Digital Image Processing (DIP) - Processing of images which are digital in nature by using digital computers.

Digital System - It is a system which is driven or made with electro mechanical elements and controlled by electronic circuits.

Ex:- computers, Pendrives etc..)

Image consists of finite number of samples and these elements are called as pixels (or) pels

represented as $f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots \end{bmatrix}$ pixel

Origin of DIP -

First application of digital images was in the newspaper industry, here the pictures are sent by submarine cable (which is a time taking process, more than a week)

lateron, Bartlane cable picture transmission system in arly 1920's reduced the time required for the transport of pictures. The visual quality is not good in the intial methods.

next, a technique based on photographic reproduction made from tapes perforated at the telegraph receiving terminal. In this, tonal quality is improved and resolution also.

In Bartlane systems, coding of images is done in five distinct gray levels. This capability was increased to 15 levels in 192c

A system for developing a film plate via light beams that were modulated by the coded picture tape improved the reproduction process.

of supporting technologies that include data storage, display and transmission.

Next John von Neumann introduced two key concepts (i) a memory to hold a stored program and data (ii) conditional branching. These two concepts gave the idea for foundation of CPU and other key advances led to computers powerful enough to be used for digital image processing.

The first computers powerful enough to carryout meaningful processing tasks appeared in 1960's. This is used to correct various types of distortions in the image. And used to enhance and restore the images of moon & in many space applications.

Digital Image processing techniques which began in the late 1960's and early 1970's are used in medical engineering, remote earth resources observations and astronomy., Computer Tomography (CT) is one of the most important events in the application of image processing in medical diagnosis.

Computer procedures in DIP are used to enhance the contrast or code the intensity levels into colour for easier interpretation of X-rays and other images used in industry, medicine and the biological sciences.

This Digital image processing techniques are used in many fields such as automatic processing of finger prints, screening of X-rays and blood samples etc., The increase in the performance of computer and expansion of networking gave scope for the growth of digital Image Processing.

Biometrics - It refers to the way of identifying human -beings based on physiological and behavioral characteristics physiological characteristics implies finger prints, face, DNA and Iris etc.,

Image processing is used to analyse and recognize finger print, face, DNA and Iris.

Medical Imaging -

Image processing is very useful in interpreting medical images, from simple diagnosis to advanced telesurgical applications etc.,

This is used in X-rays, CT, MRI, PET and ultrasound and also for combining image modalities.

Factory Automation -

Automated visual inspection is a vast field where image processing is used by industries such as aerospace, food, textiles and plastic for automated surface testing.

Factory automation includes measurement of belt width, surface quality inspection, fiber analysis etc.,

Remote sensing - The role of ~~[illegible]~~ image processing in remote sensing applications is quite immense.

weather forecasting and prediction of atmospheric changes etc.,

Environmental monitoring applications have been developed to monitor deserts, forest etc.,

Defence / Military Applications -

Many applications such as military reconnaissance systems use image processing technology. Thermal images have the ability to acquire useful images at night and under atmospheric conditions such as fog & smoke.

photography -

Imaging processing helps in creating special effects such as warping, blending, animation and other visual effects.

Entertainment -

Photography is an excellent example of how image processing is helpful to common man. The applications are video conferencing, video phones, video editing, animation and image morphing etc., extracting image components that are useful in the representation and description of shape. Here the inputs are images and outputs are attributes extracted from those images.

Segmentation - Here partition of an image in-to its constituent parts or objects is done. A rugged segmentation procedure gives successful solution of imaging problems.

Representation and Description - The output of the segmentation stag is usually a raw pixel data, constituting either the boundary of a region or the points in the region itself. So, to convert the data to more suitable form for computer processing and to decide whether the data should be represented as a boundary or as a complete region, appropriate regional representation should be there.

Description is the feature selection, which deals with extracting attributes.

Recognition - Here we assigns a label or any symbol to an object based on its descriptors. We can recognise the image by using some of the coding techniques.

Components of an Image Processing System -

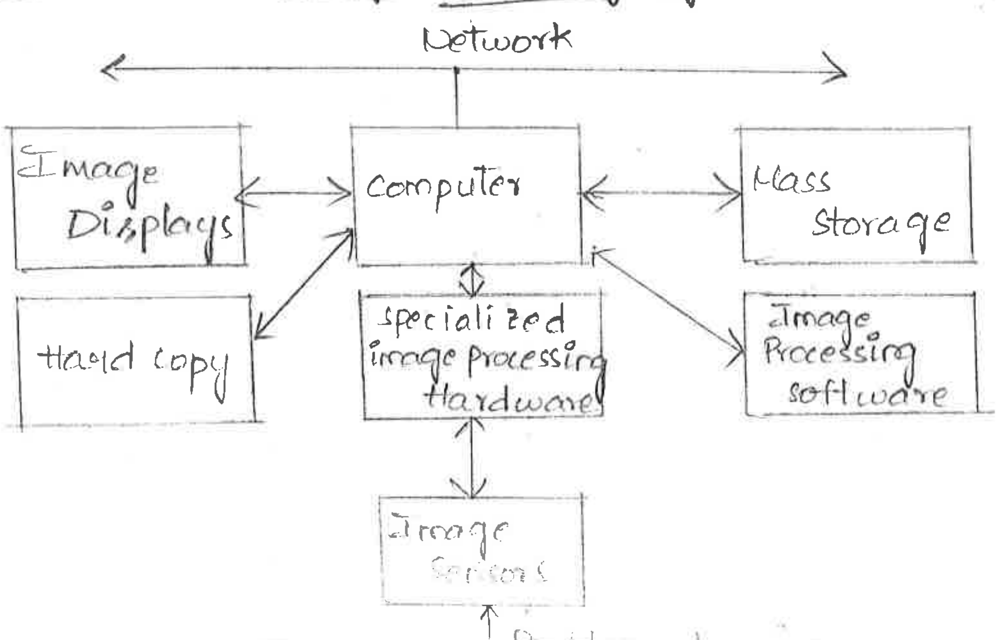


Image Sensors - Generally two elements are needed ...

digital images. One is physical device ie, we can sense the image by with the energy radiated from the object. And other element is 'digitizer' which is used to convert the sensed image in to digital format.

Specialized Image Processing Hardware - It contains hardware which is used to perform arithematic and logical operations and some other operations on all images at the same time. It gives very fast outputs.

Computer - It is a general purpose computer which can range from a PC to a super computer. Depending on level of performance needed, we use different computers ie, PC or super computer or custom computer etc. But now, we use well-equipped PC-type machine which is more suitable for off-line image programming tasks.

Image Processing Software - This contains specialized hardware modules to perform specific task. More sophisticated software packages allow the integration of those modules.

Mass Storage - If the image is not compressed, it requires lot of storage space, a single image may need 1 Megabyte of storag ie, depending on the intensity level of each pixel in the image. So to provide adequate and efficient storage, we compress the image.

Image displays - Image displays are mainly colour T.V Monitors, which are driven by the outputs of images and graphic display cards, In some cases, we use stereo displays.

Hard Copy devices - These are used for recording images, such as laser printers, film cameras, heat-sensitive devices, optical and CD ROM disks etc.) Generally, Films are preferrable because these provide highest possible resolution.

... enhancement techniques, according to our application, we use them.

Ex:- Thresholding, clipping etc..,

Image Restoration -

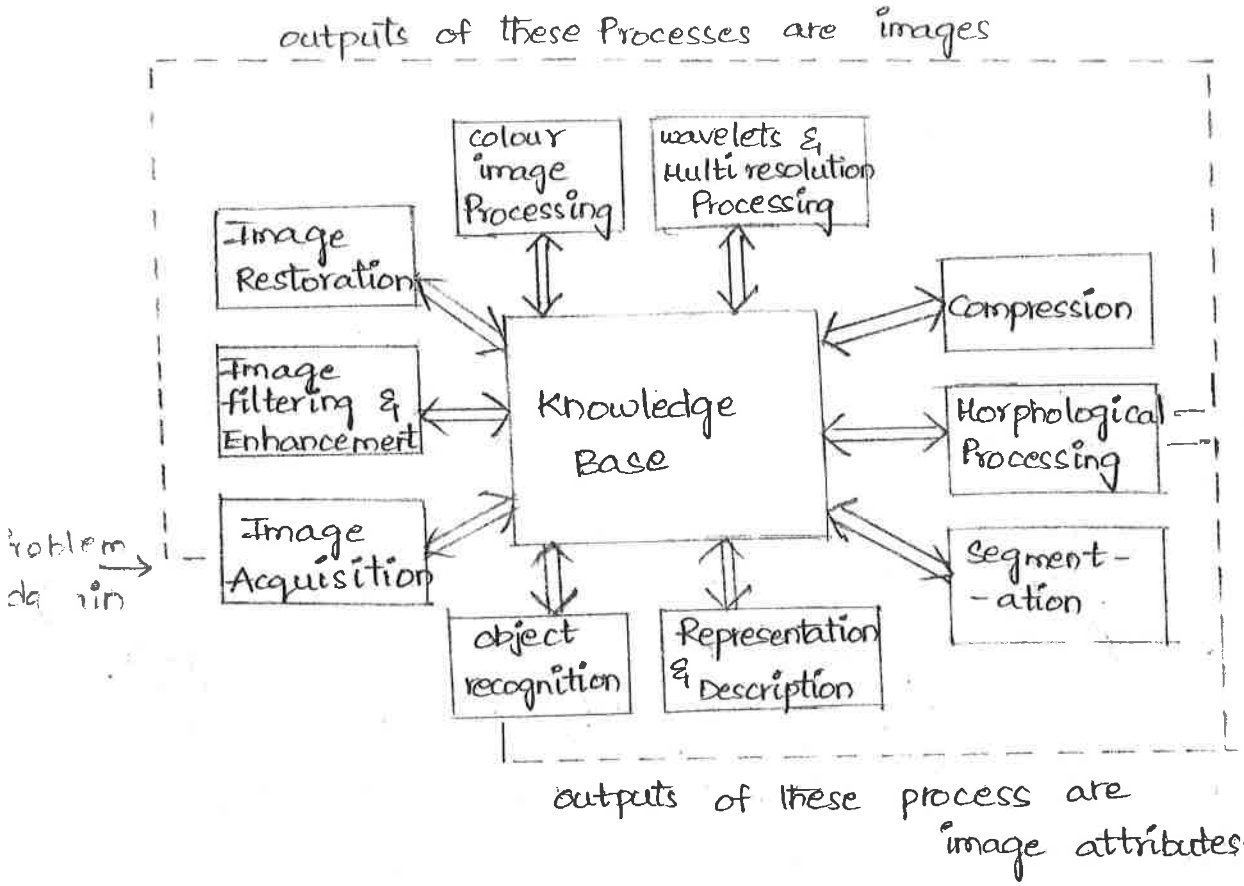
It is the process which also deals with improving the appearance of an image. But this is not similar to Enhancement technique, In Enhancement, we process the image based on human subjective preferences; where as in Restoration technique, we use mathematical or probabilistic models of image degradation ie, we use Transforms (FFT ....), Filters etc..,

Colour Image Processing - This is the area which has been gaining importance because of significant increase in the use of digital images over the internet. By using this colour, human can easily identify and analyse the image. There are different Colour Image Processing techniques.

Wavelets and Multi resolution Processing - Wavelets are the foundation for representing images in various degrees of resolution. There different wavelet transform techniques are used to make the images compress, transmit and analyse easily. Multi resolution theory is concerned with the representation and analysis of images at more than one resolution. We use this process for image data-compression and for pyramidal representation, in which images are sub divided successively into smaller regions.

Compression - This reduces the storage required to save an image, or the band width required to transmit it. We use this in the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of Image file extensions, such as jpg file extension used in JPEG (Joint Photographic Experts Group) image compression standard

Fundamental steps in DIP -



Some of the methods which are mentioned above have images as both input and output, some of the methods have images as their input whose outputs are attributes extracted from those images.

Let us have a brief overview of all the above mentioned process

Image Acquisation - This is the first process ie; we sense the image here. Generally images are generated by the combination of an "illumination" source and the reflection or absorption of energy from the source by the elements of the "scene" being imaged. Generally, the image acquisation stage involves Preprocessing sucha scaling.

Image Enhancement - It is the process of manipulating an image so that the processed image is more suitable than the original image for a specific application. When an image is processed for visual interpretation, the viewer is the ultimate judge of how well

Light and Electromagnetic Spectrum –

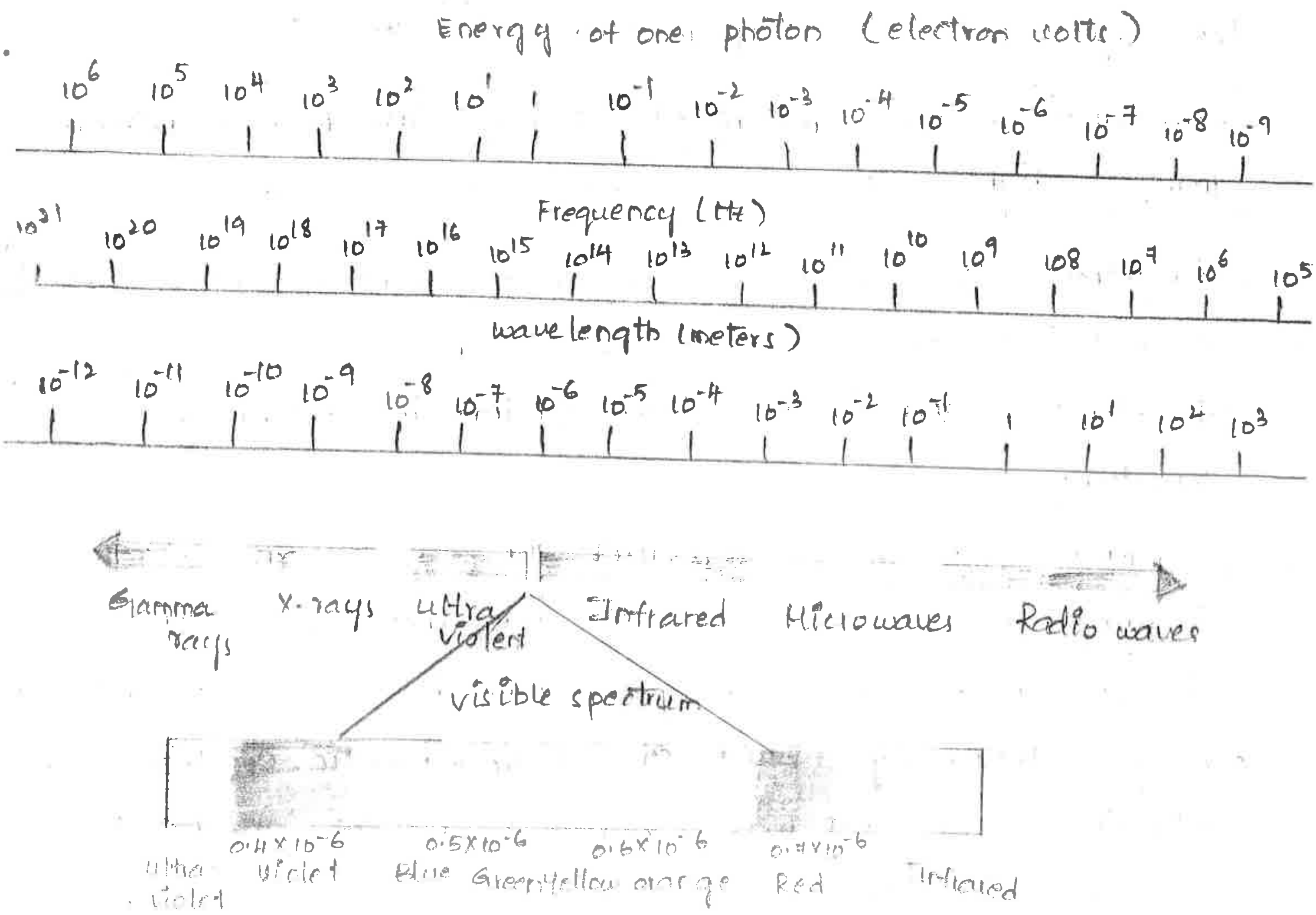
When sunlight is passed through a glass prism, the emerging beam from the glass is a continous spectrum of colours ranging from violet at one end to red at the other end.

Electromagnetic spectrum can be expressed interms of wavelength, frequency or energy. Wavelength and frequency are related by the expression

$$\boxed{\lambda = \frac{c}{v}} \qquad c = 3\times10^{8}\ \text{m/sec.}$$

The energy of the various components of the electromagnetic spectrum is given by

$$\boxed{E = h\nu}$$



Generally, Electromagnetic waves can be viewed as sinusoidal waves with wavelength 'λ' and these can be treated as a stream of massless particle with some bundle of energy. Each bundle of energy is called a "photon".

Light is a particular type of electromagnetic radiation that can be sensed by the human eye. The visible (colour) spectrum is divided in to six broad regions - violet, blue, green, yellow, orange and red.

Light that is void of colour is called Monochromatic light. The intensity of monochromatic light is perceived to vary from black to Grays and finally to white.

To describe the quality of a chromatic light source, three basic quantities are there - Radiance, luminance and brightness.

'Radiance' is the total amount of energy that flows from the light source, usually measured in watts.

'Luminance' is the amount of energy an observer perceives from a light source, usually measured in lumens.

'Brightness' is subjective descriptor of light perception that is practically impossible. It embodies the achromatic notion of intensity and is one of the factors in describing colour sensation.

Gamma radiation is important for medical and astronomical imaging. X-rays are used in industrial applications.

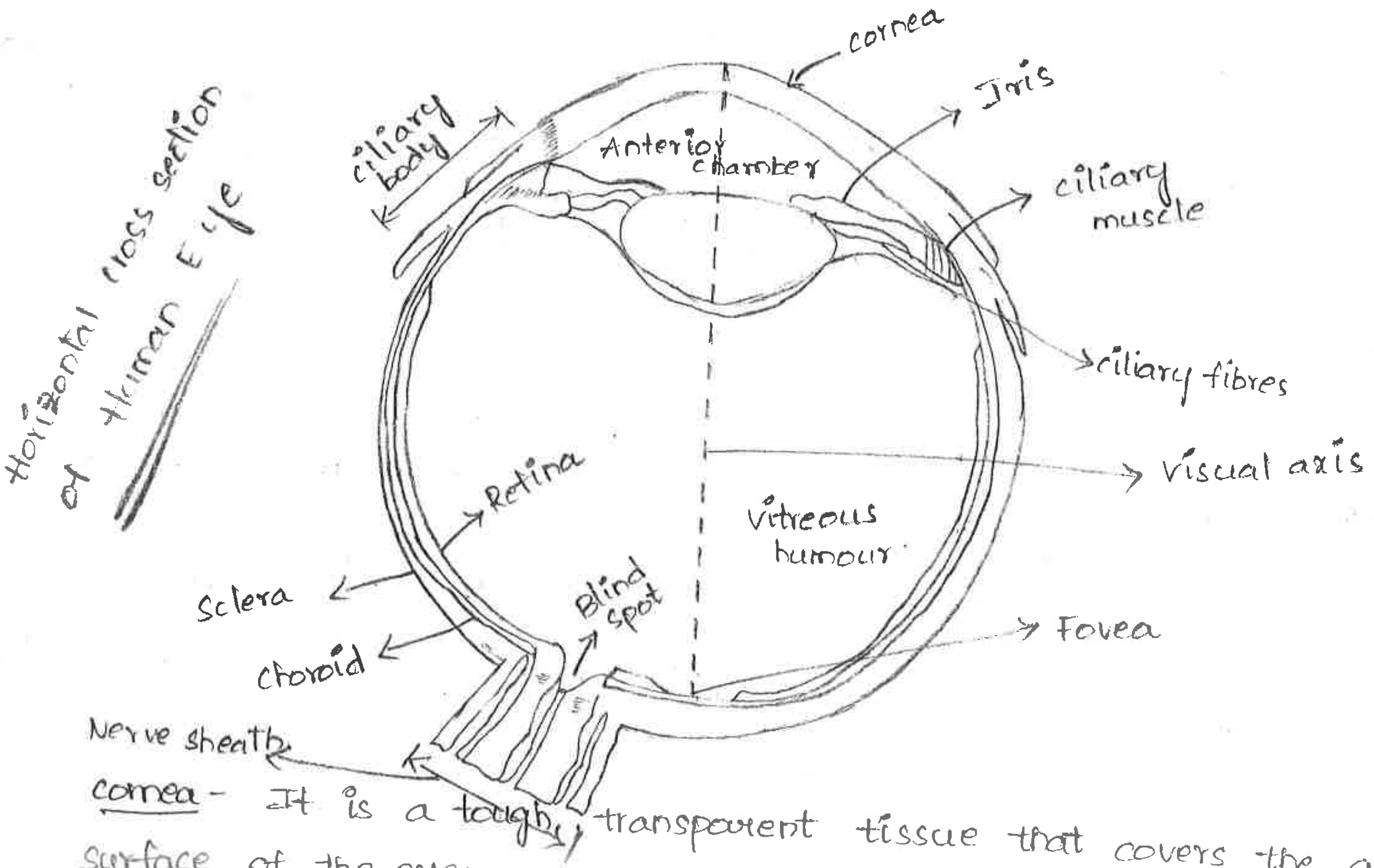
If a sensor can be developed that is capable of detecting energy radiated by a band of electromagnetic spectrum. The wavelength of an EM wave required to see an object must be of same size or smaller than the object.

Elements of visual perception -

Generally human eye has the capability to sense the image and store it. It just acts as a camera. We should know how images are formed and perceived by humans. So, we should understa The human visual perception.

structure of human eye -

The human eye is nearly in the form of sphere with diameter of approximately 20 mm. There are 3 membranes which enclose the eye, they are the 'cornea and sclera', the 'choroid', and 'Retina'



Cornea - It is a tough, transparent tissue that covers the anteric surface of the eye.

Sclera - It is an opaque membrane that makes the eye to be in sphere shape. (optic globe)

choroid - It lies directly below the sclera. This membrane contains a network of blood vessels that serve as a major source of nutrition to the eye. If any injury to choroid, leads to restriction of flow of blood (blood circulation stops).

in to ciliarybody and Iris.

These help the eye to contract or expand inorder to control the amount of light.

<u>lens</u> - lens is made up of concentric layers of fibrous cells and is suspended by the ciliary fibres. It contain 60-70% water, 6%-fat and more protein than any other tissue in the eye.

Generally, these lens are coloured by slightly Yellow pigmentation and the colour increases with the age. In some cases, excessive clouding of lens caused by affliction is referred as 'Cataracts'.

<u>Retina</u> - The inner most membrane of the eye is Retina, which lines inside of wall's entire posterior portion. While viewing an object, the light from the object outside the eye is imaged on retina.

There are 2 receptors
- Cones
- Rods

<u>Cones</u> - These cones are between 6 to 7 million in number. These lies in the central portion of the retina, which are called '<u>Fovea</u>' These are highly sensitive to colour.

<u>Rods</u> - These are larger in number ie, 75 to 150 million are distributed over the retinal surface. These give the overall picture c field of view. These are not involved in colour vision and are sensitive to low levels of illumination.

The absence of receptors results in so-called '<u>Blind Spot</u>'.

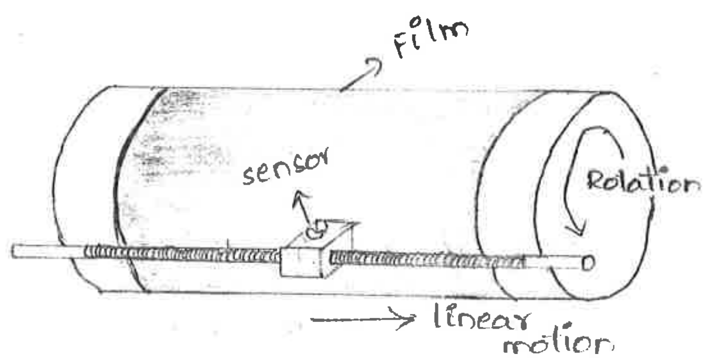
Image sensing and Acquisation-

Generally, images are generated by the combination of an "illumination" source and reflection or absorption of energy from that source by the elements of the "scene" being image. Now we can sence the image.

There are three principal sensor arrangements used to transform illumination energy in-to digital images. The basic idea is "the incoming energy from the source is transformed into the voltage by the combination of input electrical power and sensor material that is responsible to the particular type of energy being detected. The output voltage wave-form is the response of sensor and a digital quantity is obtained from each sensor by digitizing its response".

- Image Acquisation using single sensor-

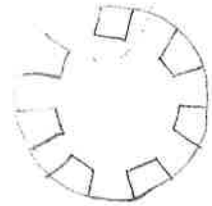
Here, we use single sensor, generally which is a photo--diode, which is constructed of silicon materials and whose output voltage waveform is proportional to light.



To generate a 2-D image, we use a single - sensor, a film. Here, a film negative is mounted on to a drum whose mechanical rotation provides displacement in one dimension. The single sensor is mounted on a lead screw that provides motion in perpendicular direction. This method is very expensive to obtain high-resolution images.

This consists of an in-line arrangement of sensors in the form of a sensor strip. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction.

linear sensor strips

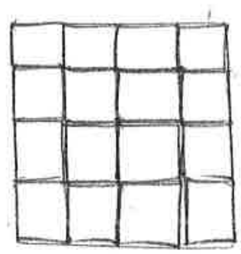


circular sensor strip.

These In-line sensors are used in airborne imaging applications. Sensor strips in ring configuration are used in medical and indus-trial imaging to obtain cross-sectional images of 3D-objects.

- Image Acquisation using Sensor Arrays -

In this, the individual sensors are arranged in the form of a 2D-array. This is also the predominant arrangement found in digital cameras. In cameras, we use CCD array which is used in other light sensing instruments.



sensor arrays.

The main principle in this is to collect incoming energy from the scene element and focus it onto an image plane. If the illumination is light, the front end of the imaging system is an optical lens that projects the viewed scene on-to the lens focal plane. The sensor array, which is coincident with the focal plane, produces output proportional to the integral of the light received at each sensor.

To process an image, it should be in digital form. But the output of most sensors is a continous wave-form. To have a digital image, we have to convert the continous wave-form to digital form. This conversion involves two processes

- Sampling
- Quantization

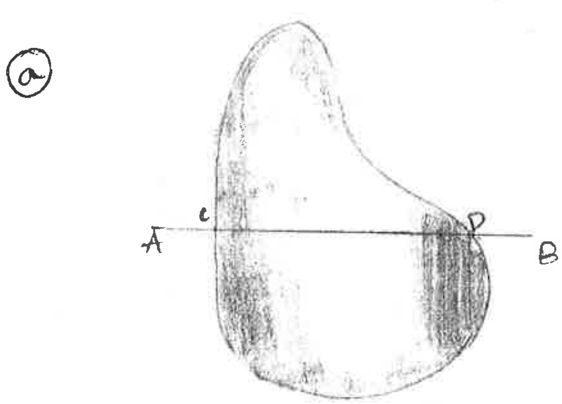
An image may be continous with respect to x and y coordinates and also in amplitude. To convert it in to digital form, we have to sample the function in both coordinates and in amplitude.

Digitizing the x- and y- coordinate values (for 2D) is called "Sampling"

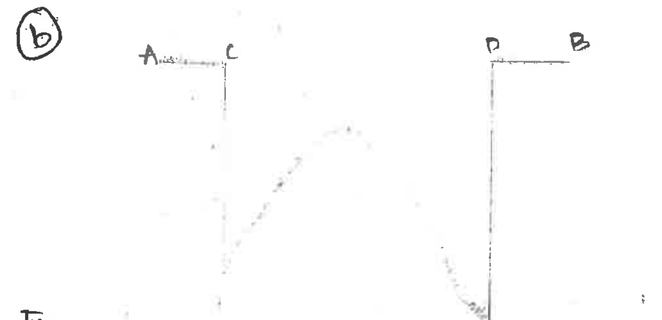
Digitizing along the amplitude values is called "Quantization"

Consider a 2D image,

Here consider along line segment AB.



continous image



From A to C, it is white and then slowly the colour variation occurs due to noise.

equally spaced samples along AB

Digital image along AB

Fig (a) shows the continous image. Fig (b) is the plot of amplitude values of continous image along the line segment 'AB'. There are some random variations which are due to noise.

Fig (c) shows the equally spaced samples along bothe the axis and the intensity scale divided in-to eight discrete intervals ranging from black to white.

Fig (d) shows the digital samples resulting from both sampling and Quantisation.

Representation of Digital images -

Consider f(s,t) is a continous image and it is sampled in to a 2D-array f(x,y), where x & y are coordinates and it contains 'M' rows & 'N' columns.

Generally, we use integer values for discrete coordinates ie, $x = 0, 1, 2 \ldots M-1$ and $y = 0, 1, 2 \ldots N-1$.

Hence the image is represented as

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

Each element of this matrix is called an "image element, picture element, pixel or pel".

The number of intensity levels typically is an integer power of 2.

$\boxed{L = 2^k}$ where 'k' is quantised number of intensity values.

and the interval is $[0 \quad L-1]$ for $M \times N$ 2-d image

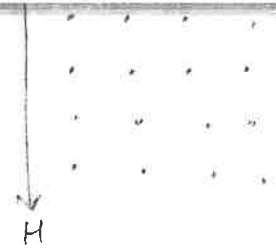
Number of bits required to store a digitized image is

$\boxed{b = M \times N \times k}$.

L = 256 then k = ?

$256 = 2^k$

$\therefore \boxed{k = 8}$ Hence the image is 8-bit image.



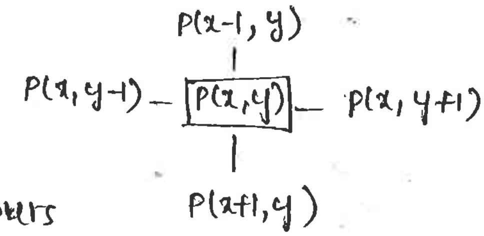
2D - representati[on] of image.

Some Basic Relationships between pixels -

Neighbours -

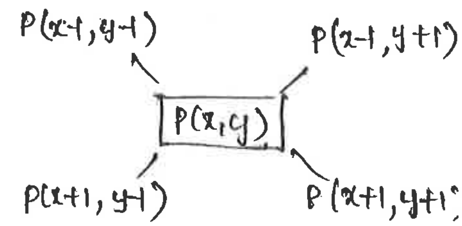
Consider a two dimensional image $f(x,y)$ and let $P(x,y)$ is the centre pixel of that image.

This centre pixel have four neighbours ie, two horizontal and two vertical.



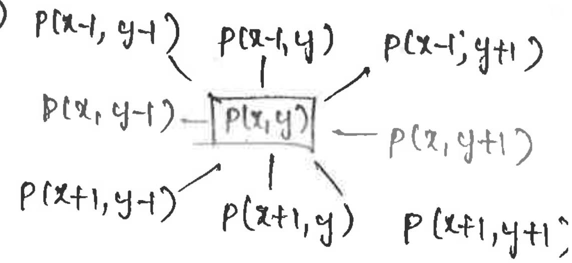
The set of these pixels is called 4-neighbours of $p(x,y)$. and is denoted as "$N_4(P)$"

This center pixel have diagnol pixels ie, it have 4 diagnol pixels and these are denoted as "$N_D(P)$"



Combination of both $N_4(P)$ and $N_D(P)$ ie, two horizontal, two vertical and four diagnol neighbours gives 8-neighbourhood of P, denoted as $N_8(P)$

$$\boxed{N_8(P) = N_4(P) + N_D(P)}$$



- **Adjacency** - Consider 'V' be a set of some intensity values

ⓐ **4-adjacency** - Let P and q are two pixels from 'V' and are said to be 4-adjacent if q is in set $N_4(P)$ $(q \in N_4(P))$

and are said to be 8-adjacent if q is in the set of $N_8(P)$
$(q \in N_8(P))$

(c) <u>m-adjacency</u> - let P and q are two pixel values from 'V' and are said to be m-adjacent if

- q is in $N_4(P)$ or
- q is in $N_D(P)$ and the set ~~$N_4(P) \cap N_D(P$~~ $N_4(P) \cap N_4(q)$ has no pixels whose values are from V.

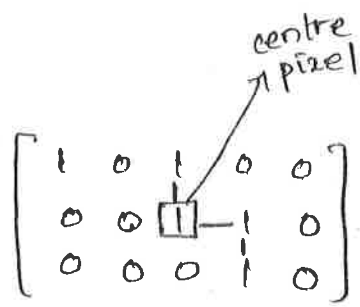
- <u>Connectivity</u> - This connectivity is an important factor in Image processing.

we can connect two pixels when (conditions for connectivity)

* The pixels are adjacency
* And those pixels must have same intensity value ie, same Gray level.

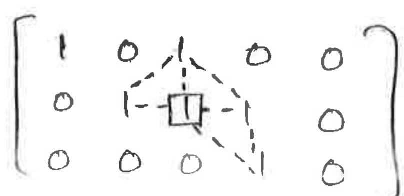
<u>4-connectivity</u> -

Consider's' some set of pixels, Here we can connect the pixels, if they are adjacent, must have same intensity value and $S \in N_4(P)$



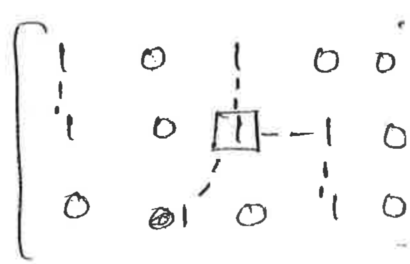
<u>8-connectivity</u> -

Here, we connect the pixels which are adjacent and diagnol having the same intensity values. $(S \in N_D(P))$



<u>M-connectivity</u> -

Intially, we check the adjacent case, if there are adjacent pixels with same intensity value, we connect them. If there are no adjacent pixels, then we check the diagnol pixels and we connect

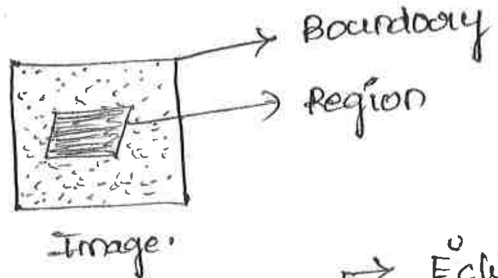


$S \in (N_4(P) \cap N_D(P))$

of intensity values of the image and the pixels must be connected.

Two regions are said to be ~~connected~~ Adjacent, if their union forms a connected set.

- **Boundary** - This should be a set of intensity values in an image and some pixels are not connected.



- Distance Measurement →
  - Euclidean distance
  - D_4 distance (or) city block distance
  - D_8 distance (or) chess board distar.

Consider two pixels $p(x,y)$ and $q(s,t)$

Euclidean distance - The Euclidean distance between P and q is

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

D_4 distance (or) City Block distance - The D_4 distance between P and q is

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

The pixels having a D_4 distance from (x,y) form a diamond centered at (x,y) (less than or equal to some value)

let,

D_4 distance ≤ 2 forms the diamond shape.

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

D_8 distance (or) chess board distance -

The D_8 distance between P and q is given as

$$\boxed{D_8(P,q) = \max(|x-s|, |y-t|)}$$

the pixels with D_8 distance from (x,y) less than or equal to some value 'r' form a square centered at (x,y))

Ex: consider D_8 distance ≤ 2 $\Rightarrow$

This structure looks like chess board.

$$\begin{matrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{matrix}$$

Introduction to the mathematical tools used in DIP -

We can perform different mathematical operations on the images. Some of such operations are

Array versus Matrix Operations -

Generally, array operation is carried out on a pixel-by-pixel basis.

Images can be viewed in the form of Matrices (ie., all set of intensity values) & operations can be performed on Matrix also.

But there is quite difference between Array operations and Matrix operation.

Ex:- Consider two images of size 2×2

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

array Product $= \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$

Matrix product $= \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix}$

To raise the intensity values of the image, we use this operation.

Arithematic operations between images are array operations and these are carried out between corresponding pixel pairs.

Addition - when the two images are added ie, the corresponding pixels will add up.

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

$s(x,y)$ → resultant image

($f(x,y)$ and $g(x,y)$ are two images.)

Here, the resultant image have more intensity values.

Subtraction - when the images are subtracted, the resultant image have low intensity values.

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

Multiplication - when images are multiplied, the corresponding pixels are multiplied. The resultant image have more and more intensity values.

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

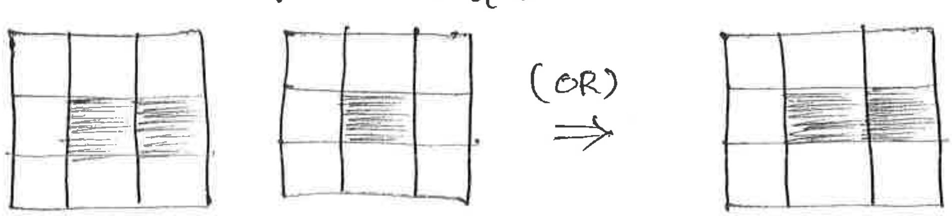
Division - Division operation is performed on the corresponding pixels of two images. Now, the resultant image have very low intensity values.

$$r(x,y) = f(x,y) / g(x,y)$$

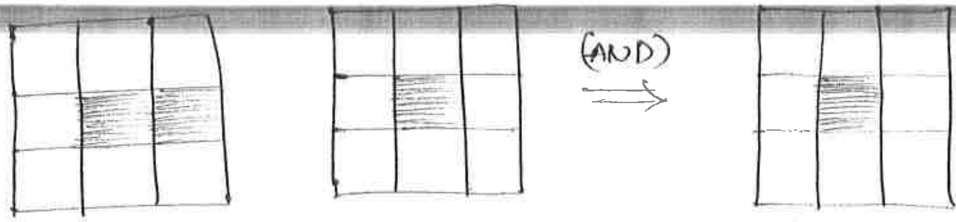
logical operations -

logical operations includes AND, OR, NOT, XOR, NAND etc,

consider two images A & B

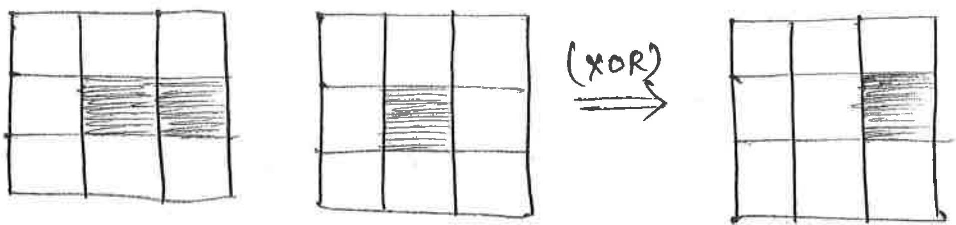


A B A (or) B



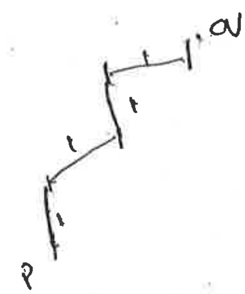
A B A and B

A



A B

Here  represent '1' and  represent '0'.



Transformations applied on an image -

Consider an image ie, $f(x,y)$

If we multiply the image with identity matrix, then there is no change in the resultant image.

$$\underbrace{\begin{bmatrix} x' \\ y' \end{bmatrix}}_{\text{Resultant image}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\text{Identity matrix}} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{Initial image}}$$

$x' = x$; $y' = y$. (Processed one is same as original one)

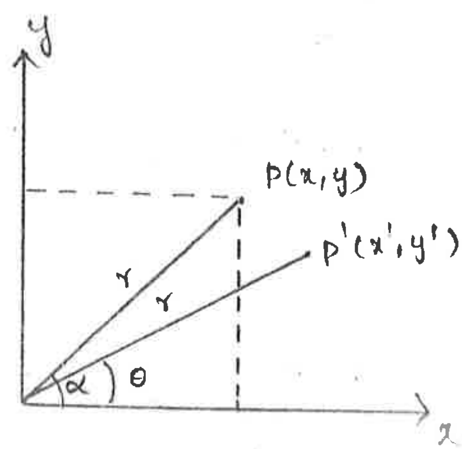
To have image scaling, we multiply the image with some valu.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

like wise image scaling is done.

$x' = c_x x_0$; $y' = c_y y_0$; $1 = 1$

Rotation of an image -



Consider an image in 2-dimensional which is at a distance 'r' and it makes an angle 'α' with x-axis.

later, the image is moved to some other point and it makes an angle 'θ' with x-axis.

From figure,

$$\cos\alpha = \frac{x}{r} \; ; \; \sin\alpha = \frac{y}{r}$$

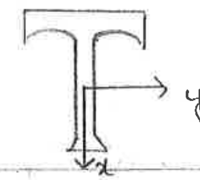
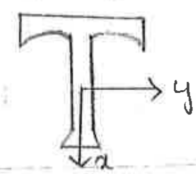
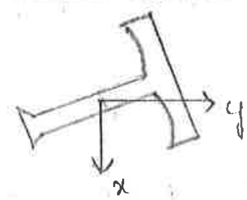
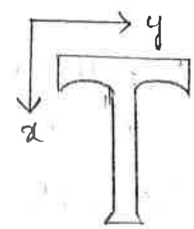
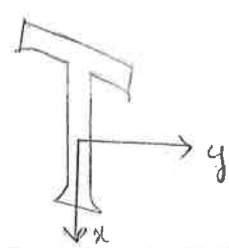
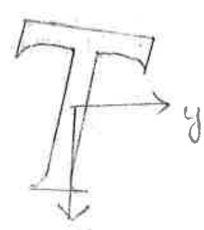
$$y' = r\sin(\alpha-\theta) = \underbrace{r\sin\alpha}_{y}\cos\theta - \underbrace{r\cos\alpha}_{x}\sin\theta = y\cos\theta - x\sin\theta.$$

$$x' = r\cos(\alpha-\theta) = \underbrace{r\cos\alpha}_{x}\cos\theta + \underbrace{r\sin\alpha}_{y}\sin\theta = x\cos\theta + y\sin\theta.$$

$$\begin{bmatrix} y' \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ 1 \end{bmatrix}$$ — Rotation Matrix

The image can be translated by multiplying the image with

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$ — translation matrix.

Transformation name	Transformation matrix	coordinate eqns	Example.
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x_0 c_x$ $y' = y_0 c_y$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta + y\sin\theta$ $y' = y\cos\theta - x\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$.	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$.	

The underlined values are Eigen values, as already determined.

The co-variance matrix C_y consists of significant diagonal values with all other values almost zero, indicating that the element of the transformed vector is made Independent.

