Roozahh Stha
Cosmos College

Chapter 1

Introduction to Digital Image Processing

- Digital Image:

⇒ A digital Image is a representation of a 2D image as a finite set of digital values called picture elements or pixels.

⇒ pixel values typically represent gray levels, colors, heights, opacity, etc.

- Digital Image Processing:

⇒ It is the processing of digital image by means of digital.

⇒ The field of Digital Image processing is related to the extracting of the attribute of the image, processing, the attributes and classification and recognization of object / Image.

- Digital Image Representation

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,m) \\ \vdots & & & \\ f(n,0) & f(n,1) & \cdots & f(n,m) \end{bmatrix}$$

⇒ A digital image is an image $f(x,y)$ that is discrete in both spatial co-ordinate & its brightness.

⇒ It is represented by 2D integer array or a series of 2D arrays for each color band.

⇒ The digitized brightness values is called gray levels.

⇒ Row & column of mixture (2D array) identify a point in the image and the corresponding matrix. element value identifies the gray levels at that point.

* Examples of fields that use Digital Image Processing:

1) Image Sharpening & restoration:

⇒ Image sharpening & restoration refers here to process images that have been captured from the modern camera to make them a better image or to manipulate those image in way to achieve desired result.

⇒ It refers to do what photoshop usually does.

⇒ This includes zooming, blurring, sharpening, grayscale to color conversion, detecting edges and vice versa, image retrieval or image recognition.

2) Medical field:

⇒ The common applications of DIP in the field of medical is

a) Gamma ray imaging
b) PET scan
c) X-ray imaging
d) Medical - CT
e) UV imaging
f) EEG, ECG, etc.

3) Office automation:

⇒ It plays an important role in any organization to automate their work.
⇒ Eg: optical character recognition document processing, logo or icon recognition.

4) Geographic Information System:

⇒ DIP techniques are used extensively to manipulate satellite imagery.
⇒ Terrain classification
⇒ Meterology.

5) Industrial Inspection

⇒ Human operators are expensive, slow and unreliable

⇒ Thus, machines are used for work purpose.

⇒ It is also used for non-destructive testing, oil & natural gas exploration testing.

⇒ Thus, industrial vision systems are used in all kinds of industries.

(6) Artistic effect:

⇒ Artistic effects are used to make images more visually appealing to aid special effects and to make composite images.

(7) PCB inspection:-

⇒ Machine inspection is used to determine that all components are present & that all shoulder joints are acceptable.

(8) Law enforcement:

⇒ Image processing techniques are used extensively by law enforcement.
⇒ Numer plate recognition for speed cameras/ automated toll system.
⇒ finger print recognition.
⇒ Enhancement of CCTV images.

(9) HCI (Human Computer Interface):

⇒ tries to make HCI more natural
⇒ It is used for face recognition and gesture recognition.

(10) Pattern recognition:

⇒ It involves the study from image processing and various other fields that include machine learning (a branch of artificial learning)

⇒ In pattern recognition, image recognition is used for identifying the processes in an image and then machine learning is used to train the system for the change in pattern.

⇒ Pattern recognition is used in computer Aided Diagonosis.

⇒ Recognition at handwriting, Recognition of image etc.

(11) Video processing:

⇒ A video is nothing but just very fast movement of picture.

⇒ The quality of the video depends on the no. of frames / pictures per minute & the quality of each frames being used.

⇒ Video processing involves noise reduction, detail enhancement, motion detection, frame rate conversion, aspect ratio conversion, color space conversion.

(12) Machine / Robot Vision:

⇒ Apart from many challenges that a robot face today, one of the biggest challenges still is to increase the vision of robot.

⇒ Make robot able to see things, identify them etc.

→ Much work has been contributed by this field and a complete other of computer vision has been introduced to work on it.

Fundamental steps in Image processing

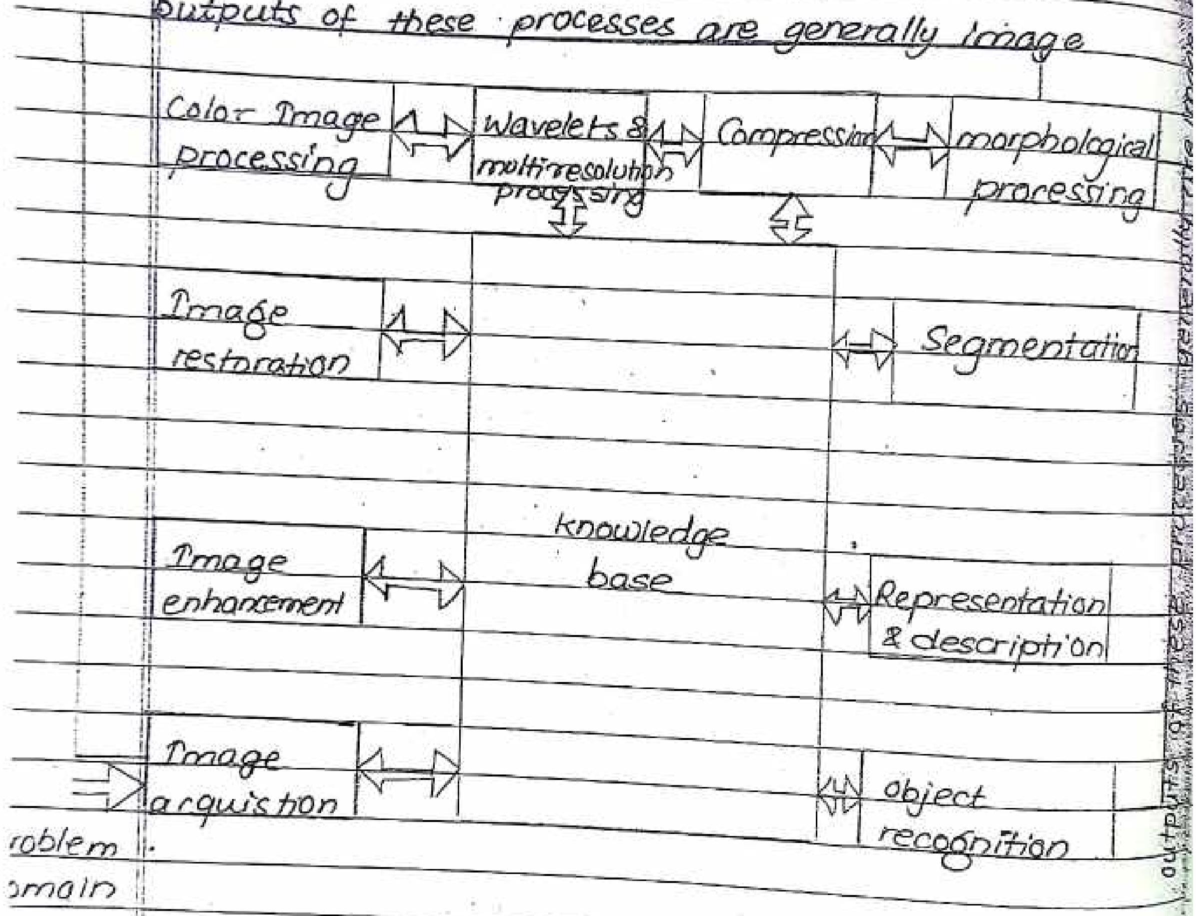


Fig:- Fundamental steps in Digital Image Processing

1. Image acquistion

⇒ This is the first process and could be as simple as being given an image that is already in digital form.

⇒ Generally, the image acquistion involves preprocessing such as scaling.

2. Image enhancement

⇒ It is the process of manipulating an image so that result is more suitable than the original for a specific application.

⇒ The word specific is important here because it establishes at the outset than enhancement techniques are problem oriented.

⇒ Thus for eg: a method that is quite usezul for enhancing x-ray image may not be the best approach for enhacing satellite images taken in the ingrared band of the electro-magnetic spectrum.

3. Image restoration.

⇒ It is an area that also deals with improving the appearances of an image by eliminating the noise of a image.

⇒ It is the objective process based on mathematical & probalistic model.

4. Color Image Processing

⇒ It is an area that has been gaining an importance because of the significant increase in the use of digital image over internet.

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

5. Wavelets and multi resolution processing

⇒ Wavelets are the foundation for the representation images in the various degree of resolution.

⇒ The output of this step will be enhance.

6. Image compression.

⇒ It deals with techniques for reducing the storage required to save an image or the band width required to transmit it.

⇒ Different image compression & decompression techniques are used in this step.

⇒ The output of this step will be an Image.

⇒ ~~The output of this step will b~~

7. Morphological processing.

⇒ It is also called contour detection or boundary detection process.

⇒ It deals for extracting image components that are useful in the representation & description of the shape.

⇒ Output of this process will be attributes of an image.

8. Segmentation:

⇒ Segmentation procedures partition and image into different parts of an image.

⇒ Autonomous segmentation is very difficult task in the digital image processing.

⇒ Rugged segmentation procedure brings the process a long way towards the sucessful solution of the image processing problem that requires object to be accurate output.

⇒ The output of this step will be attribute of an image.

9. Representation & description

⇒ It almost always follow the output of a segmentation step which is usually raw pixel data, consisting either boundary of region or all points in the region itself.

⇒ The decision that must be made is whether the data should be represented as boundary or complete region.

⇒ Boundary is appropriate where focus is on external shape such as corners.

⇒ A complete region is appropriate when follows is on internal properties. Such as text.

⇒ Choosing representation with only part of solution for transforming raw data into a form suitable form complex processing.

10. Object recognition

⇒ Recognition is the process that assign a level to the object base on its description.
⇒ It is also called image classification or object matching process.
⇒ knowledge about the problem domain is coded into the image processing system in the form of knowledge data base.

Elements of Digital Image Processing system.

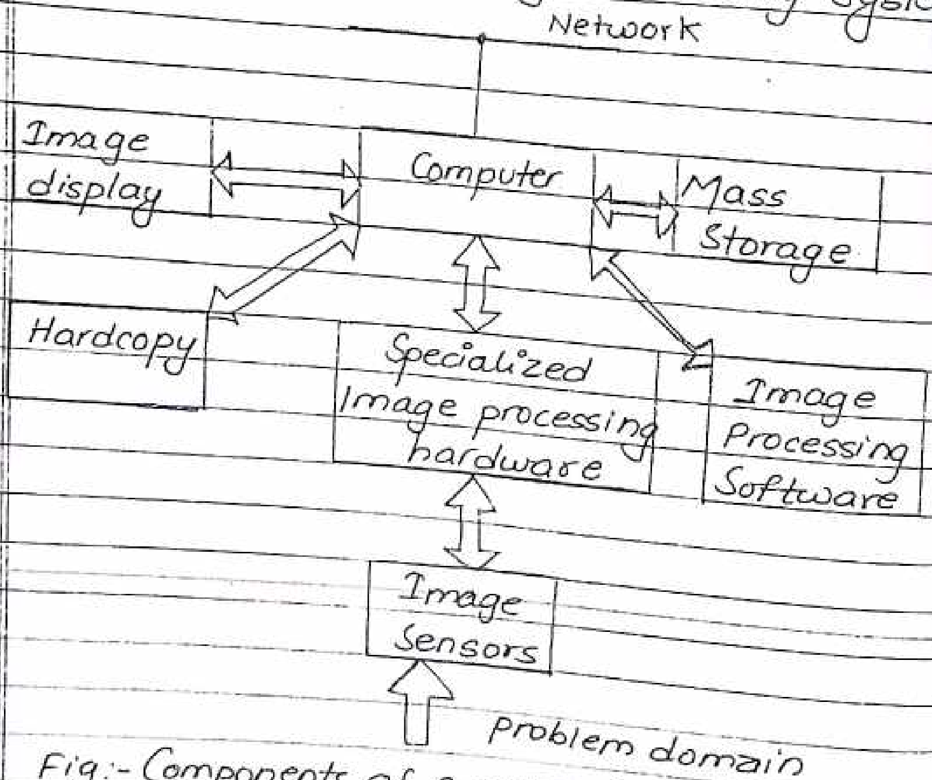


Fig:- Components of a general purpose Image processing system.

⇒ With reference to sensing, two elements are required to acquire digital images.

⇒ The first is a physical device that is sensitive to the energy radiated by the object we wish to image.

⇒ The second, called a digitizer is a device for converting the output of the physical sensing device into digital form.

⇒ For instance, in a digital video camera the sensors produce an electrical output proportional to light intensity.

⇒ The digitizer converts these outputs to digital data.

⇒ Specialized image processing hardware usually consist of the digitizer plus hardware that performs other primitive operation's such as ALU, which performs ALU operation in parallel on whole image.

⇒ Eg: of how an ALU is used is in average in image's as quickly as they are digitized for the purpose of noise reduction.

⇒ There are type of hardware sometimes is called frontend soft system and it's most distinguishing character is speed.

⇒ In other words, this unit perform's function that require fast data throughputs. (eg digitizing & averaging video images as 30 frame's) that typical main computer cannot handle.

- The computer is an image processing system is a general purpose computer and range.
- From a PC to a supercomputer. In dedicated applications sometimes specially design computers are used to achieve a required level of performance but our interest here is an general purpose image processing system.
- In this system almost any well equipped Pc type machine is suitable for offline image processing task.
- Software for image processing consists of specialized module that perform specific task.
- A well design package also include the capability for the user to write code that as a minimum utilizes the specialized modules.
- More sophisticated software packages allow the integration of those modules and general purpose software commands from atleast one computer language.
- Mass storage capability is a must in image processing applications.
- An image of size 1024 * 1024 pixels in which the intensity of each pixels is an 8 bit quantity, requires 1 Megabyte of storage space if the image is not compressed.

⇒ When dealing with 1000's or even millions of images providing adequate storage in an image processing system can be a challenge.
⇒ Digital storage for image processing application falls into 3 principle categories.

i) Short term storage for use during processing
ii) Online storage for relatively fast recall.
iii) ~~Arivol~~ Archival storage characterized by infrequent access.

⇒ Image displays in used today are mainly color TV, monitors preferably.
⇒ Flat screen monitors are driven by the output of image and graphics display cards that are an integer part of the computer system.
⇒ Seldom are these requirements for image display application that cannot be made by display card available commercially as part of the computer system.

⇒ In some cases, it is necessary to have stereo display and these are implemented in the form of gear containing two small display embedded in googles owned by the users.

⇒ Hardcopy devices for recorded images include Laser printers, film cameras, heat sensing device, inject units, digital units such as optical & CD-ROM disk.

⇒ Film provides the highest possible resolutions but paper is the obvious medium of choice for written materials.

⇒ For presentations, Images are display on film transparencies or in the digital medium if image projection equipment is used.

⇒ The later approach is gaining expectance as the standard for image presentations.

⇒ Networking is almost a default function in any computer system in used today.

⇒ Because of large amount of data inherent at image processing application, the key consideration in image transmission is bandwidth.

⇒ In dedicated networks this typically is not a problem but communication with remote sides via internet are not always an efficient.

⇒ Fortunately this situation is improving quickly as a result of optical fiber and other broadband technology.

Image Sampling & Quantization:

⇒ The output of must sensor is a continuous voltage wave form whose amplitude and spatial behaviour are related to the physical phenomenon being sensed.

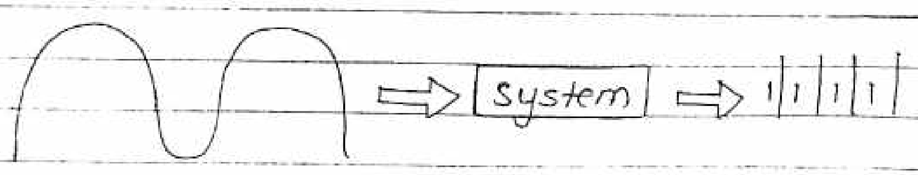
⇒ To create a digital image, we need to convert the continuous sensed data into digital form.

⇒ This involves two process:

1) Sampling
2) Quantization

⇒ Since, an Image is continuous not just its coordinate (x-axis) but also in its amplitude (y-axis).

⇒ So, the part that deals with digitizing co-ordinate is sampling and part that deals with digitizing amplitude is quantization.



Input Signal　　　　　　　　　　　　　　　　output signal

⇒ It is the process of taking measurement of the co-ordinate at regular space interval.

⇒ During the digitization process, sampling deals with digitizing (x, y) co-ordinates of an image which are finite & discrete.

Quantization:

⇒ It is the process of mapping the measured intensity to one of the finite no. of discrete level.

⇒ It is opposite to sampling.

⇒ It is done on y-axis.

⇒ Quantizing an image means dividing a signal into quanta (partitions)

⇒ On the x-axis of the signal, are the co-ordinate values and on y-axis, we have amplitudes.

⇒ So, digitizing the amplitude is known as quantization.

⇒ An image may be continuous w.r.to the x & y co-ordinates and also in amplitude.

⇒ To convert it to digital form we have to sample the function in both co-ordinates and in amplitude.

⇒ Digitizing the co-ordinate values is called sampling.

⇒ Digitizing the amplitude value is called Quantization.

⇒ In practise the method of sampling is determined by the sensor arrangement used to generate the image.

⇒ When an image is generated by single sensing element combine with mechanical motion, the o/p of sensor is quantized.

⇒ However, sampling is accomplished by selecting the no. of individual mechanical increment at which we activate the sensor to collect data.

⇒ Mechanical motion can be made very exact so in principal there is almost no. limit as to how find we can sample an image.

⇒ However, in practice limits on sampling accuracy are determined by other factors such as the quality of the optical component of the system.

Relationship between Neighbours, connectivity, distance measures between pixels

Neighbors of a pixel:

⇒ A pixel P at co-ordinates (x, y) has 4 horizontal and vertical neighbours whose co-ordinates are given by $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$.

⇒ This set of pixel called 4 neighbours of P, is denoted by $N4(P)$.

⇒ Each pixel is a unit distance from (x, y) & some of the neighbours of P lies outside the digital image if (x, y) is on the border of the image.

⇒ The 4 diagonal neighbours of P have co-ordinates $(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$

⇒ And are denoted by $ND(P)$.

⇒ These points together with the 4 neighbours are called 8 neighbours of P.

⇒ Denoted by $N8(P)$.

⇒ As before, some of the points ~~of~~ in $ND(P)$, $N8(P)$ fall outside the image if (x, y) is on the border of the image.

Connectivity:

⇒ Connectivity between pixels is a fundamental concept that ~~signifie~~ simplifies defination of numerous digital image concepts such as regions & boundaries.

⇒ To establish if two pixels are connected, it must be determined if they are neighbours and if their gray level satisfy a specific criterian of similarity.
(say if there gray levels are equal).

⇒ For instance, in a binary image with values 0 and 1, two pixels may be 4 neighbours but they are said to be connected only if they have same value.

⇒ Let 'V' be the set of gray level values used to define adjacency.

⇒ In a binary image;
$V = \{1\}$
if we are referring to adjacency of pixels with value 1.

⇒ In a grayscale image, the idea is the same, but set 'V' typically contains more elements.

⇒ For example:
In a adjacency of pixels with a range of possible gray level values 0 to 255, set 'V' could be any subset of these 256 values. We consider 3 type of adjacency.
(a) 4 adjacency

⇒ Two pixels 'p' & 'q' with values from 'V' are four adjacent if 'q' is in the set of N4(P).

(b) 8 adjacency

⇒ Two pixels 'p' & 'q' with values from 'V' are 8 adjacent if 'q' is the set of N8(P).

(c) m-adjacency / mixed adjacency:

⇒ Two pixels 'p' & 'q' with values from 'V' are m-adjacent if

(i) q is in N4(P)

(ii) q is in ND(P) &

the set has no pixels whose values from 'V'.

⇒ mixed adjacency is modification of 8 adjacency

⇒ It is introduced to eliminate ambiguities that often arise when 8 adjacency is used.

Distance measure

⇒ For pixel p, q, z with co-ordinate (x, y), (s, t) and (v, w) respectively, D is distance function or metric if

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

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

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

The Euclidean distance between P and q is defined as

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

=> For this distance measure, the pixels having a distance less than or equal to some value 'r' from (x,y) are the points contained in a disc of radius 'r' centered at (x,y).

=> The ~~D4~~ D4 distance also called cityblock distance between p & q are defined as

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

=> In this case, the pixel having a D4 distance from $(x,y) \leq$ some value 'r' form a diamond center at (x,y).

=> For example:

The pixel with D4 distance ≤ 2 from (x,y) (the center point) form the following contour some constant distance.

2
1 2
2

The pixels with $D4 = 1$ are the four neighbour are (x,y).

=> The D8 distance also called chess board distance between p and q is defined as $(x-s, y-t)$.

⇒ In this case, the pixel with D_8 distance from $(x, y) \leq$ some value 'r' form a square center at (x, y).

⇒ For example: The pixels with D_8 distance ≤ 2 from (x, y) (the center point) form the following contours of some constant distance.

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

⇒ The pixels with $D_8 = 1$ are the eight neighbours of (x, y).

⇒ Note that the D_4 and D_8 distance between p & q are independent of any paths that might exist betn the points because these distances involve only the coordinates of the points.

⇒ If we elect to consider m-adjacency, however the D_m distance between two points is defined as shortest m path between the points.

⇒ In this case, the distance betn two pixel will depend on values of pixels along the path, as well as the values of their neighbours.

⇒ For instance, consider following arrangement of pixel & assume that p, p_2 & p_4 have value 1 and p_1 & p_3 can have value of 0 or 1.

$p3 \quad p4$

$p1 \quad p2$

p

⇒ Suppose that we consider adjacency of pixel valued 1.

⇒ If p1 and p3 are 0, the length of the shortest m-path (the D_m distance between p & p4 is 2)

⇒ If p1 is 1; then p2 & p will no longer be adjacent and the length of the shortest m path be 3.
(the path goes through points p, p1, p2, p4).

⇒ Similar, comments apply if p3 = 1 & p1 is 0 in this case, the length of shortest m path is also 3.

⇒ Finally, if both p1 & p3 are 1, the length of the shortest m-path p & p4 is 4.

⇒ In the case, the path goes through the sequece of points P, p1, p2, p3, p4.

Elements of visual perception

1. Structure of human eye

=> The eye is nearly a sphere with an average diameter of approx. 20mm.

