal*** is called box or mean filter. The box filter replaces each pixel by the average of all the values in the local neighborhood. At the end of the filtering process the entire image is divided by 9

$\frac{1}{9}$ X

1	1	1
1	1	1
1	1	1

3 by 3 Box Filter

Sum of all the coefficients = 1+1+1+1+1+1+1+1+1 = 9

b) **Weighted Average Filter:**

The Name weighted average filter is given because, From the mask it's obvious that the pixels nearest to the Centre are weighted more than the distant pixels. The pixel to be updated is replaced by a sum of the nearby pixel value times the weights given in the matrix and divided by the sum of the coefficients in the matrix.

$\frac{1}{16}$ X

1	2	1
2	4	2
1	2	1

3 by 3 Weighted Average Filter

Sum of all the coefficients = 1+2+1+2+4+2+1+2+1 = 16

2.1.2 Smoothing by Non-Linear / 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 centre pixels wih the value determined by the ranking result.

a) Median Filter:

A Median filter smoothens the image by utilizing the median of the neighborhood. The concept of a median filter was first introduced by Tukey in 1977. Median filters perform the following tasks to find each pixel value in the processed image:

1. All pixels in the neighborhood of the pixel in the original image which are identified by the mask are sorted in the ascending or descending order.
2. The median of sorted value is computed and is chosen as the pixel value for the processed image.

2.2 Sharpening Spatial Filters:

The main aim of image sharpening is to highlight fine details in an image and also enhance detail which is blurred by error or by some natural effects.

a) Laplacian Filter:

The Simplest isotropic operator is the Laplacian, which is also a linear operator. For an image with two variables f(x,y), the Laplacian is defined as

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

Where,

$\frac{\partial^2 f}{\partial x^2}$ = f(x+1,y) + f(x-1,y) – 2f(x,y) (Partial 2nd order derivative in x-direction)

$\frac{\partial^2 f}{\partial x^2}$ = f(x,y+1) + f(x,y-1) – 2f(x,y) (Partial 2nd order derivative in y-direction)

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

$$= [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)]$$

From the above equation, the Laplacian mask is obtained as

0	1	0
1	-4	1
0	1	0

Sum of all the coefficients = 0

3 by 3 Laplacian Mask $\nabla^2 f(x,y)$

Thus, an enhanced image g(x,y) is obtained by

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

Uses of the Laplacian:

- The laplacian is a derivative operator and is used to highlight the gray level discontinuities and deemphasize or suppress the regions with slowly varying gray levels
- This produces an image that has dark, featureless background with grayish edge lines and other discontinuities superimposed on the background. This is used to identify the discontinuities.
- By adding the original and laplacian images, the background information can be recovered without affecting the sharpening effect.

b) Unsharp Masking:

Unsharp masking is one of the techniques typically used for edge enhancement. In this approach, a smoothened version of the image is subtracted from the original image; hence tipping the image balance towards the sharper content of the image.

$$f_s(x,y) = f(x,y) - \bar{f}(x,y)$$

where, $f_s(x,y)$ – Sharpened image

$f(x,y)$ – Original image

$\bar{f}(x,y)$ – Smoothened version of $f(x,y)$

c) High-Boost Filtering:

A high boost filter is also known as a high frequency emphasis filter. A high-boost filter is used to retain some of the low-frequency components to aid in the interpretation of an image. A high-boost filtering, the input image f(x,y) is multiplied by an amplification factor 'A' before subtracting the low pass image.

$$f_{hb} = (A-1)\, f(x,y) + f_s(x,y)$$

d) Gradient Filter:

The gradient is defined as a two-dimensional column vector. For a function 'f' at coordinates (x,y), the gradient is given by

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Image Regions

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

The Magnitude of this gradient vector is,

$$\nabla f = \mathrm{mag}(\nabla f) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \qquad (1)$$

i. Roberts Cross-Gradient Operators:

This is proposed by Roberts. It uses cross differences. The gradient at point Z_5 can be expressed as,

$G_x = Z_9 - Z_5$ and

$G_y = Z_8 - Z_6$

Using the values of G_x and G_y in equation (1)

$$\nabla f = \sqrt{[(Z_9 - Z_5)^2 + [(Z_8 - Z_6)^2]]} \qquad (2)$$

$$= |Z_9 - Z_5| + |Z_8 - Z_6| \qquad (3)$$

-1	0
0	1

(a)

0	-1
1	0

(b)

These two masks are referred to as 'Roberts cross-gradient operators'.

ii. Sobel Operators:

$G_x = (Z7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$ and

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

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

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

$$\nabla f = |(Z_7 + 2Z_8 + Z_9) - (Z_1 + Z_2 + Z_3)| + |(Z_3 + 2Z_6 + Z_9) - (Z_1 + Z_4 + Z_7)|$$

The above two masks are used to implement this equation and are referred to as 'Sobel' operators.

DIGITAL IMAGE PROCESSING

UNIT – III

3.1 Image Enhancement in Frequency Domain:

Frequency refers to the rate of repetition of some periodic events. In image processing, spatial frequency refers to the variation of image brightness with its position in space. Frequency domain is just the space defined by the values of the Fourier transform and its frequency variable. Image enhancement in frequency domain refers that the multiplication of each element of the Fourier coefficient of an image F(u,v) by a suitably chosen weighing function H(u,v) then we can accentuate certain frequency components and attenuate others.

$$G(u,v) = F(u,v) \cdot H(u,v) \qquad \ldots\ldots\ldots\ldots\ldots\ldots..(1)$$

The enhancement or suppression of frequency components is termed as ***Fourier filtering (or) Frequency domain filtering.***

The above figure shows the basic steps of filtering in frequency domain.

Where, f(x,y) = Input image

g(x,y) = Enhanced image

Frequency Domain Filtering:

Frequency domain filtering is simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter and take the inverse transform to produce the enhanced image. The filters are classified based into three categories.

- **Notch filter** – *A filter that attenuate a selected frequency (and some of its neighbors) and leave other frequencies of the Fourier transform relatively unchanged*
- **Low pass filter** – *A filter that allows only lower frequency components that attenuates the higher frequency components. (Regions of relatively uniform gray values in an image contribute to low-frequency content)*
- **High pass filter** – *A filter that allows only higher frequency components that attenuates the lower frequency components. (Regions of edges and sharp*

transitions in gray values in an image contribute significantly to high frequency content)

3.1.1 Low pass filer / Smoothing filter in frequency domain:

Smoothing in frequency domain is the process of attenuating a specified range of high-frequency components in the transform of a given image. It's also known as **blurring.**

Need for Smoothing: Sharp transitions in the gray levels of an image such as edges and noise are present as high-frequency components in its Fourier transform. To remove these unwanted contents, Smoothening is required.

3.1.1.1 Ideal low pass filter:

Ideal lowpass filter is the simplest lowpass filter. It "cuts off" all the high-frequency component of the Fourier transform which are located at a distance greater than a specified distance D_0 from the origin of the centered transform.

The transfer function of 2D-ILPF is given by,

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

Where, D_0 – Cut-off frequency

D(u, v) – Distance between the point (u, v) and the origin of the frequency rectangle.

Ringing Effect:

Ringing effect so known as Gibbs phenomenon in mathematical methods of image processing is the annoying effect in images and video appeared as rippling artifact near sharp edges. ***Ringing effect is caused by distortion or loss of high frequency information in image.***

3.1.1.2 Butterworth low pass filter:

In Butterworth lowpass filter, there is no clear cutoff frequency which decides the amount of frequencies to be passed and the amount of frequencies to be filtered. When amount of high-frequency content removed decreases, the image becomes finer in texture.

The transfer function of 2D-BLPF is given by,

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

Where, n – Filter order

Ringing Effect: In BLPF ringing effect is directly proportional to the order of the filter (n). If order “n” increases the ringing also increases. Therefore, BLPF of order “n = 2” is preferred because, it provides ***Effective lowpass filtering and Acceptable ringing characteristics.***

3.1.1.3 Gaussian low pass filter:

The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump.

The transfer function of 2D-GLPF is given by,

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

Where, σ – Measure of the gaussian curve spread.

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

The obtained filter will have ***no ringing effects.***

Applications of LPF:

- Character Recognition
- Printing and Publishing Industry and
- Processing satellite and Aerial Images

3.1.2 High pass filer / Sharpening filter in frequency domain:

Image sharpening refers to any enhancement technique that highlights edges and fine details in an image (or) Image Sharpening is the process of attenuating low frequency components without disturbing the information in the Fourier transform. Image sharpening is widely used in printing and photographic industries for increasing the local contrast and sharpening the images

Need for Sharpening: Human perception is highly sensitive to edges and fine details of an image, and since they are composed primarily by high frequency components, the visual quality of an image can be enormously degraded if the high frequencies are attenuated or completely removed. In contrast, enhancing the high-frequency components of an image leads to an improvement in the visual quality.

3.1.2.1 Ideal high pass filter:

Ideal high pass filter is the simplest high pass filter. It “cuts off” all the low-frequency component of the Fourier transform which are located at a distance lesser than a specified distance D_0 from the origin of the centered transform.

The transfer function of 2D-IHPF is given by,

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

Where, D_0 – Cut-off frequency

D(u, v) – Distance between the point (u, v) and the origin of the frequency rectangle.

3.1.2.2 <u>Butterworth high pass filter:</u>

The behavior of Butterworth high pass filters is smoother than ideal highpass filter. This means that the images produced by BHPF are better than IHPF produced images.

The transfer function of 2D-BLPF is given by,

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

Where, n – Filter order

3.1.2.3 <u>Gaussian high pass filter:</u>

The results produced by a gaussian high pass filter are smoother than the results produced by IHPF and BHPF.

The transfer function of GHPF with cutoff frequency at distance D_0 from the origin is expressed as,

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

3.2 COLOR IMAGE PROCESSING:

Color image processing is divided into two broad categories namely,

1. **Pseudo Color Processing:** In Pseudo color processing, a color is assigned to a particular monochrome intensity or range of intensities.
2. **Full Color Processing:** In Full color processing, the images are acquired using full-color sensors such as a TV camera or a Color Scanner are processed.