Inverse Hotelling transform

$$X = A^T y + m_x$$

$$X_1 = \begin{bmatrix} 0.8165 & 0.5774 & 0.7071 \\ 0.4082 & -0.5774 & -0.7071 \\ 0.4082 & -0.5774 & 0 \end{bmatrix} \begin{bmatrix} -0.8165 \\ -0.1444 \\ -0.3535 \end{bmatrix} + \begin{bmatrix} 3/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix}$$

$$X_2 = A^T \begin{bmatrix} 0 \\ 0.4331 \\ 0.3535 \end{bmatrix} \begin{bmatrix} 3/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1.25 \\ -0.25 \\ 0 \end{bmatrix}$$

lly x_3, x_4 are computed as

$$X_3 = A^T \begin{bmatrix} 0.4082 \\ -0.144 \\ -0.3535 \end{bmatrix} + \begin{bmatrix} 3/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 0.4 + 3/4 \\ 0.49 + 0.25 \\ 0.3 + 0.25 \end{bmatrix} = \begin{bmatrix} 0.75 \\ 0.75 \\ 0.55 \end{bmatrix}$$

$$X_4 = A^T \begin{bmatrix} 0.4082 \\ -0.1444 \\ -0.3535 \end{bmatrix} + \begin{bmatrix} 3/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 0.49 + 3/4 \\ 0 + 1/4 \\ 0.25 + 1/4 \end{bmatrix} = \begin{bmatrix} 1.25 \\ 0.25 \\ 0.5 \end{bmatrix}$$

Applications of KL Transform

1) Binary image alignment
2) Image Compression.

$$+\begin{bmatrix}0\\0.4331\\0.3535\end{bmatrix}\begin{bmatrix}0 & 0.4331 & 0.3535\end{bmatrix}+\begin{bmatrix}0 & 0 & 0\\0 & 0.188 & 0.1531\\0 & 0.153 & 0.125\end{bmatrix}$$

$$+\begin{bmatrix}0.4082\\-0.1444\\-0.3535\end{bmatrix}\begin{bmatrix}0.4082 & -0.144 & -0.3535\end{bmatrix}+\begin{bmatrix}0.167 & -0.06 & -0.144\\-0.06 & 0.02 & 0.05\\-0.14 & 0.05 & 0.125\end{bmatrix}$$

$$+\begin{bmatrix}0.167 & -0.06 & 0.144\\-0.06 & 0.02 & -0.05\\0.14 & -0.05 & 0.125\end{bmatrix}+\begin{bmatrix}0.4082\\-0.1444\\0.3535\end{bmatrix}\begin{bmatrix}0.4082 & -0.1444 & 0.3535\end{bmatrix}.$$

$$C_y = \frac{1}{4}\begin{bmatrix}0.994 & 0 & 0\\0 & 0.248 & 0.2\\0 & 0.2 & 0.5\end{bmatrix}$$

Covarience matrix of y can also be computed using AC_xA^T
'C_x' is the covarience matrix of X.

$$C_x = \begin{bmatrix}3/16 & 1/16 & 1/16\\1/16 & 3/16 & -1/16\\1/16 & 1/16 & 1/16\end{bmatrix}$$

A is the transformation matrix

$$A = \begin{bmatrix}0.8165 & 0.4082 & 0.4082\\0.5774 & -0.5774 & -0.5774\\0.7071 & -0.7071 & 0\end{bmatrix}$$

$$C_y = AC_xA^T = \begin{bmatrix}\underline{0.24998} & 0 & 0.03\\0 & \underline{0.0625} & 0.10\\0.07 & 0.05 & \underline{0.125}\end{bmatrix}$$

Image transforms: Image transform is basically a representation of an image. There are two for transforming an image from one representation to another. First the transformation may isolate critical components of the image pattern so that they are directly accessible for analysis. Second, the transformation may place the image data in a more compact form so that they can be stored and transmitted efficiently.

Types of Image transforms

1) Fourier transform
2) Walsh transform
3) Hadamard transform
4) slant transform
5) Discrete cosine transform
6) KL transform
7) Haar transform
8) Discrete sine transform

classification of Image transforms

Image transforms can be classified based on the nature of the basis function as

i) transforms with orthogonal basis functions
ii) transforms with non sinusoidal orthogonal basis functions
iii) transforms whose basis function depend on the statics of input data.
iv) Directional transforms.

Image transforms

- orthogonal sinusoidal basis function
- Non sinusoidal orthogonal basis function.
- Basis function depending on statistics of i/p signal
- Directional transformation

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 input signal

* KL transform
* singular value decomposition.

Directional transformation

* Hough transform
* Random transform
* Ridgelet transform
* Contourlet transform.

Introduction:

Image transforms are extensively used in image processing and image analysis. Transform is basically a mathematical tool, which allows us to move from one domain to another domain (time ~~freque~~ domain to frequency domain). The reason to migrate from one domain to another domain is to perform the task in an easier manner

Need for transform:

Transform is basically a mathematical tool to represent a signal. The need for transform is given as follows

* Mathematical Convenience
* To extract more information.

1) Mathematical convenience: Every action in time domain will have an impact in the frequency domain.

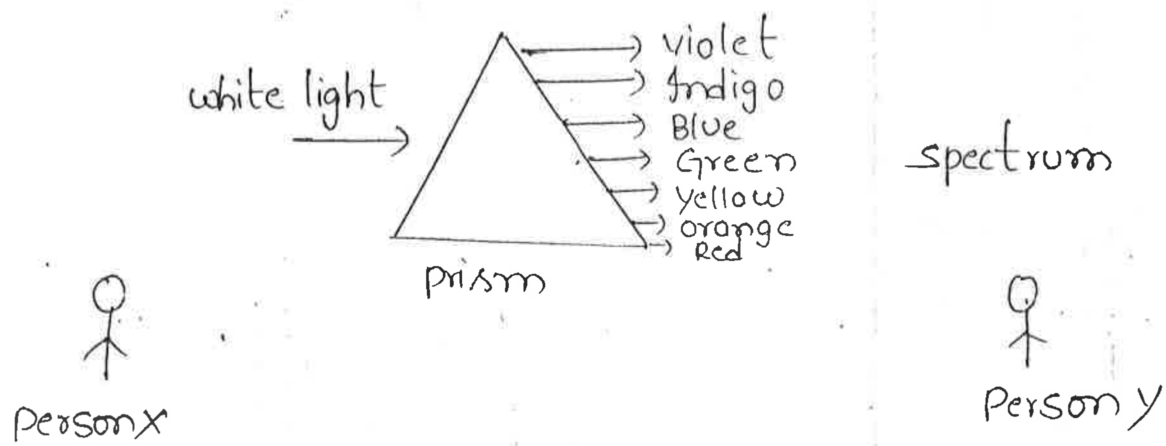
Convolution in time domain <—> Multiplication in frequency domain

2) To extract more information: Transforms allow us to extract more information. Consider the following Ex

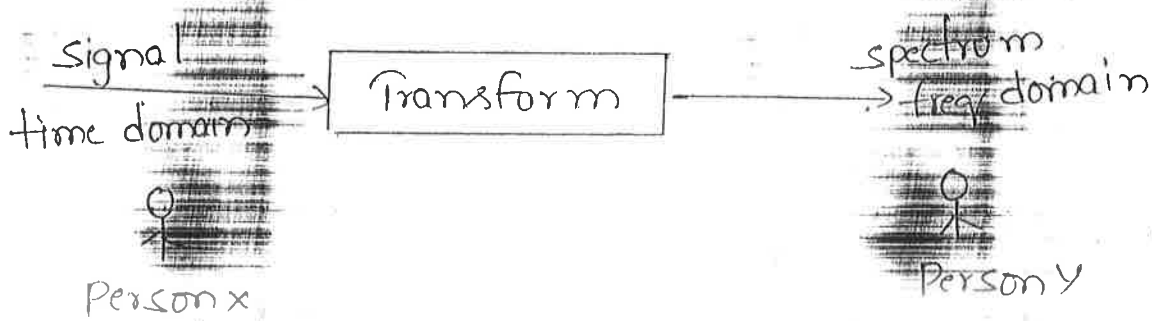
person x is on the left-hand side of the prism. where as the person y is on the right-hand side of the prism as illustrated in fig.

person x sees the light as white light where as the person y sees the white light as a combination of seven colors (VIBGYOR)

Obviously the person y is getting more information than the person x by using the prism. Similarly a transform is a tool that allows one to extract more information from a signal. Here the person x is in the time domain and the person y is in the frequency domain. The tool which allows us to move from time domain to frequency domain.



spectrum of white light.



Concept of transformation.

-) The transform which is widely used in the field of image compression is discrete cosine transform.

-) The Haar transform is the simplest example of a wavelet transform.

-) One of the important advantages of wavelet transform is that signals can be represented in diff resolutions.

-) The KL transform is considered to be the best among all linear transforms with respect to energy compaction.

Fourier transform for 1D

Fourier transform is widely used in the field of image processing. An image is a spatially varying function. One way to analyse spatial variations is to decompose an image into a set of orthogonal functions. A fourier transform is used to transform an intensity image into domain of spatial frequency.

Let us assume continous function $f(x)$. The variable x represents distance. The fourier transform of continous function is denoted as $F(u)$, where u represents the spatial frequency.

$$F(u) = \int_{-\alpha}^{\alpha} f(x)\,[\cos(2\pi ux) - \sin(2\pi ux)]\, dx$$

This can be expressed in consise manner in exponential form

$$F(u) = \int_{-\alpha}^{\alpha} f(x)\, e^{-2\pi ux}\, dx$$

Inverse fourier transform for 1D

$$F(u) = \int_{-\alpha}^{\alpha} f(x)\left[\cos(2\pi ux) + j\sin(2\pi ux)\right] dx.$$

In exponential form, it is expressed as

$$f(x) = \int_{-\alpha}^{\alpha} F(u)\, e^{+j2\pi ux}\, du$$

The fourier transform can be extended to 2D functions also.

Fourier transform for 2-D

$$F(u,v) = \int_{-\alpha}^{\alpha}\int_{-\alpha}^{\alpha} f(x,y)\, e^{-j2\pi(ux+vy)}\, dx\, dy$$

Inverse fourier transform for 2-D

$$f(x,y) = \int_{-\alpha}^{\alpha}\int_{-\alpha}^{\alpha} F(u,v)\, e^{+j2\pi(u_x+vy)}\, du\, dv$$

Discrete Fourier transform

since the images are digitized it is necessary to have a discrete formulation of the fourier transform. This is achieved by the Discrete fourier transform (DFT), which takes regularly spaced data values, and return the value of the Fourier transform by replacing the integral by a summation.

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.

Separable property : This property allows a 2D transform to be computed in two steps by successive 1D operations on rows and columns of an image.

Mathematically it is represented as

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

$$= \frac{1}{M} \sum_{x=0}^{M-1} e^{\frac{-j2\pi ux}{M}} \frac{1}{N} \sum_{y=0}^{N-1} f(x,y)\, e^{\frac{-j2\pi vy}{N}}$$

$$= \frac{1}{M} \sum_{x=0}^{M-1} e^{\frac{-j2\pi ux}{M}} f(x,v)$$

$$= F(u,v)$$

Note: The location of the factor $\frac{1}{MN}$ does not matter as some authors use it as part of inverse transform instead of the forward transform.

DFT for one-dimensional

$$F(U) = \sum_{x=0}^{M-1} f(x) \left[\cos\left(\frac{2\pi Ux}{M}\right) - j \sin\left(\frac{2\pi Ux}{M}\right) \right]$$

In exponential form

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

Inverse DFT for one-dimensional

$$F(x) = \sum_{x=0}^{m-1} f(u) \left[\cos\left(\frac{2\pi Ux}{m}\right) + i \sin\left(\frac{2\pi Ux}{m}\right) \right]$$

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

DFT for Two-dimensional

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

for $U = 0 \cdots M-1$, & $V = 0 \cdots N-1$.

Inverse DFT for two-dimesional

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

If Images are sampled in square array for $M=N$

$$F(U,V) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(Ux+Vy)/N}$$

$$f(x,y) = \sum_{U=0}^{N-1} \sum_{V=0}^{N-1} F(U,V)\, e^{j2\pi(Ux+Vy)/N}$$

spatial shift property

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

proof: Adding and subtracting x_0 to $e^{\frac{-j2\pi ux}{N}}$ in equation

$$DFT\left[f(x-x_0,y)\right] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)\, e^{\frac{-j2\pi u(x-x_0+x_0)}{M}}\, e^{\frac{-j2\pi vy}{N}}$$

→ splitting $e^{\frac{-j2\pi u(x-x_0+x_0)}{N}}$ into $e^{\frac{-j2\pi(x-x_0)u}{M}}\, e^{\frac{-j2\pi ux_0}{M}}$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)\, e^{\frac{-j2\pi(x-x_0)u}{M}}\, e^{\frac{-j2\pi ux_0}{M}}\, e^{\frac{-j2\pi vy}{N}}$$

$$= \left[\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)\, e^{\frac{-j2\pi(x-x_0)u}{M}}\, e^{\frac{-j2\pi vy}{N}}\right] e^{\frac{-j2\pi ux_0}{M}} \quad \text{— (1)}$$

from the definition of 2D- DFT we can write

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)\, e^{\frac{-j2\pi(x-x_0)u}{M}}\, e^{\frac{-j2\pi vy}{N}} = F(u,v) \quad \text{— (2)}$$

sub (2) in (1) we get

$$DFT\left[f(x-x_0),y\right] = e^{\frac{-j2\pi ux_0}{M}}\, F(u,v)$$

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

Periodicity property

The 2D-DFT of a function $f(x,y)$ is said to be periodic with a period N if

$$F(u,v) \longrightarrow F(u+pM, v+qN) \quad \text{--- (1)}$$

Proof:

$$F(u+pM, v+qN) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{\frac{-j2\pi x}{M}(u+pM)}\, e^{\frac{-j2\pi y}{N}(v+qN)} \rightarrow (2)$$

$$F(u+pM, v+qN) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{\frac{-j2\pi ux}{M}}\, e^{\frac{-j2\pi xpM}{M}}\, e^{\frac{-j2\pi yv}{N}}\, e^{\frac{-j2\pi yqN}{N}} \rightarrow (3)$$

By taking $e^{\frac{-j2\pi ux}{M}}$, $e^{\frac{-j2\pi vy}{N}}$

$$F(u+pM, v+qN) = \left[\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{\frac{-j2\pi ux}{M}}\, e^{\frac{-j2\pi vy}{N}}\right] e^{-j2\pi xp}\, e^{-j2\pi yq} \rightarrow (4)$$

we know

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{\frac{-j2\pi ux}{M}}\, e^{\frac{-j2\pi vy}{N}} \rightarrow (5)$$

(5) in (4) we get

$$F(u+pM, v+qN) = F(u,v)\, e^{-j2\pi xp}\, e^{-j2\pi yq}$$

The $e^{-j2\pi xp}$, $e^{-j2\pi yq}$ values are always '1' for any integer value of x, p, q and y.

$$\therefore F(u+pM, v+qN) = F(u,v) \times 1$$

$$F(u+pM, v+qN) = F(u,v)$$

Convolution property

Convolution is one of the most powerful operations in digital image processing. Convolution in spatial domain is equal to multiplication in freq domain.

Convolution of 2 sequences $x(n)$ and $h(n)$ is

$$x(n) * h(n) = \sum_{k=-\alpha}^{\alpha} x(k)\, h(n-k)$$

Two-dimensional convolution of two arrays (or) matrices $f(x,y)$ and $g(x,y)$ is given as.

$$f(x,y) * g(x,y) = \sum_{a=0}^{m-1} \sum_{b=0}^{N-1} f(a,b)\, g(x-a, y-b) \quad \text{--- (1)}$$

Proof: DFT of convolution of 2 sequences $f(x,y)$ and $g(x,y)$ is given by

$$DFT\{f(x,y) * g(x,y)\} = \sum_{x=0}^{m-1} \sum_{y=0}^{N-1} \left\{ \sum_{a=0}^{m-1} \sum_{b=0}^{N-1} f(a,b)\, g(x-a,y-b) \right\} e^{\frac{-j2\pi ux}{M}} e^{\frac{-j2\pi vy}{N}} \quad \text{--- (2)}$$

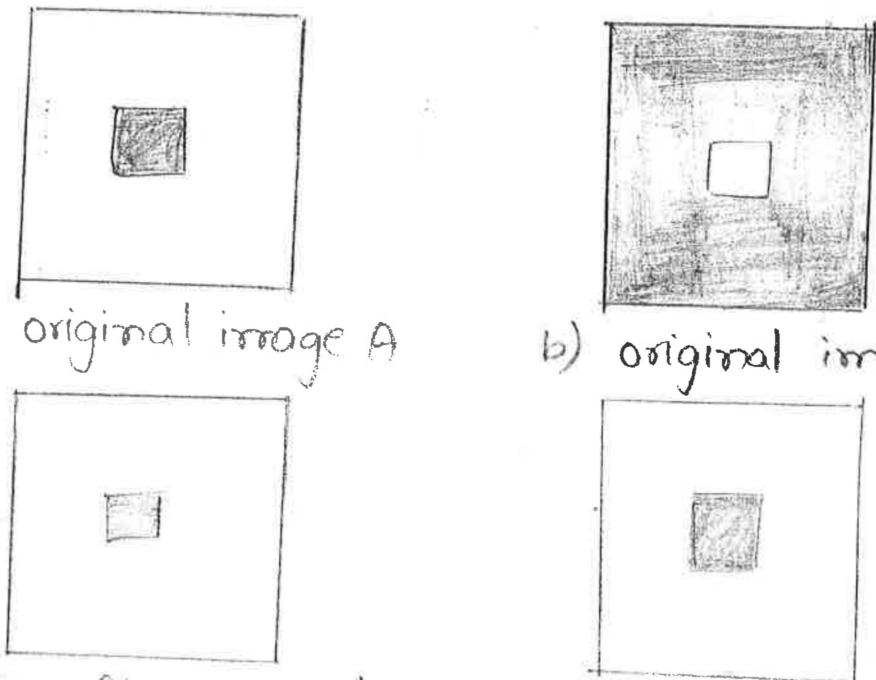
$$= \sum_{x=0}^{m-1} \sum_{y=0}^{N-1} \sum_{a=0}^{m-1} \sum_{b=0}^{N-1} f(a,b)\, g(x-a,y-b)\, e^{\frac{-j2\pi}{M}(x-a+a)u}\, e^{\frac{-j2\pi}{N}(y-b+b)v} \quad \text{--- (3)}$$

$$= \sum_{x=0}^{m-1} \sum_{y=0}^{N-1} \sum_{a=0}^{m-1} \sum_{b=0}^{N-1} f(a,b)\, g(x-a,y-b)\, e^{\frac{-j2\pi u(x-a)}{m}}\, e^{\frac{-j2\pi a u}{m}}\, e^{\frac{-j2\pi v(y-b)}{N}}\, e^{\frac{-j2\pi vb}{N}}$$

$$= \underbrace{\sum_{a=0}^{m-1} \sum_{b=0}^{N-1} f(a,b)\, e^{\frac{-j2\pi au}{m}}\, e^{\frac{-j2\pi bv}{N}}}_{F(u,v)} \; \underbrace{\sum_{x=0}^{m-1} \sum_{y=0}^{N-1} g(x-a,y-b)\, e^{\frac{-j2\pi u(x-a)}{m}}\, e^{\frac{-j2\pi v(y-b)}{N}}}_{G(u\,v)}$$

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

The convolution theorem tells us that the convolution of two functions in the spatial domain correspons to multiplication in the freq domain and vice-versa.



(a) original image A

b) original image B

(c) image after Convolution operation

(d) Image after spectral multiplication.

Correlation property: Correlation is basically used to find the relative similarity between two signals. The process of finding similarity of a signal to itself is auto Correlation, where as the process of finding of the similarity between two different signals is cross correlation

Proof: The DFT of correlation of two sequences $x(n)$ and $y(n)$ is defined as

$$DFT\{R_{x,h}\} = \sum_{m=0}^{N-1}\left\{\sum_{n=0}^{N-1} x(n)\left(h(n+m)\right)\right\} e^{\frac{-j2\pi}{N}mk} \quad \text{--- (1)}$$

Here $R_{x,h}$ denotes the correlation b/w signals $x(n)$ & $h(n)$.

By adding & subtracting to the power of the exponential term $e^{\frac{-j2\pi}{N}mk}$ in eq (1) we get.

$$DFT\{R_{x,h}\} = \sum_{m=0}^{N-1}\left\{\sum_{n=0}^{N-1} x(n)\,h(n+m)\right\} e^{\frac{-j2\pi}{N}(m+n-n)k} \quad \text{--- (2)}$$

$e^{\frac{-j2\pi}{N}(m+n-n)k}$ in to $e^{\frac{-j2\pi(m+n)k}{N}}$ & $e^{+j\frac{2\pi nk}{N}}$

$$= \sum_{m=0}^{N-1}\left\{\sum_{n=0}^{N-1} x(n)\,h(n+m)\right\} e^{\frac{-j2\pi(m+n)k}{N}}\, e^{j\frac{2\pi}{N}nk}$$

from the definition of DFT, we can write.

$$DFT\{R_{x,h}\} = H(k)\sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi}{N}n(-k)}$$

which is reduced to

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

The correlation property tell us that the correlation of two sequences in time domain is equal to the multiplication of DFT of one sequence and time reversal of the DFT. of another sequence in the frequency domain.

Scaling property:-

Scaling is basically used to increase ~~aux~~ or decrease the size of an image. According to this property, the expansion of a signal in one domain is equal to compression of the signal in another domain.

The 2D DFT of a function $f(m,n)$ is defined as

$$f(m,n) \xrightarrow{DFT} F(k,l)$$

If DFT of $f(m,n)$ is $F(k,l)$ then $DFT\left[f(am,bn)\right]$

$$= \frac{1}{|ab|} F(k/a, l/b)$$

Proof:- DFT of func^n $f(am,bn)$ is given by

$$DFT\{f(am,bn)\} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(am,bn)\, e^{\frac{-j2\pi}{N}mk}\, e^{\frac{-j2\pi}{N}nl} \quad \text{— (1)}$$

Mul & div the power of exponential term

$e^{\frac{-j2\pi}{N}mk}$ with 'a'

$e^{\frac{-j2\pi}{N}nl}$ with 'b'

$$DFT\{f(am,bn)\} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(am,bn)\, e^{\frac{-j2\pi}{N}mk(a/a)}\, e^{\frac{-j2\pi}{N}nl(b/b)} \quad \text{— (2)}$$

$$DFT\{f(am,bn)\} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(am,bn)\, e^{\frac{-j2\pi}{N}ma(k/a)}\, e^{\frac{-j2\pi}{N}nb(l/b)} \quad \text{— (3)}$$

By sub (3) in

$$F(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\, e^{\frac{-j2\pi}{M}mk}\, e^{\frac{-j2\pi}{N}nl}$$

we get

$$DFT\{f(am, bn)\} = \frac{1}{ab} F(k/a, l/b)$$

The scaling theorem tell us that compression in one domain produces a corresponding expansion in the fourier domain and vice versa.

Conjugate symmetry

If the DFT of $f(m,n)$ is $F(k,l)$ then the

$$DFT[f^*(m,n)] = F^*(-k,-l) \quad — (1)$$

Proof: The DFT of function $f(x,y)$ is defined as

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

By applying complex conjugate to $F(k,l)$ we get

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

By applying reversal to $F^*(k,l)$ in eq (2) we get

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

By applying reversal to $F^*(k,l)$ in eq (2) we get

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

Orthogonality property:

The orthogonality property of a 2D DFT is given as.

$$\frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{k,l}(m,n)\, a^{*}_{k',l'}(m,n) = \delta(k-k',\ l-l') \quad (1)$$

where $\delta(k-k', l-l')$ is the kronecker delta. This orthogonality condition can be used to derive the formula for the IDFT from the definition of the DFT.

Multiplication by Exponential :-

If the DFT of $f(m,n)$ is $F(k,l)$ then

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

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

Proof:- from the definition of a 2D-DFT, we can write

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

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

By combining $e^{j\frac{2\pi}{N}mk_0}$, $e^{-j\frac{2\pi}{N}mk}$ and $e^{-j\frac{2\pi}{N}nl}$, $e^{j\frac{2\pi}{N}nl_0}$ into a single exponential function in eq-(2) we get

$$DFT\left[e^{\frac{j2\pi}{N}mk_0} e^{\frac{j2\pi}{N}ml_0} f(m,n)\right]$$

$$= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n)\, e^{\frac{j2\pi}{N}m(k-k_0)}\, e^{\frac{j2\pi}{N}n(l-l_0)} \quad \text{--- (3)}$$

By sub 3 in

$$F(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, e^{\frac{-j2\pi}{N}mk}\, e^{\frac{-j2\pi}{N}nl}$$

This theorem proves that multiplication of a function $f(m,n)$ with an exponential in the spatial domain leads to a freq shift we get

$$DFT\left[e^{\frac{j2\pi}{N}mk_0} e^{\frac{j2\pi}{N}nl_0} f(m,n)\right] = F(k-k_0, l-l_0)$$

Rotation property:

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

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

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

$$DFT\left[f(r\cos(\theta+\theta_0), r\sin(\theta+\theta_0))\right]$$

$$\rightarrow F\left[R\cos(\phi+\phi_0), R\sin(\phi+\phi_0)\right]$$

Property	Sequence	Transform
spatial shift Property	$f(x-x_0, y)$	$e^{\frac{-j2\pi u x_0}{m}} F(u,v)$
Periodicity	$F(k+pN, l+qN)$	$= F(k,l)$
Convolution	$f(m,n) * g(m,n)$	$F(k,l) \times G(k,l)$
scaling	$f(am, bn)$	$\frac{1}{\lvert ab \rvert} F(k/a, l/b)$
Conjugate symmetry	$F(k,l)$	$= F^*(-k,-l)$
multiplication by exponential	$e^{j\frac{2\pi}{N}mk_0} e^{j\frac{2\pi}{N}ml_0} f(m,n)$	$F(k-k_0, l-l_0)$
Rotation Property	$f(r\cos(\theta+\theta_0), r\sin(\theta+\theta_0))$	$F[R\cos(\phi+\phi_0), R\sin(\phi+\phi_0)]$

Example :

Compute 2D DFT of the 4x4 gray scale image given below

$$\underset{\text{Image}}{f[m,n]} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Sol:- The 2D-DFT of the image $f(m,n)$ is rep as $F[k,l]$

$$F[k,l] = \text{kernel} \times f[m,n] \times (\text{kernel})^T \quad \text{--- (1)}$$

The kernel or basis of the fourier transform for $N=4$ is given by

The DFT basis for $N=4$ is given by $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$ --- (2)

sub (2) in (1) we get

$$F(k,l) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$F(k,l) = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Walsh transform :-

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

1-Dimensional walsh kernal

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

1-Dimensional transform of walsh

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)\, g(x,u)$$

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

2-Dimensional walsh kernal

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

2-D walsh transform

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\, g(x,y,u,v)$$

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

1-d inverse walsh kernal

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

1-d inverse walsh transform

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

2-d Inverse walsh kernal

$$h(x,y,u,v) = \frac{1}{N}\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u) + b_i(v)\,b_{n-1-i}(v)}$$

2-d inverse walsh transform

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} f(u,v)\; h(x,y,u,v)$$

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

Ex: Walsh transform for N = 4

$N = 4$

$N = 2^n$

$n = 2$

Decimal value

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	$b_1(3)=1$	$b_0(3)=1$

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

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

$$=\frac{1}{4}\left\{(-1)^{b_0(0)\,b_1(0)}\times(-1)^{b_1(0)\,b_0(0)}\right\}$$

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

Similarly

$$g(0,1)=\frac{1}{4}\left\{\prod_{i=0}^{1}(-1)^{b_0(0)\,b_1(1)}\times(-1)^{b_1(0)\,b_0(1)}\right\}=1/4$$

$$g(1,0)=\frac{1}{4}\left\{\prod_{i=0}^{1}(-1)^{b_0(1)\,b_1(0)}\times(-1)^{b_1(1)\,b_0(0)}\right\}=1/4$$

$$g(1,1)=\frac{1}{4}\left\{\prod_{i=0}^{1}(-1)^{b_0(1)\,b_0(1)}\times(-1)^{b_1(1)\,b_0(1)}\right\}=1/4$$

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

$$=\frac{1}{4}\left\{(-1)^{b_0(0)\,b_1(2)}\times(-1)^{b_1(0)\,b_0(2)}\right\}=1/4$$

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

$$= \frac{1}{4}\{(1)\times(-1)\} = -\frac{1}{4}$$

By calculating all the values in similar manner we get

$u \backslash x$	0	1	2	3	Sequency
0	$1/4$	$1/4$	$1/4$	$1/4$	zero sign change
1	$1/4$	$1/4$	$-1/4$	$-1/4$	one sign change
2	$1/4$	$-1/4$	$1/4$	$-1/4$	Three sign changes
3	$1/4$	$-1/4$	$-1/4$	$1/4$	Two sign changes.

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

→ Walsh transform for N=8. It holds the same magnitude But it is lengthy process. So we go for short cut method

Algorithm for short cut method of Walsh transform

1) find the binary representation of x.
2) find the binary values of u and consider those values in reverse binary form.
3) check for the number of overlaps of 1 between u and x.
4) If the number of overlaps of between n and k
   i) zero overlaps then the sign is positive.
   ii) Even number of overlaps then the sign is positive.
   iii) odd then the sign is negative

For Example we can go for $x=4$ and $u=3$

Step 1:- write the binary representation of $x=4$. and its binary representation is 100

Step 2:- write the binary representation of $u=3(011)$ in reverse order and it is of 110 (by reversing).

Step 3:- check for the number of overlaps between n and k

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

only one overlap;

It means odd num of overlaps so the sign is 'negative'

for $N = 8$

$$g(x,u) = \begin{bmatrix} +\frac{1}{8} & +\frac{1}{8} & \frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} \\ +\frac{1}{8} & +\frac{1}{8} & \frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ +\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ +\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} \\ +\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} \\ +\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} \\ +\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} \\ +\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} & +\frac{1}{8} & +\frac{1}{8} & -\frac{1}{8} \end{bmatrix}$$

Advantage of walsh transform

The advantage of walsh transform is fourier transform is based on the trignometric terms, where as walsh transform consists of a series expansion of basis functions whose values are only +1 & −1. These functions can be implemented more efficiently in a digital environment than the exponential basis functions of the fourier transform.

Hadamard transform :-

The Hadamard transform is basically the same the walsh transform except the rows of the transform matrix are re-ordered. The Elements of the mutually orthogonal basis vectors of a Hadamard transform are either +1 & -1, which results in very low Computational Complexity in the calculation of the transform coefficients. Hadamard matrices are easily constructed for $N = 2^n$.

one-dimensional Hadamard kernal

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

1-D hadamard transform

$$F(u) = \sum_{x=0}^{N-1} f(x)\, g(x,u)$$

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

1-D Hadamard inverse kernal

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

$$f(x) = \sum_{u=0}^{n-1} F(u)\, (-1)^{\sum_{i=0}^{n-1} b_i(x)\, b_i(u)}$$

2-D Hadamard kernal

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

2-D. Hadamard transform

$$f(u,v) = \sum_{i=0}^{n-1} f(x,y)\; g(x,y,u,v)$$

$$= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\; (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u) + b_i(y) b_i(v)}$$

2-D Hadamard Inverse kernal

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

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} f(u,v)\; h(x,y,u,v)$$

$$= \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} (-1)^{\sum_{i=0}^{n-1} b_i(x)\; b_i(u) + b_i(y) b_i(v)}$$

Ex:- for $N=2$ Hadamard kernal

$N=2$

$n = \log_2^N = 1$

$$g(x,u) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} b_i(x)\; b_i(u)}$$

$$= \frac{1}{2} (-1)^{\sum_{i=0}^{0} b_0(x) b_0(u)}$$

$$g(0,0) = \frac{1}{2} (-1)^{b_0(0) b_0(0)} = 1/2$$

$$g(1,0) = \frac{1}{2} (-1)^{b_0(1) b_0(0)} = 1/2$$

$$g(0,1) = \frac{1}{2} (-1)^{b_0(0) b_0(1)} = 1/2$$

$$g(1,1) = \frac{1}{2} (-1) = -1/2$$

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

For Hadamard transform for $N=4$

$$H_4 = \frac{1}{4}\begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}$$

$$H_4 = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & +1 \end{bmatrix}$$

For $N=8$

$$H_8 = \frac{1}{8}\begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix}$$

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

The main difference between walsh and Hadamard is only in the order of the basis function.

Haar transform :-

The Haar transform is based on a class of orthogonal matrices whose elements are either 1, -1, (or) 0 multiplied by powers of $\sqrt{2}$. The Haar transform is a computationally efficient transform as the transform of a N-point vector requires only $(2(N-1)$ additions and N multiplications

Algorithm for Haar transform

step1: Determine the order of N of the Haar basis.

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

3) Determine p and q

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

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

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

4) Determine $k \Rightarrow k = 2^p + q - 1$

5) Determine z $z \to [0,1] \Rightarrow \left[\frac{0}{N}, \frac{1}{N}, \frac{2}{N} \cdots \frac{N-1}{N}\right]$

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

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

Generate Haar basis for $N=2$

1) $N=2$
2) $n = \log_2 2 = 1$
3) i) since $n=1$, the only value of p is 0
   ii) So q takes the value of 0 (or) 1
4) Determine k $\quad k = 2^p + q - 1$

p	q	k
0	0	0
0	1	1

5) step 5: Determine z value $z \rightarrow [0, 1] \Rightarrow \left[\frac{0}{2}, \frac{1}{2}\right]$

6) $\boxed{p = 0}$

$q=0 \qquad q=1$

Case(i) if $k=0$ then $H(z) = \frac{1}{\sqrt{N}} = \frac{1}{\sqrt{2}}$;

k \ z	0	1
0	$1/\sqrt{2}$	$1/\sqrt{2}$
1	—	—
2	0	1

← since the value for k is '0' for all 'z' $H(z)$ is $\frac{1}{\sqrt{2}}$

Case ii   For $k=1$; $P=0$; $q=1$

Condition (i) $0 \le z \le 1/2$

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

Condition (iii) otherwise.

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

For $z=0$ the boundary Condition (i) satisfied

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

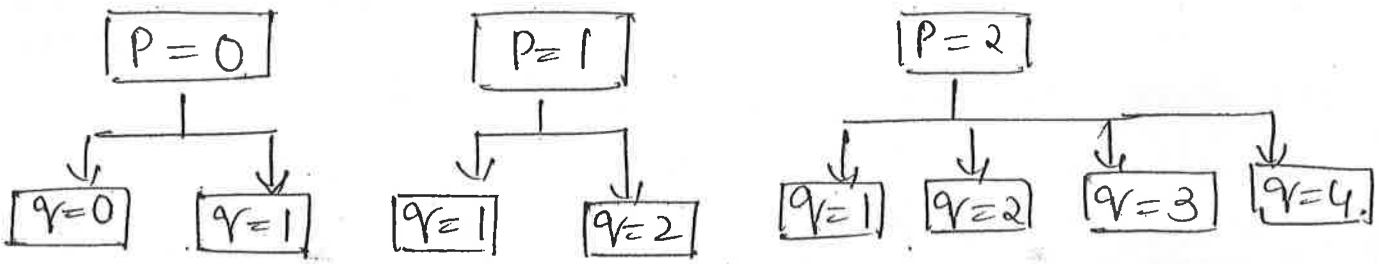
for $z = 1/2$ Condition (ii) satisfied $H(z) = \frac{-1}{\sqrt{2}} 2^{0/2} = \frac{-1}{\sqrt{2}}$

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

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

Haar basis for $N=8$

1) Determine the order of $N=8$
2) Determine n where $n = \log_2 N = 3$
3) Determine p and q
   i) $0 \leq p < 2$
   ii) If $p=0$ then $q=0$ & $q=1$
   iii) If $p \neq 0$, $1 \leq q \leq 2^p$



k values for different combinations of p and q

Combination	p	q	$k = 2^p + q - 1$
0	0	0	0
1	0	1	1
2	1	1	2
3	1	2	3
4	2	1	4
5	2	2	5
6	2	3	6
7	2	4	7

5) Determine z

$z \to [0,1] \Rightarrow [0/8, 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8]$.

6) If $k=0$ then $H(z) = \frac{1}{\sqrt{8}} = \frac{1}{2\sqrt{2}}$

otherwise

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

when $k=1$;

$(2^{p}+q-1)$

① $\underline{p=0; \; q=1}$

$\underline{\text{Condition}}$ i) $0 \le z < \frac{1}{2} \Rightarrow H_1(z) = \frac{1}{\sqrt{N}} 2^{p/2}$

ii) $\frac{1}{2} \le z < 1 \Rightarrow H_1(z) = \frac{-1}{\sqrt{N}} 2^{p/2}$

iii) otherwise $\Rightarrow H_1(z) = 0$.

a) for $z=0$ 1 condi^n satisfied so

$0 \le z < 1/2 \Rightarrow H_1(z) = \frac{1}{\sqrt{N}} 2^{p/2}$ $1/2\sqrt{2}$

(b) $z = 1/8$ 1 Condition satisfied.

$0 \le z < 1/2 \Rightarrow H_1(z) = \frac{1}{\sqrt{N}} 2^{p/2} = \frac{1}{2\sqrt{2}}$

c) for $z = 1/4$, the first condition satisfied.

$0 \le z < 1/2 \Rightarrow H_1(z) = \frac{1}{\sqrt{N}} 2^{p/2} = \frac{1}{2\sqrt{2}}$

d) $z = 3/8$, Condition 1 $\Rightarrow H_1(z) = \frac{1}{\sqrt{N}} 2^{P/2} = \frac{1}{2\sqrt{2}}$

e) $z = 1/2$, Condition 2 $\Rightarrow H_1(z) = \frac{-1}{\sqrt{N}} 2^{P/2} = -1/2\sqrt{2}$

f) $z = 5/8$, Condiⁿ 2 $\Rightarrow H_1(z) = \frac{-1}{\sqrt{N}} 2^{P/2} = -1/2\sqrt{2}$.

g) $z = 3/4$, 2 Condiⁿ $\Rightarrow H_1(z) = \frac{-1}{\sqrt{N}} 2^{P/2} = -1/2\sqrt{2}$

h) $z = 7/8$, 2 Condiⁿ $\Rightarrow H_1(z) = \frac{-1}{\sqrt{N}} 2^{P/2} = \frac{-1}{2\sqrt{2}}$

② When $k=2$ $p=1$; $q=1$

Conditions
i) $0 \le z < 1/4 \Rightarrow H_2(z) = \frac{1}{\sqrt{N}} 2^{P/2}$

ii) $1/4 \le z < 1/2 \Rightarrow H_2(z) = \frac{-1}{\sqrt{N}} 2^{P/2}$

iii) otherwise $\Rightarrow H_1(z) = 0$

a) For $z=0$, Condiⁿ 1 $\Rightarrow H_2(z) = \frac{1}{\sqrt{2}} \times 2^{P/2} = 1/2$

b) For $z = 1/8$, Condiⁿ 1 $\Rightarrow H_2(z) = 1/2$

c) For $z = 1/4$, Condiⁿ 2 $\Rightarrow H_2(z) = -1/2$

d) For $z = 3/8$, Condiⁿ 2 $\Rightarrow H_2(z) = -1/2$

e) For $z = 1/2$, Condiⁿ 3 $\Rightarrow H_2(z) = 0$

similarly

for $H_2(5/8) = H_2(3/4) = H_2(7/8) = 0$.

when $k=3$ $p=1$; $q=3$;

Conditions i) $\frac{1}{2} \le z < \frac{3}{4} \Rightarrow H_3(z) = \frac{1}{\sqrt{N}} 2^{p/2} = \frac{1}{2\sqrt{2}} \times \sqrt{2} = \frac{1}{2}$

ii) $\frac{3}{4} \le z < 1 \Rightarrow H_3(z) = \frac{-1}{\sqrt{N}} 2^{p/2} = -\frac{1}{2}$

iii) otherwise $\Rightarrow H_3(z) = 0$

a) for $z = 0, \frac{1}{4}, \frac{1}{8}, \frac{3}{8}$ satisfies 3 condition. $H_3(z) = 0$

b) for $z = \frac{1}{2}, \frac{5}{8}$ 1st condition satisfied

$$H_3(z) = \frac{1}{\sqrt{N}} 2^{p/2} = \frac{1}{2}$$

c) For $z = \frac{3}{4}, \frac{7}{8}$ the 2nd condiⁿ satisfied $= -\frac{1}{2}$

③ When $k=4$ $p=2$ $q=1$

Conditions i) $0 \le z < \frac{1}{8} \Rightarrow H_4(z) = \frac{1}{\sqrt{N}} 2^{p/2} = \frac{1}{\sqrt{2}}$

ii) $\frac{1}{8} \le z < \frac{1}{4} \Rightarrow H_4(z) = \frac{-1}{\sqrt{N}} 2^{p/2} = \frac{-1}{2\sqrt{2}} \times 2 = \frac{-1}{\sqrt{2}}$

otherwise $H_4(z) = 0$

a) for $z = 0$, 1st condition $\Rightarrow H_4(z) = \frac{1}{\sqrt{2}}$

b) for. $z = \frac{1}{8}$, 2nd condition $\Rightarrow H_4(z) = \frac{-1}{\sqrt{2}}$

c) for $z = \frac{1}{4}$, ~~3/8~~, ~~1/2~~, ~~5/8~~, $\frac{3}{8}$, $\frac{7}{8}$

3rd condition satisfied.

$$H_4(z) = 0$$

4 when $k=5,\ p=2;\ q=2$.

Condition i) $\frac{1}{4} \le z < \frac{3}{8} \Rightarrow H_5(z) = \frac{1}{\sqrt{N}}\, 2^{p/2} = \frac{1}{2\sqrt{2}} \times 2 = \frac{1}{\sqrt{2}}$

ii) $\frac{3}{8} \le z < \frac{1}{2} \Rightarrow H_5(z) = \frac{-1}{\sqrt{N}}\, 2^{p/2} = -\frac{1}{\sqrt{2}}$

iii) otherwise $\Rightarrow H_5(z) = 0$

a) for $z=0$, 3^{rd} condition $\rightarrow H_5(z) = 0$.

b) for $z = \frac{1}{4}$, 1^{st} condition $\rightarrow H_5(z) = \frac{1}{\sqrt{2}}$

c) $z = \frac{3}{8}$, 2^{nd} condition $\rightarrow H_5(z) = \frac{-1}{\sqrt{2}}$

d) $z = \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}$, 3^{rd} condition $\rightarrow H_5(z) = 0$

5 when $k=6,\ p=2,\ q=3$

Condition i) $\frac{1}{2} \le z < \frac{5}{8} \Rightarrow H_6(z) = \frac{1}{\sqrt{N}}\, 2^{p/2} = \frac{1}{2\sqrt{2}} \times 2 = \frac{1}{\sqrt{2}}$

ii) $\frac{5}{8} \le z < \frac{3}{4} \Rightarrow H_6(z) = \frac{-1}{\sqrt{N}}\, 2^{p/2} = \frac{-1}{\sqrt{2}}$

iii) otherwise $H_6(z) = 0$

a) $z = 0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{3}{4}, \frac{7}{8} \rightarrow 3^{rd}$ condition is satisfied $\rightarrow H_6(z) = 0$.

b) $z = \frac{1}{2}$, 1^{st} condition $\rightarrow H_6(z) = \frac{1}{\sqrt{2}}$

c) $z = \frac{5}{8}$, 2^{nd} condition $\rightarrow H_6(z) = \frac{-1}{\sqrt{2}}$

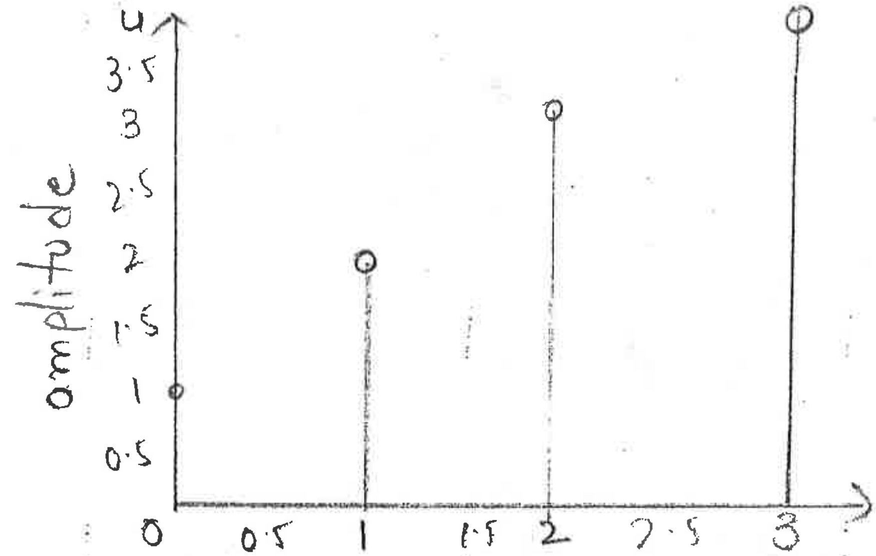
$$
= \frac{1}{2\sqrt{2}}
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\frac{7}{\sqrt{21}} & \frac{5}{\sqrt{21}} & \frac{3}{\sqrt{21}} & \frac{1}{\sqrt{21}} & \frac{-1}{\sqrt{21}} & \frac{-3}{\sqrt{21}} & \frac{-5}{\sqrt{21}} & \frac{-7}{\sqrt{21}} \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
\frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \\
\frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{3}{\sqrt{5}} \\
\frac{7}{\sqrt{105}} & \frac{-1}{\sqrt{105}} & \frac{-9}{\sqrt{105}} & \frac{-17}{\sqrt{105}} & \frac{17}{\sqrt{105}} & \frac{9}{\sqrt{105}} & \frac{1}{\sqrt{105}} & \frac{7}{\sqrt{105}} \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
\frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{1}{\sqrt{5}}
\end{bmatrix}
$$

Discrete Cosine transform:

A discrete cosine transform consists of a set of basis vectors that are sampled cosine functions. DCT is a technique for converting a signal into elementary frequency components and it is widely used in JPEG Loss image compression. DCT only use real numbers and extended as periodically and symmetrically.

