UNIT-1 :-

Introduction to Digital Image Processing :

* Introduction to Image processing :-

Image :- An 'Image' may be defined as "A two dimen-sional function, $f(x, y)$, where x and y are spatial (plane) Coordinates, and the amplitude of 'f' at any pair of Co-ordinates (x, y) is called the intensity or gray level of the image at that point."

Digital Image :- "When x, y and amplitude (intensity) values of 'f' are all finite, discrete quantities" then the image is called digital image.

Digital Image Processing :- "The processing (varying the param-eters) of digital image by means of a digital computer" is called Digital Image Processing.

Pixel :- "A digital is image is composed of a finite no. of elements, each of which has a particular location* and value**". These elements are called pixels (or) pels (or) image elements (or) Picture elements.

"location" in the above definition refered as (x, y) Co-ordinates and value means intensity or amplitude.

Need of Image Processing :- It is motivated by two major factors:

→ Improvement of pictorial information for human perception.

→ Efficient storage and transmission, autonomous machine applications.

P.T.O.

Applications of Digital Image Processing (DIP)

→ Image Sharpening and Restoration: It refers to process images that have been captured from modern cameras to make them better to achieve desired results.

This includes zooming, blurring, sharpening, gray scale to color conversion, edge detection etc.

→ Medical Field:

> Gamma Ray Imaging.
> PET (positron Emission Tomography) Scan.
> X-Ray Imaging.
> Medical CT (Computerized Tomography) Scan.
> UV Imaging.

→ Remote Sensing: To detect infractructure damages caused by an earthquake. The key steps include in the analysis are:
- The extraction of edges.
- Analysis & enhancement of various types of edges.

Remote sensing is mostly done in ~~Sl~~ Satellite Imaging

→ Transmission & Encoding: The very first image transmitted from London to New york using Submarine cable took 3 hours to reach destinition.

But Now a days we are performing live video stream throughout the world. This is because of the type of formats that have been developped.

→ Machine/Robot Vision: One of the major applications of DIP is in Robot Vision. A Robot must see and identify things/objects, Hurdles etc. Most Robots now a days work by following a line called line follower Robots.

→ Color Processing:- It involves processing of coloured images and different color spaces. Ex: RGB, YCbCr, CMYK, HSV etc.
It also involves transmission, storage & encoding of color Images. etc.

Pattern Recognition:- It involves artificial intelligence(AI) Here DIP is used for identifying the objects in an Images and then machine learning(AI) is used for identifying the objects in an images and A.I. is used to train the system for the change in pattern.

Pattern Recognition is used in CAD (Computer Aided Diagnosis), Recognition of Handwriting, Images etc.

Video Processing:- A video is nothing but the fasted movement of pictures. The quality of a video depends on number of pictures (frames) per second and quality of each frame being used.

It involves noise reduction, detail enhancement, motion Detection, frame rate conversion, Aspect Rate Conversion, Color Space Conversion etc.

---

* Fundamental Steps in Digital Image Processing:-

1) Image Acquisition:- It could be as simple as being given an image that is already in digital form. Generally this stage involves prepocessing, such as Scaling.

2) Image Enhancement:- It is the simplest and most appealing areas of DIP. The basic idea behind this technique is to bring out the detail that is obscured (or) simply to Highlight certain features of Interest in an Image. Increasing the contrast of the image to look it better is one example for enhancement. It is a subjective process.

3) Image Restoration:- It is an area that also deals with improving the appearance of an image. It is an objective process that is based on mathematical or probabilistic models of image degradation.

P.T.O.

4) Color Image Processing :- It is an area that has been gaining in importance because of the significant increase in the use of digital images over the internet.

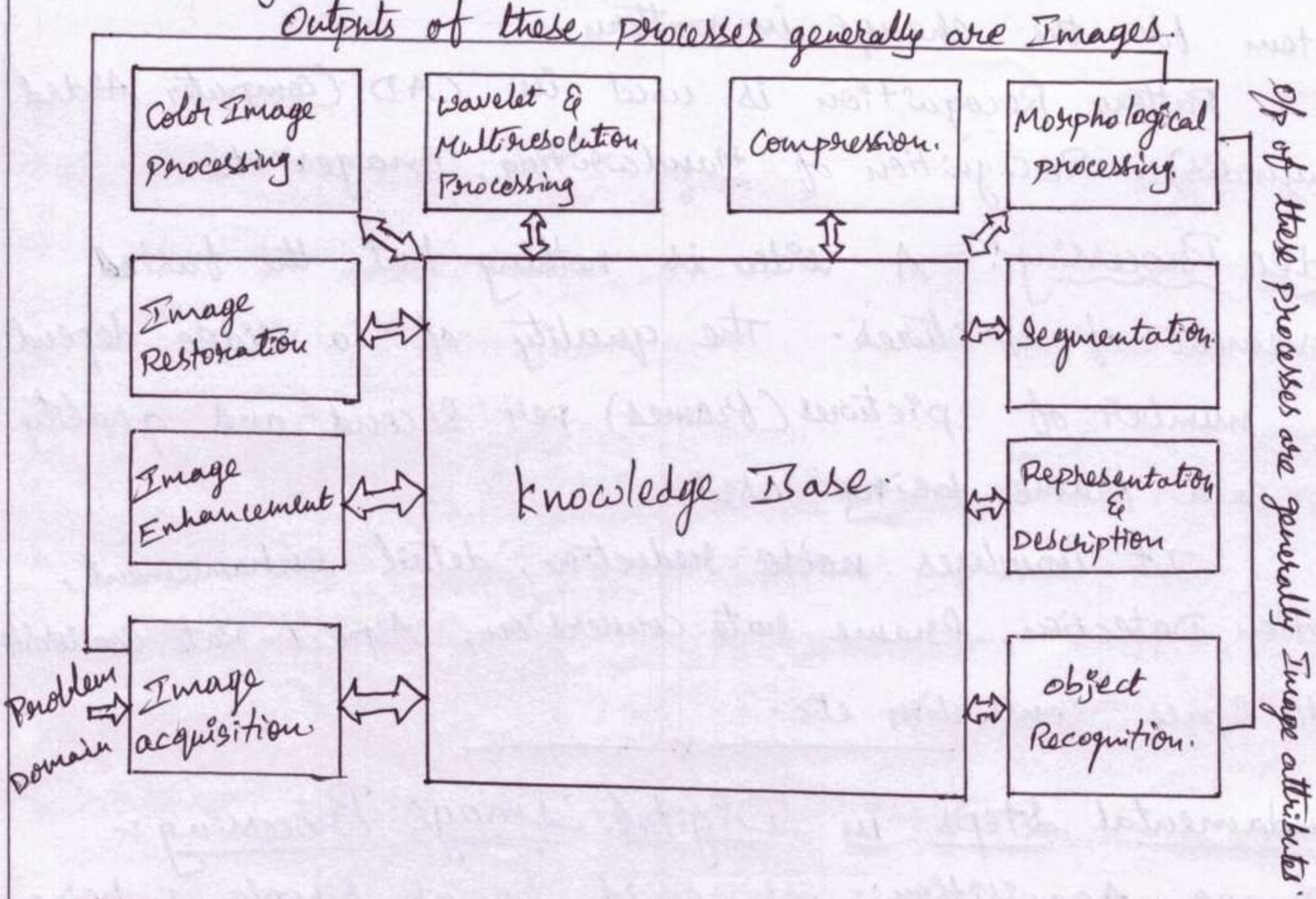


Fig:- Fundamental steps in Digital Image Processing.

5) Wavelets :- These are the fundation for representing images in various degrees of resolution. This is used for image data compression and for pyramidal representation, in which images are sub divided into successive smaller regions.

6) Compression :- It deals with the technique for reducing the storage required to save an image, & the bandwidth required to transmit it. Mostly JPEG image compression standard is used in ~~may~~ many applications.

P.T.O.

⑦ Morphological Processing:- It deals with tools for extracting image components that are useful in the representation and description of shape.

⑧ Segmentation:- It means partitioning an image into its constituent parts or objects. Atonomous segmentation is one of the most difficult tasks in DIP.

A rugged segmentation procedure brings the process a long way towards successful solution of imaging problems that require objects to be identified individually.

On the other hand, weak segmentation algorithms almost always guarantee eventual failure.

The more accurate the segmentation, the more likely recognition is to succeed.

⑨ Representation and Description:- It always follows the output of a segmentation stage, usually a raw pixel data, constituting either the boundary of a region (or) all points within the region.

In both cases it is necessary to convert (represent) the data to a form suitable for processing by computer.

Representations → Boundary
Representations → Region

→ Boundary Representation is used when the focus is on external shape characteristics (corners, inflections).

→ Regional Representation is used when the focus is on internal properties (texture or skeletal shape).

Choosing representation is only a part of the solution. Description, also called feature selection, deals with extraction of attributes results in some quantitative information.

(10) Recognition:- It is the process that assigns a label to an object based on its descriptors.

We conclude our coverage of DIP with development of methods for recognition of individual objects.

x

Representing Digital Images:- Assume that an image $f(x,y)$ is sampled so that the resulting digital image has M rows and N colums. The values of coordinates (x,y) become discrete quantities (as shown below).

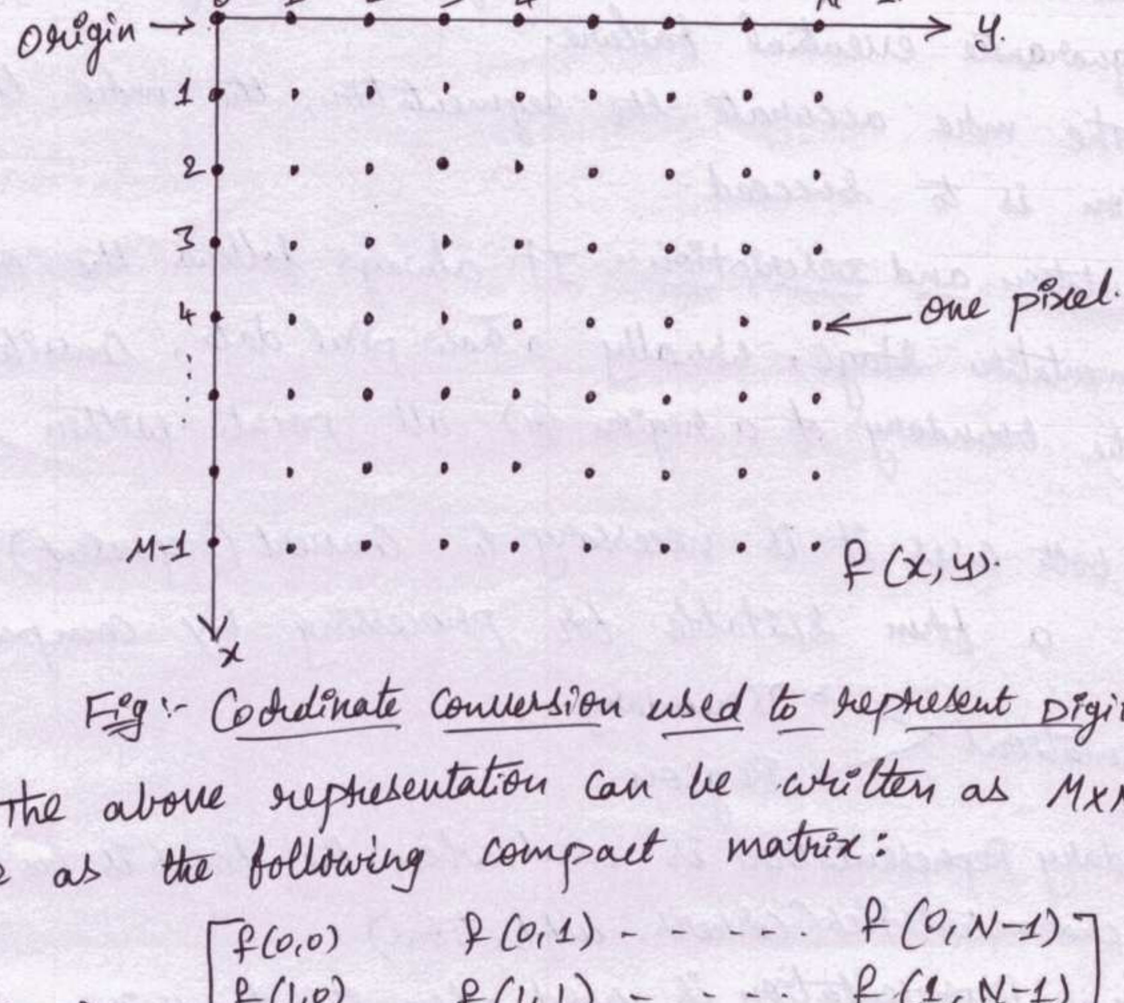


Fig:- Coordinate conversion used to represent Digital Image

The above representation can be written as $M \times N$ digital image as the following compact matrix:

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

The right sige side of the above equation is a digital image.

x

* Components of an Image Processing System:-

The following block diagram shows different components that are used in general-purpose image processing system.

