Introduction:

Scene 3D $\xrightarrow{\text{Camera}}$ 2D Image ⟶ DIP (Computer system) → o/p Image

↓ Acquisition Sensor

↓ various preprocessing techniques

processing the image by using digital systems is called Digital Image processing.

Analog : noise image not clear image

Digital : 0's & 1's.

To convert Analog into Digital we use Sampling and quantization

varies in amplitude → Sampling.

Analog x → continuous (Infinite)

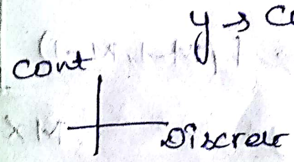
y → continuous

Sampling x → Discrete (Int)

y → continuous

Quantization x → discrete

y → discrete



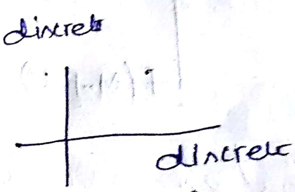
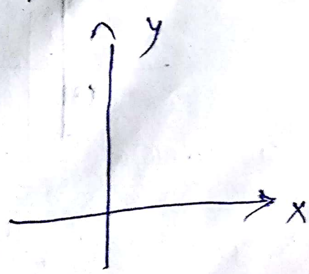


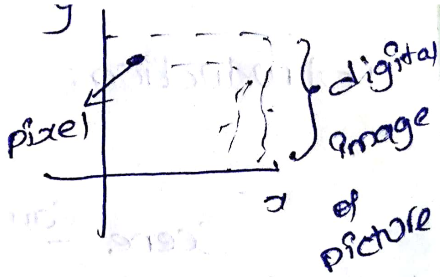
Image : 2 dimensional function f(x, y)

Where x,y are the Spatial of plane coordinate.

Light intensity function



Graylevel or intensity:



With respect to 2dimensional function the amplitude of f at any pair coordinates (x,y) is known as graylevel or intensity.

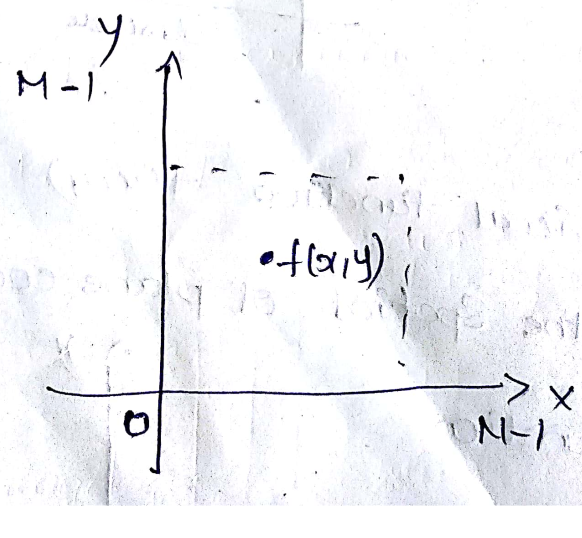
Digital image: when $f(x,y)$ and the amplitude levels of f are all finite, discrete quantities

Pixel: or pel / matrix element

A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as pixel or pel or image element.

Representation of digital image:

$$f(x,y) = \begin{bmatrix} f(0,0), & f(0,1) & \cdots & f(0,N-1) \\ f(1,0), & f(1,1) & \cdots & f(1,N-1) \\ f(2,0), & f(2,1) & \cdots & f(2,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}_{M\times N}$$



Advantages of DIP:

1. processing of image is faster and cost effective.
2. digital images can be effectively stored and efficiently transmitted from one place to another.
3. when shooting a digital image one can immediately see if the image is good or not.
4. Copying a digital image is easy.
5. whenever the image is in digital format, the reproduction of the image is both faster and cheaper.

Applications:

1. Remote Sensing:
It may be used for the purpose of monitoring the environmental conditions, traffic along the roads environmental & pollution control, natural resources survey and so on
2. Bio Medical: ECG (Electro cardio Graph), EEG, Cancer detection EMG analysis, X-ray image analysis etc.
3. Industrial Automation: Automatic Inspection System Non destructive testing, processes related to VLSI manufacturing, PCB (printed Circuit Board), Oil

and natural gas exploration, Robotics etc.

4. <u>Office Automation</u>: Optical character recognition, identification of address area and envelope, cursive script recognition etc.

5. <u>Criminology</u>: Fingerprints and footprint identification Human face recognition and matching

6. <u>Information technology</u>: (Fax (mail)), Image transmission
Fascimaile
video text, video conferencing, video phones etc.

7. <u>Astronomy & Space Application</u>: Recognisation and analysis of object containing from deep space probe machine.

8. <u>Military Applications</u>: Missile guidance and detections, Target identification, Navigation of pilotless vehicle.

Types of images:

1. Binary image or monochrome image 0's & 1's

2. Gray Scale Image for 8 bit image 0 to 255

Range of scale

0 ↓ black

1 ↓ white

3 colour Image

RGB 24 bit each 8 bit store

RGB → primary colors.

Fundamental Steps in DIP:

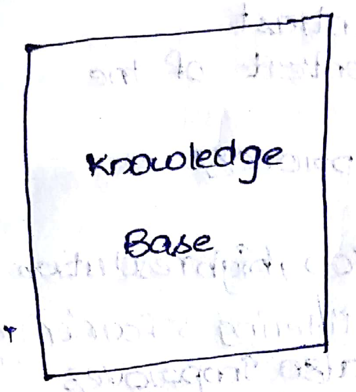
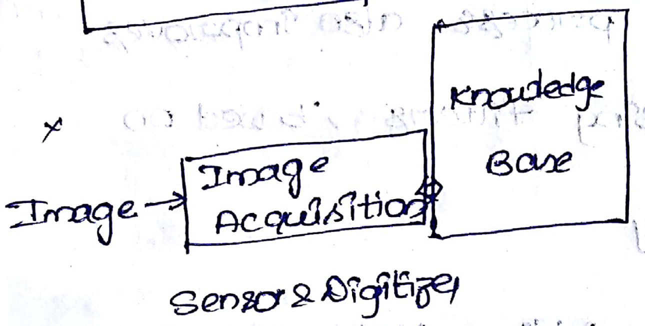
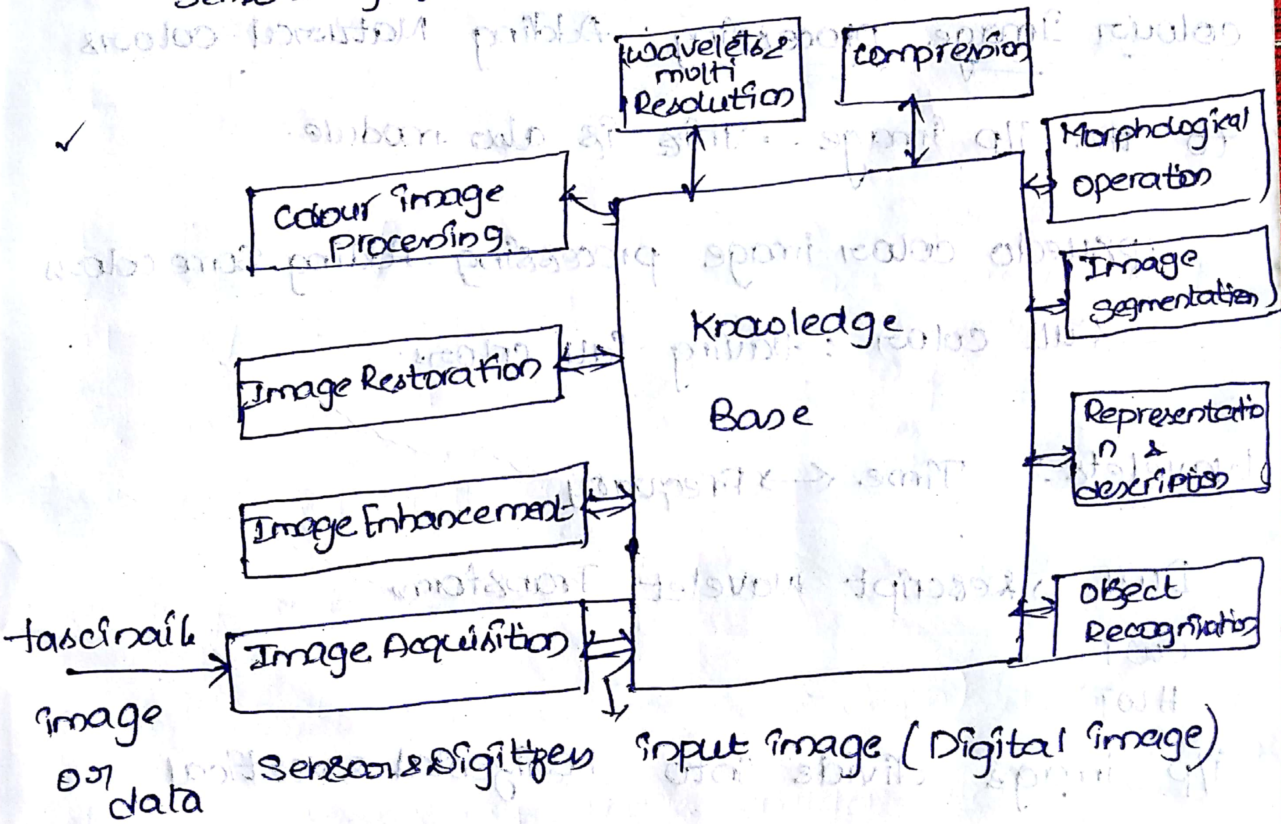


Image Acquisition having Sensor & Digitizers





All modules are interconnected To knowledge Base i.e data

sensor → illumination energy falls on (3D) object converted into electrical signal (2D) object

Digitizer Analog to digital signal.

Digital Image applied to knowledge Base

It recognises i/p then produce o/p image if not

recognised then goes to Image Enhancement

Image Enchancement → Improves the ~~Contest~~ contrast of the Image, by using filters, fine details

Image Restoration: Subjective process ⇒

based on Human Enhancement perception, high resolution

Spatial domain's time, Frequency domain filtering → Fourier Transform & F.T

Image Restoration: objective process. also Improves

quality of Image by using filters, based on

mathematical & probability

colour Image processing: Adding Natural colors

to the i/p Image. This is also module.

pseudo color image processing: Adding some colours

Full color: Having full colors.

Wavelets: Time ⟷ Frequency

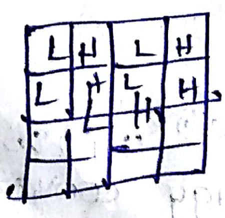
DWT → Descript wavelet Transform

CWT

HWT

i/p image divide into horizontal & vertical

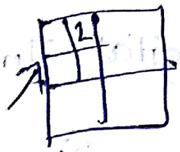
levels



Analysis | Synthesis

LPF along width decimator | HPF along width decimator

Decomposition → Decimators.



LPF low pass Filter applies to one level

compression : Reducing the data because we can reduce the Space to Save memory, Reduce Bandwidth, Speed

coding techniques → Transformcoding, Hamming code

Ex: lossless, loss compressions.

Morphological operation: concentrates on the shape deforming the shape structure of given image

Image Segmentation : i/p image divided into number of subimages

properties:

- Detection of Discontinuities → based on this property we can detect line, part & edge.
- Region Based Segmentation

↓

Tree algorithms regions splitting and merging

Representation & Description : 1. Boundary based representation mainly concentrates on external shape characteristics.