If $x(n)$ is the signal of length N, the fourier transform of the signal $x[n]$ is given by $X(k)$

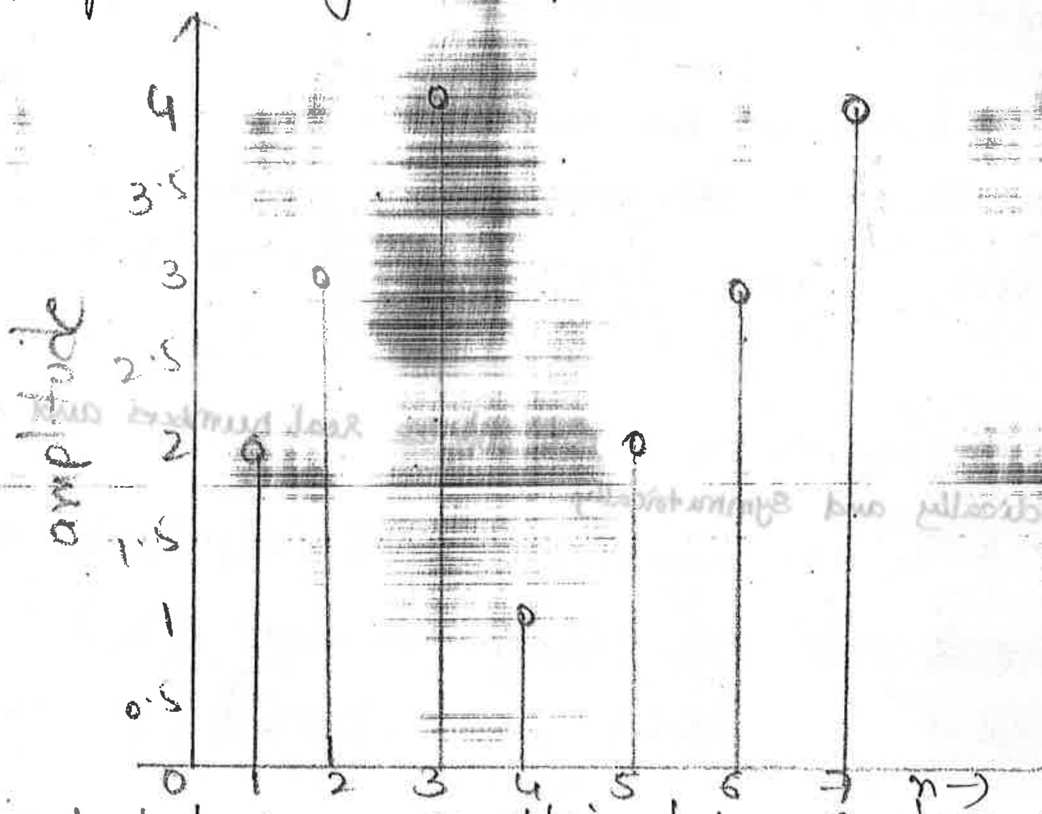
$$X[k] = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi kn}{N}}$$



n→ original sequence

original sequence of $x[n]$.

The main drawback of this method is the variation in the value of the sample at $n=3$ and at $n=4$. since the variation is drastic the phenomenon of ringing is inevitable. To overcome this, a second method of obtaining the extended sequence is by copying the original sequence in a folded manner



Extended sequence obtained by copying original

KL Transform: (KARHUNEN - LOEVE Transform)

KL transform is known as Hotelling transform & eigen vector transform. It is based on stastical properties of an image.

KL Transform is used for compression of an image by decorrelating the neighbouring pixels of an image.

Procedure (or) algorithm to KL Transform:

(i) Find the mean vector and covariance of the matrix

(ii) Find the eigen values and eigen vectors of the covariance matrix

(iii) Create Transformation matrix T, such that rows of T are eigen values.

(iv) Find KL Transform.

For example:

$$x_1 = (000)^T \quad x_2 = (100)^T \quad x_3 = (110)^T \quad x_4 = (101)^T$$

Covariance matrix of vector population

$$C_x = \frac{1}{4}\sum_{K=1}^{4} x_K x_K^T - m_K m_K^T$$

where m_K = mean of matrix, i.e calculated as

$$m_K = \frac{1}{4}[x_1+x_2+x_3+x_4] = \frac{1}{4}\left[\begin{bmatrix}0\\0\\0\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}+\begin{bmatrix}1\\1\\0\end{bmatrix}+\begin{bmatrix}1\\0\\1\end{bmatrix}\right]$$

$$= \frac{1}{4}\begin{bmatrix}3\\1\\1\end{bmatrix}$$

$$m_K . m_K^T = \frac{1}{4}\begin{bmatrix}3\\1\\1\end{bmatrix} . \frac{1}{4}[3\ 1\ 1] = \frac{1}{16}\begin{bmatrix}9&3&3\\3&1&1\\3&1&1\end{bmatrix}$$

$$\frac{1}{4}\sum_{k=1}^{4} x_k x_k^T = \frac{1}{4}\left[\begin{bmatrix}0\\0\\0\end{bmatrix}[0\,0\,0] + \begin{bmatrix}1\\0\\0\end{bmatrix}[1\,0\,0] + \begin{bmatrix}1\\1\\0\end{bmatrix}[1\,1\,0] + \begin{bmatrix}1\\0\\1\end{bmatrix}[1\,0\,1]\right]$$

$$= \frac{1}{4}\begin{bmatrix}3 & 1 & 1\\ 1 & 1 & 0\\ 1 & 0 & 1\end{bmatrix}$$

The covariance matrix $= C_x = \begin{bmatrix}3/4 & 1/4 & 1/4\\ 1/4 & 1/4 & 0\\ 1/4 & 0 & 1/4\end{bmatrix} - \begin{bmatrix}9/16 & 3/16 & 3/16\\ 3/16 & 1/16 & 1/16\\ 3/16 & 1/16 & 1/16\end{bmatrix}$

$$C_x = \begin{bmatrix}3/16 & 1/16 & 1/16\\ 1/16 & 3/16 & 1/16\\ 1/16 & 1/16 & 1/16\end{bmatrix}$$

A is the transformation matrix obtained by

$$|C_x - \lambda I| = 0$$

$$\Rightarrow \left|\frac{1}{16}\begin{bmatrix}3 & 1 & 1\\ 1 & 3 & -1\\ 1 & 1 & 1\end{bmatrix} - \lambda\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\right| = 0$$

$$\Rightarrow \begin{vmatrix}\frac{3}{16}-\lambda & \frac{1}{16} & \frac{1}{16}\\ \frac{1}{16} & \frac{3}{16}-\lambda & -\frac{1}{16}\\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16}-\lambda\end{vmatrix} = 0$$

$$\Rightarrow \lambda_1 = 0.25 \quad \lambda_2 = 0.065 \quad \lambda_3 = 0.125$$

Eigen vectors corresponding to eigen values $\lambda_1, \lambda_2, \lambda_3$ are

$$\begin{array}{c} \lambda_1 \rightarrow \\ \lambda_2 \rightarrow \\ \lambda_3 \rightarrow \end{array}\begin{bmatrix}0.8165 & 0.4082 & 0.4082\\ 0.5774 & 0.5774 & -0.5774\\ 0.7071 & -0.7071 & 0\end{bmatrix} = A$$

Transformed vector groups are obtained as

$$y_1 = A(x_1 - m_x) \qquad y_2 = A(x_2 - m_x)$$

$$y_3 = A(x_3 - m_x) \qquad y_4 = A(x_4 - m_x)$$

$$y_1 = \begin{bmatrix} 0.8165 & 0.4082 & 0.4082 \\ 0.5774 & -0.5774 & -0.5774 \\ 0.7071 & 0.7071 & 0 \end{bmatrix} \begin{bmatrix} 0 - 3/4 \\ 0 - 1/4 \\ 0 - 1/4 \end{bmatrix} = \begin{bmatrix} -0.8165 \\ -0.1444 \\ -0.3535 \end{bmatrix}$$

Simillarly $y_2 = \begin{bmatrix} 0 \\ 0.4331 \\ 0.3535 \end{bmatrix}$ $\quad y_3 = \begin{bmatrix} 0.4082 \\ -0.1444 \\ -0.35 \end{bmatrix}$ $\quad y_4 = \begin{bmatrix} 0.4082 \\ -0.144 \\ 0.3535 \end{bmatrix}$

Covariance of transposed vectors

$$C_y = \frac{1}{4} \sum_{k=1}^{4} y_k y_k^T - m_y m_y^T$$

$$m_y = \frac{1}{4}\left[y_1 + y_2 + y_3 + y_4\right] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C_y = \frac{1}{4}\left[y_1 y_1^T + y_2 y_2^T + y_3 y_3^T + y_4 y_4^T\right] - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C_y = \frac{1}{4} \begin{bmatrix} 0.994 & 0 & 0 \\ 0 & 0.248 & 0 \\ 0 & 0.2 & 0.5 \end{bmatrix}$$

Inverse Transform $x = A^T y + m_x$

Applications of KL Transform:

→ Dimentional reduction

→ Removes Random Noise without blurring stationary & moving edges

→ Reduce noise in real time images

→ Extraction of signal corresponding to small breathing displacements of human chest.

SVD Transform: (Singular Value Decomposition Transform):

The SVD transform is another popular image transform that has huge no. of applications in image Restoration, compression and object recognition. The SVD transform of an image 'f' is

$$g = SVD(A)$$

The SVD transform transforms the given matrix A into the product $U \times S \times V^{-1}$ i.e

$$A = U \times S \times V^{-1}$$

The matrix U is an orthogonal matrix. The column vectors from an orthogonal set, i.e

$$u_i^T \times u_j = \delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

The matrix V is an $n \times n$ orthogonal matrix and its columns form an orthonormal set. 'S' is the matrix of order $n \times n$ with singular values are

$$S = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$\sigma_1, \sigma_2, \cdots \sigma_n$ are called singular values which are square roots of the eigen values and form diagonal of S.

The property of SVD is that the singular values are not unique. i.e $U = AA^{-1}$ $V = \bar{A}^{\prime} \cdot A$.

Therefore the image is expressed as

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$$\Rightarrow A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

here r is called Rank of matrix

Here the Rank is nothing but no. of non zero diagonal elements.

The SVD transform is used for image compression. If the sum is truncated after r terms, the result is called r.approximation of original matrix.

The ~~distance~~ difference b/w original and approximation is called error.

Ex: Find SVD of image F.

$$F = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}$$

Sol: Step 1: Here F is square matrix. Calculate eigen value & eigen vector & characteristic equation

$$|F - \lambda I| = 0$$

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

$\lambda = 1, 1, -2$ and eigen vectors are

$x_1 = (1\ 0\ 1)$ $x_2 = (1\ 2\ -1)$ $x_3 = (-1\ 1\ 1)$

Normalized the vector $\Rightarrow$ the normalized matrix & modal matrix S

$$S = \begin{bmatrix} \frac{1}{\sqrt{1^2+1^2}} & \frac{1}{\sqrt{1^2+2^2+4^2}} & \frac{1}{\sqrt{-1^2+2^2+1^2}} \\ \frac{1}{\sqrt{1^2+1^2}} & \frac{1}{\sqrt{1^2+2^2+4-1^2}} & \frac{1}{\sqrt{-1^2+2^2+1^2}} \\ \frac{1}{\sqrt{1^2+1^2}} & \frac{1}{\sqrt{1^2+2^2+8-1^2}} & \frac{1}{\sqrt{-1^2+2^2+1^2}} \end{bmatrix}$$

$$S = \begin{bmatrix} 1/\sqrt{2} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix}$$

$\Rightarrow \quad D = S \times F \times S$

$$\Rightarrow \quad D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

UNIT - II (PART-A)

INTENSITY TRANSFORMATIONS & SPATIAL FILTERING

[IMAGE ENHANCEMENT]

INTRODUCTION :-

* Image Enhancement techniques are designed to improve the quality of an image as perceived by a human being.
* The main objective is to improve the quality of an image even by the degradation is available. This can be achieved by increasing the dominance of some features or decreasing the ambiguity b/w different regions of image.
* The Intensity transformations operate on single pixels of an image, for the purpose of contrast manipulation & image thresholding.
* Spatial filtering deals with performing operations such as image sharpening by considering every pixel in an image.

Definition :-

* Image Enhancement refers to processed image which is more suitable than original image.
* Enhancement methods are application specific and are often developed empirically

REASONS TO PREFER ENHANCEMENT TECHNIQUES :-

* Due to bad illumination sources
* For maintaining correct acceptance angle
* For good dynamic range.

CLASSIFICATION :-

Image Enhancement techniques can be done in 2 ways. They are:

(a) Spatial domain Enhancement

- Masking
- Filtering
- point operation

(b) Frequency domain Enhancement

BACKGROUND :

Basics of Intensity Transformations & Spatial Filtering :

The spatial domain processes can be denoted by the expression

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

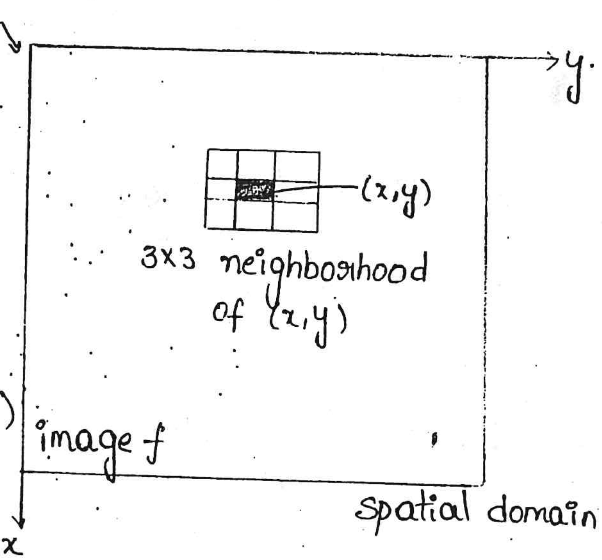
where $f(x,y)$ is i/p image

$g(x,y)$ is o/p image

T is an operator

* The operator (T) can be applied to a single image or to a set of images

Eg:- Let us consider a 3x3 neighborhood about a point (x,y) in an image in spatial domain as shown



* From the figure,
Consider an arbitrary location say (100, 150). Assuming that the origin of neighborhood is at its centre then the o/p $g(100,150)$ is obtained by considering

∴ $g(100,150)$ = Sum of [$f(100,150)$ & its 8 neighbors divided by 9]

The origin of the neighborhood is then moved to next location & the procedure is repeated as discussed above to get the o/p image.

This procedure is called <u>spatial filtering</u> & the neighborhood along with operator 'T' is known as <u>spatial filter</u>.

The smallest possible neighborhood is of size 1x1. In this case the o/p $g(x,y)$ depends only on the value of f at single point (x,y) & T becomes an intensity transformation function given by

$$\boxed{S = T(r)}$$

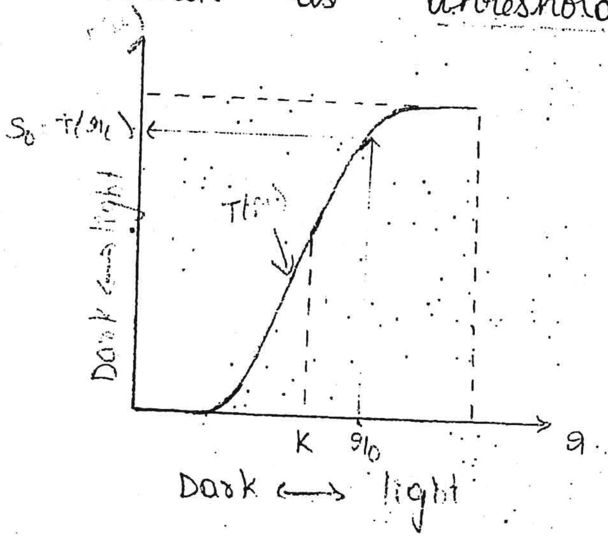
where S & r are variables

For example if $T(r)$ has the form as shown in fig (a) then the effect of applying the transformation results in 2 factors:

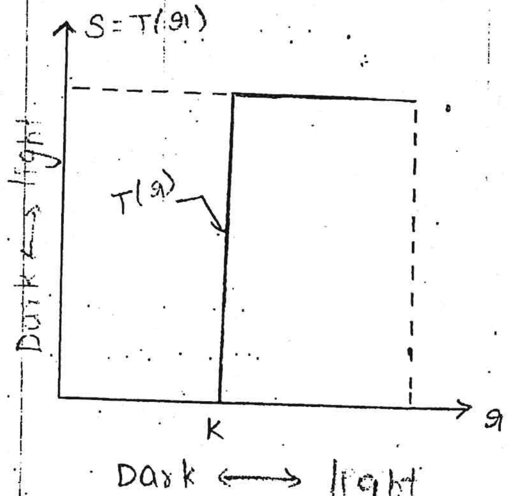
(i) To produce an image of higher contrast than the original by darkening the intensity levels below 'k' & brightening the levels above 'k'.

In this case the values of r lower than k are compressed by the transformation function which is known as contrast stretching.

(ii) To produce a two-level (binary) image by considering the values of "r" higher than 'k' which is known as thresholding function.



(a) Contrast stretching function



(b) Thresholding function

fig :- Intensity transformation functions

TECHNIQUES OF SPATIAL DOMAIN :-

1. Image Negative
2. Contrast stretching
3. Clipping
4. Thresholding
5. Log transformation
6. Level slicing
7. Bit plane slicing
8. power law transformation
9. Histogram specification
10. Histogram Equalisation

point operation techniques.

1. IMAGE NEGATIVE :-

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

$$s = L-1-r \qquad r \rightarrow \text{Intensity value}$$

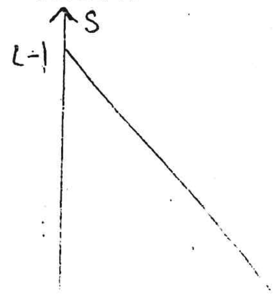
We know that $s = T(r)$ &

$L = 2^b$

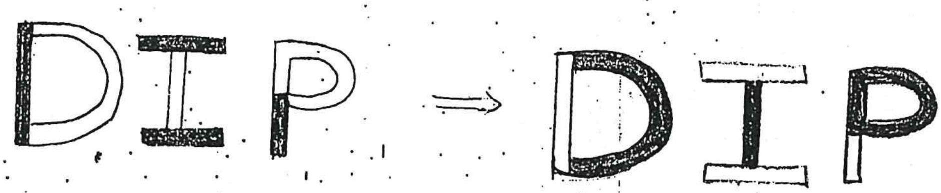
where b is bit-width

Let $b=8$, then

$$s = T(r) = 2^8-1-0$$
$$= 255$$



The main aim of this Image negative is, the dark region is converted into bright & bright into dark.



Eg:- Let us assume an image of pixels

$$\text{i/p image } f(x,y) = \begin{bmatrix} 2 & 4 & 6 & 10 \\ 6 & 7 & 0 & 0 \\ 15 & 15 & 1 & 2 \\ 0 & 15 & 15 & 0 \end{bmatrix}_{4\times4}$$

Here bit width $(b) = 4$

$$L = 2^b = 2^4 = 16$$

$$S = L-1-r = 16-1-r = 15-r$$

$$\Rightarrow \begin{bmatrix} 15-2 & 15-4 & 15-6 & 15-10 \\ 15-6 & 15-7 & 15-0 & 15-0 \\ 15-15 & 15-15 & 15-1 & 15-2 \\ 15-0 & 15-15 & 15-15 & 15-0 \end{bmatrix}$$

$$\therefore \text{processed image} = \begin{bmatrix} 13 & 11 & 9 & 5 \\ 9 & 8 & 15 & 15 \\ 0 & 0 & 14 & 13 \\ 15 & 0 & 0 & 15 \end{bmatrix}$$

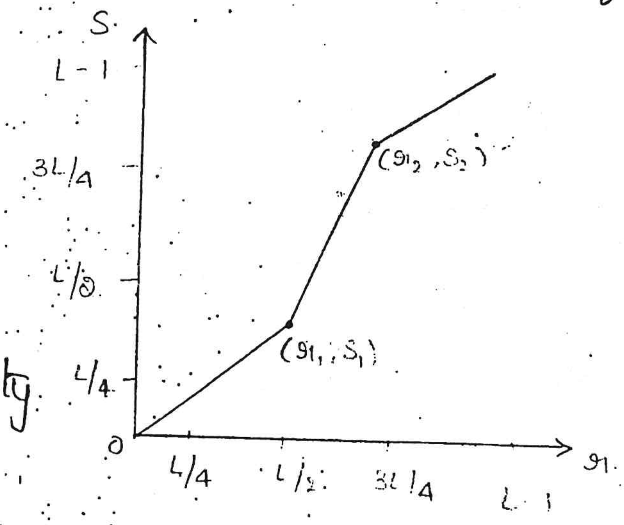
Applications :- Negatives of digital images are useful in numerous applications such as displaying medical images & photographing a screen with monochrome +ve film.

2. Contrast stretching :-

The process of expanding the range of intensity levels in an image, inorder to utilise the full range of intensity levels is known as contrast stretching.

It is one of the simplest piece-wise linear functions. Low-contrast images can result from poor illumination.

The adjacent figure shows the typical transformation used for contrast stretching.



Case-1 :- From the figure, if $r_1 = s_1$ & $r_2 = s_2$ then the transformation is linear & produces no change in intensity levels.

Case-2 :- If $r_1 = r_2$; $s_1 = 0$; $s_2 = L-1$ the transformation becomes a thresholding function which creates a binary image.

* The result of contrast stretching is obtained by assuming $(r_1, s_1) = (r_{min}, 0)$ & $(r_2, s_2) = (r_{max}, L-1)$

Eg :- Let us consider an image of 4x4 size

[illegible]

Here $L = 2^6 = 64$

∴ The values of (r_1, s_1) & (r_2, s_2) are

$(r_1, s_1) = (r_{min}, 0) = (r_{min}, 0)$

$(r_2, s_2) = (r_{max}, L-1) = (r_{max}, 63)$.

* Contrast stretching occurs due to

(a) clipping (b) Thresholding

* We know that

$$S = T(r)$$

Here
$$s = \begin{cases} \alpha r & 0 < r < a \\ \beta(r-a) + v_a & a < r < b \\ \gamma(r-b) + v_b & r > b \end{cases}$$

If $\alpha = \gamma = 0$, $\beta(r-a) + v_b$

clipping & thresholding are specical cases in this.

If $a = b = T$ threshold occurs

If $r_1 = r_2$

$\Rightarrow s_1 = 0$; $s_2 = L-1$

If $r_1 \neq r_2$; $s_1 > 0$; $s_2 > s_1$

$s = L-1$

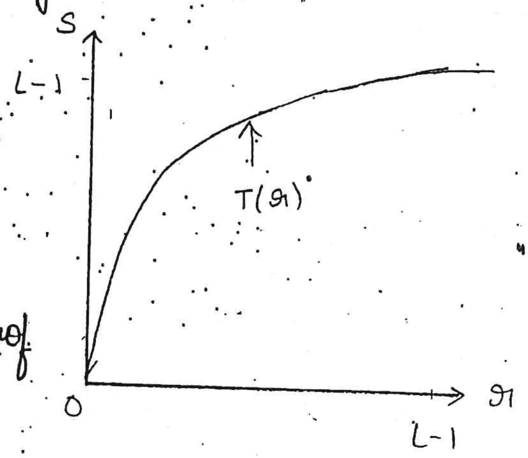
* <u>Log transformation</u> :-

The general form of log transformation is expressed as

$$S = T(r) = c \log(1+r)$$

where c is constant & it is assumed that $r \geq 0$.

The shape of the log curve shows that this transformation maps a narrow range of low gray-level values in the i/p image into a wider range of o/p levels.



The log transformation has the important characteristic of compressing the dynamic range of images with large variations in pixel values.

* LEVEL SLICING :-

The purpose of LEVEL SLICING is to highlight a specific range of gray values.

The 'Level Slicing' can be done with 2 approaches. They are:

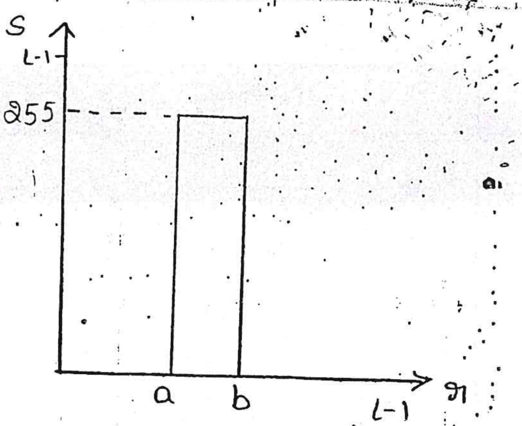
(a) Level slicing without preserving background.

(b) Level slicing with background.

(a) LEVEL SLICING WITHOUT PRESERVING BACKGROUND :-

In this all the gray levels of particular range are displayed with higher values & remaining gray levels are displayed with lower values.

From the adjacent figure we can observe that only one part is highlighted & remaining are zero's



$$S = T(r) = \begin{cases} L & a \leq r \leq b \\ 0 & \text{otherwise} \end{cases}$$

Drawback :- The main drawback of this approach is, that the background information is discarded.

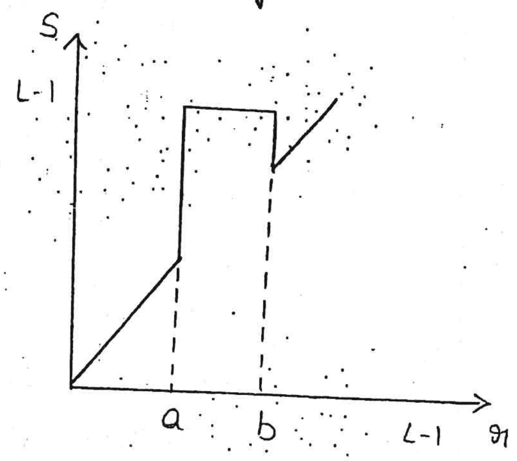
(b) LEVEL SLICING WITH BACKGROUND :-

In this high values are displayed for particular range & original gray level values in other areas.

Since the remaining gray levels does not become zero, it preserves the background of image.

Here we consider

$$S = T(r) = \begin{cases} L & a \leq r \leq b \\ r & 0 \leq r \leq L-1 \end{cases}$$



* BIT-PLANE SLICING :-

The main objective in this is instead of highlighting gray level ranges we should highlight the contribution made to total image by

Considering specific no of bits.

In Bit plane slicing the image is divided according to no. of bits.

Lower planes are considered as Bit-plane 0 (LSB bits) & Upper planes are considered as Bit-plane 7 (MSB bits).

The 3-main objectives of Bit-plane slicing are:

(a) Converting gray level image to binary image.

(b) Representing an image with fewer bits & Compressing the image to smaller size.

(c) Enhancing the image by focussing.

* For an 8-bit image, 0 is encoded as 00000000 & 255 is encoded as 11111111. Any number between 0 & 255 is encoded as one byte.

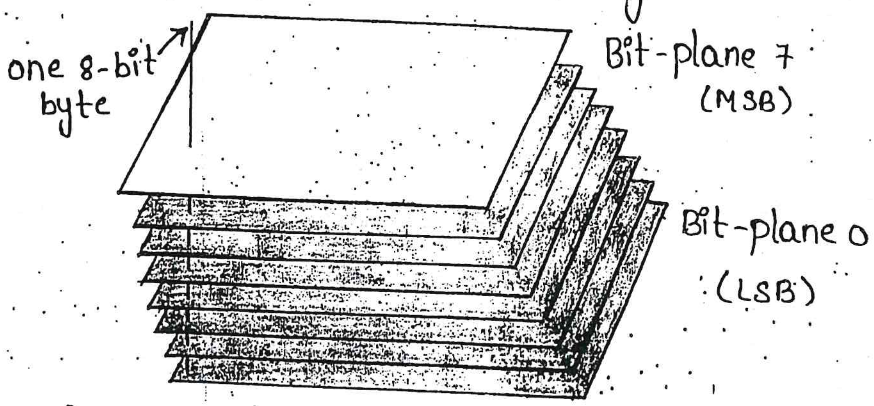


fig: Bit plane representation of an 8-bit image.

* By using Bit-plane slicing image can be compressed.

* POWER LAW TRANSFORMATION :-

Power law transformations have the basic form

$$s = c r^{\gamma} \qquad \text{--- (1)}$$

where c & γ are positive constants.

The above eqn can also be written as

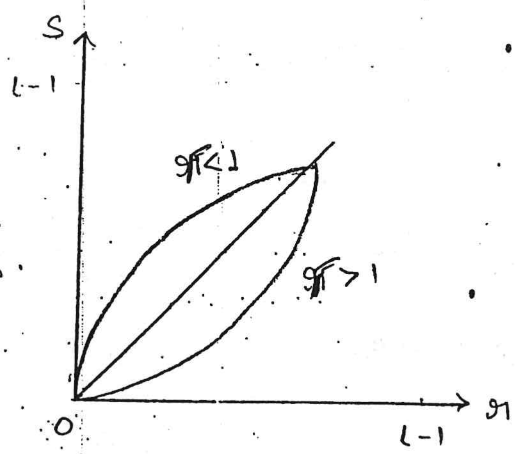
$s = (r + \epsilon)^{\gamma}$ for offset purpose (i.e., measurab o/p when the i/p is zero)

* Since gamma (γ) is used to correct the power law response phenomenon, it is known as gamma correction & this power law transformation is also known as GAMMA TRANSFORMATION.

From (1), if $r = s$ then original intensity levels of image = o/p intensity levels.

$r < 1$ for high intensities

$r > 1$ for low intensities.



Advantage :-

1. This law is used in variety of devices for image capturing, printing & display responding purpose.
2. Used for Gamma correction.
3. power-law transformations are useful for general purpose contrast manipulations.

HISTOGRAM :-

Histogram of an image is defined as the representation including relative frequency of occurance of various gray levels in the image.

Inorder to improve the visual quality of image we use histogram manipulation techniques. The histogram provides more insight about image contrast & brightness

The histogram of an image is a plot of the no. of occurances of gray levels in the image against the grey-level values.

Image near to zero values represent dark one

Image near to L-1 values represent bright one.

For a low contrast image (dark image) the histogram will not spread equally i.e, the histogram will be narrow.

For a high contrast image (bright image) the histogram will have an equal spread in grey level.

It means the histogram of dark image is clustered towards lower gray level & the histogram of bright image is clustered towards higher gray leve

Formula :-

Let r_k is the kth gray level of i/p image

n_k is the no of pixels in the i/p image.

Then the histogram of a digital image with intensity levels in the range $[0, L-1]$ is given as

$$\boxed{h(r_k) = n_k}$$

Since the histogram represents intensity values then <u>normalised histogram</u> is given as

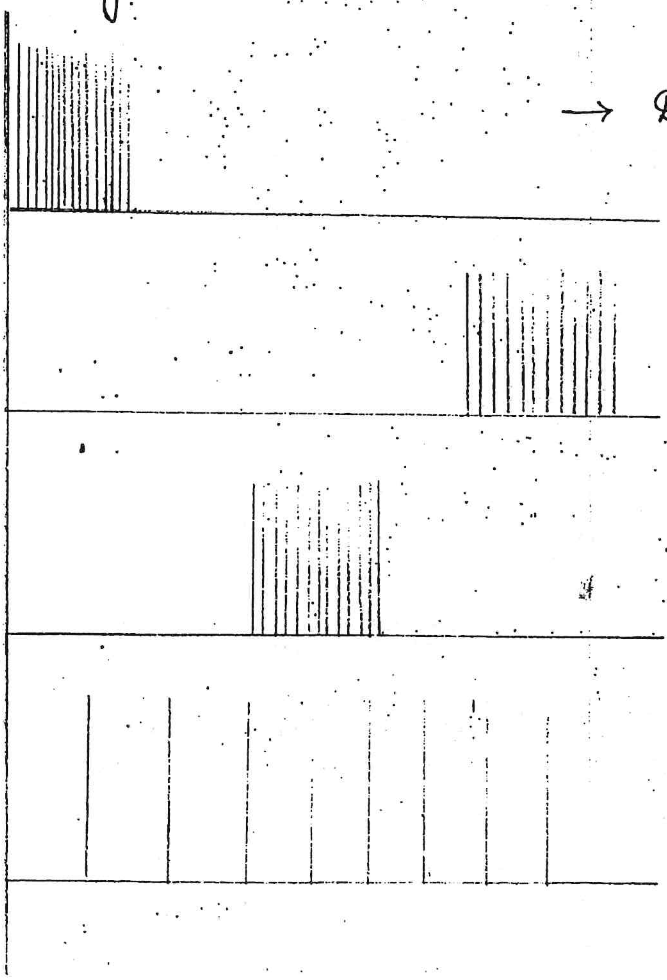
$$h(r_k) = \frac{n_k}{n}$$

when 'n' is no of intensity levels.

* plot histogram for:

(a) dark image (b) Bright image (c) Neither bright nor dark image

(d) Good image



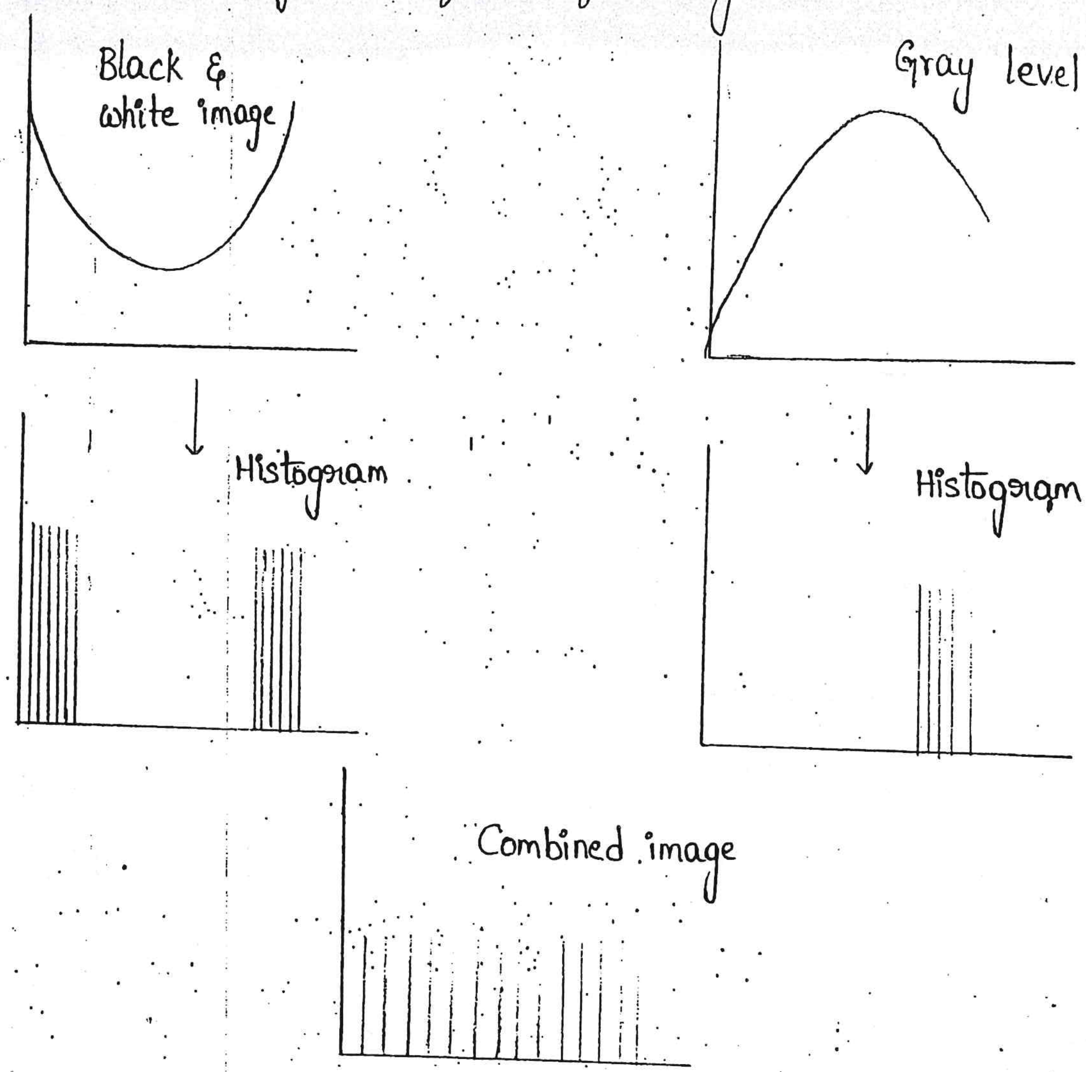
→ Dark image (or) low contrast image

→ Bright image

→ Neither bright (nor) dark image

→ Good image

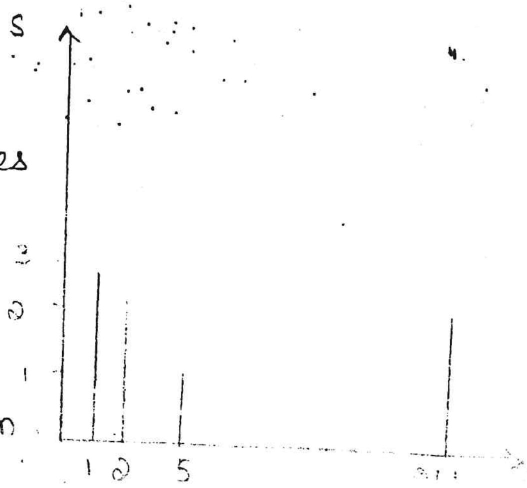
* plot the histogram for following :



Eg :-

For $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 255 & 255 & 0 \end{bmatrix}$ Draw Histogram.

From the given matrix we observe that '1' is repeated 3 times, hence its value is 3. And 2 is repeated 2 times hence its corresponding value is 2 & so on. The adjacent figure represents the corresponding histogram



* HISTOGRAM EQUALISATION (OR) HISTOGRAM LINEARIZATION :-

Let us consider r is original image &

S is processed image

So that $S = T(r)$ —— ①.

In histogram Equalisation, we consider 'r' & 's' as random variables.

The transformation function in ① should satisfy the following conditions :

(i) r limit is from $0 < r < 1$

(ii) $T(r)$ should be a single valued & monotonically increasing function.

(iii) The transformation should be continuous & differentia

* If 'r' limit is from $[0 \ 1]$ then we get black & white image.

The probability density function of transformed gray levels for ① is obtained as :

$$P_s(s)ds = P_r(r)\, dr$$

$$P_s(s) = P_r(r) \frac{dr}{ds} \quad —— ②$$

From ①,

$$S = T(r) = \int_0^r P_r(w)\, dw$$

Differentiating the eqn, we get

$$\frac{d(s)}{dr} = \frac{d}{dr}\left(\int_0^r P_r(w)\, dw\right)$$ where w is dummy variable

$$\frac{ds}{dr} = P_r(r)$$

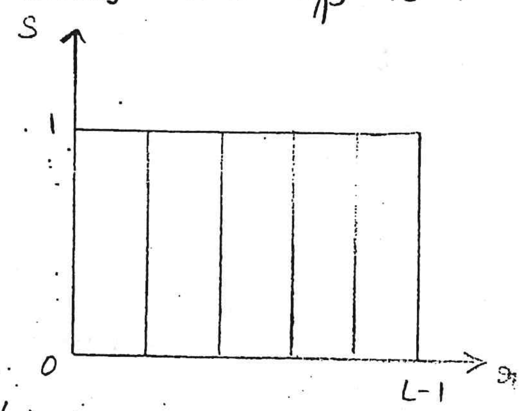
∴ From ②,

$$P_s(s) = P_r(r) \cdot \frac{1}{P_r(r)}$$

$$\boxed{P_s(s) = 1}$$

It means for all intensity levels the o/p is '1'

$P_s(s) = 1$ represents uniform Equalisation.



Drawback :-

By using Histogram Equalisation we can't get accurate manipulations.

*** Example :-**

perform Histogram Equalisation for the image

$$\begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 4 & 3 & 4 \\ 3 & 3 & 4 & 5 \\ 4 & 5 & 2 & 5 \end{bmatrix}_{4 \times 4}$$

Solution :-

The max value of image = 5

∴ we need a minimum of 3 bits to represent the number 5 (101). There are 8 possible gray levels from 0 to 7.

The histogram of i/p image is given below:

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0

Step-1 :- Compute the cumulative sum of above values:

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0
Cumulative Sum	0	0	1	4	12	16	16	16

Step-2 :- Divide the cumulative sum obtained in step-1 by total no. of pixels.

In this case, the total no. of pixels = 16

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0
Cumulative sum	0	0	1	4	12	16	16	16
Total no. of Pixels	0/16	0/16	1/16	4/16	12/16	16/16	16/16	16/16

Step-3 :- Multiply the result obtained in step-2 by the max gray level value which is 7 in this case.