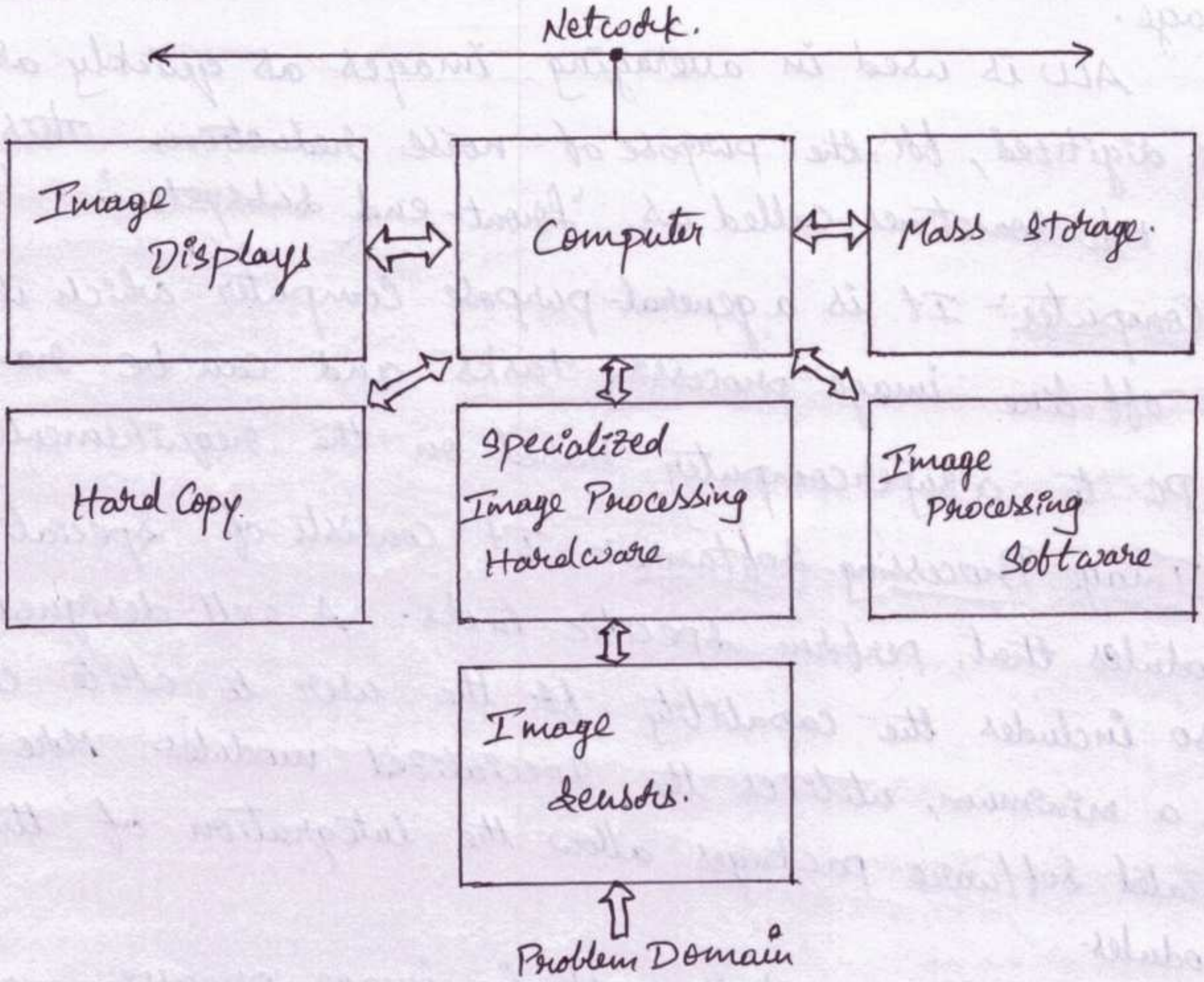


Fig:- Components of general-purpose Image Processing System:

1) Image Sensing:- Two elements are required to acquire digital images:

a) Physical device that is sensitive to energy radiated by the object we wish to image.

b) A digitizer that is used for converting the output of the physical sensing device into digital form.

For instance, in a digital video camera, the ~~sen~~ sensor produces an electrical o/p proportional to light intensity. The digitizer converts the o/p's to a digital data.

2) Specialized Image Processing Hardware :- It consists of a digiti-
-zer that is discussed before and an additional hardware
that performs other premitive operations, such as an ALU, which
performs Arithmetic and Logic operations in parallel on entire
image.

ALU is used in averaging images as quickly as they
are digitized, for the purpose of noise reduction. This type
of H/w sometimes called as 'front-end subsystem'.

3) Computer :- It is a general-purpose computer which is suitable
for off-line image processing tasks and can be range from
a PC to a supercomputer based on the requirements.

4) Image Processing Software :- It consists of specialized
modules that perform specific tasks. A well-designed package
also includes the capability for the user to write code that,
as a minimum, utilizes the specialized modules. More sophis-
-ticated software packages allow the integration of those
modules.

5) Mass Storage :- It is must in image processing application.
A image of size 1024 X 1024 pixels, in which intensity of
each pixel is 8-bits requires 1 mega byte of storage
space, if the image is not compressed.

when dealing with huge number of images, providi-
-ng adequate storage in an image processing system is a challenge.
Digital storage for DIP falls into 3 principle categories:

i) short term storage for use during processing.
ii) on-line storage for relatively fast recall. and
iii) Archival storage, used for infrequent access.

P.T.O.

6) <u>Image Displays</u>:- These are mainly color TV monitors. These are driven by the outputs of image and graphic display cards that are integral part of Computer Systems.

7) <u>Hardcopy</u>:- These devices are used for recording images include Laser printers, film Cameras, Inkjet units, digital units, such as optical and CD-ROM disks etc.

Films provide highest possible resolution, but paper is the obvious medium of choice for written material.

8) <u>Networking</u>:- It is a default function in any computer in use today. Because of larger amount of data inherent in in image processing applications, the key consideration in transmission is bandwidth. In dedicated N/w it is not a problem. But it is necessary to use optical fiber and other broadcast technologies for communication via Internet to remote sites.

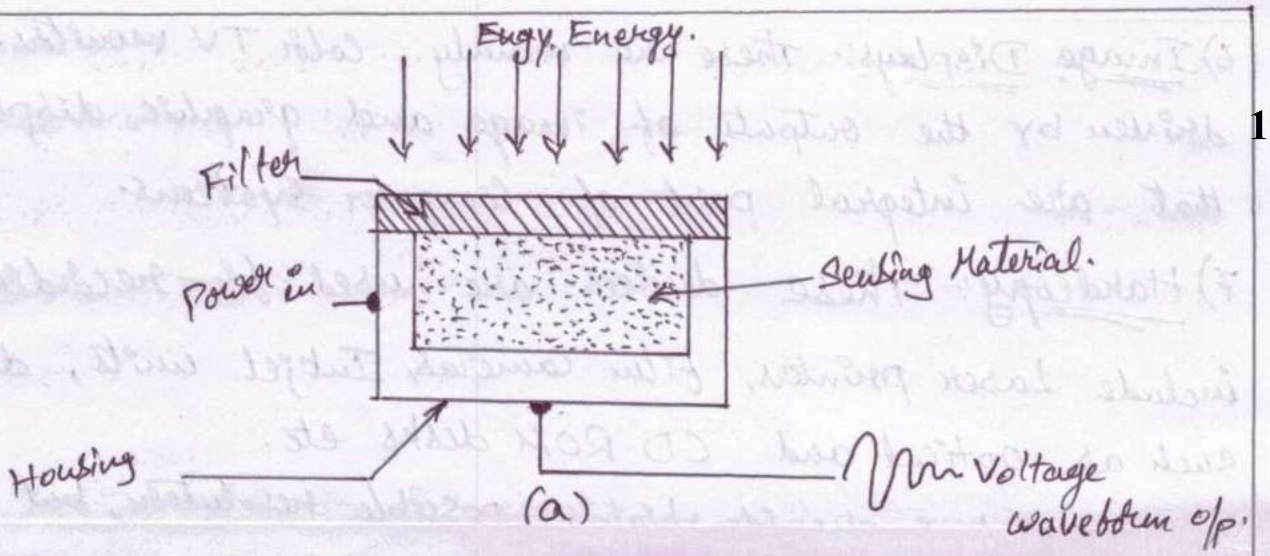
———————— x ————————

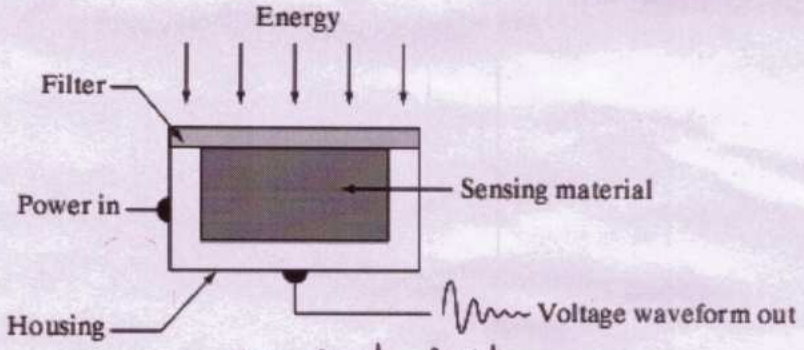
* <u>Image Sensing and Acquition</u>:- The types of images in which we are interested are generated by the combination of an "illumination" source and the reflection or absorption of energy from that source by the elements of the "scene" being imged.

The below figure shows the three principal sensor arrangements used to transform illumination energy into digital images.

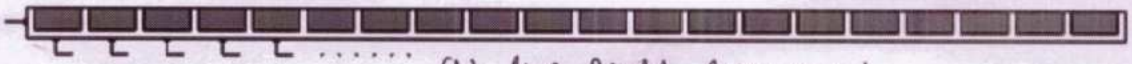
The idea is simple, the incoming energy is transformed into a voltage by combining i/p electrical power & sensor material response.

The o/p of the sensor is then digitalized by using a digitizer.

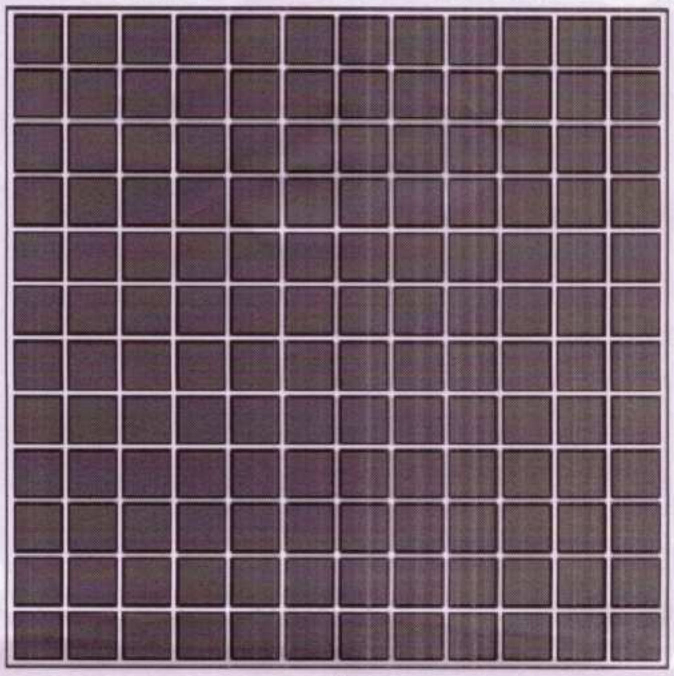




(a) Single Sensor.



(b) line Sensor (strip sensor)



(c) Array Sensor.

Figure(1):

1) Image Acquisition using Single Sensor:- Fig(a) shows the Components of image (Single) Sensor. Eg:- Photodiode. The use of filter improves selectivity. If a green filter on light sensor favors light in green band.

contd.

In order to generate a 2-D image using single sensor, there has to be relative displacement in both x-axis & y-axis between the sensor & area to be imaged.

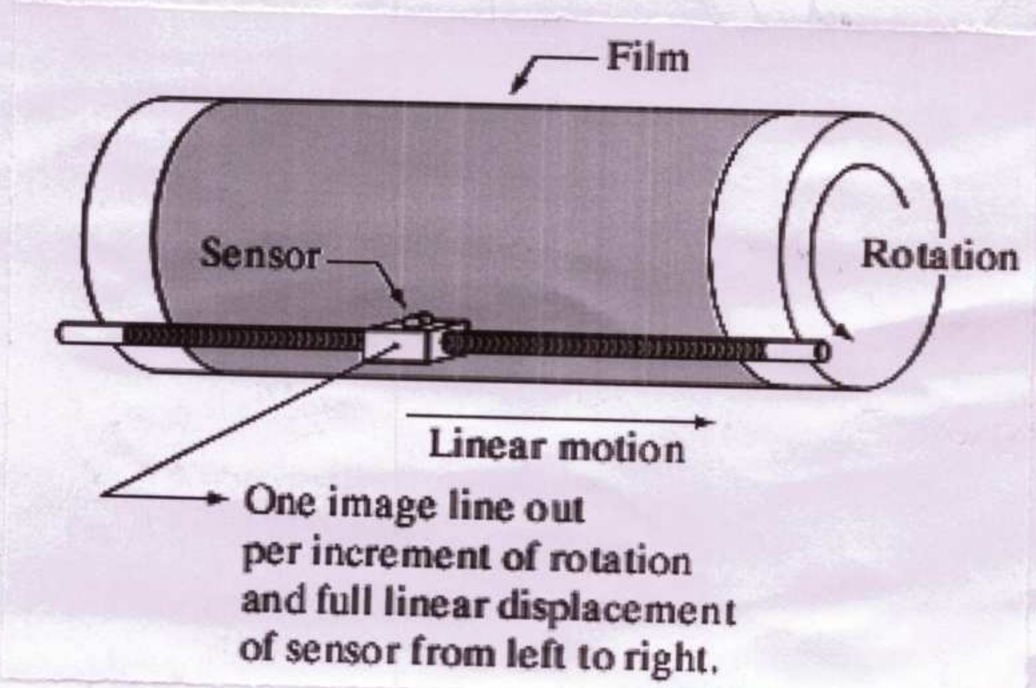
To perform this the arrangement is done as shown in below figure.

The film negative (image to be scanned) is mounted on a drum whose rotation makes one dimentional displacement.

The single sensor is mounted on a lead screw which provides displacement in another dimention.

Such arrangement provides image capturing line by line. This is a slow and inexpensive method of obtaining High-resolution Images.

Also we can replace the drum with flat bed to perform "to and fro" motion.



2) Image Acquition using Sensor strip:- Here it consists of an in-line sensor arrangement ~~of~~ in the form of a sensor strip as shown in fig 1(b).

The strip provides imaging in one dimention and its motion in perpendicular direction provides imaging in other dimension.

Figure below shows the arrangement of sensor strip which is mostly used in flat bed scanner.

Sensing devices with 4000 or more in-line sensors are possibly used.

It gives one line of image at a time and its ⊥r motion makes 2-D scanning.

Sensor strips mounted in a ring configuration are frequently used in medical applications which gives cross-sectional images of 3-D objects as shown below.