2. Region based representation :- Concentrates on the internal shape characteristics Ex: contrast, texture

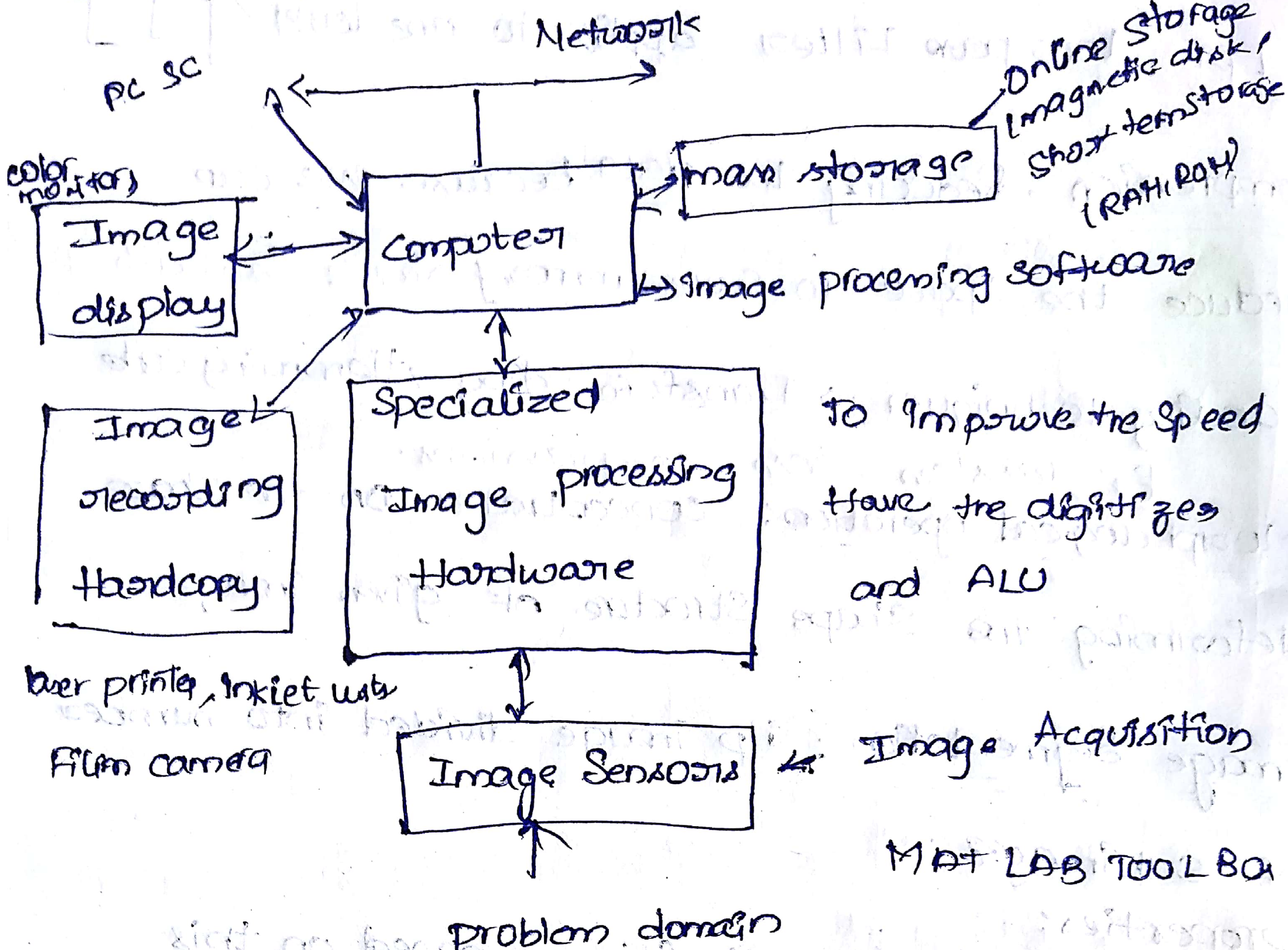
Description gives full details of data.

Object Recognition : A

↓

Assinging the label

Digital Image processing System: components:



online storage (Magnetic disk, optical media storage)

PC SC to achieve desire level the performance

Network : Inter connect to the other system

Computer : Fundamental steps

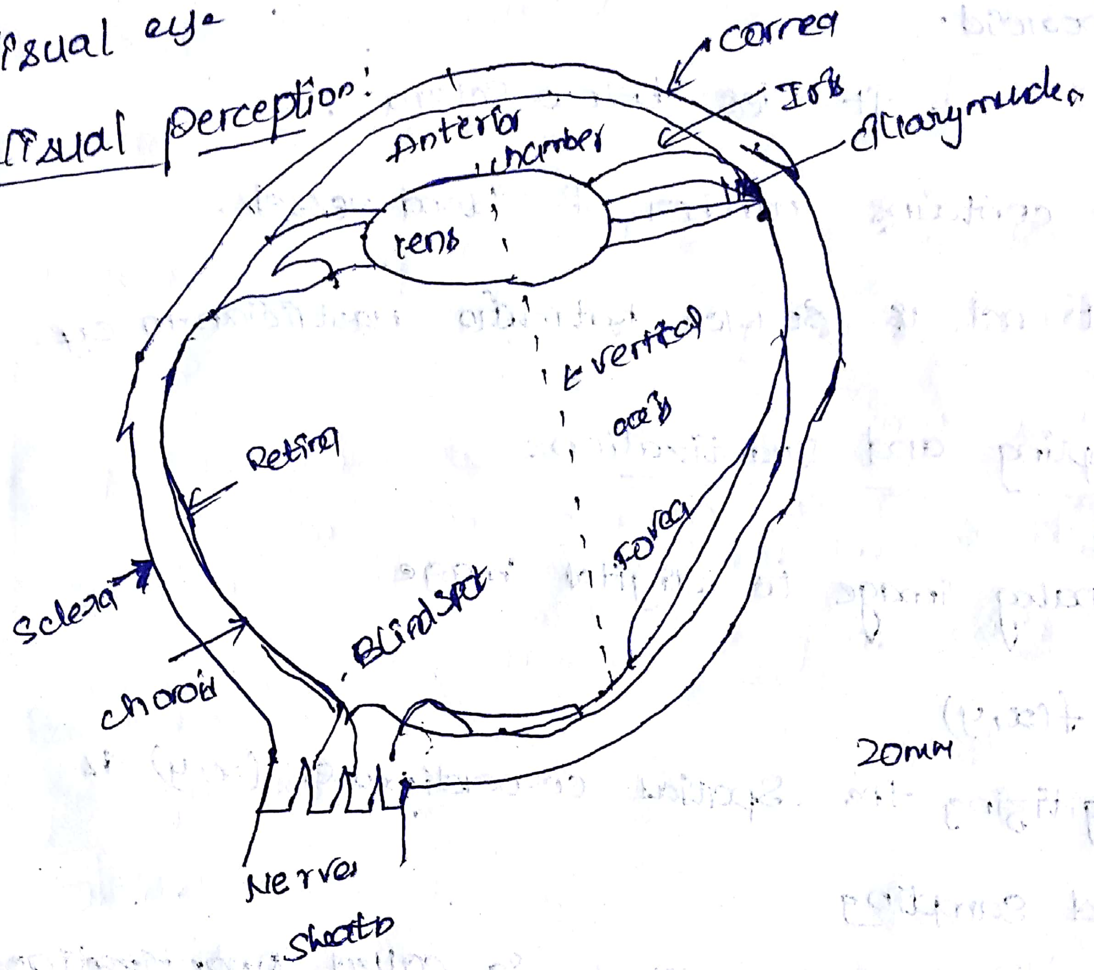
Image Sensor : Converts from Illumination energy from source into electrical signal.

Image Acquistion :- Converts 3D to 2D

digitizer : Analog to digital

Visual eye

Visual perception:



3. Retina: → cone & Rods → Doesnot depend on the colours

1. Innermost part of the eye ↓ center of Retina · It gives overall

2. Light from an object is imaged on retina fovea ↓ picture view

3. also called as Receptors Bright light vision · Dimlight vision

Iris: To protect ~~and~~ the eyelet which controls light intensity.

cornea: Transparent tissue.

· covers the Anterior surface of the eye.

Total eye divided into 3 parts,

2. Choroid:

1. Cornea & Sclera: 3. Retina

It covers outside

2. Choroid:

1. It lies below sclera

2. It contains number of blood vessels.

3. It act as source ~~nutrition~~ neutrician to eye.

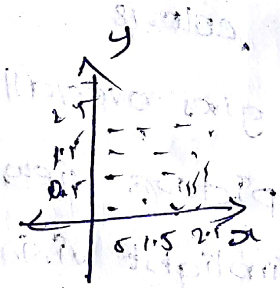
Sampling and Quantization:

Analog image to digital image

$f(x, y)$

Digitizing the spatial co-ordinates (x, y) is called sampling

Digitizing the amplitude is called Quantization.



i.e planes → Scale

digitizing (x, y) coordinate → Sampling

amplitude intensity value (gray level value)

Quantization

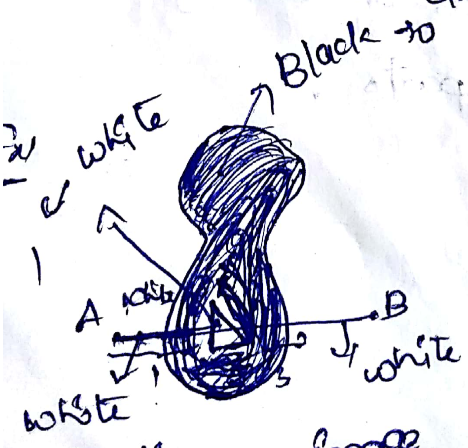
Analog → Sampling → Quantization → coding

8 bit image $[0, 1, \ldots 255]$

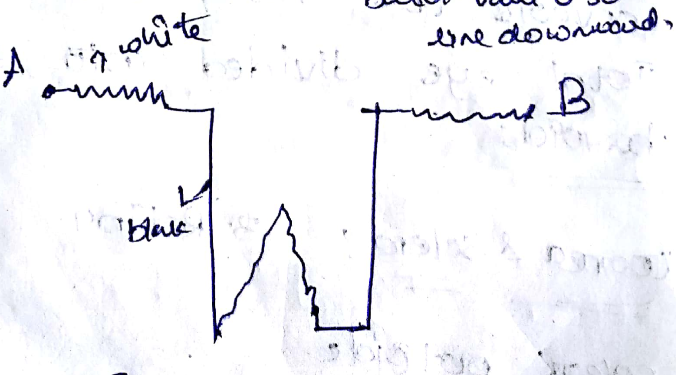
$2^8 = 256$

↓ amplitude value

Scan from A white black value 0 so line downward.