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0
Cumulative Sum	0	0	1	4	12	16	16	16
Total no. of pixels	0/16	0/16	1/16	4/16	12/16	16/16	16/16	16/16
Multiplying the result by 7.	0	0	$7/16 \cong 1$	2	5	7	7	7

Step-4 :- Mapping of gray level by one-to-one correspondence.

original gray level	Histogram equalised values.
0	0
1	0
2	1
3	2
4	5
5	7
6	7
7	7

The original image & the histogram equalised images are shown side by side.

$$\begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 4 & 3 & 4 \\ 3 & 3 & 4 & 5 \\ 4 & 5 & 2 & 5 \end{bmatrix} \xrightarrow[\text{Equalisation}]{\text{Histogram}} \begin{bmatrix} 5 & 5 & 5 & 5 \\ 7 & 5 & 2 & 5 \\ 2 & 2 & 5 & 7 \\ 5 & 7 & 1 & 7 \end{bmatrix}$$

original image — Histogram equalised image.

* HISTOGRAM SPECIFICATION (OR) HISTOGRAM MATCHING :-

The main drawback in Histogram Equalisation is, it is not suitable for interactive image Enhancement applications.

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

Histogram matching means highlighting the particular part.

Let r is i/p image

s is processed image

z is o/p image

Their probabilities are $P_r(r)$, $P_s(s)$; $P_z(z)$

We know that

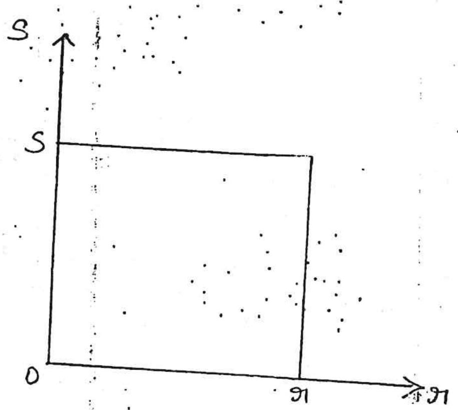
$$s = T(r)$$

$$s = G(z)$$

$$\Rightarrow z = G^{-1}(s) = G^{-1}(T(r))$$

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

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



Differentiating the above eqn, we get

$$\frac{ds}{dr} = P_r(r)$$

$$\Rightarrow \frac{ds}{dz} = P_z(z)$$

$$s = G(z) = G^{-1}(s)$$

$$= G^{-1}(T(r))$$

$$= G^{-1}(P_r(r))$$

Discrete case :-

$$s = T(r) = \sum_k \frac{n_k}{n}$$

$$P_r(r) = \sum_k \frac{n_k}{n}$$

$$S = \sum_{K} \frac{n_K}{n}, \quad K = 0 \dots L-1$$

* SPATIAL MASKING (OR) SPATIAL FILTERING TECHNIQUES :-

Masking :- Some intensities are replaced with other.

Filtering techniques
- Image Smoothing
  - Linear
  - Non-linear
- Image Sharping

Image Smoothing techniques:

1. Low pass filtering (or) Average masking } Linear filters
2. Weighted average Masking } Linear filters
3. Median filter ⟶ Non-linear filter

Image sharping techniques:

4. I order (or) Gradient Masking
5. II order (or) Laplacian Masking
6. Unsharp masking
7. High boost filtering
8. Homo-morphic filtering
9. Sobeal Masking
10. Robert Masking
11. pre-witt Masking

SPATIAL SMOOTHING FILTERS :-

Smoothing filters are used for blurring & for noise reduction.

Blurring is used in pre-processing ... such as rem

* LOW PASS FILTER (OR) AVERAGE MASKING :-

It is a linear filter.

In this the value of every pixel in an image is replaced by the average of intensity levels in the local neighborhood.

The size of the neighborhood controls the amount of filtering.

In this masking, all intensities are same.

The general form of Average masking is

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

Eg :- Let us consider a 3x3 low pass spatial mask.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{3\times 3}$$

Then its general form is taken as

$$g(x,y) = \frac{1}{9} \sum_{s} \sum_{t=-1} f(x+s, y+t)$$

Advantages :-

1. By applying low pass filter noise gets reduced
2. By applying this masking Image gets smoothed.

Dis-advantages :-

1. Average masking leads to blurring of edges; which are desirable features of an image.

2. If the average masking operation is applied to an image, which is corrupted by impulse noise then the impulse noise is attenuated & diffused but not removed.

3. A single pixel with a very unrepresentative value can affect the mean value of all the pixels in its neighborhood significantly.

* WEIGHTED AVERAGE MASKING :-

It is a linear filter

To prevent blurring at the edges, since edges consist of high pass components, we go for weighted average technique.

In this technique the pixels nearest to the centre are weighted more than the distant pixels. Since the centre pixel has more weight, blurring at edges is reduced.

Hence it is named as weighted average filter, the pixel to be updated is replaced by a sum of nearby pixel.

The general expression is as below

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

Eg :- weighted average image : $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

we can observe, that the centre pixel has more weight than remaining pixels.

Its general form is

$$g(x,y) = \frac{1}{16} \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

Advantages :-

1. Blurring at sharp edges gets reduced.
2. Noise gets reduced.
3. Image gets smoothed.

* MEDIAN FILTER :-

It is a non-linear technique.

Median filters provide excellent noise reduction capabilities than Linear smoothing filters.

Median filters are used to reduce salt-and pepper noise (impulse noise). A median filter smoothens the image by utilising the median of neighborhood.

Median filter perform the following tasks to find each pixel in the processed image.

1. All pixels in the neighborhood of the original

image are obtained by arranging them in ascending (or) descending order.

2. The median of the sorted value is computed and is choosen as the pixel value of processed image

Eg :- Compute the median value of the marked pixel shown below using 3x3 mask.

$$\begin{bmatrix} 1 & 5 & 7 \\ 2 & 4 & 6 \\ 3 & 2 & 1 \end{bmatrix}$$

Solution :- The median value of marked pixel is computed as follows :

Step-1 :- First the pixel values are arranged in ascending order

1 1 2 2 3 4 5 6 7.

Step-2 :- The median value of the ordered pixel is computed as follows :

$\cancel{1}$ $\cancel{1}$ $\cancel{2}$ $\cancel{2}$ 3 $\cancel{4}$ $\cancel{5}$ $\cancel{6}$ $\cancel{7}$

Median value = 3.

Now the original pixel value 4 is replaced by the computed median value 3..

$$\begin{bmatrix} 1 & 5 & 7 \\ 2 & 4 & 6 \\ 3 & 2 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 5 & 7 \\ 2 & 3 & 6 \\ 3 & 2 & 1 \end{bmatrix}$$

original image — After median filtering.

SPATIAL SHARPENING FILTERS :-

The main objective of sharpening is to highlight transitions in intensity.

* FIRST ORDER DERIVATIVE (OR) GRADIENT MASKING (OR) PRE-WITT MASKING

* Image differentiation enhances edges & other discontinuities & de-emphasizes areas with slowly varying intensities.

By using Gradient masking we find out the vertical & horizontal thick values only.

Gradient function, $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

$$f(x,y) = \left[\left| \frac{\partial f}{\partial x} \right|^2 + \left| \frac{\partial f}{\partial y} \right|^2 \right]$$

* Let us consider an image,

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

· Differentiation is nothing but difference b/w previous & present images.

$$\frac{\partial f}{\partial x} = w_7 - \cancel{w_4} + w_8 - \cancel{w_5} + w_9 - \cancel{w_6} + \cancel{w_4} - w_1 + \cancel{w_5} - w_2 + \cancel{w_6} - w_3$$

$$= w_7 + w_8 + w_9 - (w_1 + w_2 + w_3)$$

$$\therefore \frac{\partial f}{\partial x} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{\partial f}{\partial y} = w_3 - \cancel{w_2} + w_6 - \cancel{w_5} + w_9 - \cancel{w_8} + \cancel{w_2} - w_1 + \cancel{w_5} - w_4 + \cancel{w_8} - w_7$$

$$= w_3 + w_6 + w_9 - (w_1 + w_4 + w_7)$$

$$\therefore \frac{\partial f}{\partial y} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

From the definition ; one-dimensional function $f(x)$ is

In x-direction, $\frac{\partial f}{\partial x} = f(x+1) - f(x)$.

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

Two-dimensional function $f(x,y)$ is

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

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

$$\therefore \nabla f(x,y) = f(x+1, y) + f(x, y+1) - 2f(x,y).$$

Eg :- Take an image as $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$

Its equivalent is

$$\begin{pmatrix} 0 & 0 & 0 & & \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 & 8 \\ & 1 & 2 & 3 & 4 \end{pmatrix}$$

$$\Rightarrow \text{abs}\left[\Sigma\begin{bmatrix}0&0&0\\0&1&2\\0&5&6\end{bmatrix} * \begin{bmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{bmatrix} + \Sigma\begin{bmatrix}0&0&0\\0&1&2\\0&5&6\end{bmatrix} * \begin{bmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{bmatrix}\right]$$

$$\Rightarrow \begin{bmatrix}0&0&0\\2&2&2\\6&6&6\end{bmatrix} + \begin{bmatrix}0&0&0\\-3&0&3\\-11&0&11\end{bmatrix}$$

$$\Rightarrow 24+0 = \underline{24}$$ [Negative values are not taken]

* II-ORDER DERIVATIVE (OR) LAPLACIAN MASKING (OR) HIGH-PASS FILTER MASKING :-

By using II order we find thin lines of an image.

In this if one part gets highlighted then other parts are neglected. Usually centre part may be highlighted (or) dimmed than other pixel values.

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

Let the image is a 2-D image, then

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

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

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

Eg :-

(i) $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ (ii) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ (iii) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ (iv) $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Advantage :-

By applying Laplacian masking, brightness increases once brightness increases we can easily identify the edges & boundaries of image.

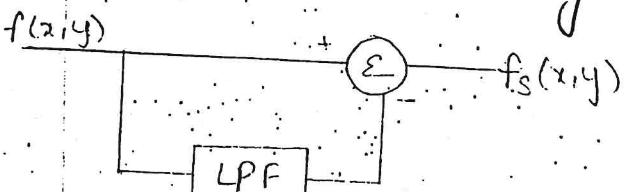
Dis-advantage :-

Because of Laplacian masking noise gets increased

* UNSHARP MASKING :-

The main objective of 'Unsharp masking' is to increase the contrast of an image.

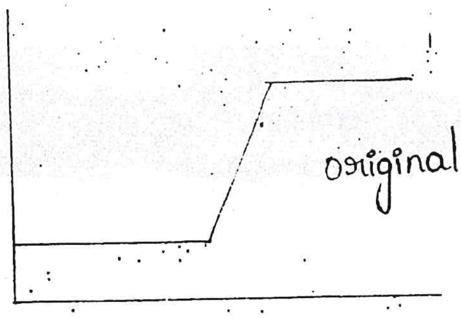
The brightness can be increased by reducing the low pass components & enhancing high pass components



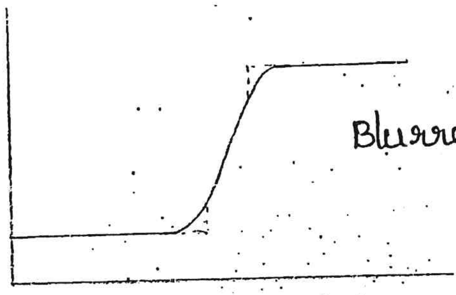
$$o/p \quad f_s(x,y) = f(x,y) - f_{LP}(x,y)$$

Unsharp masking involves the following steps :

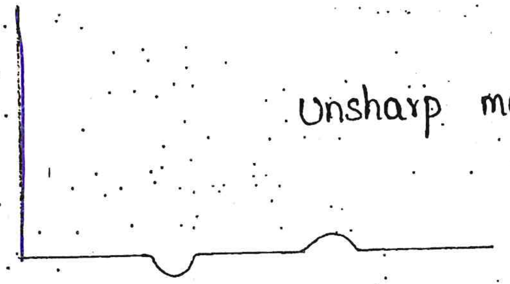
1. Blurring the original image.
2. Subtracting the blurred image from original. image and add masking to the original image



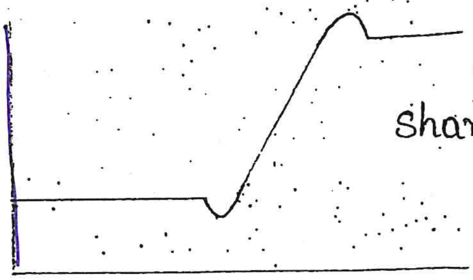
original signal



Blurred signal



Unsharp mask



Sharpened signal.

* HIGH BOOST FILTERING :-

For sharpening the image & to increase the centre pixel value we go for high boost filtering.

we know that

$$f_s(x,y) = A\,f(x,y) - f_{LP}(x,y)$$

$$= A\,f(x,y) + f(x,y) - f(x,y) - f_{LP}(x,y)$$

[Adding & Subtracting $f(x,y)$]

$$= (A-1) f(x,y) + f_s(x,y)$$

For the purpose of sharping, we use Laplacian transform which takes the form

$$f_s(x,y) = (A-1) f(x,y) + \nabla^2 f$$

↳ Laplacian function.

Eg:- The examples for high boost filtering are:

(i) $\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ (ii) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & A-4 & \phi \\ 0 & 1 & 0 \end{bmatrix}$ (iii) $\begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ (iv) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & A-8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

* HOMO-MORPHIC FILTERING :-

Image is a combination of illumination & reflectance i.e.,

$$f(x,y) = i(x,y) \cdot r(x,y) \quad \text{——} \; ①$$

Reflection term contains high pass components & Illumination term contains low pass components.

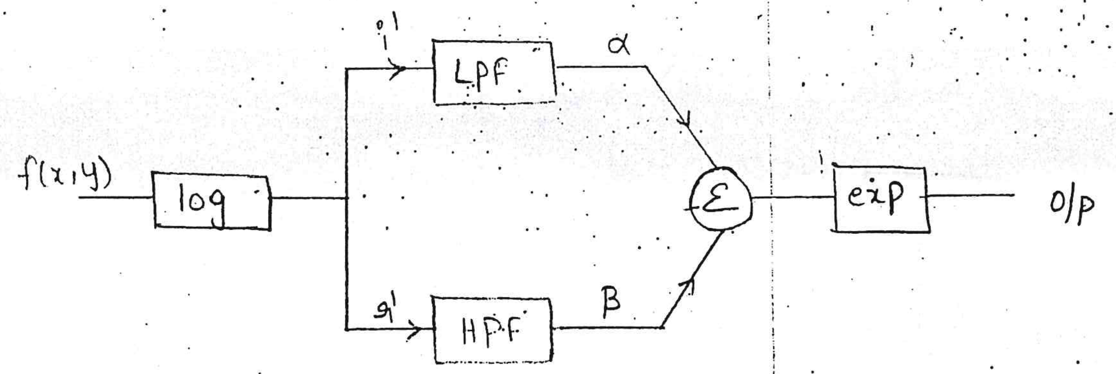
Inorder to seperate low pass & high pass components we have to apply logarithm.

① becomes

$$\log(i(x,y) \cdot r(x,y)) = \log(i(x,y)) + \log(r(x,y))$$

$$\Rightarrow \quad f'(x,y) = i'(x,y) + r'(x,y) \quad \text{——} \; ②$$

Now $i'(x,y)$ is given to LPF & $r'(x,y)$ is given to HPF. These terms (i' & r') are multiplied with α & β



NOW ② becomes,

$$f'(x,y) = \alpha\, i'(x,y) + \beta\, r'(x,y)$$

By applying exponential, we get

$$f'(x,y) = e^{\alpha i'(x,y) + \beta r'(x,y)}$$

$$= e^{\alpha \log(i(x,y)) + \beta \log(r(x,y))}$$

$$= e^{\log i(x,y)^{\alpha} + \log r(x,y)^{\beta}}$$

$$[\because m \log a = \log a^m]$$

$$= e^{\log [i(x,y)^{\alpha} \cdot r(x,y)^{\beta}]}$$

$$[\because \log m + \log n = \log mn]$$

$$\therefore f'(x,y) = i_{(x,y)}^{\alpha} \cdot r_{(x,y)}^{\beta}$$

$$\boxed{f(x,y)} \rightarrow \boxed{\log} \rightarrow \boxed{FFT} \rightarrow \boxed{IFT} \rightarrow o/p$$

This gives about Homo-morphic technique.

* SOBEAL MASKING :-

By using sobeal masking sharp edges can be found. It is also similar to Gradient filter but the centre part is doubled.

From the equations of I-order derivative, $\frac{\partial f}{\partial x}$ & $\frac{\partial f}{\partial y}$ becomes

$$\frac{\partial f}{\partial x} = (w_7 + 2w_8 + w_9) - (w_1 + 2w_2 + w_3)$$

$$\frac{\partial f}{\partial y} = (w_3 + 2w_6 + w_9) - (w_1 + 2w_4 + w_7)$$

From the above eqn's we can observe that the centre part is doubled.

$$f(x,y) = |(w_7 + 2w_8 + w_9) - (w_1 + 2w_2 + w_3)| + |(w_3 + 2w_6 + w_9) - (w_1 + 2w_4 + w_7)|$$

Eg:- The examples for Sobeal masking are

(i) $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ w.r.t 'x' (ii) $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ w.r.t 'y'

* ROBERT MASKING :-

This masking is also known as Gradient (or) First order filter.

In this masking we take the cross differences.

Let us consider an image $\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$

Here $f(x,y) = \left[(w_9 - w_5)^2 + (w_8 - w_6)^2\right]^{1/2}$.

* By applying Robert masking we can find out the diagonal values i.e., 45^0 & -45^0

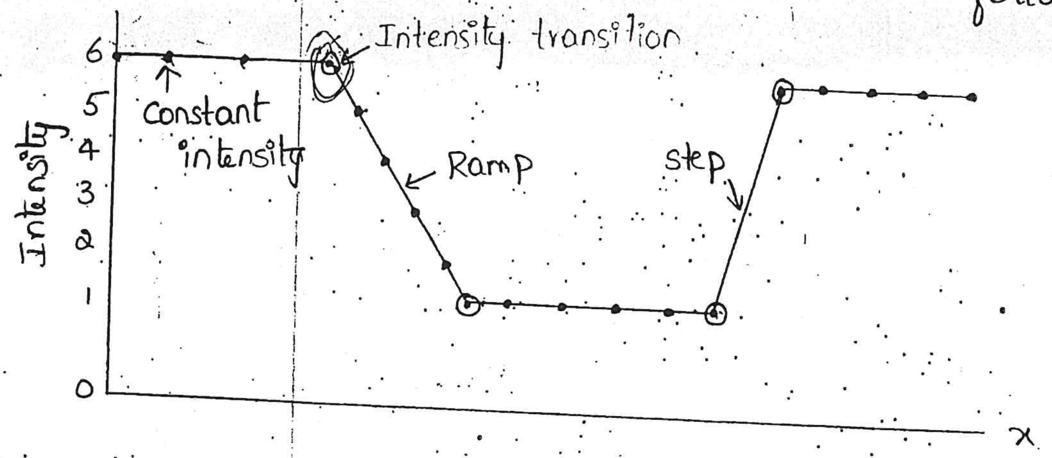
Eg :- The examples for Robert masking are

(i) $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ (ii) $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

* DIFFERENCES b/w I-ORDER & II-ORDER DERIVATIONS :-

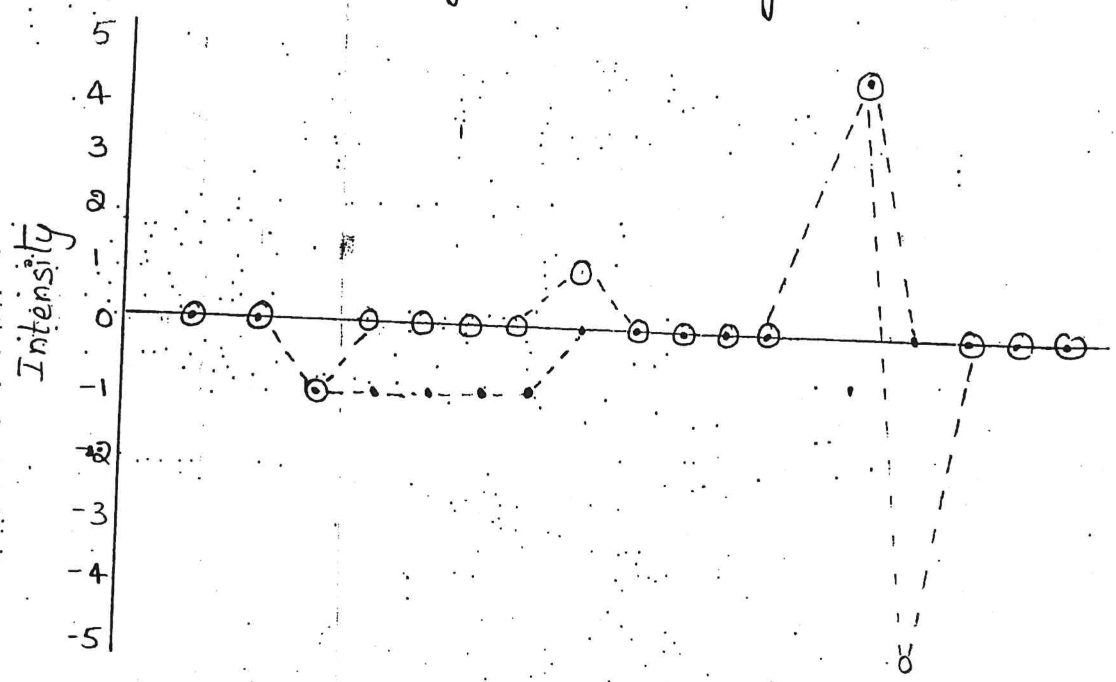
I-order derivative	II-order derivative.
1. The first order derivative of one dimensional function is $\frac{\partial f}{\partial x} = f(x+1) - f(x)$.	1. The II-order derivative of 1-D function f(x) is $\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$.
2. In areas of constant intensity the I-order derivative value is zero.	2. It is also zero at constant areas.
3. For unit-step function its value is non-zero.	3. II-order value is also non-zero at unit step function.
4. Along ramp functions also its value is non-zero.	4. This derivative value must be zero along ramp functions.
5. For isolation case, the value of I-order derivative is its peak value.	6. The II-order derivative value is doubled for isolation case.

Eg:- To get a clear view about the differences b/w I & II order derivations consider the following



Scan line	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
I- derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	5	0	0	0	0
II derivative	0	0	-1	0	0	0	0	1	0	0	0	0	5	-5	0	0	0

The corresponding o/p image is obtained as



• → I - derivative

o → II - derivative

⊙ → same values for both derivative

* COMBINING SPATIAL ENHANCEMENT METHODS :-

We know that, to obtain a task we require applications of several complementary techniques inorder to achieve an acceptable result. The main objective of combining spatial enhancement method is to enhance the image by sharpening it by combining various techniques.

We use Laplacian to highlight the prominent edges & to increase the dynamic range of the intensity level we use intensity transformation.

Median filter is used to reduce noise. However Median filtering is a non-linear process capable of removing image features, this is unacceptable in medical image processing.

The gradient has a stronger response in areas of ramp & step functions. The Laplacian function produces higher noise than Gradient. The noise can be further lowered by smoothing the gradient with averaging filter. By using Sobeal masking we can sharp the edges of an image. The smallest possible value of gradient image is zero.

By using the product of Laplacian & Smoothed gradient we can increase the sharpness of the image This type of improvement would not have been possible by using the Laplacian or gradient alone.

The dynamic range can be sharpened by using power law transformation. Histogram equalisation is not suitable for this purpose since it has dark image distributions. For this case it is better to use Histogram specification.

Eg:- These techniques are found in printing industry, in image based product inspection, in forensics, in microscopy, in surveillance.

Basics of filtering in the frequency domain:

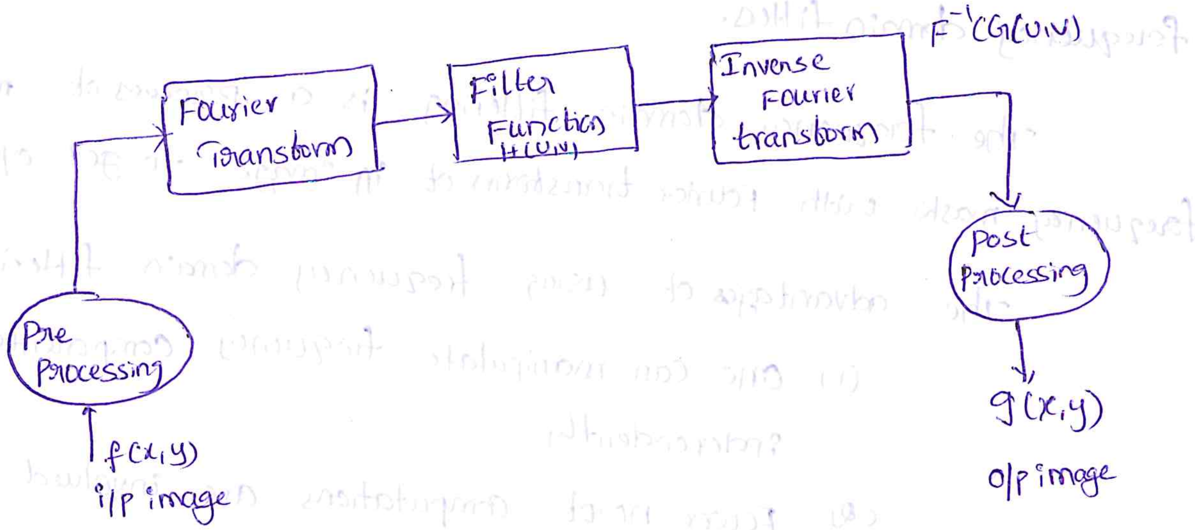


fig: Basic steps for filtering in frequency domain

The algorithm for frequency domain filtering is given as

(1) Let f(x,y) be original image for which filtering is required. obtain Fourier transform of image. Read the spectrum and multiply by $(-1)^{x+y}$ to centre the transform

(2) Design a frequency domain filter matrix function h(x,y). The mask can be any shape depends on application requirement. obtain Fourier transform of h(x,y) to get H(u,v).

(3) Multiply the Fourier spectrum of the filter with the Fourier spectrum of the image by element wise multiplication

$$G(u,v) = H(u,v) \cdot F(u,v)$$

↑ o/p image in Frequency domain

(4) Apply inverse Fourier transform to G(u,v) to retrieve the filtered image in spatial domain.

(5) Extract real components and multiply by $(-1)^{x+y}$ to offset the effect in step 1

6. Display the images and exit.

This algorithm is general and used to implement many frequency domain filters.

The frequency domain filtering is a process of multiplying frequency mask with Fourier transform of i/p image to get o/p image.

The advantages of using frequency domain filtering are

(1) One can manipulate frequency components independently

(2) Fewer no.of computations are involved.

The spatial domain filtering is flexible upto 9x9 mask but for larger masks filtering in frequency domain is preffered.

For example:

consider a mask H(D). If all the values of mask are 1 i.e

H(D) = 1. ⇒ It represents zero attenuation where all the frequency components are allowed

H(D) = 0 ⇒ It represents maximum attenuation where all the frequency components are blocked.

By Controlling the weights of mask, we can control the attenuation of frequency components.

Image smoothing in Frequency domain:

An ideal Lowpass filter which allows the frequencies up to a certain cutoff frequency and removes all frequencies beyond that, then the transfer function is given as

$$H(D) = \begin{cases} 1 & \text{for } D \le D_0 \\ 0 & \text{for } D > D_0 \end{cases}$$

By multiply F(D) in 1D preserves with H(D) which preserves the frequencies upto D_0

Similarly highpassfilter which allows the frequencies more than cutoff frequencies and removes all the frequencies below it., then the transfer function is given as

$$H(D) = \begin{cases} 1 & \text{for } D > D_0 \\ 0 & \text{for } D \le D_0 \end{cases}$$

2D image:

In general images are two Dimensional. Here the transfer function should be applied first along the rows (M) of image and the results should be stored in intermediate image. Then H(D) applied to columns of intermediate image to yield a 2D mask.

A more effective approach is to use a single filter and apply radially along the frequency range of image. Some of the masks are

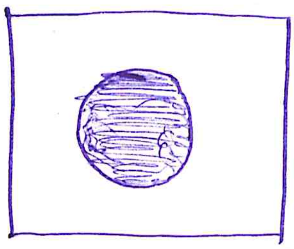


fig: low pass filter circular mask

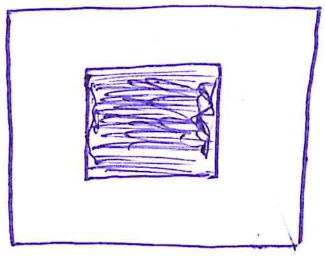


fig: square mask

The masks can be rectangular, circular & any shape

The centre of frequency rectangle is

$$(u,v) = \left(\frac{M}{2}, \frac{N}{2}\right)$$

The radial frequency

$$D(u,v) = \left[\left(u-\frac{M}{2}\right)^2 + \left(v-\frac{N}{2}\right)^2\right]^{\frac{1}{2}}$$

In 2D the radial cutoff frequency is D_0 and it is specified in terms of pixels. For circular mask the cutoff freq is the radius of circle.

For 2D image

The transfer function of lowpass filter is

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

The transfer function for a highpass filter is

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

→ The ideal lowpass filter produces ringing effect which is also known as Gibbs ringing. i.e decreasing the intensities in parallel to edges.

TO overcome this effect use

→ Gaussian lowpass filters

→ Butterworth lowpass filters

The transferfunction on Gaussian filter mask is

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

where D_0: cutoff frequency

The values of mask changes from 0 to 1. The Gaussian mask is controlled by σ, as the value of σ changes the cut off frequency changed. The Gaussian filter never cause ringing artefacts

The transferfunction on Butterworth filter mask is

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

$n \rightarrow$ order of filter

$D_0 \rightarrow$ cutoff frequency

$H : 0 \rightarrow 1$ mask magnitude

As n value increases the filter becomes sharper with increased in ringing artefacts.

if $n=0$ No ringing effect

$n=2$ Small amount of ringing present.

Image sharpening in frequency domain:

High pass filter equivalents are used to attenuate the low frequency components and allows high frequency components such as edges, boundaries and other abrupt changes of image.

The transfer function of high pass filter is

$$H_{hp}(u,v) = 1 - H_{lp}(u,v).$$

where $H_{lp}(u,v)$ = transfer function of LPF.

✓ High pass filters doesn't have ringing effects because it eliminates the zero (DC) components.

The transfer function for High pass Gaussian filter is

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

The transfer function for high pass Butterworth filter is

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

$n \rightarrow$ order of filter that gives sharpness of cut off value

→ Frequency emphasis filter is used for image sharpening this filter emphasizes frequencies by adding a portion of high frequencies to the image. It is given as

$$g(x,y) = IFFT\left[\left[1 + K(1 - H_{LP}(u,v))\right] F(u,v)\right]$$

$1 + K(1 - H_{LP}(u,v))$ is a term called as high freq emphasis filter.

The parameter k controls the proportion of high frequencies in the image. The most general form of filter is

$$g(x,y) = IFFT\{[(k_1+k_2) H_{Lp}(u,v)]\ F(u,v)\}$$

k_1 controls offset

k_2 controls contribution of high frequencies.

<u>Selective Filtering:</u>

Selective filters are allows & blocks the frequency components within its range. Some of those are

1. Band pass filters
2. Band stop filters
3. Notch filters.

<u>Band pass filters:</u>

Band pass filters allow frequency components if they fall in the range $D_l - D_h$.

For 1D Band pass filters

$$H(D) = \begin{cases} 1 & \text{for } D_l \le D \le D_h \\ 0 & \text{for } D \ge D_0 \end{cases}$$

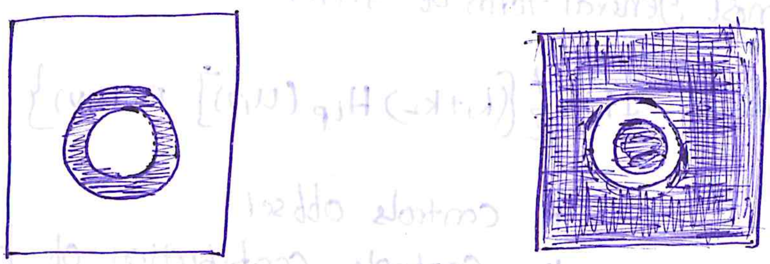
Here D_0 is cutoff frequency

For 2D Band pass filters the transfer function

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

Here D_0 is cutoff freq.

$D(u,v)$ is the Distance of the point (u,v) from the centre and W is width of Band



(a) Band pass filter

(b) Band reject filter

Frequency Domain mask.

Band reject filters:

The Band reject filter which blocks the frequency components in its range $D_l - D_h$. It is the complement of Band Pass filter.

For 1D, the transfer function is

$$H(D) = \begin{cases} 0 & \text{for } D_l < D \leq D_h \\ 1 & \text{for } \cancel{\text{else}}\ D \geq D_0 \end{cases}$$

For 2D image the transfer function is

$$H(u,v) = \begin{cases} 0 & \text{for } D_0 - \frac{\omega}{2} \leq D(u,v) \leq D_0 + \frac{\omega}{2} \\ 1 & \text{for else} \end{cases}$$

The transfer function for Band reject filter $= 1 - H_{BP}(u,v)$

Notch filters: A Notch filter is a special form Band reject filters. Instead of removing the entire range of frequencies, it only removes selective frequency components.

It is used to remove periodic noise and ringing effects and also removes the electrical interference caused by electrical disturbance.

Homomorphic filtering:

This filtering process an image with adequate brightne by simultaneous intensity range compression and contrast enhancement. i.e reducing high intensity values and enhancing dark intensity value at a time.

An image $f(x,y)$ can be expressed as the product of its illumination $i(x,y)$ and reflectance $r(x,y)$ components

$$f(x,y) = i(x,y) \cdot r(x,y). \rightarrow (1)$$

Since $F.T[x \cdot y] \neq F.T[x] \cdot F.T[y]$, so we use logarithm to equation (1) to split the terms

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

$$g(x,y) = \ln(i(x,y)) + \ln(r(x,y)) \qquad [\because g(x,y) = \ln(f(x,y))]$$

$$F.T[g(x,y)] = F.T[\ln(i(x,y)) + \ln(r(x,y))]$$

$$G(u,v) = I'(u,v) + R'(u,v) \rightarrow (2)$$

Now by applying filtering mask $H(u,v)$, the o/p image in Frequency domain is

$$G(u,v) = H(u,v) \cdot Z(u,v)$$

$$= H(u,v) \, [I'(u,v) + R'(u,v)]$$

$$G(u,v) = H(u,v) \cdot I'(u,v) + H(u,v) R'(u,v)$$

The filtered image in spatial domain is obtained by applying inverse fourier transform

$$\Rightarrow IFT[G(u,v)] = IFT[H(u,v) \cdot I'(u,v) + H(u,v) R' \ldots$$

$\therefore H(u,v$

$$g(x,y) = i'(x,y) + r'(x,y)$$

$$g(x,y) = e^{f_g(x,y)}$$

$$= e^{i'(x,y) + r'(x,y)}$$

$$e^{f(x,y)} = e^{i'(x,y)} \cdot e^{r'(x,y)}$$

$$f_0(x,y) = i_0(x,y) \cdot r_0(x,y)$$

Algorithm for applying homomorphic filters:

1. Apply log transformation to the image i.e
   $\ln(f(x,y)) = \ln(i(x,y)) + \ln(r(x,y))$
2. Apply Fourier transform to log of these components
3. Design filters seperately for illumination and reflectance components. The transfer function of these components are different
4. Apply inverse Fourier transform to filtered image
5. To offset the logarithm applied in step 1. apply antilog function.

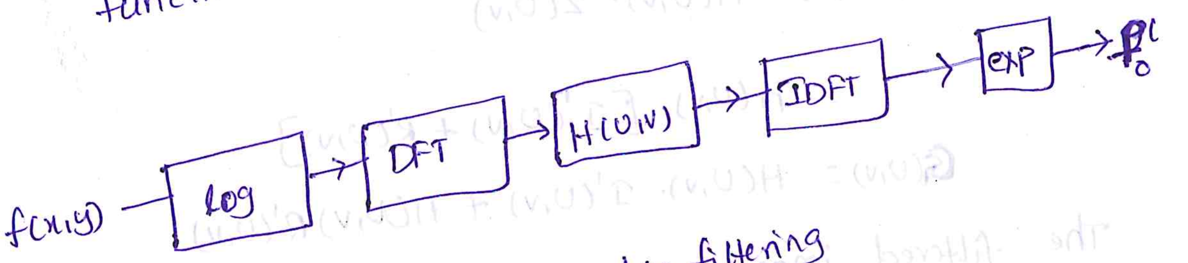


fig: Steps in Homomorphic filtering

UNIT-3

IMAGE RESTORATION

IMAGE RESTORATION:

Restoration improves image in some predefined sense. It is an objective process. Restoration attempts to reconstruct an image that has been degraded by using a priori knowledge of the degradation phenomenon. These techniques are oriented toward modeling the degradation and then applying the inverse process in order to recover the original image. Restoration techniques are based on mathematical or probabilistic models of image processing. Enhancement, on the other hand is based on human subjective preferences regarding what constitutes a "good" enhancement result. Image Restoration refers to a class of methods that aim to remove or reduce the degradations that have occurred while the digital image was being obtained. All natural images when displayed have gone through some sort of degradation:

- During display mode
- Acquisition mode, or
- Processing mode
  - Sensor noise
  - Blur due to camera mis focus
  - Relative object-camera motion
  - Random atmospheric turbulence
- Others

Degradation Model:

Degradation process operates on a degradation function that operates on an input image with an additive noise term. Input image is represented by using the notation f(x,y), noise term can be represented as η(x,y).These two terms when combined gives the result as g(x,y). If we are given g(x,y), some knowledge about the degradation function H or J and some knowledge about the additive noise teem η(x,y), the objective of restoration is to obtain an estimate *f'(x,y)* of the original image. We want the estimate to be as close as possible to the original image. The more we know about h and η , the closer f(x,y) will be to f'(x,y). If it is a linear position invariant process, then degraded image is given in the spatial domain by

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

h(x,y) is spatial representation of degradation function and symbol * represents convolution. In frequency domain we may write this equation as

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

The terms in the capital letters are the Fourier Transform of the corresponding terms in the spatial domain.

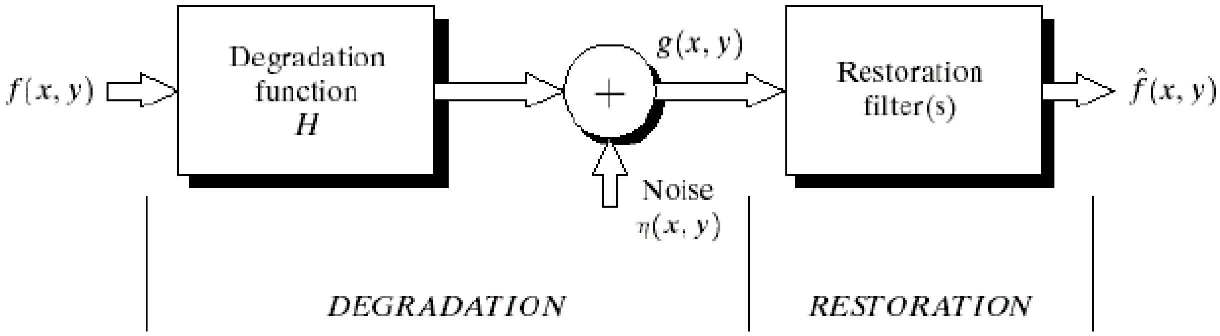


Fig: A model of the image Degradation / Restoration process

Noise Models:

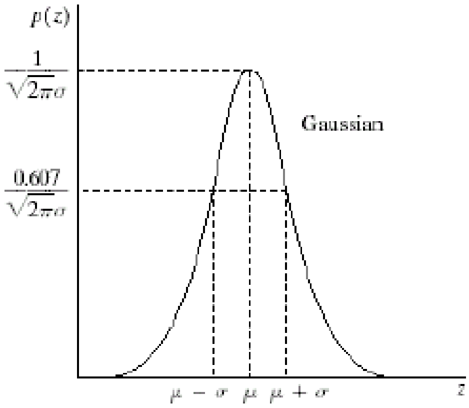
The principal source of noise in digital images arises during image acquisition and /or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition and by the quality of the sensing elements themselves. Images are corrupted during transmission principally due to interference in the channels used for transmission. Since main sources of noise presented in digital images are resulted from atmospheric disturbance and image sensor circuitry, following assumptions can be made i.e. the noise model is spatial invariant (independent of spatial location). The noise model is uncorrelated with the object function.

Gaussian Noise:

These noise models are used frequently in practices because of its tractability in both spatial and frequency domain. The PDF of Gaussian random variable is

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

Where z represents the gray level, μ= mean of average value of z, σ= standard deviation.



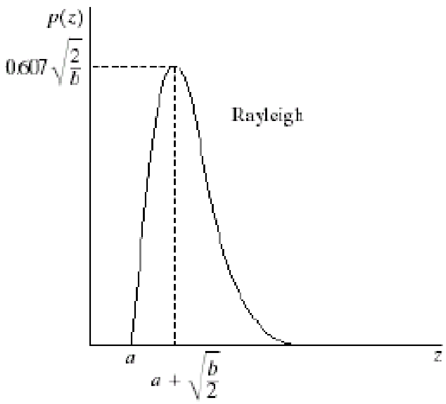
Rayleigh Noise:

Unlike Gaussian distribution, the Rayleigh distribution is no symmetric. It is given by the formula.

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

The mean and variance of this density is

$$m = a + \sqrt{\pi b/4}, \sigma^2 = \frac{b(4-\pi)}{4}$$



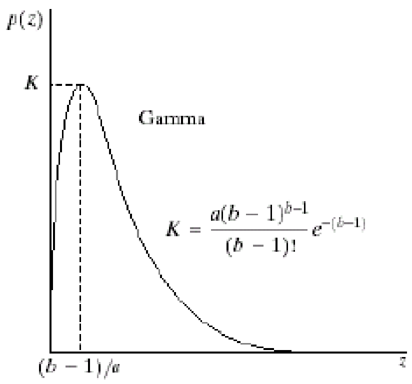
(iii)Gamma Noise:

The PDF of Erlang noise is given by

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

The mean and variance of this density are given by

$$\text{mean}: \mu = \frac{b}{a} \quad \text{variance}: \sigma^2 = \frac{b}{a^2}$$



Its shape is similar to Rayleigh disruption. This equation is referred to as gamma density it is correct only when the denominator is the gamma function.