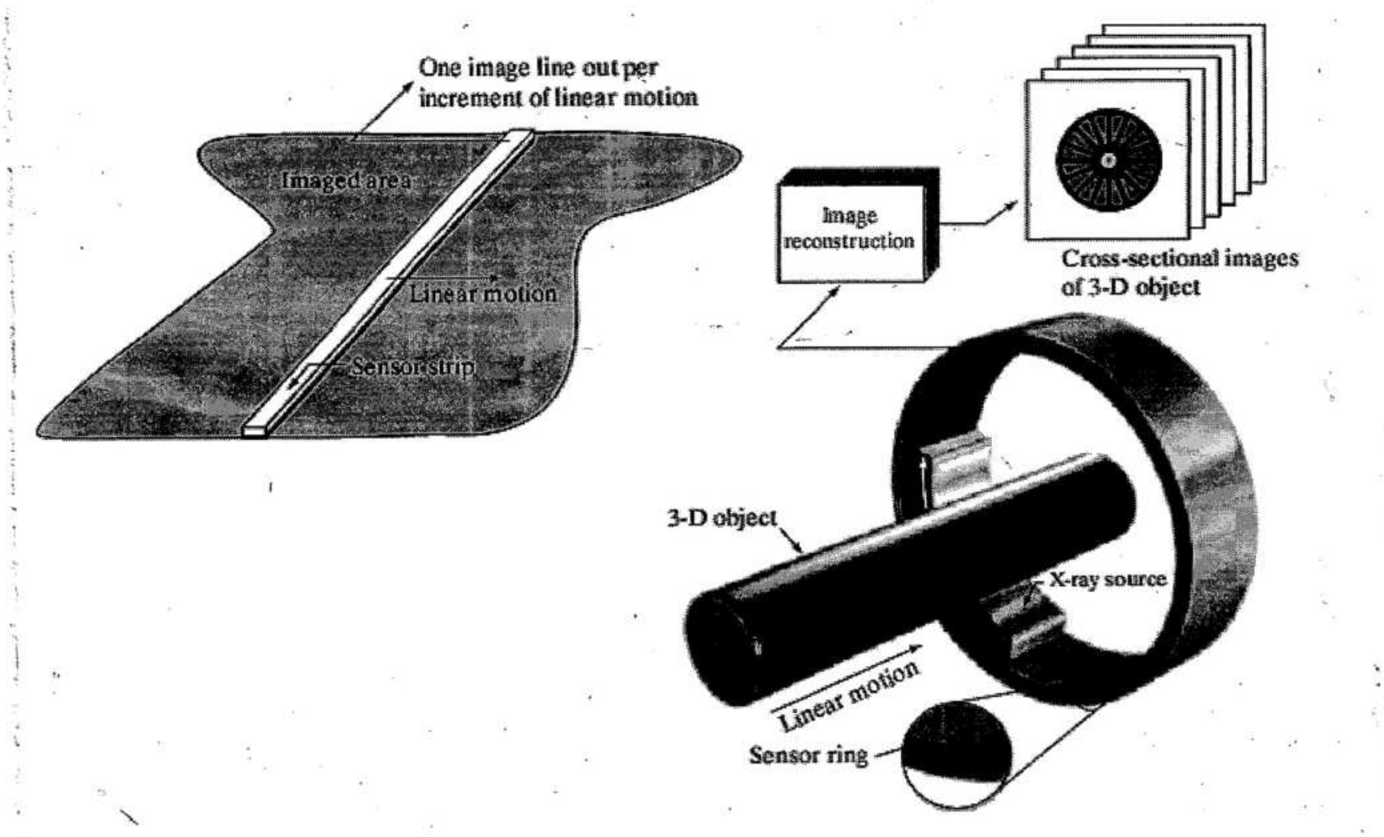


Fig:- Image Acquisition using Line Sensor

Cont...

3) Image Acquition using Sensor Arrays:- Fig 1(c) Shows individual sensors are arranges in 2-D array.

This is also the predominant arrangement found in digital cameras, electromagnetic & ultrasonic sensing devicess.

A typical sensor for these array arrangement is CCD array, that contains a packaged rugged arrays of 4000×4000 elements or more. (CCD - Charged Coupled Devices).

The principal manner in which array sensors are used is shown below.

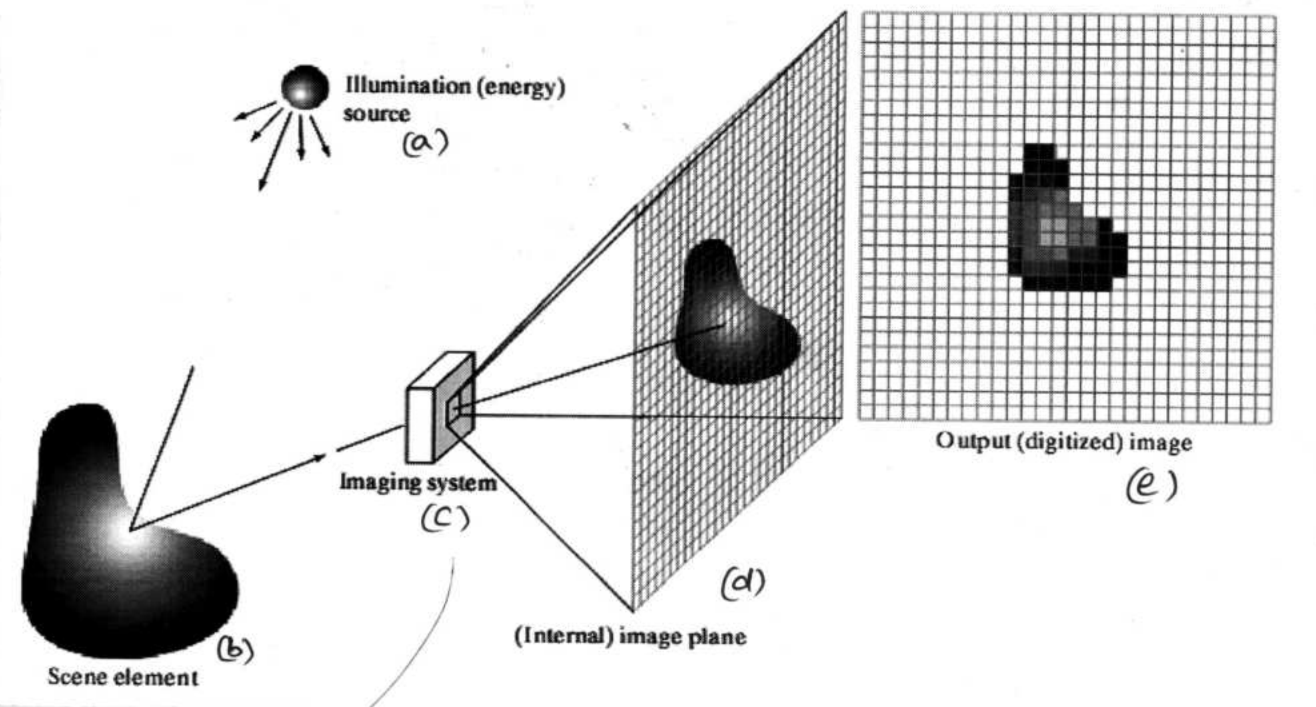


fig : Image Acquisition using Sensor Array:

→ The fig. shows the energy from illuminating source is reflected by scene element.

→ The imaging system collects the incoming energy from scene element and focus it on to an image plane.

→ The sensor array (which is coincident with the focal plane) produces outputs proportional to the integral of light received at each sensor.

→ The digitizer ckt then converts these outputs into digital form results in a digital image (shown in 'e').

⑦

* Image Sampling and Quantization:- The o/p of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed.

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

This involves two processes: 1) Sampling
2) Quantization.

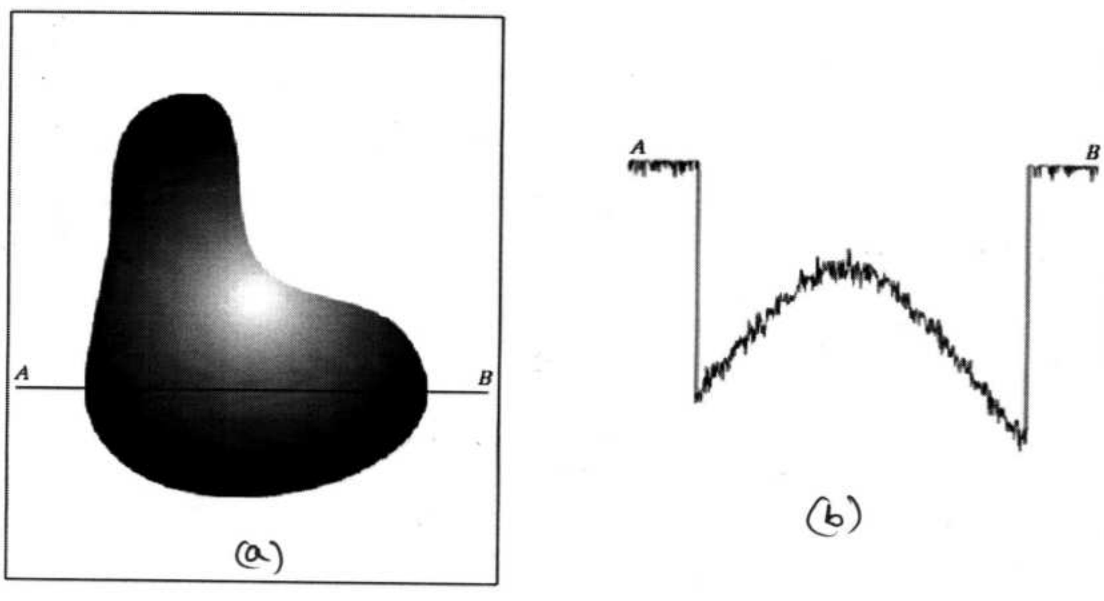
→ Digitizing the coordinate values is called Sampling.
→ Digitizing the amplitude (intensity) values is called quantization.

• Let us consider a continuous image, $f(x, y)$ whose x, y coordinates and amplitude are continuous as shown in fig(a).

• To convert the image into digital form, we have to sample the function in both the amplitude & coordinates.

• As per the definitions of sampling & quantization, we perform the operations on the image in fig(a).

• The 1-D function in fig(b) is a plot of amplitude (gray level) values of the continuous image along the line segment A↔B in fig(a).



(a) (b)

Fig:- Generation of Digital Image:

Cont...

- To sample this ~~a~~ function, we take equally spaced samples along line AB as shown in fig(c).
- The location of each sample is given by a tick mark in bottom part of the figure.
- The samples are shown as small white squares superimposed on the function. The set of these discrete location gives sampled function
- The values of the samples are still a continuous gray level values.
- In order to form a digital function, the gray-level values must be converted (quantized) into discrete quantities.
- The right side of fig(c) shows eight discrete gray-levels ranging from black to white.
- These gray-levels are assigned to each corresponding sample to obtain the quantized value.
- The digital samples obtained in sampling and quantization is represented in fig(d).
- Starting at the top of the image and carrying out this procedure line by line produces a digital image.

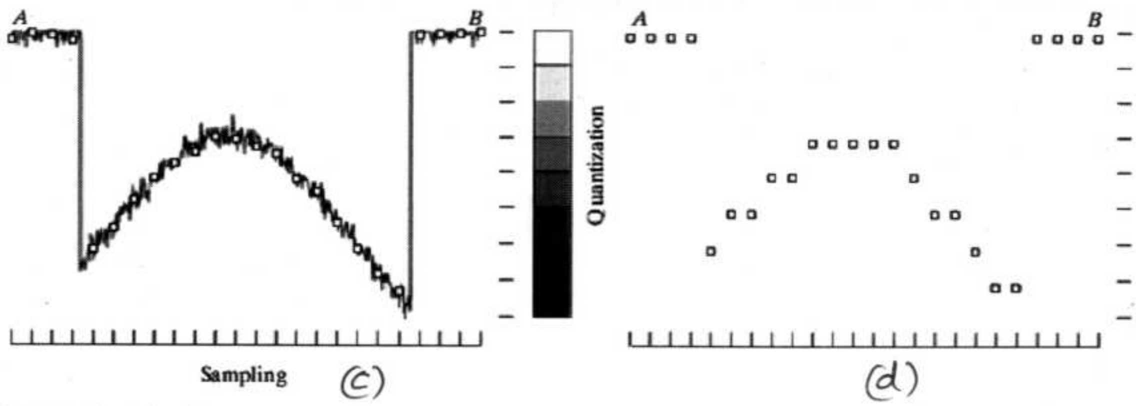


fig:- Generation of Digital Image:

- when a sensing array is used for image acquisition, there is no motion & the no. of sensors in array establishes the limits of sampling in both directions.
- The figure(2a) shows continuous image projected onto the plane ⑧

of array sensor and fig (2b) shows the image after sampling and quantization.

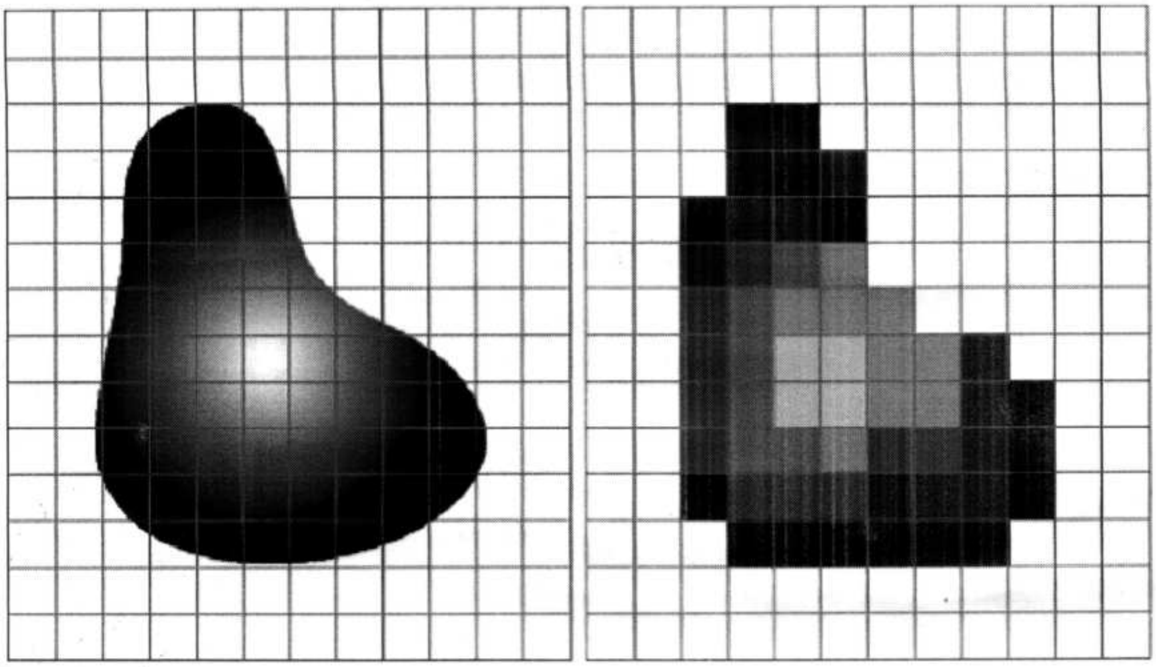


Fig 2:- (a) Continuous Image (b) Digital Image.

• Clearly, the quality of a digital image is determined to a large degree by no. of samples and discrete gray levels used in sampling & quantization

---

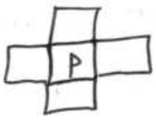
* Some basic Relationships between Pixels:-

Neighbors of a Pixel:-

• A pixel 'p' at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by $(x+1, y)$, $(x-1, y)$ $(x, y+1)$, $(x, y-1)$. This set of pixels called the 4 neighbors of 'p' and is denoted by $N_4(P)$.

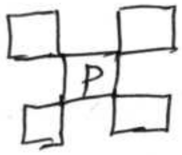
• Each pixel is a unit distance from (x, y).

• Some Neighbors of p lie outside the digital image if (x, y) is on the border of the image.

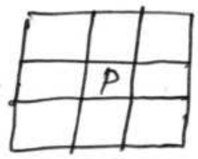


$N_4(P)$ Set:

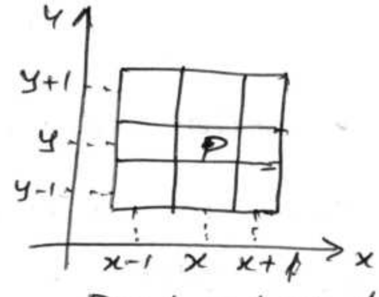
- The four diagonal neighbors of 'p' have coordinates $(x+1, y+1)$ $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ and are denoted by $N_D(P)$.
- These points, together with the 4-neighbors, are called the 8-neighbors of 'P', and is denoted by $N_8(P)$.