=> 3 membrane enclose the eye; the cornea & sclera outer cover; the choroid & the retina.

=> The cornea is top transparent tissue that covers the interior surface of the eye.

=> Continuous with cornea, the sclera is an opaque membrane that encloses the remainder of optic globe.

=> The choroid lies directly below the sclera.

=> The choroid coat is heavely pigmented & hence helps to reduce the amount of extraneous sight ~~interim~~ entering the eye & the back scatter within the optic globe.

=> At its Interior extreme, the choroid is divided into ciliary body and iris.

=> They contract or expand to control the amount of light that enters the eye.

=> The central opening of iris (pupil) varies in ~~dimension~~ diameter from approx. 2 to 8mm.

⇒ The front of iris contain the visible pigment of the eye whereas the back contains the black pigments.

⇒ The lens is made up of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body.

⇒ The lens is coloured by as a light yellow pigmentation that increases with edge.

⇒ The ~~enormous~~ inner most membrane of eye is the retina which lines the inside of the walls entire posterior portion.

⇒ Light from an object outside the eye is imaged on the retina.

⇒ Patterned vision is afforded by the distribution of discrete light receptors over the surface of retina.

⇒ There are 2 classes of receptors; cones that is highly sensitive to colour & rods that is sensitive to low level of illumination.

2. Image Formation in the eye:

⇒ In an ordinary photography camera a lens has fixed focal length and focusing at various distances is achieved by varying the distance between lens & imaging plane.

=> But in case of human eye, the distance betn the lens and the imaging region i.e. retina is fixed & focal length needed to achieve the proper focus is obtained by varying the shape of the lens.

=> This job is done by ciliary body. The distance betn the center of lens & retina along the visual axis is approximately 17mm.

=> The range of focal length is approx. 14mm to 17mm.

=> 17mm is the case when the eye is relaxed & focused at distances greater that about 3m.

3. Brightness adaption & discrimination:

=> The digital images are displayed as a discrete set of entities.

=> Due to this, the eyes ability to discriminate between diff. intensity levels is an important consideration in image processing results.

=> The range of light intensity levels to which human visual system can adopt is enormous on the order of 10^{10}.

⇒ The current sensitivity level of the visual system is called the brightness adaption level.

⇒ The ability of an eye to discriminate betn changes and the light density at any specific adaption level is discrimination.

⇒ The quantity $\frac{\Delta I_c}{I}$ where,

ΔI_c = Increment of illumination discriminable 50% of the time with background illumination I is called weber ratio.

⇒ A small value of $\frac{\Delta I_c}{I}$ means that a small percentage change in intensity is discriminable.

⇒ This represents good brightness discrimination

⇒ Conversly, a large value of $\frac{\Delta I_c}{I}$ means that a large percentage of change in intensity is required & Hence represents poor brightness discrimination.

Chapter 2

Image Enhancement in Spatial Domain:

Image Enhancement.

⇒ It is an application dependent process of emphasizing image features such as edge boundaries or constant improve visual appearance or content.

⇒ Image enhancement is an important topic in image processing because of its usefulness in almost processing application.

⇒ The principle objective of enhancement is to process an image so that the result is more suitable than the original image for specific application.

⇒ Enhancement techniques fall under two broad categories:

i) Spatial Domain Technique
- work on Image plane image
- direct manipulation of pixels in an image

ii) Frequency Domain technique
- Take the Fourier transform of the specified image.
- Modify Fourier transform coefficients of an image
- Take inverse Fourier Transform of the modified coefficients to obtain the enhanced image.

Some basic Gray level Transformation.

1. Point operation :-

Image enhacement on point operation 2D follows the following equation.

$g(x,y) = Tf(x,y)$

where T is the Transformation or image enhancement technique that is applied to the image $f(x,y)$ to obtain the processed image $g(x,y)$.

⇒ Incase of point operation, the neighbourhood size that is ~~consideratio~~ considered is of size (1 by 1) that means the operator 'T' now works on single pixel location & depending upon the intensity value of that location (x,y), it determines what will be the intensity ~~the the~~ in the corresponding location in the processed image 'g'.

⇒ This doesn't consider the pixel values at neighbouring locations.

⇒ Here, the value of each pixel is recalculated independently of all other pixels according to certain transformations.

⇒ This operation affect the appearance of an image when displayed and are often integrated into image digitizing and display applications.

2. Contrast stretching:

⇒ This process is appropriate for low contrast image or dark images.

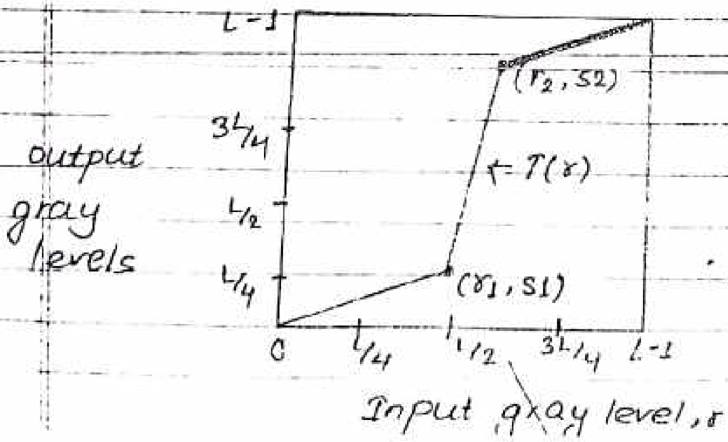
⇒ Low contrast image can result from poor illumination, lack of dynamic range in the imaging sensor or even wrong setting of a lens aparature during image aquition.

⇒ The idea behind contrast stretching is to increase the dynamic range of the gray level in the image being processed.

⇒ The contrast of a gray scale image can be adjusted by multiplying all pixel values by a constant gain, a.

$$g(x,y) = a f(x,y)$$

⇒ The overall contrast of the image is increased if $a > 1$. And decreased if $a < 1$.



⇒ Here, for contrast stretching the condition $r1 \leq r2$ and $s1 \leq s2$ must be satisfied so that the brighter part in the image ~~as brighter~~ remains as brighter and darker part remains darker.

⇒ The contrast stretched image is obtained using the transformation obtained from the eqn of the line having following points.

$(r1, s1) = (r_{min}, 0)$ and
$(r2, s2) = (r_{max}, L-1)$

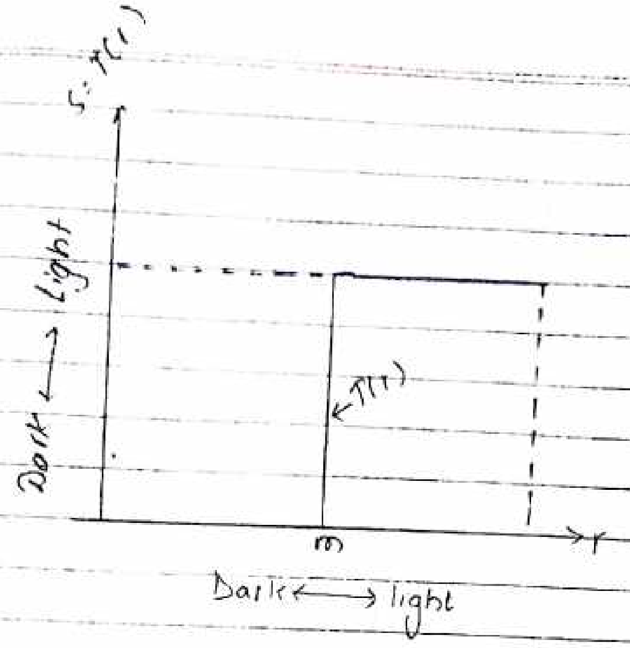
3. Thresholding:

⇒ It is the simplest method of segmentation where the resultant image becomes binary.

⇒ The simplest thresholding method replace ~~image p~~ each pixel in an image with a black pixel if the image intensity $I_{i,j}$ is less than some fixed constant T.

i.e. $I_{i,j} < T$. or a ~~br~~ white pixel if an image intensity is greater than the constant T.

⇒ Thresholding produces a binary image and for instance can be used to locate objects in an image w.r.to the plane background.



4. Digital negative:

⇒ A positive image is a normal image. A negative image is a total inversion in which light area appear dark and vice versa.

⇒ A negative color image is additionally color reversed, with ~~rate~~ a red area appears cyan, green area appear magenta & blue appears yellow.

⇒ A negative image is typically obtained by subtracting each pixel from the maximum pixel value.

⇒ Thus an 8 bit image the negative image can be achieved by reverse scaling of the gray levels according to the transformation;

$g(x,y) = 255 - f(x,y)$

where, 255 is maximum value.
$f(x,y)$ is the pixel value and $g(x,y)$ is negative value.

5. Intensity level slicing:

⇒ It can be used to segment certain gray level regions from the rest of the image.

⇒ This technique is particularly useful in the region of interest in an image is within a particular graylevel rage.

⇒ It is possible to utilize several implementation of intensity level slicing.

- Segment a particular range keeping original pixel value and set the rest to zero.

$$g(x,y) = \begin{cases} f(x,y) & f_{min} \leq f(x,y) \leq f_{max} \\ 0 & \text{otherwise} \end{cases}$$

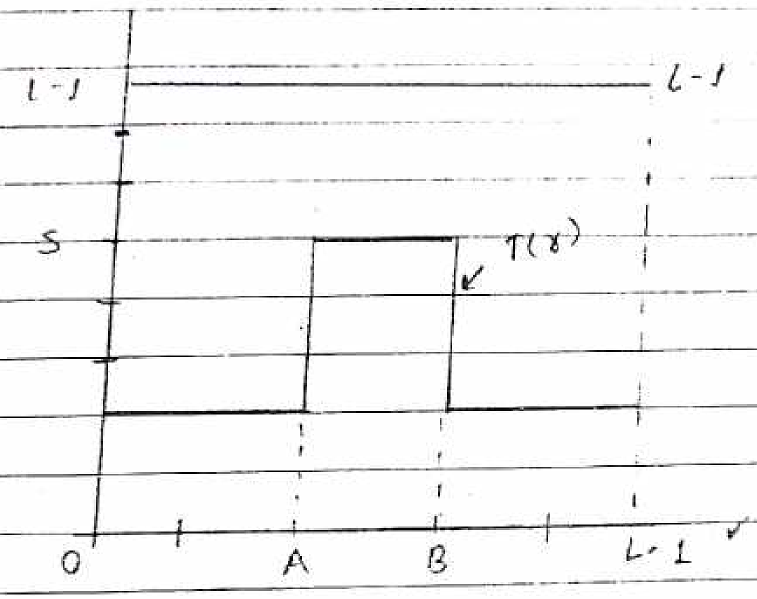
- Segment a particular range replacing pixel values by the highest value (255) and set the rest to zero.

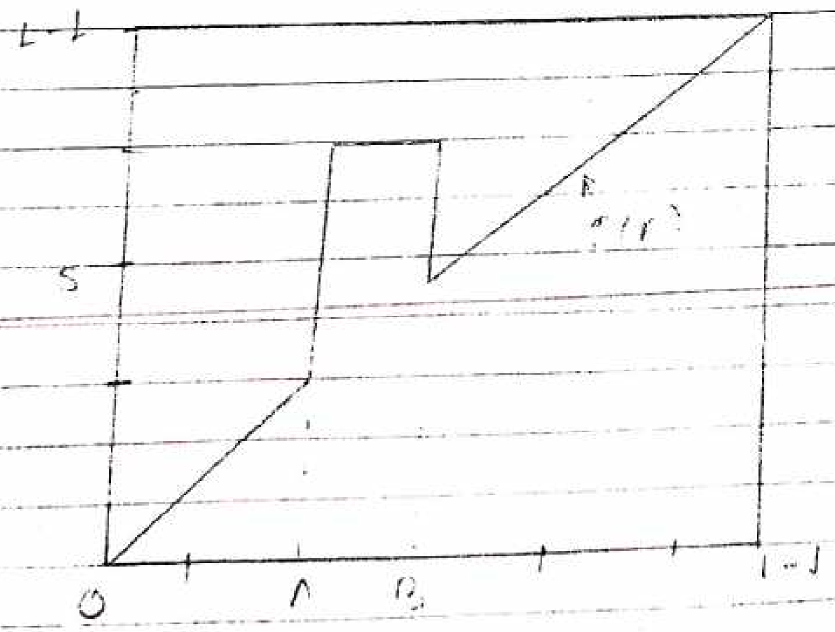
$$g(x,y) = \begin{cases} 255 & f_{min} \leq f(x,y) \leq f_{max} \\ 0 & \text{otherwise} \end{cases}$$

- Segment a particular range replacing pixel values by the highest value (255)

and keep the rest at the original pixel values.
(keep background):

$$g(x,y) = \begin{cases} 255 & f_{min} \le f(x,y) \le f_{max} \\ f(x,y) & \text{otherwise} \end{cases}$$





⇒ Intensity level slicing is sometime managed as a thresholding operation with multiple boundaries.

⇒ The lower threshold and upper threshold values we used to define the correspond-ing gray level range, and the option to keep pixel values within this range or replace then with particular value determines the operation.

⇒ In first fig, for the intensity ~~ter~~ range between A to B, the image will be enhanced and for other the pixels will be suppr-essed.

⇒ For the second figure, the transformation function shows the range between A & B, the image will be enhanced but outside this range, the original image will be retained.

Bit plane slicing:

⇒ Instead of high lightening gray level images, highlightening the contribution made to total image appearance by specific bits might be desire.

⇒ Suppose that each pixel in an image is represented by 8 bits. Imagine the Image is composed of 8, 1 bit plane ranging from bit 1-0 (LSB) to bit plane 7 (MSB)

⇒ In terms of 8 bits bytes, plane 0 contains all lowest order bits in the bytes comprising pixels in image & plane 7 contains all higher order bits.

⇒ Separating or digital image into its bit planes is useful for analysing the relative importants played by each bit of the image.

⇒ Implying, it determines the adequacy of the number of bits use to quantized each pixel useful for image compression.

⇒ Decomposing an image into its bit plane is useful for analysing the relative importance of each bit in the image.

⇒ A process that aids in determining the adequancy of no. of bits used to quantized the image.

⇒ This type of decomposing is useful for image compression in which fewer than all planes are used in a reconstructing an image.

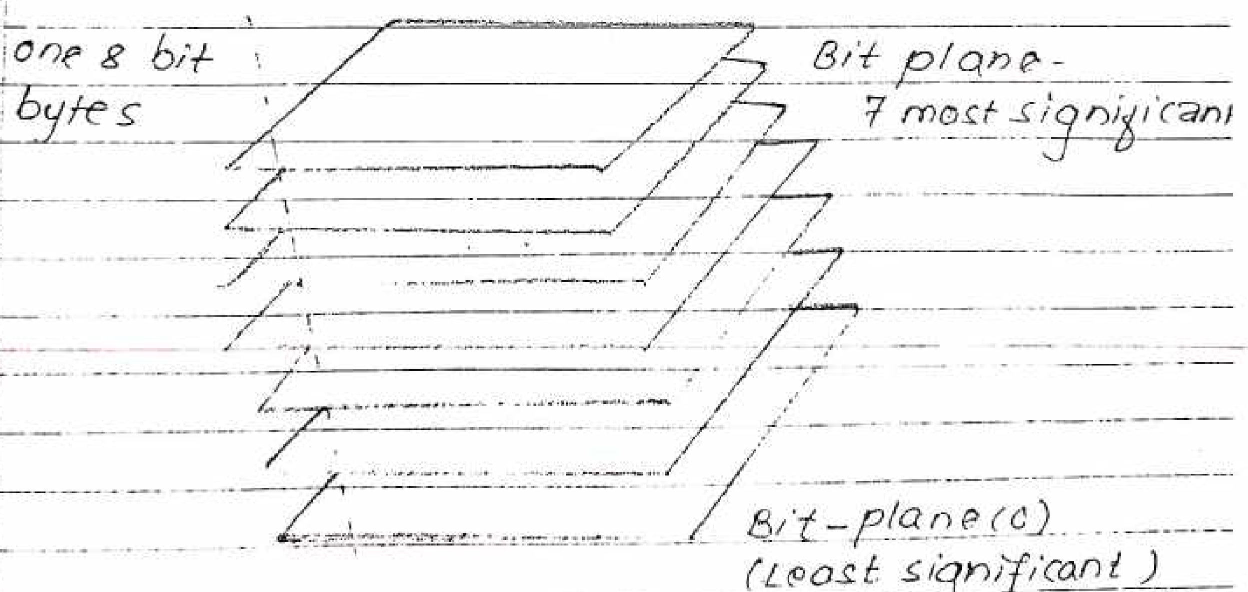


Fig:- Bit-plane representation of 8 bit.

Imp

Histogram processing & Equalization

Histogram:

⇒ Histogram of an image represents the frequency of occurance of gray level in the image.

⇒ It provides the global description of image.

⇒ These are basic for numerous spatil domain processing technique and can be used for image enhancement.

⇒ Histogram of digital image with graylevel in range (0, L-1) in a discrete fun' is

$\boxed{h(r_k) = n_k}$, $k = 0$ to $L-1$

where,

r_k = the k^{th} gray level

n_k = no. of pixels in the image having gray level r_k.

⇒ It is common practise to normalize the histogram by dividing each of its components by the total no. of pixel in the image, denoted by the product $M \times N$ where as usual M & N are row & columns dimensions of image.

⇒ Thus, a normalized histogram is given by

$$\boxed{P(r_k) = \frac{n_k}{M \times N}} \quad \text{for } k = 0, 1, 2, \ldots L-1.$$

where, $P(r_k)$: estimate of the probability of occurance of intensity level r_k in an image.

⇒ The sum of all component of normalized histogram is equal to 1.

⇒ It is massively useful in image processing specially in segmentation.

Application:

Histogram has many uses in image processing.

1) The first use is the analysis of image. We can predict about an image by just looking at its histogram. It is like looking an ex-ray of bone of a body.

2) Second use is for brightness purpose. The histogram has wide app. in image brightness. Not only in brightness, histogram are also used in adjusting contrast of an image.

3) Another importance of histogram is to equalize an image.

4) It has wide use in thresholding. This is mostly used in computer vision.

Histogram processing:

⇒ It refers to process of plotting and analyzing the histogram.

⇒ The horizontal axis of each histogram plot corresponds to intensity value r_k.

⇒ The vertical axis corresponds to the values of $h(r_k) = n_k$ or

$$P(r_k) = \frac{n_k}{M \times N} \quad (\text{for } k = 0, 1, 2 \ldots L-1 \text{ if}$$

the values are normalization.

⇒ In the dark image, the components of histogram are concentrated on the low (dark) side of the intensity scale.

⇒ The component of histogram of light image are biased towards the high side of ~~skin~~ scale.

⇒ An image with low contrast has a narrow histogram located typically toward the middle of intensity scale.

⇒ The component of histogram in high contrast image covers a wide range of intensity scale.

⇒ It is reasonable to conclude that an image whose pixels tend to occupy the entire range of possible intensity level and in addition ~~tand~~ to distributed uniformly will have appearance of high contrast and exhibit a large variety of gray tones. (Pg-142)

Histogram Equalization:

⇒ It is a process for increasing the contrast in an image by spreading the histogram out to be approximately uniformly distributed.

⇒ The graylevel of an image that has been subjected to histogram equalization are spread out and always reach white.

⇒ The increase of dynamic range produces an increase in contrast.

⇒ For image with low contrast, ~~once~~ histogram equalization has adverse effect of the increasing visual ~~grainness~~ graininess.

⇒ It requires construction of a transformation func sk.

$$S_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{M*N}$$

$$S_k = T(r_k) = \frac{L-1}{M*N} \sum_{j=0}^{k} n_j$$

where, r_k is k^{th} gray level

n_k is no. of pixel with that gray level

$M \times N$ is no. of pixel in the image &

k is $0, 1 \ldots L-1$

Here, $T(r_k)$ must satisfy two conditions

1. It must be single valued and monotonically increasing in the range $0 \le r \le 1$.
2. $T(r_k)$ must be in the range $0 \le t(r_k) \le 1$ or $0 \le r \le 1$.

Classmate
Date ______
Page ______

Histogram Specification (matching)

=> Histogram equalization has a disadvantage which is that it can ~~mat~~ generate only one type of image / O/P image.

=> With histogram specification it can specify the shape of histogram that we wish the output image to have.

=> It doesnot have uniform histogram. ~~at~~

=> When automatic enhancement is desired, this is a good approach because result from this technique are predictable and method is simple to implement.

=> In particular, it is useful sometime to be able to specify the shape of histogram that we wish the processed image have.

=> The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

Enhancement using
* Arithmetic & logic operations on image:

=> Arithmetic & logic operations on image used extensively in most image processing applications.

→ It may cover the entire image or a subset.
Arithmetic operation between p & q are defined as;

- Addition (p+q)
  used often for image averaging to reduce noise.
- Subtraction (p-q)
  used often for static background removal
- Multiplication (p*q) (or pq, p×q)
  used to connect gray level shading
- Division: (p÷q) (or p/q)
  as in multiplication.

* Arithmetic operation between pixels p and q are defined as:

- AND: p AND q (also p.q)
- OR: p OR q (also p+q)
- COMPLEMENT: NOT q (also q')
  - Form a functionally complete set.
  - Applicable to binary image.
  - Basic took in binary image processing

- used for.
  - Masking
  - Feature detecting
  - Shape analysis

⊙ Basics of Spatial Filters:

→ Spatial filtering is on of the principle tools used in image processing for a broad spectrum of application.

⇒ The name filter is borrowed from frequency domain processing where filtering refers to accepting or passing or rejecting certain frequency components.

⇒ Spatial refers to the co-ordinates also called spatial mask, kernel, template or windows.

⇒ Spatial filters consists of a neighbourhood typically a small rectangle of order MXN and a pre-defined operation that is performed on the image pixels incompassed by neighbourhood.

⇒ Filtering creates new pixels with co-ordinate equals to the co-ordinate of the centre of the neighbourhood & whose value is the result of filtering operations

⇒ A processed image is generated as the centre of the ~~first~~ filter visit each pixels in the input image.