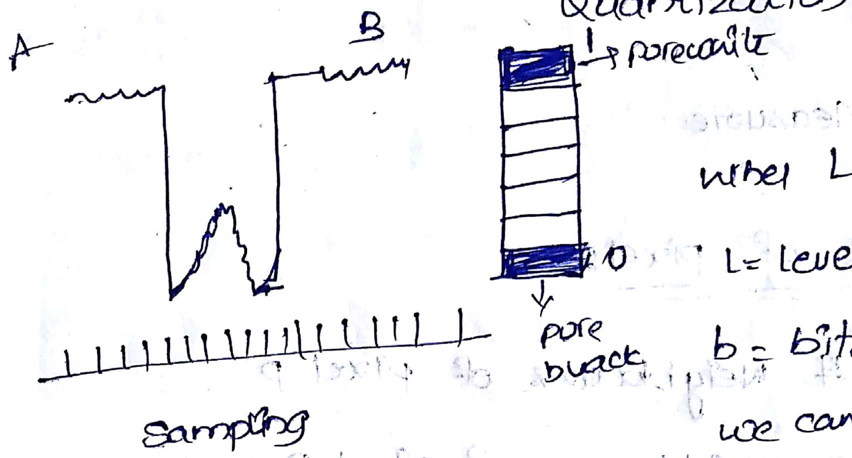


Continuous image of Analog



Gray level values of the continuous image along AB

Sampling with respect to x-axis



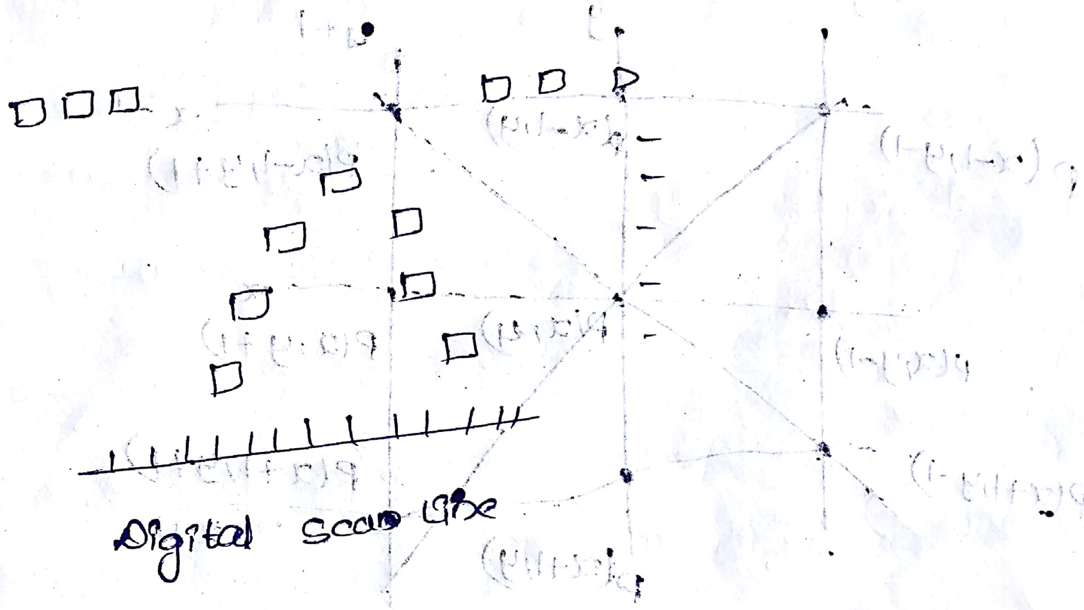
where $L = 2^b$

L = levels

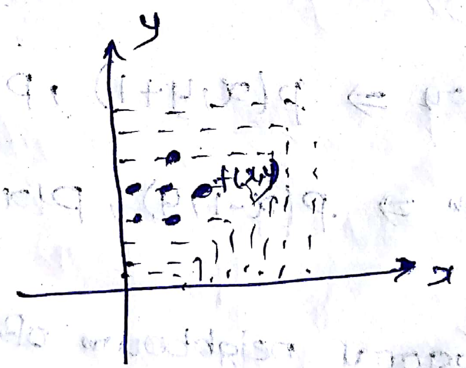
b = bits

we can store $m \times n \times b$.

Based on levels of colour converts into digital



Some Basic Relationship between pixel:



1. Neighbours of pixel
2. connectivity
3. Adjacency

4. Connected Region

5. Boundary

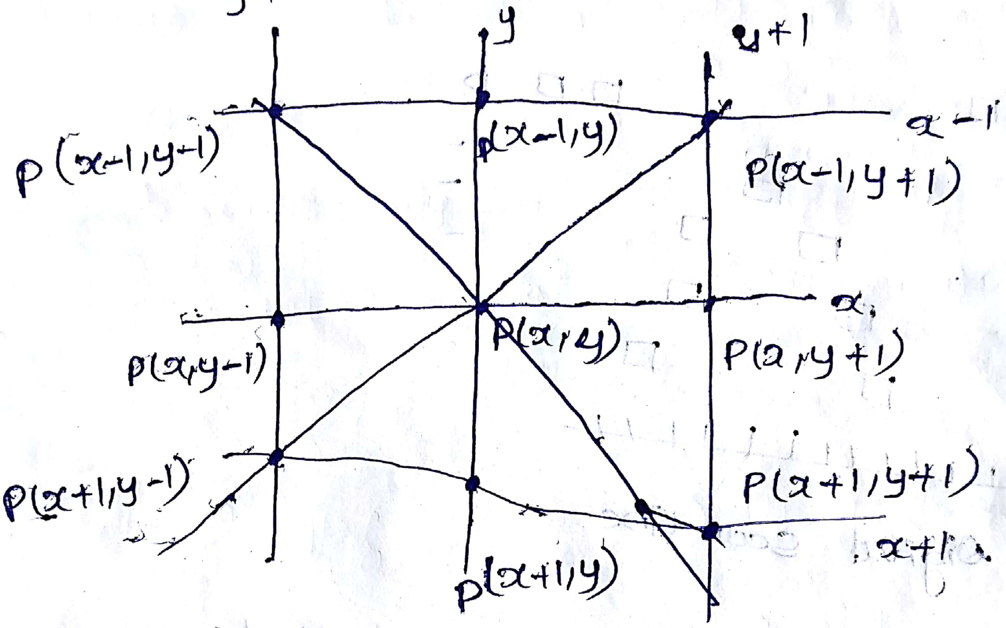
6. Distance Measure.

1. <u>Neighbours of pixel</u>:-

$N_4(P) \Rightarrow$ 4 Neighbours of pixel P

$N_8(P) \Rightarrow$ 8 Neighbours of pixel P

$N_D(P) \Rightarrow$ Diagonal Neighbours of pixel P.



consider pixel p(x,y) as origin.

1. $N_4(P) \Rightarrow$ Horizontal and vertical axis Neighbours

Horizontal Neighbours $\Rightarrow$ $p(x,y+1)$, $p(x,y-1)$

Vertical Neighbours $\Rightarrow$ $p(x-1,y)$, $p(x+1,y)$

2. $N_D(P) \Rightarrow$ ~~two~~ diagonal neighbours of pixel P.

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

$p(x+1,y+1)$

3. $N_8(P) \Rightarrow N_4(P) \cup N_D(P)$

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

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

$p(x+1, y+1)$

2. Connectivity:

1. Intensity values must be similar.

1. 4 connectivity
2. 8 connectivity
3. M-connectivity

Two pixels are said to be connected if they are adjacent in some sense $p \in q$ (or) $q \in p$. They are neighbours and their intensity values are similar.

Let 'v' be the set of gray levels used to define connectivity for two point $p, q \in v$

1. 4 Connectivity:

$p, q \in v$ and $p \in N_4(q)$

2. 8 connectivity:

$p, q \in v$ and $p \in N_8(q)$.

3. M-connectivity:

$p, q \in v$, M connected if $q \in N_4(P)$ and

$q \in N_D(P)$ and

$N_4(p) \cap N_4(q) = \phi$

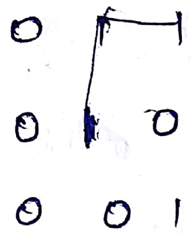
M-connectivity is

Modification ~~is~~ ~~of~~ 8-connectivity eliminate

multiple path connection that often arise

with 8-connectivity

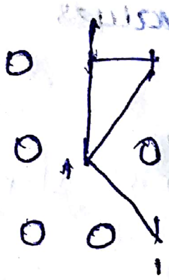
Ex: $V\{1\}$



Here I is intensity value similar

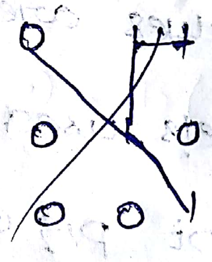
4-connectivity

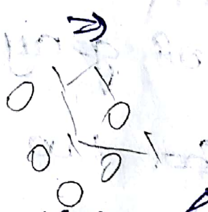
Horizontal & vertical



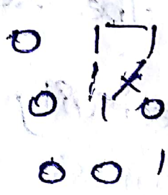
8-connectivity

Horizontal / vertical & diagonal.



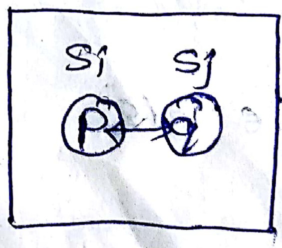


modifying 8 connectivity to avoid multiple paths



modifying M-connectivity

3. Adjacency:



→ Image

P, q are pixels

Depending on type of

1. 4 connectivity adjacency

2. 8 ~~connectivit~~ Adjacency

3. M-adjacency

1. Depending on type of connectivity used two pixels p and q are adjacent if they are connected.

2. Two image subset S_i and S_j are adjacent if $p \in S_i$ and $q \in S_j$ such that p and q are adjacent.

4. Connected Region:

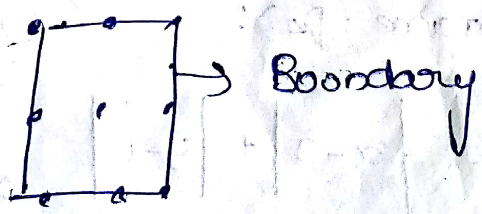
~~If~~ 1. two pixels are connected if they are adjacent in the common sense that they are neighbours their intensity value also similar.

2. Two regions to be adjacent ~~if~~ a pointing one region which is adjacent to the other region

i.e Adjacent region.

5. Boundary of Border of Contour:

1. Border of image is Boundary



2. Boundary is a set of pixels covering a region that has one or more neighbours outside the region.

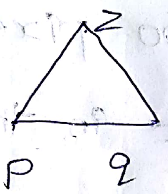
3. Boundary of the region defined as set of pixels in the first and last rows and columns of the image.

6. Distance Measure:

Let us consider three pixels $P(x, y)$, $q(s,t)$, $z(u,v)$ D is a distance function. $D(p,q) \geq 0$

$D(P,q) \geq 0$

1. $[D(P,q) = 0 \text{ if } P = q]$

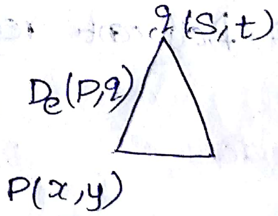


2. $D(P,q) = D(q,P)$

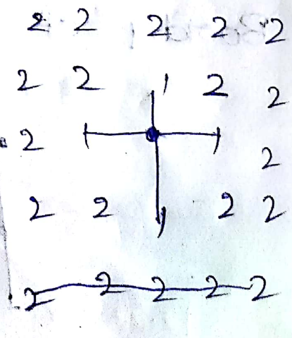
3. $D(P,z) \leq D(P,q) + D(q,z)$

1. Calculate distance between two pixels:

1. Euclidean Distance: Euclidean distance between pixels p and q.



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



2. City Block distance (D_4):

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

3. chess board distance (D_8):

$$D_8(p,q) = Max\left[\,|x-s|,\ |y-t|\,\right]$$

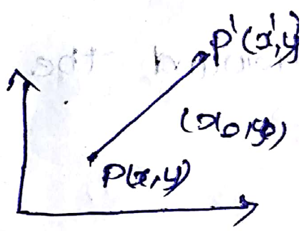
$$\begin{array}{ccccc} & 2 & 2 & 2 & \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & \bullet & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \end{array}$$

4. Quasi Ecludean Distance:

$$D_{qe}(p,q) = \begin{cases} |x-s| + (\sqrt{2}-1)\,|y-t| & \text{if } |x-s| > |y-t| \\ & \text{otherwise} \\ (\sqrt{2}-1)\,|x-s| + |y-t| & \end{cases}$$

Image Geometry:

1. Translation: If a point $p(x,y)$ is translated by a vector (x_0, y_0) to $p'(x', y')$



The new position $x' = x + x_0$

$$y' = y + y_0$$

This can be represented into matrix equation

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

In single Matrix form

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

Assymetric matrix.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{T} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$ Symmetric matrix.

consider V' Unified matrix. V

Unified Expression or unified matrix Representation.

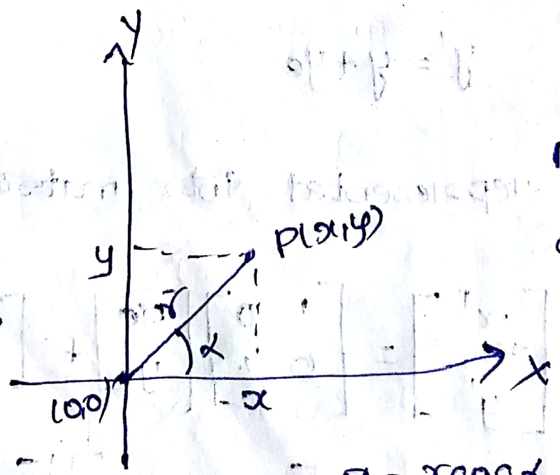
$$V' = T\ V$$

Here $V' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ $T = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$ $V = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

↓

Translation matrix.