The various color models that exist are:

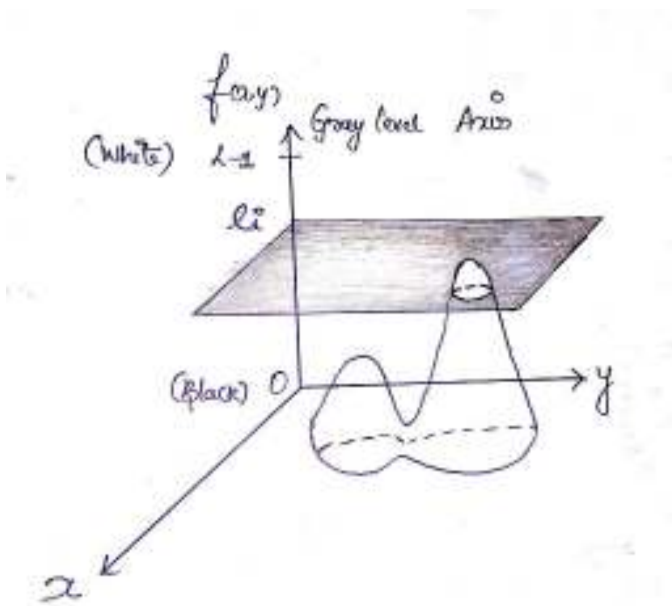
- Red Blue Green (RGB) Model (Primary color Model)
- Cyan Magenta Yellow (CMY) Model (Secondary color Model)
- Hue Saturation Intensity (HSI) Model

3.2.1 Pseudo Color Image Processing:

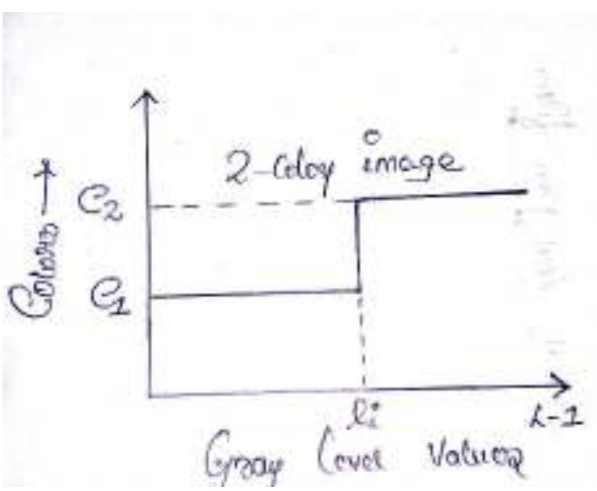
- Pseudo or False image processing is the term used to differentiate the process of assigning colors to monochrome images from the processes associate with true color images.
- Changing a black & white image or grayscale image into color image helps human to distinguish / read the images properly.

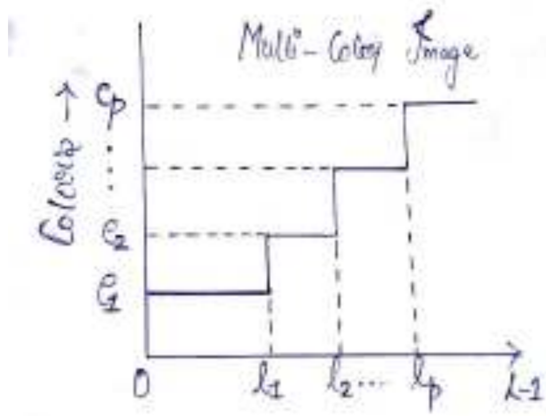
3.2.1.1 Intensity Slicing:

- It's one of the simplest methods of pseudo-color processing, here the image is represented in a 3-D manner or function.
- The X and Y axes have the pixels of the image f(x,y) and the gray level axis consist of the gray level values.
- The 3-D graph shows the various Gray level values that the image pixels have.
- ***If the graph is sliced by using a particular Gray level value (l_i), then we can assign two different colors to the pixels having Gray level values smaller than l_i and Gray level values greater than l_i.***



- This this technique converts a monochrome image into a bicolor image.



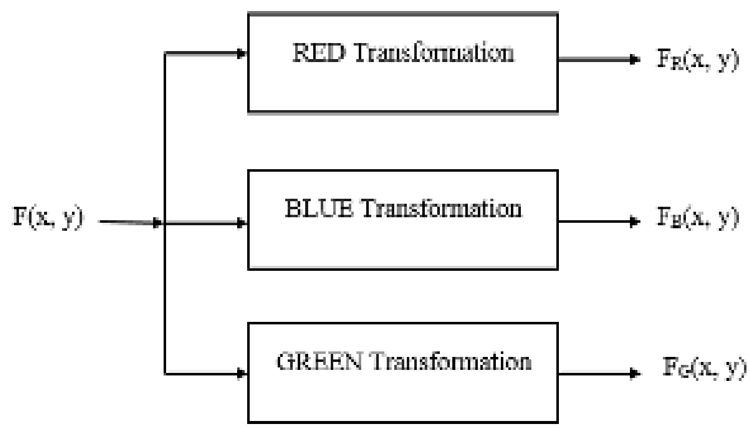


- If more than on Gray level value is used to slice an image graph. Then the slicing plane are given by $l_1, l_2, \ldots\ldots l_p$.

- Then p slicing planes will slice the graph into p+1 intervals. For p+1 intervals the pixels can be assigned C_p+1 colors, changing the input image into a multi-color image.

3.2.1.2 Gray Level to Color Transformation:

- In this method of pseudo color image processing, the input image f(x,y) is split into 3 separate channels.



- These 3 channels carry the input image to Red, Green, Blue Transformation devices.
- Here the transformation is done to each and every pixel independent of their gray level values.
- The final image that is produced is a composite made up of the 3 primary colors i.e. Red, Blue & Green

3.2.2 Full Color Image Processing:

There are two categories of Full-color image processing,

1. First approach is applied to the whole image color components individually and produce a composite color image.
2. The second approach the processing is applied to the color pixels directly.

A pixel of a colored image is a vector quantity and they process 3 color components in it,

$$C(x,y) = \begin{bmatrix} C_R(x,y) \\ C_G(x,y) \\ C_B(x,y) \end{bmatrix}$$

3.2.2.1 Color Transformation:

- Generalized transformation equation is given by,

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

- Color transformation is used to transform pixel to a single-color model. It is not implemented to converts a HIS image to RGB image.

$$S_i = T_i(r_1, r_2 \dots \dots \dots ., r_n)$$

Where, S =Output pixel, T= Transformation, r = Input pixel and n = depends on color model (i.e: For RGB n=3)

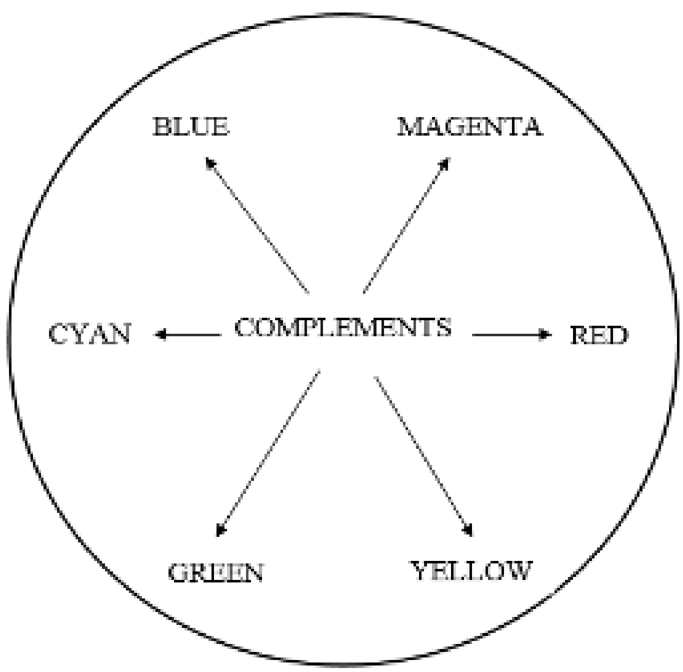
- Cost of conversion varies according to the color model in which the conversion is being done.
- For example, to modify the intensity of an image using;

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

- For HIS model, the formula is: $S_3 = kr_3; S_2 = r_2; S_1 = r_1$
- For RGB model, the formula is: $S_i = kr_i$; Where i = 1,2,3.

3.2.2.2 Color Components:

- The colors lying directly opposite to one-another in a color circle are called components.
- Color complements are analogous to gray scale negatives.



3.2.2.3 Color Slicing:

- Color slicing is used to highlight a specific range of colors. This process can be used to display the colors of interest so that they stand out from the background.
- The basic principle behind slicing a color image is to map the colors outside the region of interest to an average neutral color.
- If the colors of interest are enclosed by a cube; the pixel transformation formula is given by,

$$S_i = \begin{cases} 0.5\,, if\ \left[\left|r_j - a_j\right| > \dfrac{W}{2}\right] for\ any\ 1 \leq j \leq n \\ r_i\ , \qquad\qquad\qquad\qquad otherwise \end{cases}$$

Where, W- Width of the cube

$a_1, a_2, \ldots\ldots a_n$ are the color components

S_i – o/p pixel and r_i – i/p pixel.

- If the colors of interest are enclosed by a sphere; the pixel transformation formula is given by

$$S_i = \begin{cases} 0.5\,, if\ \sum_{j=1}^{n}\left(r_j - a_j\right)^2 > R_0^2 \\ r_i\,, \qquad\qquad Otherwise \end{cases}$$

Where, R_0 – Radius of the sphere.

3.3 IMAGE RESTORATION:

Image restoration is defined as the process of reconstructing or recovering an image which is in the degraded or distorted state. But a knowledge of the degradation function is needed for a successful restoration.

Aim:

The aim of restoration is to improve the appearance of an image. This is same as the enhancement process. But, unlike image enhancement, restoration is an ***objective*** process, which means that the process is based on mathematical or probabilistic models of image degradation.