⇒ Two type of spatial filters.
- Linear spatial filter
- Non-linear spatial filter

⇒ If operation performed on the image pixel is linear then the filter is called linear spatial filters else it is called non-linear spatial filters

Smoothening and sharpening spatial filters:

⇒ Smoothening filters are used for blurring and noise reduction. Blurring is used in pre-processing task such as removal of small details from an image to larger object extraction & bridging of small gaps in lines or curves.

⇒ The principle objective of sharpening is to highlight transition in intensities. It deals with the digital differentiation process. The strength of the response of the derivative operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied.

⇒ Thus, image differentiation enhance edge and other discontinuities & de-emphasized areas with slowly varying intensities.

Averaging:

⇒ The response of smoothing linear spatial filter is simply the average of the pixel contained in the neighbourhood of the filter mask.

⇒ These filters are also called averaging filters and low pass filters.

Date
Page

=> By replacing the value of every pixel in an image by the average of the intensity level in the neighbourhood defined by the filter mask.

=> This process results in an image with reduced sharp transition in intensities.

=> Because random noise typically consists of sharp transition in intensity level, the most obvious application of smoothing is noise reduction.

=> However, edges are also characterized by sharp intensities transitions, so averaging filters have the undesirable side effect that they blur edges.

=> Another application of this type of process include the smoothing of false contours that results from using an ~~instructi~~ insufficient intensity levels.

=> A measure use of averaging filters is in the reduction of irrelevant details in an image.

=> An example of 3x3 averaging filter is

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

=> It is one of the best known non-linear (order statistics) filters whose response is based on the ordering of the pixels contained in the image are encompass by the filter by the filter & then replacing the value of the centre pixel with the value determined by the ranking result.

=> Median filters are quite popular because for certain type of random noise they provide excellent noise reduction capability with considerably less blurring than linear smoothing filters of similar size.

=> Median filters are particularly effective in the presence of impulse noise (salt-peeper noise) because of its appearance as white & black dots super imposed on an image.

=> Here, first the values of the pixels in the ~~For all positive co~~ neighbourhood are sorted, their median is determined and that value is assigned to the corresponding pixels in filtered image.

=> Because neighbourhood pixel values are sorted & middle rank value is outputed, values that are significantly different from typical values within the neighbourhood will directly expelled.

=> Furthermore, pixels corrupted by noise will not affect surrounding pixels of the o/p as no more a weighted sum is computed as it is the case in linear low pass ~~filt~~ filtering.

For ex:-

The 3x3 median filter can be

64	65	76
56	68	61
34	55	68

→ 34 55 56 61 64 65 68 68 76

Spatial Low Pass.

⇒ A convolution using a kernel with all positive coefficient will resut in low pass filtering process.

⇒ For all positive coefficient of ~~waging~~ weighting of pixel value within the neighbourhood is carried out resulting in a smootheneding image.

⇒ To avoid an overall increase in image brightness, kernel coefficient can be normalized so that sum to unity.

⇒ Ex:- The following 3x3 uniform kernel shown in figure below are normalized so that kernel coefficient sum to unity.

⇒ These kernels result in unit waging of all pixel within the neighbourhood & result is normalized through division by the total no. of pixels within the neighbourhood.

$$\begin{bmatrix} 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \end{bmatrix}$$

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

Fig:- 3x3 kernel.

⇒ The convolution process is therefore equivalent to a ~~spat~~ spatial averaging.

⇒ Each pixel is replaced by the average or mean of the corresponding neighbourhood.

⇒ These kernels are therefore referred as mean filters.

High Pass Filtering

⇒ It is a type of sharpening spatial filters which is based on 1^{st} and 2^{nd} order derivatives to enhance the image.

⇒ The derivatives of a digital fun^c are defined in terms of differences.

⇒ There are various ways to define differences.

1) 1^{st} order derivative
2) 2^{nd} " "

- Both must be zero in areas of constant intensity. & must be non-zero at the onset of a ramp fun^c. & must be non-zero along ramp.

- A basic def^n of the 1^{st} order derivative of one dimensional fun^c $f(x)$ is the difference.

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

- We define 2^{nd} order derivative of $f(x)$ as the difference

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

⇒ 2^{nd} order derivative is used as highpass filter for image sharpening.

⇒ High pass filters are accomplished using kernel that contain negative as well as positive coefficients.

⇒ Therefore, the convolution result might be negative and either an output image representation that supports negative numbers is in order or constant bias can be added to output.

⇒ Usually a bias of half of the maximum pixel value is added so that actually a filter response of zero map onto the middle of the dynamic range, in which case negative filter o/p will be displayed darker & positive filter o/p will be brighter

⇒ High pass filter has too many uses edge detection & high frequency emphasis.

⇒ Edge detection is the process of locating parts of an image that display certain variation in gray level or color of pixels

- Edges are for instance generated by object boundaries. Hence the detection of edges is helpful in locating and possibly recognizing an image.

- Locating meaningful edges is at risky situation as noise and uninteresting features might also result in edge within the image.

- Commonly convolution kernels with the total coefficient sum of zero are used for the edge detection process.

- Due to the zero coefficient sum, the filter o/p will be zero (or comparatively small) if there is no or little variation in the gray level or the color of pixel.

- On the contrary, if there are certain change in the gray level, the filter O/P will have large amplitude either positive or negative revealing an edge in the output image.

- High frequency emphasis is the process of highlighting sharp features within an image

  - also call high boost filters, high frequency ~~empha~~ Emphasis filters are made of kernels that contain positive as well as negative coefficient but have a coefficient sum that is larger than zero.

  - This filters are commonly used for sharpen an image for improve visual appearn's.

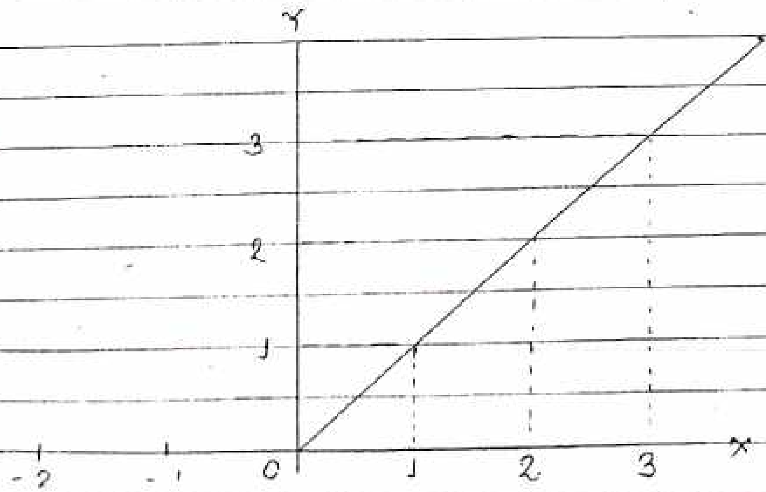


Fig:- Ramp func

$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0 & < 0 \end{cases}$$

Magnification by replication & Interpolation:

• **Zooming:**

⇒ It simply means enlarging a ~~pixe~~ picture in a sense that the detail in the image became more visible and clear.

• **Zooming by replication:**

⇒ It is also known as nearest neighbour interpolation.

⇒ In this method, we just replicate the neighbouring pixels.

⇒ Zooming is nothing but increase amount of sample or pixels.

⇒ This algorithm works on the same principle.

Algorithm.

Eg :- a 4×4 image.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

⇒ Repeat each pixel as shown in the example below.

1	1	2	2	3	3	4	4
5	5	6	6	7	7	8	8
9	9	10	10	11	11	12	12
13	13	14	14	15	15.	16	16

=> Repeat each row as shown in the example below.

1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
5	5	6	6	7	7	8	8
5	5	6	6	7	7	8	8
9	9	10	10	11	11	12	12
9	9	10	10	11	11	12	12
13	13	14	14	15	15	16	16
13	13	14	14	15	15	16	16

=> Here, 4x4 image is zoomed to 8x8 image.

=> This method can be repeated to get bigger size image.

=> Zooming by replication can be implemented on the computer by using replication mask (window).

=> Steps using mask.

* Steps using mask :

Interlace the original image with zeros. This is known as zero's interlacing. Adding zero to every other pixel of the first rows, we get 0's interlacing.

(I) 10 20 30 40

This is known as zero interlacing. Now, insert a row full of zero, we get:

1	0	2	0	3	0	4	0
0	0	0	0	0	0	0	0
5	0	6	0	7	0	8	0
0	0	0	0	0	0	0	0
9	0	10	0	11	0	12	0
0	0	0	0	0	0	0	0
13	0	14	0	15	0	16	0
0	0	0	0	0	0	0	0

II) On the above image, we run replication mask to get zoomed image.

1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
5	5	6	6	7	7	8	8
5	5	6	6	7	7	8	8
9	9	10	10	11	11	12	12
9	9	10	10	11	11	12	12
13	13	14	14	15	15	16	16
13	13	14	14	15	15	16	16

⇒ Zooming by linear interpolation:

Instead of replicating each pixels average of two adjacent pixels along rows is taken and placed between two pixels.

⇒ The same operation is then performed along the column.

⇒ Linear interpolation is performed on zero interlaced image.

(i)

1	1.5	2	2.5	3	3.5	4	2
0	0	0	0	0	0	0	0
5	5.5	6	6.5	7	7.5	8	4
0	0	0	0	0	0	0	0
9	9.5	10	10.5	11	11.5	12	6
0	0	0	0	0	0	0	0
13	13.5	14	14.5	15	15.5	16	8

(ii) Find the average value along row & place betn pixels.

1	1.5	2	2.5	3	3.5	4	2
3	3.5	4	4.5	5	5.5	6	3
5	5.5	6	6.5	7	7.5	8	4
7	7.5	8	8.5	9	9.5	10	5
9	9.5	10	10.5	11	11.5	12	6
11	11.5	12	12.5	13	13.5	14	7
13	13.5	14	14.5	15	15.5	16	8

(iii)

1	2	2	3	3	4	4	2
3	4	4	5	5	6	6	3
5	6	6	7	7	8	8	4
7	8	8	9	9	10	10	5
9	10	10	11	11	12	12	6
11	12	12	13	13	14	14	7
13	14	14	15	15	16	16	8

- Zooming is interpolation is more efficient then zooming by replication

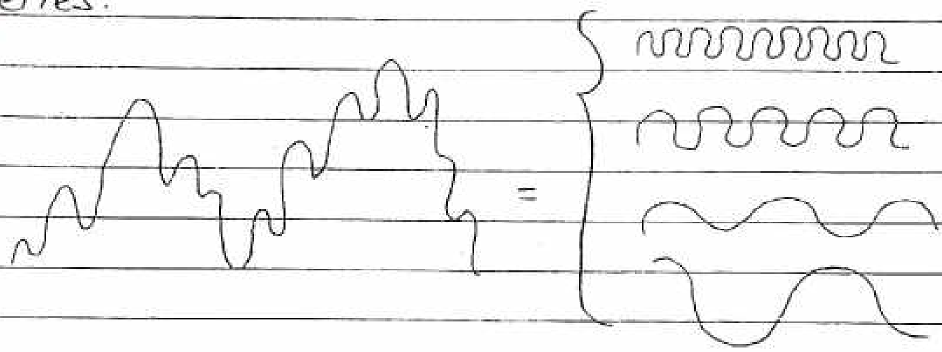
* Combining spatial enhancement method.
- Successful image enhancement is typically not achieved using a single operation.
- Rather we combine a range of techniques in order to achieve final result.

Chapter 3

Image Enhancement in frequency domain

3.1 Introduction to fourier transform & the frequency domain.

⇒ Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient - a fourier series.



⊙ **Frequency:**

The no. of times that the periodic function repeats the same sequence of values during a unit variation of the independent variables.

⊙ **Fourier Transform**

Let $f(x)$ be a continuous function of a real variable x.

The fourier transform of $f(x)$ denoted by $\mathcal{F}\{f(x)\}$ is given by:

$$\mathcal{F}\{f(x)\} = f(u) = \int_{-\infty}^{\infty} f(x) \exp[-j2\pi ux]\, dx$$

where, $j = \sqrt{-1}$.

Given function $F(u)$, $f(x)$ can be obtained by using the inverse fourier transform,

$$\mathcal{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) \exp[j 2\pi u x] du$$

- These two equations, called fourier transform pair, exist if $f(x)$ is continuous & integrable & $F(u)$ is integrable.

- These conditions are almost always satisfied in practice.

- We are concerned with function $f(x)$ which are real. However the fourier transform of a real fun^c is generally complex. So,

$$F(u) = R(u) + j I(u)$$

where,

$R(u)$ & $I(u)$ denote the real & imaginary component of $F(u)$ respectively.

- Expressed in exponential form, $F(u)$ is

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

where,

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

and

$$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

(4) The magnitude function $|F(U)|$ is called the fourier transform spectrum of $f(x)$ & $\phi(U)$ is the phase angle.

(5) The square of spectrum

$P(U) = |F(U)|^2 = R^2(U) + I^2(U)$ is commonly called the power spectrum (or the spectral density) of $f(x)$.

⇒ The variable U is often called frequency variable. This name arises from the expression of the exponent term $\exp[-j2\pi ux]$ in terms of sines and cosines.

⇒ From Euler's formula:

$$\exp[-j2\pi ux] = \cos(2\pi ux) - j\sin(2\pi ux)$$

⇒ $F(U)$ is composed of an infinite sum of sine and cosine terms.

⇒ Each value of U determines the frequency of its corresponding sine-cosine pair.

2-D Fourier Transform

⇒ The Fourier transform can be extended to 2 dimensions:

$$\mathcal{F}\{f(x,y)\} = F(U,V) = \iint_{-\infty}^{\infty} f(x,y) \exp[-j2\pi(Ux+Vy)]\,dx\,dy$$

and inverse transform.

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y) = \iint_{-\infty}^{\infty} F(u,v)\exp[j2\pi(Ux+Vy)]dxdy$$

⇒ The 2D Fourier spectrum is:

$$|F(U,V)| = \sqrt{R^2(U,V) + I^2(U,V)}$$

⇒ The phase angle is:

$$\phi(U,V) = \tan^{-1}\left[\frac{I(U,V)}{R(U,V)}\right]$$

⇒ The power spectrum is:

$$P(U,V) = |F(U,V)|^2 = R^2(U,V) + I^2(U,V)$$

Discrete Fourier Transform:

⇒ Suppose a continuous function, $f(x)$ is discreted into a sequence:

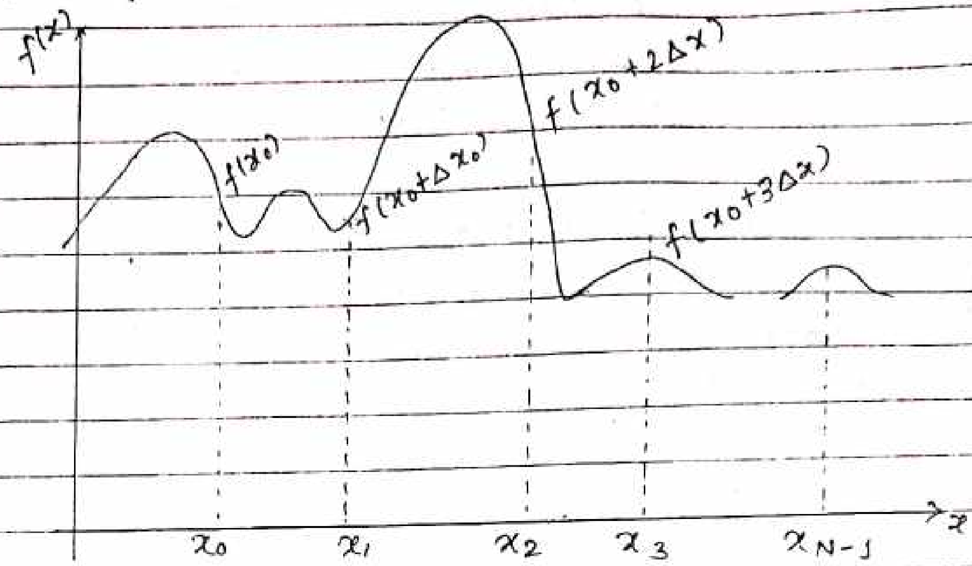
$\{f(x_0), f(x_0+\Delta x), f(x_0+2\Delta x), \dots f(x_0+[N-1]\Delta x)\}$
by taking N samples Δx units apart

⇒ Let x refer to either a continuous or discrete value by saying

$$f(x) = f(x_0 + x\Delta x)$$

⇒ where x assumes the discrete values $0, 1, \dots N-1$ and
$\{f(0), f(1) \dots f(N-1)\}$ denotes any N uniformly

spaced samples from a corresponding continuous function.



→ The discrete Fourier transform is given by:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j 2\pi u x / N]$$

for $u = 0, 1, \ldots N-1$

→ The discrete inverse fourier transform is given by

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[j 2\pi u x / N].$$

for $x = 0, 1 \ldots N-1$

→ The values of $u = 0, 1 \ldots N-1$ in the discrete case correspond to samples of the continuous transform at $0, \Delta u, 2\Delta u, \ldots (N-1)\Delta u$ and Δu and Δx are related by:

$$\Delta u = \frac{1}{(N\Delta x)}$$

2-D Discrete Fourier transform:

⇒ In the 2D case:

$$F(U,V) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-j2\pi\left(\frac{Ux}{M} + \frac{Vy}{M}\right)\right]$$

for $U = 0 \to M-1$ and $V = 0 \to N-1$

$$f(x,y) = \sum_{U=0}^{M-1} \sum_{V=0}^{N-1} F(U,V) \exp\left[j2\pi\left(\frac{Ux}{M} + \frac{Vy}{N}\right)\right]$$

for $x = 0 \to M-1$ and $y = 0 \to N-1$

⇒ The discrete function $f(x,y)$ represents samples of the continuous function at $f(x_0 + x\Delta x, y_0 + y\Delta y)$

$\Delta u = \frac{1}{(M\Delta x)}$ and $\Delta v = \frac{1}{(M\Delta y)}$

⇒ For the case when $N = M$ (such as in a square image)

$$F(U,V) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[\frac{-j2\pi(Ux + Vy)}{N}\right]$$

and

$$f(x,y) = \frac{1}{N} \sum_{U=0}^{N-1} \sum_{V=0}^{N-1} F(U,V) \exp\left[\frac{j2\pi(Ux + Vy)}{N}\right]$$

Note: each expression in this case has a $1/N$ term. The grouping of the constant multiplier terms in the fourier transform pair is arbitary.

Properties of 2-D Fourier Transform

⇒ The dynamic range of Fourier spectra is generally higher than can be displayed.

⇒ A common technique is to display the funᶜ:

$$D(U,V) = C \log[1 + |F(u,v)|]$$

where 'C' is a scaling factor and the logarithm function performs a "compression" of the data.

⇒ 'C' is usually choosen to scale the data into the range of the display device. [0-255] typically ([1-256] for 256 gray-level MATLAB image).

Smoothing Frequency Domain Filters: (LPF)

⇒ Smoothing is achieved in the frequency domain by dropping out the high frequency components.

⇒ The basic model for filtering is:

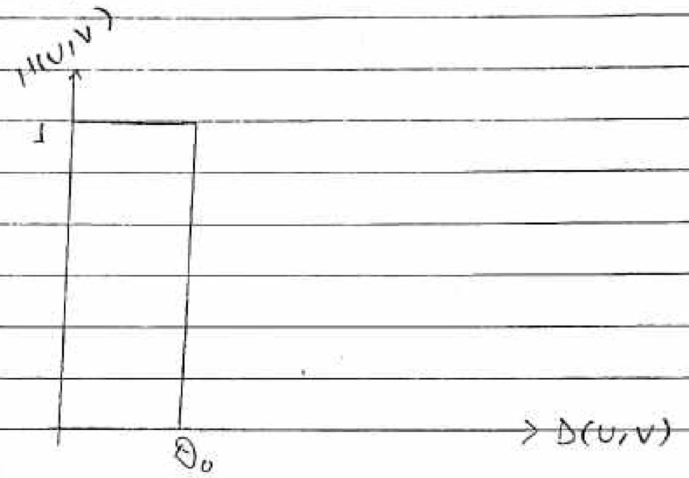
$$G(U,V) = H(U,V) F(U,V).$$

⇒ where $F(U,V)$ is the fourier Transform of the image being filtered and $H(U,V)$ is the filter transform function.

⇒ Low pass filters - only pass the low frequencies, drop the high ones.

Ideal Low pass Filter

⇒ Simple cut off all high frequency components that are a specified distance D_0 from the origin of transform.



Transfer function:

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

where $D(U,V)$ is given as:

$$D(U,V) = \left[(U - M/2)^2 + (V - N/2)^2\right]^{1/2}$$

Sharpening in the frequency Domain.

⇒ Edges and fine detail in images are associated with high frequency components.

⇒ High pass filters - only pass the high frequencies, drop the low ones.

⇒ High pass frequencies are precisely the reverse of low pass filters, so:

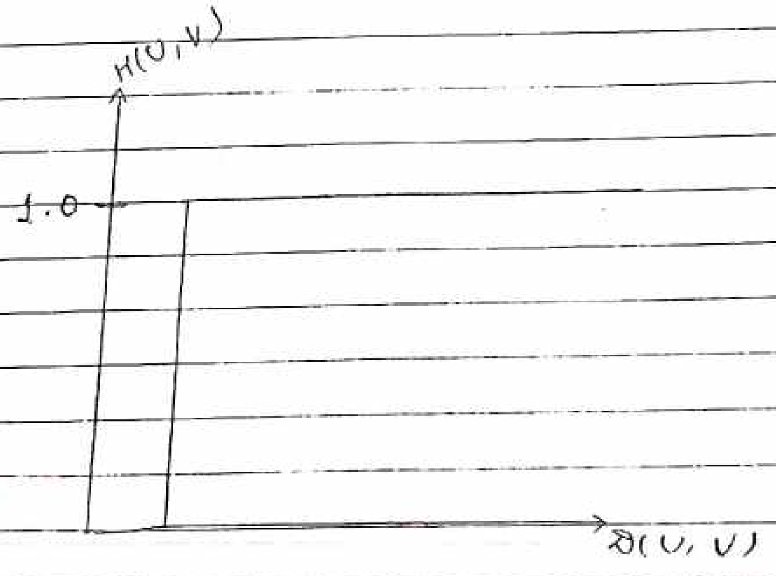
$$H_{hp}(U,V) = 1 - H_{lp}(U,V)$$

Ideal High pass filters:

⇒ The ideal high pass filters is given as

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

where, D_0 is the cut off distance as before



Fast Fourier Transform:

⇒ The reason that Fourier based techniques have become so popular is the development of the Fast Fourier Transform (FFT) algorithm.