2. Rotation: let $P(x,y)$ is rotated by an angle α around the clockwise direction.

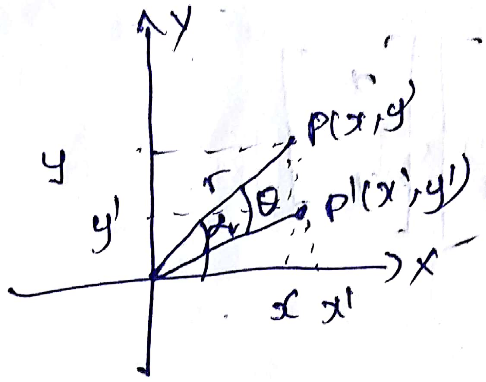


Rotate into clockwise direction

r = distance

$x = r\cos\alpha$ — (1)

$y = r\sin\alpha$ — (2)



$x = r\cos\alpha \qquad y = r\sin\alpha$

$x' = r\cos(\alpha - \theta)$

$y' = r\sin(\alpha - \theta)$

$x' = r\cos(\alpha - \theta) = r\cos\alpha\cos\theta + r\sin\alpha\sin\theta.$

from ① and ②

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

$y' = r\sin(\alpha - \theta) = r\sin\alpha\cos\theta - r\cos\alpha\sin\theta$

$= y\cos\theta - x\sin\theta$

$y' = y\cos\theta - x\sin\theta.$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

3. Scaling:

S_x Scaling in x-direction

S_y Scaling in y-direction

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

3D

Coordinate System → 3D cartesian coordinate (X, Y, Z)

(x, y) $\cancel{(X, Y, Z)}$ (x, y, z)

1. Translation: $x' = x + x_0$

$y' = y + y_0$

$z' = z + z_0$

Matrix form Representation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_e \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

A point $p(x, y, z)$ is translated to a new location coordinate system $p'(x', y', z')$ using a displacement vector (x_0, y_0, z_e)

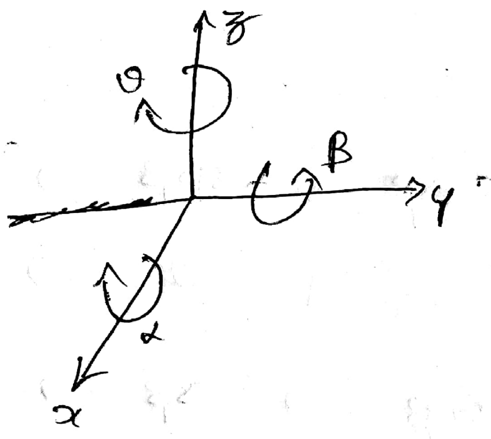
$$V' = T \cdot V$$

$$V' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$ ⇒ column vector Translated coordinates

$$T = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ → Transformation matrix.

$$V = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ → column vector of original coordinates.

2. Rotation:



1. Rotation of a point about the z-axis by an angle 'θ' achieved by using transformation

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

2. Rotation of a point about the X-axis by an angle 'α' is performed by using transformation α

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Rotation of a point about the Y-axis by an angle 'β'

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

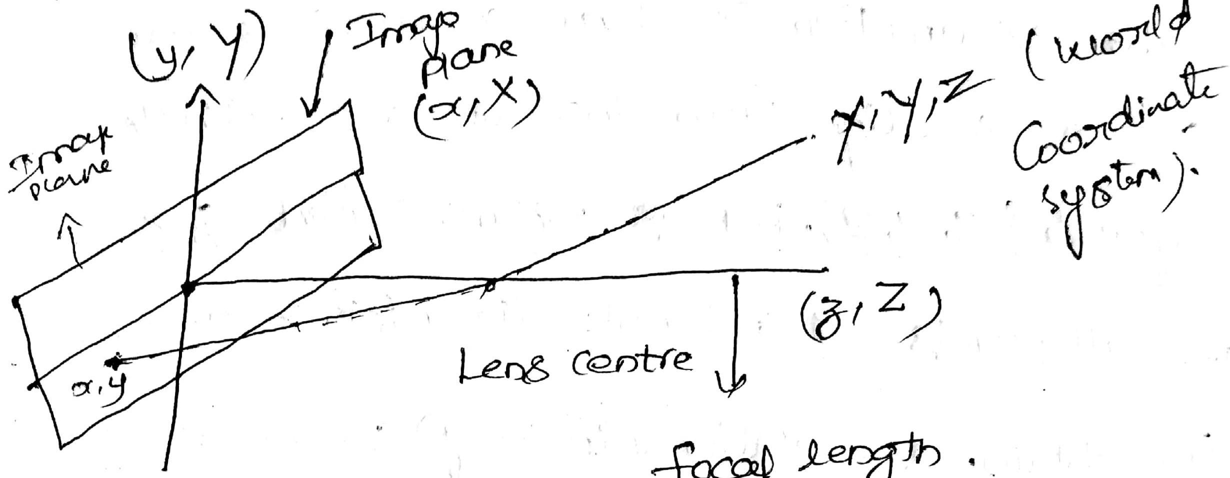
3. Scaling :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Image transformation also called as

perspective transformation:

Perspective Transformation (Image Transformation):



x, y, z → plane coordinates of camera-coordinates System

(0, 0, λ) → center of the lens

(X, Y, Z) → world Coordinate System align with the camera coordinate System.

Assume Z > λ

1. Camera Coordinate system (x, y, z) has the image plane coincide with the (x, y) plane and the optical axis established by the center of the lens along the z-axis.

2. The center of the image plane is at Origin and center of the lens is at coordinates

$(0,0,\lambda)$. If the camera is ~~focussed~~ in focus for distinct objects, λ is the focal length of the length.

3. The assumption is that camera coordinate system is align with the world coordinate system (X, Y, Z). Let us assume that $Z > \lambda$ i.e all points of interest lie infront of lens

4. To obtain a relationship (x, y) coordinates of the projection of the points (X, Y, Z) on to the image plan. This is accomplish by the use of similar triangle.

$$\frac{x}{\lambda} = -\frac{X}{Z-\lambda}$$

$$\frac{x}{\lambda} = \frac{X}{\lambda - Z} \quad \text{— } 1$$

$$\frac{y}{\lambda} = \frac{-Y}{Z-\lambda}$$

$$\frac{y}{\lambda} = \frac{Y}{\lambda - Z} \quad \text{— } 2$$

$$x = \frac{X\lambda}{\lambda - Z} \quad \text{— } 3 \qquad y = \frac{\lambda Y}{\lambda - Z} \quad \text{— } 4$$

(3) & (4) are non linear expression because they involve division by the variable

Z.

Above equation are non linear convert into linear matrix form this can be accomplished by using homogeneous coordinates.

Cartesian coordinate system in vector form

$$w = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

homogeneous world coordinate system

$$w_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix}$$

perspective transformation (P).

$$\boxed{c_h = P\, w_h}$$

c_h = Camera coordinate system

w_h = homogeneous world coordinate

Spon

$$C_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{bmatrix} \begin{bmatrix} kx \\ ky \\ kz \\ k \end{bmatrix}$$

$$C_h = (P) \quad \omega_h$$

C_h is converted into cartesian coordinate system.

$$C_h = \begin{bmatrix} kx \\ ky \\ kz \\ \frac{-kz}{\lambda} + k \end{bmatrix}$$

C_h = Homogenous camera coordinate system.

To convert into cartesian coordinate system

By dividing each of the first 3 components by 4th

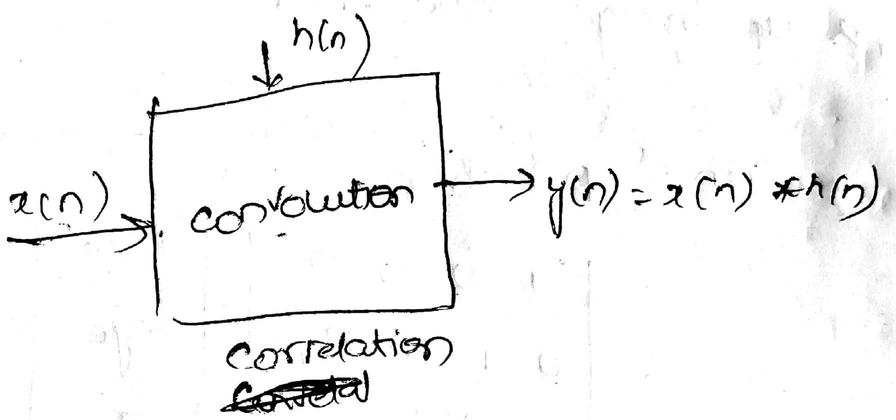
$$c = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} \frac{\lambda x}{\lambda - z} \\ \frac{\lambda y}{\lambda - z} \\ \frac{\lambda z}{\lambda - z} \end{bmatrix}$$

$$X = \frac{x_0}{\lambda}(x-z)$$

$$Y = \frac{y_0}{\lambda}(x-z)$$

Image Transforms:

1. Useful for fast computation of ~~cog~~ convolution and correlation



2. Spatial domain (time) $\xrightarrow{\text{Image Transform}}$ Frequency domain

Fourier transform

3. 1D

continuous time Fourier (CTFT) Transform

$$x(t) \xrightarrow{1D} \int_{-\infty}^{\infty} x(t) e^{-\omega t} dt = X(\Omega)$$

Discrete time Fourier transformation

$$x(n) \xrightarrow{1D} \sum_{n=0}^{N-1} x(n) e^{-j\omega n} = X(k)$$

Unit-II

2D

1D CTFT continuous line Fourier Transformation

$$CTFT\left[x(t)\right] = x(\Omega) = \int_{-\infty}^{\infty} x(t) e^{-T\Omega t} dt$$

1D DTFT Discrete Time Fourier Transformation:

$$DTFT\left[x(n)\right] = X(k) = \sum_{n=-\infty}^{\infty} x(n) e^{-T\omega t} dt$$

2D DTFT

1D

$$x(n) \xrightarrow{1D\ DFT} X(k)$$

or

$$DFT\left[x(n)\right] = X(k)$$

$$2D\ DFT\left[x(m,n)\right] = X(k,l)$$

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

** Square image is i/p

$$DFT\left[x(m,n)\right] = \bar{X}(k,l)$$

$$DFT\left[x(m,n)\right] = X(k,l)$$

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

or

$$x(m,n) \xleftrightarrow{2D\ DFT} X(k,l)$$

$k, l = 0$ to $N-1$

2D

Square Image → N×N

Rectangular image → M×N

IDFT

$$DFT\left[x(n)\right] = \sum_{n=-\infty}^{\infty} x(n) e^{-j\frac{2\pi}{N}nk}$$

$$= X(k) \quad k = 0 \text{ to } N-1$$

**

Inverse 2D-DFT

$$x(m,n) = IDFT\left[X(k,l)\right] = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l}^{N-1} X(k,l)\, e^{\frac{j2\pi}{N}mk}\, e^{\frac{j2\pi}{N}nl}$$

Input image is Rectangular image $(M \times N)$

$$2D\ DFT\left[x(m,n)\right] = X(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n)\, e^{-j\frac{2\pi}{M}nk}\, e^{-j\frac{2\pi}{N}nl}$$

$k,l = 0$ to $N-1$

$$IDFT\left[x(k,l)\right] = x(m,n) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(k,l)\, e^{\frac{j2\pi}{M}mk} \cdot e^{\frac{j2\pi}{N}nl}$$

where $m,n = 0$ to $N-1$

***

properties of 2D Discrete Fourier Transformation:

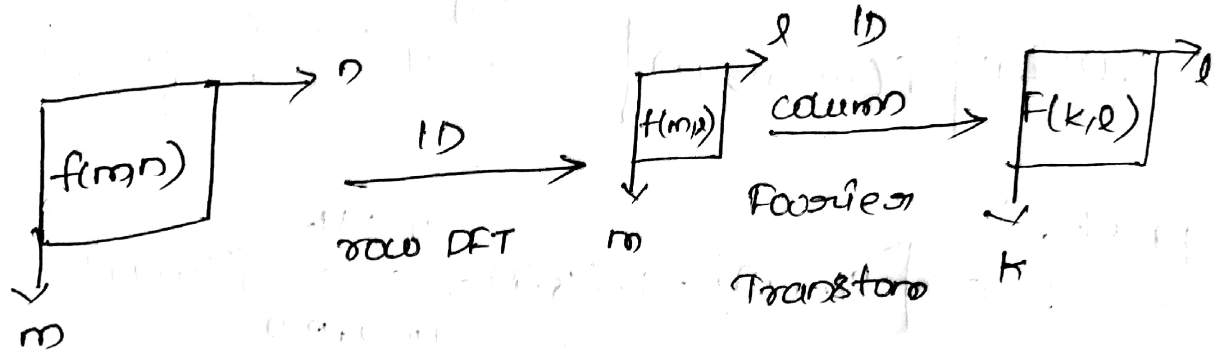
1. Separable property :-

performing a 2D discrete Fourier transform is equivalent to performing to 1D transform.

