Unit-5 Image Registration & Segmentation

Registration

Introduction, Geometric Transformation - Plane to plane transformation, Mapping, Stereo imaging - Algorithm to establish correspondance, Algorithm to recover depth.

Segmentation

Introduction, Region extraction, Pixel based approach, Multilevel thresholding, Local thresholding, Region based approach, Edge & Line detection: Edge detection, Edge operators, Pattern fitting Approach, Edge linking & Edge following, Edge elements extraction by Thresholding, Edge detector performance, Line detection, Corner Detection

---

Image registration :-

⇒ Image registration is the process of overlaying two or more images of the same scene taken at different times or at the same time by different sensors.

⇒ It is an essential step where the final information is gained from combining various data.

Application :- ① In medical imaging
Computer Tomography (C.T.) & P.E.T. (Positron Emission Technology)

② Remote sensing imaging

③ Image fusion

Scanned by CamScanner

Image registration basically involves finding a correspondence ie matching of identical shapes in the related image pair.

This matching is achieved using geometric transformation

Geometric transformation:

Pixel by pixel comparison b/w two images of the same object acquired using different sensors

A few geometric transformation are-

① Euclidian transformation:

It is either translation, rotation or reflection

Translation:

$(x, y) \xrightarrow{\text{translated}} (x', y')$

$x' = x + h$

$y' = y + k$

This relationship can be expressed as in matrix form

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

similarly

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -h \\ 0 & 1 & -k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Scanned by CamScanner

Rotation :-

$(x, y) \xrightarrow[\text{about origin}]{\text{Rotated by an angle } \theta} (x', y')$

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

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

Origin of the image is usually the top left corner To rotate about the centre, we need to do a transformation of the origin to the centre of image.

Translation & Rotation can be implemented using a single equation

Rotation followed by Translation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & h \\ -\sin\theta & \cos\theta & k \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & -h\cos\theta - k\sin\theta \\ -\sin\theta & \cos\theta & h\sin\theta - k\cos\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Translation followed by Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & h\cos\theta - k\sin\theta \\ \sin\theta & \cos\theta & h\sin\theta + k\cos\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scanned by CamScanner

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & -h \\ -\sin\theta & \cos\theta & -k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

In general the Eucledian transformation can be written as-

$$T = \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Affine Transformation :-

In Eucledian transformation, length & orientation preserved but in affine transformation lines transform to lines but circles become ellipses

Affine transformation is the combine effect of translation, rotation, scaling & shear

Scaling : An object is said to be scaled if it gets multiplied by constant c_1.

Scaling transformation shrink or stretch an object which perform changes in length & angle.

$$x' = xc_1$$

$$y' = yc_2$$

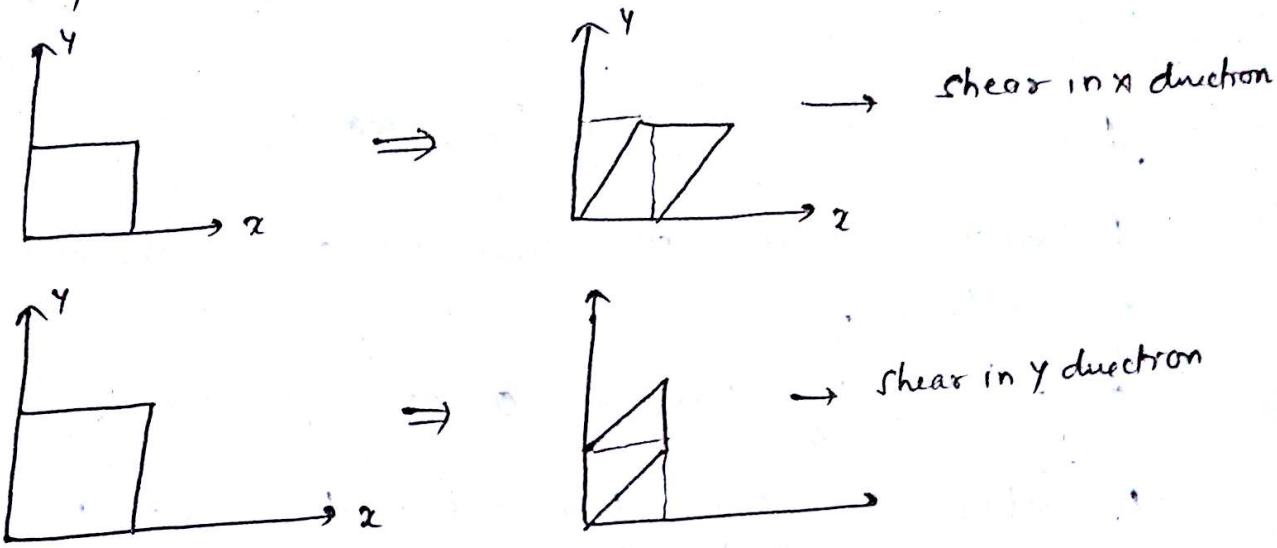
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1/c_1 & 0 & 0 \\ 0 & 1/c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Scanned by CamScanner

Shear :

Shear transformation has an effect of pulling or stretching an object in the direction parallel to the coordinate axis.



x direction

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & s_1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

similarly $$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -s_1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

$x' = x + s_1 y$, $y' = y$ (in x direction)

$y' = s_2 x + y$, $x' = x$ (in y direction)

y direction :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ s_2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -s_2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Scanned by CamScanner

In general Affine transformation are written in the form

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Perspective Transformation (Projection Transform)

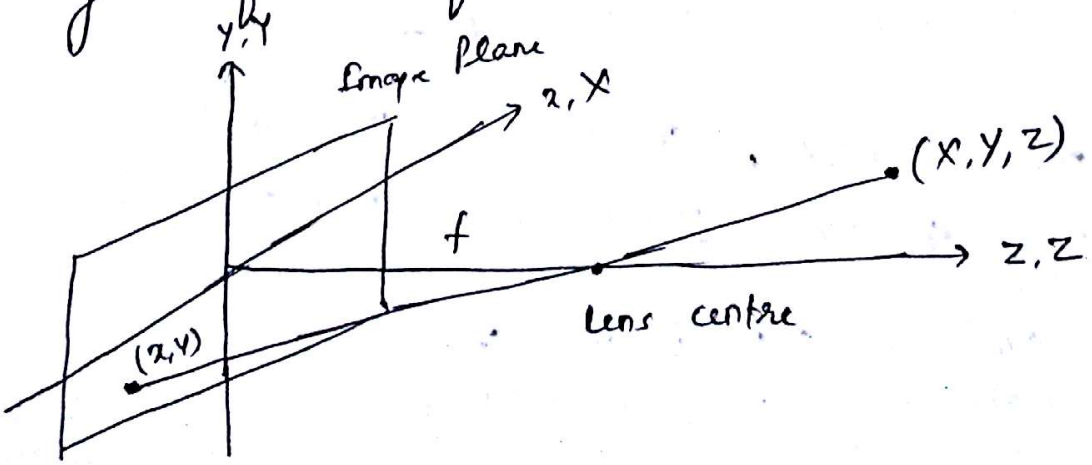
Projection is a term used when we change the dimension of something.

Ex: Projecting a 3D scene on to a 2D monitor.

Perspective is something that refers to the way an objects size get affected by distance.

Perspective projection transform is ref responsible for making objects near the camera appear bigger than object that are far away

It provide an approximation to the manner in which an image is formed from a 3D scene



Scanned by CamScanner

⇒ Here the pt (X, Y, Z) from the real world mapped as a pt (x, y) on the 2D image plane.