⇒ Allows the Fourier Transform to be carried out in a reasonable amount of time.

⇒ Reduces the amount of time required to perform a Fourier tranform by a factor of 100-600 times.

contd...

Frequency Domain filtering & spatial Domain filtering

⇒ Similar jobs can be done in the spatial and frequency domains.

⇒ Filtering in the spatial domain can be easier to understand.

⇒ Filtering in the frequency domain can be much faster especially for large images.

contd..

The discrete Fourier transform is

$$F(U) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi x/M}$$

⇒ This takes $O(M^2)$ be compute

If we let:

$$W_M = e^{-j2\pi/M}$$

the discrete Fourier Transform can be written as:

$$F(U) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) W_M^{Ux}$$

If M is a multiple of 2, M = 2K for some positive number K. Substituting 2K for M gives:

$$F(U) = \frac{1}{2K} \sum_{x=0}^{2K-1} f(x) W_{2K}^{Ux}$$

Separting out the K even terms and K odd terms:

$$F(U) = \frac{1}{2} \left\{ \frac{1}{K} \sum_{x=0}^{K-1} f(2x) W_{2K}^{U2(x)} + \frac{1}{K} \sum_{x=0}^{K-1} f(2x+1) W_{2K}^{U(2x+1)} \right\}$$

Notice that:

$$W_{2K}^{2(U+1)} = W_{2K}^{2U} W_{2K}^{U} = W_K^U W_{2K}^U$$

So,

$$F(U) = \frac{1}{2} \left\{ \frac{1}{K} \sum_{x=0}^{K-1} f(2x) W_K^{Ux} + \frac{1}{K} \sum_{x=0}^{K-1} f(2x+1) W_K^{Ux} W_{2K}^U \right\}$$

$$F(U) = \frac{1}{2} \left\{ F_{even}(U) + F_{odd}(U) W_{2K}^U \right\}$$

Using this to get the first k terms ($u = 0, \ldots k-1$), then reuse there parts to get the last k terms ($u = k \ldots 2k-1$)

$$F(u+k) = \frac{1}{2}\{F_{even}(u) - F_{odd}(u)W_{2k}^{u}\}$$

(Only have to do $u = 0 \ldots M/2-1$, $x = 0 \ldots M/2-1$

and a little extra math)

To derive $F(u+k)$ use: $W_k^{u+k} = W_k^{u}$ and $W_{2k}^{u+k} = -W_{2k}^{u}$

Computational complexity:

Discrete Fourier Transform $O(N^2)$
Fast Fourier Transform $O(N \log N)$

Note:
Remember that, the fast Fourier Transform is just a Faster algorithm for computing the Discrete Fourier Transform - it is not any different in result

FFT for 2-D:

Remember, we can use seperability of the Fourier transform to break a 2-D transform into 2N, 1-D transforms

DFT — $O(N^4)$
DFT using seperability $O(N^3)$
FFT using separability $O(N^2 \log N)$

Discrete Cosine Transform

⇒ There are four definitions of the discrete Cosine transform, sometimes denoted DCT-I, DCT-II, DCT-III and DCT-IV.

⇒ The most commonly used DCT in image processing and compression is DCT-II for a square $N \times N$ image, the discrete transform matrix can be expressed as:

$$C_{NN}(k,l) = \begin{cases} \frac{1}{\sqrt{N}} & \text{for } l = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2k+1)l\pi}{2N}\right] & \text{all other } k, l \end{cases}$$

$$F = C_{NN} f C_{NN}^{T} \quad , \quad f = C_{NN}^{T} F C_{NN}$$

⇒ In 2-D case, the formula for normalized version of the discrete cosine transform (forward cosine transform DCT-II) may be written as:

$$F(U,V) = \frac{2C(U)\,C(V)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \cos\left(\frac{2m+1}{2N} U\pi\right) \cos\left(\frac{2n+1}{2N} V\pi\right)$$

$$U = 0, 1 \ldots N-1 \; ; \; V = 0, 1 \ldots N-1$$

where,

$$C(K) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k = 0 \\ 1 & \text{otherwise} \end{cases}$$

and inverse cosine transform is

$$f(m,n) = \frac{2}{N} \sum_{U=0}^{N-1} \sum_{V=0}^{N-1} c(u)c(v)F(U,V)\cos\left(\frac{2m+1}{2N} U\pi\right) \cos\left(\frac{2n+1}{2N} V\pi\right)$$

$m = 0, 1 \ldots N-1; \; n = 0, 1, \ldots N-1$

- Note that the DCT computation can be based on the Fourier transform all N coefficients of the DCT may be computed using a 2N-point FFT. DCT forms the basis of JPEG image Compression.

The Haar Transform:

⇒ The Haar Transform or the Haar wavelet transform is one of the group of related transforms knowns as discrete wavelet transform.

⇒ It is in particular can frequently be made very fast using matrix transformation.

⇒ It uses non-sinusoidal basic waveform.

⇒ So, it has great application related to digital siagnal processing.

⇒ The starting point for the definition of a

Haar transform is the Haar functions $h_K(z)$, which are defined in the closed interval $[0,1]$.

⇒ The order 'k' of the function is uniquely decomposed into two integers p, q.

$$k = 2^p + q - 1 , \quad k = 0, 1, \ldots, L-1 \;\&\; L = 2^n$$

where,

$$0 \le p \le n-1 , \quad 0 \le q \le 2^p \text{ for } p \ne 0$$

$$\text{and } q = 0 \quad \text{or for } p = 0$$

The Haar function are:

$$h_k(z) \equiv h_{pq}(z) = \frac{1}{\sqrt{L}} \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \le z < \frac{q-1/2}{2^p} \\ -2^{p/2} & \frac{q-1}{2^p} \le z < \frac{q/2}{2^p} \\ 0 & \text{otherwise in } [0,1] \end{cases}$$

$$h_0(z) \equiv h_{00}(z) = \frac{1}{\sqrt{L}} , \quad z \in [0,1]$$

⇒ The Haar transform matrix of order L consists of two resulting from the preceding function computed at the points.

$$z = m/L , \quad m = 0, 1, 2, \ldots, L-1 .$$

⇒ The N Haar functions can be scapled at $t = m/N$ to form an N by N matrix for discrete Haar transform. For Eg: when $N=2$. We have,

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$ last ko -1

when $N=4$

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

when $N=8$

$$H_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

ladder

Algorithm to generate the Haar Basis:

Step 1 : Determine the order of N of the Haar Basis.

Step 2 : Determine n where $n = \log_2 N$.

Step 3 : Determine p and q.

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

ii) If $p = 0$ then $q = 0$ or $q = 1$

iii) If $p \neq 0$, $1 \leq q \leq 2^p$

Step 4: Determine k

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

Step 5 : Determine z

$$z \to [0, 1] \Rightarrow \left[\frac{0}{N}, \frac{1}{N}, \ldots \frac{N-1}{N} \right]$$

Step 6 : If $k = 0$ then $H(z) = \frac{1}{\sqrt{N}}$

otherwise;

$$H_k(z) = H_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} +2^{p/2} & \frac{(q-1)}{2^p} \leq z < \frac{(q-1/2)}{2^p} \\ -2^{p/2} & \frac{(q-1/2)}{2^p} \leq z < q/2^p \\ 0 & \text{otherwise.} \end{cases}$$

For $N=2$

Step 1: $N=2$

Step 2: $n=\log_2 2=1$

Step 3: (i) Since $n=1$, the only value of P is 0

(ii) so q takes the values of 0 or 1.

Step 4: Determine the value of k using formula

$k=2^P+q-1$

p	q	k
0	0	0
0	1	1

for $p=0$ and $q=0$

for $p=0$ & $q=1$

Step 5: Determine the value of z,

$z \rightarrow [0,1] \Rightarrow \left\{\frac{0}{2}, \frac{1}{2}\right\}$

Step 6:

$$\boxed{P=0} \rightarrow \boxed{q=0},\ \boxed{q=1}$$

Case -1: If $k=0$ then $H(z)=\frac{1}{\sqrt{N}}=\frac{1}{\sqrt{2}}\ ;\in \forall z$

k\n	0	1
0	$1/\sqrt{2}$	$1/\sqrt{2}$
1	–	–
z	0	1

case-2:

for $k=1$; $p=0$; $q=1$

Condition:

i) $0 \le z < 1/2$

ii) $1/2 \le z < 1$

iii) otherwise

$$H_k(z) = H_{pq}(z) = \frac{1}{\sqrt{2}} \left\{ +2^{p/q} \frac{(q-1)}{2^p} \ldots\ldots \right.$$

For $z=0$ boundary condition (i) is satisfied

so, $H(z) = \frac{1}{\sqrt{2}} \cdot 2^{0/2} = \frac{1}{\sqrt{2}}$

For $z=1$, boundary condition (ii) is satisfied

so, $H(z) = \frac{-1}{\sqrt{2}} \cdot 2^{0/2} = -\frac{1}{\sqrt{2}}$

The haar basis for $N=2$ is given below

k/n	0	1
0	$1/\sqrt{2}$	$1/\sqrt{2}$
1	$1/\sqrt{2}$	$-1/\sqrt{2}$

Hadavard Transform:

=> The fourier tranform consist of a projection onto a set of orthogonal sinusodial wave form.

=> The fourier transform coeff. are called frequency component & waveforms are ordered by frequency.

⇒ It consist of a projection onto a set of square waves called Walsh Functions.

⇒ The Hadavard transform coeff. are called sequence components and the walsh functions are ordered by the number of their zero crossings.

⇒ The Walsh functions are real not complex and take only the values +1 & -1.

⇒ The Hadavard matrix H_{jj} is a symmetric JxJ matrix with element +1 and -1.

⇒ The Hadavard matrix of 2nd order is given by:

$$H_{22} = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

⇒ A Hadavard matrix of order 2J can be written as

$$H_{2j2j} = \begin{vmatrix} H_{jj} & H_{jj} \\ H_{jj} & H_{-jj} \end{vmatrix}$$

⇒ Hadavard matrices of orders other the powers of 2 exist, but they are not wisely used in image processing.

=) Inverse Hadavard matrices are easily computed as :

$$H_{jj}^{-1} = \frac{1}{j} H_{jj}$$

Pooznih Siha

Image Restoration

Chapter - 4

- Image Restoration

⇒ Image Restoration attempts to resture image that have been degraded.

⇒ It identify the degradation process & attempt to reverse it.

⇒ It is similar to image enhancement but more objective.

⇒ It attempts to recover an image that has been degraded by using a prior knowledge of the degradation phenomenon. Thus, this techniques are oriented towards modeling the degradation & appling the inverse process to recover the original image.

Image Degradation / Restoration process:

⇒ Degration / Restoration process is modelled as a degradation function that together with an addictive noise term, operates on an input image $f(x,y)$ to produce a degraded image $g(x,y)$

⇒ Given $g(x,y)$, some knowledge about the degradation function 'H' and some knowlege about the addictive noise term $\eta(x,y)$. The objective of restoration is to obtained and estimate $\hat{f}(x,y)$ of the original image. The more that is known about 'H' & η the closure the estimate can be.

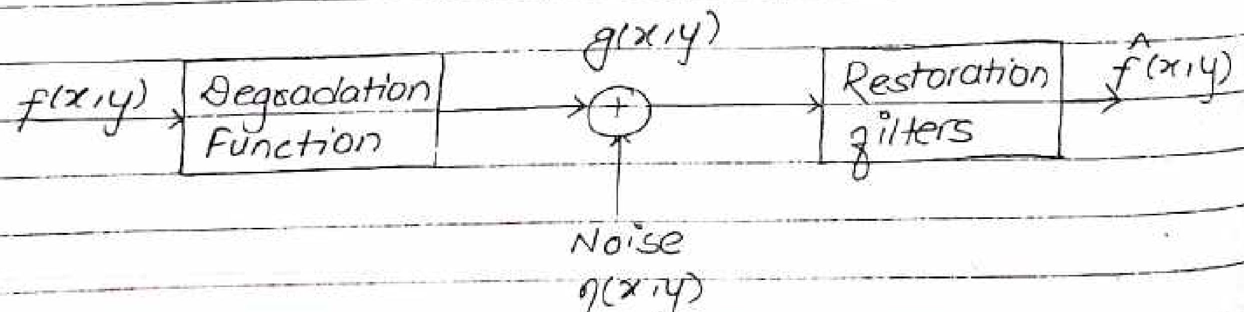
⇒ If H is a linear position invariant process then the degraded image can be describe as the convolution of 'h' & 'f' with an added noise term.

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

where,
$h(x,y)$ is the spatial representation of the degradation function.

⇒ The convolution of the spatial domain is analogous to multiplication in the frequency domain. So, in the frequency domain the representation is

$$G(U,V) = H(U,V)F(U,V) + N(U,V)$$



Noise Models Restoration in the presence of noise only spatial filtering:

⇒ Image Noise is random (not present in the object image) variation of brightness or color information in images & usually an aspect of electronic noise.

⇒ The principle sources of noise in digital image arise during image acquisition or transmission

⇒ The performance of imaging sensors is affected by variety of factors such as environmental conditions during image acquistion & by the quality of sensing element themselves.

For instance in acquiring image with th CCD cameras, light levels & sensor temperature are measure factors affecting the amount of noise in the resulting image.

⇒ Images are corrupted during transmission principally due to interference in the channel for transmission.

For eg :- An image transmitted using a wireless network might be corrupted as a result of Lightening or other atmospheric disturbance.

Spatial and frequency properties of noise.

⇒ Noises independent of spatial co-ordinate expect spatially periodic noise.

⇒ It is uncorrelated with respect to image itself.

⇒ Frequency properties refers to the frequency contain of noise to the fourier sense.

For eg :- When the fourier spectrum of the noise is constant. The noise usually is called white noise. This terminology is a carry over from the physical properties of white light which contains all frequency in the visible spectrum in equal proportions.

Noise models:

⇒ Noise component of the model are random variables characterized by probability density functions (PDF)

⇒ The most common PDFs in image processing applications are:

1) Gaussian noise
2) Rayleigh noise
3) Erlang noise (Gamma)
4) Exponential noise
5) Uniform noise
6) Impulse (Salt & pepper) noise

1) Gaussian Noise:

- It is also called an electronic noise because it arises in amplifiers or detectors.

- Caused by natural sources such as thermal vibration of atoms & discrete nature of radiation of warm ~~objectives~~ objects

- It disturbs the grey level values in digital image.

- The PDF of Gaussian random variable z is given by

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

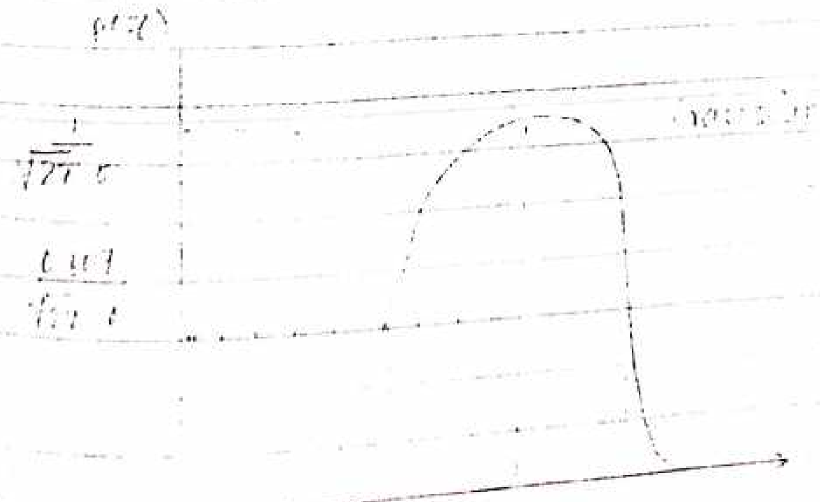
where.

z represents intensity

$\bar{z}$ represents the mean (average) value of z.

σ is standard deviation

σ^2 is variance of z.



⇒ 70% of the value of z will be within one S.D and 95% of the values of z will be within two S.D.

2) Rayleigh Noise

⇒ The PDF of the Rayleigh noise is given by:

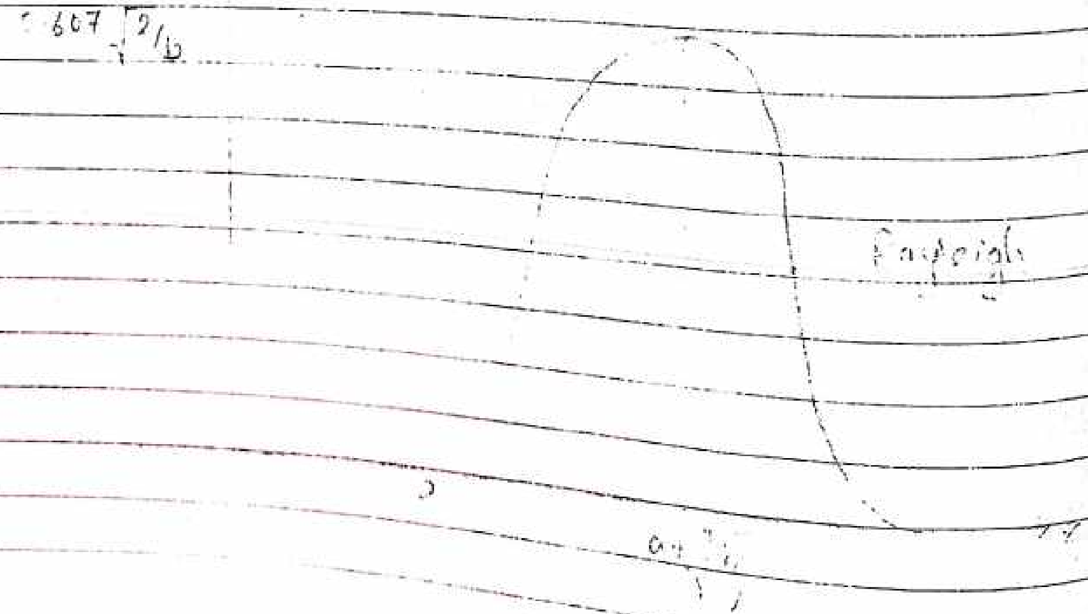
$$P(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & z < a \end{cases}$$

where, z = intensity

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

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

⇒ This model can be quite useful for approximating skewed histograms



3. Erlang Noise (Gamma):

- Generally seen in the laser based image.
- It obeys the gamma distribution
- The PDF is given as:

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

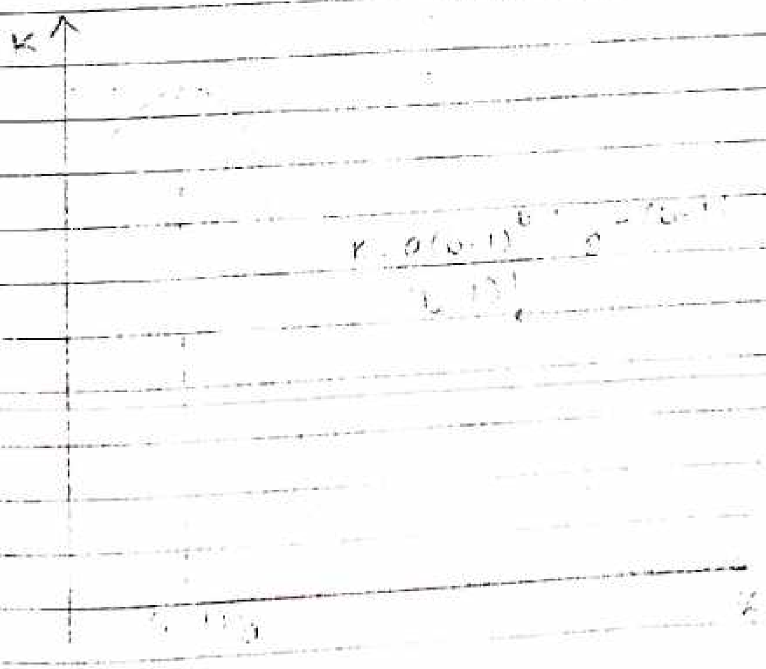
where,

z : intensity

$\bar{z} = b/a$

$\sigma^2 = b/a^2$

$a > 0$, b is positive integer



4) Exponential Noise:

⇒ The PDF is given by:

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

where,

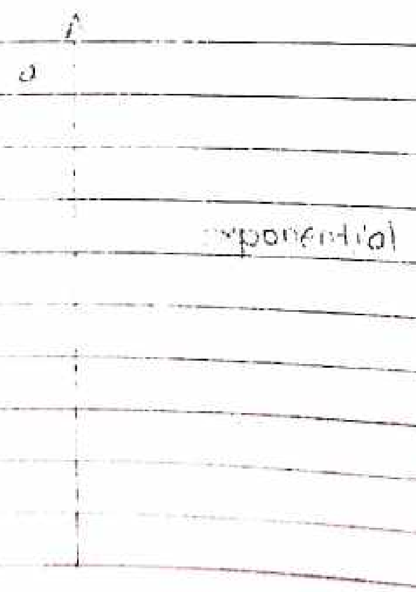
z = intensity

$\bar{z} = 1/a$

$\sigma^2 = 1/a^2$

$a > 0$

⇒ This PDF is a spatial case of the Erlang PDF with $b = 1$.