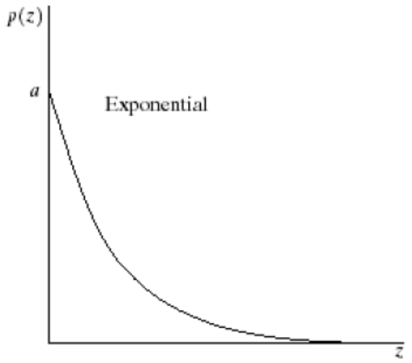
(iv)Exponential Noise:

Exponential distribution has an exponential shape. The PDF of exponential noise is given as

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

Where a>0. The mean and variance of this density are given by

$$m = \frac{1}{a}, \quad \sigma^2 = \frac{1}{a^2}$$



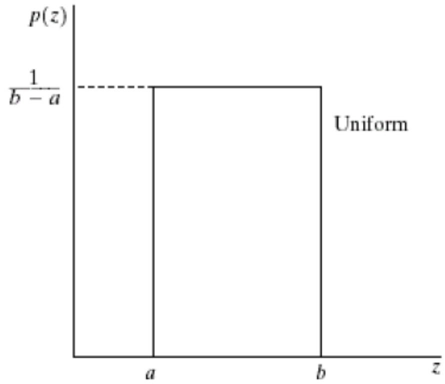
(v)Uniform Noise:

The PDF of uniform noise is given by

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

The mean and variance of this noise is

$$m = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}$$



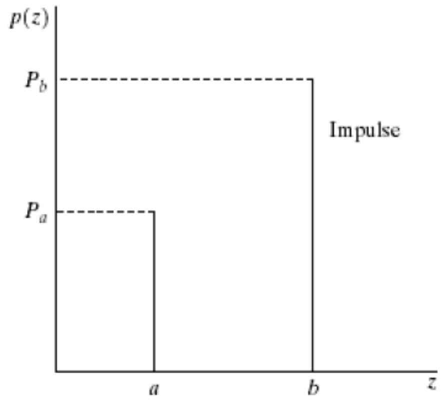
(vi)Impulse (salt & pepper) Noise:

In this case, the noise is signal dependent, and is multiplied to the image.

The PDF of bipolar (impulse) noise is given by

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

If b>a, gray level b will appear as a light dot in image. Level a will appear like a dark dot.



Restoration in the presence of Noise only- Spatial filtering:

When the only degradation present in an image is noise, i.e.

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

or

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

The noise terms are unknown so subtracting them from g(x,y) or G(u,v) is not a realistic approach. In the case of periodic noise it is possible to estimate N(u,v) from the spectrum G(u,v).

So N(u,v) can be subtracted from G(u,v) to obtain an estimate of original image. Spatial filtering can be done when only additive noise is present. The following techniques can be used to reduce the noise effect:

i) Mean Filter:

ii) (a)Arithmetic Mean filter:

It is the simplest mean filter. Let Sxy represents the set of coordinates in the sub image of size m*n centered at point (x,y). The arithmetic mean filter computes the average value of the corrupted image g(x,y) in the area defined by Sxy. The value of the restored image f at any point (x,y) is the arithmetic mean computed using the pixels in the region defined by Sxy.

$$\hat{f}(x,y)=\frac{1}{mn}\sum_{(s,t)\in Sxy} g(s,t)$$

This operation can be using a convolution mask in which all coefficients have value 1/mn A mean filter smoothes local variations in image Noise is reduced as a result of blurring. For every pixel in the image, the pixel value is replaced by the mean value of its neighboring pixels with a weight .This will resulted in a smoothing effect in the image.

(b)Geometric Mean filter:

An image restored using a geometric mean filter is given by the expression

$$\hat{f}(x,y)=\left(\prod_{(s,t)\in Sxy} g(s,t)\right)^{1/mn}$$

Here, each restored pixel is given by the product of the pixel in the sub image window, raised to the power 1/mn. A geometric mean filters but it to loose image details in the process.

(c)Harmonic Mean filter:

The harmonic mean filtering operation is given by the expression

$$\hat{f}(x,y)=\sum_{(s,t)\in Sxy} g(s,t)^{Q+1} \Big/ \sum_{(s,t)\in Sxy} g(s,t)^{Q}$$

The harmonic mean filter works well for salt noise but fails for pepper noise. It does well with Gaussian noise also.

(d)Order statistics filter:

Order statistics filters are spatial filters whose response is based on ordering the pixel contained in the image area encompassed by the filter. The response of the filter at any point is determined by the ranking result.

(e)Median filter:

It is the best order statistic filter; it replaces the value of a pixel by the median of gray levels in the Neighborhood of the pixel.

$$\hat{f}(x,y) = \underset{(s,t)\in Sxy}{\text{median}}\{g(s,t)\}$$

The original of the pixel is included in the computation of the median of the filter are quite possible because for certain types of random noise, the provide excellent noise reduction capabilities with considerably less blurring then smoothing filters of similar size. These are effective for bipolar and unipolor impulse noise.

(e)Max and Min filter:

Using the l00th percentile of ranked set of numbers is called the max filter and is given by the equation

$$\hat{f}(x,y) = \max_{(s,t)\in Sxy}\{g(s,t)\}$$

It is used for finding the brightest point in an image. Pepper noise in the image has very low values, it is reduced by max filter using the max selection process in the sublimated area sky. The 0th percentile filter is min filter.

$$\hat{f}(x,y) = \min_{(s,t)\in Sxy}\{g(s,t)\}$$

This filter is useful for flinging the darkest point in image. Also, it reduces salt noise of the min operation.

(f)Midpoint filter:

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by

$$\hat{f}(x,y) = \left(\max_{(s,t)\in Sxy}\{g(s,t)\} + \min_{(s,t)\in Sxy}\{g(s,t)\}\right)\Big/2$$

It comeliness the order statistics and averaging .This filter works best for randomly distributed noise like Gaussian or uniform noise.

Periodic Noise by Frequency domain filtering:

These types of filters are used for this purpose-

Band Reject Filters:

It removes a band of frequencies about the origin of the Fourier transformer.

Ideal Band reject Filter:

An ideal band reject filter is given by the expression

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

D(u,v)- the distance from the origin of the centered frequency rectangle.

W- the width of the band

Do- the radial center of the frequency rectangle.

Butterworth Band reject Filter:

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

Gaussian Band reject Filter:

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

These filters are mostly used when the location of noise component in the frequency domain is known. Sinusoidal noise can be easily removed by using these kinds of filters because it shows two impulses that are mirror images of each other about the origin. Of the frequency transform.

FIGURE From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Band pass Filter:

The function of a band pass filter is opposite to that of a band reject filter It allows a specific frequency band of the image to be passed and blocks the rest of frequencies. The transfer function of a band pass filter can be obtained from a corresponding band reject filter with transfer function Hbr(u,v) by using the equation

$$H_{BP}(u,v)=1-H_{BR}(u,v)$$

These filters cannot be applied directly on an image because it may remove too much details of an image but these are effective in isolating the effect of an image of selected frequency bands.

Notch Filters:

A notch filter rejects (or passes) frequencies in predefined neighborhoods about a center frequency.

Due to the symmetry of the Fourier transform notch filters must appear in symmetric pairs about the origin.

The transfer function of an ideal notch reject filter of radius D_0 with centers a (u_0 , v_0) and by symmetry at ($-u_0$, v_0) is

$$D_1(u,v) = \sqrt{(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2}$$
$$D_2(u,v) = \sqrt{(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2}$$

Ideal, butterworth, Gaussian notch filters

$$H(u,v) = \begin{cases} 0 & \text{if} \quad D_1(u,v) \le D_0 \text{ or } D_2(u,v) \le D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$H(u,v) = 1 \Bigg/ \left[1 + \left(\frac{D_0^2}{D_1(u,v) D_2(u,v)} \right)^n \right]$$

$$H(u,v) = 1 - \exp\left[-\frac{1}{2} \left(\frac{D_1(u,v) D_2(u,v)}{D_0^2} \right) \right]$$

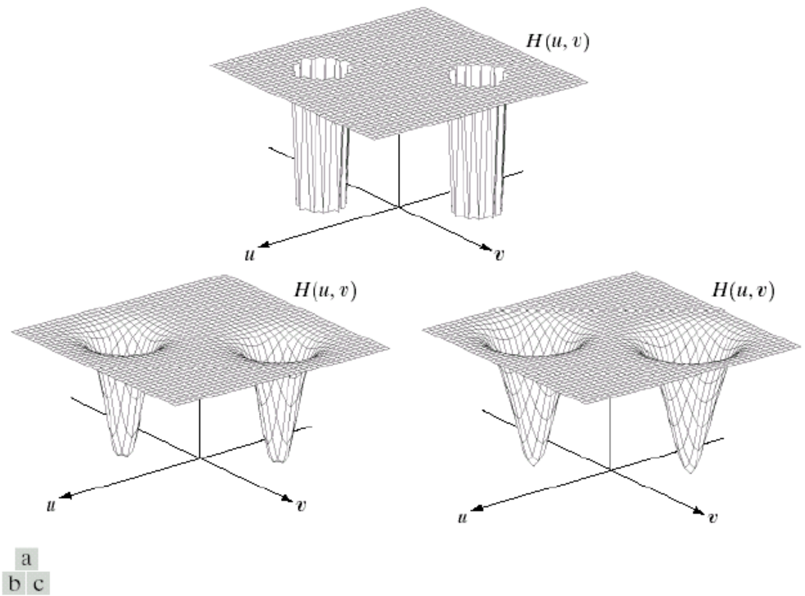


FIGURE Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Inverse Filtering:

The simplest approach to restoration is direct inverse filtering where we complete an estimate $\hat{F}(u,v)$ of the transform of the original image simply by dividing the transform of the degraded image G(u,v) by degradation function H(u,v)

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

We know that

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

Therefore

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

From the above equation we observe that we cannot recover the undegraded image exactly because N(u,v) is a random function whose Fourier transform is not known.

One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin.

We know that H(0,0) is equal to the average values of h(x,y).

By Limiting the analysis to frequencies near the origin we reduse the probability of encountering zero values.

Minimum mean Square Error (Wiener) filtering:

The inverse filtering approach has poor performance. The wiener filtering approach uses the degradation function and statistical characteristics of noise into the restoration process.

The objective is to find an estimate $\hat{f}$ of the uncorrupted image *f* such that the mean square error between them is minimized.

The error measure is given by

$$e^2 = E\{[f(x) - \hat{f}(x)]^2$$

Where E{.} is the expected value of the argument.

We assume that the noise and the image are uncorrelated one or the other has zero mean.

The gray levels in the estimate are a linear function of the levels in the degraded image.

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

Where H(u,v)= degradation function

H*(u,v)=complex conjugate of H(u,v)

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

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

$S_f(u,v)=|F(u,v)|^2$= power spectrum of the underrated image

The power spectrum of the undegraded image is rarely known. An approach used frequently when these quantities are not known or cannot be estimated then the expression used is

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

Where K is a specified constant.

Constrained least squares filtering:

The wiener filter has a disadvantage that we need to know the power spectra of the undegraded image and noise. The constrained least square filtering requires only the knowledge of only the mean and variance of the noise. These parameters usually can be calculated from a given degraded image this is the advantage with this method. This method produces a optimal result. This method require the optimal criteria which is important we express the

$$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y)$$ in vector-matrix form

$$\mathbf{g} = \mathbf{Hf} + \boldsymbol{\eta}$$

The optimality criteria for restoration is based on a measure of smoothness, such as the second derivative of an image (Laplacian).

The minimum of a criterion function C defined as

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

Subject to the constraint

$$\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\|^2 = \|\boldsymbol{\eta}\|^2$$

Where $\|\mathbf{w}\|^2 \triangleq \mathbf{w}^T\mathbf{w}$ is a euclidean vector norm $\hat{\mathbf{f}}$ is estimate of the undegraded image. ∇^2 is laplacian operator.

The frequency domain solution to this optimization problem is given by

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

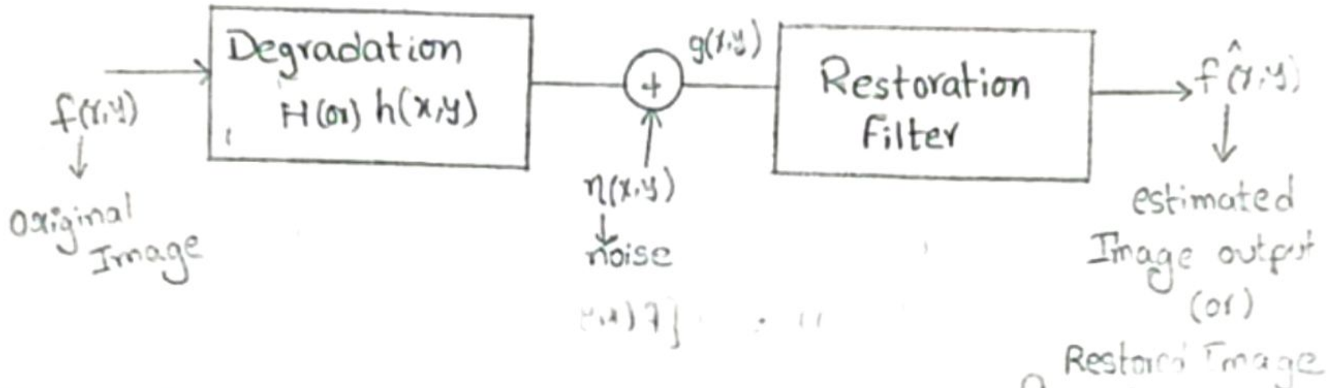
Where γ is a parameter that must be adjusted so that the constraint is satisfied.

P(u,v) is the Fourier transform of the laplacian operator

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

IV Image Restoration And Reconstruction

Image Degradation Model:-



I) Image Degradation:-

The process of degrading (or) corrupting original image $f(x,y)$ by using Degradation function 'H' is known as Image Degradation Process. When we transmit the original degraded image through wired (or) wireless communication channel, there may be possibility of occurence of noise in the transmitted image. Therefore the degraded image is given as

$$g(x,y) = f(x,y) * h(x,y) + \eta(x,y) \longrightarrow (1)$$

Ex:- Example of degradation process is Image blurring

In image blurring process, we use average filter to get blurred image (i.e) multiplying (Convolution) the degraded function with original image gives degraded (or) blurred image.

Image Restoration:-

The process of obtaining closest match (or) estimated output image from degraded image is known as

Image Restoration Process.

Ex:- Getting original image from blurred image is known as Image Restoration Process.

Linear Position - Invarient Degradation:-

The expression for degraded image is given as

$$g(x,y) = H[f(x,y)] + \eta(x,y) \longrightarrow (1)$$

If $\eta(x,y) = 0$, then $\rightarrow (1)$ becomes as

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

Linear Properity:-

A system is said to be linear if

$$H[a f_1(x,y) + b f_2(x,y)] = a H[f_1(x,y)] + b H[f_2(x,y)] \longrightarrow (2)$$

a, b are scalar constants.

$f_1(x,y)$, $f_2(x,y)$ are any two input images

If $a=b=1$ then eq (2) becomes as

$$H[f_1(x,y) + f_2(x,y)] = H[f_1(x,y)] + H[f_2(x,y)] \longrightarrow (3)$$

Equation (3) is called the additive property; this property says that, the response to a sum of two inputs is equal to sum of individual responses.

In eq (2) if $f_2(x,y) = 0$, then eq (2) can be written as

$$H[a f_1(x,y)] = a H[f_1(x,y)] \longrightarrow (4)$$

Equation (4) is known as property of Homogenity; this property says that response to a constant (or) multiple of any input is equal to the response to that input multiplied by same constant 'a'.

Thus linear operator posses both additive and homogenity properties.

(x) An operator (or) system is said to be position invarient if

$$H[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta). \text{ for any } f(x,y) \,\&\, \alpha, \beta \rightarrow (5)$$

Equation (5) says that response of $f(x,y)$ at any point (α, β) depends only on the value of the input at the point (α,β) but not on its position.

$f(x,y)$ in terms of impulse response is given as

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\, \delta(x-\alpha, y-\beta)\, d\alpha\, d\beta \longrightarrow (6)$$

$$g(x,y) = H[f(x,y)] = H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha, y-\beta)d\alpha d\beta\right] \quad [\because \text{from } (a)]$$

If 'H' is a linear operator, then above equation can be written as

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta) H[\delta(x-\alpha, y-\beta)]\, d\alpha\, d\beta \longrightarrow (6a)$$

let $h(x,\alpha,y,\beta) = H[\delta(x-\alpha, y-\beta)] \longrightarrow (7)$
$\downarrow$
Impulse response of 'H'.

Substituting (7) in (6a), we get

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\, h(\alpha,x,\beta,y)\, d\alpha\, d\beta$$

If 'H' is position invarient, then eq (7) becomes as

$$h(x-\alpha, y-\beta) = H[\delta(x-\alpha, y-\beta)]$$

Substitute above equation in (6a) gives

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\, h(x-\alpha, y-\beta)\, d\alpha\, d\beta \longrightarrow (8)$$

eq (8) is in the form of convolution expression

$$\left[\text{ie} \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau = f(t) * g(t)\right]$$

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

If we consider noise terms, then we get

$$\boxed{g(x,y) = f(x,y) * h(x,y) + \eta(x,y)} \longrightarrow (9)$$

Convolution in spatial domain is equal to product in frequency domain.

$$\Rightarrow G(u,v) = F(u,v)H(u,v) + N(u,v) \longrightarrow (10)$$

Noise Models:-

1. Impulse [salt-and-pepper] noise :-

The PDF of (bipolar) impulse noise 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 $b>a$, intensity 'b' will appear as a light dot in image. Conversely, level 'a' will appear like a dark (or) dot.

If either P_a (or) P_b is zero, the impulse noise is called unipolar.

If neither probability is zero, (if equal), impulse noise will resemble salt-and-pepper granules distributed over image.

Hence for this reason, bipolar impulse noise is also called as Salt-and-pepper noise.

2. Uniform Noise :

The PDF of uniform noise is given by

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

• The mean of this density function is given by

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

and its variance by

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

The Uniform density is useful as basis for numerous random number generators that are used for Simulations.

3. Exponential Noise :

The PDF of exponential noise is given by

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

where $a > 0$.

The mean and variance of this density function are

$$\bar{z} = \frac{1}{a}$$

$$\sigma^2 = \frac{1}{a^2}$$

The Exponential function density find application in Laser imaging.

4. Gamma (or) Erlang Noise :

The PDF of Erlang noise is given by

$$P(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

where $a > 0$; b is a positive integer.

The mean and Variance of this density function are given by

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

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

Finds application in Laser Imaging.

5. <u>Gaussian Noise</u>:

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 $z \rightarrow$ intensity

$\bar{z} \rightarrow$ mean value of z

$\sigma \rightarrow$ standard derivation, $\sigma^2 \rightarrow$ variance of z.

Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and high temperature.

6. <u>Rayleigh Noise</u>:

The PDF of rayleigh noise is given by

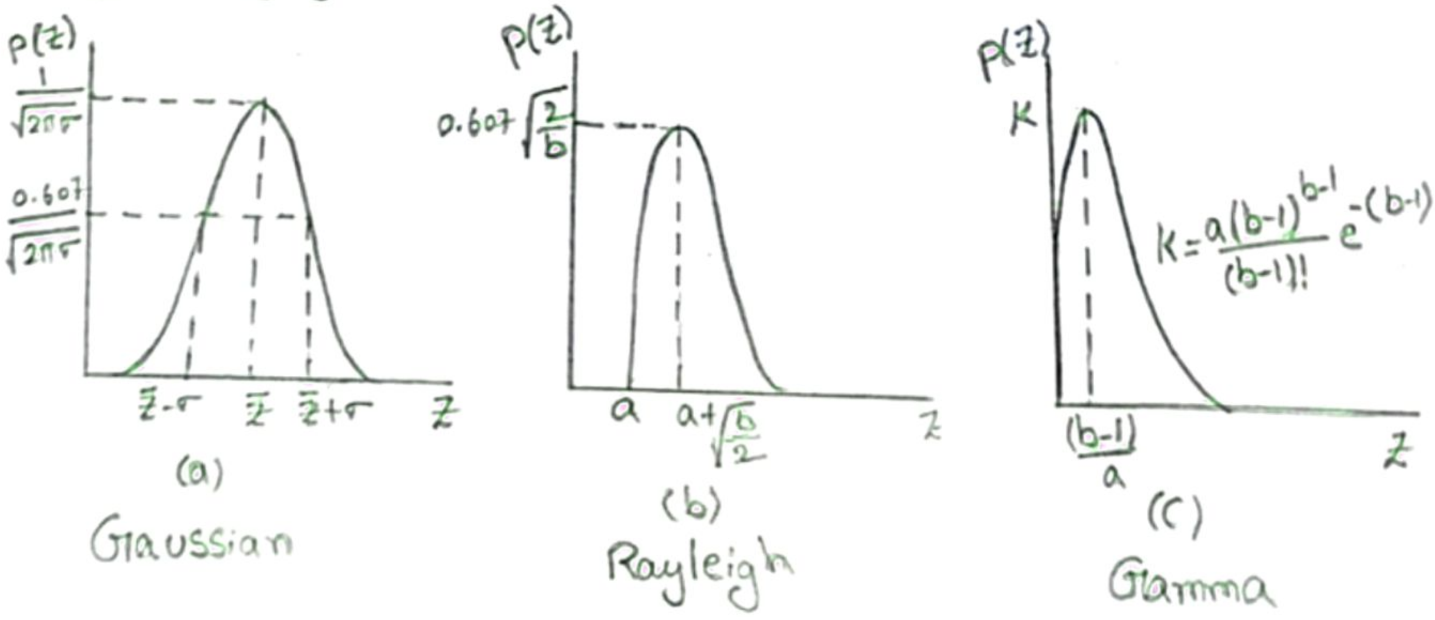
$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

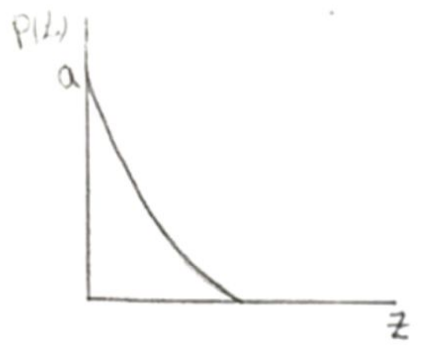
The mean and variance of this density are given by

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

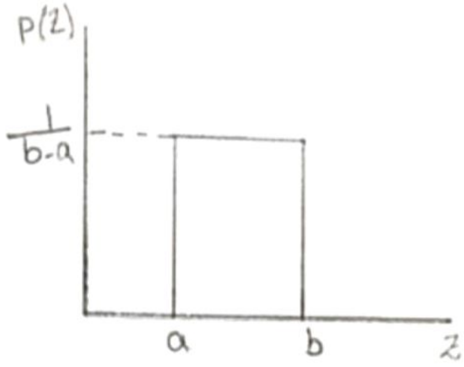
$$\sigma^2 = \frac{b(4-\pi)}{4}$$

Rayleigh density helpful in characterising noise phenomena in range imaging.

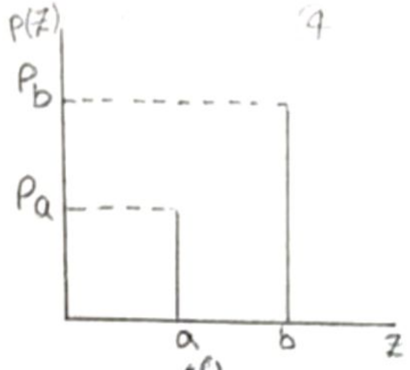




(d)
Exponential



(e)
Uniform



(f)
Impulse

Image Restoration:-

Restoration in presence of noise only - Spatial filtering:

When only degradation present in an image is noise, now the equations

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

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

become

$$g(x,y) = f(x,y) + \eta(x,y)$$
$$G(u,v) = F(u,v) + N(u,v)$$

The noise terms are unknown, so subtracting them from $g(x,y)$ (or) $G(u,v)$ is not a realistic option. In the case of periodic noise, it is usually possible to estimate $N(u,v)$ from the spectrum of $G(u,v)$. In this case $N(u,v)$ can be subtracted from $G(u,v)$ to obtain an estimate of the original image.

Spatial filtering is the method of choice in situations when only additive random noise is present.

1. Mean Filters:

Here we discuss briefly the noise reduction capabilities of the spatial filters and develop several other filters whose performance is in many cases superior to the filters.

• Arithmetic Mean Filters:

This is the simplest of the mean filters. Let S_{xy} represent the set of coordinates in a rectangular subimage window of size $m \times n$, centered at point (x,y). The arithmetic mean filters compute the average value of corrupted image $g(x,y)$ in the area defined by S_{xy}. The value of restored image f at point (x,y) is simply the arithmetic mean computed using the pixels in the region defined by $S(x,y)$.

In other words,

$$\bar{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

This operation can be implemented using a spatial filter of size $m \times n$ in which all coefficients have value $\frac{1}{mn}$. A mean filter smooths local variations in an image, and the noise is reduced as a result of blurring.

• Geometric Mean Filters:

An image restored using a Geometric mean filter is given by expression

$$\bar{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

Here each restored ~~point~~ pixel is given by the product of the pixels in the subimage window, raised to the power $1/mn$. A GM filter achieves smoothing comparable to AM filter, but it tends to lose less image detail in the process.

2. Order-Static Filters:

Order static filters are the spatial filters, whose response is based on ordering the values of the pixels contained in the image area encompassed by the filter. The ranking result determines the response of the filters. 5

- Median Filter:

The best known order-statistic filter is the median filter, which, as its name implies replaces the value of a pixel by the median of the intensity levels in the neighbourhood of that pixel.

$$\hat{f}(x,y) = \underset{(s,t) \in S_{xy}}{\text{median}} \{g(s,t)\}$$

The value of the pixel at (x,y) is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size.

- Max and Min filters:

The median represents the 50th percentile of a ranked set of numbers, but you will recall from basic statistics that ranking leads itself to many other possibilities, using the 100th percentile results in the so called max filter, given by

$$\hat{f}(x,y) = \underset{(s,t) \in S_{xy}}{\max} \{g(s,t)\}$$

- Harmonic Mean filters:

The harmonic mean filtering is given by the expression

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$$

The HM filters works well for salt noise, but fails for pepper noise. It does well also with other types of noise like Gaussian Noise.

• Contraharmonic Mean Filter:

The contraharmonic mean filters yields a restored image based on the expression

$$\bar{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

where Q is the order of the filter. For +ve values of Q, the filter eliminates pepper noise, for -ve values of Q, it eliminates salt noise. It cannot do both simultaneously. Note that the contraharmonic filter reduces to Arithmetic mean filter if Q = 0 and to the harmonic mean filter if Q = -1.

This filter is useful for finding the brightest points in an image.

The '0'th percentile filter is the min filter

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

This filter is useful for finding the darkest points in an image.

• Mid point filter:

The midpoint filters simply computes the mid point between the max and min values in the area encompassed by filter

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{g(s,t)\} + \min_{(s,t)\in S_{xy}} \{g(s,t)\}\right]$$

It works best for randomly distributed noise like Gaussian (or) uniform noise.

- **Alpha-trimmed mean filter:** (6)

Suppose that we delete the d/2 lowest and the d/2 highest intensity values if $g(s,t)$ in the neighbourhood S_{xy}. Let $g_{r}(s,t)$ represent the remaining $mn-d$ pixels. A filter formed by averaging these remaining pixels is called an 'Alpha-trimmed mean filters'.

$$\bar{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_{r}(s,t)$$

$$d = 0,1,2 \ldots mn-1$$

if $d=0$, it reduces to AM filter

if $d=mn-1$, it becomes median filter

3. **Adaptive filter:**

"A filter whose behaviour changes based on statistical characteristic of image inside the mask region."

An adaptive expression for obtaining $\hat{f}(x,y)$ is given as

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_n^2}{\sigma_L^2}\left(g(x,y) - m_L\right) \longrightarrow (1)$$

σ_n^2 = variance of overall noise corrupting $f(x,y)$ to form $g(x,y)$

$g(x,y)$ = value of noisy image at (x,y)

m_L = Local mean of the pixels (mean value under mask region)

σ_L^2 = Local variance of the pixels (varien under the mask)

1. from eq $\rightarrow$ (1), if $\sigma_n^2 = 0$ then $\hat{f}(x,y) = g(x,y)$
2. if $\sigma_L^2 > \sigma_n^2$ then $\hat{f}(x,y) \simeq g(x,y)$ i.e, filter output approaches to $g(x,y)$

Generally at edges of image $\sigma_L^2 > \sigma_n^2$

3. if $\sigma_L^2 = \sigma_n^2$ then $\hat{f}(x,y) = m_L$ i.e, filter output will be equal to local mean.

We get the ratio $\frac{\sigma_n^2}{\sigma_L^2} = 1 \Rightarrow \sigma_n^2 = \sigma_L^2$; when

$\sigma_n^2 > \sigma_L^2$ occurs

Adaptive Median Filter:-

Median filters performs well if impulse noise (Salt & pepper) is not Larger i.e, ($<$) not greater than '0.2'.

→ Adoptive median filters can handle impulse noise with probabilities Larger than 0.2 (i.e) it removes ~~stan~~ salt and pepper noise effectively.

→ Adoptive median filters perserves the image details while smoothing non-impulse noise, which a median filter can't do.

→ Adoptive filters changes the mask size during filter operation i.e it increases the mask size.

Adoptive median - filter algorithm:-

This algorithm works in two ~~ways~~ stages.

Stage A:

$$A_1 = Z_{med} - Z_{min}$$

$$A_2 = Z_{med} - Z_{max}$$

→ If $A_1 > 0$ and $A_2 < 0$ i.e $Z_{med} - Z_{min} > 0$ AND $Z_{med} - Z_{max} < 0 \Rightarrow Z_{med} > Z_{min}$ AND $Z_{med} < Z_{max}$

$\Rightarrow Z_{min} < Z_{med} < Z_{max}$

Above condition states that if Z_{med} is between min & max values of 'z' then go to stage B

(7)

Else

increase the window size [ie mask 'S_{xy}' size]

If window size => 'S_{xy}' < S_{max} repeat stage A

Else

Output Z_{med}

Stage B:

$B_1 = Z_{xy} - Z_{min}$

$B_2 = Z_{xy} - Z_{max}$

If $B_1 > 0$ AND $B_2 < 0$ => if $Z_{min} < Z_{xy} < Z_{max}$

give Z_{xy} as a output

else

represent Z_{med} as output.

Z_{min} = minimum intensity in window (or) mask S_{xy}

Z_{max} = maximum intensity value in S_{xy}

Z_{med} = median of intensity values in S_{xy}

Z_{xy} = Intensity value at coordinates (x,y)

S_{max} = maximum allowed size of S_{xy}.

Summary of algorithm:

In stage A we are checking, whether Z_{med} is a salt (max intensity - 1) (or) pepper noise (min intensity - 0) i.e, whether $Z_{med} = Z_{min}$ (or) $Z_{med} = Z_{max}$

if Z_{med} is not equal to Z_{min} and Z_{max} & if it lies between Z_{min} & Z_{max}

i.e, $Z_{min} < Z_{med} < Z_{max}$, it means if Z_{med} is not a noise (salt & pepper) then we go to second stage ie stage 'B'. If $Z_{min} < Z_{med} < Z_{ma}$

Condition fails, increase the mask size (i.e, increase S_{xy} size) & if the increased mask size is less than the specified (or) maximum allowable mask size again repeat stage A. otherwise display Z_{med}

In stage B, we check each and every intensity value of mask S_{xy} i.e

Z_{xy} of S_{xy}

If $Z_{min} < Z_{xy} < Z_{max}$ display Z_{xy}

else

display Z_{med}

Periodic Noise Reduction by Frequency Domain Filtering:

Periodic noise appears as concentrated bursts of energy in the Fourier transform, at locations corresponding to the frequencies of periodic interference. The approach is to use a selective filter to isolate the noise.

The three types of selective filters (band reject, bandpass and notch) are used in for basic periodic noise reduction.

Selective Filters

- Band Pass and Band Reject
- Notch filters

A band pass filter is obtained by substracting bandrejection filter response from '1' i.e,

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

Ideal Bandrejection filter transfer function is given as

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

where D_0 = cut off frequency

W = Width of the band

D = Distance from centre of filter ie $D(u,v)$

Ideal BandPass filter transfer function is given as

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

Transfer function for Butterworth Band rejection filter given as

$$H(U,V) = \frac{1}{1+\left[\frac{DW}{D^2-D_0^2}\right]^{2n}}$$

n = filter order

Transfer function for Butterworth Band Pass filter is given as

$$H(U,V) = \frac{1}{1+\left[\frac{D^2-D_0^2}{DW}\right]^{2n}}$$

Transfer function for Gaussian Band rejection filter is given as

$$H(U,V) = 1 - e^{-\left[D^2-D_0^2/DW\right]^2}$$

Transfer function for Gaussian Band Pass filter is given as

$$H(U,V) = e^{-\left[D^2-D_0^2/DW\right]^2}$$

• Notch Filter:

A Notch filter rejects (or) passes frequencies. The transfer function for Notch Filter is given as

$$H_{NR}(U,V) = \prod_{K=1}^{Q} H_K(U,V)\, H_{-K}(U,V)$$

$H_K(U,V)$ = high pass filter output at (U_K, V_K)

$H_{-K}(U,V)$ = high pass filter output at $(-U_K, -V_K)$

For Ex:- Butterworth notch reject filter of order 'n' given as

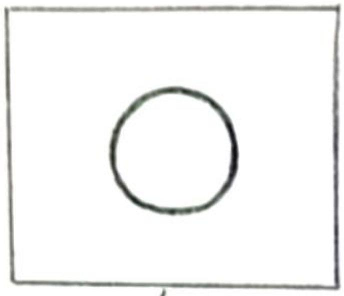
$$H_{NR}(U,V) = \prod_{K=1}^{Q} \left[\frac{1}{1+\left[D_{0K}/D_K(U,V)\right]^{2n}}\right]\left[\frac{1}{1+\left[D_{0K}/D_{-K}(U,V)\right]^{2n}}\right]$$

$$D_K(U,V) = \left[\left(u-\left(M/2+U_K\right)\right)^2 + \left(v-\left(N/2+V_K\right)\right)^2\right]^{1/2}$$

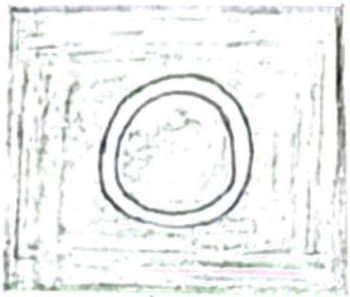
$$D_{-K}(U,V) = \left[\left(u-\left(m/2-U_K\right)\right)^2 + \left(v-\left(N/2-V_K\right)\right)^2\right]^{1/2}$$

Notch pass filter is obtained from a notch reject filter by using the expression

$$\boxed{H_{NP}(u,v) = 1 - H_{NR}(u,v)}$$



Gaussian band rejection Notch filter



Gaussian band pass Notch filter

Optimum Notch filtering :-

When several interference components are present, we use this filtering technique because, it won't remove the image details which is the advantage of this method over other filtering methods.

This filtering procedure consists of, first separating (isolating) the contributions of noise patterns & then substracting a weighted portion of pattern from the corrupted image.

This method can be used to remove multiple periodic interference noise patterns.

→ The first step in this process is, we extract the frequency components of noise patterns by using Notch pass filter ($H_{NP}(u,v)$) at the location of noise.

If this notch pass filter is designed to pass only noise components then fourier transform of this noise pattern is given as

$$N(u,v) = H_{NP}(u,v)\, G(u,v) \quad \rightarrow (1)$$

$G(U,V)$ = Fourier transform of corrupted image

$H_{NP}(U,V)$ = Frequency domain response of Notch pass filter

$N(U,V)$ = Noise patterns in frequency domain

Spatial domain response is obtained by taking inverse fourier transform of eq (1)

$$\Rightarrow \eta(x,y) = F^{-1}\left[H_{NP}(U,V)\, G(U,V)\right] \rightarrow (2)$$

Since corrupted image is formed by the addition of noise to original image ($f(x,y)$) i.e $g(x,y) = f(x,y) + \eta(x,y)$. Substracting noise from degraded image gives estimation of original image

i.e, $g(x,y) - \eta(x,y) = f(x,y) \rightarrow (3)$

$$f(x,y) = g(x,y) - \eta(x,y) \rightarrow (3a)$$

Instead of substracting $\eta(x,y)$ from $g(x,y)$, we substract weighted portion of $\eta(x,y)$ i.e $\omega(x,y) \times \eta(x,y)$ from $g(x,y)$ i.e,

$$\hat{f}(x,y) = g(x,y) - \omega(x,y)\eta(x,y)$$

$$\underbrace{\hat{f}}_{\text{estimate of } f(x,y)}(x,y) = g(x,y) - \omega(x,y)\,\eta(x,y) \rightarrow (4)$$

$\omega(x,y)$ = weighting (or) modulation function

We need to find 'ω' from eq (4)

Consider a neighbourhood of size $\underbrace{(2a+1)\text{by}(2b+1)}$ about a point (x,y)

Ex: $3\times3 = (2(1)+1)\text{by}(2(1)+1)$

$5\times5 = (2(2)+1)\text{by}(2(2)+1)$

The $\underbrace{\text{local variance}}_{\text{variance under mask}}$ of $\hat{f}(x,y)$ at (x,y) is given as

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left[\hat{f}(x+s, y+t) - \bar{\hat{f}}(x,y)\right]^2 \rightarrow (6)$$

$\bar{\hat{f}}(x,y)$ = average value of $\hat{f}$ in the neighbourhood.

(10)

$$\Rightarrow \hat{f}(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \hat{f}(x+s, y+t) \rightarrow (7)$$

Substitute (4) & (7) in (6), we get

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left\{ \left[g(x+s, y+t) - w(x+s, y+t)\eta(x+s, y+t) \right] - \left[\bar{g}(x,y) - \overline{w(x,y)\eta(x,y)} \right] \right\}^2 \rightarrow (8)$$

let $w(x,y)$ remains constant over a given neighbourhood, then

$$w(x+s, y+t) = w(x,y)$$

$$\Rightarrow \overline{w(x,y)\eta(x,y)} = w(x,y)\,\bar{\eta}(x,y) \rightarrow (9)$$

Substitute eq-(9) in (8) we get

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left\{ \left[g(x+s, y+t) - w(x,y)\eta(x+s, y+t) \right] - \bar{g}(x,y) - w(x,y)\bar{\eta}(x,y) \right] \right\}^2$$

to get minimum variance $\frac{\partial \sigma^2}{\partial w(x,y)} = 0$, by solving partial derivative, we get

$$w(x,y) = \frac{\overline{g(x,y)\eta(x,y)} - \bar{g}(x,y)\bar{\eta}(x,y)}{\overline{\eta^2}(x,y) - \bar{\eta}^2(x,y)} \rightarrow (10)$$

after obtaining $w(x,y)$ we calculate $\hat{f}(x,y)$ by using following expression

$$\hat{f}(x,y) = g(x,y) - w(x,y)\eta(x,y)$$

Therefore by using optimum notch filtering technique, we can estimate closest approximation to original image ie $\hat{f}(x,y)$ from degraded image.

11

~~[Notch Pass filter is obtained from a notch reject filter by using the expression~~

$$\cancel{H_{NP}(u,v) = 1 - H_{NR}(u,v)} \,]$$

Estimating the Degradation Function:-

There are 3 principal ways to estimate the degradation function

(1) Observation

(2) Experimentation

(3) Mathematical modeling.

The process of restoring an image by using a degradation function that has been estimated in some way something times is called blind deconvolution.

(1) Estimation by Image Observation:

Consider a degraded image without knowing the degradation function H. Based on assumption that image was degraded by a linear, position-invarient process, one way to estimate H is to gather information from image itself. For example, if image is blurred, we can look at small rectangular section of image containing sample structures To reduce noise effect we consider area with strong signal content. The next step would be to process subimage to get unblurred image.

For example, this can be done by sharpening subimage with a sharpening filter

If $g_s(x,y)$ is the subimage and $\hat{f}_s(x,y)$ be the subimage that is processed and assume effect of noise is negligible because of the choice of strong signal area, the function is given as

$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

(2) Estimation by Experimentation :-

Images similar to the degraded fun image can be acquired with various system settings until they are degraded as closely as possible to image we wish to restore. Then obtain impulse response of degradation by imaging an impulse using same system settings. An impulse is simulated by a bright dot of light, to reduce effect tof noise to negligible values. The fourier transform of an impulse is a constant

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

$G(u,v)$ is fourier transform of observed image

A is a constant describing strength of impulse.

(3) Estimation by Modeling :-

Degradation modeling has been used because of insight it affords in the image restoration problem. Major approach in modeling is to derive a mathematical model starting from basic principles.

The total exposure at any point of recording medium is obtained by integrating the instantaneous exposure over the time interval during which the imaging system shutter is open.

If T is the duration of exposure, it follows that

(2)

$$g(x,y) = \int_0^T f[x - x_0(t), y - y_0(t)]\,dt \rightarrow (1)$$

where $g(x,y)$ is the blurred image

Fourier transform of eq (1) is

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\, e^{-j2\pi(ux+vy)}\,dx\,dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\int_0^T f[x - x_0(t), y - y_0(t)\,dt]\right] e^{-j2\pi[ux+vy]}\,dx\,dy \rightarrow (2)$$

Reversing order of integration, eq (2) is expressed in the form

$$G(u,v) = \int_0^T \left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)]\, e^{-j2\pi(ux+vy)}\,dx\,dy\right] dt$$

The term inside the outer brackets is the fourier transform of the displaced function $f[x - x_0(t), y - y_0(t)]$.

$$G(u,v) = \int_0^T F(u,v)\, e^{-j2\pi[ux_0(t)+vy_0(t)]}\,dt$$

$$= F(u,v)\int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\,dt \rightarrow (3)$$

Where the last step follows from fact that $F(u,v)$ is independent of 't'.

By defining

$$H(u,v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\,dt \rightarrow (4)$$

Eq (3) can be expressed in the familiar form

$$G(u,v) = H(u,v)\,F(u,v)$$

If variables $x_0(t)$ and $y_0(t)$ are known, transfer function $H(u,v)$ can be obtained directly from eq (4). Suppose image undergoes uniform linear motion in x-direction, at rate given by $x_0(t) = at/T$. When $t = T$, the image displaced by a total distance 'a'. With $y(t) = 0$, eq (4) becomes

$$H(u,v) = \int_0^T e^{-j2\pi u x_0(t)}\,dt$$

$$H(u,v) = \int_0^T e^{-j2\pi ua/T}\, dt$$

$$= \frac{T}{\pi ua} \sin(\pi ua) e^{-j\pi ua}$$

H vanishes at values of 'u' given by $u = n/a$, n is an integer.

If y-component also vary with motion given by $y_0 = bt/T$, the degradation function becomes

$$\boxed{H(u,v) = \frac{T}{\pi(ua+vb)} \sin\left[\pi(ua+vb)\right] e^{-j\pi(ua+vb)}}$$

h

Inverse Filtering :-

The output of degraded image (or) corrupted image in frequency domain is given as

$$G(u,v) = H(u,v)\,F(u,v) + N(u,v) \longrightarrow (1)$$

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

$$F^{\wedge}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \longrightarrow (2) \quad \left[\because \hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}\right]$$

$\hat{F}(u,v)$ = Estimated output of degraded image

$H(u,v)$ = Degradation function

$N(u,v)$ = Noise in frequency domain.

From eq (2), we can observe one thing (ie)

If $H(u,v) =$ '0' (or) small then the fraction $\frac{N(u,v)}{H(u,v)}$ becomes high (or) infinity. So, for this reason we need to shift $H(u,v)$ to $H(0,0)$. At $H(0,0)$, the value is maximum (i.e) $H(0,0)$ is high, therefore the fraction $N(u,v)\ H(u,v)$ becomes very low, in estimated image $\hat{F}(u,v)$. the noise component will be less. So, we can get closest match to $F(u,v)$ ie $\underset{\text{original image in frequency domain}}{F(u,v)} \simeq \hat{F}(u,v)$

In eq (2) it is not possible to estimate $N(u,v)$ pratically.