$N_D(P)$



$N_8(P)$.



Pixel coordinates.

- Some of the points in $N_D(P)$ & $N_8(P)$ fall outside the image if (x, y) is on the border of image.

Adjacency, Connectivity, Regions and Boundaries:

- Connectivity b/w pixels is a fundamental concept the simplifies the definitions of such as regions and boundaries
- Two neighbor pixels satisfy a specified criterion of similarity in their gray levels (equal) then they are said to be "Connected."
- Let 'V' be the set of gray-level values used to define adjacency.
- In a binary image, $V = \{1\}$ if ~~we~~ we are referring to adjacency of pixels with value '1'.
- Same idea in the grayscale image, but here 'V' contains more elements in b/w 0 – 255.
- Three types of adjacency:

a) 4-adjacency: 2 pixels 'p' and 'q' with values from 'V' are 4-adjacent if 'q' is in the set $N_4(P)$.

(b) 8-adjacency: 2-pixels 'p' and 'q' with values from 'V' are 8-adjacent if 'q' is in the set $N_8(P)$.

(c) m-adjacency (mixed): Two pixels 'P' & 'q' ~~are~~ with values from 'V' are m-adjacent if,

i) 'q' is in $N_4(P)$, or

ii) 'q' is in $N_D(P)$ & the set $N_4(P) \cap N_4(q)$ has no pixel whose values are from 'V'.

- mixed-adjacency is a modification of 8-adjacency which is introduced to eliminate ambiguities that often arise with 8-adjacency.
- Consider the following pixel arrangement to show the ambiguity of 8-adjacency and its elimination is m-adjacency.

$$\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Binary Image Pixel arrangement 8-adjacency m-adjacency.

- Let 'R' be a subset of pixels in an image.
- If 'R' is a connected set the 'R' is said to be "Region".
- If one or more neighbors in the set 'R' are not connected, then it is said to be "boundary" of the Region 'R'.

Distance Measures:- For pixels P, q & z, with coordinates (x, y), (s, t), (v, w), respectively, 'D' is a distance function or metric if

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

(b) $D(P,q) = D(q,P)$ &

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

Cont...

- The "Euclidean Distance" between P & q,

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

- D_4 distance (city-block distance) between P & q,

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

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

Eg: $\underline{D_4(P,q) \leq 2}$

- D_8 distance (chess board Distance) b/w P & q,

$$D_8(P,q) = \max\left(|x-s|, |y-t|\right)$$

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

Eg :- $\underline{D_8(P,q) \leq 2}$

o

* Introduction to the mathematical tools used in DIP:-

I. Array verses Matrix Operations:-

- Consider two 2X2 images: $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ & $\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$.

- Array Product : $\begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$.

- Matrix Product : $\begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix}$.

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

II. Linear verses Non-Linear Operations:-

- General operator H, that produces an o/p image $g(x,y)$ for a given I/p image $f(x,y)$: $H[f(x,y)] = g(x,y)$.
- 'H' is said to be linear operator if,

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

where, a_i, a_j are arbitrary constants.

& $f_i(x,y), f_j(x,y)$ are two images of same size.

- If 'H' is a sum operator Σ:

$$\Sigma(a_1 f_1(x,y) + a_2 f_2(x,y)) = \Sigma a_1 f_1(x,y) + \Sigma a_2 f_2(x,y)$$
$$= a_1 \Sigma f_1(x,y) + a_2 \Sigma f_2(x,y)$$
$$= a_1 g_1(x,y) + a_2 g_2(x,y)$$

Here, 'Σ' is linear operator.

Cont...

- Consider "max" operation,

  let, $f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}$, $f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$, $a_1 = 1$ & $a_2 = -1$.

- Now,

  LHS: $\max \left\{ (1) \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1) \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} = \max \left\{ \begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix} \right\} = -2$

  RHS: $(1) \max \left\{ \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \right\} + (-1) \max \left\{ \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} = 3 - 7 = -4$

  $\therefore$ LHS $\neq$ RHS.

  $\therefore$ "Max" is non-linear operation.

III - <u>Arithmetic Operations</u>:-

- Arithmetic operations are array operations that are carried out b/w corresponding pixels.
- Four Arithmetic operations:

  i) Sum — $s(x,y) = f(x,y) + g(x,y)$.
  ii) Difference — $d(x,y) = f(x,y) - g(x,y)$
  iii) Product — $P(x,y) = f(x,y) * g(x,y)$
  iv) Division — $v(x,y) = f(x,y) / g(x,y)$.

  where, $x = 0,1,2 \ldots M-1$ & $y = 0,1,2 \ldots N-1$.

  All images are of same size $M \times N$.

IV - <u>Set Operations</u>:- Basic set operations are Union, Intersection, Compliment, etc.

- Let 'A' is a set composet of ordered pairs of real Numbers.
- If a pixel $a = (x,y)$ is an element in 'A'; $a \in A$.
- If 'a' is not in 'A'; $a \notin A$.
- If 'A' has no elements then is a "null set" (ϕ).
- Every element of 'A' are in 'B' then, $A \subseteq B$.

Cont...

- Union of two sets : $C = A \cup B$.
- Intersection of two sets : $D = A \cap B$.
- If two sets A & B are disjoint or mutually exclusive then,

$$A \cap B = \phi$$

- Complement of a set 'A' is the set of elements not in A,

$$A^c = A' = \{ \omega \mid \omega \notin A \}.$$

- Difference of two sets A & B, $A - B = \{ \omega \mid \omega \in A, \omega \notin B \} = A \cap B^c$

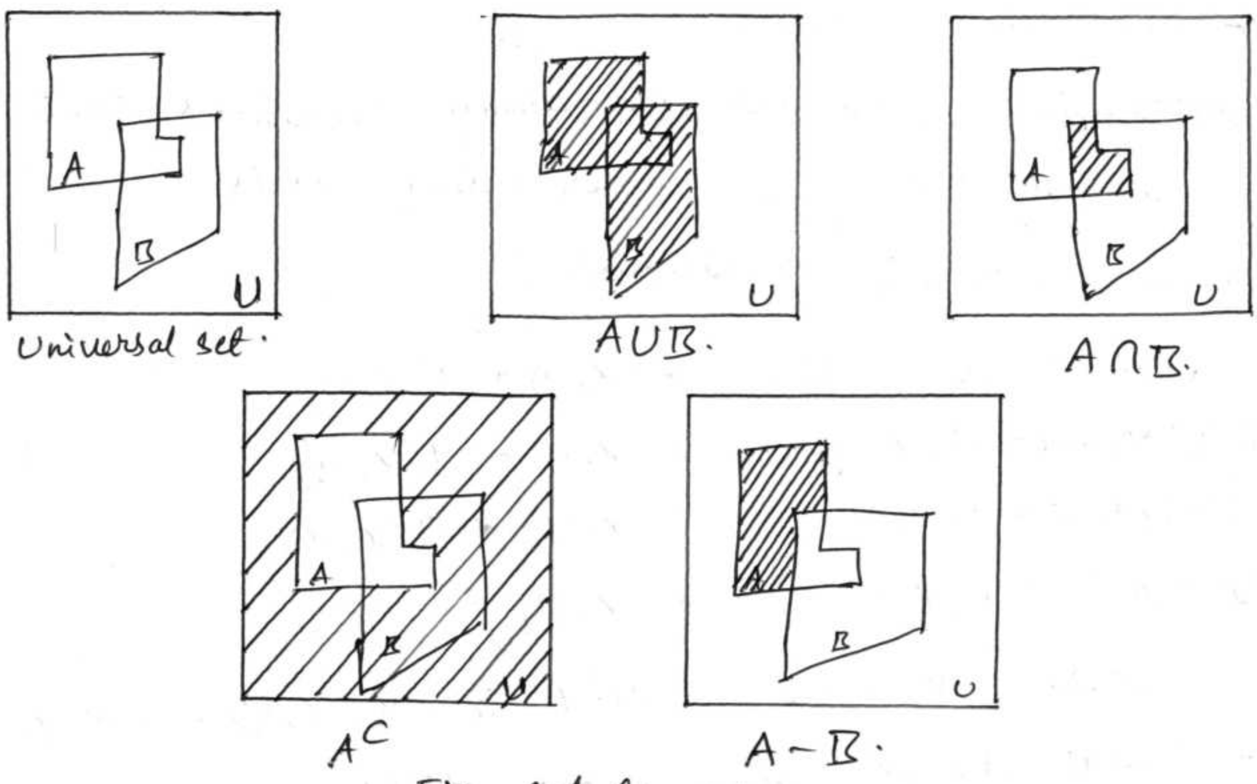


Fig:- Set Concept

IV - Logical Operations :-

- Logical operations like AND, OR, NOT, NAND, XOR are performed on Images at pixel level.
- the Illustrations are shown in the below figures.
- Consider ~~a~~ two images A & B, the operations are ~~per~~ performed at Pixel to pixel level.

Contd...

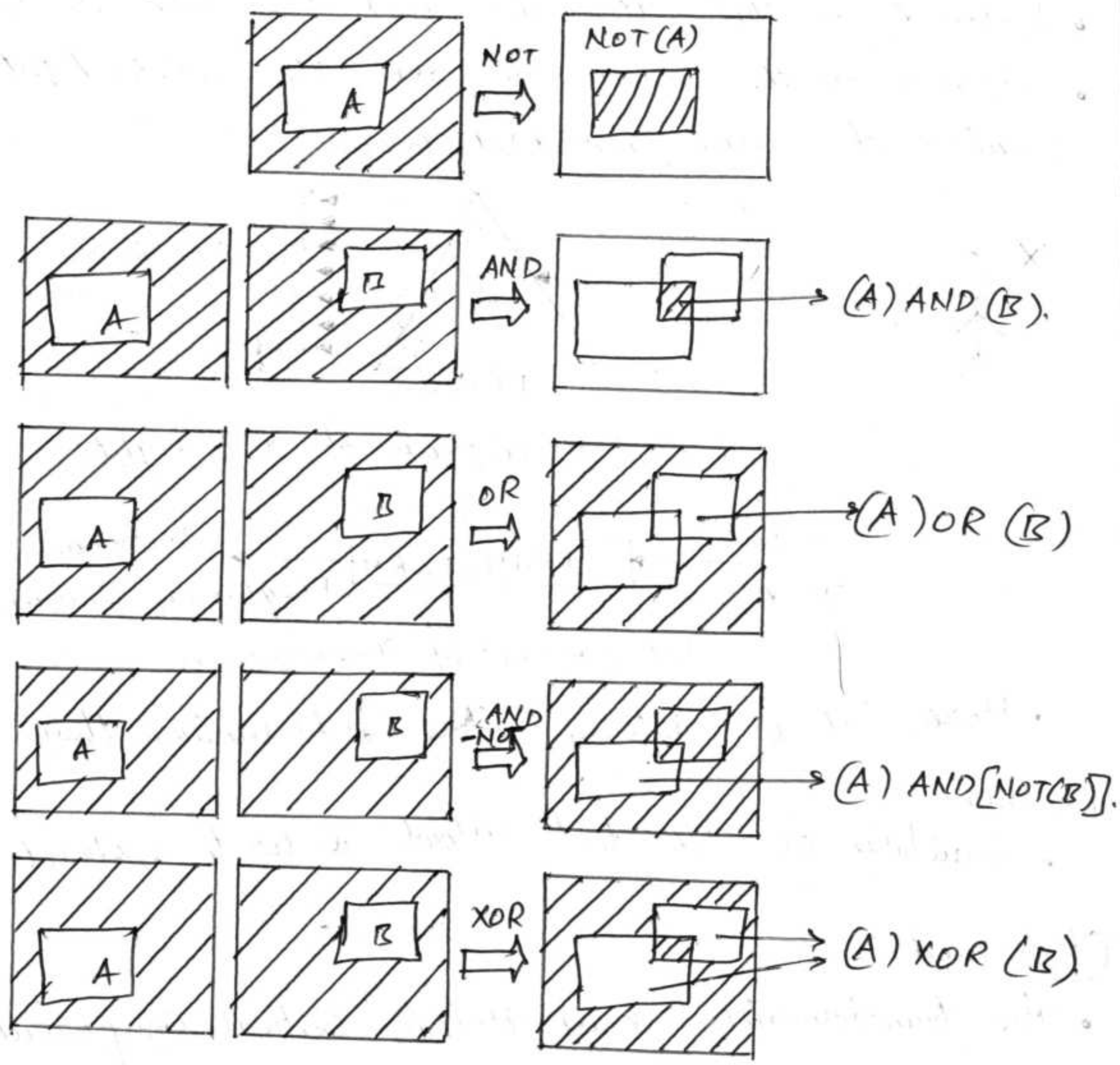


Fig:- Logic Operations on Image:

---

* Need for Image Transform:-

"Transform" is basically a mathematical tool to represent a signal.

The need for transforms is as follows:

i) Mathematical Convenience: Every action in time domain will have an impact in frequency domain.

Convolution in Time Domain $\Longleftrightarrow$ Multiplication in frequency Domain.

Convolution in Time Domain → Complex.

Multiplication in frequency Domain → Simple.

ii) To Extract more Information:- It allow us to extract more relevant information.

- To illustrate this, Consider the example of Prism experiment. (12)

- Person 'X' on the left side sees the light as white light;
- Person 'Y' on the right side sees the white light as a combination of seven colors (VIBGYOR).

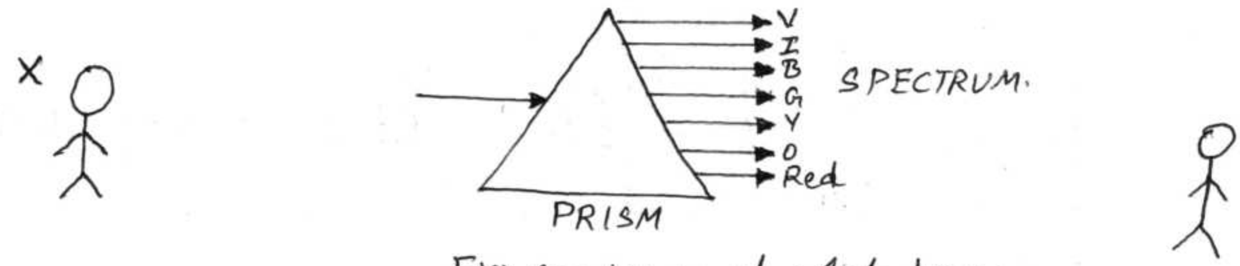


Fig:- spectrum of white light:

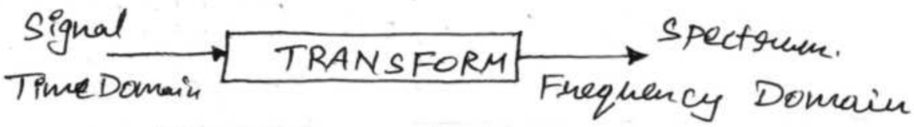


Fig:- Concept of Transform:

- Here, 'Y' is getting more information than 'X', by using prism.
- Similarly the T/f tool allows us to extract more information

iii) Other needs:-

- The transformation may isolate critical components of the image pattern, so that they are directly accessible to analysis.
- The transformation may place the image data in a more compact form so that they can be stored & transmitted efficiently.