⇒ The camera coordinate system is represented by (x, y, z) while the real world coordinates are represented by (X, Y, Z)

⇒ The optical axis which is obtained by from the centre of the lens is along z axis

⇒ The coordinate of the centre of image plane is $(0, 0, f)$. where f is the focal length of the camera.

We assume that camera & real world coordinate are aligned. Now we need to find the relationship b/w (x, y) & (X, Y, Z).

$$\frac{x}{f} = \frac{X}{f - z} \quad \text{(By similarity of triangle)} \quad -(1)$$

here we assume that $z > f$ i.e all points to be mapped are in front of camera.

$$\text{similarly } \frac{y}{f} = \frac{Y}{f - z} \quad -(2)$$

from eqn (1) & eqn (2)

$$x = \frac{Xf}{f - z} \quad -(3)$$

$$y = \frac{Yf}{f - z} \quad -(4)$$

Let (aX, aY, aZ, a) be the homogeneous coordinates of the real world point (a is any non zero constant) A point in cartesian world is expressed in vector form

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

Scanned by CamScanner

and its homogeneous counter part is

$$W_h = \begin{bmatrix} aX \\ aY \\ aZ \\ a \end{bmatrix}$$

The perspective transformation matrix is defined as

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix}.$$

to get the transformed values we use

$$C = P \cdot W_h$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \begin{bmatrix} aX \\ aY \\ aZ \\ a \end{bmatrix}$$

$$C = \begin{bmatrix} aX \\ aY \\ aZ \\ \frac{-aZ}{f} + a \end{bmatrix}$$

These are the camera coordinate in homogeneous form. They can be converted into the cartesian coordinate simply by dividing the first three components of C by the 4th component of C.

for the 1st coordinate

$$\frac{aX}{\frac{-aZ}{f} + a} = \frac{aX}{\frac{-aZ + af}{f}} = \left(\frac{Xf}{f - Z}\right)$$

Scanned by CamScanner

$$C = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{Xf}{f-z} \\ \frac{Yf}{f-z} \\ \frac{zf}{f-z} \end{bmatrix}$$

3rd component is of no interest as we are mapping a 3D point onto a 2D plane (xy).

The inverse perspective transformation maps an image back to the 3D space.

$$\omega_h = P^{-1} C_h$$

let $(ax, ay, 0, a)$ be the homogeneous vector form. since point on the image plane is 2D, z component is 0.

$$\therefore \quad C_h = \begin{bmatrix} ax \\ ay \\ 0 \\ a \end{bmatrix}$$

$$\omega_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix} \begin{bmatrix} ax \\ ay \\ 0 \\ a \end{bmatrix} \Rightarrow \begin{bmatrix} ax \\ ay \\ 0 \\ a \end{bmatrix}$$

For cartesian coordinate system we devide by the 4th component

$$W = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

This simply means that when 2D point is mapped onto a 3D scene, the z component is zero

Scanned by CamScanner

we need to do something if we need to recover the 3rd dimension (z coordinate)

$$C_h = \begin{bmatrix} ax \\ ay \\ az \\ a \end{bmatrix}$$

here z component is a free variable instead of zero.

we know

$$W_h = P_h C_h$$

$$W_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix} \begin{bmatrix} ax \\ ay \\ az \\ \frac{az+a}{f} \end{bmatrix}$$

for cartesian coordinates

$$\frac{ax}{\frac{az+a}{f}} = \frac{xf}{f+z}$$

$$\therefore W = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{xf}{f+z} \\ \frac{fy}{f+z} \\ \frac{fz}{f+z} \end{bmatrix}$$

solving for z in terms of z in last value & substituting in the first two expression

we get

$$x = \frac{x}{f}(f-z)$$

$$y = \frac{y}{f}(f-z)$$

Scanned by CamScanner

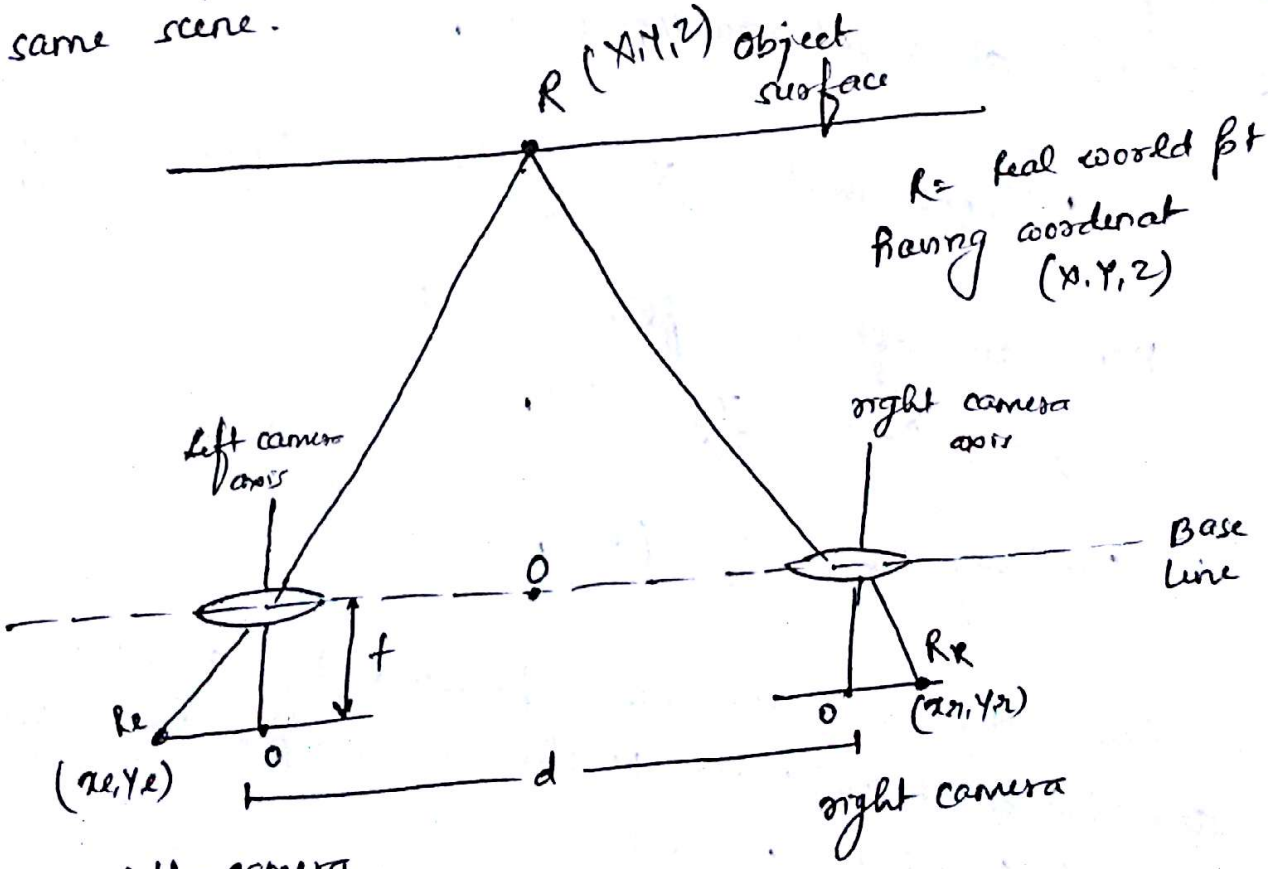
Stereo Imaging :-

⇒ In perspective transformation when we map a 3D scene onto a 2D image plane, we lose the z axis information. This z axis represents depth.

⇒ This depth information can be obtain using stereo imaging.

⇒ We posses a pair of eyes, two images of the same scene are captured. These two images are processed by brain and depth information is extracted

Hence to compute the depth information, we need two sensors placed at a particular distance capturing the same scene.