3.3.1 Image Restoration / Degradation model:

The figure 3.1 shows that the degradation process is modeled as a degradation function that together with an additive noise term operates on an input image f(x,y) to produce degraded image g(x,y). Given g(x,y), some knowledge about the degradation function H, and some knowledge about the additive noise term η(x,y), the objective of restoration is to obtain an estimate f^(x,y) of the original image.

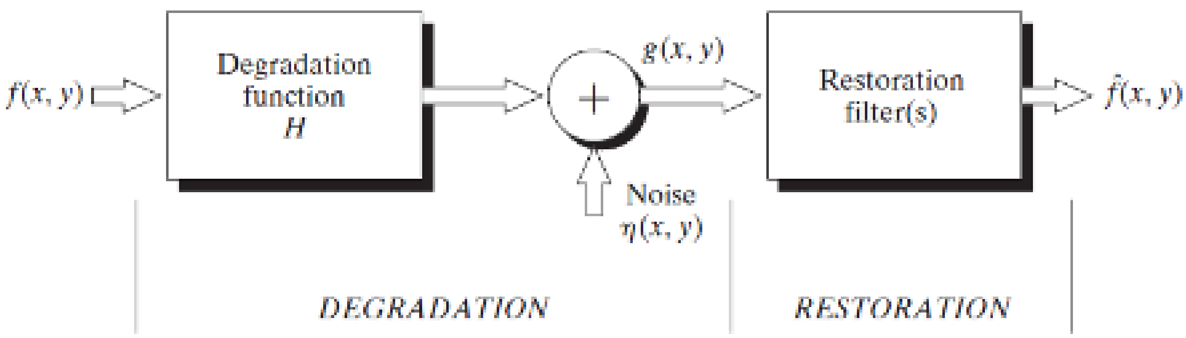


Fig: 3.1 A model of the image degradation / restoration process.

Degradation Process:

The Degradation unction here operates with an additive noise term. It works on the input image f(x,y) and produces a degraded image g(x,y)

Restoration Process:

The objective of restoration process is to obtain an estimate or approximation f^(x,y) of the original image. This approximation should be as close as possible to the original input image. When knowledge about H and n increases, the approximation f^(x,y) gets more closer to f(x,y).

Spatial Domain Representation:

Assuming that H is linear, position invariant process, the degraded image can be represented in the spatial domain as

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

Where, g(x,y) – Degraded image in spatial domain

h(x,y) – Spatial representation of the degradation function

f(x,y) – Original image

n(x,y) – Additive noise

Frequency Domain Representation:

Convolution in spatial domain is equal to the multiplication in the frequency domain, therefore, the degraded image in the frequency domain is represented by,

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

Where, H(u,v) – Fourier transform of h(x,y)

F(u,v) – Fourier transform of f(x,y)

N(u,v) – Fourier transform of n(x,y)

3.3.2 Types of Image Blur:

Blur can be introduced by an improperly focused lens, relative motion between camera and the scene or atmospheric turbulence. Blurring is a form of bandwidth reduction of an ideal image owing to the imperfect image formation process.

1) **Gauss Blur:**

Gauss blur is defined by the following point-spread function:

$$h(x,y) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Here, σ is called the variance of the blur occurs due to long time atmosphere exposure.

2) **Out-of-focus Blur:**

This blurring is produced by a defocused optical system. It distributes a single point uniformly over a disk surrounding the point. The point spread function of the out-of-focus blur is given by

$$h(x,y) = c\begin{cases} 1, \sqrt{(x-c_x)^2 + \left(y-c_y\right)^2} \ \le \ r \\ 0, \qquad\qquad\qquad Otherwise \end{cases}$$

Where, r is the radius and (c_x, c_y) is the centre of the out-of-focus point spread function.

3) Motion Blur

Motion blur is due to relative motion between the recording device and the scene. When an object or the camera is moved during light exposure, a motion blurred image is produced

$$h(x,y,L,\phi) = \begin{cases} \frac{1}{L} \ if \ \sqrt{x^2+y^2} \ \le \frac{L}{2} and \frac{x}{y} = \ -tan\phi \\ 0 \qquad\qquad\qquad Otherwise \end{cases}$$

Where, L – Length of motion

Φ – Angle of radiance

4) Atmospheric Turbulence Blur:

Atmospheric turbulence blur is introduced by variety of factors like temperature, wind speed, exposure time, for long exposures, the point spread function is given by,

$$h(x,y,\sigma_G) = C\ exp\left(-\frac{x^2+y^2}{2\sigma_G^2}\right)$$

Where, σ_G – Amount of spread of the blur

3.3.3 Algebraic Approach:

3.3.4 Inverse Filtering:

- Inverse filtering is defined as the process of ***recovering the input*** of a system from its output.
- Direct inverse filtering is the ***simplest technique*** of restoration.

Concept:

Let an image f(x,y) is degraded by a degradation function H. the obtained degraded image is denoted as g(x,y).

$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \qquad (1)$$

The inverse filtering divides the transform of the degraded image G(u,v) by the degradation function H(u,v) and determines an approximation of the transform of the original image.

It is expressed as,

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \tag{2}$$

Where, F$(u,v) - Transform\ of\ the\ original\ image$

$\hat{F}(u,v) - Approximation\ of\ F(u,v)$

Drawbacks:

The inverse filtering divides the transform of the degraded image G(u,v) by the degradation function H(u,v).

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \tag{3}$$

Substituting this in equation (2) gives,

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

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \tag{4}$$

An exact inverse filtering expression can be obtained by substituting eq (2) in eq (4) as

$$F(u,v) = \frac{G(u,v)}{H(u,v)} - \frac{N(u,v)}{H(u,v)} \tag{5}$$

In addition to the transform of the original image, there is an extra term N(u,v)/H(u,v) in the right hand side of equation (4). This shows that the approximated function is not equal to the original function.

- The original or undegraded image can't be fully recovered by this method. Due to that the fourier transform of N(u,v) is unknown since it's a random function.
- Zero or small-value problem (If the degradation function value is ***small or zero***, the second term in eq(4) will have higher value than the first term,

F(u,v). This indicates the ***poor performance*** of the system which results in bad approximation of the original function. This is known as the ***'Zero or small-value problem'***

- If noise is present in the region of vertical high value strips, the inverse filtering corrupts the restored image.
- This method has no explicit provision to handle the noise.

Limitation:

- Inverse filtering is highly sensitive to noise.

Application:

- Removal of blur caused by uniform linear motion.

3.3.5 Wiener (or) Least Mean Square (LMS) Filtering:

- For restoration of an image, this method considers the degradation function as well as the statistical properties of noise.
- Therefore, it is used to improve direct inverse filtering because, it has a provision to handle the noise.

Objective:

The main objective of LMS filtering is to approximate the original image in such a way that the mean square error between the original and approximated images is minimized.

Mean Square Error:

The mean square error is found by taking the expected value of the difference between two images. It is expressed as,

$$\boldsymbol{e^2 = E\{(f-\hat{f})^2\}} \qquad (6)$$

Where, E{x} -Expected value of x

f – Uncorrupted (or) original image.

$\boldsymbol{\hat{f}}$- Approximation of 'f'

Assumptions Made:

The following assumptions are made to perform LMS error filtering.

- The image and noise are uncorrelated, (i.e) they have no relation.
- Either image or noise has zero mean.
- The approximated gray levels are a linear function of the degraded gray levels.

Approximated Image:

Based on the assumptions made, the approximated image in frequency domain which satisfies the minimum error function is given by

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

Simplifying the above equation gives,

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

Multiplying and Dividing the above equation by H(u,v) gives,

$$\hat{F}(u,v) = \left[\left(\frac{1}{H(u,v)}\right)\frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}}\right] G(u,v) \qquad (7)$$

[$H(u,v) . H^*(u,v) = |H(u,v)|^2$]

Where, H(u,v) – Transform of the degradation function.

$H^*(u,v)$ – Complex conjugate of H(u,v)

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

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

G(u,v) – Transform of degraded image

Equation (7) is called ***wiener filter*** and the terms inside the ***square brackets*** are referred to as the ***minimum mean square error filter*** or ***least mean square filter***.

If N(u,v) = 0;

When the noise N(u,v) = 0, then eq(7) becomes,

$$\hat{F}(u,v) = \left[\left(\frac{1}{H(u,v)}\right)\frac{|H(u,v)|^2}{|H(u,v)|^2 + 0}\right]G(u,v)$$

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \qquad (8)$$

The Eq(8) is similar to Eq(2) for inverse filtering. Thus, if noise is zero, Winer filter reduces to inverse filter.

With unknown quantities:

If power spectrum of the undegraded image $S_f(u,v)$ is unknown, the Eq(7) can be written as,

$$\hat{F}(u,v) = \left[\left(\frac{1}{H(u,v)}\right)\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right]G(u,v) \qquad (9)$$

Where, K- Specified constant

Here, the noise is assumed to be ***White noise,*** Whose spectrum $|N(u,v)|^2$ is a constant.

Advantages:

- Wiener filter has no small or zero-value problem
- The results obtained are more closer to the original image than the inverse filtering

Disadvantages:

- It requires the power spectrum of the undegraded image and noise to be known, which makes the implementation more difficult.
- Wiener filter is based on minimizing a statistical criterion. Therefore, the results are optimal only in an average sense.

3.3.6 Constrained Least Squares Filtering:

- Constrained least squares filtering is a restoration technique which uses only the mean and variance of the noise.

Matrix Formulation:

The basic operation of the restoration process is given by an equation

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

The above equation can be written in ***vector-matrix*** form as

$$g = Hf + n \qquad (10)$$

If g(x,y) has the size MxN

- g, f and n matrices will have dimensions MNx1.
- H matrix will have MNxMN dimensions.

Problems Encountered:

There are two main problems in the matrix calculation of constrained least squares filtering. They are:

(1) Even for medium size image, the matrix dimensions will be very large, which makes the computation difficulty.
(2) The matrix ‘H’ in Eq(10) is highly sensitive to noise.

Handling Noise Sensitivity:

The noise sensitivity problem can be reduced by performing the restoration based on some measures of smoothness, such as the Laplacian.