---o---

➤ Classification of Image Transforms:- Image transforms can be classified on the nature of the "basis Function".

i) Transforms with orthogonal basis function.
ii) Transforms with non-sinusoidal orthogonal basis function.
iii) Transforms whose basis functions depend on the statistics of the input data.
iv) Directional Transformation.

Cont...

IMAGE TRANSFORMS

- Orthogonal Sinusoidal Basis Function
  - Fourier Transform
  - Discrete Cosine Transform
  - Discrete Sine Transform
- Non-Sinusoidal Orthogonal Basis function
  - Haar Transform
  - Walsh Transform
  - Hadamard Transform
  - Slant Transform
- Basis function Depending on Statistics of i/p Function
  - KL Transform
  - Singular Value Decomposition (SVD)
- Directional Transformation
  - Hough Transform
  - Radon Transform
  - Ridgelet Transform
  - Contourlet Transform

Fig:- Classification of Image Transforms

o

* **Discrete Fourier Transform :-** Fourier Transform is widely used in the field of image processing. An image is a spatially varying function. A fourier T/f is used to transform an intensity image into the domain of spatial frequency.

The discrete fourier t/f of a finite duration sequence $x(n)$ is defined as,

$$DFT(x(n)) : \boxed{X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} \quad k = 0,1,2 \cdots (N-1)}$$

Unitary Transform:- A discrete linear transform is unitary ~~its~~ if its transform matrix conforms the unitary condition,

i.e., $A \times A^H = I$ where, $A \rightarrow$ Transformation matrix

$A^H \rightarrow$ Hermitian matrix.

$A^H = (A^*)^T$

Problem :- Check whether DFT matrix is unitary or Not?

Sol :- Step 1: Determine Matrix 'A':

Finding 4-Point DFT ($N=4$)

$$X(k) = \sum_{n=0}^{4-1} x(n)\, e^{-j\frac{2\pi}{4}kn}$$

where, $k = 0,1,2 \cdots (4-1)$

$$X(0) = \sum_{n=0}^{3} x(n)\, e^{0}$$

$$\boxed{\therefore X(0) = x(0) + x(1) + x(2) + x(3).}$$

$$X(1) = \sum_{n=0}^{3} x(n)\, e^{-j\frac{2\pi}{4}n} = \sum_{n=0}^{3} x(n)\, e^{-j\frac{\pi}{2}n}$$

$$X(1) = x(0)e^{-j\frac{\pi}{2}(0)} + x(1)e^{-j\frac{\pi}{2}(1)} + x(2)e^{-j\frac{\pi}{2}(2)} + x(3)e^{-j\frac{\pi}{2}(3)}.$$

$$\boxed{\therefore X(1) = x(0) - j\,x(1) - x(2) + j\,x(3).}$$

$$\left[\because e^{j\theta} = \cos\theta + j\sin\theta\right]$$

$$X(2)=\sum_{n=0}^{3}x(n)\,e^{\frac{-j2\pi}{4}(2)n}=\sum_{n=0}^{3}x(n)\,e^{-j\pi n}$$

$$=x(0)e^{0}+x(1)e^{-j\pi}+x(2)e^{-j2\pi}+x(3)e^{-j3\pi}$$

$$\boxed{\therefore X(2)=x(0)-x(1)+x(2)-x(3)}$$

$$X(3)=\sum_{n=0}^{3}x(n)\,e^{-j\frac{3\pi}{2}n}.$$

$$=x(0)+x(1)e^{-j\frac{3\pi}{2}}+x(2)e^{-j3\pi}+x(3)e^{-j\frac{9\pi}{2}}.$$

$$\boxed{\therefore X(3)=x(0)+jx(1)-x(2)-jx(3)}$$

Now by collecting the coefficients of $x(0)$, $x(1)$, $x(2)$ & $x(3)$, we get A!

$$X(k)=A=\begin{bmatrix}1&1&1&1\\1&-j&-1&j\\1&-1&1&-1\\1&j&-1&-j\end{bmatrix}$$

Step-2: Computing A^H:

$$A^H=(A^*)^T \quad \text{i.e.,} \quad A \xrightarrow{\text{Conjugate}} A^* \xrightarrow{\text{Transpose}} A^H$$

Now,
$$A^*=\begin{bmatrix}1&1&1&1\\1&j&-1&-j\\1&-1&1&-1\\1&-j&-1&j\end{bmatrix}.$$

$$\therefore A^H=[A^*]^T=\begin{bmatrix}1&1&1&1\\1&j&-1&-j\\1&-1&1&-1\\1&-j&-1&j\end{bmatrix}$$

Step: 3:- Determining $A\times A^H$.

$$\Rightarrow A\times A^H=\begin{bmatrix}1&1&1&1\\1&-j&-1&j\\1&-1&1&-1\\1&j&-1&-j\end{bmatrix}\times\begin{bmatrix}1&1&1&1\\1&j&-1&-j\\1&-1&1&-1\\1&-j&-1&j\end{bmatrix}=\begin{bmatrix}1+1+1+1&1+j-1-j&1-1+1-1&1-j-1+j\\1-j-1+j&1+1+1+1&1+j-1-j&1-1+1-1\\1-1+1-1&1-j-1+j&1+1+1+1&1+j-1-j\\1+j-1-j&1-1+1-1&1-j-1+j&1+1+1+1\end{bmatrix}$$

$$=\begin{bmatrix}4&0&0&0\\0&4&0&0\\0&0&4&0\\0&0&0&4\end{bmatrix}=4\times\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\Rightarrow A\times A^H=4\times I.$$

$\therefore$ The resultant is a identity matrix that shows DFT ~~satisfy~~ satisfies unitary condition.

2D - DFT :- The 2D-DFT of a rectangular image $f(m,n)$ of size $M \times N$ is represented as $F(k,l)$.

$$f(m,n) \xrightarrow{\text{2D-DFT}} F(k,l).$$

where,
$$F(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{M}mk} e^{-j\frac{2\pi}{N}nl}$$

For a square image $f(m,n)$ of size $N \times N$, the 2D DFT is defined as

$$F(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}mk} e^{-j\frac{2\pi}{N}nl}$$

The inverse 2D-DFT,

$$f(m,n) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l)\, e^{j\frac{2\pi}{N}mk} e^{j\frac{2\pi}{N}nl}$$

The Fourier Transform $F(k,l)$ is given by,

$$F(k,l) = \underbrace{R(k,l)}_{\text{Real part}} + j\, \underbrace{I(k,l)}_{\text{Imaginary part}}$$

Then, the polar form of $F(k,l) = |F(k,l)|\, e^{j\,\psi(k,l)}$

where, $|F(k,l)| = \sqrt{R^2\{F(k,l)\} + I^2\{F(k,l)\}}$ → magnitude spectrum.

& $\psi(k,l) = \tan^{-1} \dfrac{I\{F(k,l)\}}{R\{F(k,l)\}}$ → Phase angle (or) Phase spectrum.

Note:- The efficient computational form of DFT is FFT which can be represented in two ways:

a) Standard Representation:

High frequencies are grouped at center & low frequencies are at edges.

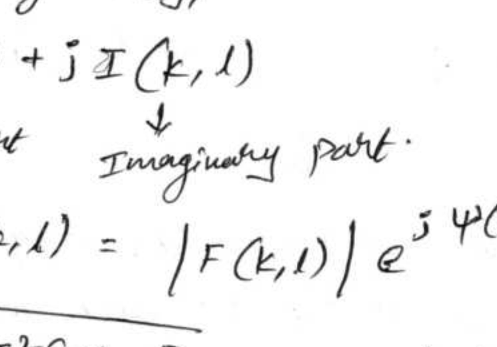


Fig (a)

b) Optical Representation:

Low frequencies are grouped at center & High frequencies are located at edges.

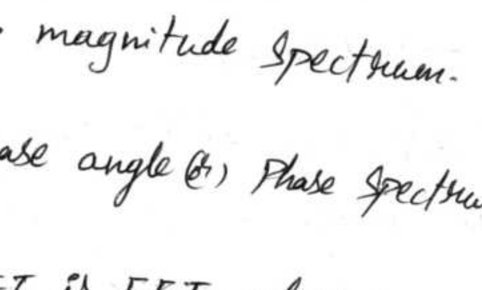


Fig (b).

* Properties of 2D DFT :-

1) Separable Property
2) Spatial Shift Property
3) Periodicity Property
4) Convolution Property
5) Correlation Property
6) Scaling Property
7) Conjugate Symmetry Property.
8) Rotation Property.

① Separable Property:

The separable property allows a 2D t/f to be computed in two steps by successive 1D operations on rows & columns of an image.

Proof:- Mathematically it is represented as, $F(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}mk}\, e^{-j\frac{2\pi}{N}nl}$

$$\Rightarrow F(k,l) = \sum_{m=0}^{M-1}\Big[\underbrace{\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}nl}}_{1D}\Big] e^{-j\frac{2\pi}{N}mk} = \sum_{m=0}^{M-1} \underbrace{F(m,l)}_{1D}\, e^{-j\frac{2\pi}{N}mk} = \underbrace{F(k,l)}_{2D}$$

Thus, performing a 2D Fourier t/f is equivalent to two 1D T/f's :

→ Performing a 1D t/f on each row of image $f(m,n)$ to get $F(m,l)$

→ Performing a 1D t/f on each column of $F(m,l)$ to get $F(k,l)$

x

② Spatial Shift Property:- The 2D-DFT of a shifted version of the image $f(m,n)$, i.e., $f(m-m_0, n)$ is given by;

$$f(m-m_0, n) \xrightarrow{DFT} \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m-m_0,n)\, e^{-j\frac{2\pi}{N}mk}\, e^{-j\frac{2\pi}{N}nl} \quad \text{①}$$

where, 'm_0' is representing no. of times the function $f(m,n)$ shifted.

Proof:- Adding and subtracting "m_0" to $e^{-j\frac{2\pi}{N}mk}$ in ①,

$$DFT[f(m-m_0,n)] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m-m_0,n)\, e^{-j\frac{2\pi}{N}(m-m_0+m_0)k}\, e^{-j\frac{2\pi}{N}nl}.$$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m-m_0,n)\, e^{-j\frac{2\pi}{N}(m-m_0)k} \cdot e^{-j\frac{2\pi}{N}m_0k} \cdot e^{-j\frac{2\pi}{N}nl}$$

$$= \left[\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m-m_0,n)\, e^{-j\frac{2\pi}{N}(m-m_0)k}\, e^{-j\frac{2\pi}{N}nl}\right] e^{-j\frac{2\pi}{N}m_0k}.$$

$$\therefore DFT\{f(m-m_0,n)\} = F[k,l]\, e^{-j\frac{2\pi}{N}m_0k}.$$

Hence, DFT of shifted function is "unaltered" except phase factor.

x

(3) Periodicity Property :- The 2D-DFT of a function $f(m,n)$ is said to be periodic with a period "N" if, $F(k,l) \longrightarrow F(k+pN, l+qN)$.

Proof :- Consider, $F(k+pN, l+qN) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}(k+pN)m}\, e^{-j\frac{2\pi}{N}(l+qN)n}$.

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}km} \cdot e^{-j\frac{2\pi}{N}pNm} \cdot e^{-j\frac{2\pi}{N}ln} \cdot e^{-j\frac{2\pi}{N}qNn}$$

$$= \left[\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}km}\, e^{-j\frac{2\pi}{N}ln}\right] e^{-j2\pi pm}\, e^{-j2\pi qn}$$

$$\therefore F(k+pN, l+qN) = F(k,l)\, e^{-j2\pi pm} \cdot e^{-j2\pi qn}.$$

Here, $e^{-j2\pi pm}$ & $e^{-j2\pi qn}$ are always '1' $\forall$ q & n

$\left[\because e^{-j2\pi} = \cos 2\pi - j\sin 2\pi = 1\right]$

Hence, $\boxed{F(k+pN, l+qN) = F(k,l)}$

(4) Convolution Property :- Convolution is one of the most important tool in DIP.

"Convolution in spatial domain is equals to multiplication in time domain".

Proof :- convolution of two sequences $x(n)$ & $h(n)$ is defined as,

$$x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k) \longrightarrow (1).$$

From (1) we can write 2D-convolution of arrays (or) Matrices $f(m,n)$ & $g(m,n)$ as,

$$f(m,n) * g(m,n) = \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\, g(m-a, n-b).$$

[$\because$ arry/matrix has finite no. of elements]

Now,

$$DFT\{f(m,n) * g(m,n)\} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\left\{\sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\, g(m-a, n-b)\right\} e^{-j\frac{2\pi}{N}mk}\, e^{-j\frac{2\pi}{N}nl}$$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\, g(m-a, n-b)\, e^{-j\frac{2\pi}{N}(m+a-a)k}\, e^{-j\frac{2\pi}{N}(n-b+b)l}$$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\, g(m-a, n-b)\, e^{-j\frac{2\pi}{N}(m-a)k}\, e^{-j\frac{2\pi}{N}ak}\, e^{-j\frac{2\pi}{N}(n-b)l}\, e^{-j\frac{2\pi}{N}bl}$$

$$= \left(\sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\, e^{-j\frac{2\pi}{N}ak}\, e^{-j\frac{2\pi}{N}bl}\right)\left(\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} g(m-a, n-b)\, e^{-j\frac{2\pi}{N}(m-a)k}\, e^{-j\frac{2\pi}{N}(n-b)l}\right)$$

$\boxed{\therefore DFT\{f(m,n) * g(m,n)\} = F(k,l) \times G(k,l)}$ Hence Proved.

⑤ Correlation Property :- The cross correlation of two sequences $x(n)$ & $h(n)$ is equivalent to performing the convolution of one sequence with the folded version of the other sequence.

Proof :- The DFT of correlation of two sequences $x(n)$ & $h(n)$ is defined as

$$DFT\{R_{x,h}\} = \sum_{m=0}^{N-1} \left\{ \sum_{n=0}^{N-1} x(n)\, h(n+m) \right\} e^{-j\frac{2\pi}{N} mk}.$$

$$= \sum_{m=0}^{N-1} \left\{ \sum_{n=0}^{N-1} x(n)\, h(n+m) \right\} e^{-j\frac{2\pi}{N}(m+n-n)k}.$$

$$= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(n)\, h(n+m)\, e^{-j\frac{2\pi}{N}(n+m)k}\, e^{-j\frac{2\pi}{N}(-n)k}.$$

$$= \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}(-n)k} \cdot \sum_{m=0}^{N-1} h(n+m)\, e^{-j\frac{2\pi}{N}(n+m)k}.$$

$$= H(k) \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N} n(-k)}$$

$$\boxed{DFT\{R_{x,h}\} = H(k) \cdot X(-k)}$$

It shows that the correlation in time domain equals to multiplication of DFT of one seq. & time reversal of DFT of another sequence.