Simple arrangement for stereo imaging

Scanned by CamScanner

⇒ We have two cameras C_1 & C_2 with their optical axis parallel and seperated by a distance (d).

⇒ The line connecting the two camera lenses is called the base line.

⇒ Here we made two assumption

1) The base line is perpendicular to the line of sight of the two cameras.
2) The two cameras are perfectly aligned, differing only in location of their origin

The two cameras capture pt R on two diffrent images This pt is represented as (x_l, y_l) in left image plane & (x_r, y_r) at right image plane.

We know that $X = \frac{x}{f}(f - z)$ (From perspective Transformation)

Hence on left image plane we get

$$X_L = \frac{x_l}{f}(f - z_1)$$

Similarly $X_R = \frac{x_r}{f}(f - z_2)$

Since the cameras are seperated by a distance d and the z coordinates is the same for both camera coordinate ($z_1 = z_2$) we get

$$X_R = X_L + d$$
$$z_1 = z_2 = z$$

Scanned by CamScanner

Substituting it in eqn (2)

$$X_L + d = \frac{x_r}{f}(f - z) \quad \text{——— (3)}$$

Subtract eqn (1) from eqn (3)

$$X_L + d - X_L = \frac{x_r}{f}(f - z) - \frac{x_l}{f}(f - z)$$

$$d = \frac{x_r - x_l}{f}(f - z)$$

$$\therefore fd = (x_r - x_l)(f - z)$$

$$\frac{fd}{x_r - x_l} = f - z$$

$$\boxed{z = f - \frac{fd}{x_r - x_l}}$$

Therefore if we know the difference b/w right & left image plane coordinates, base line distance d and the focal length, we can compute the z coordinate of the real world pt.

This is depth information.

Application :

3D Movies

Scanned by CamScanner

Image segmentation

It is a image processing method where I/p is an image and O/p is image attributes.

Segmentation subdivides an image into its constituent regions or objects

Segmentation is used to analyze the content of an image.

Segmentation is used when we want the computer to make a decision.

segmentation is used when we automate a particular activity. The aim in an automated system is to extract important feature from image data from which description, interpretation or understanding of the scene can be provided by machine It is not use for human interpretation

Ex:

1. Automated blood cell counting
2. Finger print matching in forensic studies

Most of segmentation algorithms are based on one of the two basic properties.

1. segmentation based on discontinuity
2. segmentation based on similarity

discontinuity Partition an image based on sudden change in intensity such as edges.

Scanned by CamScanner

Similarity: Partition an image into regions that are similar according to a set of predefined criteria.

Eg Thresholding, region growing, region splitting & region merging.

segmentation < Discontinuity / Similarity

Image segmentation based on discontinuities

There are 3 basic discontinuities in the intensity -

1. Points
2. Lines
3. Edges

The easiest way is to use mask which have property to detect these discontinuities.

Point detection -

The point detection is based on the second derivative. using the Laplacian.

$$\nabla^2 f(x,y) = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

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

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

Scanned by CamScanner

The Laplacian is then

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

The o/p is obtain using the following ~~reason~~ expression -

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases}$$

0	1	0
1	-4	1
0	1	0

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

T = Threshold value (Non negative)

$R(x,y)$ = Response of mask at centre point

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

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

The intensity of the isolated point will be quite diffuent from its surroundings and thus will be easily detectable by this type of mask

1	1	1
1	-8	1
1	1	1

NOTE: $|R(x,y)|$ is taken because we want to detect both kind of points i'e white point on a black background as well as black point on a white background.

Scanned by CamScanner

(2) Line detection :-

Detection of lene is done using the mask shown below.

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

Horizontal

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

+45°

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

vertical

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

-45°

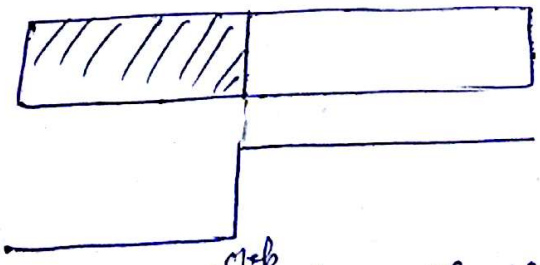
In an image, lines can be in any direction and detecting these lines would need different mask.

(3) Edge detection :-

An edge can be defined as a set of connected pixels that form a boundary b/w two disjoint regions.

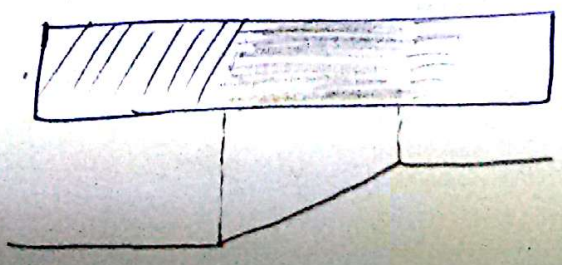
Edge models are classified according to their intensity profiles –

Step edge - It involves transition b/w two intensity levels occuring ideally over a distance of 1 pixel.



Step edge with their corresponding intensity profile

Ramp edge :-



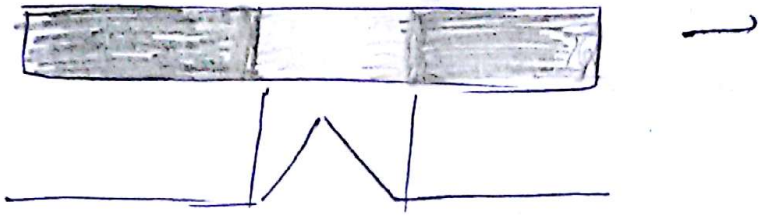
Ramp edge

Scanned by CamScanner

digital image have edges that are blurred and noisy. In such situation, edges are more closely modeled as having an intensity ramp edge.

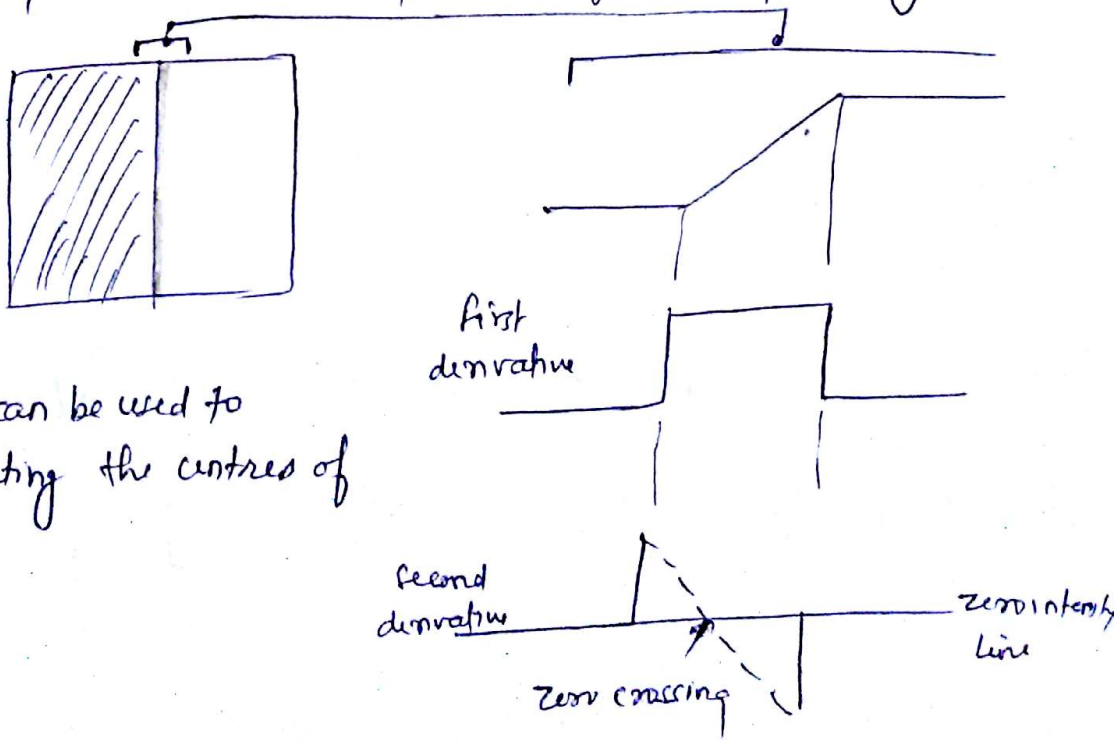
Slope of ramp is inversely proportional to degree of blurring.