In such a case, the minimum value of a criterion function, C should be found. The function is defined as

$$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\nabla^2 f(x,y)\right]^2 \qquad (11)$$

The ***constraint*** here is:

$$\left|\left|g - H\hat{f}\right|\right|^2 = \left|\left|n\right|\right|^2 \qquad (12)$$

Solution:

The solution to this problem in frequency domain is given by,

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |P(u,v)|^2}\right] G(u,v) \qquad (13)$$

Where, γ – Parameter adjusted to satisfy the equation (12)

P(u,v) – Fourier Transform of the Laplacian operator given by,

$$p(x,y)=\begin{bmatrix}0 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 0\end{bmatrix}$$

If γ = 0, then equation (13) becomes,

$$\hat{F}(u,v)=\left[\frac{H^*(u,v)}{|H(u,v)|^2+0.|P(u,v)|^2}\right]G(u,v)$$

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

Which is the inverse filtering operation. Thus, when γ = 0, constraint least square filter reduces to inverse filter.

Computation of γ:

'γ' is an important parameter which is adjusted until the desired results are obtained. Therefore, the computation of γ value should be done carefully.

Iteration Method:

1) Define a residual vector r given by

$$r=g-H\hat{f}$$

2) Now, it can be proved that ϕ(γ) is a monotonically increasing function of γ defined as,

$$\phi(\gamma)=r^T.r=\left||r|\right|^2$$

3) Next, adjust γ to satisfy the expression

$$\left||r|\right|^2=\left||n|\right|^2\pm a$$

Where, a – Accuracy Factor.

Newton – Raphson Algorithm:

This algorithm is used to improve the speed of restoration process in constrained least square filtering. Two quantities are needed to implement this algorithm.

(i) $||r||^2$
(ii) $||n||^2$

(i) To find $||r||^2$

The steps to find $||r||^2$ are:

i. The Residual vector equation can be rewritten as,

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

ii. Taking the inverse transform of R(u,v) gives r(x,y).

iii. Now, $||r||^2$ can be obtained by the equation,

$$||r||^2 = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} r^2(x,y)$$

(ii) To find $||n||^2$

The steps to find $||n||^2$ are:

i. Calculate the sample mean of the noise from the expression

$$m_n = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y)$$

ii. Find the variance of the noise over the entire image using the sample average method. This is expressed as

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

iii. Now calculate $||n||^2$ by using

$$||n||^2 = MN[\sigma_n^2 - m_n]$$

Advantages:

- Only the mean and variance of the noise are required to be known to implement this method.
- For each input image, it produces an optimal result.

Unit - IV Image Segmentation

- Segmentation subdivides an image into its constitute regions or objects according to a pre-defined condition.
- Segmentation stops once the Region of Interest (ROI) has been isolated.

Detection of Discontinuities:

- The various discontinuities that exist in a gray level image are points, lines and edges.
- By using a 3x3 mask we can detect the various discontinuities.
- The procedure includes calculating the response of the mask, centered at the centre of the region.

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9$$

$$\boxed{R = \sum_{i=1}^{9} w_i z_i}$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Here w_i is the gray level of the image pixels and z_i are the filter co-efficients

Point Detection :

-1	-1	-1
-1	8	-1
-1	-1	-1

- Detection of isolated points is performed by the above mask.
- If the response, at the point on which the mask is centered, is greater than the pre-defined threshold, then it can be assumed that a point has been detected.

$$\boxed{|R| \geq T.}$$

- T is a non-negative threshold.
- This process isolates any pixel whose gray level value is significantly different from the background.
- The mask will produce a zero response for an area of same gray level. This is because the sum of all the co-efficients is zero for the mask.

Detection of Lines :

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

Horizontal 0°

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

+45°

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

Vertical 90°

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

-45°

- The above masks are used to detect lines of different orientations.
- The central line of the masks are heavily weighted and hence gives higher response for the pixels present under it.
- The 0° mask is used for detecting horizontal lines and the 90° mask is used for detecting vertical lines.
- The +45° and -45° masks are used to detect lines oriented at 45° and 135° respectively.
- The sum of all the filter co-efficients is zero.

- The response of various filters is given by :
  R_0 , R_{45} , R_{90} and R_{135}
- If the four masks are ~~made~~ to applied to a single image consisting of lines oriented at a single direction then :
  - if $|R_0| > |R_{45}|, |R_{90}| \& |R_{135}|$
  → we can conclude that a Horizontal Line is present in the image.
  - if $|R_{45}| > |R_0|, |R_{90}| \& |R_{135}|$
  → we can conclude that a Line oriented at 45° is present in the image.
  - Similarly if the response $|R_{90}| > |R_0|, |R_{45}| \& |R_{135}|$
  → we can conclude that a vertical line is present
  - lastly if $|R_{135}| > |R_0|, |R_{45}| \& |R_{90}|$
  → we conclude that a line oriented at 135° is present.
- If we are interested in detecting lines of a particular orientation in an image, then we only use line masks having the same orientation.

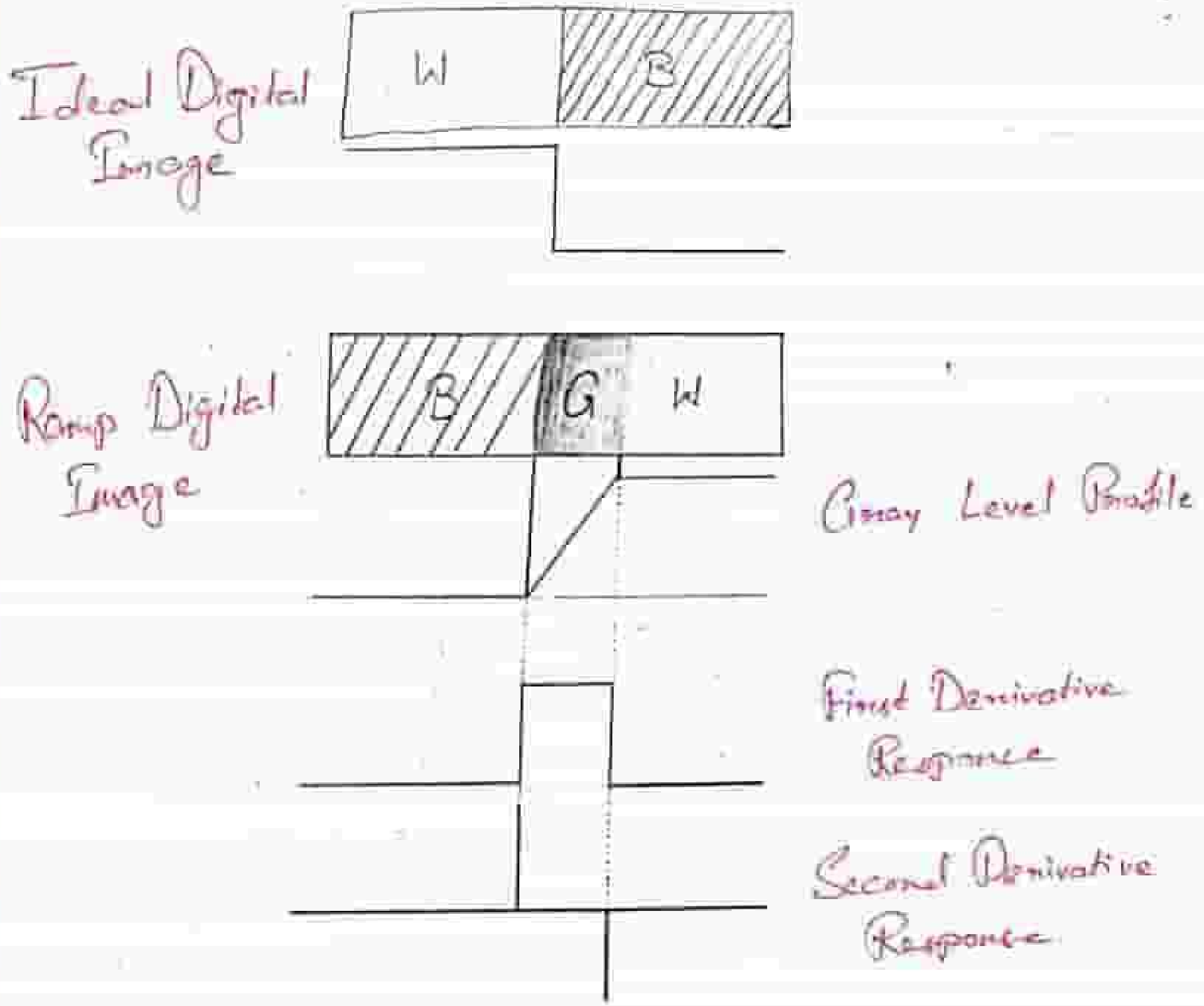
EDGE DETECTION :

- An edge is a set of connective pixels that lie on the boundary between two regions.
- The two types of edges present are :

  Ideal edge : is a set of connected pixels each of which is located at an orthogonal step transition in gray level.

  Ramp edge : is a set of pixels that show a gradual gray level transition from black-to-white. The thickness of edge is determined by the length of the ramp.

- For detection of ramps we used derivative operators. There are 2 kinds of derivative operators, i.e. First and Second order derivative.
- The 1st derivative is (+)ve at the transition in and out of the ramp. It is constant for the points on the ramp. It gives 0 response for areas with constant gray level.
- The 2nd derivative gives a (+)ve spike at transition associated with the dark side of the edge and a (-)ve transition associated with the light side of the edge. The response is 0 for points on the ramp and having a constant gray level.



- First derivative magnitude can be used to detect the presence of an edge at a point in an image.
- The sign of second derivative can be used to determine whether the edge pixel lies on the dark or light side of the image.
- The zero-crossing property of a 2° derivative is useful in determining the center of a thick edge.

- The gradient of an image $f(x,y)$ is given as a vector :

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

- The magnitude of a 1st derivative is given by ;

$$\nabla f = mag(\nabla f) = \left[G_x^2 + G_y^2 \right]^{1/2}$$

- The direction of the gradient vector is given by ;

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

- Computation of the gradient of an image is based on computing the partial derivatives $\partial f / \partial x$ and $\partial f / \partial y$ at every pixel location.