1. performing 1D transform of each row of image $f(m,n)$ to get $F(m,l)$

2. performing a 1D transform on each column of $F(m,l)$ to get $F(k,l)$

Proof



$$2D\ DFT\left[f(m,n)\right] = F(k,l)$$

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

$f(m,n) = x(m,n)$ = input image

$F(k,l)$ = output image.

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

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

$$= F(k,l)$$ $\left[\because \overset{1D}{x(n)} = X(k) = \sum_{n=0}^{\infty} x(n)\, e^{-j\frac{2\pi}{N}nk}\right.$ or $\displaystyle\sum_{n=0}^{N-1}$

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

2. Spatial Shifting property:

$$2D\,DFT\left[\underset{\Rightarrow\, i/p\ image}{f(m-m_0,n)}\right] = e^{-j\frac{2\pi}{N}m_0k}\cdot \underset{\Rightarrow\, o/p\ image}{F(K,l)}$$

proof.

$$2D\,DFT\left[f(m,n)\right] = F(K,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}ml}\cdot e^{-j\frac{2\pi}{N}nl} \quad K,l = 0 \text{ to } N-1$$

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

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

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

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

$$= F(K,l)\, e^{-j\frac{2\pi}{N}m_0k}$$

$= R.H.S$

Hence proved.

5. Correlation property:

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

correlation is basically used to find the relative similarity between two signals. The correlation of two sequences $x(n)$ and $h(n)$ is equivalent to performing the conclusion of one sequence with folded version of the other sequence.

proof: To prove this expression the DFT of two sequences $x(n)$ and $h(n)$ is defined as

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

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

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

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

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

$$= H(K) \cdot X(-K)$$

$\left[\because DFT[x(n)] = X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}nk}\right]$ 1D DFT

6. Multiplication by Exponential on Frequency Shifting

property:

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

proof:

we know

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

2D DFT

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

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

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

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

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

$$= F(k-k_0, l-l_0)$$

= R.H.S

7. Scaling property:

Scaling is basically used to increase or decrease the size of the Image

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

proof:

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

~~DFT~~ Let

$$f(m,n) = f(am, bn)$$

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

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(am, bn)\, e^{-j\frac{2\pi}{N}\frac{mK}{a}(a)} \cdot e^{-j\frac{2\pi}{N}\frac{nl}{b}(b)}$$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(am, bn)\, e^{-j\frac{2\pi}{N}ma\left(\frac{K}{a}\right)} \cdot e^{-j\frac{2\pi}{N}nb\left(\frac{l}{b}\right)}$$

$$IDDFT\left[F(an)\right] = \frac{1}{a} F\left[\frac{K}{a}\right]$$

$$\left[\text{i.e } f(an) = \frac{1}{a} F\left[\frac{K}{a}\right]\right]$$

$\Downarrow$ Seperable property.

$$= \frac{1}{|ab|} F\left(\frac{K}{a}, \frac{l}{b}\right)$$

= R.H.S.

8. Conjugate property:

$$DFT\left[f^*(m,n)\right] = F^*(-k,-l)$$

$$F[k,l] = F^*(-k,-l)$$

proof:

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

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

conjugate mean (-ve) value changes to (+ve) value

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

$$\cancel{= F(k,l)}$$

$$= F[k,l]$$

$$= R.H.S$$

Hence proved.

9. orthogonality property:

Both signals are independent.

$$\frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{k,l}(m,n)\, a^*_{k',l'}(m,n) = \delta(k-k', l-l')$$

$\delta \rightarrow$ Impulse signal

Here $\delta(k-k', l-l')$ is the Kronecker Delta.

10. Rotation property: states that if a function is rotated by the angle, its fourier transform also rotated by an equal amount.

$$f(m,n) \xrightarrow{\text{Rotation}} F(r\cos\theta ; r\sin\theta)$$

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

$$DFT\left[f(r\cos(\theta+\theta_0), r\sin(\theta+\theta_0))\right] \longrightarrow F\left[R\cos(\varphi+\varphi_0), R\sin(\varphi+\varphi_0)\right]$$

Fast Fourier Transform:

Formula of DFT.

$$DFT\left[x(n)\right] = X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}nk} \qquad k = 0 \text{ to } N-1.$$

$$x(0) + x(1)\, e^{-j\frac{2\pi}{N}1k} + x(2)\, e^{-j\frac{2\pi}{N}2k} \cdots + x(N-1)\, e^{-j\frac{2\pi}{N}(N-1)k}$$

DFT $\rightarrow$ Direct computation.

Direct calculation of discrete fourier transform for one value of k

- N complex multiplications

· N-1 complex addition are required,

· N-1 value of K.

· N2 complex multiplication required,

· N(N-1) complex Addition are required.

or $N^2 - N$

Fast Fourier transform is to calculate

N-point DFT.

convert time to frequency domain,

viceversa also possible.

following props.

DIT DIF

Decimation Decimation Frequency

Time

N = 8

4 4

2 2 2 2

I/p image having

time divided into

no of Subgroups

2. In Fast Fourier transformation the number of complex multiplication is

$$m(n) = \frac{1}{2} 2^n \log_2 2^n$$

Here $2^n = N$

$n = \log_2 N$

$= \frac{1}{2} N \log_2 N$

$$\boxed{m(n) = \frac{1}{2} N \cdot n} \quad n \geq 1$$

3. Additions

$$A(n) = 2^n \log_2 2^n$$

$$= N \log_2 N$$

$$\boxed{A(n) = N \cdot n} \quad n \geq 1$$

Decomposition procedure for fast fourier Transformation:

$$DFT\left[f(x)\right] = F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)\, \omega_N^{ux}$$

$x \to$ time

$u \to K$ here ω_N = Twiddle factor $= e^{-j\frac{2\pi}{N}}$ or

exponential $\exp\left[-j\frac{2\pi}{N}\right]$

Here $N = 2^n$ n = positive integer.

N can be written or expressed as

$$N = 2M$$

Where M = positive integer

$$DFT\left[f(x)\right] = \frac{1}{2M}\sum_{x=0}^{2M-1} f(x)\, \omega_{2M}^{ux}$$

f(x) should be separated even & odd term.

Let $x = 2x$ for even

$x = 2x+1$ for odd

$$DFT\left[f(x)\right] = \frac{1}{2}\left[\underbrace{\frac{1}{M}\sum_{x=0}^{M-1} f(2x)\, \omega_{2M}^{u2x}}_{\text{even}} + \underbrace{\frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)\, \omega_{2M}^{u(2x+1)}}_{\text{odd}}\right]$$

from properties of twiddle factor

$$\omega_{2M}^{u2x} = \omega_{M}^{ux}$$

$$= \frac{1}{2}\left[\frac{1}{M}\sum_{x=0}^{M-1} f(2x)\, \omega_{M}^{ux} + \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)\, \omega_{M}^{ux} \cdot \omega_{2M}^{u}\right]$$

Define $F_{even}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x)\, \omega_M^{ux}$

$$F_{odd}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1)\, \omega_M^{ux} \cancel{\omega_{2M}}$$

$u = 0, 1, \ldots, M-1$

$$\boxed{\therefore F(u) = \frac{1}{2}\left[F_{even}(u) + F_{odd}(u) \cdot \omega_{2M}^{u} \right]}$$

3. periodicity property:

Function $f(m,n)$ is said to be periodic

with period N.

If $F(K,l) = F(K+pN, l+qN)$

1D $x(n) = x(n+N)$ for all value of 'n'

$N \to$ fundamental period.

proof: $F(K,l) = F(K+pN, l+qN)$

$$\text{2D DFT}\left[f(m-m_0,n)\right] = F(K,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}mk}\, e^{-j\frac{2\pi}{N}nl}$$

$K, l = 0$ to

Let $F(K,l) = F(K+pN, l+qN)$

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

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

$\because e^{-j2\pi} = \cos 2\pi - j\sin 2\pi$

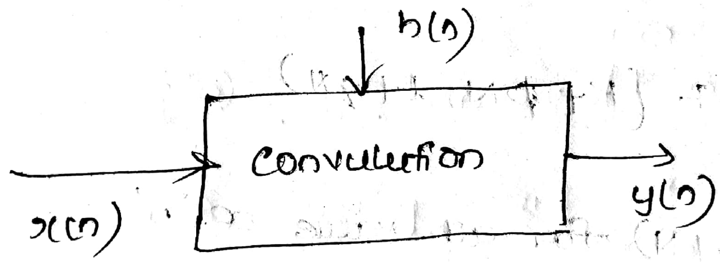
$= F(k,l)$

& p, q, m, n are

$= R.H.S$

Hence proved

4. Convolution property:



or

$f(n)$

$= x(n) * h(m)$

1D

$$y(n) = x(n) \circledast h(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k)$$

2D Convulution Response

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

~~Walsh function are real and they tak~~

2D statement:

$$DFT\left[f(m,n) \circledast g(m,n)\right] = F(k,l) \cdot G(k,l).$$

proof

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

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

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

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

$$= \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} f(a,b)\, e^{-j\frac{2\pi}{N}ak} \cdot e^{-j\frac{2\pi}{N}bl}.$$

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

$$= F(k,l) \cdot G(k,l)$$

proved

Convolution theorem states that the convolution of the function in spacial time domain corresponds to multiplication in frequency domain.

Walsh Transform

Walsh function are real and they take only two values +1 or -1

- 1D Walsh function Transform basis can be given by the following equation

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

$N = 2^n$, n = Time index = 0 to N-1

N = order of the Matrix

K = Frequency index = 0 to N-1

$M = \log_2 N$ = The no. of bits to represent a number.

$b_i(n)$ = the ith bit (LSB) bit of the binary value of n decimal number represented in binary from

1. Find the 1D Walsh basis for the fourth order system (N=4)

N=4, n = 0 to N-1

= 0 to 3

= 0,1,2,3

$M = \log_2 4 = \log_2 2^2 = 2\log_2 2 = 2.$

$K = 0 \text{ to } 3 = 0,1,2,3$

$i = 0 \text{ to } M-1 = 0 \text{ to } 1 = 0,1$

when $n=0, K=0$, $g(0,0) = \frac{1}{N} = \frac{1}{4}$

construction of walsh basis for $N=4$

K(or) n	b_1	b_0	$b_1(n)$	$b_0(n)$	$b_1(K)$	$b_0(K)$
0	0	0	$b_1(0) = 0$	$b_0(0) = 0$	0	0
1	0	1	$b_1(1) = 0$	$b_0(1) = 1$	0	1
2	1	0	$b_1(2) = 1$	$b_0(2) = 0$	1	0
3	1	1	$b_1(3) = 1$	$b_0(3) = 1$	1	1

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

$g(1,0) = \frac{1}{4}$

$$\begin{array}{cc} & \begin{array}{cccc} n\to & 0 & 1 & 2 & 3 \end{array} \\ g(n,k) = \begin{array}{c} K\downarrow \\ 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & -1/4 & -1/4 \\ 1/4 & -1/4 & 1/4 & -1/4 \\ 1/4 & -1/4 & -1/4 & 1/4 \end{bmatrix}_{4\times4} \end{array}$$

$g(1,2) = \frac{-1}{4}$

$$g(n,k) = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Sequency

0 for 1st row because no sign change

1   1 to -1 one sign change

3   1 to -1 → 1   -1 to 1 → 2   1 → -1 → 3

2

~~2D~~ WT

2D Walsh Transforms:

$$F(k,l) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n) \times (-1)^{\sum_{i=0}^{p-1} b_i(m) \cdot b_{p-1-i}(k) + b_i(n) \cdot b_{p-1-i}(l)}$$

Hadamard Transform:

Hadamard Transform are either +1 or -1. Hadamard Transform elements are Mutually orthogonal basis vectors.