Roof edge :



Roof edges are models of line through a region, with the base (width) of roof edge being determine by the thickness and sharpness of line.

These models allow us to write mathematical expression for edges in the development of image processing algo.



Zero crossing can be used to identify or locating the centres of thick edges

Scanned by CamScanner

Basic edge detection

The process of edge detection is carried out by derivative approach.

The image gradient and its property

For finding edge strength and direction at location (x,y) of an image f, the gradient is used and denoted by ∇f, and defined as vector

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$$

∇f = It points in the direction of the greatest rate of change of f at location (x,y).

The magnitude (length) of vector ∇f, denoted as $M(x,y)$

where $M(x,y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$

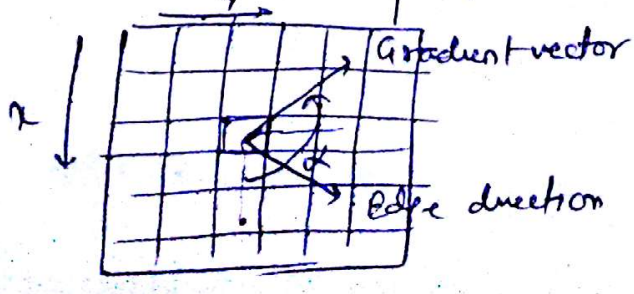
$M(x,y)$: Value of the rate of change in the direction of gradient vector

The direction of gradient vector is given by an angle

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

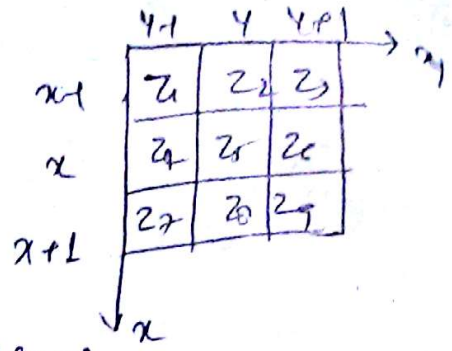
measured w.r.t x axis.

NOTE: The direction of an edge at an arbitrary point (x,y) is orthogonal to the direction, $\alpha(x,y)$, of the gradient vector at that point



Scanned by CamScanner

Finding gradient using mask :



$$g(x) = \frac{\delta f}{\delta x} = \lim_{h \to 0} \frac{f(x+h) - f(x,y)}{h}$$

$$g_y = \frac{\delta f}{\delta y} = \lim_{k \to 0} \frac{f(x,y+k) - f(x,y)}{k}$$

in discrete domain $h = k = 1$

$$\therefore \frac{\delta f}{\delta x} = f(x+1, y) - f(x,y) = z_8 - z_5$$

$$\frac{\delta f}{\delta y} = f(x, y+1) - f(x,y) = z_6 - z_5$$

$$\nabla f = \frac{\delta f}{\delta x} + \frac{\delta f}{\delta y}$$

$$\text{mag } \nabla f = \left[\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2 \right]^{1/2}$$

$$= \left[(z_8 - z_5)^2 + (z_6 - z_5)^2 \right]^{1/2}$$

$$|\nabla f| = |z_8 - z_5| + |z_6 - z_5|$$

$$= |z_5 - z_8| + |z_5 - z_6|$$

This is the first order difference gradient. This can be implemented using two masks

1
-1

mask 1

1	-1

mask 2

This is known as ordinary operator

Scanned by CamScanner

Step to compute the gradient of an image

1. Convolve the original image with mask 1. This gives the gradient along x direction
2. Convolve the original image with mask 2 " "
   " " " y direction
3. Add the result of 1 & 2

The another direct method of implementing the ordinary operator is

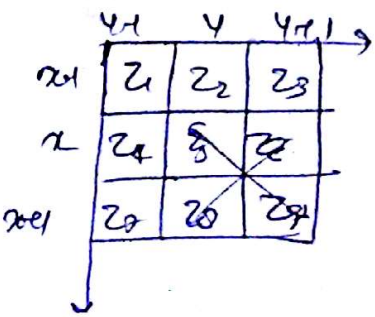
1. Add mask 1 & mask 2
2. Convolve the original image with the resultant mask to get the gradient image

Roberts mask:

He stated that better result could be obtained if cross diffrence were taken instead of the straight diffrence

Ordinary mask $|\nabla f| \approx |z_5 - z_8| + |z_5 - z_6|$

Roberts mask $= |z_5 - z_9| + |z_6 - z_8|$



1	0
0	-1

0	1
-1	0

Drawback: The mask is of even size.

Scanned by CamScanner

Prewitts & Sobel operators:

2×2 mask are simple but they as useful for computing edge direc... mask that are symmetrical about centre p...

Prewitts operator:

It uses 3×3 mask. It while approximating the first derivative, assign similar weight to all the neighbours of the candidate pixel whose edge strength is being calculated.

$$|\nabla f| = \underbrace{|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)|}_{x \text{ gradient}} + \underbrace{|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|}_{y \text{ gradient}}$$

$$g_x = \frac{\delta f}{\delta x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

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

g_x

$$g_y = \frac{\delta f}{\delta y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

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

g_y

Prewitts mask

Sobel operator:

$$\nabla f = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

using 2 in centre location provides image smoothing.

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

g_x

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

g_y

Scanned by CamScanner

Prewitts mask are simpler to implement than the sobel masks but the sobel mask have better noise suppression (smoothing) characteristics

$$M(x,y) \simeq |g_x| + |g_y|$$

* Noise are also high frequency content when the image is noisy and we implement high pass mask then the derivative of noise terms is always high

For noisy image we perform prewitt & sobel operator.

-1	-1	-1
0	0	0
+1	+1	+1

⇒

-1
0
1

H.P

×

1	1	1

L.P

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

⇒

1	1	1

L.P

×

-1	0	1

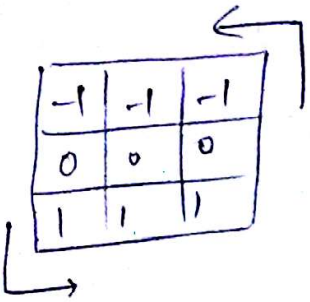
H.P

Compass operator :-

Edges in horizontal & vertical direction are enhanced with prewitt's or sobel's operator. There are application we need edges in all the directions

A simple method would be to rotate the prewitt's or sobel's mask in all the possible direction

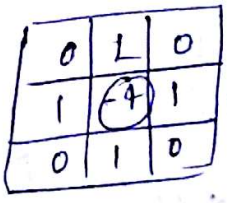
Scanned by CamScanner



This operator is known as compass operator and is very useful for detecting edges in diagonal direction also

Image segmentation using the second derivative :