- The simplest way to obtain a first-order-derivative at a point (z_5) is by using Roberts cross gradient operators :

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$G_x = (z_9 - z_5)$$

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

-1	0
0	1

G_x

0	-1
1	0

G_y

- A 2×2 mask like Robert's Operator is not having a clear center. So we go for a 3×3 mask.

- The basic 3×3 mask is a Prewitt operator, given by the formula:

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

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

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

G_x

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

G_y.

- With a slight variation, i.e. by multiplying the central coefficient of the mask by 2. The operators are called as Sobel Operators. The perform noise removal.

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

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

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

G_x

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

G_y.

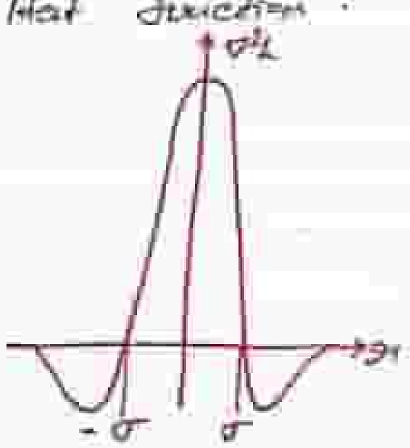
- Hence $r^2 = x^2 + y^2$ and σ is the standard deviation.
- The amount of blurring depends on the value of σ.
- Now the Laplacian is applied to the blurred image $h(r)$. The resulting function is given as;

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

- This function is called as Laplacian of Gaussian (LoG).
- The LoG function is digitally represented by the mask;

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

- Due to the shape of the LoG function is sometimes referred as the Mexican Hat function.



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

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

- A 3x3 Laplacian mask used for edge detection is given by the formula;

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

or

$$\nabla^2 f = 8z_5 - [z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9]$$

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

- The Laplacian operator is not used for edge detection because it is highly ~~and~~ susceptible to noise and it produces undesirable double edges.
- Moreover it is unable to detect the direction of edges.
- Hence the Laplacian is combined with a smoothing process as a precursor.
- The smoothing function is given by $h(x) = -e^{-\frac{x^2}{2\sigma^2}}$

EDGE LINKING & BOUNDARY DETECTION.

- The edge detection methods are followed by edge linking procedure to group the pixels into meaningful edges.

Local Processing –

- In Local processing, analysis of the pixels in a region, characterized as an edge, is done.
- All the pixels that are similar according to a pre-defined criteria is linked.
- This forms an edge that is continuous in nature.
- The two principal properties that are analyzed are
  ① The strength of the response, i.e. the magnitude of the gradient operator "∇f".
  ② The direction of the gradient vector.
- Therefore, if we consider two pixels (x_0, y_0) and (x, y). Then we can say that they lie in the same edge region if;

$$|\nabla f(x,y) - \nabla f(x_0,y_0)| \leq E \quad \text{and}$$

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

- Here 'E' is a non-negative magnitude threshold and 'A' is a non-negative angle threshold.
- Any two pixels satisfying the above given criteria are joined forming an edge.

GLOBAL PROCESSING VIA HOUGH TRANSFORM :

- Here the edge pixels are linked only if they lie on a curve of specified shape.
- If we take 'n' number of points, then to analyze whether they are ~~belonging to a~~ connected via one single line we have to find $n(n-1) \sim n^2$ lines and then perform $(n)(n(n-1))/2 \sim n^3$ comparisons of every point to all lines.
- To reduce the computation time / steps we go for Hough Transform.

infinite no.

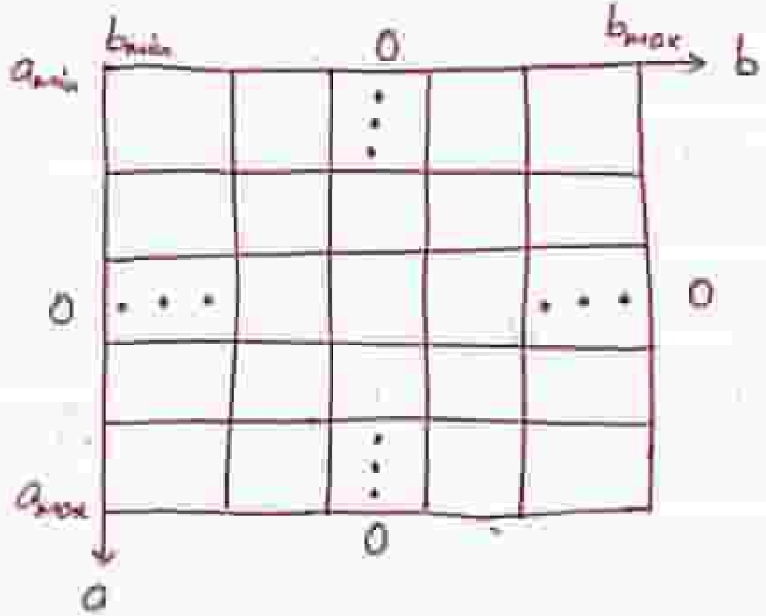
- According to Hough Transform, the lines passing through (x_i, y_i) are given by the equation ; $y_i = ax_i + b$. If the same equation is written in the a-b plane or the parameter space, it is given by $b = -x_i a + y_i$.

- The resulting 'b' values are then rounded-off to the nearest value in the b-axis.

- If a value of 'a' say 'a_p' yields a solution 'b_q', we let $A(p,q) = A(p,q) + 1$. This increases the total count of the points in the accumulator cell.

- If at the end of the computation there are Q values present in an accumulator cell $A(i,j)$. This concludes that there are Q points in the X-Y plane lying on the line $y = a_i x + b_i$.

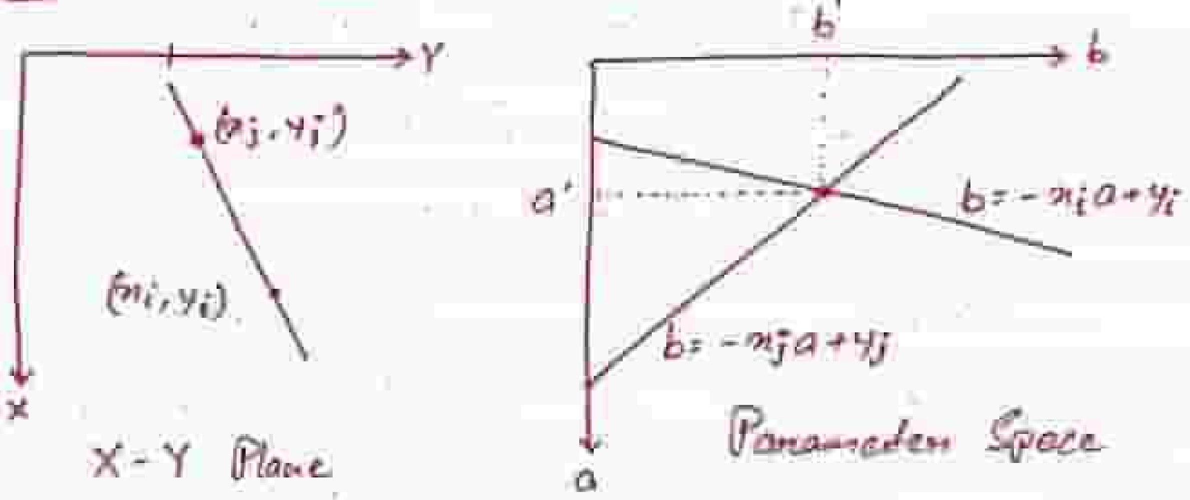
- The total number of computation points for an image having n-pixels and K-subdivisions in the parameter space is equal to 'nK'.



∴ The equation $b = -x_i a + y_i$ defines that only one line passes through the point (x_i, y_i) in the parameter space.

- If we consider another point (x_j, y_j), then the line passing through (x_j, y_j) intersects the line passing through (x_i, y_i) at (a', b').

- Here a' is the slope and b' is the intercept.



- The parameter space is divided into Accumulator Cells. The cell at co-ordinates (i, j) is given by $A(i, j)$.

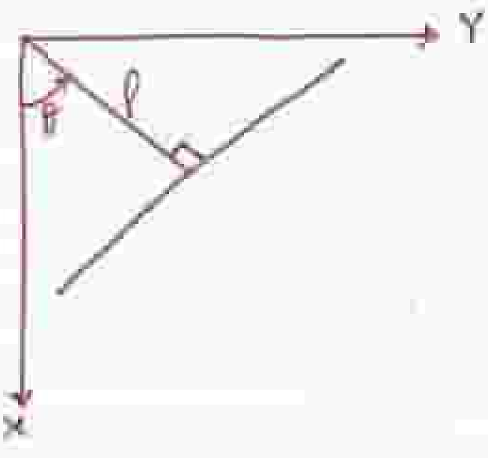
- The initial values of the Accumulator Cells is given as zero (0). For every point in the image (x_k, y_k), we calculate the value of 'b' using the formula
$$b = -x_k a + y_k.$$

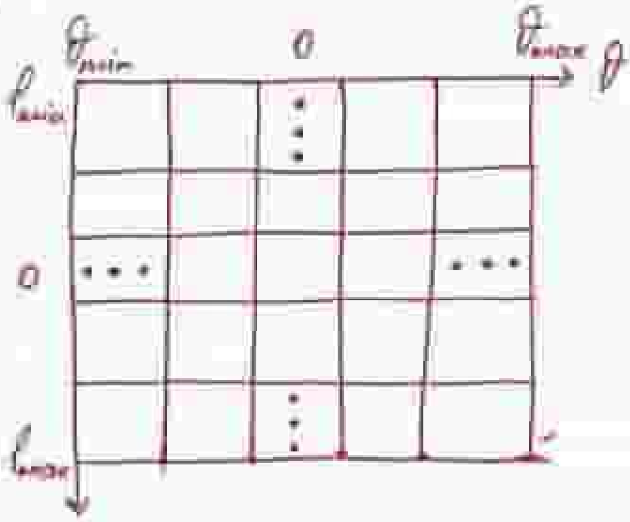
- The parameter 'a' equal to each of the allowed subdivisions values on the a-axis.