Note :- Correlation is basically used to find relative similarity b/w two signals.

———— x ————

⑥ Scaling Property :- It is used to increase or decrease the size of an image.

It states that the expansion of signal in one domain is equal to the compression of signal in another domain.

i.e., if DFT of $f(m,n)$ is $F(k,l)$ then $DFT[f(am, bn)] = \frac{1}{|ab|} F(k/a, l/b)$.

Proof :- Consider, $DFT[f(am, bn)] = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(am, bn)\, e^{-j\frac{2\pi}{N} km}\, e^{-j\frac{2\pi}{N} ln}.$

Now, multiply & Divide the exponential terms $e^{-j\frac{2\pi}{N} km}$ with 'a' & $e^{-j\frac{2\pi}{N} ln}$ with 'b'.

Cont...

$$\Rightarrow \text{DFT}\left[f(am, bn)\right] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(am, bn)\, e^{-j\frac{2\pi}{N}(am)(k/a)}\; e^{-j\frac{2\pi}{N}(bn)(l/b)}$$

$$\therefore \text{DFT}\left[f(am, bn)\right] = \frac{1}{|ab|}\, F\left(k/a,\, l/b\right).$$

---

7) Conjugate Symmetry:- If the DFT of $f(m,n)$ is $F(k,l)$ then, $\text{DFT}\left[f^*(m,n)\right] = F(-k,-l)$

i.e., $F^*(k,l) = F(-k,-l)$

Proof:- The DFT of a function $f(m,n)$ is defined as,

$$F(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}mk}\, e^{-j\frac{2\pi}{N}nl}.$$

Applying complex conjugate,

$$F^*(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{+j\frac{2\pi}{N}mk}\, e^{+j\frac{2\pi}{N}nl}$$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}m(-k)}\, e^{-j\frac{2\pi}{N}n(-l)}$$

$$\therefore F^*(k,l) = F(-k,-l)$$

---

(8) Multiplication by Exponential:- If DFT of $f(m,n)$ is $F(k,l)$ then

$$\text{DFT}\left[e^{j\frac{2\pi}{N}mk_0}\, e^{j\frac{2\pi}{N}l_0 n} f(m,n)\right] = F(k-k_0,\, l-l_0).$$

Proof:-

Consider, $\text{DFT}\left[e^{j\frac{2\pi}{N}mk_0}\, e^{j\frac{2\pi}{N}nl_0} f(m,n)\right]$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \left[e^{j\frac{2\pi}{N}mk_0}\, e^{j\frac{2\pi}{N}nl_0} f(m,n)\right] e^{-j\frac{2\pi}{N}mk}\, e^{-j\frac{2\pi}{N}nl}.$$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}m(k-k_0)}\, e^{-j\frac{2\pi}{N}n(l-l_0)}$$

$\left[\because a^m \cdot a^n = a^{m+n}\right]$ (16)

$$\therefore \text{DFT}\left[e^{j\frac{2\pi}{N}mk_0}\, e^{j\frac{2\pi}{N}nl_0} f(m,n)\right] = F(k-k_0,\, l-l_0).$$

(a) Rotational Property: The rotation property states that if a function is rotated by an angle, then its Fourier transform also rotates by an equal amount.

$$f(m,n) \longrightarrow f(r\cos\theta,\ r\sin\theta).$$

$$DFT\left[f(r\cos\theta,\ r\sin\theta)\right] \longrightarrow F\left[R\cos\phi,\ R\sin\phi\right].$$

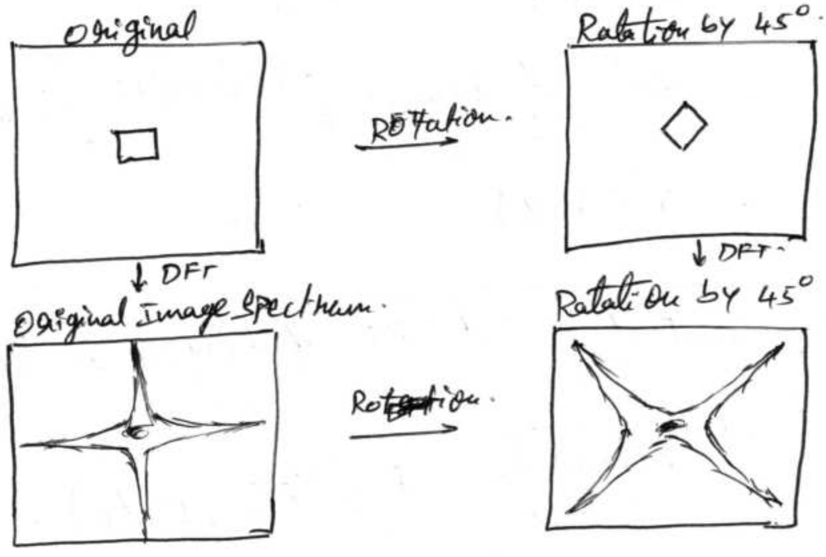
For simplicity purpose consider the polar form of the image

$$\Rightarrow \quad m = r\cos\theta \ , \quad n = r\sin\theta$$

$$\& \quad k = \omega\cos\phi \ , \quad l = \omega\sin\phi$$

then we can write the rotated transform pair as,

$$f(r,\ \theta+\theta_0) \Longleftrightarrow F(\omega,\ \phi+\theta_0).$$



* <u>Importance of Phase</u>:- The Fourier transform $F(k,l)$ can be expressed in polar coordinates as,

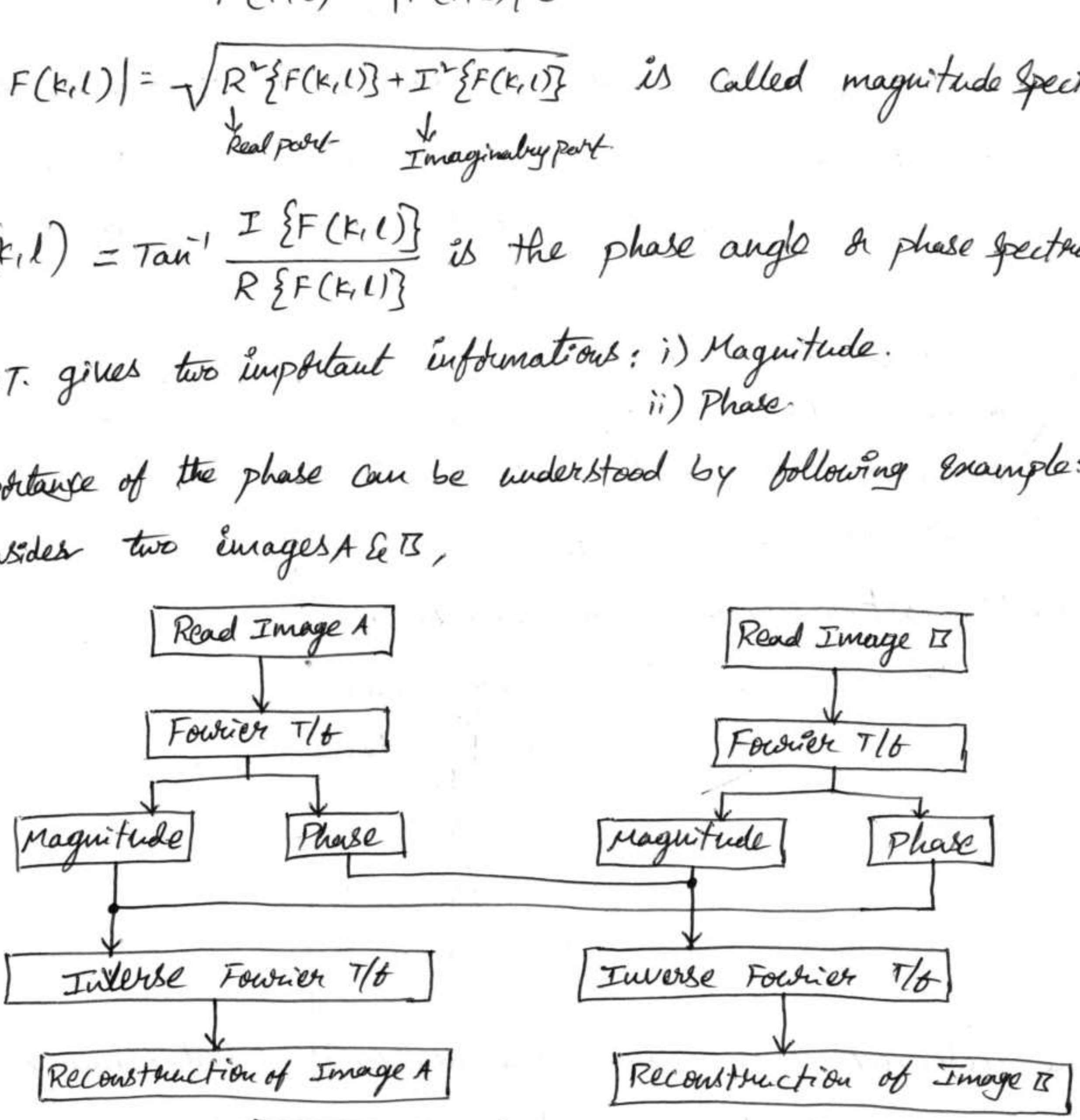
$$F(k,l) = |F(k,l)|\, e^{j\phi(k,l)}.$$

where, $|F(k,l)| = \sqrt{\underbrace{R^2\{F(k,l)\}}_{\text{Real part}} + \underbrace{I^2\{F(k,l)\}}_{\text{Imaginary part}}}$ is called magnitude spectrum

& $\phi(k,l) = \tan^{-1} \dfrac{I\{F(k,l)\}}{R\{F(k,l)\}}$ is the phase angle or phase spectrum.

The F.T. gives two important informations: i) Magnitude.
ii) Phase.

The importance of the phase can be understood by following example:
Consider two images A & B,



---

* <u>Walsh Transform</u>:- In 1923 Walsh introduced a complete set of orthogonal square-wave functions to represent an image function.
→ The Walsh-T/f is simple as it deals with real values and it takes only two values which are either '+1' or '-1'.

The 1D-Walsh T/f basis is, $\boxed{g(n,k) = \frac{1}{N}\prod_{i=0}^{m-1}(-1)^{b_i(n)\, b_{m-1-i}(k)}}$

where; $n \to$ Time index (0 to N-1)
$k \to$ frequency index (0 to N-1)
$N \to$ Order.
$m \to$ no. of bits ($m = \log_2 N$)

$b_i(n) \to i^{th}$ bit of binary value of 'n' represented in binary.
Note:- If $n=0$ or $k=0$ then $g(n,k) = 1/N$.

The 2D- Walsh transform of a function $f(m,n)$ is,

$$F(k,l) = \frac{1}{N} \sum_m \sum_n f(m,n) \prod_{i=0}^{P-1} (-1)^{[b_i(m) b_{P-1-i}(k) + b_i(n) b_{P-1-i}(k)]}$$

Problem:- Find the 1D walsh basis for the fourth-order system.

Sol:- Given, the system is of order-4 i.e., $N=4$.

$$m = \log_2 N = \log_2 4 = \log_2 (2^2) = 2(\log_2 2) = 2.$$

$\therefore \underline{m=2}$ & $\underline{N=4}$.

'n' ranges from 0 to $N-1$ i.e., $n = 0, 1, 2, 3$.

& 'k' ranges from 0 to $N-1$ i.e., $k = 0, 1, 2, 3$.

'i' varies from 0 to $m-1$ i.e., $i = 0, 1$

Note:- if $n=0$ or $k=0$ then $g(n,k) = \frac{1}{N}$

Table 1: Construction of walsh basis for $N=4$:

Decimal Value (n)	Binary Values: $b_i(n)$	
	$b_1(n)$	$b_0(n)$
0	$b_1(0) = 0$	$b_0(0) = 0$
1	$b_1(1) = 0$	$b_0(1) = 1$
2	$b_1(2) = 1$	$b_0(2) = 0$
3	$b_1(3) = 1$	$b_0(3) = 1$

The coefficients of the walsh representation is called Sequency Components. The sequency for $N=4$ is shown in below table:

Table 2: Walsh T/f basis for $N=4$

k \ n	0	1	2	3	Sequency:
0	1/4	1/4	1/4	1/4	zero sign change (DC value.
1	1/4	1/4	−1/4	−1/4	one sign change
2	1/4	−1/4	1/4	−1/4	Three sign change.
3	1/4.	−1/4	−1/4	1/4	Two sign change.

Each value in the above table has to be calculated using the formula, $g(n,k) = \frac{1}{N} \prod_{i=0}^{m-1} (-1)^{b_i(n)\, b_{m-1-i}(k)}$.

Cont...

letus Consider the calculation of $g(2,1)$,

$$g(2,1) = \frac{1}{4} \prod_{i=0}^{1} (-1)^{b_i(2) b_{m-1-i}(1)}$$

$$= \frac{1}{4} \left\{ (-1)^{b_0(2) b_1(1)} \times (-1)^{b_1(2) b_0(1)} \right\}$$

$$= \frac{1}{4} \left\{ (-1)^0 \times (-1)^1 \right\} = \underline{\underline{-1/4}}.$$

likewise we have to calculate all values of walsh T/f.

$$\therefore \quad g(u,k) = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & -1/4 & -1/4 \\ 1/4 & -1/4 & 1/4 & -1/4 \\ 1/4 & -1/4 & -1/4 & 1/4 \end{bmatrix}$$

* we can <u>easily</u> find the sign of each value in table (or) matrix without using formula, by following the below steps,

step1: write the binary value of 'n'.
step2: write the binary value of 'k' in reverse order.
step3: Check for the no. of pairs of '1'.
i) if it is even then +ve sign (include '0').
ii) if it is odd then -ve sing

<u>Ex</u>:- Consider same, $g(2,1)$.

Here $n = 2 \longrightarrow 1\,0 \longrightarrow \boxed{1}\,0$
$k = 1 \longrightarrow 0\,1 \xrightarrow{\text{reverse}} \boxed{1}\,0$

1 pair which is odd!

$\therefore$ $g(2,1)$ coefficient has -ve sign (Satisfied).

* **Hadamard Transform**:- It is basically the same as the Walsh t/f except the rows of the Transform matrix are re-ordered.

→ The elements of mutually orthogonal basis vectors of a Hadamard T/f are either +1 or -1; which results in very low computational complexity in the calculation of the transform coefficients.

→ Hadamard matrices are easily constructed for $N = 2^n$ by the following procedure.