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



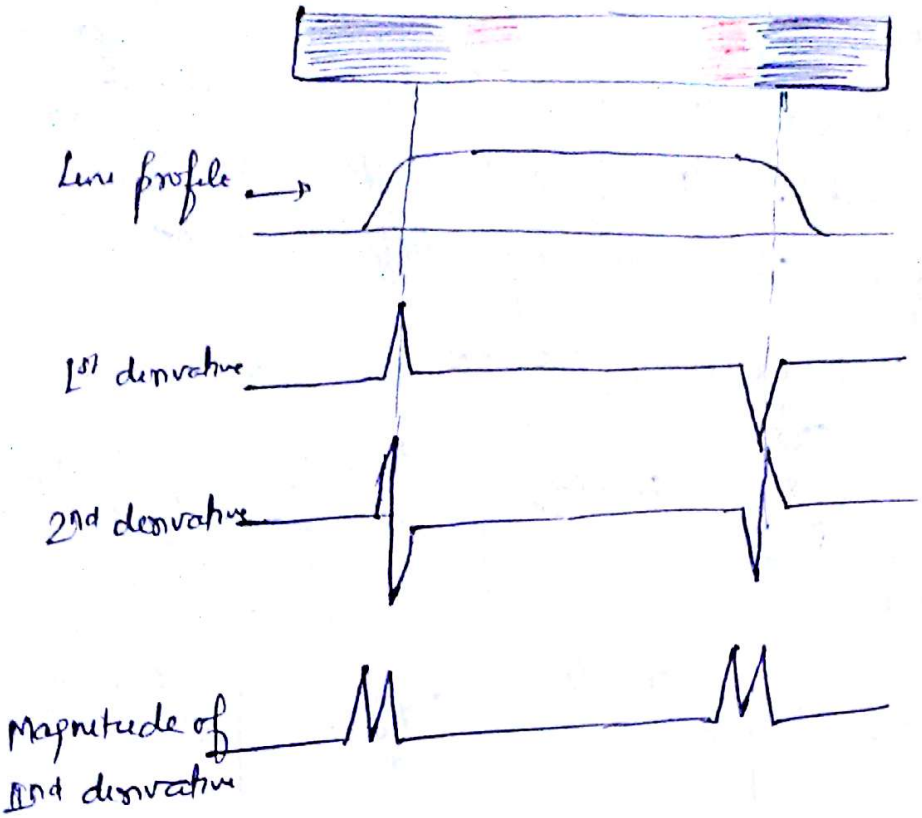
we have a laplacian mask still we can not use it directly to find out the edges in the image due to the following reason-

① Since laplacian is 2nd derivative filter, it is very sensitive to noise. Hence if there is any noise in the image, the laplacian gives very large values and damage the entire image.

② The magnitude of laplacian produces double edges. The magnitude produces 2 peaks for a single edge.

Zero crossing property of laplacian is used to detect edges

Scanned by CamScanner



The Marr-Hildreth edge detector (Laplacian of gaussian LoG):

The Laplacian mask results very strong response to noise pixels. Hence if some noise cleaning is done prior to the application of Laplacian operator, better result could be obtained.

Hence Laplacian is proceeded by a smoothing operation (Gaussian). The resultant algorithm is known as a Laplacian of Gaussian (LoG) or Marr-Hildreth operator

Marr & Hildreth argued that the most satisfactory operator fulfilling their condition is the filter $\nabla^2 G$ where ∇^2 is the laplacian operator $(\partial^2/\partial x^2 + \partial^2/\partial y^2)$ and G is the 2D Gaussian function.

Scanned by CamScanner

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

with standard deviation σ (sometimes σ is called the space constant). To find an expression for $\nabla^2 G$ we perform the following differentiations:

$$\nabla^2 G(x,y) = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2}$$

$$\Rightarrow \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-(x^2+y^2)/2\sigma^2}\right] + \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}\right]$$

$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-(x^2+y^2)/2\sigma^2} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-(x^2+y^2)/2\sigma^2}$$

$$\boxed{\nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{(x^2+y^2)}{2\sigma^2}}}$$

This expression is called the Laplacian of a Gaussian (LOG).

$$G(x,y) = e^{-x^2+y^2/2\sigma^2}$$

Let $x^2+y^2 = r^2$ $\quad G(r) = e^{-r^2/2\sigma^2}$

$$\frac{\partial G(r)}{\partial r} = \frac{-2r}{2\sigma^2} e^{-r^2/2\sigma^2}$$

$$= \frac{-r}{\sigma^2} e^{-r^2/2\sigma^2}$$

$$\frac{\partial^2 G(r)}{\partial r} = \frac{\partial}{\partial r}\left[\frac{-r}{\sigma^2} e^{-r^2/2\sigma^2}\right]$$

$$\Rightarrow -\frac{1}{\sigma^2}\left[r \frac{\partial}{\partial r} e^{-r^2/2\sigma^2} + e^{-r^2/2\sigma^2} \frac{\partial}{\partial r} r\right]$$

$$= -\frac{1}{\sigma^2}\left[\frac{-r^2}{\sigma^2} e^{-r^2/2\sigma^2} + e^{-r^2/2\sigma^2}\right]$$

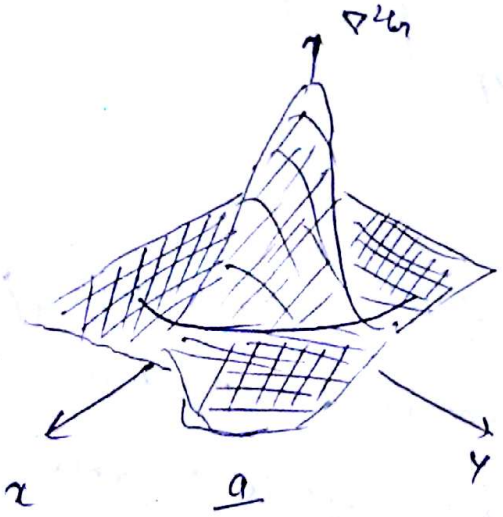
Scanned by CamScanner

$$\frac{e^{-r^2/2\sigma^2}}{\sigma^2}\left[\frac{r^2}{\sigma^2}-1\right]$$

$$= \frac{e^{-r^2/2\sigma^2}}{\sigma^2}\left[\frac{r^2-\sigma^2}{\sigma^2}\right]$$

$$\nabla^2 G(r) = \left(\frac{r^2-\sigma^2}{\sigma^4}\right) e^{-r^2/2\sigma^2}$$

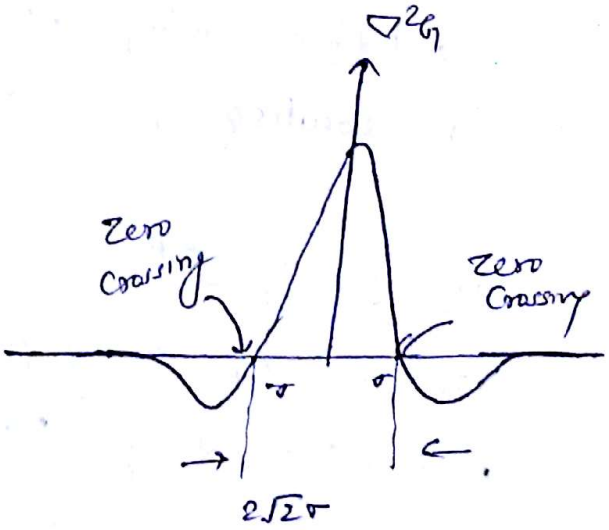
$$\boxed{\nabla^2 G(x,y) = \left(\frac{x^2+y^2-\sigma^2}{\sigma^4}\right) e^{-(x^2+y^2)/2\sigma^2}}$$



(a)

3D plot of (-)ve of LoG

$\Rightarrow$



Cross section of (a) showing zero crossing

(zero cross section of the LoG occur at $x^2+y^2 = 2\sigma^2$. which defines a circle of radius $\sqrt{2}\sigma$ centred on the origin.)