Minimum - Mean Square Error Filtering (or) Weiner Filtering :-

let $f(x,y)$ represents original image

$\hat{f}(x,y)$ represents estimated image

$$e^2 = E\left\{(f-\hat{f})^2\right\}, \quad E\{\} \text{ indicates estimate (or) average}$$

Mean Square Error is given as square of the difference between $\hat{f}, f$

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

If the difference in eq (1) is high, then we can't get closest match to $f(x,y)$ i.e, $\hat{f}(x,y)$ may not be equal to $f(x,y)$. So, if we want to get estimated image (or) restored image similar to original image $f(x,y)$ then MSE should be minimum.

"The filters which uses (or) implements, the minimum mean square concept, that filter is known as 'Weiner filter' (or) minimum mean square error filter."

The estimated output of weiner filter in frequency domain is given as

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

multiplying and dividing eq (2) with $H(u,v)$, we get

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

$$= \left[\frac{1}{H(u,v)} \quad \frac{|H(u,v)|^2 S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v) \longrightarrow (3)$$

$$\left[\because HH^* = |H|^2\right]$$

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

By taking $S_f(u,v)$ as common term we get

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \quad \frac{|H(u,v)|^2}{|H(u,v)|^2 + \dfrac{S_\eta(u,v)}{S_f(u,v)}}\right] G(u,v) \longrightarrow (4)$$

From eq (4)

$H^*(u,v)$ = Complex Conjugate of $H(u,v)$

$H(u,v)$ = Degradation function

$S_\eta(u,v)$ = Noise Power Spectrum

$S_f(u,v)$ = Signal Power Spectrum i.e, undegraded image power spectrum.

$\hat{F}(u,v)$ = estimated (or) restored output.

If $\dfrac{S_\eta(u,v)}{S_f(u,v)} = K$, then eq (4) becomes

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \times \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v) \longrightarrow (5)$$

From eq (4) (or) (5), even if $H(u,v) = 0$ (or) low, the ratio $\dfrac{|H(u,v)|^2}{|H(u,v)|^2 + K}$ doesn't become high as it did in inverse filtering.

Similar to inverse filtering, it is not possible to estimate Noise power spectrum in eq(4). So, we use some constant 'K' shown in eq (5). If $K=0$, then eq (5) becomes

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \rightarrow \text{Inverse filtering}$$

(17)

Signal to Noise ratio is given as

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} |F(u,v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} |N(u,v)|^2}$$

$|F(u,v)|^2$ = Signal Power

$|N(u,v)|^2$ = Noise Power

Signal to Noise ratio is defined as ratio of signal power to noise power. Images with low noise tend to have a high SNR

(or)

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

Geometric Mean Filter:-

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2}\right]^{\alpha} \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \beta\left[\frac{S_\eta(u,v)}{S_f(u,v)}\right]}\right]^{1-\alpha} G(u,v) \longrightarrow (1)$$

α, β are positive real constants

If $\alpha = 1$ then eq (1) becomes as

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

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

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

If $\alpha = 1$, then eq (1) becomes as a inverse filter

If $\alpha = 0$, then eq (1) becomes as a weiner filter shown in eq (3)

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \beta \frac{S_\eta(u,v)}{S_f(u,v)}} \right] G(u,v) \longrightarrow (3)$$

If $\beta = 1$, $\alpha = 1/2$ then eq (1) becomes product of two terms with same power, thats why the name Geometric mean filters.

Constrained Least Squares Filtering:-

In weiner filter i.e.

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

We can't estimate power spectra of undergraded image and noise. ie $\frac{S_\eta(u,v)}{S_f(u,v)}$ ratio. We can achieve good results (o/p) for weiner filter if we assume 'K' in the place of $\frac{S_\eta(u,v)}{S_f(u,v)}$

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

However estimation of constant is not always a suitable solution.

$\Rightarrow$ In constrained least square restoration process, we require information about mean and variance of noise only; this is the advantage of constrained least square restoration over weiner filter.

From convolution (i.e, linear position Invarient degradation model) we know that

$$\boxed{g = Hf + \eta} \longrightarrow (1)$$

g = degraded image

H = degradation function

f = original Image

η = noise

(14)

Since 'H' is very sensitive to Noise, we use second order derivative response to measure this noise i.e, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ operator (or) Laplacian operator.

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

We have to find 'c' (criteria) by using Laplacian operator & we should minimize 'c' subjected to constraint.

$$\boxed{\|g - H\hat{f}\|^2 = \|\eta\|^2} \longrightarrow (3)$$

$$\therefore \|W\| = W^T W = \sum_{K=1}^{n} w_K^2 \longrightarrow \text{Eucleadian Norm form}$$

The frequency domain solution (or) representation of $\hat{f}$ is given as

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

'γ' is a constant

$H(U,V)$ = degradation function in frequency domain

$P(U,V)$ = frequency domain response of laplacian mask $P(x,y)$

$$P(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & 0 & 0 \end{bmatrix} = \text{Laplacian mask}$$

$G(U,V)$ = Degraded Image in frequency domain

In constrained least square restoration process, we estimate 'γ' i.e. we select some constant value 'γ' & then we estimate $\hat{F}(U,V)$ by using inverse fourier transform we get $\hat{f}(x,y)$, from the obtained $\hat{f}(x,y)$ we calculate

$g - H\hat{f}$, so that it should satisify the equation

$$\|g - H\hat{f}\|^2 = \|\eta\|^2$$

i.e, by selecting 'γ' manually we calculate the noise $\|\eta\|$

=> 'γ' can be estimated interactively, until desired results can be obtained interactive process includes the following procedure.

(i) Select $r = g - H\hat{f}$

Since $\hat{f}$ is a function of 'γ' => $\hat{f}$ will also be a function of 'γ'

=> r will also be a function of 'γ'

So, => $\phi(\gamma) = r^T \cdot r = \|r\|^2$

we have to adjust 'γ' until it satisifies $\|r\|^2 = \|\eta\|^2 + a$ equation, where 'a' is accuracy factor.

=> The following steps describes how to select 'γ'.

Steps :-

1. Select a value for 'γ'.
2. Calculate $\|r\|^2$.
3. Stop if $\|r\|^2 = \|\eta\|^2 + a$ satisified

   otherwise

   go to step 2 after increasing 'γ'.

   if $\|r\|^2 < \|\eta\|^2 - a$

   go to step 2 after decreasing 'γ' if $\|r\|^2 > \|\eta\|^2 + a$
4. Use the new value of 'γ' to calculate $\|r\|^2$

* Newton - Raphson algorithm:-

→ To use this algorithm, we need $\|r\|^2$ & $\|\eta\|^2$

Since we know $r = g - H\hat{f}$

In frequency domain we get

(16)

$$R(u,v) = G(u,v) - H(u,v)\hat{F}(u,v) \longrightarrow (1)$$

If we calculate inverse fourier transform of eq(1), we get 'r'

$$\Rightarrow \underbrace{\|r\|^2}_{\text{Eucleadian Norm form}} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} r^2(x,y) \qquad \text{since } \|w\| = w^T w = \sum_{k=1}^{n} w_k^2$$

M, N are rows and coloumns of given image

Compute $\|\eta\|^2$ by using laplacian operator.

→ Consider the variance of noise over entire image i.e

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

$$\left[\because \sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2\right.$$

z_i ↓ i/p image intensity values

$\bar{z}$ = mean of i/p image

$$m_n = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\eta(x,y)] \longrightarrow (3)$$

from (2) and (3), if we consider zero mean ($m_\eta = 0$) noisy image we get

$$MN\sigma^2 = MN \cdot \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\eta(x,y) - m_\eta]^2$$

$$= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\eta(x,y)]^2$$

$$MN\sigma^2 = \|\eta\|^2 \qquad \because \|r\|^2 = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} r^2(x,y)$$

$$\Rightarrow \|\eta\|^2 = MN\sigma^2 \longrightarrow (4)$$

If we consider mean of noise too in eq (4) we get

$$\|\eta\|^2 = MN[\sigma^2 + m_\eta] \longrightarrow (5)$$

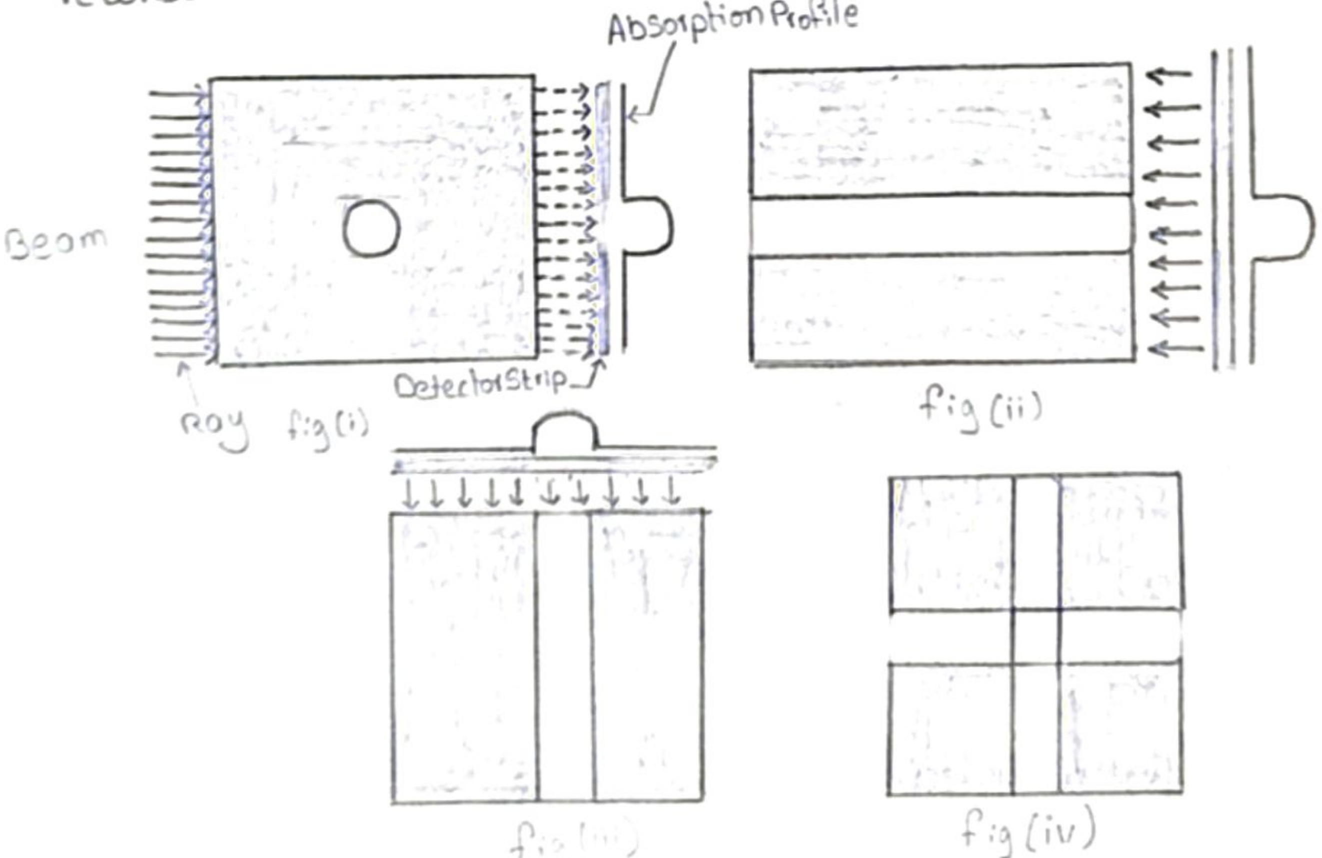
Equation (5) tells us that we can implement optimum (constrained least squares) algorithm by having knowledge of only mean and variance of the noise.

Image Reconstruction from Projection:-

Assume a object on a uniform background as shown in fig (i), pass a thin beam of x-rays from left to right. Object absorbs some of x-ray energy. by using detector strip (x-ray absorbtion detectors) on the other side of the object, the detector yields absorption profile as shown in fig (ii) which is a 1-D absorption signal. If we have more than one object in the x-ray path, i.e, in 1-D projection we can't detect them. So, we back project the x-ray beam. The process of back projecting X-ray 1-D signal across 2D Signal is known a Smearing (i.e duplicating 1-D Signal across 2-D area) shown in fig (iii).

The above described process is known as 'back-Projection'.

We continue the projections at different angles and reconstruct the image as shown in fig (iv).



First Generation (G1) CT scanners :-

First - generation CT scanners employ a "Pencil" X-ray beam and a single detector as shown in fig (a). For a given angle of rotation, the detector pair is translated incrementally along the linear direction and projection is generated by measuring the output of detector at each increment of translation. After complete linear translation, the detector is rotated & procedure is repeated to generate projections for different angles in range [0°, 180°] and generate complete set of projections. The image hence is generated by back projection.

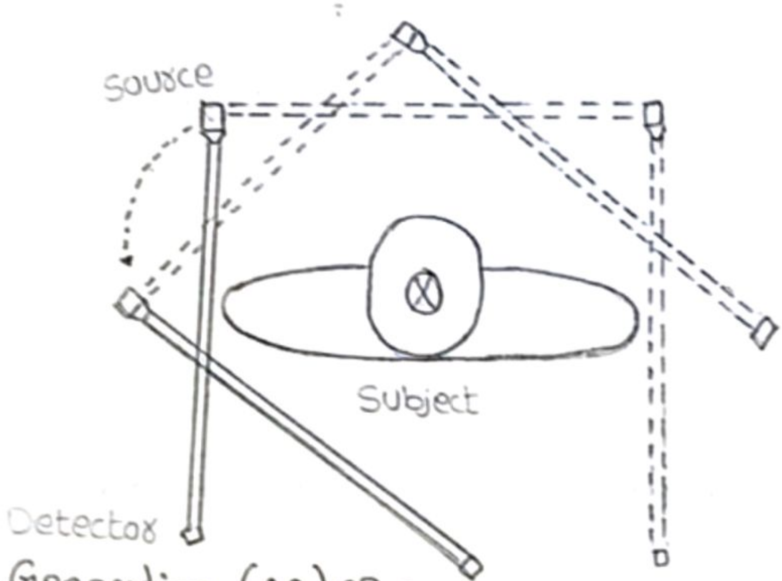


fig (a):
Ist Generation
CT scanners

Second - Generation (G2) CT Scanners :-

Second generation CT scanners operate on the same principle as G1 scanners. Only difference is that the beam is in the shape of a fan. This uses multiple detectors. Hence number of translations are less than that of in G1 scanners. The second generation scanner is as shown in fig (b).

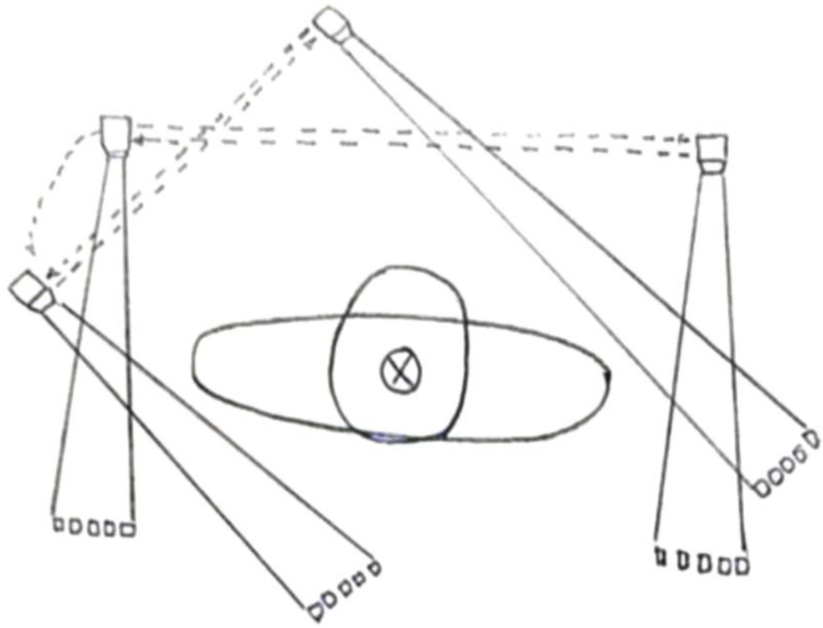


fig (b): IInd Generation CT Scanners

Third - Generation (G3) Scanners :-

Third - generation scanners are a significant improvement over the earlier two generations. These scanners use a bank of detectors long enough (order of 1000 individual detectors) to cover the entire field of view of a wider beam. Each increment in angle, produces entire projection thus eliminate the need to translate detector pair as done in G1 and G2 scanners.
The third - generation scanner is as shown in fig (c)

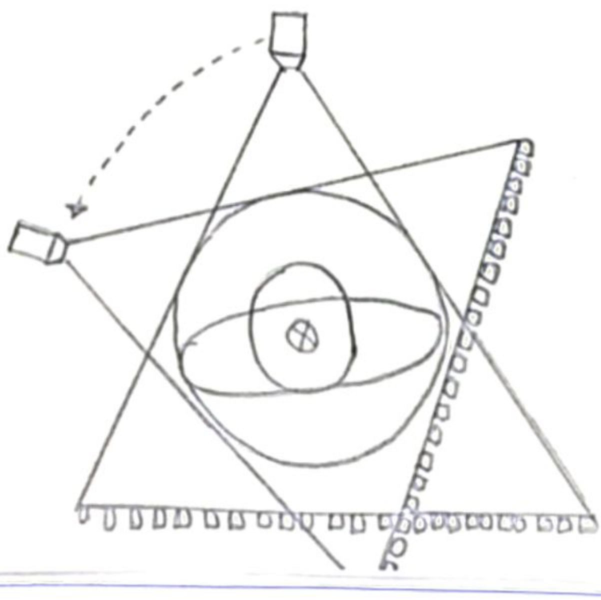


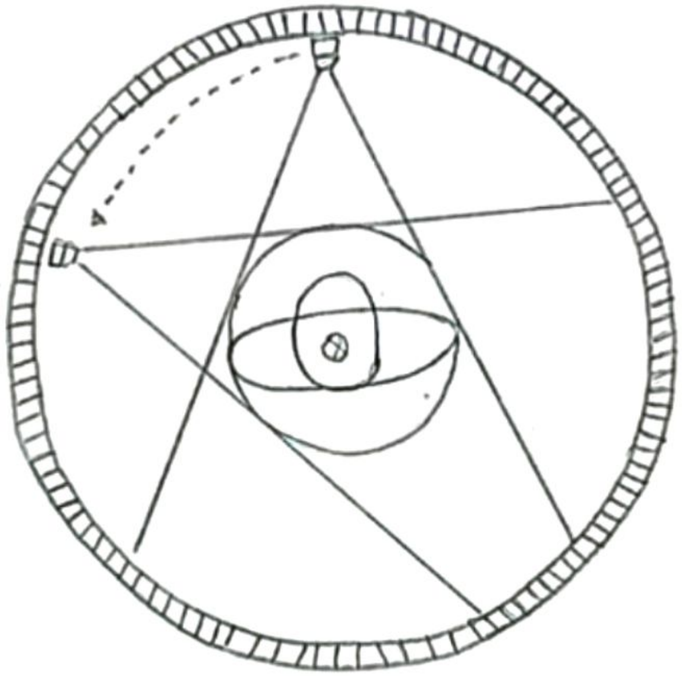
fig (c): IIIrd Generation CT Scanners

Fourth-Generation (G4) Scanners :-

Fourth-generation scanners uses a circular ring of detectors (order of 5000 individual detectors) and only the source has to rotate.

Advantage of G3 and G4 scanners is their speed.

Disadvantages are cost and greater X-ray scatter that requires high doses than G1 & G2 to achieve comparable Signal to Noise characteristics.

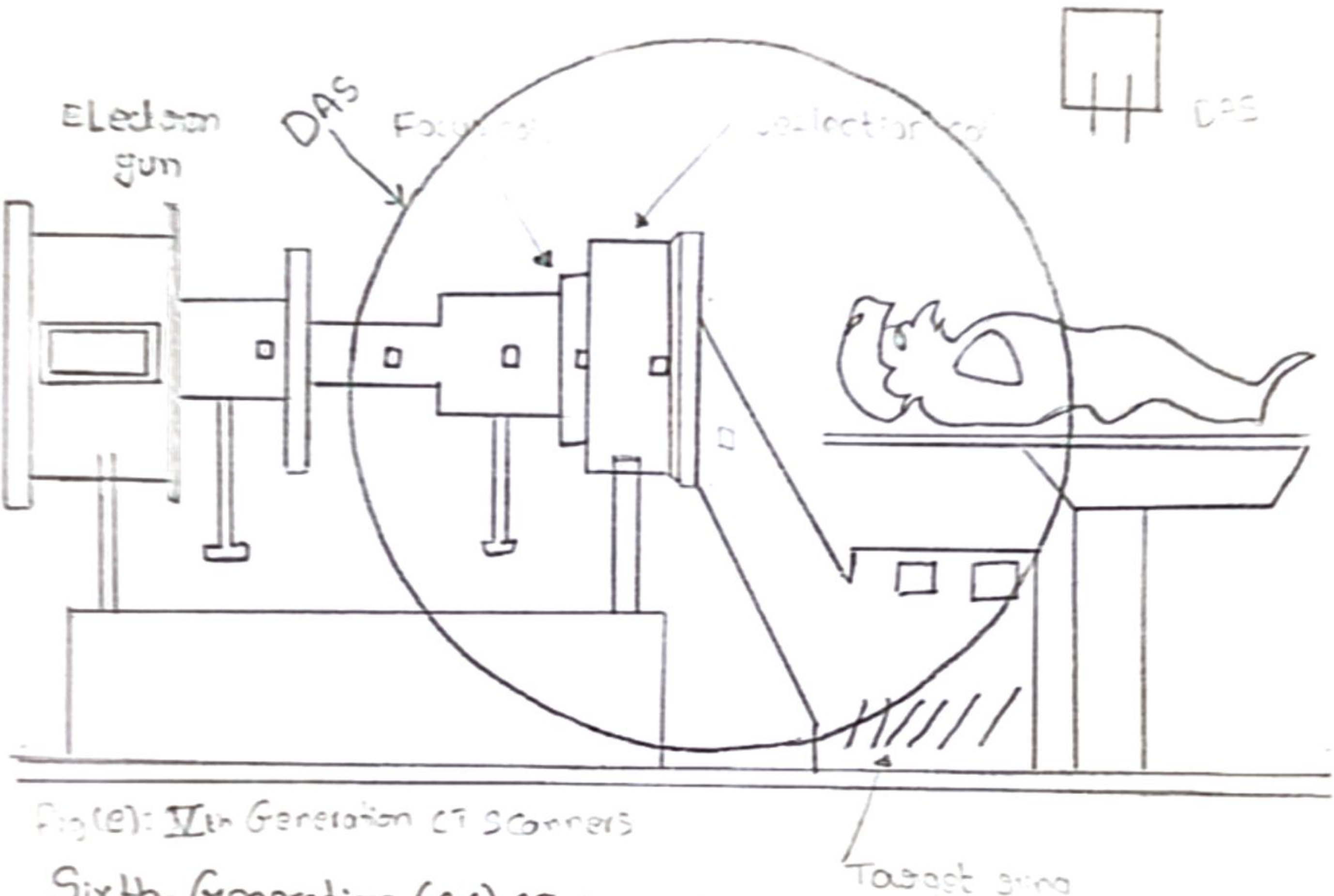


fig(d): IVth Generation CT scanners

Fifth-Generation (G5) CT scanners :-

Fifth-generation (G5) scanners are also known as 'electron beam computed Tomography (EBCT)' scanners. These eliminate all mechanical motion by employing electron beam controlled electromagnetically. By striking tungsten anodes that encircle the patient, these beams generate X-rays that are then shaped into fanbeam that passes through the patient and excites a ring of detectors as in G4 scanners.

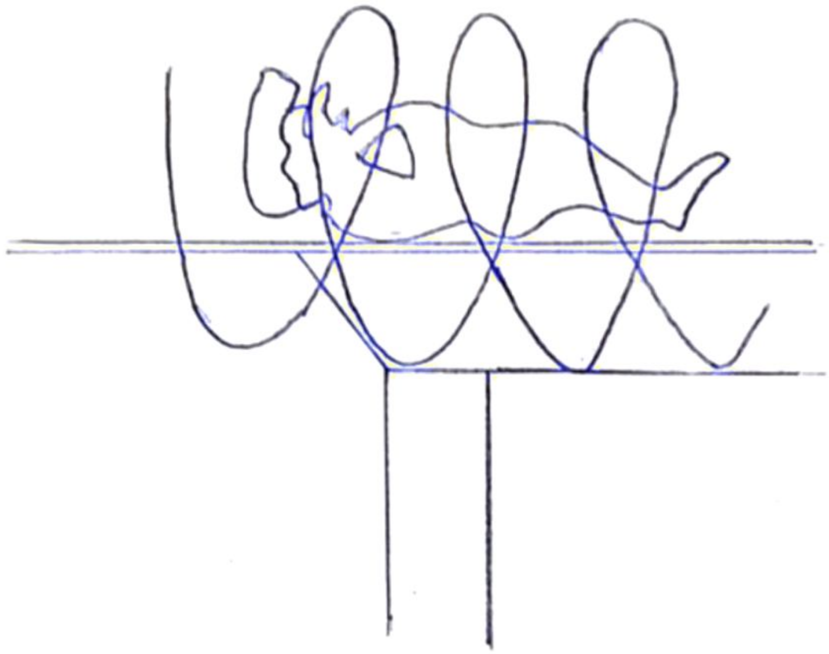
Although an image may be obtained in less than one second, there are procedures that require several minutes which is drawback.



Fig(e): IVth Generation CT Scanners

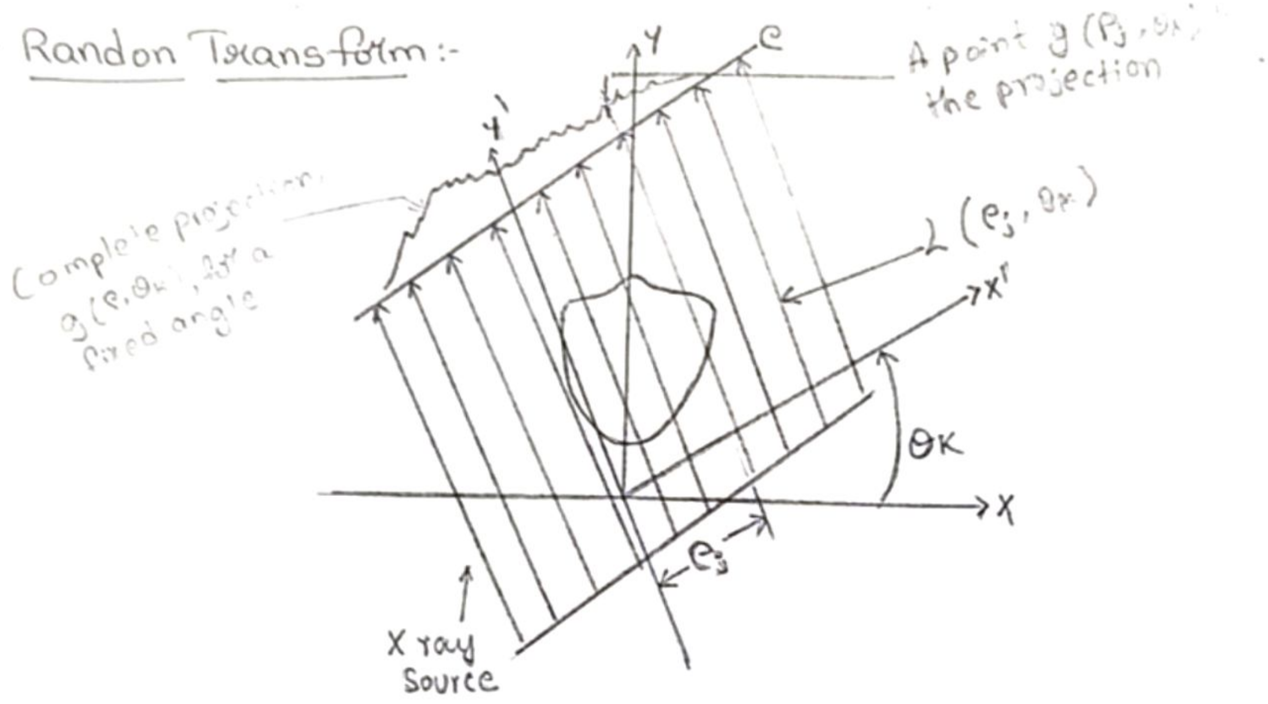
Sixth-Generation (G6) CT Scanners:-

In Sixth-generation (G6) scanners, G3 (or) G4 Scanner is configured using slip rings that eliminate the need for electrical and signal cabling between the detectors and processing unit. The detector pair thus rotates continuously through 360° and the patient is moved at constant Speed along axis perpendicular to the scan. The result is a continuous helical data that is processed to obtain individual slice images.



fig(f): $\overline{\text{VI}}$th generation CT scanners

+ Randon Transform :-



The projection of a parallel-ray beam is a set of lines as shown in above figure.

Distance from x axis to specified point is 'ρ_j'

An arbitrary point in projection signal is given by raysum along line $x\cos\theta_K + y\sin\theta_K = \rho_j$

where θ_K is the angle of inclination at a point.

At particular time, the Captured intensity $g(\rho_j, \theta_K)$ is given as

$$g(\rho_j, \theta_K) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, \delta(x\cos\theta_K + y\sin\theta_K - \rho_j)\, dx\, dy \longrightarrow (1)$$

$$R(f(x,y)) = g(\rho, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, \delta(x\cos\theta + y\sin\theta - \rho)\, dx\, dy \longrightarrow (2)$$

[∵ by considering all values of ρ and θ]

Thus eq (2) is the required randon transform (or) line integral transform.

$g(\rho,\theta)$ hence can be represented as $R\{f(x,y)\}$ (or) $R\{f\}$.

- In discrete cases, eq (2) becomes

$$g(\rho,\theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, \delta(x\cos\theta + y\sin\theta - \rho)$$

where x, y, ρ and θ are discrete variables.

Fourier Slice Theorem :-

This derives fundamental result relating 1-D fourier transform of projection and 2-D fourier transform of region from which projection was obtained.

1-D fourier transform of a projection with respect to ρ is

$$G(\omega,\theta) = \int_{-\infty}^{\infty} g(\rho,\theta)\, e^{-j2\pi\omega\rho}\, d\rho \longrightarrow (1)$$

We have by Randon transform

$$R\{f(x,y)\} = g(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, \delta(x\cos\theta + y\sin\theta - \rho)\, dx\, dy \longrightarrow (2)$$

Substitute eq (2) in eq (1)

$$eq(1) \Rightarrow G(\omega,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) \left[\int_{-\infty}^{\infty} \delta(x\cos\theta + y\sin\theta - \rho)\, d\rho\right] e^{-j2\pi\omega\rho}\, dx\, dy \longrightarrow (3)$$

from eq (3), we know

$$\delta(x) = 1 \quad \text{at } x = 0$$

$$\delta(x\cos\theta + y\sin\theta - \rho) = 1 \text{ at}$$

$$x\cos\theta + y\sin\theta - \rho = 0$$

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

By substituting 'ρ' value in eq (3), we get

$$G(\omega,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi\omega(x\cos\theta + y\sin\theta)}\, dx\, dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi[x(\omega\cos\theta) + y(\omega\sin\theta)]}\, dx\, dy$$

$$= F(u,v)$$
$$u=\omega\cos\theta,\ v=\omega\sin\theta$$

$$\boxed{\therefore\ G(\omega,\theta) = F(\omega\cos\theta,\ \omega\sin\theta)}$$

where $F(u,v)$ is the 2-D fourier transform of $f(x,y)$

Conclusion:-

The fourier theorem thus states that the 1D fourier transform of a projection of given image is a slice of two dimensional fourier transform of given image.

Image Reconstruction by using Parallel beam filter back Projection:-

2-D inverse fourier transform is given by

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}\,du\,dv \quad \text{—} r(1)$$

Let us assume

$$u = \omega\cos\theta$$
$$v = \omega\sin\theta$$

To get dudv value, we use Jacobian Matrix form.

By Jacobain matrix we can calulate partial derivative and it is given as

if $x = r\cos\theta$

$y = r\sin\phi$

$z = r\sin\beta$

$$\frac{\partial x}{\partial\theta}\frac{\partial y}{\partial\phi}\frac{\partial z}{\partial\beta} = \begin{vmatrix} \frac{\partial x}{\partial\theta} & \frac{\partial x}{\partial\phi} & \frac{\partial x}{\partial\beta} \\ \frac{\partial y}{\partial\theta} & \frac{\partial y}{\partial\phi} & \frac{\partial y}{\partial\beta} \\ \frac{\partial z}{\partial\theta} & \frac{\partial z}{\partial\phi} & \frac{\partial z}{\partial\beta} \end{vmatrix}$$

if only x and y are considered

$$\frac{\partial x}{\partial\theta}\frac{\partial y}{\partial\phi} = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial\theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial\phi} \end{vmatrix}$$

for $u = \omega\cos\theta$
$v = \omega\sin\theta$

$$\frac{\partial u}{\partial \theta}\frac{\partial v}{\partial \omega} = \begin{vmatrix} \frac{\partial u}{\partial \omega} & \frac{\partial u}{\partial \theta} \\ \frac{\partial v}{\partial \omega} & \frac{\partial v}{\partial \theta} \end{vmatrix}$$

$$\frac{\partial u}{\partial \omega} = \cos\theta \qquad \frac{\partial u}{\partial \theta} = -\omega\sin\theta$$

$$\frac{\partial v}{\partial \omega} = \sin\theta \qquad \frac{\partial v}{\partial \theta} = \omega\cos\theta$$

$$\Rightarrow \frac{\partial u}{\partial \theta}\frac{\partial v}{\partial \omega} = \begin{vmatrix} \cos\theta & -\omega\sin\theta \\ \sin\theta & \omega\cos\theta \end{vmatrix}$$

We know the determinant of 2x2 matrix is given as

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Hence for above matrix

$$\frac{\partial u}{\partial \omega}\frac{\partial v}{\partial \theta} = \begin{vmatrix} \cos\theta & -\omega\sin\theta \\ \sin\theta & \omega\cos\theta \end{vmatrix} = \omega\cos^2\theta + \omega\sin^2\theta$$
$$= \omega$$

$$du\,dv = \omega\, d\omega\, d\theta$$

Hence eq (1) becomes as

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\omega\cos\theta, \omega\sin\theta)\, e^{j2\pi(x\omega\cos\theta + y\omega\sin\theta)}\, |\omega|\, d\omega\, d\theta \rightarrow (2)$$

From fourier slice theorem

$$G(\omega,\theta) = F(\omega\cos\theta, \omega\sin\theta)$$

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(\omega,\theta)\, e^{j2\pi(x\omega\cos\theta + y\omega\sin\theta)}\, |\omega|\, d\omega\, d\theta \rightarrow (3)$$

$$f(x,y) = \frac{1}{2}\int_{0}^{2\pi}|\omega|\int_{-\infty}^{\infty} G(\omega,\theta)\, e^{j2\pi(x\omega\cos\theta + y\omega\sin\theta)}\, d\omega\, d\theta \longrightarrow (4)$$

[$\because$ θ spans from 0 to π and to get complete image we do smearing]

By frequency domain enhancement process we have

$$f(x,y) = F^{-1}\left[G(u,v)\,H(u,v)\right] \quad \rightarrow (5)$$

Compare eq (4) and eq (5)

$$G(u,v) = G(\omega,\theta)$$
$$H(u,v) = |\omega|$$

Which is a ramp signal and it has infinite response. This is the drawback as our image is a finite one. The way of limiting the range is by using windowing technique. By using windowing technique, we get finite response by multiplying ~~f(x,y) with~~ $G(\omega,\theta)$ with $|\omega|$ and hence can obtain original image.

Windowing function given as

$$h(c) = \begin{cases} c + (c-1)\cos\dfrac{2\pi\omega}{M-1} & 0 \le \omega \le (M-1) \\ 0 & \text{otherwise} \end{cases}$$

where c is a constant

if $c = 0.54$, function is Hamming window

$c = 0.5$, function is Hann window

Thus

$$f(x,y) = \int_0^{\pi}\int_{-\infty}^{\infty} G(\omega,\theta)\,|\omega|\,d\omega\, e^{j2\pi\omega\theta\rho}\,d\theta$$

$$\boxed{f(x,y) = \int_0^{\pi} g(\rho,\theta) * s(\rho)\,d\theta} \qquad \left[\because f(t)*h(t) = \int_{-\infty}^{\infty} f(\tau)\,h(t-\tau)\,d\tau\right]$$

From above ~~d~~expression, we can say that original image $f(x,y)$ can be obtained by performing convolution between $g(\rho,\theta)$ and $s(\rho)$

$$s(\rho) = \int_{-\infty}^{\infty} |\omega|\, e^{j2\pi\omega\rho}\,d\omega$$

Image Reconstruction Using Fan Beam Filtered Back Projections:

For this we use fan beam rays i.e, III generation (G3) Scanners.

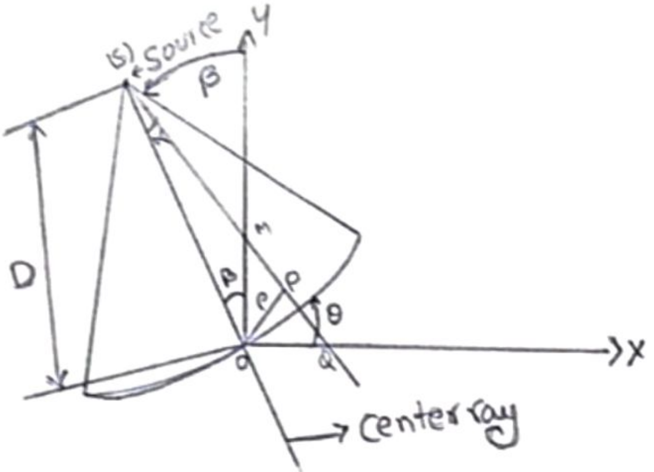
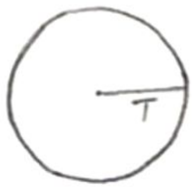


fig (1)



'α' is the angular displacement from center ray to any one of sensors.

'β' is the angular displacement of source with respect to Y-axis

'ρ' is the perpendicular distance

'θ' is the inclination with respect to X axis.

$$\alpha + (\beta + x) = 90^\circ$$

$$\theta + x = 90^\circ$$

$$\alpha + \beta + 90^\circ - \theta = 90$$

$$\boxed{\alpha + \beta = \theta}$$

from Δ^{le} OSP, $\sin\alpha = \frac{\rho}{D}$

$$\boxed{\rho = D\sin\alpha}$$

Assume object is of circular shape with radius T

Image in Spatial domain is given as

$$f(x,y) = g(\rho,\theta) * S(\rho)$$

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\rho,\theta)\, \delta(x\cos\theta + y\sin\theta - \rho)\, d\rho\, d\theta \rightarrow (1)$$

let $x = r\cos\psi$ $\qquad \rho = D\sin\alpha$

$y = r\sin\psi$ $\qquad d\rho = D\cos\alpha\, d\alpha$

$$\theta = \alpha + \beta$$

$$d\theta = d\beta \quad [\because \alpha \text{ is constant}]$$

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

$$r\cos\psi\cos\theta + r\sin\theta\sin\psi = \rho$$

$$r\cos(\theta-\psi) = \rho$$

$$r\cos(\theta-\psi) - D\sin\alpha = 0$$

Substitute above values in eq (1)

$$f(r,\psi) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(D\sin\alpha, \alpha+\beta)\, \delta\big(r\cos(\alpha+\beta-\psi) - D\sin\alpha\big)\, D\cos\alpha\, d\alpha\, d\beta \rightarrow (2)$$

Now we need to calculate $\delta(R\sin\alpha)$ and $\big(r\cos(\alpha+\beta-\psi) - D\sin\alpha\big)$ values.

we know that $\delta(\rho) = \int_{-\infty}^{\infty} \omega e^{j2\pi\omega\rho}\, d\omega \rightarrow (3)$

$$\delta(R\sin\alpha) = \int_{-\infty}^{\infty} \omega e^{j2\pi\omega(R\sin\alpha)}\, d\omega$$

let $\omega' = \frac{\omega R\sin\alpha}{\alpha} \Rightarrow \omega = \frac{\omega'\alpha}{R\sin\alpha}$

$d\omega' = \frac{R\sin\alpha}{\alpha} d\omega$, $\quad d\omega = \frac{\alpha}{R\sin\alpha} d\omega'$

$$\delta(R\sin\alpha) = \int_{-\infty}^{\infty} \omega'\left[\frac{\alpha}{R\sin\alpha}\right] e^{j2\pi\omega'\left(\frac{\alpha}{R\sin\alpha}\right)R\sin\alpha}\, d\omega' \left(\frac{\alpha}{R\sin\alpha}\right)$$

$$= \left[\frac{\alpha}{(R\sin\alpha)}\right]^2 \int_{-\infty}^{\infty} \omega' e^{j2\pi\alpha\omega'}\, d\omega'$$

$$= \left(\frac{\alpha}{R\sin\alpha}\right)^2 \delta(\alpha)$$

let $h(\alpha) = \delta(R\sin\alpha)$

$h(\alpha' - \alpha) = \delta(R\sin(\alpha' - \alpha))$

$R\sin(\alpha'-\alpha)$

$$\therefore \boxed{r\cos(\alpha+\beta-\psi) - D\sin\alpha = R\sin(\alpha'-\alpha)}$$

we have

$$\theta = 0 \text{ to } 2\pi$$

$$\alpha+\beta = 0 \text{ to } 2\pi$$

if $2\pi = \alpha+\beta$, if $\alpha+\beta = 0$

$\beta = 2\pi - \alpha$ $\quad\quad$ $\beta = -\alpha$

also $\rho = T$

$$D\sin\alpha = T$$

$$\alpha = \sin^{-1}(T/D)$$

Thus

$$f(r,\psi) = \frac{1}{2}\int_{-\alpha}^{2\pi-\alpha}\int_{-\sin^{-1}(T/D)}^{\sin^{-1}(T/D)} p(\alpha,\beta)\, s(R\sin(\alpha'-\alpha))\, D\cos\alpha \, d\alpha \, d\beta$$

The response along particular line is same for either fan beam (or) parallel beam

$$\Rightarrow g(\rho,\theta) = p(\alpha,\beta)$$

we know $s(R\sin\alpha) = \left(\frac{\alpha}{R\sin\alpha}\right)^2 s(\alpha)$

also $s(\rho) = \int_{-\infty}^{\infty} \omega e^{j2\pi\omega\rho} d\omega$

$$h(\alpha) = s(R\sin\alpha) = \left(\frac{\alpha}{\sin\alpha}\right)^2 s(\alpha)$$

$$h(\alpha'-\alpha) = s(R\sin(\alpha'-\alpha)) = \left[\frac{\alpha'-\alpha}{R\sin(\alpha'-\alpha)}\right]^2 s(\alpha'-\alpha)$$

Thus

$$f(r,\psi) = \frac{1}{2}\int_{0}^{2\pi}\int_{-\alpha}^{\alpha} p(\alpha,\beta)\, h(\alpha'-\alpha)\, D\cos\alpha \, d\alpha \, d\beta$$

8

(23)

Now to find $\overline{IV}$ $r\cos(\alpha+\beta-\psi)-D\sin\alpha$ value:

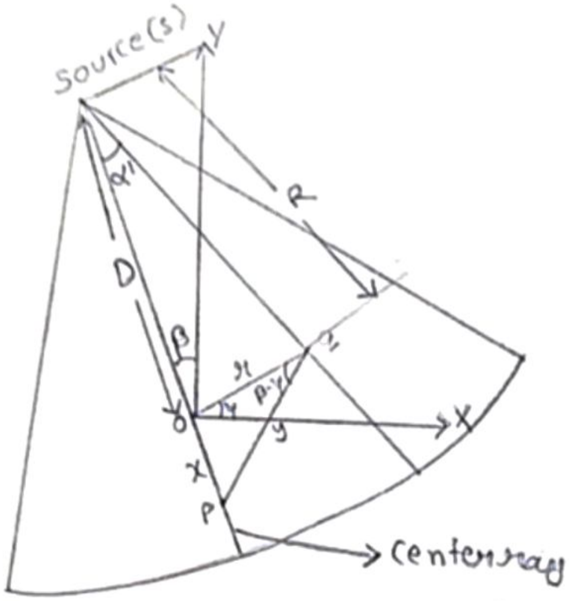


fig (2):

α' is the angle between center ray to specified point.

r is the perpendicular distance from origin.

from Δ^{le} opq, $\sin(\beta-\psi) = \frac{x}{r}$

$$x = r\sin(\beta-\psi)$$

$$\cos(\beta-\psi) = \frac{y}{r}$$

$$y = r\cos(\beta-\psi)$$

from Δ^{le} SPq, $\sin\alpha' = \frac{r\cos(\beta-\psi)}{R}$

$$\Rightarrow R\sin\alpha' = r\cos(\beta-\psi)$$

$$\cos\alpha' = \frac{D+r\sin(\beta-\psi)}{R}$$

$$\Rightarrow R\cos\alpha' = D+r\sin(\beta-\psi)$$

$$r\left[\cos\alpha\cos(\beta-\psi)\right]-\sin\alpha\left[r\sin(\beta-\psi)+D\right]\Rightarrow$$

$$\Rightarrow \cos\alpha.R\sin\alpha'-\sin\alpha\left[R\cos\alpha'-D+D\right]$$

$$R\left[\cos\alpha\sin\alpha'-\sin\alpha\cos\alpha'\right]$$

Sampling theorem:

The Fourier transform provides additional insight into the sampling process. "Sampling" is a process of converting a continuous function into a discrete signal. To achieve this, the signal is convolved with a continuous train of the impulse function.