The order $N=2$ the Hadamard matrix is given as,

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

The Hadamard matrix of order 2N can be generated by "Kronecker Product" operation:

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

Now, substitute $N=2$ in above eqn.,

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & +1 \end{bmatrix}$$

$||^{ly}$ for $N=4$, we get,

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

The Hadamard matrix of order $N = 2^n$ may be generated from the order 'two' core matrix. It is not desired to store entire Matrix

* **Haar Transform :-** It is based on a class of ~~&~~ orthogonal matrices whose elements are either 1, −1, or 0 multiplied by powers of $\sqrt{2}$.

→ The Haar transform is computationally efficient transform as the transform of an N-point vector requires only $2(N-1)$ additions & 'N' multiplications.

Algorithm to Generate Haar Basis:

step 1: Determine the order of Haar basis "N."

step 2: Determine 'n' where, $n = \log_2 N$.

step 3: Determine P & q.

i) $0 \le P \le n-1$

ii) If $P=0$ then, $q=0$ or $q=1$.

iii) If $P \neq 0$ then, $1 \le q \le 2^P$.

step 4: Determine 'K', $K = 2^P + q - 1$.

step 5: Determine 'Z'

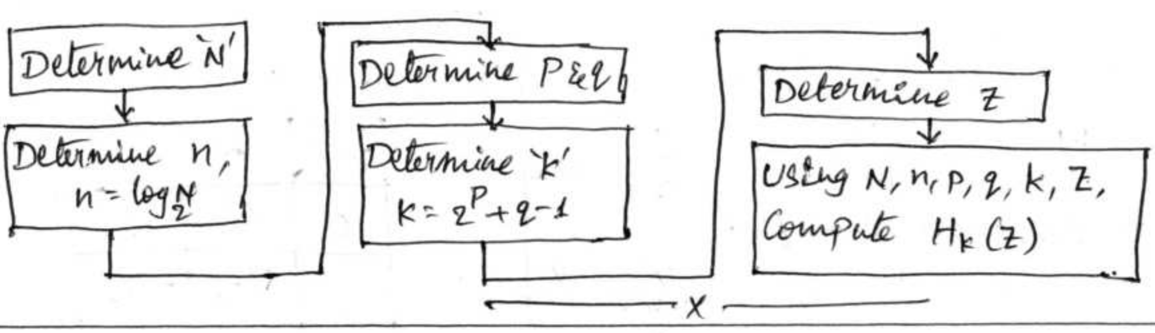
$$Z \rightarrow [0, 1] \Rightarrow \left\{ \frac{0}{N}, \frac{1}{N}, \cdots \frac{N-1}{N} \right\}$$

step 6: If $K=0$ then $H(Z) = \frac{1}{\sqrt{N}}$

otherwise,

$$H_K(Z) = H_{Pq}(Z) = \frac{1}{\sqrt{N}} \begin{cases} +2^{P/2} & \frac{(q-1)}{2^P} \le Z < \frac{(q-1/2)}{2^P} \\ -2^{P/2} & \frac{q-1/2}{2^P} \le Z < q/2^P \\ 0 & \text{otherwise} \end{cases}$$

Flow chart Format :-



——— x ———

Problem:- Generate one Haar Basis for $N=2$.

Sol:- Step1: $N = 2$

Step2: $n = \log_2 2 = 1$.

Step3: i) we have, $n=1$ then $P = 0,1,2 \cdots n-1$
i.e, $P = 0$.

ii) As $P=0$, $q=0$ or $q=1$.

Step4: $k = 2^P + q - 1 = 2^0 + 0 - 1$ (for $q=0$)

$k = 1-1 = 0$

& $k = 2^0+1-1 = 1+1-1 = 1$ (for $q=1$)

P	q	k
0	0	0
0	1	1

Step5: $z \to [0,1) \Rightarrow \left\{\frac{0}{2}, \frac{1}{2}\right\} \Rightarrow \left\{0, \frac{1}{2}\right\}$.

Step6: If $k=0$, then $H(z) = \frac{1}{\sqrt{N}}$

$\therefore$ for $q=0 \Rightarrow k=0$ $H(z) = \frac{1}{\sqrt{2}}$ //

for $q=1 \Rightarrow k=1$, $H_k(z) = H_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{P/2} & \frac{q-1}{2^P} \le z < \frac{q-1/2}{2^P} \\ -2^{P/2} & \frac{q-1/2}{2^P} \le z < \frac{q}{2^P} \\ 0 & \text{e.w.} \end{cases}$

$$\Rightarrow H_k(z) = \frac{1}{\sqrt{2}} \begin{cases} 2^{0/2} & \frac{1-1}{2^0} \le z < \frac{1-\frac{1}{2}}{2^0} \\ -2^{0/2} & \frac{1-1/2}{2^0} \le z < \frac{1}{2^0} \\ 0 & \text{e.w.} \end{cases}$$

$$\Rightarrow H_k(z) = \frac{1}{\sqrt{2}} \begin{cases} 1 & 0 \le z \le \frac{1}{2} \\ -1 & \frac{1}{2} \le z \le 1 \\ 0 & \text{otherwise.} \end{cases}$$

Now,

for $z=0$, first condition satisfies,

$H(z) = \frac{1}{\sqrt{2}}(1) = \frac{1}{\sqrt{2}}$ //

for $z = \frac{1}{2}$, II condition satisfies,

$H(z) = \frac{1}{\sqrt{2}}(-1) = -1/\sqrt{2}$ //

Haar basis for $N=2$ is,

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

—o—

* Slant Transform :- A slant basis vector (sawtooth waveform) is monotonically decreasing in constant steps from maximum to minimum has a sequency property and has a fast computational Algorithm.

Let S_N denote an $N \times N$ slant matrix with $N = 2^n$.

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

The S_4 matrix is obtained by following opertation :

$$S_2 = \frac{1}{\sqrt{2}} \left[\begin{array}{cc:cc} 1 & 0 & 1 & 0 \\ a & b & -a & b \\ \hdashline 0 & 1 & 0 & -1 \\ -b & a & b & a \end{array}\right] \left[\begin{array}{c:c} S_1 & 0 \\ \hdashline 0 & S_1 \end{array}\right]$$

Note :- $b_n = (1 + 4a_{n-1}^2)^{-1/2}$, $a_1 = 1$

$a_n = 2 b_n a_{n-1}$, $N = 2^n$.

If $b = \frac{1}{\sqrt{5}}$, $a = 2b = \frac{2}{\sqrt{2}}$ then we get slant matrix,

$$\cancel{S_4 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 2/\sqrt{5} & 1/\sqrt{5} & -2/\sqrt{5} & \end{bmatrix}}$$

$$S_4 = \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 2/\sqrt{5} & 1/\sqrt{5} & -2/\sqrt{5} & 1/\sqrt{5} \\ 0 & 1 & 0 & -1 \\ -1/\sqrt{5} & 2/\sqrt{5} & 1/\sqrt{5} & 2/\sqrt{5} \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

on simplifying,

$$\Rightarrow S_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{3}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix} \begin{array}{l} \rightarrow \text{zero sign change.} \\ \rightarrow \text{one} \quad " \quad " \\ \rightarrow \text{Two} \quad " \quad " \\ \rightarrow \text{Three} \quad " \quad " . \end{array}$$

Here the sequency property show the sign changes are linear variant.

$\therefore$ The slant T/f reproduces linear variationes of brightness very well.

It is mostly used in edge detection applications

Note :- Observe the reference concept in last page of unit for Higher order slant T/f.

* Discrete Cosine Transform :- (DCT).

→ The DCT's are members of real-valued discrete sinusoidal unitary transforms.

→ A DCT consists of a set of basis vectors that are sampled cosine functions.

→ A DCT is a technique for converting a signal into elementary frequency components.

→ It is widely used in image compression (JPEG).

→ DCT and DFT works on sinusoidal functions, the difference is DCT has only real values whereas DFT has complex values.

→ DCT is obtained by extending the Discrete sine seq. of DFT.

$$f(x) \xrightarrow{\text{1D DCT}} F(u)$$
$$f(x,y) \xrightarrow{\text{2D DCT}} F(u,v).$$

1D-DCT: $F(u) = C \cdot f(x)$

2D-DCT: $F(u,v) = C \cdot f(x,y) \cdot C^T$.

where,

C = Cosine T/f matrix.

$f(x)$ & $f(x,y)$ = Spatial Domain function

$$C(u,v) \begin{cases} = \sqrt{\frac{1}{N}} & \text{for } u=0 \text{ \& } 0 \le v \le N-1. \\ = \sqrt{\frac{2}{N}} \cos\left[\frac{(2v+1)\pi u}{2N}\right] & 1 \le u \le N-1 \text{ \& } 0 \le v \le N-1 \end{cases}$$

---

* Problem :- Obtain DCT matrix for $N=4$.

Sol Given, $N=4$ → $u = 0,1,2,3$.
& $v = 0,1,2,3$.

Now,

$$C(u,v) = \begin{array}{c|c} & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.2705 & -0.2705 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2705 & -0.653 & 0.653 & -0.2705 \end{bmatrix} \end{array}$$

$$\therefore \begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.2705 & -0.2705 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2705 & -0.653 & 0.653 & -0.2705 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix}.$$

DCT Properties:-

1) DCT is real & orthogonal.
2) If 'A' is a DCT Matrix & if 'A' is orthogonal $A \times A^T = I$.
3) $A^{-1} = A^T$.

---

* KL Transform (Karhunen - Loeve):- It is a series expansion method for Continuous random Processes.

→ It is also called "Hotelling Transform." (or) "Eigen vector T/f."

→ It is a reversible linear T/f that exploits the statistical properties of a vector representation.

→ The basis functions of KL T/f are orthogonal eigen vectors of covariance matrix of a data set.

→ More image Compression can be achieved by decorrelating data from neighbouring pixels.

Steps to find KL Transform:-

i) Find the mean vector & Covariance matrix of given image 'x'.

ii) Find the Eigen values & then the Eigen vectors of Covariance matrix.

iii) Create the transformation matrix 'T', ∋ rows of 'T' are Eigen vectors.

iv) Find K.L. Transform.

Drawbacks of KL Transform:- There are two serious drawbacks.

i) It is i/p-dependent & the basis function has to be calculated for each signal model on which it operates. It has no specific mathematical structure, leads to false implementation.

ii) It requires $O(m^2)$ multiply/addition operations whereas the DFT & DCT need $O(\log_2 m)$ multiplications only.

Applications of KL Transform:-

i) Clustering Analysis:- It is used in clustering analysis to determine a new Coordinate system for sample data.

Cont...

ii) Image Compression:- It is heavily used in performance evaluation of compression algorithms, since KL Spectrum contains large number of zero-valued Coefficient.

---

* Singular Value Decomposition (SVD) Transform:-

The SVD of a rectangular matrix 'A' is a decomposition of the form: $A = UDV^T$.

where, 'A' is an $m \times n$ matrix.

'U','V' are orthogonal matrices.

'D' is a diagonal Matrix formed by singular values of 'A'.

The singular values $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_n \geq 0$ appear in descending order along the main diagonal of D.

These singular values are obtained by taking square root of Eigen values of AA^T & A^TA.

$$\Rightarrow A = UDV^T = [u_1, u_2, u_3 \cdots u_n] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ \vdots \\ v_n^* \end{bmatrix}.$$

The relation b/w SVD & eigen values:

$$A = UDV^T$$

Now, $AA^T = (UDV^T)(UDV^T)^T = UDV^TVDU^T = \underline{UD^2U^T}$

& $A^TA = (UDV^T)^T(UDV^T) = VDU^TUDV^T = \underline{VD^2V^T}$

→ Thus U & V are calculated as Eigen vectors of AA^T & A^TA respectively.

→ If matrix 'A' is real then singular values are always real & U, V also.

Properties of SVD:-

i) The singular values ($\sigma_1 - \sigma_n$) are unique & U, V are not unique.

ii) $A^TA = (UDV^T)^TUDV^T = VD^TDV^T$; hence 'V' diagonalises A^TA. 'V' can be obtained by A^TA.

iii) 'U' can be obtained by AA^T.

iv) The rank of 'A' is equals to no. of its non-zero singular values.

Applications:-

1. SVD approach can be used in image compression.
2. SVD can be used in face recognition
3. SVD can be used in water marking.
4. SVD can be used for Texture classification.

* <u>Radon Transform</u>:- The Radon transform is named after the Austrian ~~Scientist~~ Mathematician Johann Karl August Radon.

→ It is used to compute the projection of an object in its image.

→ Applying Radon transform to an image $f(m,n)$ for a given set of angles can be thought of as computing the projection of the image along the given angles.

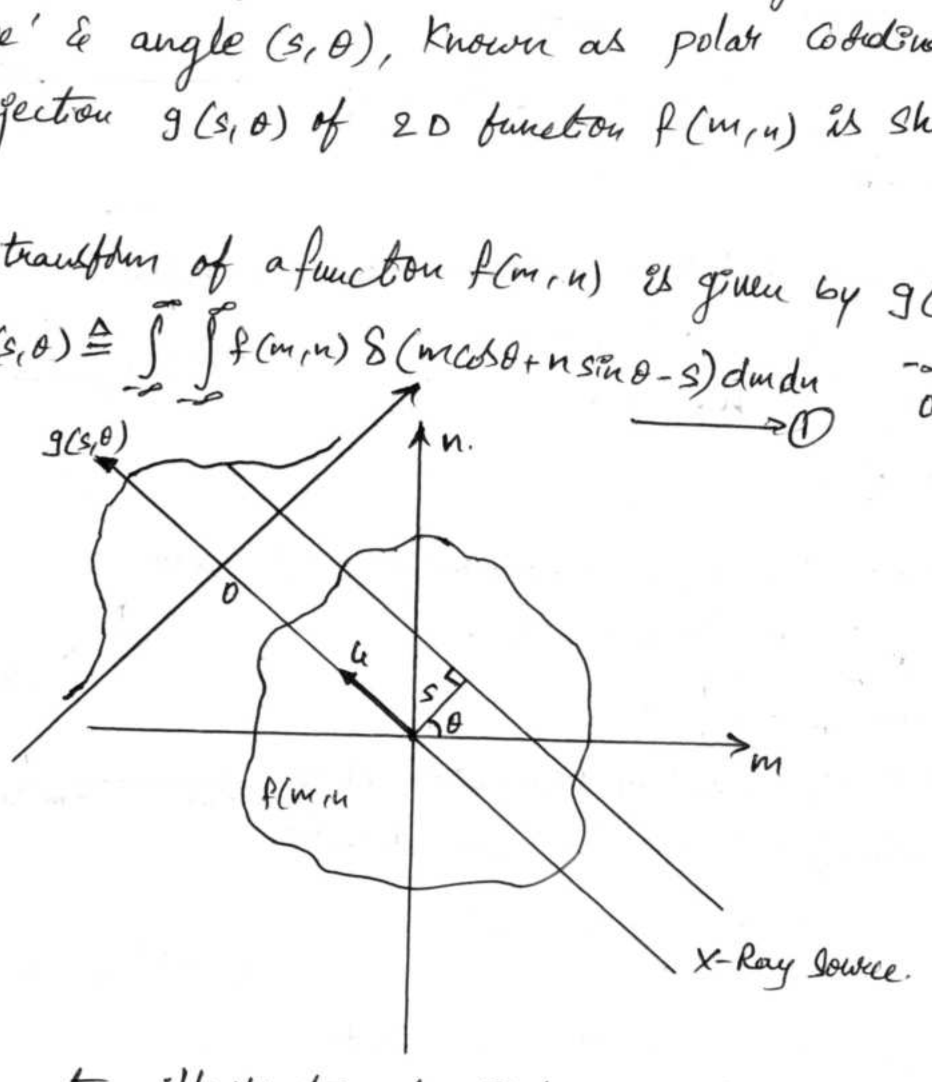
→ The resulting projection is the sum of the intensities of the pixels in each direction, i.e., line integral.