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

5x5 approximation to the shape in (a)

Scanned by CamScanner

Due to the shape of LoG, it commonly refered to as the Mexican Hat function

This mask is not unique, we could use any values that approximate the shape of the LoG.

The Marr-Hildreth algo consist of convoling the ~~mask~~ LoG filter with an

Canny edge detector:

I/P image, $f(x,y)$

$$g(x,y) = [\nabla^2 G(x,y)] * f(x,y)$$

and then finding the zero crossing of $g(x,y)$ to determine the location of edges in $f(x,y)$

Zero crossing at P:

Zero crossing at any fixed P of the filtered image $g(x,y)$ is based on using 3x3 neighbourhood centred at P.

It implies that at least two of its opposite neighbor fixed must differ.

If the values of $g(x,y)$ are being compared against a threshold then not only signs are different but the absoluted value of their numerical diffrence must also exceed the threshold before we can call P a zero crossing fixed.

$$DOG = \frac{1}{2\pi\sigma_1^2} e^{-x^2+y^2/2\sigma_1^2} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \quad \text{with } \sigma_1 > \sigma_2$$

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2-\sigma_2^2} \ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$$

for same zero crossing pt.

Scanned by CamScanner

Canny edge detector :

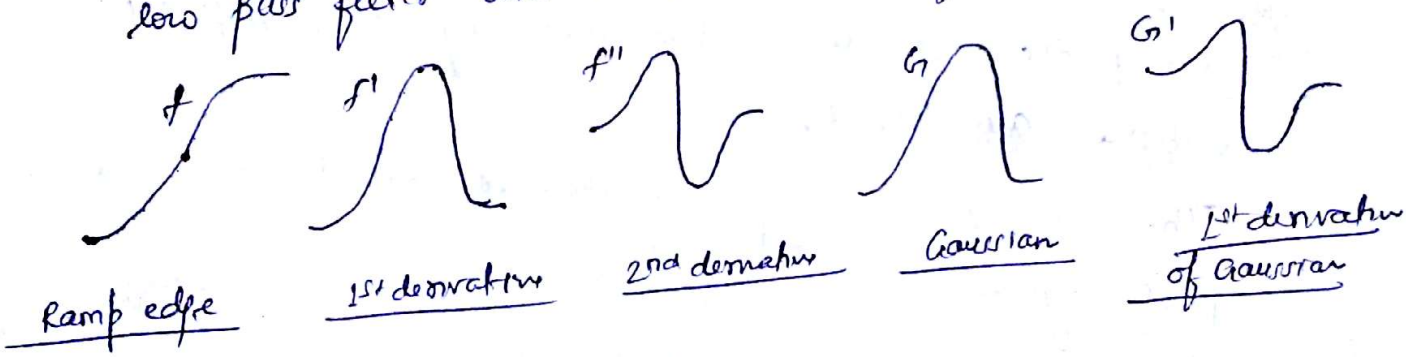
Canny's approach is based on 3 basic properties-

1. Low error rate
2. Edge point should be well localized
3. Single edge point response.

Canny operator is a 1st derivative edge detector coupled with noise cleaning.

Like in LOG, a gaussian function is used to smoothen the noise

In this, we first smoothen the image using a Gaussian low pass filter and then take the first derivative

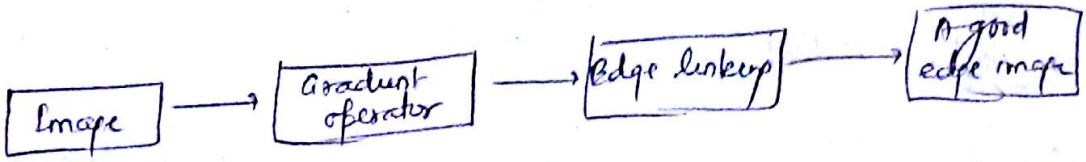


Edge linking :-

The gradient operators like Roberts, sobel, Prewitt enhance the edges. but produce usually breaks in lines.

Due to this edge detection algo are generally followed by edge linking which form edges from edge pixels.

Scanned by CamScanner



There are 3 major technique for edge ~~detection~~, linkup...

① Local processing (Edge point in local region)

② Global processing (works with an entire edge image)

③ Points on the boundary of region be known

Local processing:

A simple approach for edge linkup is to analyze the characteristics of pixels in small neighbourhood about every pt (x,y) that has been declared an edge point. All points that are similar according to predefined criteria are linked.

The two common properties that we consider to link pixels together are.

① The strength (magnitude)

② The direction of gradient vector

① Strength of gradient:

Let S_{xy} = Set of coordinates of neighbourhood centered at point (x,y)

An edge pixel (s,t) in S_{xy} is similar in magnitude to the pixel at (x,y) if

$$|M(s,t) - M(x,y)| \leq E$$

where E is +ive threshold.

Scanned by CamScanner

The duction of gradient:

An edge pixel with coordinate (s,t) in S_{xy} has an angle similar to the pixel at (x,y) if

$$|\alpha(s,t) - \alpha(x,y)| \leq A$$

where A is (+)ive angle threshold.

The duction of edge at (x,y) is perpendicular to the duction of gradient vector at that point

A pixel with coordinat (s,t) in S_{xy} is linked to the pixed at (x,y) if both magnitude and duction criteria are satisfied This process is repeated at every location in an image

Global processing via graph theoretic techniques:

This is a global approach for detecting edges. We represent edges in the form of graph and then search for a path which has a lowest cost. This low cost path correspond to the most significant edge.

We assume two pixel P & Q such that P & Q are 4 neighbours. Each edge element, defined by pixel P & Q, has an associated cost

Scanned by CamScanner

$$c(P,Q) = \max[I] - [f(P) - f(Q)]$$

where max(I) is the maximum gray level in the image.

$$I = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 5 & 6 & 1 \\ 2 & 6 & 7 & 0 \\ 3 & 7 & 1 & 3 \end{array}$$

$c(P,Q) = \max[I] - [f(P) - f(Q)]$ to navigate

We assume that edges start from top row & to last row.

A =
	1	2	3
1	5	6	1
2	6	7	0
3	7	1	3

B =
	5	6	1
	6	7	0
	7	1	3

D =
	5	6	1
	6	7	0
	7	1	3

C =
	5	6	1
	6	7	0
	7	1	3

E =
	5	6	1
	6	7	0
	7	1	3

F =
	5	6	1
	6	7	0
	7	1	3

G =
	5	6	1
	6	7	0
	7	1	3

H =
	5	6	1
	6	7	0
	7	1	3

Scanned by CamScanner

Global processing using Hough Transform :-

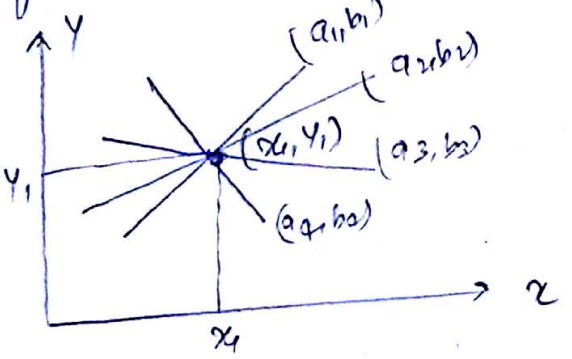
Hough transform is the method of pixel linking and curve detection

Given a set of discrete pixels, the Hough transform checks if these points lie on a straight line and if yes, it draws a line joining all these points

Consider a point (x_1, y_1). A line passing through this point can be written in slope intercept form as

$$y_1 = ax + b$$

using this eqn & varying the values of a & b, infinite number of lines pass through this point (x_1, y_1)



Let the eqn is written as

$$b = -ax + y_1$$

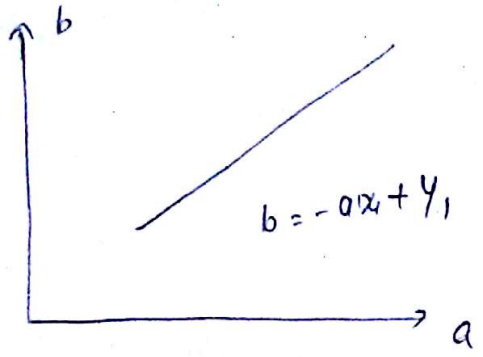
& consider the ab plane we get a single line for a point (x, y_1).