→ ~~Ha~~ Less complexity in Calculation of

Transform coefficient.

→ 1D

DFT[f(n)]

$$= H(K) = \frac{1}{N}\sum_{n=0}^{N-1} f(n)\,(-1)^{\sum_{i=0}^{P-1} b_i(n)\, b_i(K)}$$

$$f(n) = IDFT\left[H(K)\right] = \sum_{K=0}^{N-1} H(K)\,(-1)^{\sum_{i=0}^{P-1} b_i(n)\, b_i(K)}$$

2D

$$H(K,l) = \frac{1}{N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\,(-1)^{\sum_{i=0}^{P-1} b_i(m).\, b_i(K) + b_i(n).\, b_i(l)}$$

The order $N=2$,

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

Kronecker product Recursion

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

If $N=2$,

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

If $N=4$

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

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & +1 & -1 & +1 \\ 1 & 1 & -1 & -1 & -1 & -1 & +1 & +1 \\ 1 & -1 & -1 & 1 & -1 & +1 & +1 & -1 \end{bmatrix}_{8\times 8}$$

Sequency

0

7

3

1

6

2

5

properties of Hadamard Transform:

1. Hadamard Transform is real, symmetric and orthogonal.

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

2. Hadamard Transform is a fast transform

3. The natural order of the Hadamard transform coefficient turns out to be equal to the bit reverse to the gray code representation of its sequency

4. Hadamard transform has good to very good energy comparison for highly correlated images.

Applications

- Data compression, filtering and design of course

Drawback:

- Easy to stimulate but difficult to analyse.

Discrete cosine Transform:

- Discrete cosine Transform consists of a set of basis vectors that are samped cosine functions.
- It is a real function

1D

$$X(K) = \alpha(K) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{(2n+1)\pi K}{2N}\right)$$

$K = 0$ to $N-1$

Here

$$\alpha(K) = \sqrt{\frac{1}{N}} \quad K=0$$

$$\sqrt{\frac{2}{N}} \quad K\neq 0$$

IDCT

$$x(n) = IDCT\left[X(K)\right]$$

$$= \alpha(K) \sum_{K=0}^{N-1} X(K)\cos\left[\frac{(2n+1)\pi K}{2N}\right]$$

$n=0$ to $N-1$

1. Compute the DCT matrix for N=4.

Sol $$X(K) = \alpha(K) \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)\pi K}{2N}\right]$$

$K=0$ to $N-1$

$$\alpha(K) = \begin{cases} \sqrt{\frac{1}{N}} & K=0 \\ \sqrt{\frac{2}{N}} & K\neq 0. \end{cases}$$

$K = 0$ to $N-1$

$= 0$ to 3

$K = 0, 1, 2, 3$

$k = 0, 1, 2, 3$

let $k = 0$,

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{4}} & k=0 \\ \sqrt{\frac{1}{2}} & k \neq 0 \end{cases}$$

$$X(0) = \alpha(0) \sum_{n=0}^{3} x(n) \cdot 1$$

$$= \frac{1}{2} \left[x(0) + x(1) + x(2) + x(3) \right]$$

$k = 1$,

$$X(1) = \alpha(1) \sum_{n=0}^{3} x(n) \cos \left[\frac{(2n+1)\pi}{2 \times 4} \right]$$

$$= \frac{1}{\sqrt{2}} \cdot \left[x(0) \cos \frac{\pi}{8} + x(1) \cos \frac{3\pi}{8} + x(2) \cos \frac{5\pi}{8} + x(3) \cos \frac{7\pi}{8} \right]$$

$k = 2$

$$X(2) = \alpha(2) \sum_{n=0}^{3} x(n) \cos \left[\frac{(2n+1)\pi(\cancel{2})}{\cancel{2} \times 4} \right]_0 \quad \frac{\pi}{4}$$

$$= \frac{1}{\sqrt{2}} \left[x(0) \cos \frac{\pi}{4} + x(1) \cos \frac{3\pi}{4} + x(2) \cos \frac{5\pi}{4} + x(3) \cos \frac{7\pi}{4} \right]$$

$k = 3$

$$X(3) = \alpha(3) \sum_{n=0}^{3} x(n) \cos \left[\frac{(2n+1)\pi \cdot 3}{2 \times 4} \right] \quad \begin{matrix} 2 \times 3 \\ 6+1 \end{matrix}$$

$$= \frac{1}{\sqrt{2}} \left[x(0) \cos \frac{3\pi}{8} + x(1) \cos \frac{9\pi}{8} + x(2) \cos \frac{15\pi}{8} + x(3) \cos \frac{21\pi}{8} \right]$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.27 & -0.27 & -0.6 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

2DDCT:

$$F(k,l) = \alpha(k)\,\alpha(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \cos\frac{(2m+1)\pi k}{2N} \cdot \cos\frac{(2n+1)\pi l}{2N}$$

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k=0 \\ \sqrt{\frac{2}{N}} & k \neq 0 \end{cases} \qquad \alpha(l) = \begin{cases} \frac{1}{\sqrt{N}} & l=0 \\ \sqrt{\frac{2}{N}} & l \neq 0. \end{cases}$$

properties DCT

1. CT is real and orthogonal.
2. It is a fast transform
3. Excellent energy compaction for Highly correlated Data.

4. The NXN cosine Transform is very close to the KL Transform (Hotelling Transform).

5. Separable property.

Unit-III

Image Enhancement

- First step of the DIP. system.

Input → Enhancement → output Image

- Improves the quality of the image
- Enhancement is subjective process
- Subject image is modify the image to image of person to person

Image Enhancement

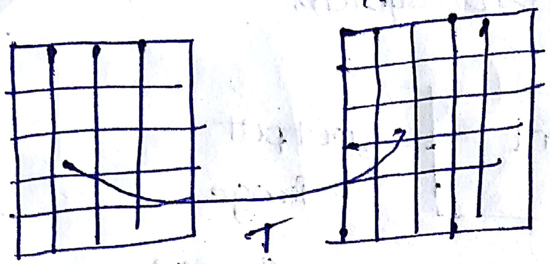
- Spatial domain method
  - ⇓ Analysing the image in terms of time (Direct method)
- Frequency domain Method. (Indirect method)
  - = Frequency
  - DFT & IDFT

Spatial domain

point processing | Neighbourhood processing

point processing:

I/P | O/P



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

$T \Rightarrow$ operator of Transformation

$T = 1 \times 1$

- Each pixel is modified by using this equation

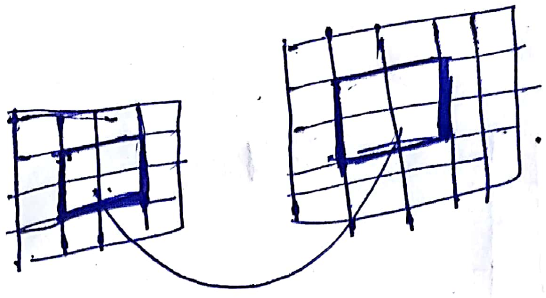
$$g(x,y) = T[f(x,y)]$$

$f(x,y) \Rightarrow$ i/p image

$g(x,y) \Rightarrow$ Enhanced of o/p image

Neighbourhood processing:

- Neighbour of the pixels considering



T = 3X3 or 5X5 or 7X7 i/p image

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

Neighbourhood operation apply on the i/p image.

group of neighbourhood of pixels

Example of point processing:

1. Image Negative
2. Contrast Stretching
3. Thresholding
4. Gray level Slicing
5. Bit plane slicing
6. Dynamic Range compression of Log Transformation.

1. Image Negative:

Ex in opposite color of

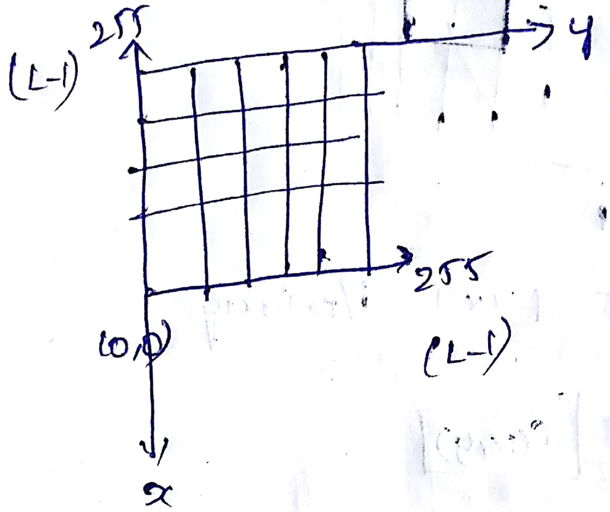
Input Image

images will appear.

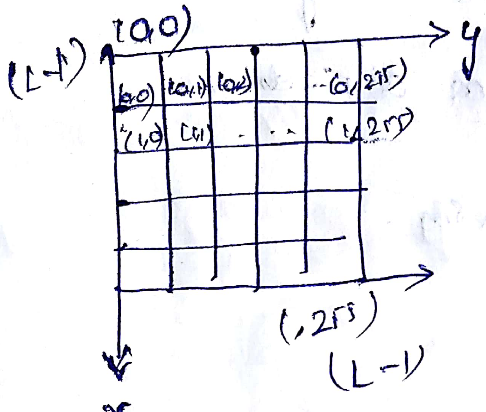
Grayscale Image

(0 to 255)

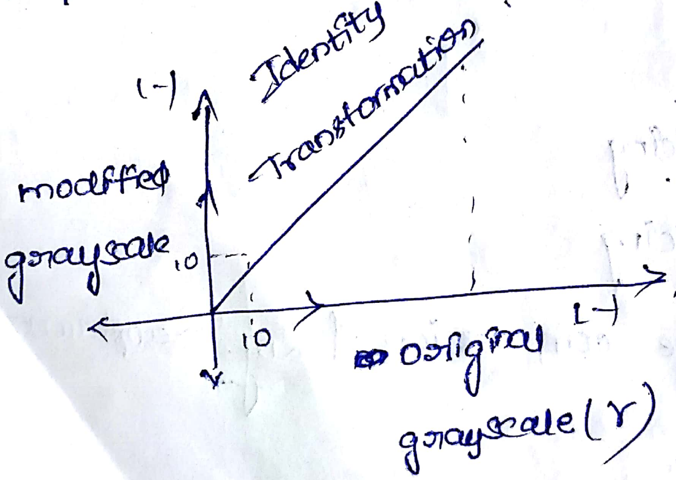
- Dimension: 255 x 255



or



Ideal Transformation of Ideal case:



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

⇒ O/P follows the i/p.

or

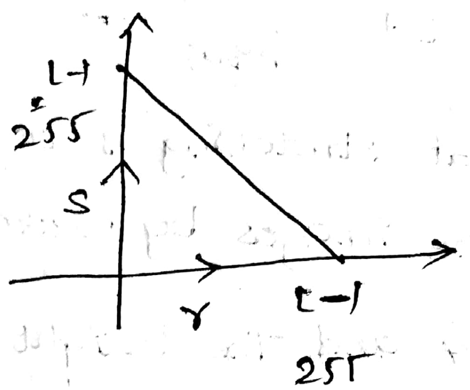
$$s = T[r]$$

or

$$g(m,n) = T[f(m,n)]$$

Ex:

X-rays



$$s = (L-1) - r$$

or

$$\boxed{s = 255 - r}$$

If $r = 255$ then $s = 0$ black

$r = 0$ then 255 white

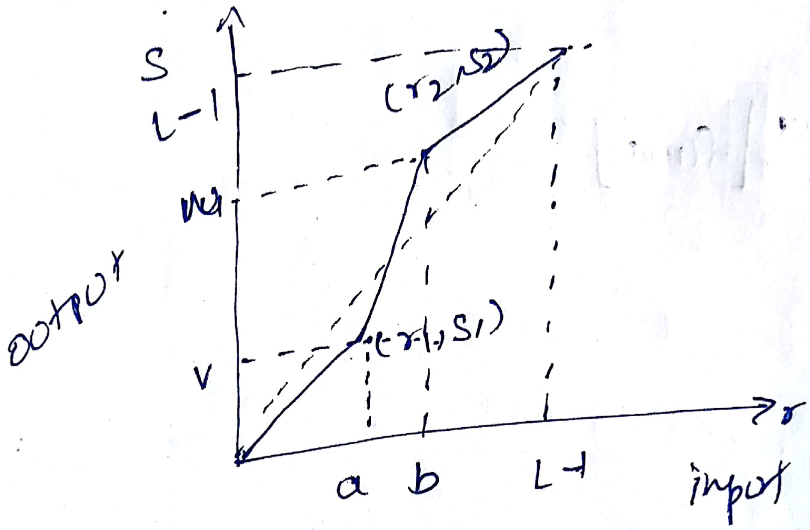
0 ... 255 Shaded colors.

↓ ↓

black white

i/p range increases output gray value increases i.e Image Negative.

2. Constrat stretching :



1. A idea behind constrat stretching is to increase the constrat of the images by making the dark portions darker and the bright portion brighter.

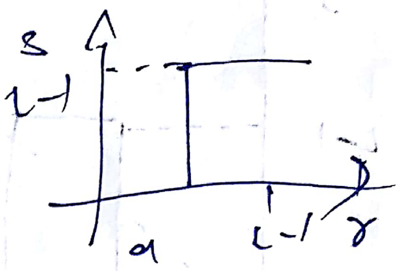
2. The dark gray levels darker by assigning a slope <1 and make the bright gray levels brighter by assigning a slope >1.

3. The formulation of the constrat stretching algorithm is given as

$$s = lr \quad 0 \le r < a$$
$$= m(r-a)+v \quad a \le r \le b$$
$$= n(r-b)+w \quad b \le r < L-1$$

where l, m, n are slopes from the figure l and n slopes are less than 1 m slope 71

$$s = \begin{cases} L-1 & r > a \\ 0 & r < a \end{cases}$$



Threshold

1. Extreme constrat stretching yields a thresholding.
If observe the constrat stretching diagram closely, that if the first and last row are made zero, and the centre slope increased, we get a threshold transformation.

2. To achieve thresholding $s = \begin{cases} L-1 & r > a \\ 0 & r < a \end{cases}$

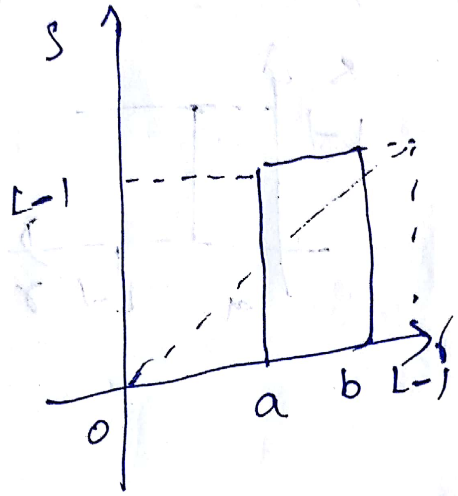
3. Threshold image has the maximum contrast as it has only black and white gray levels.

Graylevel Slicing: / Intensity Slicing

1. It is highlight specific range of gray levels.

2 types

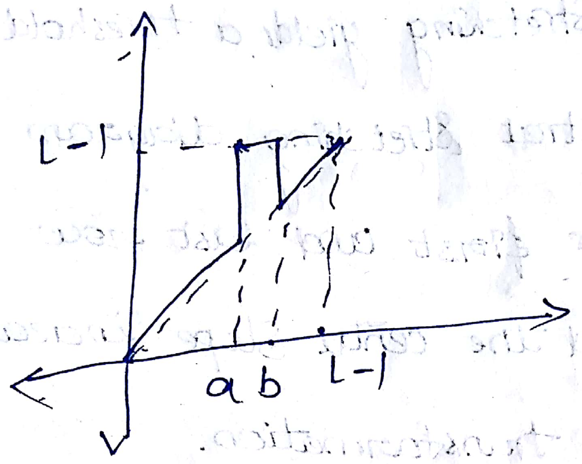
1. Graylevel slicing without background.
2. " " with "



$a \le r \le b$

otherwise 0

$$S = \begin{cases} L-1 & a \le r \le b \\ 0 & \text{otherwise} \end{cases}$$



$$S = \begin{cases} L-1 & a \le r \le b \\ r & \text{otherwise} \end{cases}$$

5. Bit plane slicing:

$256 \times 256 \times 8$

Any image is defined as a $256 \times 256 \times 8$ image.

In this 256×256 is the number of pixels presenting in the image.

8 → indicates no. of bits required to represent each pixel.

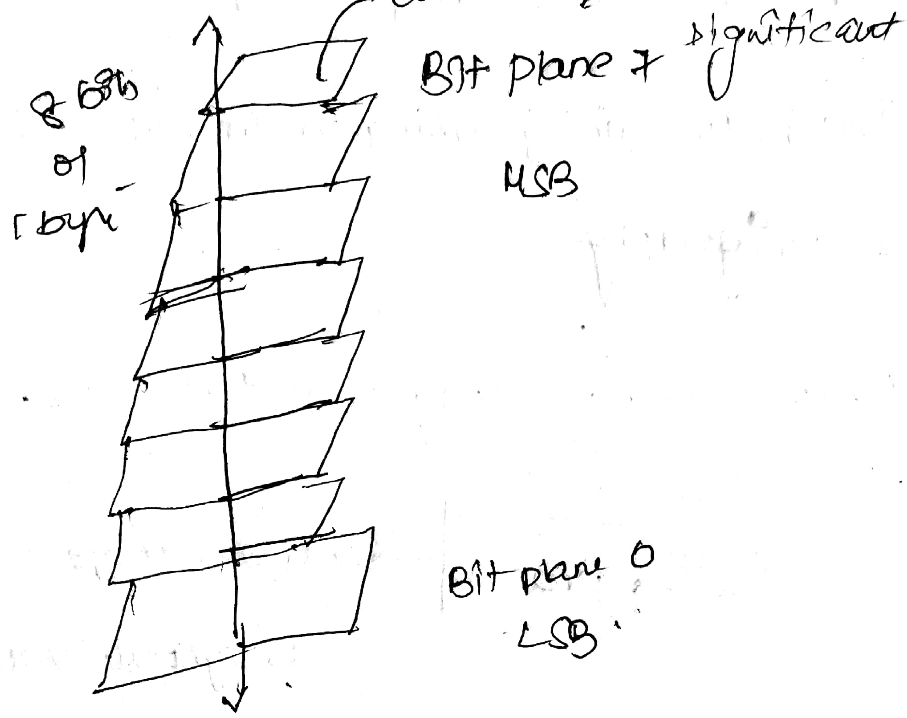
MSB

0 → 0 0 0 0 0 0 0 0 LSB

255 → 1 1 1 1 1 1 1 1 LSB

MSB

considering ~~MSB~~ MSB rather than LSB



In an 8-bit image 0 encoded as

0 → 0 0 0 0 0 0 0 0

and 255 encoded as

255 → 1 1 1 1 1 1 1 1

any number between 0-255 encoded as 1 byte.

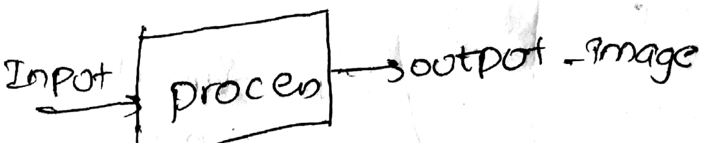
2. The bit in the far left side is referred as the Most Significant bit because a change in the bit would significantly change the value encoded by the bit.

3. The bit in the far right side is referred as the LSB because a change in the bit doesnot change the encoded graylevel value much.

Bit plane transmit only the higher order bit and remove the lower order bit.

3. Bit plane slicing using in image compression & Steriography.

6. Dynamic Range Compression of log Transformation.



brightest values.

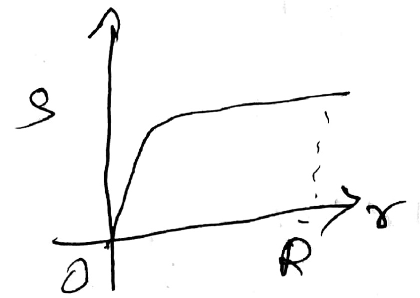
lowest values

$$g(m,n) = c \log_2 \left(1 + \left[f(m,n)\right]\right)$$

or

$$s = c \log_2 (1 + |r|)$$

- Sometimes the dynamic range of a processed image far exceeds the capability of the displayed device, in which only the brighter parts of the image visible on the display screen low parts of image are not visible. To spreadout the lower gray level (low parts of the image) Using dynamic range compression.



Neighbourhood processing:

Second type of spatial domain method is

Neighbourhood processing.

Group of pixels is input image

i.e 3x3, 5x5, 7x7

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

↓

Neighbourhood operator

3x3, 5x5

↓

Mask or window or Template

Mask operator:

w01	w02	w03
w04	w05	w06
w07	w08	w09

3x3 Mask

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

3x3 Neighbourhood.

Input image multiplying with mask.

$$g(x,y) = f(x-1,y-1) \times \omega_1 +$$

$$f(x-1,y) \times \omega_2 +$$

$$f \cdots \cdots +$$

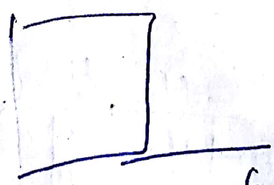
$$f(x+1,y+1) \times \omega_9$$

Image filtering operation can be corrected by using neighbourhood processing

over a region It is considered to have a low frequencies. If the gray level change very rapidly, that region considered to have High frequencies

In the most of the image the background considered to the low frequency region whereas the edges considered to be high region.

Low pass filtering or Average filtering removes (blurring) the edges. while High pass filter removing background



fc f

The standard low pass filter mask (3x3)

Each element of the Mask is the average value

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

3x3

- Sum of the coefficient value is 9

All coefficients are same (positive)

$$\frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

5x5

1. Box filter

$$\frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

3x3.

2. weighted mask filter

$$\frac{1}{6} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Q9 $\frac{1}{10} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ or $\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Low pass median filter (Non Linear Filter):-

1. Median filter is used in to remove the salt and pepper noise and impulse noise
2. Replace the value of a pixel by the median of the intensity values in the neighbourhood of that pixels.

Ex: Compute the median value of the marked pixel shown in figure.

$$\begin{bmatrix} 1 & 5 & 7 \\ 2 & 4 & 6 \\ 3 & 2 & 1 \end{bmatrix}$$
3x3

1. First arrange ascending or descending order.

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

2. discard 1st and last values.