→ Radon T/f is mapping from Cartesian rectangular coordinates to the 'distance' & angle (s,θ), known as polar coordinates.

→ The 1D projection $g(s,\theta)$ of 2D function $f(m,n)$ is shown in below figure.

→ The Radon transform of a function $f(m,n)$ is given by $g(s,\theta)$ where,

$$g(s,\theta) \triangleq \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(m,n)\,\delta(m\cos\theta + n\sin\theta - s)\,dm\,dn \longrightarrow (1) \qquad -\infty < s < \infty,\ 0 \leq \theta < \pi$$



<u>Fig(1)</u>: <u>Schematic</u> <u>illustration of</u> <u>Radon Transform in Tomography</u>:

The angle of projection is illustrated clearly in Figure(2) below which is derived from above figure.

P.T.O...

From the beside figure,

$$s = m\cos\theta + n\sin\theta \longrightarrow (2)$$

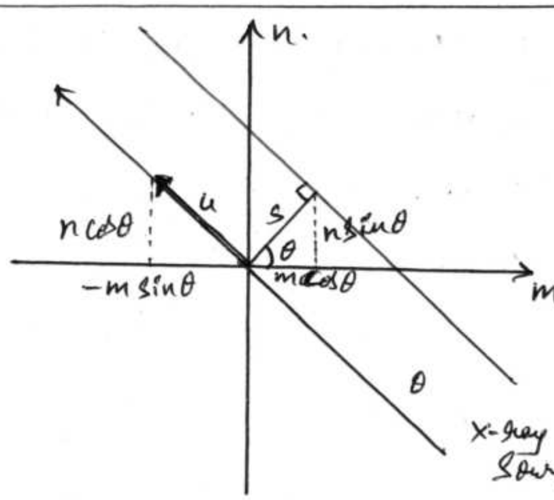


Fig2-Angle of projection.

From fig 4,

$$u = -m\cos(90-\theta) + n\sin(90-\theta).$$

$$u = -m\sin\theta + n\cos\theta \longrightarrow (3)$$

Multiplying (2) with 'sinθ' on both sides,

$$s \cdot \sin\theta = m\cos\theta\sin\theta + n\sin^2\theta \longrightarrow (4)$$

Multiplying (2) with 'cosθ' on both sides,

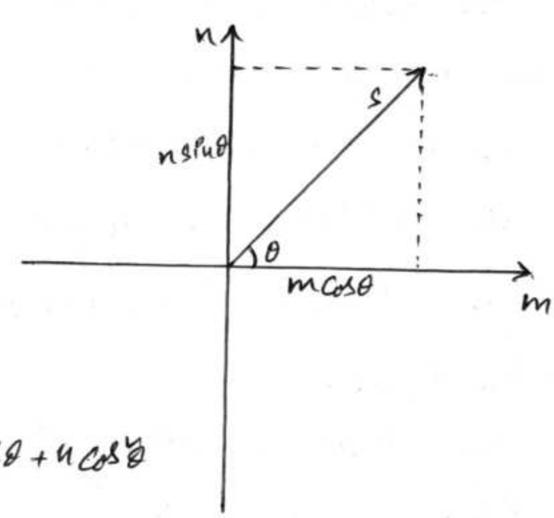
$$s\cos\theta = m\cos^2\theta + n\sin\theta\cos\theta \longrightarrow (5)$$

Multiplying (3) with 'sinθ' on both sides,

$$u\sin\theta = -m\sin^2\theta + n\cos\theta\sin\theta \longrightarrow (6)$$

Multiplying (3) with 'cosθ' on both sides,

$$u\cos\theta = -m\sin\theta\cos\theta + n\cos^2\theta \longrightarrow (7)$$



Add (4) & (7):

$$s\sin\theta + u\cos\theta = m\cos\theta\sin\theta + n\sin^2\theta - m\sin\theta\cos\theta + n\cos^2\theta$$

$$= n(\sin^2\theta + \cos^2\theta)$$

$$\boxed{\therefore s\sin\theta + u\cos\theta = n} \longrightarrow (8)$$

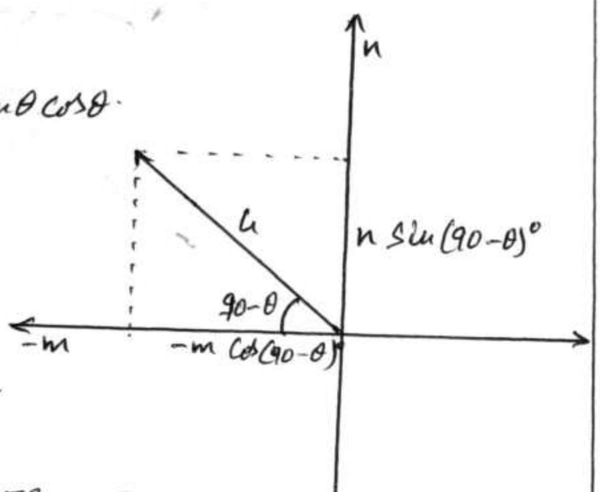
Fig 3: Projection of Distances into m & n planes.

subtract (5), (6):

$$s\cos\theta - u\sin\theta = m\cos^2\theta + n\sin\theta\cos\theta + m\sin^2\theta - n\sin\theta\cos\theta$$

$$= m(\cos^2\theta + \sin^2\theta)$$

$$\boxed{\therefore (s\cos\theta - u\sin\theta) = m} \longrightarrow (9)$$



we know that, the resulting projection is the "sum of intensities of pixels in each direction,

i.e., $g(s,\theta) = \int_L f(m,n)\,du$

from (8), (9),

$$\Rightarrow g(s,\theta) = \int_0^L f(s\cos\theta - u\sin\theta,\ s\sin\theta + u\cos\theta)\,du.$$

Fig 4: Projection of vector u in m & n planes.

i.e., Radon T/f, $R_f = \int_0^L f(s\cos\theta - u\sin\theta,\ s\sin\theta + u\cos\theta)\,du = g(s,\theta).$

In freq. domain, $G(\omega,\theta) = \int_{-\infty}^{\infty} g(s,\theta)\, e^{-j2\pi\omega s}\,ds.$ [Fourier T/f]

on simplification we get, $G(\omega,\theta) = F(\omega\cos\theta,\ \omega\sin\theta)$

Prob :- Perform KL transform for the following matrix: $X = \begin{bmatrix} 4 & -2 \\ -1 & 3 \end{bmatrix}$

Sol: 1) Formation of vectors:

Given is 2x2 matrix; hence two vectors can be extracted:

$$x_0 = \begin{bmatrix} 4 \\ -1 \end{bmatrix} \quad \& \quad x_1 = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$$

2) Covariance Matrix:

$$Cov(x) = E[xx^T] - \bar{x}\bar{x}^T$$

$$E[xx^T] = \frac{1}{M}\sum_{k=0}^{M-1} x_k x_k^T$$

Here, $M=2$,

$$E[xx^T] = \frac{1}{2}\sum_{k=0}^{2-1} x_k x_k^T$$

$$= \frac{1}{2}\left\{ x_0 x_0^T + x_1 x_1^T \right\}$$

$$= \frac{1}{2}\left\{ \begin{bmatrix} 4 \\ -1 \end{bmatrix}\begin{bmatrix} 4 & -1 \end{bmatrix} + \begin{bmatrix} -2 \\ 3 \end{bmatrix}\begin{bmatrix} -2 & 3 \end{bmatrix} \right\}$$

$$= \frac{1}{2}\left\{ \begin{bmatrix} 16 & -4 \\ -4 & 1 \end{bmatrix} + \begin{bmatrix} 4 & -6 \\ -6 & 9 \end{bmatrix} \right\}$$

$$E[xx^T] = \frac{1}{2}\left\{ \begin{bmatrix} 20 & -10 \\ -10 & 10 \end{bmatrix} \right\} = \begin{bmatrix} 10 & -5 \\ -5 & 5 \end{bmatrix}.$$

Now, Min. value of $E[xx^T] = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$.

$\bar{x}\bar{x}^T$ calculation,

$$\bar{x} = \frac{1}{M}\sum_{k=0}^{M-1} x_k = \frac{1}{2}\sum_{k=0}^{1} x_k$$

$$= \frac{1}{2}\left\{ \begin{bmatrix} 4 \\ -1 \end{bmatrix} + \begin{bmatrix} -2 \\ 3 \end{bmatrix} \right\} = \frac{1}{2}\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

$$\bar{x}\bar{x}^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Hence,

$$Cov(x) = E[xx^T] - \bar{x}\bar{x}^T$$

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

$$\therefore Cov(x) = \begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix}.$$

3) Eigen values of Covariance Matrix:

$$|Cov(x) - \lambda I| = 0$$

$$\left| \begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0.$$

$$\begin{vmatrix} 1-\lambda & -2 \\ -2 & -\lambda \end{vmatrix} = 0.$$

$$\Rightarrow (1-\lambda)(-\lambda) - (-2)(-2) = 0$$

$$\Rightarrow -\lambda + \lambda^2 - 4 = 0.$$

$$\Rightarrow \lambda^2 - \lambda - 4 = 0$$

Roots of above eqn are eigen values,

$$\lambda = \frac{+1 \pm \sqrt{(-1)^2 - 4(1)(-4)}}{2(1)} = \frac{1 \pm 4.1231}{2}$$

$$\Rightarrow \lambda_0 = \frac{1+4.1231}{2} \qquad \lambda_1 = \frac{1-4.1231}{2}$$

$$\lambda_0 = \underline{2.5615} \qquad \lambda_1 = \underline{-1.5615}$$

4) Eigen vector Calculation:

The first Eigen vector ϕ_0 is found from,

$$(Cov(x) - \lambda_0 I)\phi_0 = 0.$$

$$(Cov(x) - \lambda_0 I)\phi$$

$$= \left\{ \begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} 2.5615 & 0 \\ 0 & 2.5615 \end{bmatrix} \right\}\begin{bmatrix} \phi_{00} \\ \phi_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

$$\Rightarrow \begin{bmatrix} -1.5615 & -2 \\ -2 & -2.5615 \end{bmatrix}\begin{bmatrix} \phi_{00} \\ \phi_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

using row detection method, we found ϕ_{01} to be free variable, let $\phi_{01} = 1$.

$$\Rightarrow -1.5615\phi_{00} - 2\phi_{01} = 0.$$

$$\Rightarrow -1.5615\phi_{00} = 2$$

$$\Rightarrow \phi_{00} = -1.2808$$

$$\therefore \text{Eigen vector } \phi_0 = \begin{bmatrix} -1.2808 \\ 1 \end{bmatrix}.$$

IIIy for ϕ_1 Eigen vector,

$$(Cov(x) - \lambda_1 I)\phi_1 = 0$$

$$\Rightarrow \left\{ \begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} -1.5615 & 0 \\ 0 & -1.5615 \end{bmatrix} \right\} \begin{bmatrix} \phi_{10} \\ \phi_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

let, $\phi_{11} = 1$,

$\Rightarrow 2.5615\phi_{10} - 2\phi_{11} = 0$

$$\Rightarrow \quad \phi_{10} = \frac{2}{2.5615}$$

$$\phi_{10} = 0.7808$$

$\therefore$ Eigen vector, $\phi_1 = \begin{bmatrix} 0.7808 \\ 1 \end{bmatrix}$.

⑤ Normalizing Eigen vectors:

$$\frac{\phi_0}{\|\phi_0\|} = \frac{1}{\sqrt{\phi_{00}^2 + \phi_{01}^2}} \begin{bmatrix} \phi_{00} \\ \phi_{01} \end{bmatrix}$$

$$= \frac{1}{\sqrt{(-1.2808)^2 + 1^2}} \begin{bmatrix} -1.2808 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.7882 \\ 0.6154 \end{bmatrix}$$

$\parallel^{ly}$ for $\dfrac{\phi_1}{\|\phi_1\|} = \begin{bmatrix} 0.6154 \\ 0.7882 \end{bmatrix}$.

⑥ The transformation matrix "T":

$$T = \begin{bmatrix} \phi_0 & \phi_1 \end{bmatrix} \text{ (or) } \begin{bmatrix} \phi_0 \\ \phi_1 \end{bmatrix}.$$

$$\therefore T = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}.$$

⑦ K.L. Transform:-

$$Y = T \cdot X$$

$$Y_0 = T X_0$$

$$\Rightarrow Y_0 = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \end{bmatrix}.$$

$$\therefore Y_0 = \begin{bmatrix} -3.7682 \\ 1.6734 \end{bmatrix}.$$

$$Y_1 = T X_1$$

$$= \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} -2 \\ 3 \end{bmatrix}.$$

$$\therefore Y_1 = \begin{bmatrix} 3.4226 \\ 1.1338 \end{bmatrix}.$$

$$\therefore Y = \begin{bmatrix} -3.7682 & 3.4226 \\ 1.6734 & 1.1338 \end{bmatrix}.$$

which is K-L-Transform.

— o —

Note:- For reconstruction of original matrix,

$$X = T^T Y. \text{ i.e.,}$$

$$X_0 = T^T Y_0$$

$$\& \; X_1 = T^T Y_1$$

— x —

* Slant transform generalized matrix:

$$S_n = \frac{1}{\sqrt{2}} \times \left[\begin{array}{cc|c|cc|c} 1 & 0 & 0 & 1 & 0 & 0 \\ a_n & b_n & & -a_n & b_n & \\ \hline \multicolumn{2}{c|}{0} & I & \multicolumn{2}{c|}{0} & I \\ \hline 0 & 1 & 0 & 0 & -1 & 0 \\ -b_n & a_n & & b_n & a_n & \\ \hline \multicolumn{2}{c|}{0} & I & \multicolumn{2}{c|}{0} & -I \end{array}\right] \times \left[\begin{array}{c|c} S_{n-1} & 0 \\ \hline 0 & S_{n-1} \end{array}\right]$$

$$S_1 = \frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad N = 2^n.$$

$$b_n = (1 + 4a_{n-1}^2)^{-1/2} \quad \& \quad a_1 = 1.$$

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

$\longrightarrow$ END $\longleftarrow$