This entire line in ab plane is due to a single point in xy plane and different values of a & b.

Now consider another point (x_2, y_2) in the xy plane

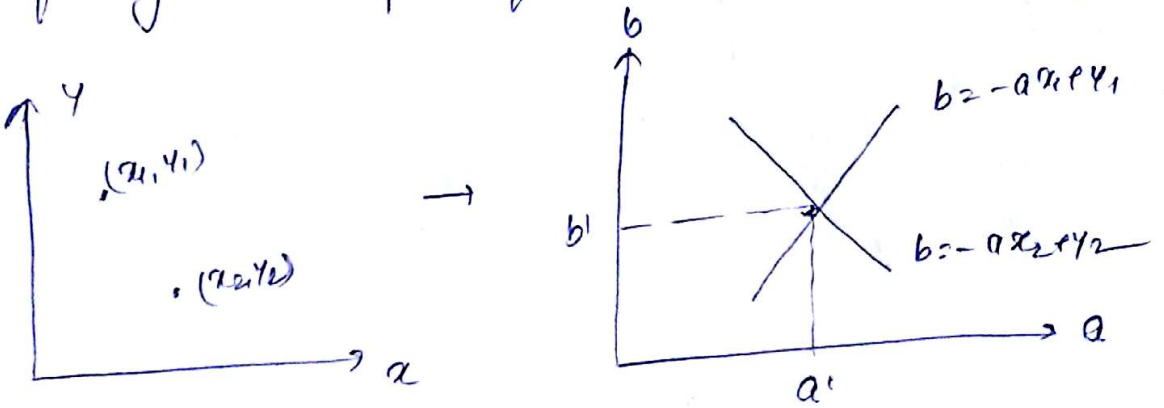
$$y_2 = ax_2 + b$$

Scanned by CamScanner



$b = -ax_2 + y_2$

this is another line in ab plane. These two lines will intersect each other somewhere in ab plane only if they are a part of a straight line in xy plane

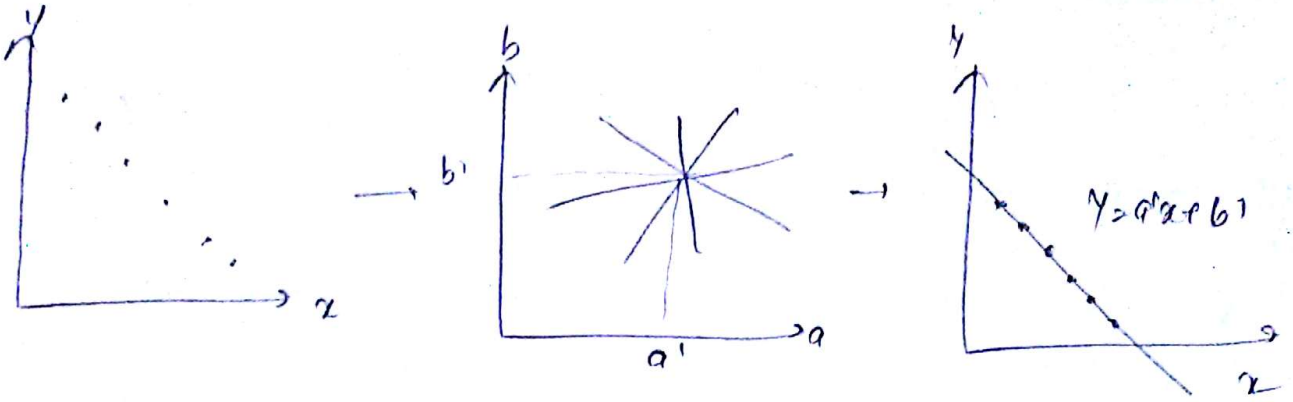


then from $y = a'x + b'$ we get a line that passes through points (x_1, y_1) & (x_2, y_2) in the xy plane.

Now if we have n points that lie on a straight line we get n lines in the ab plane and all these lines would intersect at (a', b') in ab plane.

Using these value of a' & b' in eqn $y = a'x + b'$, we would get a line that would pass through all these points in the xy plane.

Scanned by CamScanner



The ab plane is called the parameter space

Image segmentation based on similarities :-

segmentation based on thresholding

Im segmentation thresholding is also used to produce regions of uniformity within the given image based on some threshold criteria T.

Let we have an image $f(x,y)$ compared of light objects on a dark background. One way to extract the objects from the background is to select a threshold T.

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

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

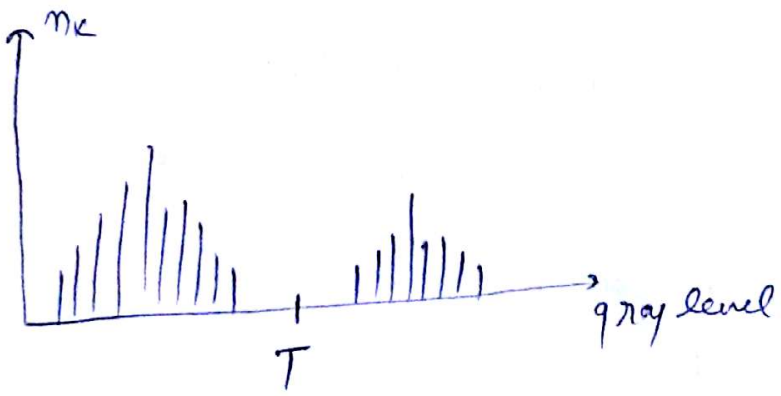
when T is constant over an entire image, then this eqn is refered to as global thresholding.

Scanned by CamScanner

When the value of T changes we use term variable thresholding.

⇒ Local or regional thresholding is used some time to denote variable thresholding in which the value of T at any point (x,y) in an image depends on properties of a neighborhood of (x,y).

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



Multiple thresholding:

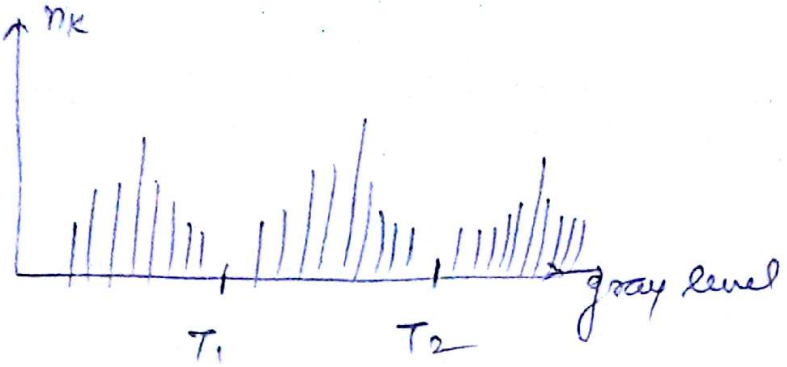
It involve a histogram with 3 dominant modes.

Eg An image with 2 different type of dark objects against light background.