5) Uniform Noise:-

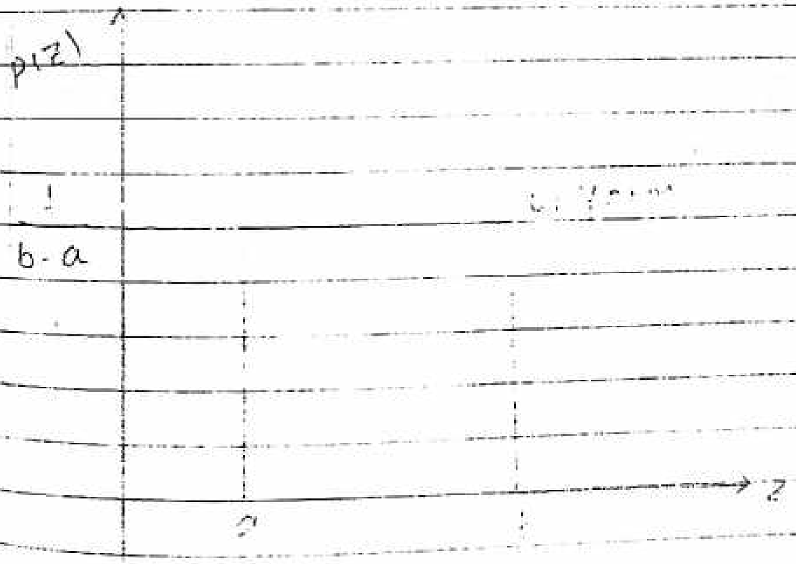
→ The PDF is given by:

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

where,

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

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



6) Impulse (salt & Pepper) noise:-

→ PDF is given by

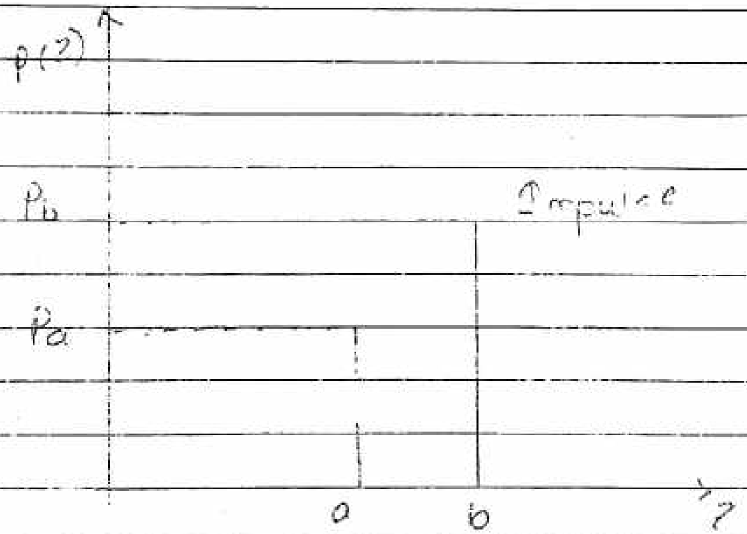
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

• If b>a then any pixel with intensity b will appear as a light dot in the range.

• pixels with intensity 'a' will appear as a dark dot.

⇒ An image containing impulse noise will have dark pixels in bright regions and bright pixels in dark regions.

⇒ It is caused by analog on digital conversion error bit errors in transmission, etc.



Periodic Noise Reduction by Frequency Domain Filtering:

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

⇒ This type of noise can be reduced significantly via frequency domain filtering

⇒ The basic idea is that periodic noise appears as concentrated burst of energy in the fourier transform at locations corresponding to the frequencies of the periodic interference.

⇒ The approach is to use a selective filter to isolate the noise.

⇒ There are 3 types of selective filters used for periodic noise reduction:

1. Band reject filters

⇒ It is used in applications where the general location of the noise components in the frequency domain is approximately known.

⇒ Removing periodic noise from an image involves removing a particular range of frequencies from that image.

reject filter with transfer function. $H_{BR}(U,V)$ by

$$H_{BP}(U,V) = 1 - H_{BR}(U,V)$$

=> It is not a common practise to directly implement band pass filtering or image since it removes too much of image details.

=> It is quite useful in isolating the effects on an image caused by selected frequency bands.

=> This filtering helps to isolate the noise patterns which is independent of the image contained.

3) Notch Filters:-

=> It rejects or passes frequencies in predefined neighbourhood about a center frequency.

=> This approach reduces the noise in the image without introducing the applicable blurring in the image.

=> Due to the symmetry of the fourier transform, notch filters much appear in symmetric pairs about the origin in order to obtain meaningful results.

⇒ Band reject filters can be used for this purpose.

⇒ An idle band reject filter is given as follows:

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - W/2 \\ 0 & \text{if } D_0 - W/2 \le D(u,v) \le D_0 + W/2 \\ 1 & \text{if } D(u,v) > D_0 + W/2 \end{cases}$$

Types:

a) Butterworth:

⇒ A butterworth band reject filter of order 'n' is given by the expression

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

b) Gaussian:

⇒ A Gaussian band reject filters is given by:-

$$H(u,v) = 1 - e^{-1/2\left[\frac{D^2(u,v) - D_0^2}{D(u,v)W}\right]^2}$$

2. Band pass filter:

⇒ A band pass filters performs the opposite operation of band reject fiters

⇒ The transfer function $H_{BP}(u,v)$ of a Band pass filter is obtained from a corresponding band

Image Compression

poczahh Stha

Chapter 5

Cosmos College

⇒ Image Compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unexceptable level.

⇒ The objective of image compression is to ~~irreliv~~ irrelevant and redundancy of the image data in order to be able to store for transmit data in an efficient form.

⇒ The reduction in file size allows more image to be stored in a given amount of disc or memory space.

⇒ It also reduces the time required for images to be sent over the internet or downloaded from web pages.

⇒ Why image compression is required?

- Digital images are very large in size and hence occupy larger storage space.

- Due to their larger size they take larger bandwidth and more time for upload or download through the internet. This makes it inconvienent for storage as well as file share.

- To Combat with this problem, the images are compressed in size with special Technique. This compression not only helps in saving storage space but also enables easy sharing of file.

Image Compression app. reduce the size of an image file without causing major degradation to the quality of image.

Eg:-
One 90 minutes color movie, each second plays 24 frames. When digitize it, each frame has 512 x 512 pixels, each pixels has three components R, G and B each one occupies 8 bits respectively. The total byte number is:

$$90 \times 60 \times 24 \times 3 \times 512 \times 512 = 97,200 \text{ MB}.$$

Eg:

A CD may save 600 megabytes of data, the movie needs 160 CDs to save. So number of CDs to store the megabytes data can be reduced by using various compression techniques.

Applications:

1. Progressive transmission of image (internet)

2. Video coding (HDTV, Teleconferency)

3. Digital libraries and image databases.

4. Medical imaging.

5. Satellite images.

Coding Redundancy:

⇒ It is used to reduce the amount of data representing same information.

⇒ This type of coding is always reversible and usually implementing using Look up table (LUTs)

⇒ Eg of image coding scheme that explore coding redundancy are Huffman codes & arithmetic coding technique

⇒ In this we utilize formulation to show how the gray level histogram of an image also can provide a great deal of insight into the construction of codes to reduce the amount of data used to represent it.

⇒ Let us assume that a discrete random variable r_k in the interval [0,1] represent the graylevel of an image that each r_k occurs with probability

$P_r(r_k) = \frac{n_k}{n}$ where, $k = 0, 1, 2 \dots L-1$

where L is the no. of graylevels,
n_k is the no. of times that the k^{th} graylevel appears in the image
n is the total no. of pixels in the image.

⇒ if the no. of bits used to represent each value of r_k is $l(r_k)$ then the average no. of bits required to represent each pixel is

$$L_{avg} = \sum_{K=0}^{L-1} l(r_K) P_r(r_K)$$

i.e. the average length of the code words assign to the various graylevel values is found by summing the product of no. of bits used to represent each graylevel & the probability that graylevel occurs. Thus, the total no. of bits required to code MXN image is MXN Lavg.

Huffman coding:

⇒ also known as Huffman encoding, an algorithm for the lossless compression of files based on the frequency of occurance of a symbol in the file that is being compressed.

⇒ The Huffman algorithm is based on statistical coding, which means that the probability of symbol has a direct bearing on the length of its representation.

⇒ The more probable of the occurance of the symbol is the shorter will be its bit size representation.

⇒ In any file, certain characters are used more than others

⇒ Using binary representation, no. of bits required to represent each character depends upon the no. of characters that have to be represented.

⇒ Using 1 bit we can represent two characters. i.e. 0 represent the first character & 1 represent the 2nd character.

⇒ Using 2 bit we can represent 4 characters & so on.

Example:

Consider a 6 symbol source

	a_1	a_2	a_3	a_4	a_5	a_6
$P(a_i)$:	0.1	0.4	0.06	0.1	0.04	0.3

Step 1: Source reduction

Original Source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Step 2: Code assignment procedure

Sym	Prob.	code	1	2	3	4	bits
a2	0.4	0	0.4	0.4	0.4	0.6	1
a6	0.3	10	0.3	0.3	0.3	0.4	0
a1	0.1	111	0.1	0.2	0.3		
a4	0.1	1100	0.1	0.1			
a3	0.06	11010	0.1				
a5	0.04	11011					

The code is instantaneous uniquely decodable without refering sucessing symbols.

Average length:

$0.4 \times 1 + 0.3 * 2 + 0.1 \times 3 + 0.1 * 4 + (0.06 + 0.04) * 5$

$= 2.2$ bits (symbol)

Interpixel Redundancy:

⇒ It includes:

i) Spatial Redundancy

ii) Geometric "

iii) Interframe "

Run Length Coding:

→ A simple data compression scheme in which sequences of the same item are replaced by one such item and a count. For example the text BBBBB stored as a single B with count of 5.

⇒ Run length encoding (RLE) is very simple form of data compression in which runs of data (i.e. sequences in which the same data value occurs in many consequtive data elements) are stored as a single data value and count rather than as the original run.

⇒ This is most useful on data that contains many such runs consider for example simple graphic images such as icons, linedrawings & animations. It is not useful with files that doesn't have many runs as it could greatly increase the file size.

⇒ Run length encoding performs lossless data compression.

⇒ Run length encoding is used in fax machines. It is relatively efficient because must faxed documents are generally white space with occasional interruptions of black.

⇒ For example: Consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space & many short runs of black pixels within the text.

Let us take a hypothetical single scanline, with B representing a black pixel & W representing white.

WWWWWBWWWWWWBBBWWWWWWBB

if we apply the run length encoding data Compression algorithm to the above hypothetical scan line. We get the following:

5W1B5W3B6W2B

~~Psychologic~~ Psychovisual Redundancy:

⇒ It performs lossy data Compression.

⇒ The brightness of a region as percieved by the eye depends on factors other than simply the light reflected by the region.

⇒ For example:
Intensity variations can be percived in an area of constant intensity. Such phenomena result from the fact that the eye doesnot respond with equal sensitivity to all visual information certain information simply has less relative importance than other information in normal visual processing.

This information is said to be psycho-visually redundant. It can be eliminated without significantly imporing the quality of image perception.

⇒ The psycho visual redundancy exist should not come as a surprise because human perception of the information in an image normally doesnot involved quantitative analysis of every pixel value in the image. In general an observer searches for distinguishing features such as edges or textual regions & mentally combines them into recognizable groupings.

⇒ The brain then co-relates these groupings with s prior knowledge in order to complete the image interpertation process.

⇒ Unlike cooling an interpixel redundancy, psycho-visual redundancy is associated with real or quantifiable information.

⇒ Its elimination is possible only because the information itself is not essential for normal visual processing since the elimination of psychovisually redundancy data results in a loss of quantitative information.

⇒ It is commonly referred to as quantization.

⇒ This terminology is consistent with normal use of the word which generally means the mapping of a broad range of input value to a limited no. of output values as it is an irreversible operation, quatization result in lossy data Compression.

Image Compression Models:

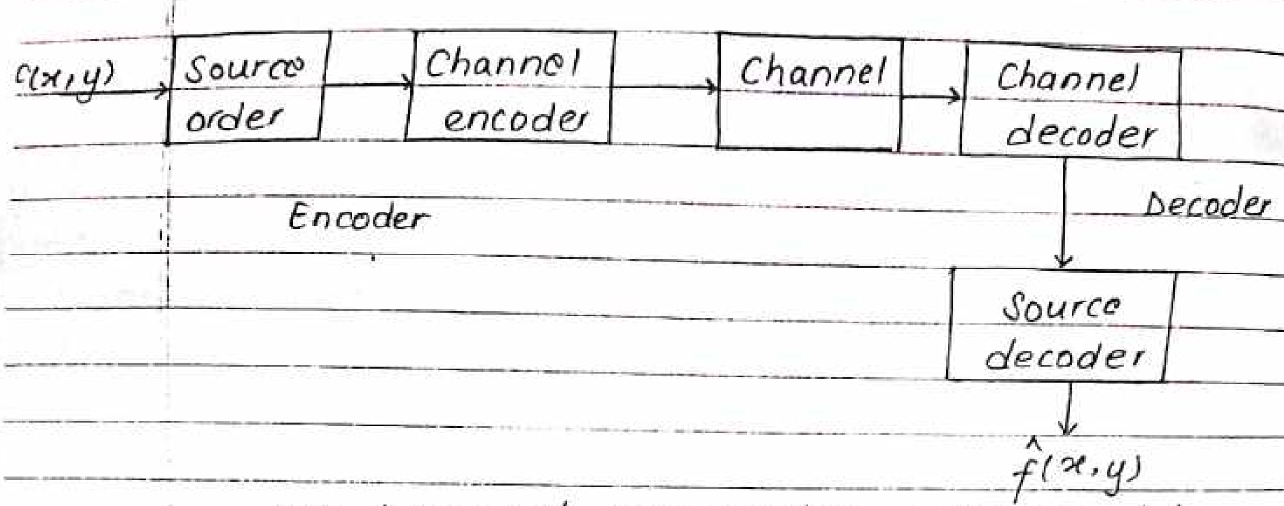


Fig:- A general compression system model.

=> A general system model for compression and decompression is shown above.

=> An input image $f(x,y)$ fed into the encoder which creates a set of symbols from the input data after transmission over the channel the encoded representation is fed to the decoder where a reconstructed output image $\hat{f}(x,y)$ is generated.

=> In general $\hat{f}(x,y)$ may or maynot be an exact replica of $f(x,y)$. If it is the system is errorfree or information preserving if not some level of distortion is present in the reconstructed image.

=> If the transmission of storing channel is error free, the channel encoder and decoder is omitted otherwise extra data bits can be

added to be able to detect.
(For eg: parity, cyclic redundancy checks) or correct (error correcting code for memory) error often used in special hardware.

Encoder:

⇒ A general model for a source encoder is:

$f(x,y) \rightarrow$ [Mapper] $\rightarrow$ [Quantizer] $\rightarrow$ [Symbol encoder] $\rightarrow$ channel

Fig:- Source encoder

⇒ The mapper transforms the data to a format suitable for reducing the interpixel redundancies.

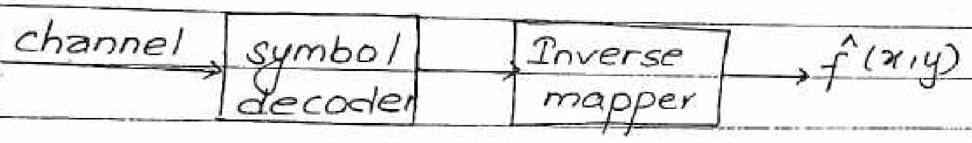
This step is generally irreverssible and can reduce the amount of data; used for run length encoding but not in transformations to the sources or Discrete cosine domains

⇒ The 'quantizer' reduced the precision of the output of the mapper according to the determined reliability criteria This's especially reduce psycho-visual redundancies and is irreversible. It is therefore only used for "Lossy" compression

⇒ The "symbol encoder" makes a static or variable length of code to represent quantizers output. It reduces the coding redundancy & is reversible.

Decoder

The general model belonging to the source decoder is:



source decoder.

⇒ Symbol decoder decodes the incoming data / information on the channel.

⇒ Inverse mapper transforms the transmitted compressed data to a format suitable compressed lossy image.

Imp # Lossless & Lossy compression:

- Lossless Compression:

⇒ It is used when the data has to be uncompressed exactly as it was before compression.

⇒ No information is loss during processing.

⇒ Text files are stored using loseless technique because loosing the single character can in the worst case, make the text dangerously misleading.

⇒ It is based on the knowledge of coloured image & human perception.

⇒ Image can be restored without any loss of information.

⇒ For eg:- Medical images, GIS, etc.

- Lossy Compression:

⇒ It reduces bits by idenfying marginally important information & removing it.

⇒ It works on the assumption that data doesnot need to be store perfectly.

⇒ Much information can still be or expectable quality in image, videos and audio data when compressed.

⇒ Perfect recovery isnot possible but it does provide the larger compression of data.

⇒ For eg:- TV signals, Tele-conferencing, etc.

Predictive coding:

⇒ future value will be predicted on the basis of past pixel value.

⇒ Predictive coding simply transmit the difference.

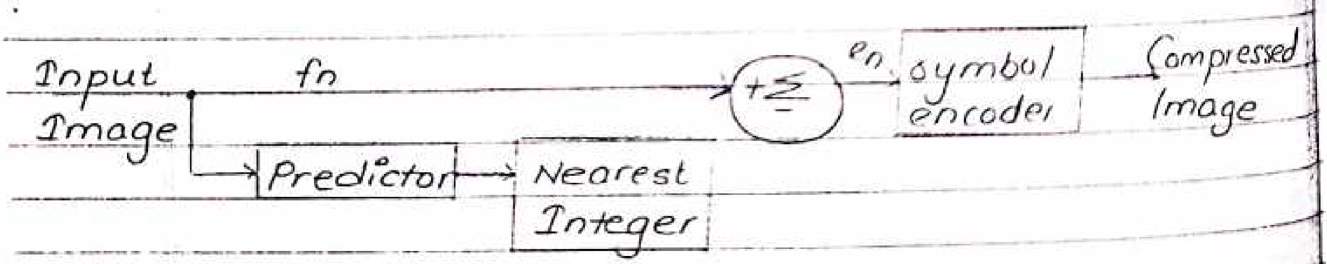
⇒ Predict the next sample as been equal to the current sample.

⇒ Instead of sending current sample, send the error in the previous assumption.

Types of predictive coding

1) Lossless predictive coding
2) Lossy predictive coding

1) Lossless predictive coding:



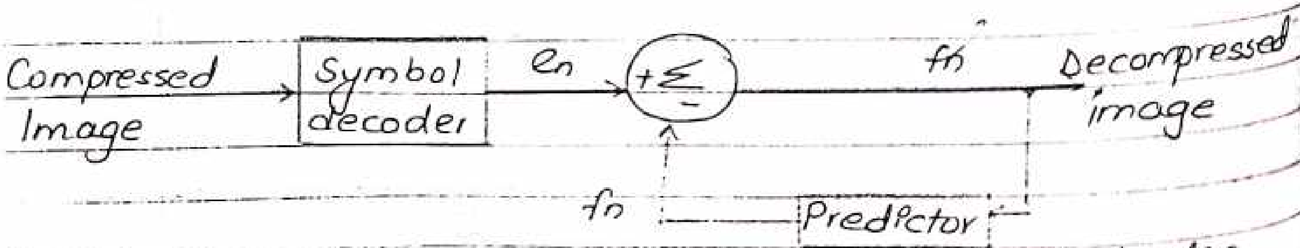


Fig-A lossless predict. model encoder & decoder

⇒ It is based on eliminating the interpixel redundancies of closely spaced pixels by extracting & coding only the information in each pixel.

⇒ The basic principle of lossless predictive coding is to code only new information which is obtained as the different between predictive value of pixel & actual value of pixel.

⇒ The new information of a pixel is defined as a difference between the actual & predicted value of that pixel.

⇒ The system copists of an encoder & a decoder is containing & identical predictor. As each sucessive pixel of the input image denoted f_n, is introduced to the encoder, the predictur generates the anticipated value of that pixels based on some number of post inputs.

⇒ The output of the predictor is then rounded to the nearest integer denoted $\hat{f_n}$ and used to form the difference or prediction error which is coded using the variable length code by the symbol encoder to generate the next element of the compressed data stream.

$$e_n = f_n - \hat{f_n}$$

⇒ The decoder reconstructs e_n from the recieve variable length code words & performs the inverse operations.

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

⇒ In some cases, the predictor is form by the linear combination of previous pixels i.e.

$$\hat{f}_n = \text{round} \sum_{i=1}^{M} d_i f_{n-1}$$

where, d_i = prediction coefficient.

(2) Predictive model for Lossy Compression:

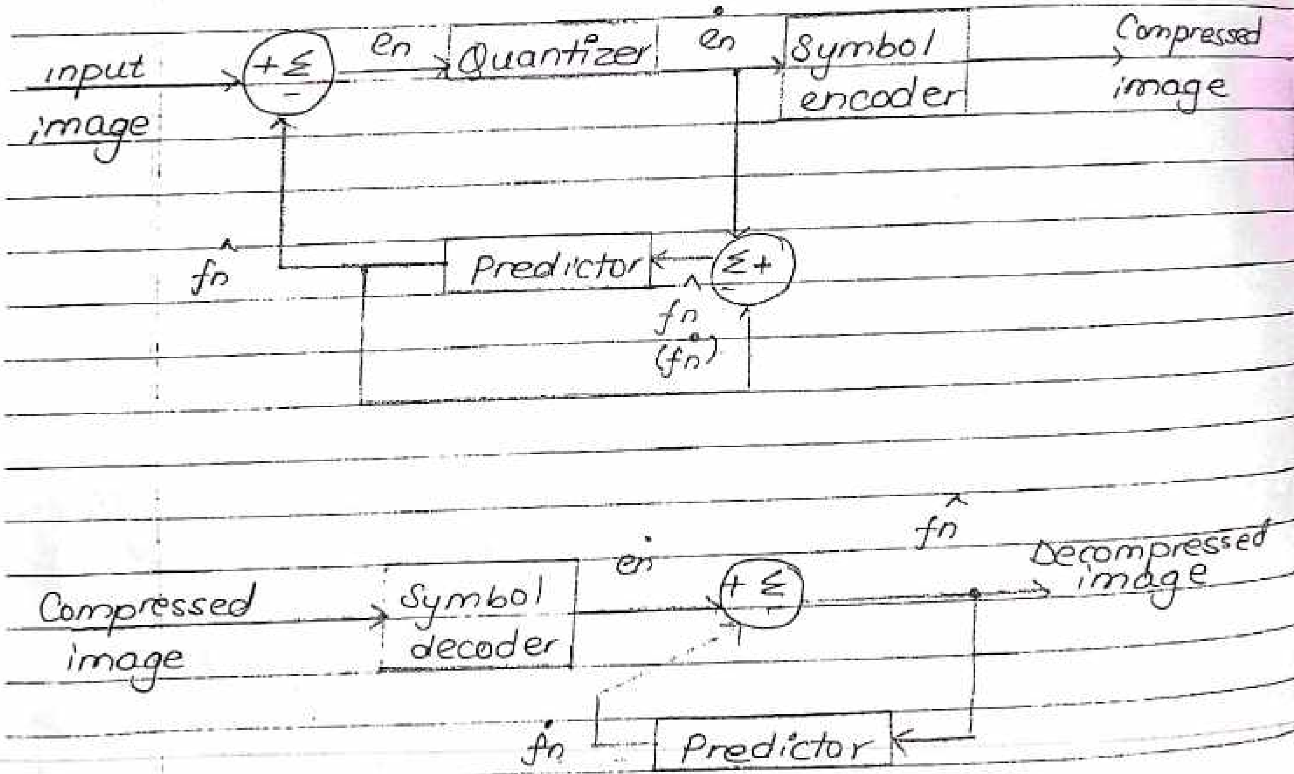


Fig:- A lossy predictive model encoder & decoder

⇒ Here, a quantizer is added to the lossless predictive model & an examined the resulting trade of between reconstruction accuracy

& compressed performance.