Time domain statement:

A band limited signal having no frequency components higher than 'f_m' Hz may be completely recovered from the knowledge of its samples taken at the rate of atleast '$2f_m$' samples per second.

$f_s \geq 2f_m$ (or) $\omega_s \geq 2\omega_m$.

Frequency domain statement:

A band limited signal having no frequency components higher than 'f_m' Hz is completely ~~recovered~~ described by its samples at uniform intervals less than (or) equal to '$1/2f_m$' seconds apart $T_s \leq 1/2f_m$

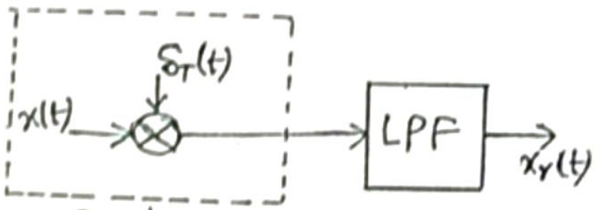


Fig 1. Sampling process Reconstruction process

The uniform sampling and reconstruction process is illustrated in fig 1.

Let us consider a band limited signal $x(t)$ having no frequency components beyond f_m Hz i.e., $X(\omega)$ is zero for $|\omega| > \omega_m$,

where $\omega_m = 2\pi f_m$

When this signal is multiplied by a periodic impulse train $\delta_T(t)$ (with period 'T'). The product yields a sequence of impulses located at uniform intervals of T seconds.

The strength of resulting impulses is equal to the value of $x(t)$ at the corresponding instants

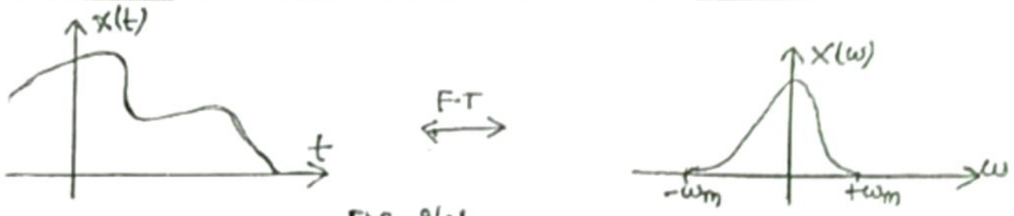


Fig. 2(a)

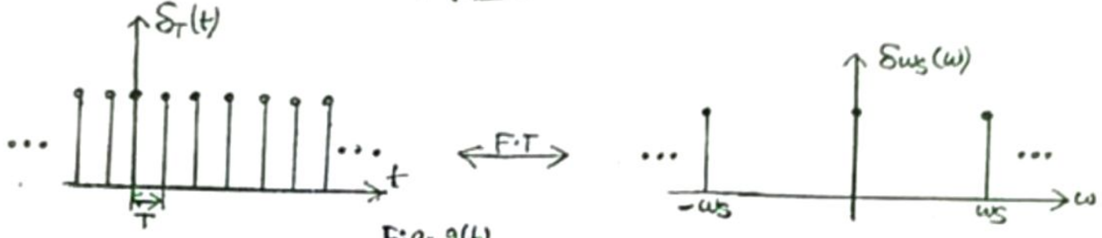


Fig. 2(b)

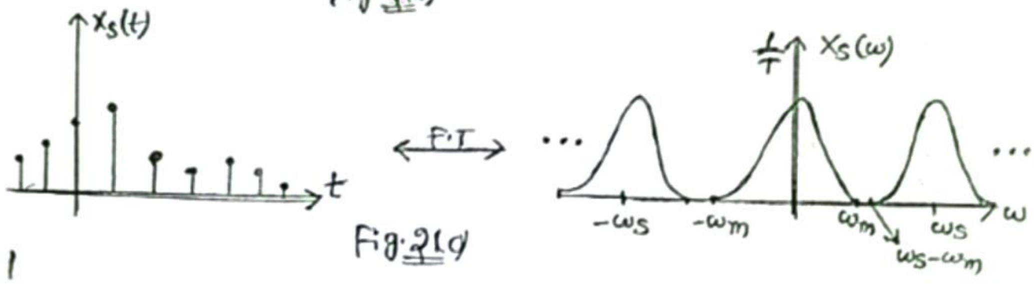


Fig. 2(c)

The Fourier transforms of the bandlimited signal $x(t)$ and the impulse train $\delta_T(t)$ are $X(\omega)$ and $\delta_{\omega s}(\omega)$ as shown in fig. 2(a) + (b) respectively.

The product of $x(t)$ + $\delta_T(t)$ yields a discrete time signal $x_s(t)$ as shown in fig. 2(c). The corresponding spectrum $X_s(\omega)$ can be determined by "frequency convolution" theorem as below.

We know that the fourier transform of periodic impulse train is also periodic function $\delta_{\omega s}(\omega)$ and can be written as the sum of impulses located at $\omega = 0, \pm\omega_s, \pm 2\omega_s, \ldots$

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \xleftrightarrow{F.T} \omega_s \delta_{\omega s}(\omega) = \omega_s \sum_{m=-\infty}^{\infty} \delta(\omega - m\omega_s)$$

Then, the sampled signal is given by

$$x_s(t) = x(t)\,\delta_T(t) \xleftrightarrow{F.T} X_s(\omega) = \frac{1}{2\pi}\left[X(\omega) * \omega_s \delta_{\omega s}(\omega)\right]$$

$$= \sum_{n=-\infty}^{\infty} x_n \delta(t - nT)$$

$$= \frac{\omega_s}{2\pi}\left[X(\omega) * \delta_{\omega s}(\omega)\right]$$

$$= \frac{1}{T}\left[X(\omega) * \delta_{\omega s}(\omega)\right]$$

Unit - 8

The process of subdividing an image into regions (or) objects is known as Image Segmentation. The segmentation process will stop when objects (or) region of interest have been detected in an application.

Example :- In the automatic inspection of electronic assemblies, we can analyse the images to detect broken connection paths and missing components.

Image segmentation can be done based on similarity and discontinuity of intensity values.

In discontinuity based segmentation, we segment the image based on change intensities (Ex:- Edges)

In similarity based segmentation, we segment the image i.e partitioning the image based on predefined criteria like thresholding, region growing, region splitting and merging.

Segmentation

- discontinuity of intensity values
  - Points
  - Lines
  - Edges
- Similarity of intensity values
  - Thresholding
  - Regiongrowing
  - Region splitting
  - Region merging

Fundamentals:

Let 'R' represents the entire spatial region occupied by an image partitioning the image Region 'R' into n Subregions, $R_1, R_2 \ldots\ldots R_n$ such that

R

R_1	R_2	R_3
R_4	R_5	R_6
R_7		R_n

(i) $R = \bigcup_{i=1}^{n} R_i$

(ii) R_i is a connected set $i = 1, 2 \ldots n$

(iii) $R_i \cap R_j = \phi$ for all i, j $i \neq j$

(iv) $Q(R_i) = TRUE$ $i = 1, 2 \ldots n$

(v) $Q(R_i \cup R_j) = FALSE$ for adjacent regions R_i and R_j

Condition (i) indicates that image region 'R' is the union of 'n' subimage regions, i.e, segmentation must be complete.

Condition (ii) requires that points in Region 'R' must be connected (4 (or) 8 - connected)

Condition (iii) indicates that regions must be disjoint i.e, there shouldn't be any common element.

Condition (iv) indicates that all pixels in R_i, $i = 1 \ldots n$ have same intensity level

Condition (v) indicates that two adjacent regions R_i & R_j must be different.

Example:-

Following figures explain how to segment an image based on (i) Similarity and (ii) Discontinuity of intensity values

fig(a) shows image region of constant intensity on a dark backgro

fig(b) shows image, which is obtained by performing image segmentation based on discontinuity of image intensity values.

fig(c) shows image, obtained by performing segmentation based on similarity of intensity values

fig(a) fig(b) fig(c)

fig (b) is obtained by selecting the regions of image which is having atleast one background pixel as a neighbour (i.e) select the regions of image where there is sharp change in intensity values (discontinuity).

Now fig(b) contains an image whose input intensity values (inner regions of image) & background intensities are same (zeros).

Now we need to segment the image (fig(b)) based on similarity of intensity values since inner regions of image and background intensities are same (i.e 0).

fig(c) is obtained by assigning maximum intensity value to the inner regions of fig (b) & black (min intensity) to exterior regions of image and to boundary pixels of image.

fig(d), fig(e), fig(f) shows the images which are obtained by region-based segmentation.

fig(d) fig(e) fig(f)

fig(e) shows the image whose inner regions form a textured pattern.

In region based segmentation, we segment the image based on predefined properties like same intensity values.

Segmentation based on discontinuity of Image Intensity Values:

a) Point detection

b) Line detection

c) Edge detection

We use partial derivatives to detect the changes in intensity values i.e, 1st order and 2nd order partial derivatives.

First Order derivative properties : $(\delta f/\delta x)$

→ $\frac{\partial f}{\partial x} = 0$ in areas of constant intensity

→ $\frac{\partial f}{\partial x} \neq 0$ i.e, first derivative must be non-zero at the onset of step (or) ramp.

→ $\frac{\partial f}{\partial x} \neq 0$ ie, first order derivative must be nonzero along ramp intensity.

Second Order Derivative Properties:

1. Second order derivative $(\partial^2 f/\partial x^2)$ must be nonzero at the onset and end of intensity step (or), ramp.
2. Second order derivative is zero in areas of constant intensity
3. Second order derivative must be zero along ramp intensity.

For a given mask (or) selected mask, we calculate the $\frac{\partial f}{\partial x}$, $\frac{\partial^2 f}{\partial x^2}$ as shown below:

Z_1	Z_2	Z_3	→ x-1 axis
Z_4	Z_5	Z_6	→ x-axis
Z_7	Z_8	Z_9	→ x+1 axis
↓ y-1 axis	↓ y-axis	↓ y+1 axis	→ 3x3 mask

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

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

$$\frac{\partial f}{\partial x} = (Z_7 + Z_8 + Z_9) - (Z_4 + Z_5 + Z_6)$$

$$\frac{\partial f}{\partial y} = (Z_3 + Z_6 + Z_9) - (Z_2 + Z_5 + Z_8)$$

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

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

$$\frac{\partial^2 f}{\partial x^2} = (z_7+z_8+z_9)+(z_1+z_2+z_3)-2[z_4+z_5+z_6]$$

$$\frac{\partial^2 f}{\partial y^2} = (z_3+z_6+z_9)+(z_1+z_4+z_7)-2[z_2+z_5+z_8]$$

Laplacian Operator:-

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

$$\boxed{\nabla^2 f(x,y) = f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)-4f(x,y)}$$

→ First order derivatives provides thicker edges in an image

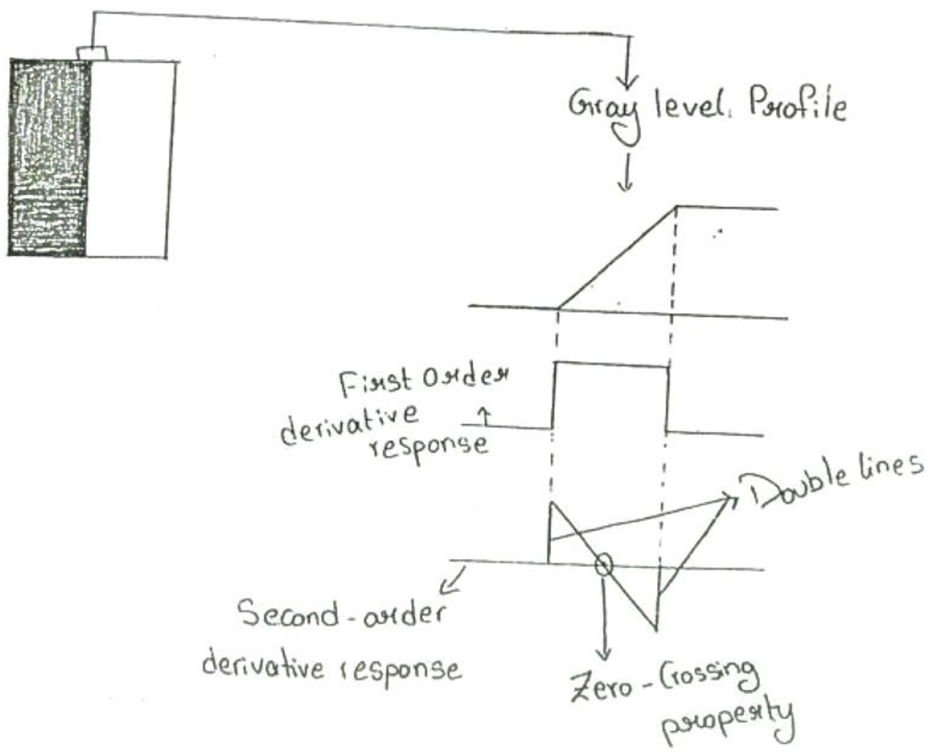
→ Second order derivatives have strong response for thin lines, isolated points, noise.

→ Second order derivatives provides double lines

→ Second order derivatives have zero crossing property.

→ Sign of second order derivative can be used to determine whether a pixel belongs to darkside (or) light side.

Following figure shows the above mentioned properties



The approach for finding (or) computing first and second order derivative at any pixel location in an image is, to use spatial filters (mask processing)

ω_1	ω_2	ω_3
ω_4	ω_5	ω_6
ω_7	ω_8	ω_9

3x3

Mask

z_1	z_2	z_3			
z_4	z_5	z_6			
z_7	z_8	z_9			

$\rightarrow f(x,y)$ = image

$m \times N$

let $\omega_1, \omega_2, \omega_3 \cdots \omega_9$ indicates the mask coefficients

$z_1, z_2, \cdots z_9$ indicates the image intensity values

The mask processing output is, pixel by pixel multiplication and addition of mask coefficients and image intensity values.

ie $R(x,y) = \omega_1 z_1 + \omega_2 z_2 + \omega_3 z_3 + \omega_4 z_4 + \omega_5 z_5 + \omega_6 z_6 + \omega_7 z_7 + \omega_8 z_8 + \omega_9 z_9$

$$\boxed{R(x,y) = \sum_{i=1}^{9} \omega_i z_i} \rightarrow (1)$$

→ We use second order derivative i.e, laplacian operator to detect points, lines.

Point detection:-

We say a point has been detected in an image, if the absolute response of mask centred at (x,y), exceeds specified threshold. For intensity values which are greater than threshold value, we assign '1' and '0' for remaining intensity values

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases}$$

↓

o/p image intensity values

T = Threshold value

R is given in (1)

1	1	1
1	-8	1
1	1	1

a, point detection mask

Line Detection:-

For line detection we use laplacian mask, which is isotropic (independent of direction).

To detect the lines in a horizontal, vertical, diagonal directions we use the following masks.

Let R_1, R_2, R_3, R_4 denotes the response of masks, R's are calculated by using eq-(1) i.e

$$R(x,y) = \sum_{i=1}^{K} z_i w_i$$

If $R_1 > R_2$ at a given point (x,y) then we can say, that point (x,y) is associated with (1st mask) line in the direction of mask 'K' i.e, Horizontal Line.

If $R_2 > R_3$ at (x,y), then we can say point at (x,y) is associated with vertical line (mask 2)

=> In general, $|R_K| > |R_j|$ for all $j \neq K$, then we can say that point has been associated with mask 'K'.

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

Horizontal
mask 1

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

Vertical
mask 2

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

$-45°$
mask 3

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

$+45°$
mask 4

Edge detection:-

a) Edge models: Edge models are classified according to their intensities profile.

(i) Step Edge:- A step edge involves a transition between two intensity levels occuring ideally over a distance of one pixel

fig(a): Vertical step edge

fig(b): Intensity Profile

Ex:- In animations (computer animations), where images are generated by computer, in such applications this kind of edge we can see.

(ii) Ramp edge:- In practise, digital image have edges that are blurred and noisy.

We get blurred images because of improper focusing mechanism (Ex: Improper adjustment of lenses)

We get noisy images, if the electronic components of imaging system (Ex:- Camera) won't work properly (or) due to poor illumination.

This kind of blurred and noisy images are modelled by intensity ramp profile.

The slope of ramp is inversly proportional to degree of blurring in the edge.

(a) Ramp edge

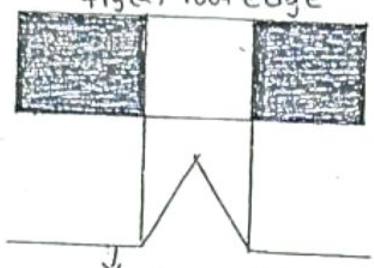
(b) ramp edge profile
↳ analysis of characteristics

Roof edge:-

It arises when objects are closer to the sensor, when objects are closer, they appear brighter. This kind of edges are modelled by roof edge as shown below

Ex:- Satellite images, where thin features such as rods can be modelled by this type of edge.

fig(a) roof edge



fig(b) roof edge profile

Edge Detection:-

Detecting changes in intensities for the purpose of finding edges can be done by using first order and second order derivatives.

→ The strength of an edge and its direction at location (x,y) of a image f, is calculated (or) found out by using gradient (∇f)

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} \rightarrow (1)$$

$$\text{Magnitude of gradient} = mag(\nabla f) = \sqrt{g_x^2 + g_y^2} \rightarrow (2)$$

Gradient vector points in a direction, where there is maximum rate of change of 'f' at location (x,y)

→ The direction of gradient is given as

$$\alpha(x,y) = Tan^{-1}\left[\frac{g_y}{g_x}\right] \rightarrow (3)$$ measured with respect to x-axis.

Advanced Techniques for Edge Detection:-

Previous methods based on simple filtering of image with one (or) more masks but not considering edge characteristics and noise content. These advanced techniques improved edge detection by considering the above factors.

Marr-Hildreth Edge Detector:- (Algorithm)

According to this detector, an operator used for edge detection should have two features

- It should be a differential operator capable of computing analog & d first (or) second derivative at every point in image
- It should be capable of tuned to act at any scale so that large operators can detect blurry edges and small operators detect sharply focused fine detail.

Thus according to this, the operator that fulfilling this conditions is filter $\nabla^2 G$, where

∇^2 is the Laplacian operator $\left(\partial^2/\partial x^2 + \partial^2/\partial y^2\right)$

G is the 2D Gaussian function given as

$$\boxed{G(x,y) = e^{-x^2+y^2/2\sigma^2}}$$

where σ is standard deviation

$\nabla^2 G$ is given as

$$\nabla^2 G = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2}$$

$$= \frac{\partial}{\partial x}\left[\frac{\partial G(x,y)}{\partial x}\right] + \frac{\partial}{\partial y}\left[\frac{\partial G(x,y)}{\partial y}\right]$$

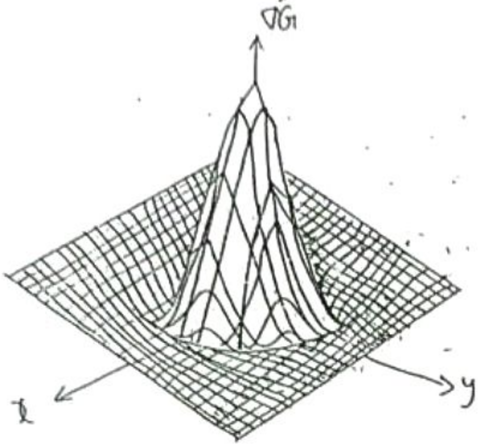
$$= \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-x^2+y^2/2\sigma^2}\right] + \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} e^{-x^2+y^2/2\sigma^2}\right]$$

$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-x^2+y^2/2\sigma^2} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-x^2+y^2/2\sigma^2}$$

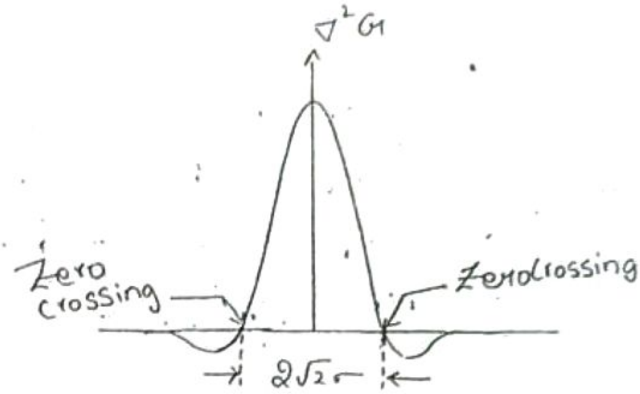
$$\boxed{\nabla^2 G = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-x^2+y^2/2\sigma^2}}$$

This is the expression called as Laplacian of Gaussian (LoG)

Because of the shape (illustrated in below figure) the LoG function sometimes is called the <u>Mexican hat operator</u>.



(a)
3D plot of negative of LoG



(b)
Cross section of (a) showing Zero crossings

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

(C)
5x5 mask approximation to shape in (a)

This algorithm consists of convolving LoG filter with an input image $f(x,y)$

$$g(x,y) = [\nabla^2 G(x,y) * f(x,y)]$$

and then finding zero crossings to determine location of edges in $f(x,y)$. This indicate that we can smoothe image first with Gaussian filter and then compute laplacian of result. They both give identical results.

<u>Conclusion</u>:

1. Smooth image with gaussian $G(x,y) = e^{-x^2+y^2/2\sigma^2}$
2. Calculate Laplacian of Gaussian

$$LoG = \nabla^2 G$$

3. Perform mask processing ie convolution of LoG and image

$g(x,y) = \nabla^2 G * f(x,y)$

4. Find zero crossings of image from step 3.

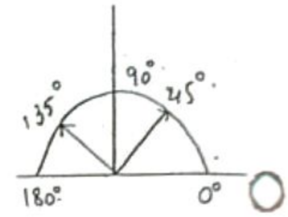
Canny Edge Detector Algorithm:-

Steps:

1. Smooth the given image with gaussian function to remove noise. (Compute the gradient value)

2. Compute the gradient magnitude

3. Thin gradient $= \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$



→ magnitude of gradient $= \sqrt{g_x^2 + g_y^2}$

→ Angle $\alpha(x,y) = \tan^{-1}(g_y/g_x)$

By varying g_x & g_y, the value of angle gets varied. Consider four possible angles for 'α' as $0°, 45°, 90°, 135°$. If angle is between 0 to 22.5° make it 0°

If angle is between 22.5° to 45° make it 45°

If angle is between 180 to 120° make it 135° and so on

This is done to reduce minimum response and to increase maximum response that produces thick lines.

3. Non-Maximum Suppression:

While using Sobel filter, the edges it finds can be either very thick (or) very narrow depending on intensity across the edge and how much images was blurred first. One would like to have edges that are only one pixel wide. The 'non-maximal suppression' step keeps only those pixels on an edge with highest gradient magnitude. These maximal

~~magnitudes should occur right at edge boundary~~, and gradient magnitude should fall of with distance from edge.

So three pixel in 3x3 around pixel (x,y) are examined:

- If $\theta'(x,y) = 0°$, then pixels $(x+1,y)$, (x,y) and $(x-1,y)$ are examined.
- If $\theta'(x,y) = 90°$, then pixels $(x,y+1)$, (x,y) and $(x-1,y-1)$ are examined.
- If $\theta'(x,y) = 45°$, then pixels $(x+1,y+1)$, (x,y) and $(x-1,y-1)$ are examined.
- If $\theta'(x,y) = 135°$, then pixels $(x+1,y-1)$, (x,y) and $(x-1,y+1)$ are examined.

If pixel(x,y) has highest gradient magnitude of three pixel examined, it is kept as an edge. If one of the other two pixels has a highest gradient magnitude, then pixel (x,y) is not on 'corner' of edge and should not be ~~cons~~ classified as an edge pixel.

4. Hysterisis Thresholding :-

Some of the edges detected by steps 1-3 will not actually be valid, but will just be noise. This noise is to be filtered out. Eliminating pixels whose gradient magnitude D falls below some threshold removes the worst of this problem, but introduces a new problem.

A simple threshold may actually remove valid parts of a connected image edge, leaving a disconnected final edge image. This happens in region where the edge's gradient magnitude fluctuations between just above and just below the threshold. Hysterisis is one way of solving this problem. Instead of choosing a single threshold, two thresholds t_{high} and t_{low} are used.

· Pixels with great magnitude $D < t_{low}$ are discarded immediately. However pixels with $t_{low} \leq D < t_{high}$ are only kept if ~~an~~ they form a continuous edge line with pixels with highest gradient magnitude.

This is actually tricky part to implement on a GPU - to do it completely correctly is difficult. However, for this assignment we can implement a partially correct version:

- If pixel (x,y) has gradient magnitude less than t_{low} discard edge (write out black).
- If pixel (x,y) has gradient magnitude greater than t_{high}, keep the edge (write out white).
- If pixel has gradient magnitude between t_{low} and t_{high} and any of its neighbours in a 3x3 region around it have gradient magnitude greater than t_{high}, keep edge (write out white)
- If none of pixel (x,y)'s neighbours have high gradient magnitude but atleast one falls between t_{low} and t_{high} search 5x5 region to see if any of these pixels have a magnitude greater than t_{high}. If so keep the edge (write out white).
- Else discard the edge (write out black).

There may be breaks (or) discontinuities occuring in the edges of images due to lack of proper illumination. Hence edge linking process tells how to link two edges that are discontinued and thus gets boundary information.

Two approaches for edge linking are:

- Local Processing.
- Global Processing

Local Processing:

In this, analyses, the characteristics of pixels in a small neighbourhood about every point (x,y) in that is declared as an edge point. All points that are similar according to predefined criteria are joined, form an edge of pixels that share common properties.

The properties used to find similarity between edge pixels for this kind of analysis are

→ Strength (magnitude)

→ direction (angle) of gradient vector.

An edge pixel with coordinates (s,t) in S_{xy} is similar to the magnitude to pixel (x,y) if

$$|M(s,t) - M(x,y)| \leq E$$

E is the positive threshold.

An edge pixel has an angle similar to pixel at (x,y) if

$$|\alpha(s,t) - \alpha(x,y)| \leq A$$

A is positive angle threshold.

A pixel with coordinates (s,t) in S_{xy} is linked to pixel at (x,y) if both magnitude and direction criteria are satisified. This process is repeated at every location in image.

Histogram Processing:

The number of lines in parameter space (m&c) gives all points (x_k, y_k) in x-y plane. By identifying the collinear points in parameter spaces, we can connect any two points (i.e., broken edge points) in xy coordinate system

→ ~~But from~~

Hough Transform:

→ Consider a point in spatial domain i.e., (x,y)

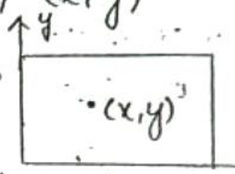
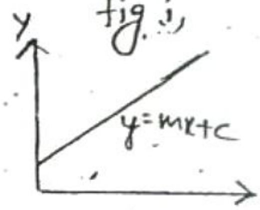


fig. i,

Slope- Intercept form of a line passing through (x,y) is given as

$$y = mx + c \quad \text{—} \; ①$$, shown in fig ii,



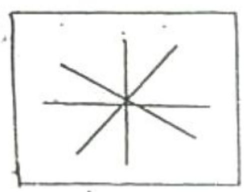
fig

For example $(x,y) = (1,1)$

By substituting $(x,y) = (1,1)$ in eqn-① we get eqn-②

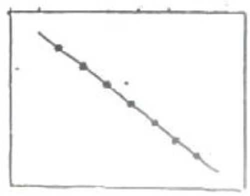
$$1 = m + c \quad \text{—} \; ②$$

We get different lines i.e., (∞ lines) which satisfies eqn-②



If we consider the parameter space i.e., (m&c) coordinate system, we get only one line.

i.e., $c = -mx + y$ —— ③



The '∞' lines which we obtained in ② are nothing but '∞' points on one line given by eqn-③

m&c coordinate system" is called Hough transform

→ The number of lines in parameter space (m&c) gives all points (x_k, y_k) in x-y plane. By identifying the collinear points in parameter space, we can connect any two points (i.e., broken edge points) in 'xy' coordinate system.

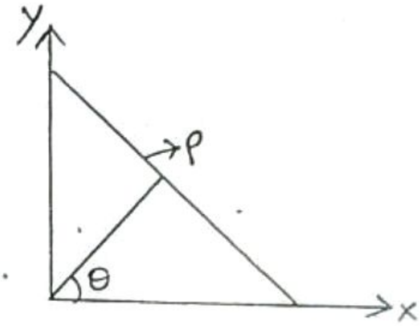
→ But from eqn-①, If ~~slope~~ $\theta = 90°$ then slope $m = \tan\theta = \tan 90° = \infty$.

It means we can't find the vertical lines responses If we use eqn-①. So, we consider slope-Intercept form in polar coordinate system

i.e.,

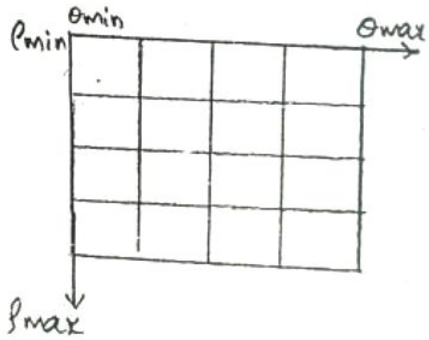
$$x\cos\theta + y\sin\theta = \rho \quad \text{— ④}$$

By using eqn-④ we can find both horizontal and vertical lines responses.



→ Since our image is a finite quantity. So, we divide the m&c coordinate system into finite number of cells known as accumulator cell.

→ $A(i,j)$ represents accumulator cell at i^{th} row and j^{th} column.



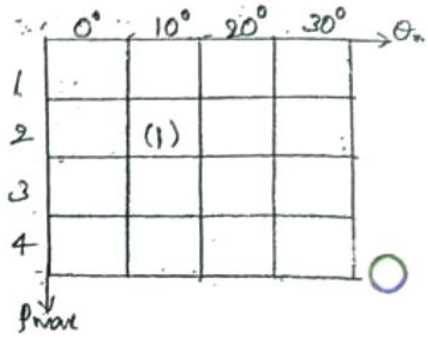
→ Initially set $A(i,j)$ i.e., accumulator cell to zero i.e., $[A(i,j) = 0]$. If we take different allowable values for θ i.e., from θ_{min} to θ_{max} ($-90°$ to $90°$) and we know one edge point (x_k, y_k).

By substituting θ & (x_k, y_k) in eqn – ④ we get the normal distance i.e., 'ρ'

$$x_k \cos\theta + y_k \sin\theta = \rho$$

→ When we get 'ρ' for the corresponding 'θ' value increment the accumulator value by one.

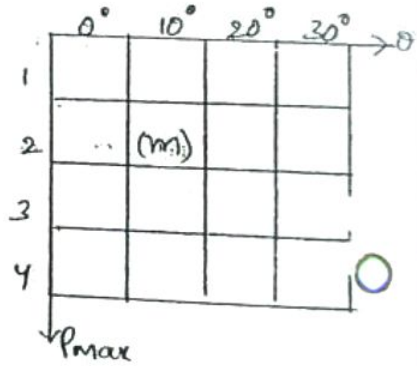
i.e., $A(i,j) = A(i,j) + 1$



Ex! Let us consider for $\theta = 10°$ we get $\rho = 2$.

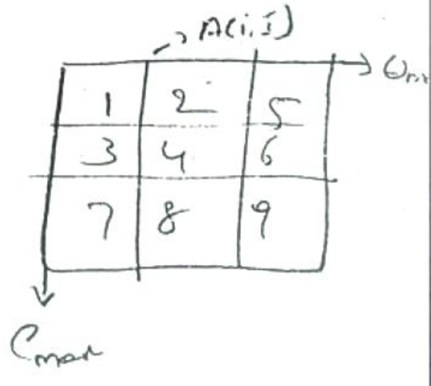
Initially $A(2, 10°)$ is set to zero

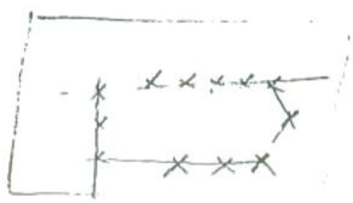
$$\therefore \quad A(2,10°) = A(2,10°) + 1$$
$$= 0 + 1$$
$$= 1$$



If $A(i,j) = m$, it means accumulator cells contains 'm' collinear points for corresponding 'θ' value.

For example if we specify to connect three points in a given Image then, we get





Thresholding:

The point beyond which the response is obtained is called Threshold. Image thresholding is the major process in applications of image segmentation because of its properties, easy to implent and computational speed.

The basics of intensity thresholding:

Consider the intensity of histogram of an image $f(x,y)$ composed of light objects on dark background. The object and background pixels intensity values are grouped into two dominant modes.

For extracting objects from background, select threshold value that separates these modes. Then point (x,y) in image at which $f(x,y) > T$ is called an object point otherwise the background point.

Segmented image $g(x,y)$ is given by

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

T is the selected Threshold.

Thus the process of dividing image into subimages, one with region of interest and other undesired is called as "Global Thresholding".

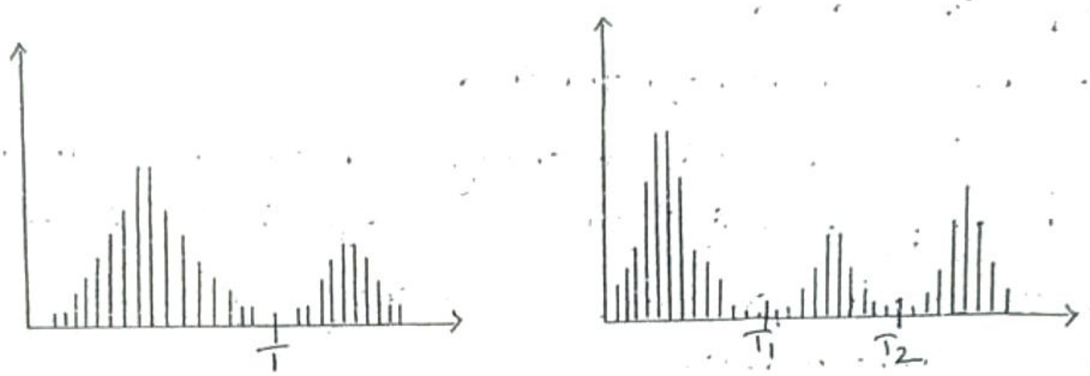
The value of T if changes over an image, it is Variable (or) Local (or) Regional Thresholding. In this, the value of 'T' depends on properties of neighbourhood ie average intensity of pixels in neighbourhood.

Multiple thresholding considers more than one threshold value for image which divides it into no. of subimages

The segmented image is given by

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

where a, b, c are three distinct intensity values.



(a) (b)

Intensity histograms that can be partitioned

(a) by single threshold

(b) by dual threshold

Basic Global Thresholding:-

When intensity distribution of objects and background pixels are distinct, we use single (global) threshold over entire image. A suitable algorithm is capable to estimate the threshold value for each image. The following iterative algorithm can be used for this purpose:

1. Select an initial estimate for the global threshold, T.
2. Segment the image using T. This will produce two groups of pixels (sub images)

   G_1 with pixel intensity values $> T$

   G_2 with pixel intensity values $\le T$
3. Compute the average (mean) intensity values m_1 and m_2 for pixels in G_1 and G_2,
4. Compute new threshold value:

$$T = \frac{\text{mean of } G_1 + \text{mean of } G_2}{2} = \frac{m_1 + m_2}{2}$$

If $T - T_{next} < \Delta T$, then stop the process. Otherwise repeat the process from step(2) to step(4)

where ΔT is the predefined value

→ Larger ΔT, fewer iterations will be performed.

→ Initial threshold should be selected between minimum and maximum intensity level in image. The average intensity is good initial choice for 'T'.

Optimum Global Thresholding Using Ostu's Method:-

This method, maximises the between-class variance.

Let $f(x,y)$ be the image of size $M \times N$ pixels and n_i be the number of pixels with intensity i.

The total number of pixels in image is MN and is given

$$MN = n_0 + n_1 + n_2 + \cdots + n_{L-1}$$

Let $\{0, 1, 2 \ldots L-1\}$ be the L distinct intensity levels in a digital image.

Algorithm:-

1. Calculate normalised histogram which has components

$$P_i = n_i / MN \text{, from which it follows}$$

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

2. Select a threshold $T(k) = 0$, $0 < k < L-1$

3. As a resut of step 2, two subimages clasc 1 (C_1) and class 2 (C_2) are obtained.

C_1 with intensity values of range $[0, k]$

C_2 with intensity values of range $[k+1, L-1]$

The probability of class C_1 is given as

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

The probability of class C_2 occurring is

$$P_2(K) = \sum_{i=K+1}^{L-1} P_i = 1 - P_1(k)$$

According to Baye's Theorem

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

The mean intensity value of pixels in class C_1 is

$$m_1(K) = \sum_{i=0}^{K} i \cdot P(i/C_1)$$

$$= \sum_{i=0}^{K} i \, P(C_1/i) \, P(i) / P(C_1)$$

$$\boxed{m_1(K) = \frac{1}{P_1(K)} \sum_{i=0}^{K} i \, P_i}$$

$P_i \rightarrow$ probability of i^{th} value.

The mean intensity value of pixels in class C_2 is

$$m_2(K) = \sum_{i=K+1}^{L-1} i \cdot P(i/C_2)$$

$$\boxed{m_2(K) = \frac{1}{P_2(K)} \sum_{i=K+1}^{L-1} i \, P_i}$$

The cumulative mean (average intensity) upto level K is given

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

The average intensity of entire image, i.e global mean is given

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

By substituting above values, the result of two equations can be obtained as

(i) $P_1 m_1 + P_2 m_2$

$$= P_1 \cdot \frac{1}{P_1(K)} \sum_{i=0}^{K} i \, P_i + P_2 \cdot \frac{1}{P_2(K)} \sum_{i=K+1}^{L-1} i \, P_i$$

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

In Region based segmentation, the segmentation is based on similarity on intensity values and on finding the regions directly.

Region Based Segmentation
- Region Growing
- Region Splitting and Merging.

Region Growing:-

In this, we group pixels (or) subimages of same properties to form larger regions.

→ Select a seed point in the image. Consider neighbourhood around seed point and connect those.

→ Process is continued till the pixels are found belonging to certain category.

→ Seed point is obtained by selecting random values (or) based on threshold.

Region-growing algorithm based on 8-connectivity is:

$f(x,y)$ be input image array, $S(x,y)$ denote seed array. 'f' and 'S' are of to be same size.

1. Find all connected components in $S(x,y)$. Label connected pixel as '1' and not conected as '0'.
2. Form an image 'f_Q' such that at pair of coordinates (x,y),

$$f_Q(x,y) = \begin{cases} 1 & \text{satisfying for given predicate } Q \\ 0 & \text{otherwise} \end{cases}$$

3. Let 'g' be an image formed by connecting the 8-neighbourhood pixels to seed point.

4. label each component in 'g' with different labels. This is the segmented image obtained by region growing.

Region Splitting and Merging:-

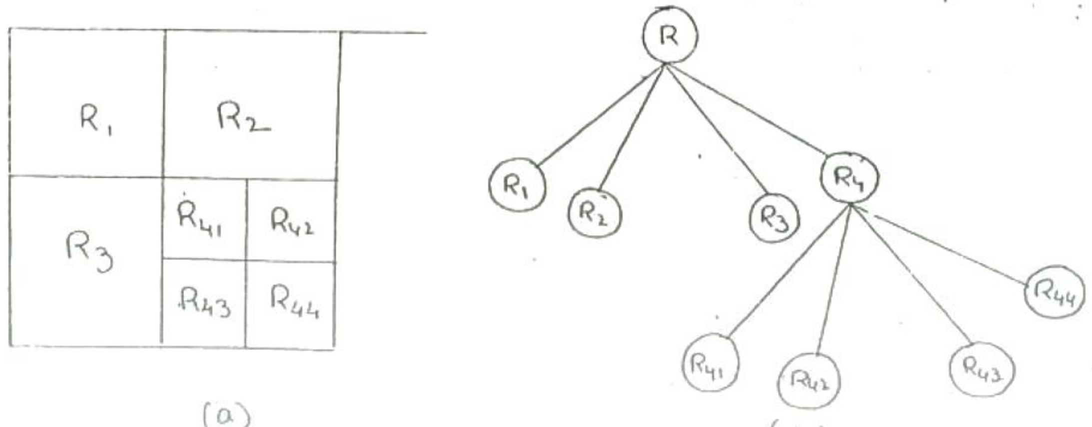
In this method image is first subdivided into a set of arbitrary regions. If these regions are different from one another, keep on splitting. Splitting is done to subimage until equal intensities are obtained. Then merge those regions.