- The disadvantage of using the equation $y = ax + b$ to solve Hough Transform is that the slope reaches '∞' as the line becomes vertical.
- To overcome this problem we go for the following equation:

$$\rho = x\cos\theta + y\sin\theta.$$

- The method of solving by constructing a table of accumulators is the same as the slope - intercept method.
- Here the loci are sinusoidal curves in the $\rho\theta$ - plane.





- Here the values of ρ is calculated by incrementing the θ values. Thus at the end of the process there will be a total of Q entries in an accumulator cell $A(i, j)$.
- Hough Transform is applied to any function of the form $g(v, c) = 0$.

- For the points lying on a circle given by the equation;

$$(x-c_1)^2 + (y-c_2)^2 = c_3^2 .$$

- Though the overall process of Hough Transform remains same. The only difference is in the shape of the accumulator cells.
- As they have to accomodate 3 values i.e. c_1, c_2 and c_3, the accumulator cells have a 3-D structure or cube like structure.

- The whole process of Hough Transform can be explained in the following 4 steps.

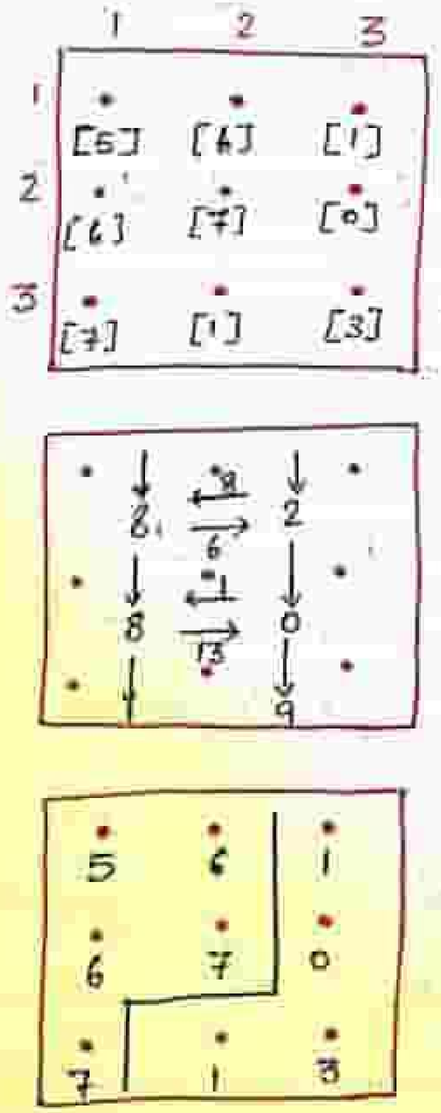
1. Compute the gradient of an image and threshold it to obtain a binary image.
2. Specify sub-divisions in the $\rho\theta$-plane
3. Examine the count of accumulator cells for high pixel concentrations.
4. Examine the relationship between the pixels.

- An edge element is a boundary between 2 pixels p and q. Where, p and q are 4-neighbours.

- The cost associated with each pair of pixel is given by the formula;

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

- Where H is the highest gray level value in the given region of image and $f(p)$ & $f(q)$ are the gray level values of the pixels p and q.



- Here a graph shows pixels on nodes having particular gray level values given in the bracket '[ ]'.
- We start calculating the cost of the paths from the top-most row, i.e. (1,1) (1,2) and (1,2) (1,3).
- For calculating the cost of the right and left path we denote the pixel lying on the right side of the path as 'p' and left side as 'q'.
- By joining all the lowest cost paths we generate an edge segment that starts from the first row and ends at the last row.

GLOBAL PROCESSING VIA GRAPH-THEORETIC METHOD.

- Here the edge segments are represented into the form of a graph and searching the graph for a low-cost path corresponding to significant edges.

- This method produces reliable edge linking in the presence of noise. But this method has a higher processing time.

- A Graph $G = (N, U)$ is a finite non-empty set of ~~nodes finite non-empty~~ nodes 'N' and a set of 'U' unordered pairs of distinct elements of 'N'.

- Two points in a graph is called a pair (n_i, n_j) and each pair of U is called an 'arc'. A graph which has arcs directed then the graph is called a directed graph.

- If an arc is directed from n_i to n_j, the n_i is the parent node and n_j is the successor node. The process of identifying a successor node is called expansion of the node.

- By expansion of nodes, a sequence of nodes are generated in the form of $n_1, n_2 \ldots n_k$. this is called a path from n_1 to n_k. The cost of the entire path is given by the formula: $C = \sum^{k} c(n_{i-1}, n_i)$.

REGION BASED SEGMENTATION

- Here segmentation is performed by finding regions directly based on the following conditions;
When an image represented by 'R' is segmented into n subregions. $R_1, R_2 \ldots R_n$

ⓐ $\bigcup_{i=1}^{n} R_i = R$.

ⓑ R_i is a connected region, $i = 1, 2, \ldots, n$.

ⓒ $R_i \cap R_j = \phi$ for all i and j & $i \neq j$.

ⓓ $P(R_i)$ = True for $i = 1, 2, \ldots, n$.

ⓔ $P(R_i \cup R_j)$ = False for $i \neq j$.

- $P(R_i)$ is a logical predicate defined over the pixels in the region R_i.
- For example $P(R_i)$ = True if all pixels in R_i have the same gray level.

REGION GROWING

- It is a procedure that groups pixels or subregions into larger regions based on predefined criteria.
- The most used approach for Region Growing is to start with a set of 'seed' points and then growing the region by connecting all the neighbouring pixels having similar properties as the 'se...

- In the absence of information about the image pixels, we have to check all the pixels for a predefined condition. These pixels which have the same set of properties can be called as the 'seed' points.

- The major descriptors necessary for region analysis are:
  1. Gray level information.
  2. Spatial properties such as moments and textures.
  3. Connectivity information.
  4. Pixel adjacency information.

- The formulation of the stopping criteria is a major step in successful region growing process.

- For generating a robust stopping criteria, we need to have knowledge ~~abt~~ about the following descriptors:
  1. History of region growth.
  2. Concept of size.
  3. The shape of the region being grown.
  4. Comparison of the gray level of candidate pixel and the average gray level of the grown region.

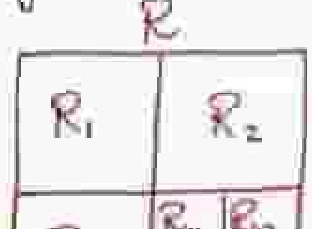
Region Splitting & Merging :

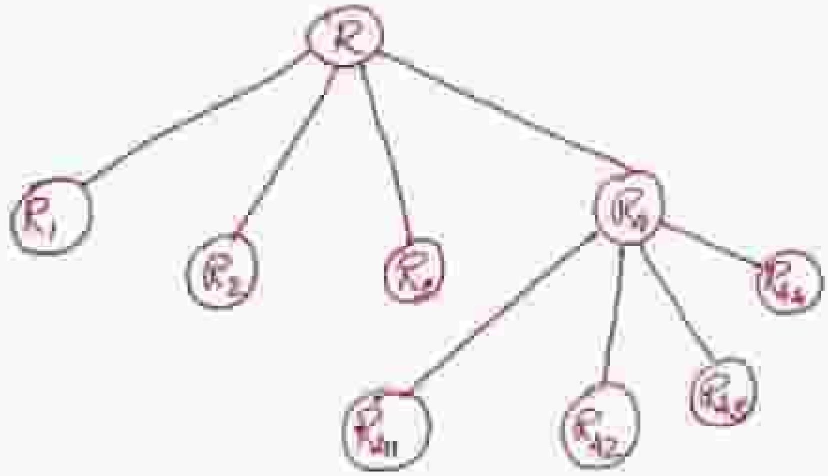
- In this method of region formation, images are sub-divided into a set of arbitrary, disjointed regions initially.

- Then merge or split the regions to satisfy a pre-defined condition.

- The following procedure can be used to perform region splitting and merging :

  1. Split into 4 disjoint quadrants any region R_i for which $P(R_i) = \text{False}$.
  2. Merge any adjacent regions R_j and R_k for which $P(R_j \cup R_k) = \text{True}$.
  3. Stop when no further merging or splitting is possible.





- When R is the original image and P is the predicate then we sub-divide R into smaller segments R_i, such that $P(R_i)$ is True.

- If $P(R_i)$ = False , after the initial arbitrary segmentation , then we further sub-divide the quadrant into sub-quadrants.

- This splitting technique is generally termed as quadtree method of segmentation.

- The root of the tree corresponds to the entire image and each node corresponds to a single sub-division.

- Splitting is followed by merging process. Two regions are merged only and only if they satisfy the same predicate.

- For 2 regions R_j and R_k , they are merged if $P(R_j \cup R_k)$ = True.

THRESHOLDING.

It plays a central position in application of image segmentation.

Foundation

Previously regions were identified by first finding edge segments and then attempting to link the segments into boundaries.

Now in thresholding, partitioning images directly into regions based on intensity values / properties of these values.

Basics

Any point (x,y) in the image at which $f(x,y) > T$ is called object point otherwise background point.

segmented image.

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

When T is a constant applicable on an entire image, then the process in this equation is referred to as global thresholding.

When the value of T changes over an image it is variable Thresholding.

When the variable thresholding in which the value of T at any point (x,y) in an image depends on properties of a neighborhood of (x,y) (the average intensity of pixels in the neighborhood) is called Local (or) Regional thresholding.

If T depends on the spatial coordinates (x,y) themselves, then variable thresholding is often referred to as dynamic (or) adaptive thresholding.

Multiple Thresholding - It is the threshold obtained based on 2 types of objects.

eg two types of light objects on a dark background.

(x,y) as belonging to the background if $f(x,y) \leq T_1$, if $T_1 < f(x,y) \leq T_2$ and other object class if $f(x,y) > T_2$

So the segmented image

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) > T_2 \\ b & \text{if } T_1 < f(x,y) \leq T_2 \\ c & \text{if } f(x,y) \leq T_1 \end{cases}$$

where a, b, c are any distinct intensity value.

The success of intensity thresholding is directly related to the width & depth of the values. The key factors affecting the properties of value

i) The separation between peaks.

ii) Noise content of an image.

iii) The relative size of objects & background.

iv) The uniformity of the illumination source.

v) The uniformity of the reflectance properties of the image.