⇒ The quantizer which absorbs the nearest integer functions of the error free encoder is inserted between the symbol encoder & the point at which the prediction error is formed.

⇒ It maps the prediction error into a limited range of o/p denoted e_n which established amount of compression & distortion associated with lossy predictive coding.

⇒ In order to accomodate the insertion of the quantization steps, the error free encoder of figure must be altered so that the predictions generated by the encoder & decoder are equivalent

⇒ This is accomplished by placing the lossy encoder prediction with-in a feedback loop where its input denoted $\dot{f}_n$ is generated as a function of past prediction & the correspon-ding quantized error.

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

⇒ This close loop configuration prevents error build up at the decoders outputs.

Introduction to Morphological Image Processing

Chapter-6

⇒ The word Morphology commonly denotes the branch of biology that deals with the form and str. of animal & plant.

⇒ The same word in Morphological Image processing is used in context of mathematical morphology as a tool for extracting image component that are useful in representation & description of region shape such as boundaries, skeletons & convex hull.

⇒ Morphological technique is concerned with pre or post processing such as morphological filtering, thinning and pruning.

⇒ The field of mathematical morphology contributes a wide range of operators to image processing all based around a few simple mathematical concept from set theory.

⇒ The op. are particularly useful for analysis of binary image & common use are age detation noise removal, image enhancement & image segmentation.

⇒ Morphological techniques typically probe an image with a small shape or template known as structuring element.

⇒ Str. element is positioned at all possible locations in the image & is compared with corresponding neighbourhood of pixels.

=> Morphological operation differ in how they carryout this comparision.

Applications:

- Preprocessing
  - => Filtering
  - => Shape simplification

- Segmentation using object shape.

- Object qualification
  - => Area, perimeter, etc.

- Enhancing object structure
  - => Skeleton, thinning, thickening, convex hull, object marketing.

Logic operations involving binary images:

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

Fig. The Logical operations that can be done on binary images (i.e. AND, OR, NOT).

⇒ A binary image is a digital image that has only two possible values for each pixel.

⇒ Typically, the two colors used for a binary image are black & white though any two colors can be used.

⇒ The color used for the objects in the image is the fore ground color while the rest of the image is the background colour.

⇒ In the binary image; Black: representing object denoted by 1
White: representing object denoted by 0

Dilation & Erosion

• Dilation:
With A & B are set in two dimension i.e. Z^2 the dilation of image element A by structuring element B by $A \oplus B$, is denoted as:

$A \oplus B = \{P \in Z^2$ such that $p = a+b, a \in A, b \in B\}$
This means dilation of A by B consist of a set of point 'p' in two dimensional space such that p is the sum of every points of A with every points of B where $a \in A, b \in B$

⇒ It uses the process of vector addition.

⇒ This process "grows" or "thickness" object in a binary image.

⇒ The specific manner & extent of thickning is controlled by the shape of structuring element used.

⇒ One of simplest application of dilation is for bridging the gaps.

⇒ The basic effect of the operator on a binary image is to gradually enlargethe boundary of region of or ground pixels.

⇒ Thus, the areas of fore ground pixel grow in size while holes within those region become smaller.

⇒ It removes the noise present in the image by filling of the gap in bet^n the images. But the boundary of image gets expanded on dilation.

- Erosion:

⇒ with A & B sets in 2-D i.e. Z^2, the erosion of image element A by structuring element B i.e. $A \ominus B$ and defined as:

$$A \ominus B = \{p \in Z^2 \text{ such that } p+b \in A, \text{ for every } b \in B\}$$

This means erosion of A by B consist of a set of point P in 2-D space such that the sum of the point P and structuring element co-ordinate b∈A for every points of b∈B.

⇒ It uses the process of vector subtraction.

⇒ This process causes object to shrink in the image.

⇒ The specific manner & extent of this thickening is controlled by the shape of structuring element used.

⇒ One of the simplest application of erosion is for removing the external noise associated with the boundaries.

⇒ The basic effect of the operator on a binary image is to erode away the boundaries of regions of fore ground pixels.

⇒ Thus, areas of fore ground pixels shrine in size and holes within those areas become larger.

Opening and Closing:

⇒ Opening generally smoothens contour of an object breaks narrow isthemuses and eliminates protrusions.

⇒ Closing also smoothens the section of contours but an oppose to opening but generally fuses narrow breaks & long thin gulfs, eliminates small holes & fills gaps in contour.

⇒ The opening of set A by structuring element B denoted by: $A \circ B$ is defined as:

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

Thus, the opening A by B is the erosion A by B, followed by dilation of the result by B.

⇒ Similarly, the closing of set A by structuring element B denoted by $A \bullet B$ is defined as:

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

⇒ Thus, says that closing of A by B is simply the dilation of A by B followed by the erosion of the result by B.

Image Segmentation

chapter - 7

⇒ Image segmentation is process of sub dividing an image into constituent part into image

⇒ The main purpose of sub dividing an image into its constituent parts or object is to ease in further analysis of this constituents in order to extract some desired information.

⇒ The level of segmentation of an image is application dependend for example

- If we are interested in detecting the movement of vehicle on a road/on a busy road our interest is to detect the moving vehicle on the road.

So, the first level of segmentation should be the extraction of road from those aerial images After the identification of road further analysis can be done to identify every individual vechicle on road There after motion analysis of vehicle can be done Thus. level of sub division can be terminated after extraction of desire component from image.

⇒ Hence, image segmentation is an application dependent process of sub dividing an image into constituent part such that the desire information can easily be extracted from an image.

Approaches to segmentation.

⇒ Image Segmentation approaches are mainly of two different types.

i) Discontinuity based
ii) Similarity based

i) Discontinuity based approach:

⇒ Here, the paition or sub division of image is carried out based on some abrutt change in intesity level in an image or abrutt change of graylevel of ".

⇒ Major focus is on the identification of isolated points or lines or edges present in the image

⇒ The detection of isolated points or lines or edges is done using mask.

⇒ In mask processing operation, mask is shifted over the entire image to calculate some weighted sum of pixel at a particular location.

⇒ The response of mask is evaluated in detection of abrutt changes by using the formula for the below image & mask.

⇒ The response of mask is defined w.r.to its center location.

Image

	(x,y)	

Mask

$W_{-1,1}$	$W_{0,1}$	$W_{1,1}$
$W_{-1,0}$	$W_{0,0}$	$W_{1,0}$
$W_{-1,-1}$	$W_{0,-1}$	$W_{1,-1}$

$$R = \sum_{i=-1}^{1}\sum_{j=-1}^{1} W_{i,j}\, f(x+i, y+j)$$

⇒ Depending upon what are the coefficient values of this mask that we choose, we can have different types of image processing operations.

a. Point Detection

⇒ The basic idea of point detection is that an isolated point (whose grey level is significantly difference from its background and which is located in homogeneous area) will be quite different from its surroundings & thus can easily be detected by mask.

⇒ Here, a point is considered to be detected at the location on which the mask is centered if $|R| > T$, where T is a non negative threshold & R is the response of the mask.

⇒ An isolated point can be detected using the mask show below:

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

⇒ This formulation measures the weighted difference between the center point & its neighbours.

b) Line Detection:

⇒ Line can be detected using various masks.

⇒ The horizontal mask, +45° mask, vertical mask & -45° mask assist in line detection.

⇒ For detection of horizontal lines, a mask can be used with values at the center row or the middle row having all equal to 2 & the top row & the bottom row having all values equal to minus 1. By moving this mask over the entire image, it detects all these points which lies on a horizontal line.

⇒ Similar is with the use of +45° mask, vertical mask & -45° mask to detect the tilted & vertical lines.

-1	-1	-1
2	2	2
-1	-1	-1

Fig: Horizontal line detection mask

-1	-1	2
-1	2	-1
2	-1	-1

Fig +45° line detection mask.

-1	2	-1		2	-1	-1
-1	2	-1		-1	2	-1
-1	2	-1		-1	-1	2

Fig: Vertical line detection mask   Fig - -45° line detection mask

⇒ Suppose, that the four masks (R1 to R4) are seen individually through an image then if at a certain point of image, $|R_i| > |R_j|$ for all $i \neq j$, that point is said to be more likely associated with a line in the direction of mask i.

⇒ For eg:- If at a point in the image, $|R_1| > |R_j|$ for j = 2, 3, 4 then that particular point is said to be more likely associated with a horizontal line.

c) Edge Detection

⇒ It is the most common approach for detecting meaningful discontinuities in gray level.

⇒ Basic formulation and initial assumptions in edge detections are:

- An edge is boundary between two regions with relatively distinct gray level properties.
- Regions are sufficiently homogeneous so that the transition between the regions can be determined on the basis of gray level discontinuity alone.

⇒ If this is not valid, some other techniques will be used.

⇒ The basic idea behind most edge detection techniques is the computation of a local derivative operator.

⇒ Basically, two derivative operators are involved in it.

- The 1st order derivative & 2nd order derivative.

⇒ The 1st order derivative tells us where an edge is and 2nd order derivative can be used to show the edge direction i.e. whether an edge pixel lies on the dark or light side of an edge.

⇒ 1st order derivative can be computed using various gradient operator (ROBERTS, Perwitt and Sobel operators)

⇒ For a given 3x3 region of an image

$$\begin{matrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{matrix}$$

⇒ Computation of the gradient of an image is based on obtaining partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$

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

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

⇒ For Roberts operator

$G_x = (Z_9 - Z_5)$

and

$G_y = (Z_8 - Z_6)$

⇒ For perwitt

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

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

⇒ For Sobel

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

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

⇒ 2nd order derivative can be computed using Laplacian operator.

* Similarly based approach.

⇒ The basic idea in similarity based approach for image segmentation is to group those pixels in an image which are similar in some sense.

⇒ This approach can be implemented in two ways.

a) Segmentation by thresholding

b) Region extraction

a) Segmentation by thresholding:

⇒ Here, the segmentation is done with the aid of threshold value.

⇒ The threshold value is choosen following some criteria or feature of an image.

⇒ Then, every pixel of the image is compared to threshold value.

⇒ On comparision all the pixels having intensities value greater than the threshold value belong to one region and the rest pixels belong to other region.

⇒ So, this is the simplest thresholding operation that can be used for image segmentation purpose.

Thresholding can be done by following ways:

1. Feature thresholding

⇒ It is the simplest method of image segmentation.

⇒ Conside an image that contains two type of regions & the distinctness of the region is reflected by the feature value at the pixel belonging to them. Suppose there exist a threshold (t) such that the feature values of all pixels that actually belong to region of 1st type are less than or equal to t. The gray level values of all pixels that actually belong to region of the 2nd type are greater than t.

Hence, segmented image is represented as

$$b(r,c) = \begin{cases} 1 & \text{if } p(r,c) \le t \\ 0 & \text{if } p(r,c) > t \end{cases}$$

where, $p(r,c)$ is the feature value at pixel (r,c) & values 1 and 0 can be replaced by other symbols too.

⇒ The threshold can be treated as a class boundary.

⇒ For the proper image segmentation by thresholding operation, the appropriate choice of feature must be done to achieve the desired segmentation. Also the selection of optimum threshold should be made that would incur least classification error.

ii) Amplitude thresholding or window slicing:

⇒ It is as useful whenever the amplitude features sufficiently characterised the object.

⇒ The appropriote amplitude feature values are callibrated so that a given amputude interval represents unique object characteristics. ~~For ex:- in medical~~

For ex:- in medical x-ray images, the gray level amplitude represents the absorption characteristics of a body masses and enables discrimination of bones from tissue or healthy tissue from diseased tissue.

Methods of selecting a Threshold value:

1. Threshold Selection using histogram.

⇒ Our objective is to segment an image ~~$a(r,c)$~~ $g(r,c)$ of size $M \times N$ that contains two types of region R1 & R2 by graylevel thresholding.

⇒ Threshold can be selected using the information contain by graylevel histogram of image.

⇒ The threshold value is selected from the histogram in such a way that the pixel in image are uniformly distributed.

⇒ This method can't be applied to unknown class of image.

2. Multi-level thresholding:

⇒ Suppose an image contains more than 2 types of region i.e. it has more than one valley in the histogram.

⇒ In such condition, the feature histogram of image is expected to contain n no. of peaks, each of which corresponds to ~~distinct~~ distinct type of region.

⇒ Thus, we have $(n-1)$ valley in the histogram & select gray level corresponding to the bottom of these valleys as threshold for image segmentation.

3. Local thresholding

⇒ Consider due to variation in illumination, overall brightness of the image varies widely from corner to the other.

⇒ In this case, the graylevel distribution for object as well as further background overlap eachother significantly.

⇒ No single graylevel threshold can segment the image properly.

⇒ Thus, to solve this we divide the whole image into smaller once so that variation in illumination over these sub-image is neglible.

⇒ Each sub image is segmented independently & segmented sub images are put together in appropriate order to get the segment version of the original image.

Region Extraction

⇒ Suppose spatial domain on which image is defined is denoted by V.

⇒ The image segmentation techniques divides V into n regions. Denoted by ~~R_i~~
R_i $(i = 1, 2, \ldots n)$ such that

$$\bigcup_{i=1}^{n} R_i = V$$

$$R_i \cap R_j = \emptyset \text{ or } \phi$$

$prop(R_i) = True$

$prop(R_i \cup R_j) = False$

⇒ The first condition say that every pixel of image domain is mapped to one region or the other.

⇒ The second condition ensures that a pixel is mapped to only one region

⇒ Third condition indicate that the regions are defined on some property.

⇒ finally, maximality of each region is assured by the forth condition.

Region Growing:

⇒ let us pick an arbitary pixel (r,c) from the domain of the image to be separated

⇒ This pixel is called seed pixel and this pixel belongs to some other regions

⇒ Examine the nearest neighbours (4 or 8 neight) depending on the connectivity of (r,c) 1x1 & a neighbouring pixel is accepted to belong to the same region as (r,c) if they together satisfy the homogeneity property of region

⇒ Once a new pixel is accepted as a member of current region, the nearest neighbour of this new pixel are examined.

⇒ This pixel goes on recursively until no more pixel is accepted.

⇒ All pixels of the current region are marked with unique label.

⇒ Another seed pixel is picked up and same procedure is repeated

⇒ Region labeling is done until every pixel is assigned to some region or the other.

⇒ Main problem of this approach in addition to large execution time are the selection of property to be satisfied and selection of an appropriate seed point.

Region splitting.

⇒ Suppose we try to satisfy homogenity property over a rectangular region.

⇒ If the graylevel present in the region doesnot satisfy the property then the region is further divided in four equal quadrants

⇒ If the property is satisfied, the region is left as it is

⇒ In terms of graph theory, the region is split into 4 children if it doesnot satisfy the given property otherwise the node is left unaffected.

⇒ This method is applicable to images whose number of rows & column are integer power of 2.

Region Merging:

⇒ This method is exactly opposite to the region splitting method.

⇒ Region merging method is a ~~region~~ bottom up method.

⇒ This method is also applicable to images whose no. of rows & no. of columns are some integer power of 2.

⇒ At any level of merging, one can check if 4 adjacent regions arranged in 2x2 together satisfies the homogenEity property.

⇒ If yes the regions can be merged to a single homogeneous region otherwise the regions are left as it is.

⇒ In terms of graph theory, the above explanation can referred as child nodes are removed if the parent node satisfies the homogeneity.

Split & merge:

⇒ If most of the homogeneous region are small, it takes more time to split the regions.

⇒ Further, if we donot have priori knowledge about size of region, in that case too it takes more time.

⇒ Thus, this is a hybrid approach which combines both spliting & merging techniques.

⇒ Suppose if we start with rectangular region of size $m \times m$ pixels then for each region, homogeneity property is tested.

⇒ If test fails, the regions are split into 4 quadrants each of size $m/2 \times m/2$

⇒ If the region satisfies the homogeneity property then merging process is followed to form region of size $2m \times 2m$.

Histogram Equalization Algorithm:

→ Histogram of a digital image with gray levels in the range [0, L-1] is a discrete function:

$$h(r_K) = n_K$$

where,

r_K = the K^{th} gray level.

n_K : the no. of pixels in the image having gray level n_K

$h(r_K)$: histogram of a digital image with gray level r_K.

Steps:

I) Find the "running sum" of the histogram values.

II) Normalize the values from step 1 by dividing the total number of pixels.

III) Multiply the values from step 2 by the maximum gray level value & round.

IV) Map the gray level values to the results from step 3 using one-to-one correspondance.

Example

Input Image

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4×4 Image

Grayscale: [0, 9]

Solution :

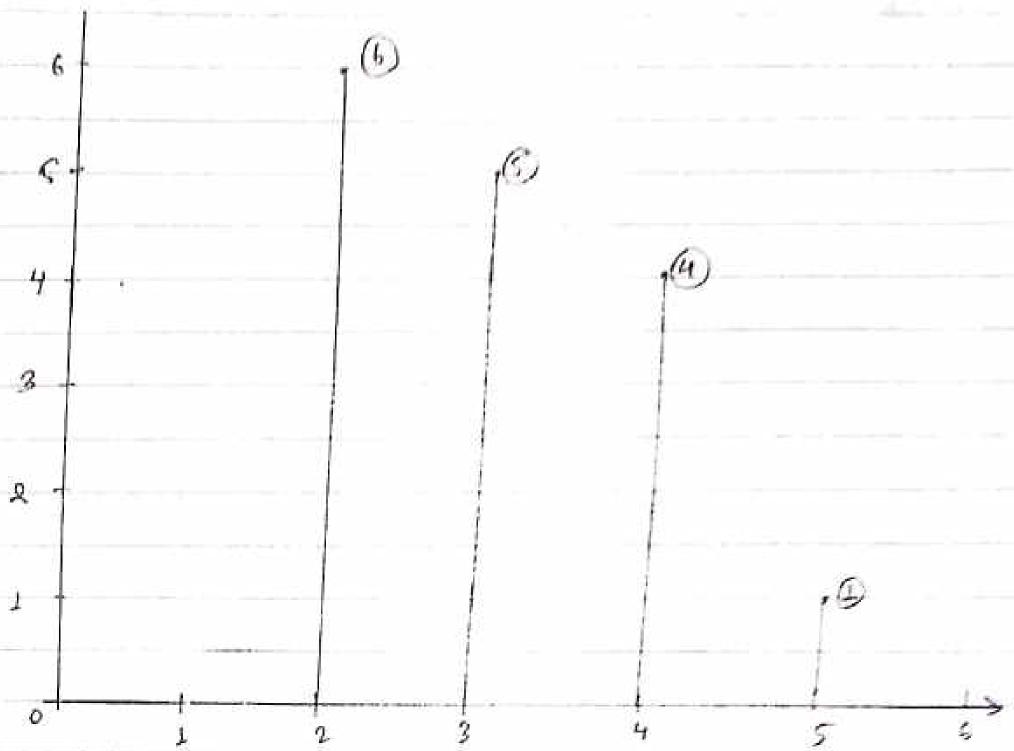


Fig:- histogram of give image

Total N = 6+5+4+1 = 16

Gray level j	0	1	2	3	4	5	6	7
No of pixel	0	0	6	5	4	1	0	0
Running sum	0	0	6	11	15	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	$\frac{0}{16}$	$\frac{0}{16}$	$\frac{6}{16}$	$\frac{11}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$
Multiply by max graylevel $s * 9$	$\frac{0}{16}*9$ = 0 = 0	$\frac{0}{16}*9$ = 0 = 0	$\frac{6}{16}*9$ = 3.37 = 3	$\frac{11}{16}*9$ = 6.18 = 6	$\frac{15}{16}*9$ = 8.43 = 8	$1*9$ = 9 = 9	$1*9$ = 9 = 9	$1*9$ = 9 = 9
output gray level	0	0	3	6	8	9	9	9

0 $\xrightarrow{\text{replaced}}$ 0

1 ⟶ 0

2 ⟶ 3

3 ⟶ 6

4 ⟶ 8

5 ⟶ 9

6 ⟶ 9

7 ⟶ 9.