Let R represent entire region and select a predicate 'Q'. Subdivide it into successive smaller and smaller regions. If $Q(R) =$ False, divide image into quadrants and if Q is false for any quadrant, divide it into subquadrants. This is also called as quadtrees. In this image splitting technique is also called as quadtrees. In this image R, is divided into 4 quadrants and R_4 is further divided into 4 subquadrants. Final partition contains adjacent regions with identical properties.

Then merging is done to these regions that satisifying predicate Q.

Steps followed:

1. Split image into 4 disjoint quadrants for $Q(R) =$ False
2. If no further splitting possible, merge adjacent regions R_j and R_k for $Q(R_j \cup R_k) =$ True
3. Stop when no further merging is possible



(a) (b)

fig,

(a): Noisy shaded image

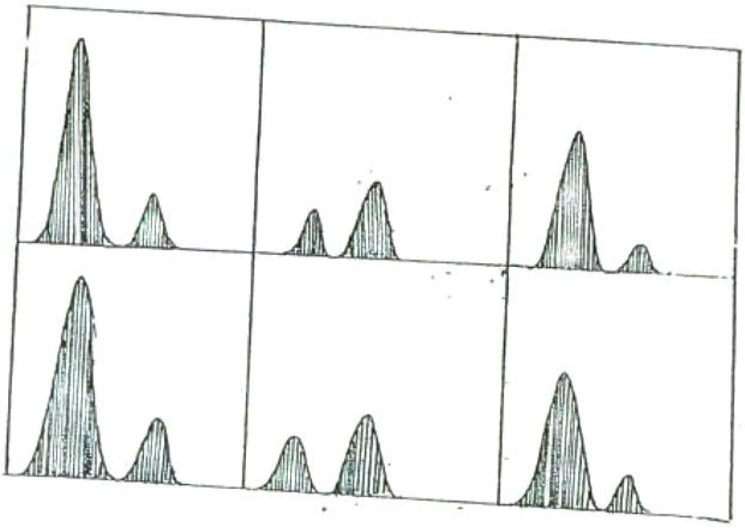
(b): histogram of (a)

(c): Segmentation of (a) using iterative global thresholding

(d): Image subdivided into six ~~image~~ subimages.

(e): Result of applying Ostu's method to each subimage individually.

Histograms of six subimages:



Consider a image and its histogram as shown in fig (a) and (b). The output in fig (c) shows that the image can not be segmented with global threshold and Ostu's method properly.

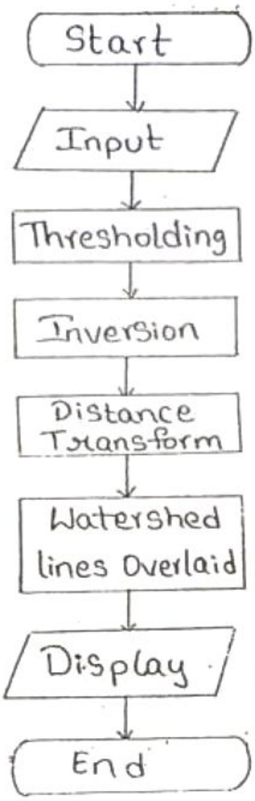
To overcome this problem, subdivide the original image into six subregions and apply Ostu's global method to each subimage. This produces a reasonable image result By seeing the histograms of these subimages, the improvement in resultant image is clearly observed.

Segmentation Using Morphological Watersheds:

A grayscale image can be divided into distinct catchment basins which are then filled by water. Dams are built wherever it is necessary to prevent the merging of two adjacent basins. Once the surface is immersed in water, the dams outline the watershed lines. Thus, watershed lines mark the boundaries of the catchment basins. These segment the image into the desired regions.

Watershed Segmentation Process can be expressed with an Algorithm as shown below:

Watershed Algorithm:



The problem of this method is that this algorithm leads to over segmentation.

To overcome this problem, image is smoothened before the segmentation process.

Multiple Thresholds:

In this multiple thresholding, more than one value is selected that produces more classes.

For K classes, $C_1, C_2 \ldots C_K$, the between-class variance is given as

$$\sigma_B^2 = \sum_{k=1}^{K} P_k (m_k - m_G)^2$$

where

$$P_k = \sum_{i \in C_k} P_i$$

$$m_k = \frac{1}{P_k} \sum_{i \in C_k} i P_i$$

K classes are separated by K-1 thresholds and the values $k_1^*, k_2^* \ldots k_{k-1}^*$ maximises σ_B^2 and is given as:

$$\sigma_B^2 (k_1^*, k_2^* \ldots k_{k-1}^*) = \max_{0<k_1<k_2<\ldots k_{n-1}<L-1} \sigma_B^2 (k_1, k_2 \ldots k_{k-1})$$

For 3 classes with 3 intensity values, σ_B^2 is given as:

$$\sigma_B^2 = P_1 (m_1 - m_G)^2 + P_2 (m_2 - m_G)^2 + P_3 (m_3 - m_G)^2$$

where

$$P_1 = \sum_{i=0}^{k_1} P_i \qquad m_1 = \frac{1}{P_1} \sum_{i=0}^{k_1} i P_i$$

$$P_2 = \sum_{i=k_1+1}^{k_2} P_i \qquad m_2 = \frac{1}{P_2} \sum_{i=k_1+1}^{k_2} i P_i$$

$$P_3 = \sum_{i=k_2+1}^{L-1} P_i \qquad m_3 = \frac{1}{P_3} \sum_{i=k_2+1}^{L-1} i P_i$$

Relationships:

$$P_1 m_1 + P_2 m_2 + P_3 m_3 = m_G$$

$$P_1 + P_2 + P_3 = 1$$

k_1^*, k_2^* are the threshold value that maximise $\sigma_B^2 (k_1, k_2)$

$$\sigma_B^2 (k_1^*, k_2^*) = \max_{0<k_1<k_2<L-1} \sigma_B^2 (k_1, k_2)$$

The thresholding image is given as:

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) \le k_1^* \\ b & \text{if } k_1^* < f(x,y) \le k_2^* \\ c & \text{if } f(x,y) > k_2^* \end{cases}$$

and $$\eta(k_1^*, k_2^*) = \frac{\sigma_B^2(k_1^*, k_2^*)}{\sigma_G^2}$$

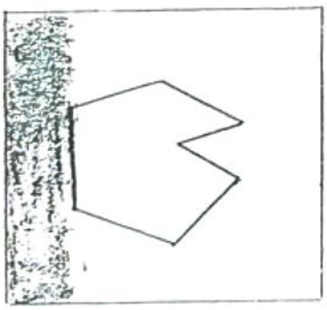
where a, b, c are three intensity values

σ_G^2 is total image variance.

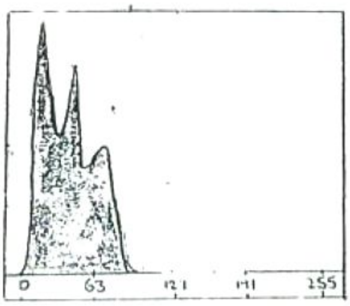
Variable Thresholding:-

Image Partitioning:

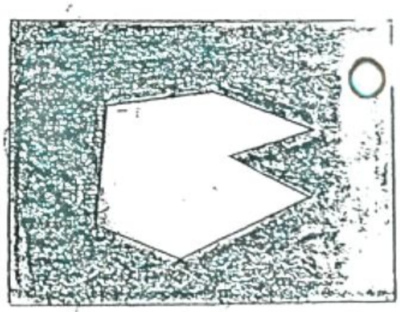
One of the simplest approaches to variable thresholding is to subdivide an image into nonoverlapping rectangles. This approach is used to compensate for non-uniformities in illumination and/(or) reflectance. The rectangles are chosen small enough so that illumination of each is approximately uniform.



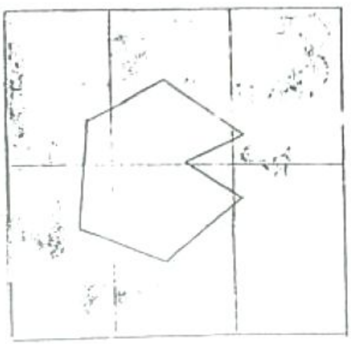
(a)



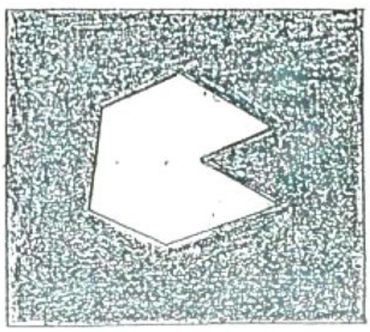
(b)



(c)



(d)



(e)

$$\boxed{P_1 m_1 + P_2 m_2 = m_G}$$

(ii) $P_1 + P_2 = r$

$$= \sum_{i=0}^{K} P_1(K) + \sum_{i=K+1}^{L-1} P_2(K).$$

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

$$\boxed{P_1 + P_2 = 1}$$

ie sum of overall probabilities is equal to '1'.

4. To evaluate 'goodness' of threshold at level K, we use dimensionless metric 'η' given as

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

where,

σ_B^2 is the between-class variance.

σ_G^2: is the global variance ie intensity variance of all pixels in image.

σ_B^2 is defined as

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

$$= P_1\left[m_1 - (P_1 m_1 + P_2 m_2)\right]^2 + P_2\left[m_2 - (P_1 m_1 + P_2 m_2)\right]^2$$

$$= P_1\left[m_1(1-P_1) - P_2 m_2\right]^2 + P_2\left[m_2(1-P_2) - P_2 m_2\right]^2$$

$$= P_1 P_2^2\left[m_1 - m_2\right]^2 + P_2 P_1^2\left[m_2 - m_1\right]^2$$

$$= P_1 P_2\left[m_1 - m_2\right]^2\left[P_1 + P_2\right]$$

we have $P_2 = 1 - P_1$

$$m_1 = \frac{1}{P_1(K)} \sum_{i=0}^{K} i P_i = m_G / P_1(K)$$

$$m_1 = \frac{m_G}{P_1(K)}$$

Similarly $m_2 = \frac{m_G}{P_2(K)}$

$$\Rightarrow \sigma_B^2 = \frac{(m_G P_1 - m)^2}{P_1(1-P_1)}$$

(or)

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

5. The final results are

$$\eta(K) = \frac{\sigma_B^2(K)}{\sigma_G^2}$$

$$\sigma_B^2(K) = \frac{\left[m_G P_1(K) - m(K)\right]^2}{P_1(K)(1 - P_1(K))}$$

K^* will be the optimum threshold value, that maximizes $\sigma_B^2(K)$:

$$\boxed{\sigma_B^2(K^*) = \max_{0 \le K \le L-1} \sigma_B^2(K)}$$

Evaluate the above equation for all K values and select 'K' value that yielded maximum $\sigma_B^2(K)$ as threshold. If it exists for more than one value of K, take the average of those values.

6. After obtaining K^* value, input image $f(x,y)$ is segmented as:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > K^* \\ 0 & \text{if } f(x,y) \le K^* \end{cases}$$

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

For this normalised metric will be $\eta(K^*)$ and lies in range

$$0 \le \eta(K^*) \le 1$$

(b): Corresponding quadtree. R represents entire image region.

The Use of Motion in Segmentation :-

let $f(x,y,t_i)$ represents the image taken at time 't_i' and consider that image as a still image.

$f(x,y,t_j)$ represents the image taken at time 't_j' and cosider the image as a motion image (i.e the picture taken when object is in motion).

Now calculate the difference between $f(x,y,t_i)$ and $f(x,y,t_j)$. If the difference is greater than 'T' (threshold) specified, mark it as '1' in difference image

i.e, $$d_{ij}(x,y) = \begin{cases} 1 & \text{if } f(x,y,t_i) - f(x,y,t_j) > T \\ 0 & \text{otherwise} \end{cases} \quad \longrightarrow (1)$$

All the pixels in $d_{ij}(x,y)$ with value '1' is because object motion.

→ Sometimes because of noise, we get 1s in $d_{ij}(x,y)$ to eliminate this noise pixels:

We use 4 (or) 8 connectivity i.e, we consider only 4 (or) 8 connected components of 1s in $d_{ij}(x,y)$ and then ignore any region that has less than a predetermined number of elements.

→ Accumulative image :-

Let $f(x,y,t_1)$ is a reference image and $f(x,y,t_2)$ $f(x,y,t_3)$ ..... $f(x,y,t_n)$ are sequence of image frames.

The difference between $f(x,y,t_1)$ and $f(x,y,t_2)$ .... $f(x,y,t_n)$ is known (ADI) Accumulative Difference Image

A counter for each pixel location in the accumulative image is incremented every time a difference occurs at that pixel location between the reference and an image in the sequence. Thus when the Kth frame is being compared with the reference, the entry in a given pixel of accumulative image gives number of times the intensity at that point was different.

The following are the three types of Accumulative image:

a) Absolute :

$$A_k(x,y) = \begin{cases} A_{k-1}(x,y)+1 & \text{if } |R(x,y)-f(x,y,k)| > T \\ A_{k-1}(x,y) & \text{otherwise} \end{cases}$$

b) Positive:

$$P_k(x,y) = \begin{cases} P_{k-1}(x,y)+1 & \text{if } [R(x,y)-f(x,y,k) > T \\ P_{k-1}(x,y) & \text{otherwise} \end{cases}$$

c) Negative:

$$N_k(x,y) = \begin{cases} N_{k-1}(x,y)+1 & \text{if } [R(x,y)-f(x,y,k)] < -T \\ N_{k-1}(x,y) & \text{otherwise} \end{cases}$$

UNIT-4

Color Image Processing

Introduction

The use of color in image processing is motivated by two principal factors. They are Color is a powerful descriptor that often simplifies object identification and extraction from a scene. Humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. Color in image processing is divided into two major areas,

Full-color processing: Images acquired with a full-color sensor, such as color TV camera or Color scanner.

Pseudo-color processing: Assigning a color to a particular monochrome intensity or range of Intensities.

4.1. Color Fundamentals

Color of an object is determined by the nature of the light reflected from it. In 1666, Sir Isaac Newton discovered that when a beam of sunlight passes through a glass prism, the emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other. As the following Fig. shows that the color spectrum may be divided into six broad regions: violet, blue, green, yellow, orange, and red.

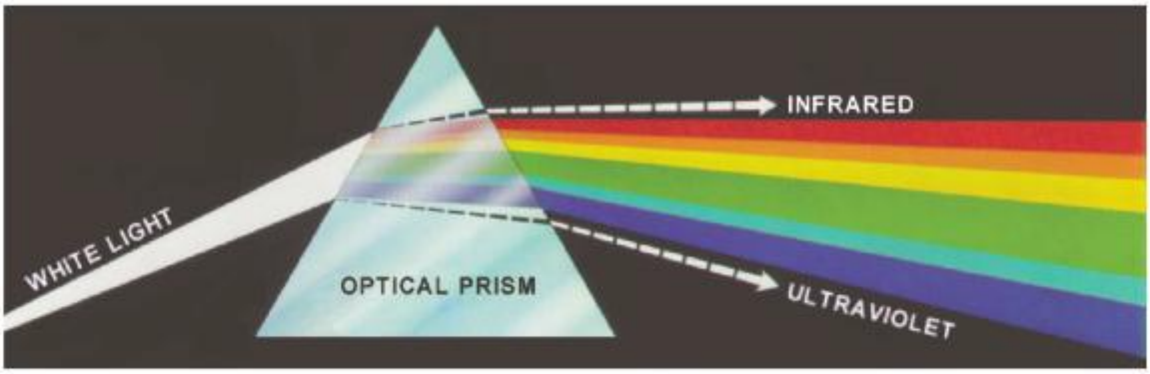


Fig. Color spectrum seen by passing white light through a prism

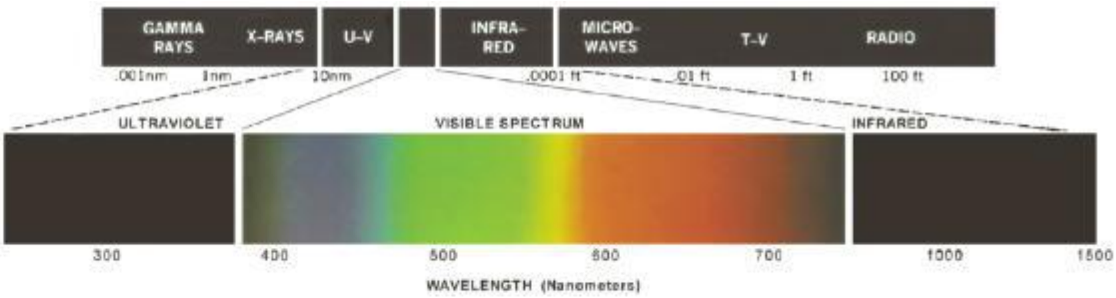


Fig. Wavelengths comprising the visible range of the electromagnetic spectrum

Visible light is composed of a relatively narrow band of frequencies in the electromagnetic spectrum. A body that reflects light that is balanced in all visible wavelengths appears white to the observer. However, a body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color. For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelengths. Characterization of light is central to the science of color. If the light is achromatic (void of color), its only attribute is its intensity, or amount. Achromatic light is what viewers see on a black and white television set.

Chromatic light spans the electromagnetic spectrum approximately from 400 to 700nm. Three basic quantities are used to describe the quality of a chromatic light source: radiance, luminance, and brightness.

Radiance: Radiance is the total amount of energy that flows from the light source, and it is usually measured in watts (W).

Luminance: Luminance, measured in lumens (lm), gives a measure of the amount of energy an observer perceives from a light source.

Brightness: Brightness is a subjective descriptor that is practically impossible to measure.

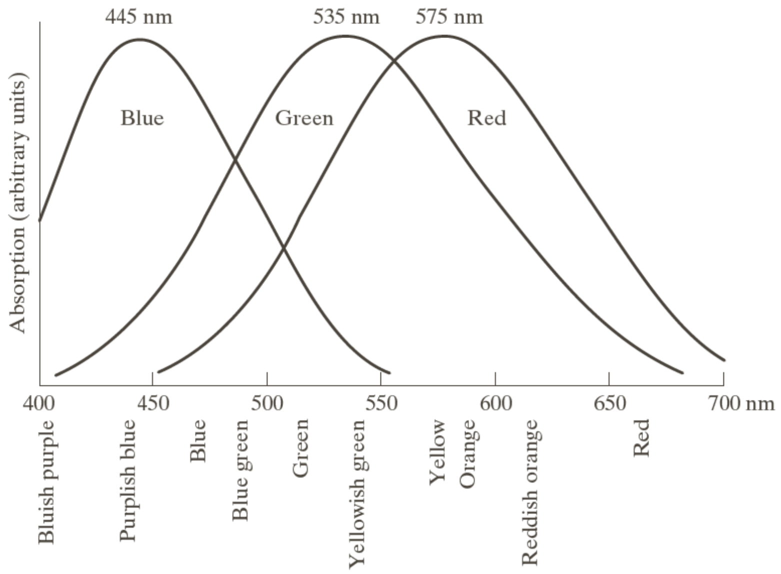


Fig. Absorption of light by the red, green and blue cones in the human eye as a function of wavelength

Cones are the sensors in the eye responsible for color vision. Detailed experimental evidence has established that the 6 to 7 million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue.

Approximately 65% of all cones are sensitive to red light, 33% are sensitive to green light, and only about 2% are sensitive to blue (but the blue cones are the most sensitive). The above figure shows average experimental curves detailing the absorption of light by the red, green, and blue cones in the eye. Due to these absorption characteristics of the human eye, colors are seen as variable combinations of the so- called primary colors red (R), green (G), and blue (B).

The primary colors can be added to produce the secondary colors of light --magenta (red plus blue), cyan (green plus blue), and yellow (red plus green). Mixing the three primaries or a secondary with its opposite primary color, in the right intensities produces white light.

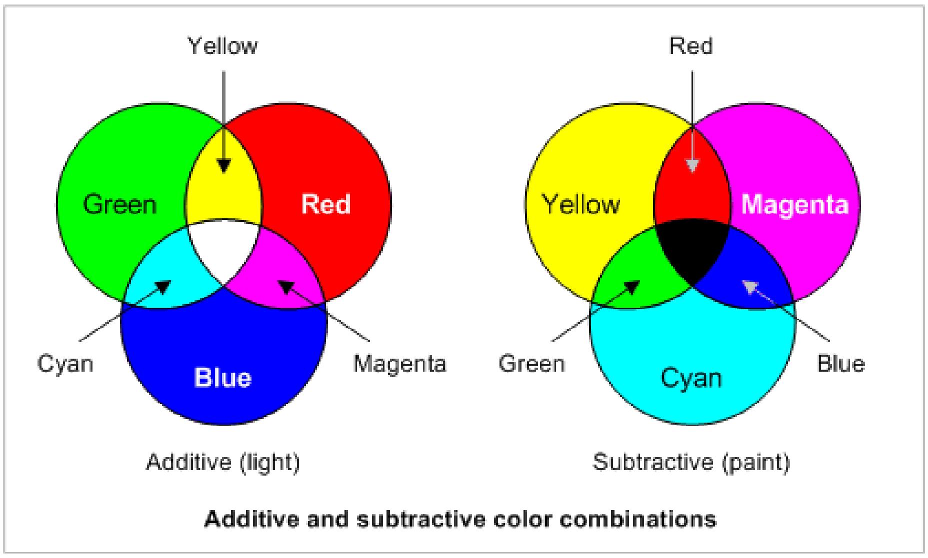


Fig. Primary and Secondary Colors of light and pigments

The characteristics generally used to distinguish one color from another are brightness, hue, and saturation. Brightness embodies the chromatic notion of intensity. Hue is an attribute associated with the dominant wavelength in a mixture of light waves. Hue represents dominant color as perceived by an observer. Saturation refers to the relative purity or the amount of white light mixed with a hue. The pure spectrum colors are fully saturated. Colors such as pink (red and white) and lavender (violet and white) are less saturated, with the degree of saturation being inversely proportional to the amount of white light-added. Hue and saturation taken together are called chromaticity, and. therefore, a color may be characterized by its brightness and chromaticity. The amounts of red, green and blue needed to form any particular color are called the tristimulus values and are denoted by red (X), green (Y) and blue (Z) needed to form a particular color. A color can be specified by its trichromatic coefficients and defined as

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

$$y = \frac{Y}{X+Y+Z} \qquad\qquad x+y+z=1$$

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

4.2. Color Models

The purpose of a color model (also called color space or color system) is to facilitate the specification of colors in some standard, generally accepted way. In essence, a color model is a specification of a coordinate system and a subspace within that system where each color is represented by a single point.

Most color models used are oriented either toward hardware or toward applications where color manipulation is goal. The most commonly used hardware-oriented models are RGB (Red, Green, Blue) for color monitors and video cameras, **CMY** (Cyan, Magenta, Yellow) and **CMYK** (CMY + Black) for color printing and HSI (Hue, Saturation, Intensity) which corresponds closely with the way humans describe and interpret color.

4.2.1. The RGB Color Model:

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

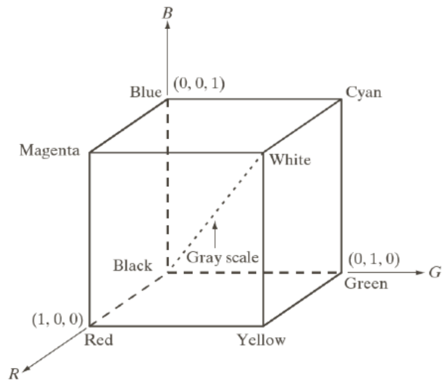


Fig. Schematic of the RGB color cube

Images represented in the RGB color model consist of three component images, one for each primary color. When fed into an RGB monitor, these three images combine on the phosphor screen to produce a composite color image.

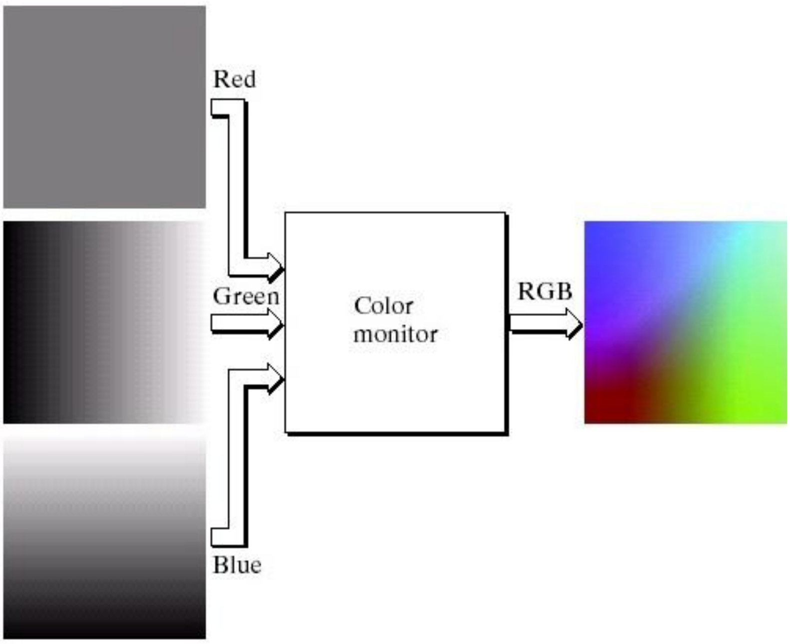


Fig. Generating the RGB image of the cross Sectional color plane

The number of bits used to represent each pixel in RGB space is called the pixel depth. Consider an RGB image in which each of the red, green, and blue images is an 8-bit image. Under these conditions each RGB color pixel [that is, a triplet of values (R, G, B)] is said to have a depth of 24 bits C image planes times the number of bits per plane). The term full-color image is used often to denote a 24-bit RGB color image. The total number of colors in a 24-bit RGB image is $(28)^3$ = 16,777,216.

4.2.2. The CMY and CMYK Color models

Cyan, magenta, and yellow are the secondary colors of light or, alternatively, the primary colors of pigments. For example, when a surface coated with cyan pigment is illuminated with white light, no red light is reflected from the surface. That is, cyan subtracts red light from reflected white light, which itself is composed of equal amounts of red, green, and blue light. Most devices that deposit colored pigments on paper, such as color printers and copiers, require CMY data input or perform an RGB to CMY conversion internally. This conversion is performed using

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

Where, again, the assumption is that all color values have been normalized to the range [0, 1]. The above equation demonstrates that light reflected from a surface coated with pure cyan does not contain red (that is, $C = 1 — R$ in the equation). Similarly, pure magenta does not reflect green, and pure yellow does not reflect blue. So, the RGB values can be obtained easily from a set of CMY values by subtracting the individual CMY values from 1. Equal amounts of the pigment primaries, cyan, magenta, and yellow should produce black. In practice, combining these colors for printing produces a muddy-looking black. So, in order to produce true black, a fourth color, black is added, giving rise to the CMYK color model.

4.2.3. HSI color model

When humans view a color object, we describe it by its hue, saturation, and brightness. Hue is a color attribute that describes a pure color (pure yellow, orange, or red), whereas saturation gives a measure of the degree to which a pure color is diluted by white light. Brightness is a subjective descriptor that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.

Intensity (gray level) is a most useful descriptor of monochromatic images. This quantity definitely is measurable and easily interpretable. The HSI (hue, saturation, intensity) color model, decouples the intensity component from the color-carrying information (hue and Saturation) in a color image. As a result, the HSI model is an ideal tool for developing image processing algorithms based on color descriptions that are natural and intuitive to humans.

In the following figure the primary colors are separated by 120° and the secondary colors are 60° from the primaries, which means that the angle between secondaries is also 120°.

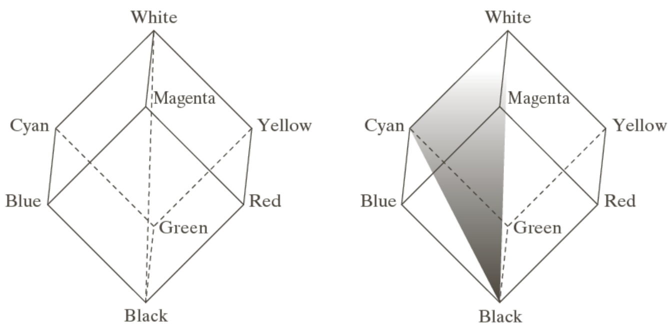


Fig. The relation between RGB and HSI color model

The hue of the point is determined by an angle from some reference point. Usually (but not always) an angle of 0° from the red axis designates 0 hue, and the hue increases counter clockwise from there. The saturation (distance from the vertical axis) is the length of the vector from the origin to the point. The origin is defined by the intersection of the color plane with the vertical intensity axis. The important components of the HSI color space are the vertical intensity axis, the length of the vector to a color point, and the angle this vector makes with the red axis.

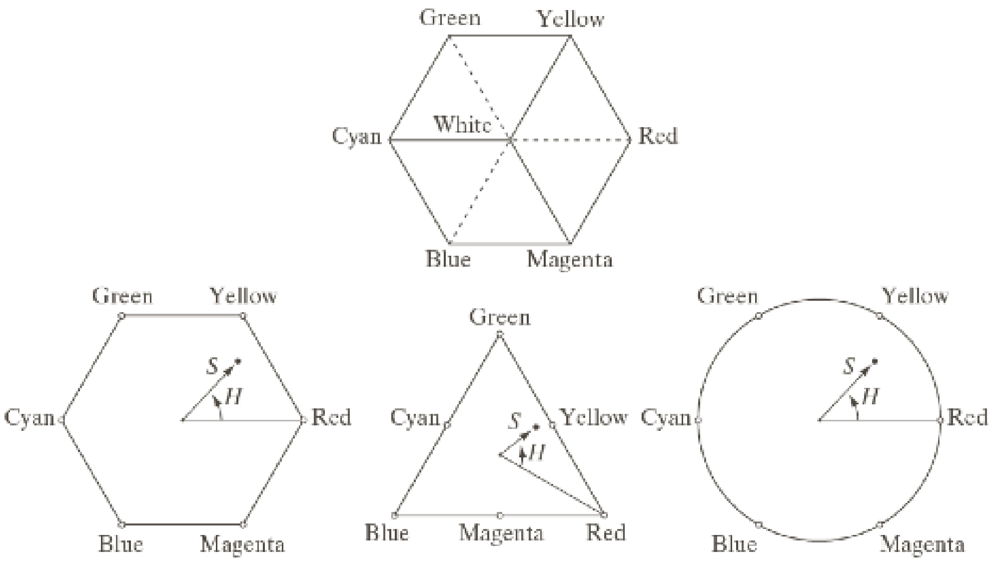


Fig. Hue and saturation in the HSI color model

4.2.4. Conversion from RGB color model to HSI color model

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

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

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

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right]$$

$$I = \frac{1}{3}(R+G+B)$$

It is assumed that the RGB values have been normalized to the range [0, 1] and that angle θ is measured with respect to the red axis of the HST space. The SI values are in [0,1] and the H value can be divided by 360o to be in the same range.

4.2.5. Conversion from HSI color model to RGB color model

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

RG sector ($0^o \le H < 120°$):

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

$$B = I(1 - S)$$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

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

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

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

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

Then the RGB components are

$$H = H - 120^\circ$$
$$R = I(1 - S)$$
$$G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$
$$B = 3I - (R + G)$$

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

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

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

Then the RGB components are

$$G = I(1 - S)$$
$$B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$
$$R = 3I - (G + B)$$

4.3. Pseudo color image processing

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

- Intensity Slicing
- Gray Level to Color Transformation

4.3.1. Intensity Slicing:

The technique of intensity (sometimes called density) slicing and color coding is one of the simplest examples of pseudo color image processing. If an image is interpreted as a 3-D function (intensity versus spatial coordinates), the method can be viewed as one of placing planes parallel to the coordinate plane of the image; each plane then "slices" the function in the area of intersection. The following figure shows an example of using a plane at f(x, y) = li to slice the image function into two levels.

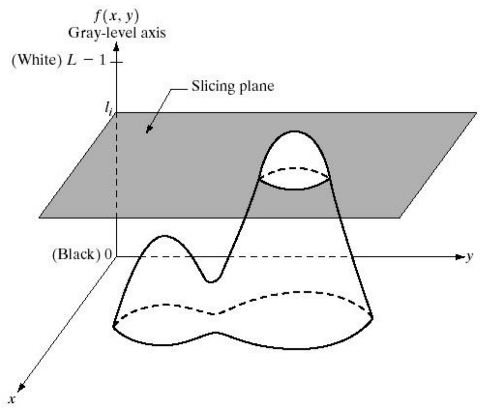


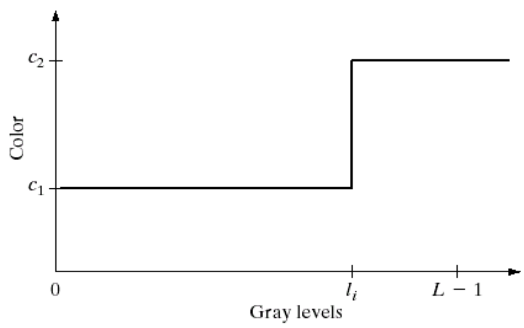
Fig. Geometric interpretation of the intensity slicing technique

If a different color is assigned to each side of the plane shown in the above figure any pixel whose gray level is above the plane will be coded with one color and any pixel below the plane will be coded with the other. Levels that lie on the plane itself may be arbitrarily assigned one of the two colors. The result is a two-color image whose relative appearance can be controlled by moving the slicing plane up and down the gray-level axis.

In general, the technique may be summarized as follows. Let [0, L - 1] represent the gray scale, level l_o represent black [f(x, y) = 0], and level l_{L-1} represent white [f(x, y) = L - 1]. Suppose that P planes perpendicular to the intensity axis are defined at levels $l_1, l_2, \ldots, l_p$.. Then, assuming that 0 < P < L – 1, the P planes partition the gray scale into P + 1 intervals, V1, V2,..., Vp + 1. Gray-level to color assignments are made according to the relation

$$f(x, y) = c_k \text{ if } f(x, y) \in V_k$$

Where c_k is the color associated with the k^{th} intensity interval V_k defined by the partitioning planes at l = k - 1 and l = k. An alternative representation defines the same mapping according to the mapping function shown in the following figure. Any input gray level is assigned one of two colors, depending on whether it is above or below the value of *li*. When more levels are used, the mapping function takes on a staircase form.



An alternative representation of the intensity-slicing technique

4.3.2. Gray Level to Color Transformation:

This approach is to perform three independent transformations on the gray level of any input pixel. The three results are then fed separately into the red, green, and blue channels of a color television monitor. This method produces a composite image whose color content is modulated by the nature of the transformation functions. These are transformations on the gray-level values of an image and are not functions of position. In intensity slicing, piecewise linear functions of the gray levels are used to generate colors. On the other hand, this method can be based on smooth, nonlinear functions, which, as might be expected, gives the technique considerable flexibility. The output of each transformation is a composite image.

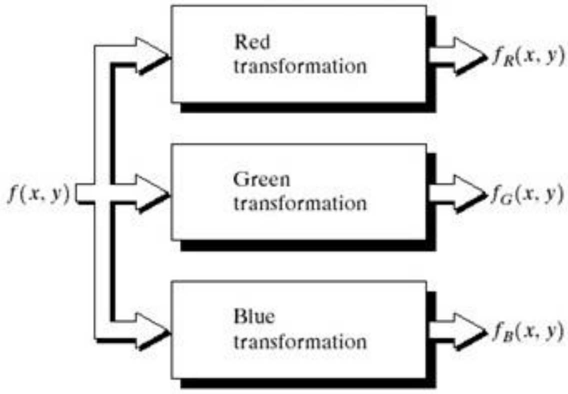


Fig. Functional block diagram for pseudo color image processing

4.4. Full color image processing

Full-color image processing approaches fall into two major categories. In the first category, each component image is processed individually and then forms a composite processed color image from the individually processed components. In the second category, one works with color pixels directly. Because full-color images have at least three components, color pixels really are vectors. For example, in the RGB system, each color point can be interpreted as a vector extending from the origin to that point in the RGB coordinate system.

Let c represent an arbitrary vector in RGB color space:

$$\mathbf{c} = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

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

$$\mathbf{c}(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

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

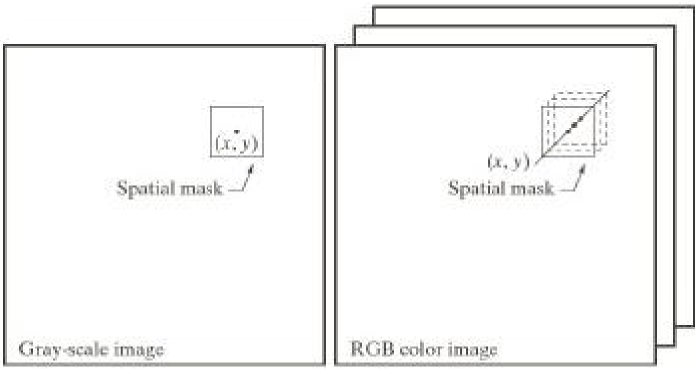


Fig. Spatial masks for (a)gray-scale and (b) RGB color images.

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

4.5. Color Transformations

Color transformations deal with processing the components of a color image within the context of a single color model, without converting components to different color space.

4.5.1. Formulation

We can model color transformations using the expression

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

Where f(x, y) is color input image, g(x, y) is the transformed color output image and T is the operator over a spatial neighborhood of (x, y). Each f(x, y) component is a triplet in the chosen color space. For a given transformation the cost of converting from one color space to another is also a factor to implement it. Hence, we wish to modifying intensity of an image in different color spaces, using the transform

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

When only data at one pixel is used in the transformation, we can express the transformation as:

$$s_i = T_i(r_1, r_2, \ldots\ldots, r_n) \quad i = 1, 2, \ldots, n$$

Where r_i = color component of $f(x, y)$

s_i = color component of $g(x, y)$

In RGB color space,

$$s_R(x,y) = kr_R(x,y)$$
$$s_G(x,y) = kr_G(x,y)$$
$$s_B(x,y) = kr_B(x,y)$$

In HSI color space,

$$s_I(x,y) = kr_I(x,y)$$

In CMY color space,

$$s_C(x,y) = kr_C(x,y) + (1-k)$$
$$s_M(x,y) = kr_M(x,y) + (1-k)$$
$$s_Y(x,y) = kr_Y(x,y) + (1-k)$$

4.5.2. Color Complements

Color complement replaces each color with its opposite color in the color circle of the Hue component. This operation is analogous to image negative in a gray scale image. Color complements are used to enhance the details in dark regions of a color image.

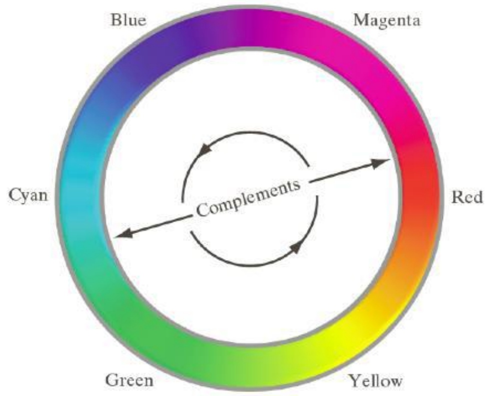


Fig. Complements on the Circle

4.5.3. Color Slicing

Color slicing is the process of highlighting a specific range of colors in an image is useful for separating object from their surroundings. It is more complex than gray-level slicing, due to multiple dimensions for each pixel. This can be done by selecting the region that needs to be high spotted in a cube of width '*w*'. The outside region must be mapped with a neutral color. Then the transformation is given by

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

Set to gray

Keep the original color

$i = 1, 2, \ldots, n$

or

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

Set to gray

Keep the original color

$i = 1, 2, \ldots, n$

4.5.4. Tone and Color Corrections

Effectiveness of these transformations judged ultimately in print. But developed, refined and evaluated on monitors. Need to maintain a high degree of color consistency between monitors used and eventual output devices. *Device-independent* color model, relating the color gamut's of the monitors and output devices. The success of this approach is a function of the quality of the color profiles used to map each device to the model and the model itself. The model of choice for many color management system (CMS) is the CIE *L*a*b* model.

$$L^* = 116 \cdot h\left(\frac{Y}{Y_w}\right) - 16$$

$$a^* = 500\left[h\left(\frac{X}{X_w}\right) - h\left(\frac{Y}{Y_w}\right)\right]$$

$$b^* = 200\left[h\left(\frac{Y}{Y_w}\right) - h\left(\frac{Z}{Z_w}\right)\right]$$

where

$$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.787q + 16/116 & q \le 0.008856 \end{cases}$$

$X_w, Y_w,$ and Z_w are reference white tristimulus values

Like the HIS system, the *L*a*b* system is an excellent decoupler of intensity (represented by lightness *L**) and color (represent by *a** for red minus green and *b** for green minus blue). The tonal range of an image, also called its *key type*, refer to its general distribution of color intensities. Most of the information in *high-key* images are located predominantly at low intensities; *middle-key* images lie in between.

4.5.5. Histogram Processing

Histogram processing transformations can be applied to color images in an automated way. As might be expected, it is generally unwise to histogram equalize the component of a color image independently. This results in erroneous color. A more logical approach is to spread the color intensities uniformly, leaving the colors themselves (e.g., hues) unchanged. The HSI color space is ideally suited to this type of approach.

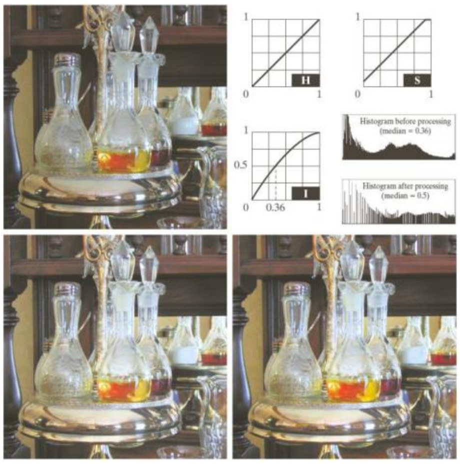


Fig. Histogram Equalization in the HSI Color Space

4.6. Color segmentation process

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

Segmentation in HSI Color Space:

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

Segmentation in RGB Vector Space:

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

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

Where the subscripts R, G, and B, denote the RGB components of vectors **a** and **z.** The locus of points such that $D(z, a) \leq D_o$ is a solid sphere of radius D_o. Points contained within or on the surface of the sphere satisfy the specified color criterion; points outside the sphere do not. Coding these two sets of points in the image with, say, black and white, produces a binary segmented image.

A useful generalization of previous equation is a distance measure of the form

$$D(z, a) = [(z-a)^{T} C^{-1} (z-a)]^{1/2}$$

Where C is the covariance matrix1 of the samples representative of the color to be segmented and the above equation represents an ellipse with color points such that $D(z, a) \le D_o$.

PREVIOUS QUESTIONS

1. Explain about RGB, CMY and CMYK color models?
2. What is Pseudocolor image processing? Explain.
3. Explain about color image smoothing and sharpening.
4. Explain about histogram processing of color images.
5. Explain the procedure of converting colors from RGB to HSI and HSI to RGB.
6. Discuss about noise in color images.
7. Explain about HSI colour model.
8. Consider the following RGB triplets. Convert each triplet to CMY and YIQ
   i) (1 1 0) ii) (1 1 1) iii). (1 0 1)
9. Explain in detail about how the color models are converted to each other.
10. Discuss about color quantization and explain about its various types.
11. What are color complements? How they are useful in image processing.
12. What is meant by Luminance, brightness, Radiance & trichromatic Coefficients.