<u>Basic Global thresholding</u>

Single global thresholding applicable over the entire image.

<u>Automatic thresholding algorithm.</u>

1) Select an initial estimate for the global threshold T

2) Segment the image using T as per

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

This produce two groups of pixels;

G_1 - consisting of all pixels with intensity value $> T$

G_2 - consisting of pixels with value $\leq T$.

3) Compute the average (mean) intensity values m_1 and m_2 for pixels in G_1 & G_2 respectively.

4) Compute the new threshold value

$$T = \frac{1}{2}(m_1 + m_2)$$

5) Repeat steps 2 to 4 until the different between values of T in successive iterations is smaller than a predefined parameter ΔT.

Global Thresholding Using Otsu's Method.

The objective is to minimise the average error incurred in assigning pixels to two (or) more groups (also called classes). The solution is based on only two parameters; the PDF (Probability Density Function) of intensity levels of each class and the probability that each class occurs in a given application.

Let $\{0, 1, \ldots \ldots L-1\}$ denote the L-distinct intensity levels in a digital image of size $M \times N$ pixels.

$n_i \rightarrow$ denotes the number of pixels with intensity i

The total number, MN of pixels in the image is

$MN = n_0 + n_1 \ldots \ldots + n_{L-1}$

Normalised histogram has components $P_i = n_i / MN$

4	3	2	1	0
5	4	3	2	1
1	1	1	2	4
2	2	1	4	5

$P_{(1)} = 6/20$

$P_{(2)} = 5/20$

$P_{(1)} = 4/20$

Global variance (i.e. - the intensity variance of all the pixels in the image)

$$\sigma^2_G = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$$

Between class variance

$$\sigma^2_B = P_1 (m_1 - m_G)^2 + P_2 (m_2 - m_G)^2$$

$$= P_1 P_2 (m_1 - m_2)^2$$

$$\sigma^2_B = \frac{(m_G P_1 - m)^2}{P_1 (1 - P_1)}$$

$$\sigma^2_B (k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

$k^* \rightarrow$ Otsu's threshold as the value of k for which $\sigma^2_B(k)$ is maximum

Final result $\eta(k) = \dfrac{\sigma^2_B(k)}{\sigma^2_G}$ at $k = k^*$

$$0 \leq \eta(k^*) \leq 1.$$

Algorithm

1) Compute the normalised histogram of input image
2) Compute the cumulative sums.
3) Compute the cumulative means.
4) Compute the global intensity means.
5) Compute the global variance
6) Compute the between class variance.

$$\sum_{i=0}^{L-1} P_i = 1 \quad P_i \geq 0$$

Threshold $T(k) = k$, $0 < k < L-1$. use it to threshold the input image in to two classes C_1 & C_2 where $C_1 \rightarrow$ consist of all the pixels in the image with intensity value in $[0, k]$ and $C_2 \rightarrow$ consists of the pixels with the values in the range $(k+1, L-1)$

Cumulative sum $P_1(k)$ for $k = 0, 1, 2, \ldots L$

using
$$P_1(k) = \sum_{i=0}^{k} P_i$$

$$P_2(k) = \sum_{i=k+1}^{L-1} P_i$$

$$= 1 - P_1(k)$$

mean intensity value of pixels

$$m_1(k) = \sum_{i=0}^{k} i\, P(i/C_1) \rightarrow$$ Probability value of i, given that i come from C_1

$$m_2(k) = \sum_{i=k+1}^{L-1} i\, (P_i/C_2) \rightarrow$$ belong to class C_2

Cumulative mean (average intensity) up to level k

$$m(k) = \sum_{i=0}^{k} i\, P_i$$

Average intensity of the entire image (global mean)

$$m_G = \sum_{i=0}^{L-1} i\, P_i$$

$$P_1 m_1 + P_2 m_2 = m_G$$

3) Threshold the image using T value to produce a binary image $g_T(x,y)$

4?) This $g_T(x,y)$ is used as a mask k, image in following steps to select pixels from $f(x,y)$ corresponding to strong edge pixels.

4) Compute the histogram using only the pixels in $f(x,y)$ that corresponds to the location of the pixels in $g_T(x,y)$

5) Use the histogram to segment $f(x,y)$ globally using Otsu's method.

7) Obtain Otsu's threshold, k^* as the value of k for which $\sigma^2_B(k)$ is maximum.

8) Obtain the separability measure, η^*

Image Smoothening to improve Global Thresholding.

Noise in the image can turn a simple thresholding problem into an unsolvable one, when noise cannot be reduced at the source, and thresholding is the segmentation method of choice. Smooth the image prior to thresholding.

Each black point in one white region of white point in black region is the thresholding error. This can be rectified through image smoothening.

Edges to improve Global Thresholding

One approach for improving the shape of histogram is to consider only those pixels that lie on (or) near the edges between objects and the background. So the resulting histogram have peaks of approximately the same height.

Algorithm

1) Compute an edge image using magnitude of gradient or by computing laplacian of $f(x,y)$

2) Specify a threshold value T.

IMAGE COMPRESSION

Image compression is defined as the process of reducing the amount of data needed to represent a digital image. This is done by removing the redundant data. *The* ***objective of image compression*** *is to decrease the number of bits required to store and transmit without any measurable loss of information.*

Two types of digital image compression are

1. Lossless (or) Error – Free Compression
2. Lossy Compression

Need for Data Compression:

1. If data can be effectively compressed wherever possible, significant improvements in data ***throughput*** can be achieved.
2. Compression can reduce the file sizes up to 60-70% and hence many files can be combined into one compressed document which makes the ***sending*** easier.
3. It is needed because it helps to reduce the consumption of excessive resources such as ***hard disc space*** and ***transmission bandwidth***.
4. Compression can fit more data in small memory and thus it reduces the ***memory space*** required as well as the ***cost*** of managing data.

5.1 Data Redundancy:

Relative Data Redundancy:

Let there are two data sets that represent the same information and n_1 and n_2 are the number of information carrying units (i.e. bits) in the data sets.

The relative data redundancy R_D is given by,

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

Where, C_R – Compression Ratio

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

Based on n_1 and n_2 values there can be three different cases as below,

i. If $n_2 = n_1$, no redundant data is present in first set (i.e. Input image)
ii. If $n_2 \ll n_1$, highly redundant data is present in first set.

iii. If $n_2 \gg n_1$, highly redundant data is present in second set.

5.1.1 <u>Types of Redundancies:</u>

(i) <u>Coding Redundancy</u>:

A code is system of symbols used for representing information. A code word is a sequence of symbols representing a piece of information. Thus, the code length is defined as the number of symbols in each code word. ***A resulting image is said to have coding redundancy if its gray levels are coded using more code symbols than actually needed to represent each gray level.***

Average length of code words:

The gray-level histogram of an image is used to construct codes to reduce data redundancy. Multiplying the number of bits used to represent each gray level with the probability of occurrence of that gray level and adding all such products gives the average length of code words assigned to different gray levels. The average number of bits needed to represent each pixel is given below,

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

Where, $l(r_k)$ – Number of bits used to represent each gray level

$P_r(r_k)$ – Probability of occurrence of gray level r_k

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

Where, L – Number of gray levels

n_k – Number of times the k^{th} gray level appear in image

n – Total number of pixels

Total number of bits:

The total number of bits required to code an M x N image is given by,

$$L_{MxN} = M.N.L_{avg}$$

To Avoid Redundancy:

Coding redundancy can be avoided and thus data compression can be achieved by assigning fewer bits to more probable gray levels and more bits to less probable gray levels.

(ii) Interpixel Redundancy:

It's caused by the interpixel correlations within an image. Interpixel correlations are the structural and geometric relationships between objects in the image.

1. Autocorrelation Coefficient:

The autocorrelation coefficients can be computed using,

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

Where, A(Δn) – Scaling factor given by

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

Where, n - Number of pixels

` ***To Avoid Redundancy***

Inter-pixel redundancies in an image can be reduced by transforming the 2-D pixel array into a more efficient format. Transformations used for this purpose are known as mappings. If the original image elements can be reconstructed from the transformed data set, the mappings are called reversible mappings.

(iii) Psychovisual Redundancy:

The human eye doesn't respond to all information with equal sensitivity. Because, some information may be given less importance when comparing to other information in normal visual processing. Such information is said to be psychovisually redundant.

5.1.2 Fidelity Criteria:

Fidelity criteria is used evaluate the information loss during data compression

Need:

There is a loss of real or quantitative visual information when redundancy is removed. Thus, there is a chance for losing the information of interest. Therefore, there is a need to quantify the nature and extent of the information loss using repeatable and reproducible criteria.

Types:

i. Objective Fidelity Criteria
ii. Subjective Fidelity Criteria

(i) Objective Fidelity Criteria:

The objective fidelity criterion expresses the level of information loss as a function of two parameters. They are

- The original or input image
- The compressed and successively decompressed output image

RMS Error:

The root-mean-square (rms) error between the input and output image is a very good example for objective fidelity criteria.

The error between two images f(x,y) and $\hat{f}$(x,y) is given by

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

The total error between two MxN size images is

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

The root-mean-square error e_{rms} between $\hat{f}$(x,y) and f(x,y) is obtained by

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

Advantage:

- Simple and convenient technique to evaluate information loss

(ii) Subjective Fidelity Criteria

Since human perception is based on subjective quality, it is more suitable to use subjective fidelity criteria to evaluate the information loss.

Concept:

- This method is implemented by showing a ‘typical’ decompressed image to a number of viewers.
- Then, their evaluation is averaged to take a decision.

- These evaluations may be done using a rating scale or side-by-side comparisons such as excellent, better, same, worse, etc.

5.2 Image Compression Models:

Fig. 5.1 shows, an image compression system is composed of two distinct functional components: an encoder and a decoder. The encoder performs compression, and the decoder performs the complementary operation of decompression.