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

⟹

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

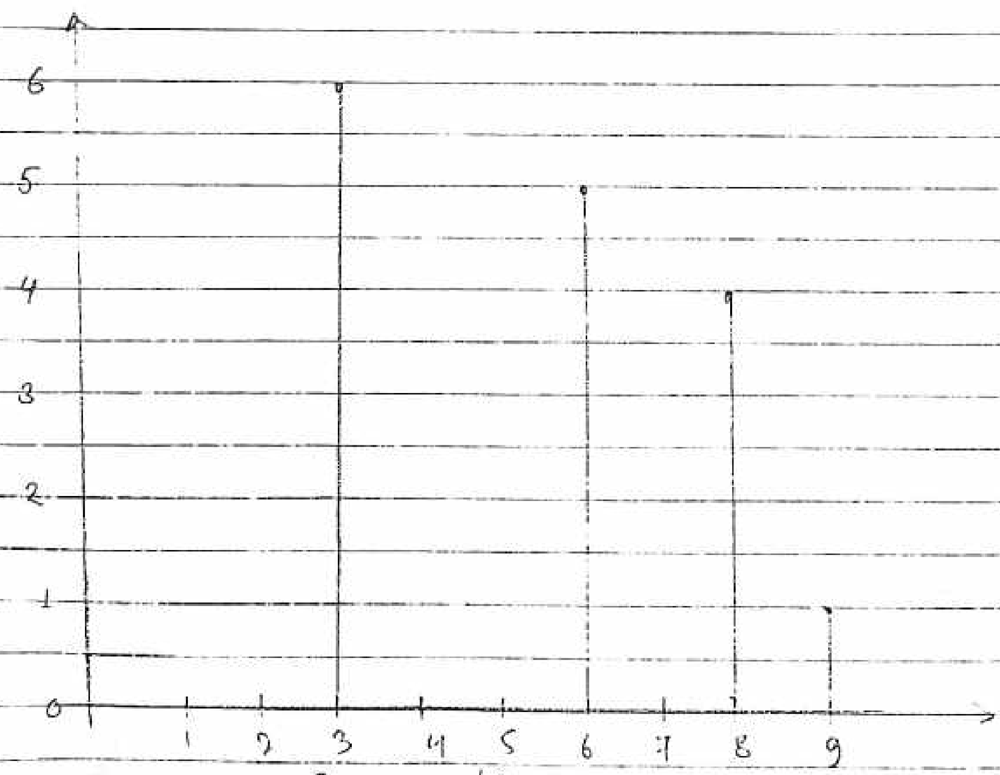


Fig:- Output histogram.

Q. Construct Huffman code for each gray level given and find the compression ratio & code efficiency.

Graylevel	0	1	2	3	4	5	6	7
No. of pixel	30	35	38	10	15	10	38	80

Gray level	n_K	$P_K (r_K/N)$
0	30	$30/256 = 0.12$
1	35	$35/256 = 0.136$
2	38	0.15
3	10	0.04
4	15	0.06
5	~~N=~~	0.04
6	38	0.15
7	80	0.32
	$N = 256$	

Step 2: Arranging in descending order & reducing it to binary huffman

r_K	$P(r_K)$	1	2	3	4	5	6 (suppose)
7	0.32	0.32	0.32	0.32	0.32	0.4	0.61(0)
2	0.15	0.15	0.15	0.25	0.24	0.32	0.4(1)
6	0.15	0.15	0.15	0.15	0.25	0.29	
1	0.13	0.13	0.14	0.15	0.15		
0	0.12	0.12	0.13	0.14			
4	0.06	0.08	0.12				
3	0.04	0.06					
5	0.04						

continue.....

Poozabh Siha

Representation & description

Chapter-8

Introduction

⇒ After an image is segmented ~~in two~~ into regions, the resulting aggregate of segmented pixel usually is represented & described in a form suitable for further computer processing.

⇒ Representation can be done in two ways:
1) In terms of external Characteristics.
2) " " internal "

⇒ An external representation is choosen when the primary focus is on shape, characteristics like length, orientation of st. line joining its extreme point.

⇒ An internal representation is chosen when primary focus is on regional properties such as colour or texture.

⇒ The next step is to describe the represented image.

⇒ While describing or representing an image, the features choosen for representation must be insensitive to any transformations.

Descriptors:

(1) Chain code.

⇒ They are used to represent a boundary by a connected sequence of st. line segments of specified length & direction

- This representation is based on 4 or 8 connectivity of the segment.

- The direction of each segment is coded by using numering skill scheme.

- A boundary code form as a sequence of such directional number is referred to as free man chain code.

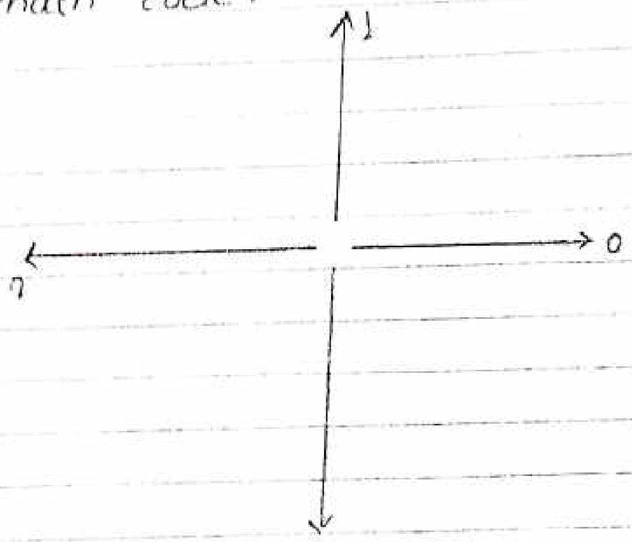
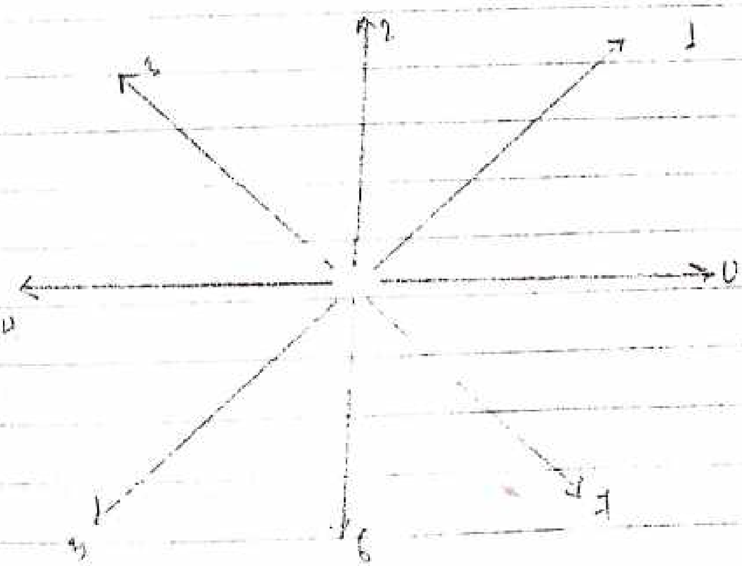


Fig. 4-connectivity segment



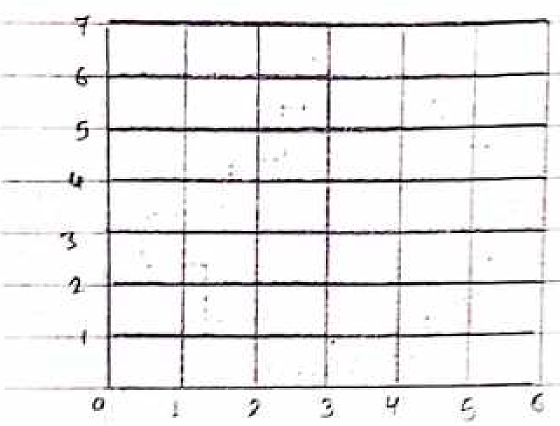


fig digital boundary with resampling grids superimposed

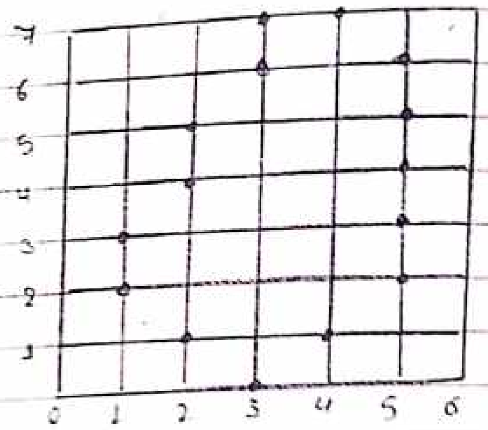


fig b result of resampling

fig:- 4 directional chain coded boundary

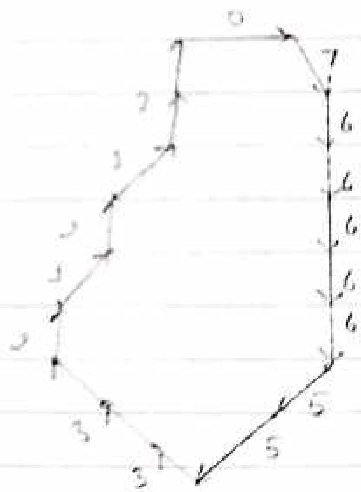


fig:- 8 directional chain coded boundary.

:) Digital images ~~are~~ usually are acquired are processed in a grid format with equal spacing in x,y direction so chain code can be generated by a boundary in shape of clockwise direction &

assigning a direction to the segment connecting every pair of pixel.

=> As the boundary in fig(a) is traversed, a boundary point is assigned to each node of the large grid depending on the proximity of the digital & original boundary to that node as in fig(b)

=> The resampled boundary obtain in this way then can be represented by a 4 or 8 code

=> This method is generally unexceptable becouse of:

i) The resulting chain tends to be quite long
ii) Any small disturbances along the boundary due to noise or imperfect segmentation cause changes in the code that may not be related to the principle shape feature of boundary.

=> In order to overcome these problems is to resample the boundary by selecting a larger grid spacing.

=> This approach of selecting larger grid spacing results in scaling as well as translation in variant but rotation in variant feature of the discriptor is not achieved.

=> To overcome this problem of rotation variant.

the differential chaincode is introduced where the first difference of chain code is used instead of code itself.

⇒ This difference is obtained by counting the no. of direction changes (in counterclockwise direction) that separates two adjacent element of code. ***

Continue....

Step 3: Coding

r_k	$P(r_k)$	1	2	3	4	5	6
7	00	00	00	00	00	1	0
2	11	11	11	10	01	00	1
6	010	010	010	11	10	01	
1	100	100	011	010	11		
0	101	101	100	011			
4	0111	0110	101				
3	01100	0111					
5	01101						

Average length = $0.32 * \overset{\text{bits}}{2} + 0.15 * 2 + 0.15 * 3 + 0.13 * 3 + 0.12 * 3 + 0.06 * 4 + 0.04 * 5 + 0.04 * 5$

$= 2.78$ bits / symbol

Coding Redundancy $(CR) = \frac{n1}{n2} = \frac{3}{2.78} = 1.08$

where,

n1 = traditional coding bits per symbol

n2 = bits per symbol obtained from Huffman

$0 \rightarrow 7$ $\qquad\qquad$ $\log_a b = \dfrac{\log_{10} b}{\log_{10} a}$

$2^3 = 3$ bit

n_1 can be calculated from given number of grey levels : i.e. 0 to 7

so, number of bits $n_1 = 3$ to code 0 to 7

Compression ratio $= \dfrac{n_1}{n_2} = \dfrac{3}{2.78}$

Coding efficiency $= \dfrac{H(3)}{L}$

where,

$H(s) = \sum_{i=p}^{n-1} P_i \log_2 \dfrac{1}{P_i}$ where $i = 0, 1, 2 \ldots 7$

$= 0.32 \log_2 \dfrac{1}{0.32} + 0.15 \log_2 \dfrac{1}{0.15} + 0.15 \log_2 \dfrac{1}{0.15} +$

$0.13 \log_2 \dfrac{1}{0.13} + 0.12 \log_2 \dfrac{1}{0.12} + 0.06 \log_2 \dfrac{1}{0.06} +$

$0.04 \log_2 \dfrac{1}{0.04} + 0.04 \log_2 \dfrac{1}{0.04}$

$= 2.71$ bits/symbol

Coding Redundancy (CR) $= \dfrac{n_1}{n_2} = \dfrac{3}{2.78} = 1.08$

where,

n_1 = traditional coding bits per symbol.

n_2 = bits per symbol obtained from Huffman.

n_1 can be calculated for given number of gray levels

i.e. 0 to 7. so, number of bits $n_1 = 3$ to code 0 to 7.

Compression ratio $= \frac{n_1}{n_2} = \frac{3}{2.78}$

code efficiency $= \frac{H(S)}{L}$

where,

$$H(S) = \sum_{i=p}^{n-1} P_i \log_2 \frac{1}{P_i} \quad \text{where } i = 0, 1, 2 \ldots 7$$

$$= 0.32 \log_2 \frac{1}{0.32} + 0.15 \log_2 \frac{1}{0.15} + 0.15 \log_2 \frac{1}{0.15} + 0.13 \log_2 \frac{1}{0.13}$$

$$+ 0.12 \log_2 \frac{1}{0.12} + 0.06 \log_2 \frac{1}{0.06} + 0.04 \log_2 \frac{1}{0.04} + 0.04 \log_2 \frac{1}{0.04}$$

$$= 2.71$$

$$\text{Efficiency} = \frac{H(S)}{l} = \frac{2.71}{7} = 0.39$$

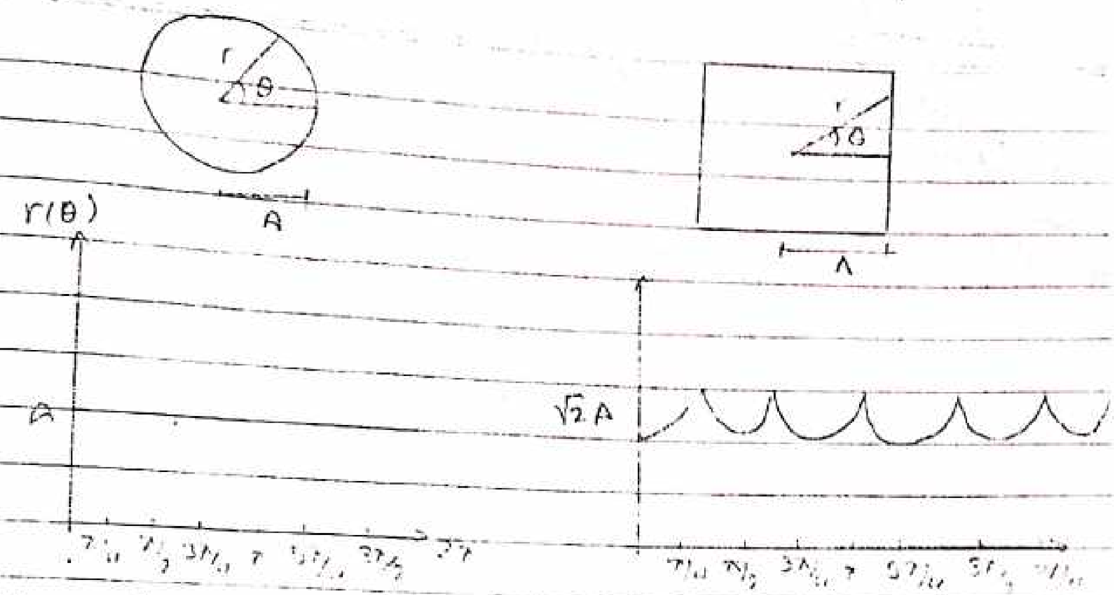
***

b) Signature:

⇒ A signature is a 1-D functional representation of a boundary and may be generated in various ways.

⇒ One of the simplest way is to plot the distance from the centroid to the boundary as a function of an angle.

⇒ This approach is invarient to translation but is dependent on rotation & scaling

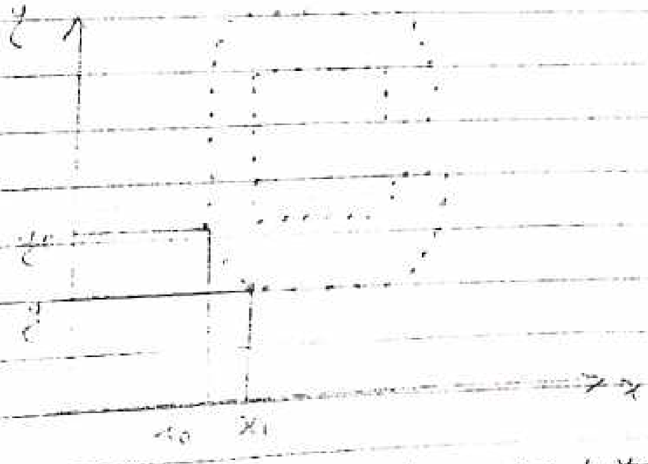


→ Here, in case of circle, the radius is constant through out the angle and is shown in figure.

⇒ In the case of square and angle 45°, it posses the longer distance as composed at 90°. Thus, the graph/signature consists of repetation of pattern.

$$r(\theta) = \begin{cases} A\sec\theta & \text{for } 0 \le \theta \le \pi/4 \\ A\cos\theta & \text{for } \pi/4 \le \theta \le \pi/2 \end{cases}$$

c) Fourier Descriptors.



⇒ Fig A:- digital boundary and its representation as a complex sequence. The points (x_0, y_0)

and (x_1, y_1) shown are (arbitarily) the first two points in the sequence.

⇒ Figure above shows a k-point digital boundary in the xy-plane.

⇒ Starting at arbitary point (x_0, y_0) co-ordinate points $(x_0, y_0), (x_1, y_1), (x_2, y_2) \cdots (x_{k-1}, y_{k-1})$ are encountered in traversing the boundary say in the counter clockwise direction.

⇒ These co-ordinates can be expressed in the form $x(k) = x_k$ and $y(k) = y_k$ with these notation, the boundary can be represented as the sequence of co-ordinates $s(k) = [x(k), y(k)]$ for $k = 0, 1, 2 \ldots k-1$.

⇒ If each co-ordinate pair is treated as a complex number so that $s(k) = x(k) + jy(k)$ for $k = 0, 1, 2 \ldots k-1$. This means the x-axis is treated as the real axis and the y-axis as the imaginary axis of a sequence of complex number.

⇒ This representation reduces a 2-D problem to 1-D problem.

⇒ The discrete Fourier Transform of $s(k)$ is
$$a(u) = \sum_{k=0} s(k) e^{-j2\pi uk/k} \quad \text{for } u = 0, 1, 2 \ldots k-1$$

⇒ The complex coefficient $a(u)$ are called Fourier Descriptors of the boundary.

⇒ The inverse Fourier transform of these coefficients restores $s(k)$

i.e. $s(k) = \frac{1}{K} \sum_{u=0}^{K-1} a(u) e^{j2\pi uk/K}$ for $k = 0, 1, 2 \ldots K-1$

⇒ However, instead of all Fourier coefficient, only the first P coefficients are used. This results in the following approximation of $s(k)$:

$$\hat{s}(k) = \frac{1}{P} \sum_{u=0}^{P-1} a(u) e^{j2\pi uk/P} \quad \text{for } k = 0, 1, 2 \ldots k-1$$

⇒ This means the same number of points exists in the approximate boundary, but not as many terms are used in the reconstruction of each point. Thus, the smaller 'p' becomes the more detail is lost on the boundary.

(d) Shape number:

⇒ The shape number of a boundary is defined as the first difference of the smallest magnitude.

⇒ The first difference of a chain coded boundary depends on starting point.

⇒ The order 'n' of a shape number is defined as the number of digits in each representation.

⇒ Here, the first difference is computed by treating the chain as a circular sequence. Although the first difference of the code is independent of rotation, in general, the coded boundary depends on the orientation of the grid.

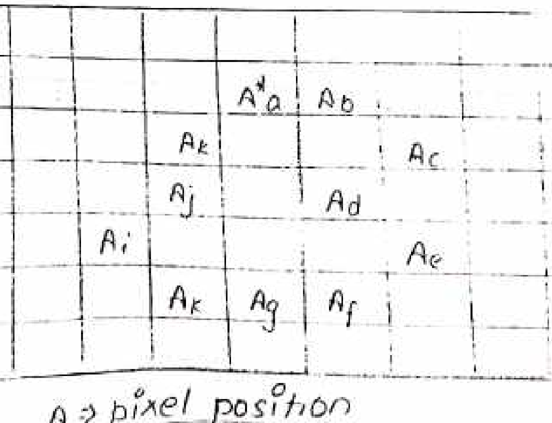
⇒ Depending upon the starting point, the chain code is different but the cycle remain same. Among all the various code combination thus formed, such a number is choosen as a starting point that leads to the lowest numerical value of the chain code. which is known as a shape number.

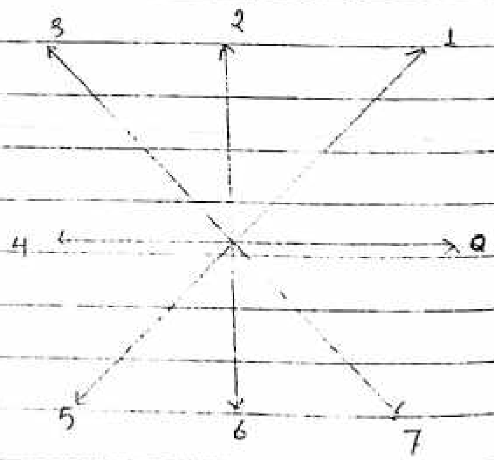
⇒ Shape number is a boundary scriptor which is invariant to rotation, translation and scaling.

⇒ Shape number helps in shape understanding of the object.

⇒ Here the chaincode is treated as a circular sequence of direction number and starting point is redefined so that the resulting sequence of number form an integer of minimum magnitude which is a shape number.

Qn. Given an image, write down the 8-chain code and find shape number of it.