3 is the median value.

$$\begin{bmatrix} & 3 & \end{bmatrix}$$

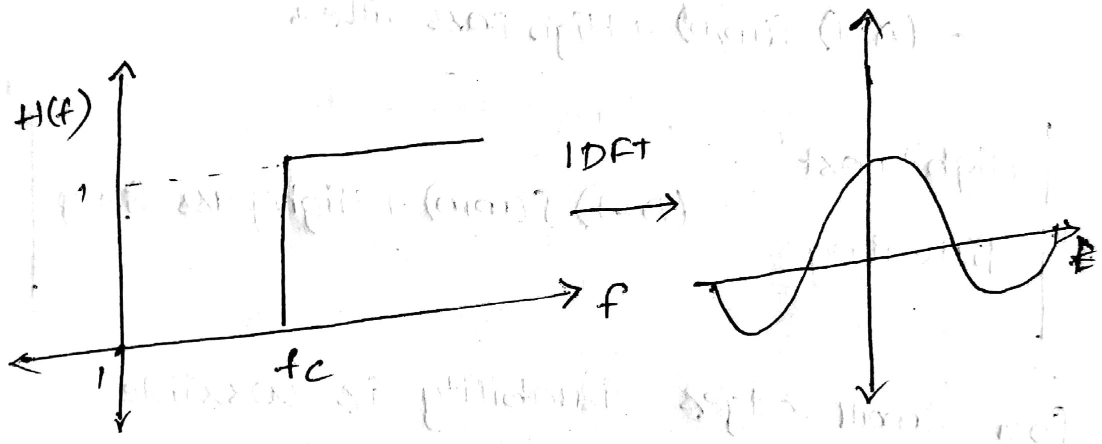
High pass filtering: or sharpening filter:

It allows high frequency signal

- Not allows the low frequency
- Highlights the fine details in the image.

⇓

Edge of Sharp



$|H(f)| = 1$ $f_c > f$ of   Spatial Domain.

0 otherwise $H(k,l)$

↓ $= 1 - H_{LPF}(k,l)$

Transfer function.

$$\rightarrow \frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

3×3 mask

- Sum of the coefficients value are zero.
- Middle coefficient value positive as well as high value
- Boundaries values are Negative

High Boost filtering (or) High Frequency emphasis

Filtering:

High Boost filtering is defined as

$A \times f(m,n)$ - Low pass filtering

A → Amplification factor.

= constant

= $(A-1)\,f(m,n) + f(m,n)$ - Low pass filter

$f(m,n)$ - low pass filter = High pass filter.

= $(A-1)\,f(m,n)$ + High pass filter.

High Boost filtering	= $(A-1)\,f(m,n)$ + High pass filter

for small edges visuability is possible

Frequency domain:

$$f(m,n) \circledast h(m,n) \xrightarrow{\text{2D-DFT}} F(k,l) \cdot H(k,l)$$

↓ Input image

↓ Filter kernel

Convolution in time Domain

$F(k,l)$ → Spectrum of the input image

$H(k,l)$ → Spectrum of the filter kernel.

Already proved in convulution property

$$2D\ DFT\left[f(m,n) \circledast h(m,n)\right] = F(k,l) \cdot H(k,l)$$

Low pass filtering (or) Smoothing Filtering:

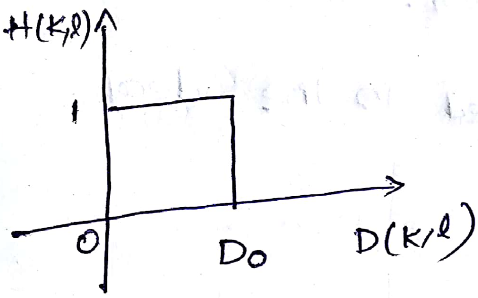
reduces the edges, noise Smoothing image

2 types

1. Non-Separable filtering
2. Separable filtering

Non-Separable filtering

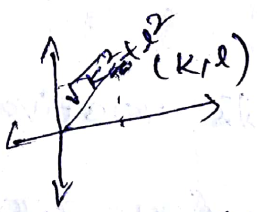
1\.



Do → cutoff frequency

1 → gain value

$$H(k,l) = \begin{cases} 1 & \sqrt{k^2+l^2} \le Do \\ 0 & \text{otherwise} \end{cases}$$



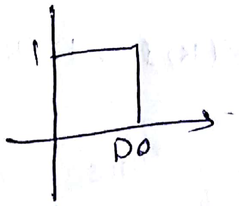
$$D(k,l) = \sqrt{k^2+l^2}$$

Allows the low-frequency

Separable filtering

1\. $H(k,l) = \begin{cases} 1 & k \le D_k \text{ or } l \le D_l \\ 0 & \text{otherwise} \end{cases}$

Butterworth LPF:



practical approximate to the low pass filter.

$$\cancel{H(K,l)} = \frac{1}{1 + \cancel{\left[\frac{\sqrt{K^2+l^2}}{Do} \right]}^{2n}}$$

n → order of the filter.

. Butterworth LPF approximated to the ideal LPF.

$$H(K, l) = \frac{1}{1 + \left(\frac{Do}{\sqrt{K^2+l^2}} \right)^{2n}}$$

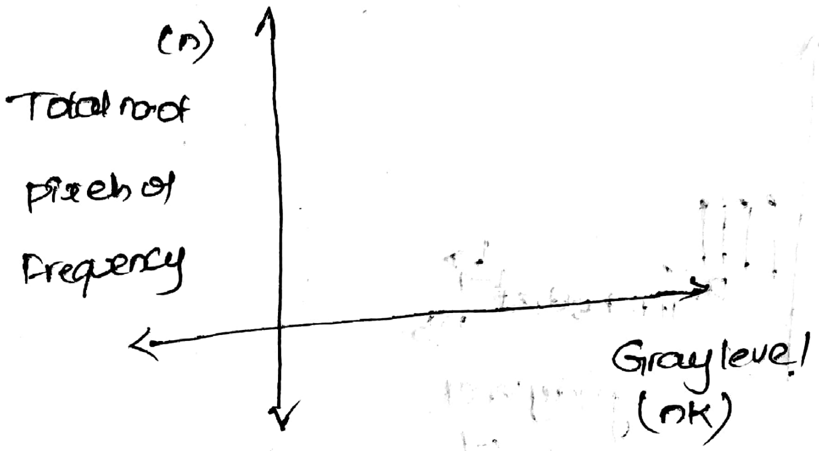
n = order of the filter.

Histogram:

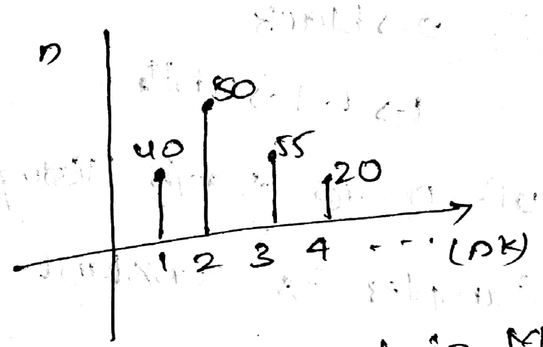
1. Histogram of an image represents relative frequency of occurance of the various gray levels in an image.

2. probability occurance of various image in an image also called as Histogram

1. 2 graphs

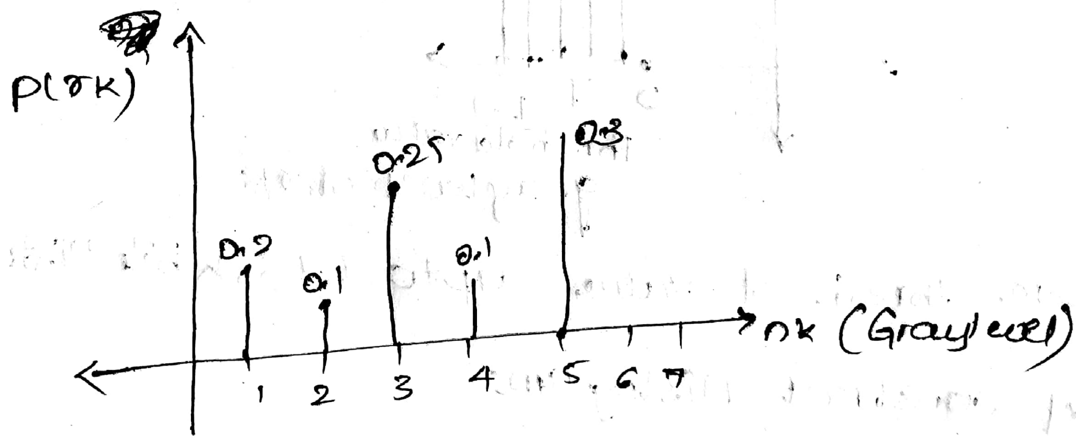


Ex:



this is the command used in MAT-LAB.

2. $p(r_K) = \frac{n_K}{n}$



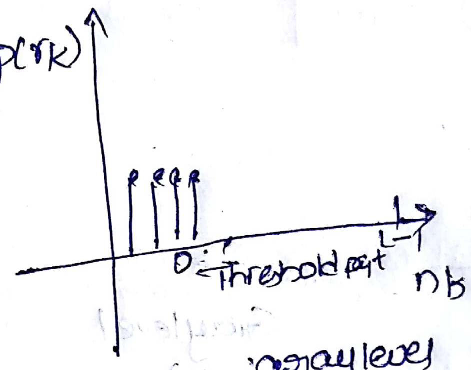
n_K = no. of pixels in the kth gray level.

n = total number of pixels.

r_K → kth gray level.

Typical Histogram:

Dark Histogram:



$K = L-1 = 255$

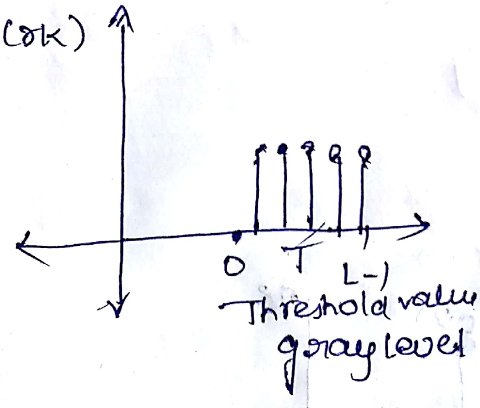
0 → black

1 → L-1 ⇒ white.

below Threshold of .0 is Dark Histogram.

Distance between samples is constant.

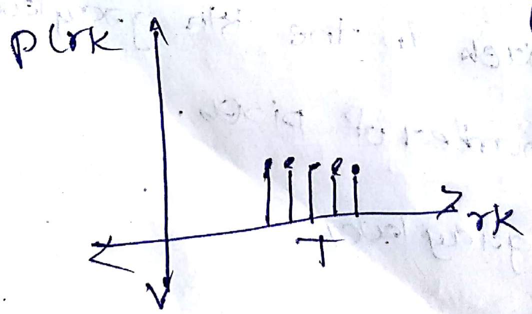
White Histogram:



Above threshold value upto L-1 is white Histogram

Low constract Histogram:

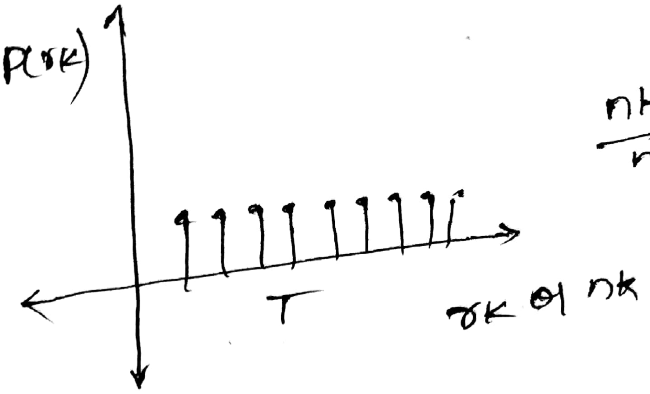
(narrow range)



edge middle

cover only the narrow

High Contrast Histogram:



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

Spreads into all range.

Histogram Equilization:

Assignment - I

1. Explain Digital Image processing System with block diagram.

2. Discuss about the relation between pixels.

3. State & prove any 3 properties of 2D DFT.