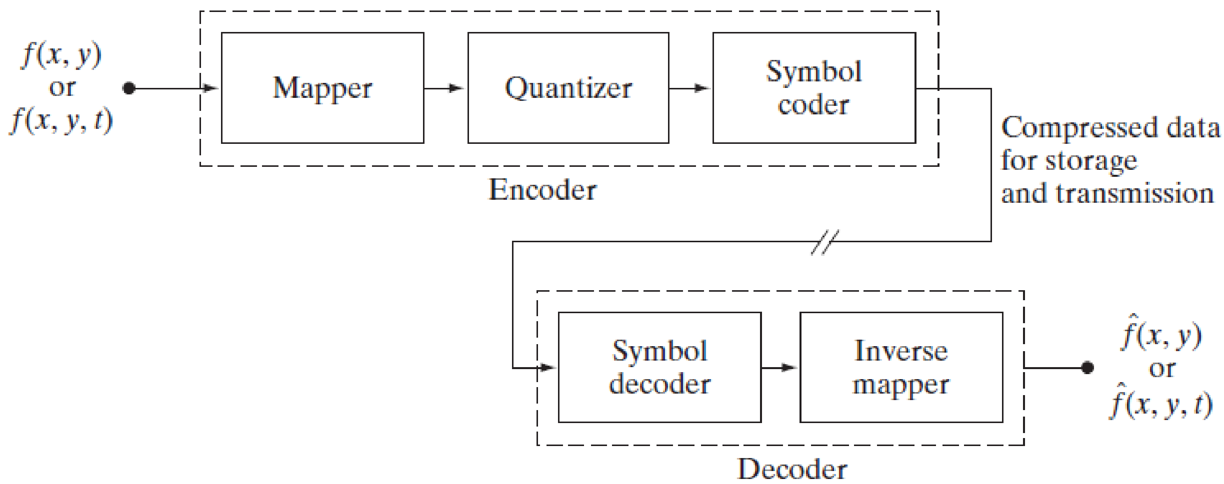


Fig: 5.1 Functional block diagram of a general image compression system.

Input image f(x,y) is fed into the encoder, which creates a compressed representation of the input. This representation is stored for later use, or transmitted for storage and use at a remote location. When the compressed representation is presented to its complementary decoder, a reconstructed output image $\hat{f}(x, y)$ is generated. In general $\hat{f}(x, y)$, may or may not be an exact replica of f(x,y). If it is, the compression system is called *error free*, *lossless*, or *information preserving*. If not, the reconstructed output image is distorted and the compression system is referred to as *lossy*.

5.2.1 Source Encoder and Decoder:

The Encoding / Compression Process: The encoder of Fig. 5.1 is designed to remove the redundancies.

A ***mapper transforms*** f(x,y) into a (usually nonvisual) format designed to reduce spatial and temporal redundancy. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image. Run-length coding is an example of a mapping that normally yields compression in the first step of the encoding process. The mapping of an image into a set of less correlated transform coefficients.

The ***quantizer*** in Fig. 5.1 reduces the accuracy of the mapper's output in accordance with a pre-established fidelity criterion. The goal is to keep irrelevant information out of the compressed representation.

the ***symbol coder*** of Fig. 5.1 generates a fixed- or variable-length code to represent the quantizer output and maps the output in accordance with the code. In many cases, a variable-length code is used. The shortest code words are assigned to the most frequently occurring quantizer output values thus minimizing coding redundancy. This operation is reversible. Upon its completion, the input image has been processed for the removal of each of the three redundancies

The Decoding / Decompression Process:

The decoder of Fig. 5.1 contains only two components: a symbol decoder and an inverse mapper. They perform, in reverse order, the inverse operations of the encoder's symbol encoder and mapper. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general decoder model. In video applications, decoded output frames are maintained in an internal frame store and used to reinsert the temporal redundancy that was removed at the encoder.

5.3 <u>Error Free Compression / Lossless Compression:</u>

- Error-free compression is the acceptable data reduction method since there is no data loss.
- This method is applicable to both binary and gray-scale images.
- It provides compression ratios ranging from 2 to 10.

Operations:

(i) Forming an alternative representation for the given image by which its ***interpixel redundancies*** are reduced

(ii) Coding the representation to remove its ***coding redundancies.***

Methods:

Four important methods to achieve lossless or error free compression are:

- Variable-Length Coding
- Bit-Plane coding
- LZW Coding
- Lossless Predictive Coding

5.3.1 <u>Variable-Length Coding:</u>

- Variable-Length coding assigns shortest possible code words to the most probable gray levels and vice versa.
- It is the simplest approach to reduce coding redundancy that is present in any natural binary coding of gray levels.

Objective:

The objective is to code the gray levels of an image in such a way that the average number of bits required to represent each pixel, L_{avg} should be reduced.

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

The entropy of the generated code can be found by,

$$H = \sum_{k=0}^{L-1} p_r(r_k).\log_2(\frac{1}{p_r(r_k)}) \qquad (2)$$

5.3.1.1 Huffman Coding:

Huffman coding provides the smallest number of code symbols per source i.e. original symbol. Therefore, it is the most popular technique to remove coding redundancy. It is introduced by Huffman in the year 1952.

Coding Procedure:

Let a set of six source symbols with probabilities given below are to be coded.

$$\{a_1, a_2, a_3, a_4, a_5, a_6\} = \{0.1, 0.4, 0.06, 0.1, 0.04, 0.3\}$$

(i) ***Step 1:*** The first step is to order the probabilities of the given symbols in the descending order. Now, the source symbol becomes

$$\{a_2, a_6, a_1, a_4, a_3, a_5\} = \{0.4, 0.3, 0.1, 0.1, 0.06, 0.04\}$$

(ii) ***Step 2:*** Next, a source reduction is created by adding the lowest i.e. bottom two probabilities into a single symbol. This symbol is known as the compound symbol. Then, the two probabilities are replaced by the compound symbol and its probability in the next source reduction

Steps 1 and 2 are repeated until a source with only two symbols is obtained. These steps are illustrated in Table 5.1 for the given source symbols.

For example, in the third source reduction step, the lowest two probabilities are 0.3 and 0.3. These two are added to form a compound symbol with probability 0.6. thus, the probabilities are replaced by 0.6 in the fourth reduction step. At last they are ordered in the decreasing order.

(iii) ***Step 3:*** In this step, each reduced source is coded. It starts from the smallest source obtained in the last step and goes back to the original source.

The minimal length binary codes used are: 0 and 1

This Huffman code assignment procedure is illustrated in Table 6.2

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Table 5.1: Huffman code assignment procedure.

Here, the reduced symbols 0.6 and 0.4 in the last column are assigned 0 and 1 first.

- Since 0.6 was generated by adding 0.3 and 0.3 in the third column a '0' and '1' are appended.
- Then a '0' and '1' are appended with 01 since its symbol 0.3 was generated by adding 0.2 and 0.1 in the second column. This produces the codes 010 and 011.

This procedure is repeated until it reaches the original symbols as shown in Table 5.2.

5.4 <u>Lossy Compression</u>:

Lossy compression methods can achieve high rates of compression. But they reduce the accuracy of the reconstructed images by producing some distortions. The distortions produced may be visible or invisible.

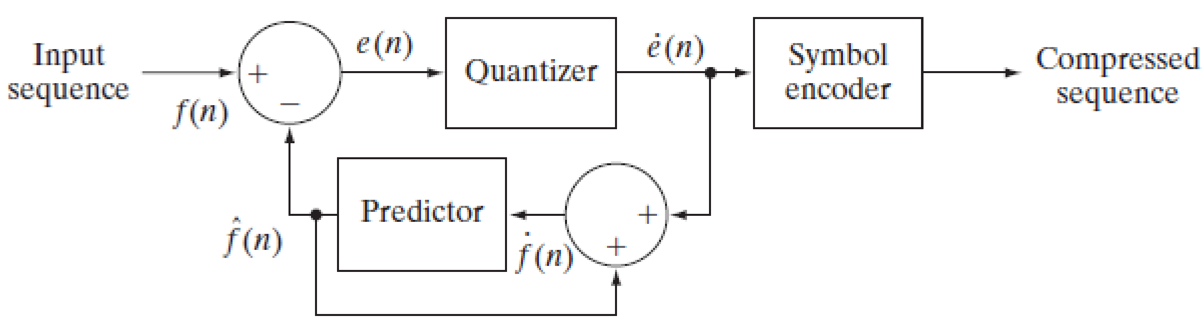
5.4.1 Lossy predictive coding:

Lossy predictive coding is a spatial domain method, because it operates directly on image pixels. The lossless predictive coding model contains.

- An encoder and
- A decoder

<u>Encoder:</u>

The encoder block diagram of the lossy predictive coding is shown in figure



The differences of this encoder from the lossless encoder are that the nearest integer block is replaced by a quantizer and the predictor has feedback.

Quantizer:

- The quantizer is placed between the symbol encoder and the prediction error point.
- It does the function of nearest integer block in the lossless predictive encoder.
- This quantizer maps the prediction error e_n into a limited range of outputs, $\dot{e}_n$.

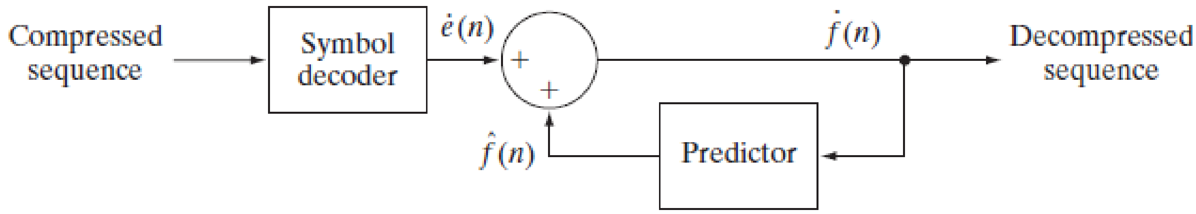
Predictor:

- The predictor and quantizer are designed independently of each other.
- The predictions produced by the encoder and decoder should be equivalent. For this purpose, the predictor of this encoder is placed within the feedback loop.
- Its input $\dot{f}_n$ is a function of past predictions and the corresponding quantized errors.

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

Decoder:

The decoder block diagram of the lossy predictive coding is shown in figure



The error which will be produced at the output of the decoder is avoided as the predictor of encoder is placed in a closed loop.

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