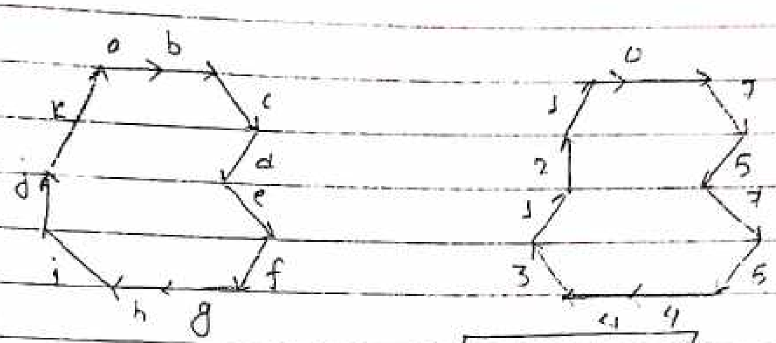


A ⇒ pixel position

a* → starting position

Fig:-8-chain

→ Solⁿ

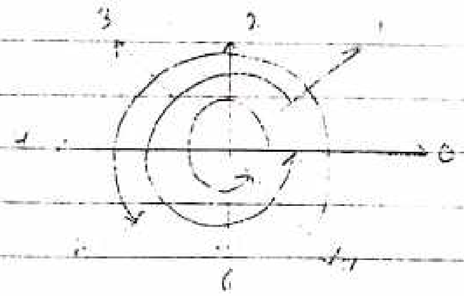


Chain code = 0757544 3121

First difference:- 776267076 1 7

Shape no·- start from smallest no.)

0761 7 7 776267



Dilation and Erosion / Opening & closing :-

Image

A :

{1	0	0}	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	1	0

structuring element

B :

1	1	1

center element for circle

[first 3 element are taken since 3 structuring elements are given.]

CamScanner

⇒ Solution.

Erosion $E(A)$ =

1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1 0 0
1 1 1
1 0 0

smallest is 0.
so, it is taken in A_{12}.

Dilation $D(A)$ =

1	1	0	0	0
0	1	1	0	0
0	1	1	1	0
0	0	1	1	0
0	0	1	1	0

Opening:- Opening is done by first process of erosion and by dilation

$D[E(A)]$ =

1	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Closing :-

$E[D(A)]$ =

1	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Hough Transform:

⇒ The Hough Transform can be used to detect lines, circles or other parametric curves.

⇒ It was introduced in 1962 & first used to find lines in images a decade later.

⇒ The goal is to find the location of lines in images The problem could be solved by example morphology and a linear structuring element or by correlation. Then we would need to handle rotation, zoom, distortion, etc.

⇒ Hough Transform can detect lines, circles and other structures if there parametric eq^n is known.

⇒ It can give robost detection under noise and partial occlusion.

⇒ Two point (x,y) and (z,k) define a line in (x,y) plane.

⇒ These two points give rise to two difference lines in (a,b) space.

⇒ In (a,b) space these lines will intersects in point (a',b').

where a' = rise

b' = intersect of the line defined by (x,y) & (z,k) in (x,y) space.

⇒ The fact is that all points on the line defined by (x, y) & (z, k) in (x, y) space will parameterized lines the intersect in (a', b') in (a, b) space)

⇒ Points that lie on a line will form a "cluster of crossings" in the (a, b) space.

Hough Transform algorithm:

⇒ Quantize the parameter space (a, b), that is divide it into cells.
cells.

⇒ This quantized space is often referred as accumulator.

⇒ In fig, amin = minimal value of a.

⇒ • Count the no. of times line intersect a given cell.

• For each point (x, y) with value 1 in binary image, find value (a, b) in range [(amin, amax), (bmin, bmax)] defining the line corresponding to this points.

• Increase the value of accumulator for [a', b']

• Then proceed the next point in range.

⇒ Cells recieving a minimum no. of 'votes' are assumed to correspond to lines in this space.

⇒ lines can be found as peaks in this accumulator space.

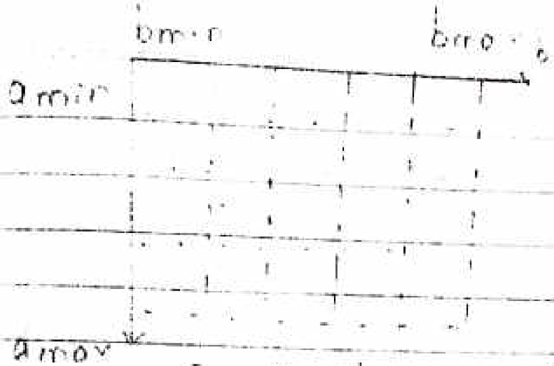


Fig - Hough accumulator cells.

Example: Eq^n of a line

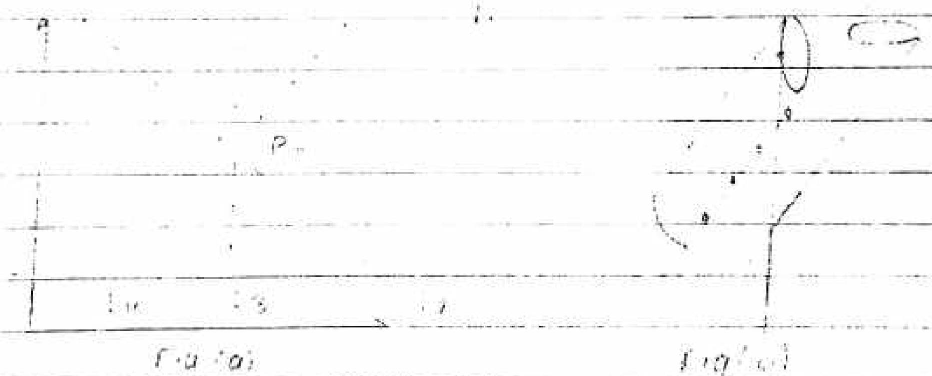
$$y = kx + d$$

where, k = slope

d = intercept

Hough transform: starts with edge point (x, y)

- find (slope k, intercept d) that passes through the most edge points.



- Consider the point $(x, y) = (1, 1)$
- Many lines at different slope k & intercept 'd' can pass through $(1, 1)$
- Thus, we can rewrite equation of line through $(1, 1)$ as:

$$1 = k \cdot 1 + d$$

$$\Rightarrow d = -k + 1$$

→ Set of all lines passing through (1,1) fig a can be shown as line parameter space in following fig:-

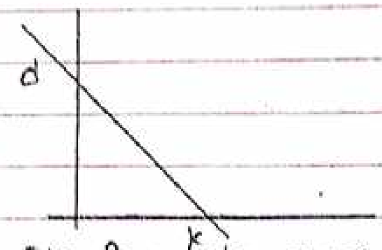
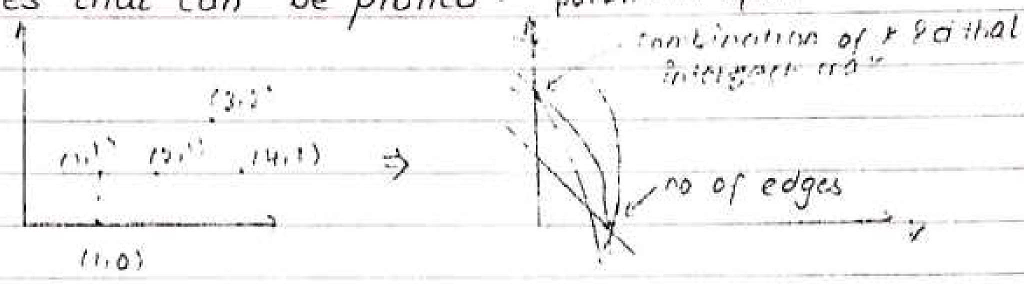


Fig · Parameter space

⇒ Suppose image has 5 points:
(1,0), (1,1), (2,1), (4,1) & (3,2)

⇒ Each of these points corresponds to the following lines that can be plotted.



$(1,0) \Rightarrow d = -k$

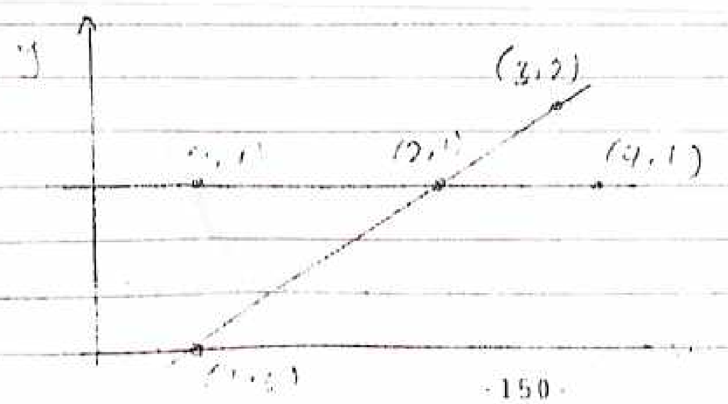
$(1,1) \Rightarrow d = -k+1$

$(2,1) \Rightarrow d = -2k+1$

$(4,1) \Rightarrow d = -4k+1$

$(3,2) \Rightarrow d = -3k+2$

⇒ After finding optional values of k & d, we can draw them as:



Object Recognition

Chapter - 9

Introduction to patterns & pattern classes

Pattern:

⇒ A pattern is an arrangement of descriptors
⇒ The name feature is used often in pattern recognition literature to denote a descriptors.

Pattern classes:

⇒ A pattern class is a family of patterns that share some common properties.

⇒ Pattern classes are denoted by $\omega_1, \omega_2 \ldots \omega_\omega$ where ω is the number of classes.

⇒ Pattern recognition by machine involves techniques for assigning patterns to their respective classes automatically and with a little as human intervention as possible.

⇒ Three common pattern arrangements are used in practise which are vectors (for quantitative description) & strings and trees (for structural description)

Recognition Based on Decision Theoretic Methods

⇒ Decision-Theoretic approaches to recognition are based on the use of decision function.

⇒ Let $x = (x_1, x_2 \ldots, x_n)^T$ represent an n^{th} dimensional pattern vector.

⇒ For ω pattern classes $\omega_1, \omega_2 \dots \omega_\omega$, the basic problem in decision theoretic pattern recognition is to find ω decision functions $d_1(X), d_2(X) \dots d_\omega(x)$ with the property that if a pattern x belongs to class ω then $d_i(x) > d_j(x)$ $j = 1, 2, 3 \dots \omega$; $j \neq i$.

⇒ In another words, an unknown pattern X is said to belong to the ith pattern class if, upon substitution of X into all decision functions, $d_i(X)$ yields the largest numerical value. Ties are resolved arbitarily.

⇒ The decision boundary separating class ω_i from ω_j is given by values of X for which $d_i(x) = d_j(x)$ or equivalently by values of X for which $d_i(x) - d_j(x) = 0$.

⇒ Common practise is to identify the decision boundary between two classes by the single function, $d_{ij}(x) = d_i(x) - d_j(x) = 0$. Thus, $d_{ij}(x) > 0$ for patterns of ω_i and $d_{ij}(x) < 0$ for patterns of class ω_j.

Matching:

⇒ Recognition technique based on matching represent each class by a prototype pattern vector

⇒ An unknown pattern is assigned to the class to which it is closest in terms of a predefine matric.

⇒ The simplest approach is the minimum distance classifier which computes the distance between the unknown and each of the prototype vectors.

⇒ It chooses the smallest distance to make a decision.

Imp • Minimum distance classifier.

⇒ Suppose we define the prototype of each pattern class to be the mean vector of the patterns of that class.

$$m_j = \frac{1}{N_j} \sum_{x \in \omega_j} x_j \quad , j = 1, 2, 3, \dots, W$$

where,

N_j = number of pattern vectors from class ω_j

W = no. of pattern classes.

⇒ Using the Euclidean distance to determine closeness, reduces the problem to computing the distance measures:

$$x' = \{x_1, x_2, \dots x_m\}$$

$$D_j(x) = ||X - m_j|| \quad j = 1, 2, \dots w$$

where,

$||a|| = (a^T a)^{1/2}$ is the Euclidean norm.

We assign x to class w_i if $D_i(x)$ is the smallest distance.

i.e. $d_j(x) = x^T m_j - \frac{1}{2} m_j^T m_j$, $j = 1, 2, 3 \dots N$

assign x to class w_i if $d_i(x)$ yields the largest numerical value.

⇒ The decision boundary between classes w_i and w_j for a minimum distance classifier is:

$$d_{ij}(x) = d_i(x) - d_j(x)$$

$$= x^T(m_i - m_j) - \frac{1}{2}(m_i - m_j)^T(m_i + m_j) = 0$$

This equation is the perpendicular bisector of the line segment joining m_i and m_j.

Example:

Two pattern classes denoted by w_1 and w_2 respectively, have sample mean vectors $m_1 = (4.3, 1.3)^T$ and $m_2 = (1.5, 0.3)^T$. Find decision boundary between them.

Soln

$$d_1(x) = x^T m_1 - \frac{1}{2} m_1^T m_2$$

$$= 4.3 m_1 + 1.3 x_2 - 10.1$$

$(4.3, 1.3)^T \begin{bmatrix} 4.3 \\ 1.3 \end{bmatrix}$

$= 4.3 \times 4.3 + 1.3 \times 1.3$

$\frac{(4.3)^2 + (1.3)^2}{2}$

$d_2(X) = X^T m_2 - \frac{1}{2} m_2^T m_2$

$= 1.5 m_2 + 0.3 x_2 - 1.17$

$\therefore$ The boundary eqⁿ is:

$d_{12}(X) = d_1(X) - d_2(X)$

$= 2.8 x_1 + 1.0 x_2 - 8.9$

Matching by Correlation:

⇒ We know correlation of a mask $w(x,y)$ of size $m \times n$, with an image $f(x,y)$ may be expressed in the form

$$c(x,y) = \sum_s \sum_t w(s,t) f(x+s, y+t) \quad \text{——— (i)}$$

where,

- Limits of summation are taken over the region shared by w and f.
- x and y are the displacement variables so that all elements of w visit every pixel of 'f', where f is assumed to be larger than w.

- Just as spatial convolution is related to the fourier transform of the function via the convolution theorem, spatial correlation is related to the transforms of the functions via the correlation theorem:

$$f(x,y) \circledast w(x,y) \Longleftrightarrow F^*(u,v)\, W(u,v) \quad \text{—— (11)}$$

where $\circledast$ ⇒ represent spatial convolution
F^* ⇒ complex conjugate of F

Normalized correlation coefficient is:

$$\Gamma(x,y) = \frac{\sum_s \sum_y [w(s,t) - \bar{w}] \sum_s \sum_t [f(x+s,y+t) - \bar{f}(x+s,y+t)]}{\left\{ \sum_s \sum_t [w(s,t) - \bar{w}]^2 \sum_s \sum_t [f(x+s,y+t) - \bar{f}(x+s,y+t)]^2 \right\}^{1/2}}$$

where, limits of summation are taken over the region shared by w and f, $\bar{w}$ is the average value of the mask and $\bar{f}(x+s,y+t)$ is the average value of 'f' in the region coincident with w. Often, 'w' is referred to as a template and correlation is referred to as "template matching."

Bayes Classifier for Gaussian pattern class:

- Consider a 1-D problem ($n=1$) involving two pattern classes ($W=2$)
- Gaussian densities, with mean m_1 and m_2.
- Standard deviation σ_1 and σ_2.

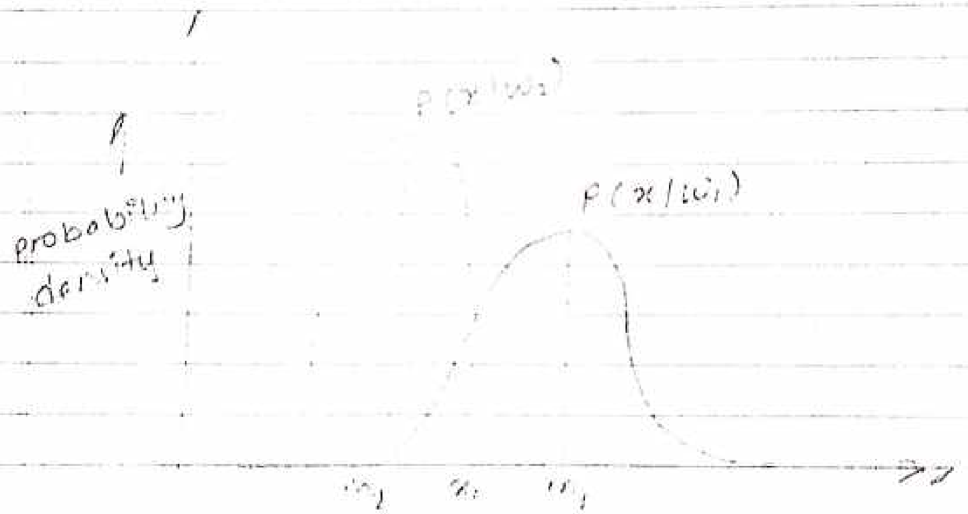
We have, Baysean decision function from the equation.

$d_j(x) = P(x/\omega_j)P(\omega_j)$, $j = 1, 2, 3 \ldots W$

is:

$$d_j(x) = \frac{1}{\sqrt{2\pi}\,\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j) \quad , j = 1, 2$$

where, the patterns are now scalars, denoted by x.



⇒ boundary between two classes is a single point declared by x_0, such that

$d_1(x_0) = d_2(x_0)$

⇒ If the two classes are equally likely to occur, then

$P(\omega_1) = P(\omega_2) = 1/2$

⇒ decision boundary is the value of x_0 for which,

$P(x_0/\omega_1) = P(x_0/\omega_2)$

⇒ In the n-dimensional case, the Gaussian density of the vectors in the j^{th} pattern class has the form:

$$P(X/\omega_j) = \frac{1}{(2\pi)^{n/2}|C_j|^{1/2}} e^{-\frac{1}{2}(x-m_j)^T C_j^{-1}(x-m_T)}$$

where,

m_j = mean vector

C_j = covariance matrix

$m_j = E_j\{x\}$ [E_j is expected value]

and

$C_j = E_j\{(x-m_j)(x-m_j)^T\}$

- Neural Networks in Image Processing
  (Read from AI note)

1 hr.

Pattern Recognition

Chapter 10

- Science of making inference from perceptual data.

- From automated speech recognition, fingerprint identification, optical character recognition, DNA sequence identification

- Pattern is defined as composite of features that are characteristic of an individual

Block diagram / Steps:

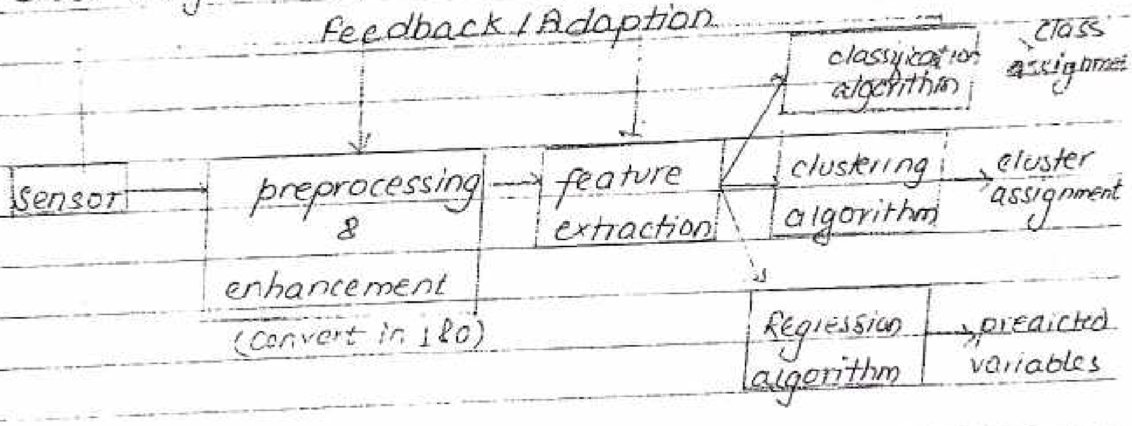


Fig: Pattern recognition System

Approaches to pattern recognition

- statistical
- structural

Basis of	Statistical	Structural
Foundation	Statistical decision Theory	Human perception & cognition.
Description	Quantitative feature fixed number of features semantics from feature position.	Morphological primitives variable number of primitive captures primitive relationship semantic from primitive encoding.
Classification	Statistical classifiers	Passing with syntatic grammar.

- Back propagation
- Neural networks
- classifiers - supervised & unsupervised
- Sampling

from AI

Numerical

Hadamard Transform

1D: $F = H \cdot f$ $\qquad [H^T = H]$

2D: $F = H \cdot f \cdot H^T$

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{2\times 2}$$

positive

$$\begin{array}{c} \quad (1) \qquad\qquad (-1) \\ H = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & +1 \end{array}\right]_{4\times 4} \end{array}$$

$$\begin{array}{c} \quad (+1) \qquad\qquad\qquad (+1) \\ H = (+1)\left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & +1 & 1 & -1 & -1 & +1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & +1 & -1 & 1 & 1 & -1 \end{array}\right](-1) \end{array}$$

For Hadamard

$H = H^T$

Q. 1D:

$f(x) = \{1, 2, 0, 3\}$

Find Hadamard Transform

Sol^n

For 1D: $F = H \cdot f$

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}_{4\times4} \times \begin{bmatrix} 1 \\ 2 \\ 0 \\ 3 \end{bmatrix}$$

$= 1\times1 + 1\times2 + 1\times0 + 1\times3$

$$= \begin{bmatrix} 6 \\ -4 \\ 0 \\ 2 \end{bmatrix}$$ //

Q. 2D

$$f(x) = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 2 \end{bmatrix}_{4\times4}$$

Sol^n

2D: $F = H \cdot T \cdot H^T$

$$= \begin{bmatrix} H \end{bmatrix}_{4\times4} * \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 2 \end{bmatrix} * \begin{bmatrix} H \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 8 & 12 & 8 \\ -2 & 0 & 0 & 0 \\ 0 & -2 & -2 & -2 \\ 0 & -2 & -2 & -2 \end{bmatrix} * \begin{bmatrix} H \end{bmatrix}_{4\times4}$$

$$= \begin{bmatrix} 34 & 2 & -6 & -6 \\ 2 & 2 & 2 & 2 \\ -1 & 2 & 2 & 2 \\ -6 & 2 & 2 & 2 \end{bmatrix}$$

Haar Transform

$$f = \begin{bmatrix} 3 & -1 \\ 6 & 2 \end{bmatrix}$$

Haar Transform for N = 2

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

Now,
Haar Transform can be expressed as:

$T = H \cdot f \cdot H^T$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} * \begin{bmatrix} 3 & -1 \\ 6 & 2 \end{bmatrix} * \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$T = \begin{bmatrix} 5 & 4 \\ 3 & 0 \end{bmatrix}$$