In such case we required 2 threshold value T1 & T2 to seperate the two objects from their background.

Thus is known as multiple or multilevel thresholding.

Scanned by CamScanner



point (x,y) belongs to background if $f(x,y) \leq T_1$

" " " one object class if $T_1 < f(x,y) \leq T_2$

" " " other " " if $f(x,y) > T_2$

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

when a, b, c are 3 distinct intensity values

Basic global thresholding:

When the intensity distributions of objects and background pixels are sufficiently distinct, it is possible to use single (global) threshold over the entire image

An algorithm capable of estimating automatically the threshold value for each image is required.

Following iterative algorithm can be used for this-

① Select an initial estimate for the global threshold T.

② Segment the image using T, this will produce two group of pixels: G_1 (All pixel) $> T$
G_2 (" ") $\leq T$

③ Compute the avg intensity value m_1 & m_2 for the pixels in G_1 & G_2 respectively.

Scanned by CamScanner

④ Compute a new threshold value.

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

⑤ Repeat steps 2 through 4 until the difference b/w value of T in successive iterations is smaller than a predefined parameter ΔT

Optimum global Thresholding:

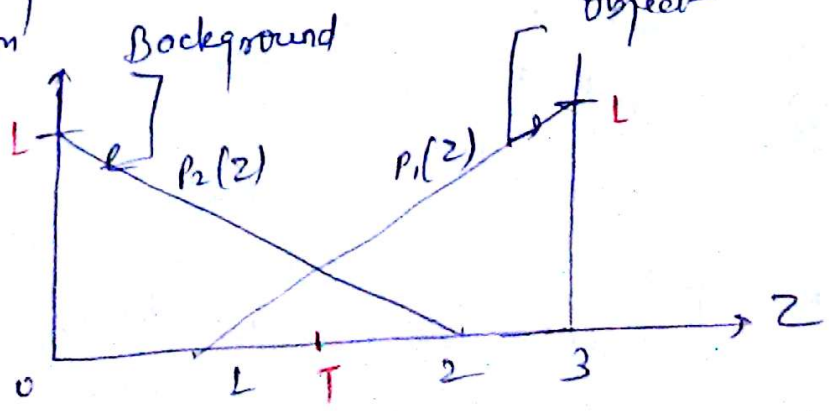
The objective of optimum thresholding is to minimize the avg error incurred in assigning pixels to two or more groups.

Ex: Suppose that an image has the gray level probability density function as shown. $P_1(z)$ corresponds to the object & $P_2(z)$ corresponds to the background. Assume $P_1 = P_2$. Find the optimum threshold b/w object & background pixels.

→ The soln is based on only two parameters-

→ The P.D.F of intensity level of each class

→ Probability that each class occurs in given application



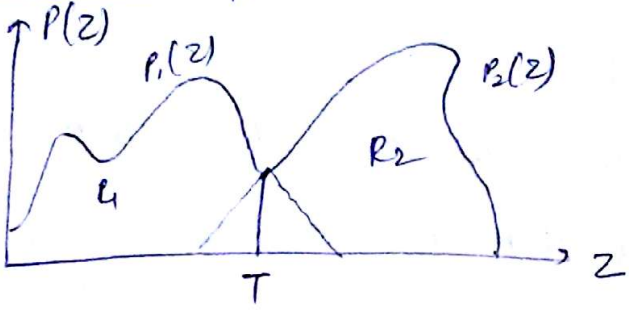
Scanned by CamScanner

optimum global thresholding is 2 type -

① Parametric approach

② Non parametric approach (Otsu's method)

① Parametric approach :



⇒ Parametric methods use metrics that are derived from data

⇒ Let us assume two classes class 1 with probability P_1 & class 2 with probability P_2 of the gray scale value z

⇒ $\therefore P_1 + P_2 = 1$

$$P_1(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu_1)^2}{2\sigma^2}}$$

$$P_2(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu_2)^2}{2\sigma^2}}$$

μ_1 = Mean gray level of R_1

μ_2 = Mean " " of R_2

$$P_1 P_1(z) = P_2 P_2(z)$$

The normalized histogram for the mixture probability density function can then be written as follows when $z = T$

Scanned by CamScanner

$P_1 P_1(T) = P_2 P_2(T)$

$P(T) = P_1 + P_2$

Once this T is chosen, the probability of falsely classifying a pixel that actually belong to R_2 to R_1 is $P_1(T)$ & similarly R_1 to R_2 is $P_2(T)$

Our task is to keep this error to its min value

After doing this The optimum threshold is

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{(\mu_1 - \mu_2)} \ln\left(\frac{P_2}{P_1}\right)$$

if we assume 2 different value of σ_1 & σ_2 for R_1 & R_2 —

$$T = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$a = \sigma_1^2 - \sigma_2^2$$

$$b = 2(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)$$

$$c = \sigma_1^2 \mu_2^2 - \sigma_2^2 \mu_1^2 + 2\sigma_1^2 \sigma_2^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right)$$

Non parametric (Otsu's method)

Refer Page No. 742 of Rafel C. Golzanez, 3rd edition.

Scanned by CamScanner

It is given that $P_1 = P_2$ (A priori probability)

Probability calculated by logically examining existing information or making conclusion based on deductive reasoning rather than research or calculation

We can write eqn for $P_1(z)$ as well as $P_2(z)$

$$P_1(z) = \begin{cases} 0 & z < 1 \\ \frac{1}{2}z - \frac{1}{2} & 1 \le z \le 3 \\ 0 & z \le 3 \end{cases}$$

$$P_2(z) = \begin{cases} 0 & z < 0 \\ -\frac{1}{2}z + 1 & 0 \le z \le 2 \\ 0 & z < 2 \end{cases}$$

Let T be the optimal thresholding for the above histogram

$$P_1 P_1(T) = P_2 P_2(T)$$

$$\because P_1 = P_2$$

we get $P_1(T) = P_2(T)$

$$\frac{1}{2}T - \frac{1}{2} = -\frac{1}{2}T + 1$$

$$\frac{1}{2}T + \frac{1}{2}T = 1 + \frac{1}{2}$$

$$T = 1.5$$

Hence the optimal threshold is $z = T = 1.5$

Scanned by CamScanner

Region based segmentation :-

Let R represents the entire spatial region occupied by an image. Region R is divided into subregions $R_1, R_2, \ldots R_n$ such that -

(I) $\bigcup_{i=1}^{n} R_i = R$

(II) R_i is connected set, $i = 1, 2, \ldots n$

(III) $R_i \cap R_j = \phi$ for $\forall i \& j$, $i \neq j$

(IV) $Q(R_i) =$ True for $i = 1, 2, \ldots n$

$Q(R_k)$ is the logical predicate defined over the points in set R_k

(V) $Q(R_i \cup R_j) =$ FALSE for any adjacent region R_i & R_j

(Two adjacent region must be different in the sense of predicate Q)

Region based segmentation partion an image into subregions based on some predefine criteria.

Region based segmentation can be performed in following ways. (I) Region merging (II) Region splitting

(1) Region growing / Merging by eg. region growing.

⇒ It is a procedure that groups pixels or subregions into larger regions based on predefine criteria for growth.

⇒ The basic approach is to select seed pixel and grow region from the seed pixel.

Scanned by CamScanner

Q. Segment this image using region growing technique

Predicate : $\max g(x,y) - \min\{g(x,y)\} < th$

$th \leq 3$

seed pixel = (6)

5	6	6	6	7	7	6	6
6	7	(6)	7	5	5	4	7
6	6	4	4	3	2	5	6
5	4	5	4	2	3	4	6
0	3	2	3	3	2	4	7
0	0	0	0	2	2	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	4

Scanned by CamScanner

Region merging

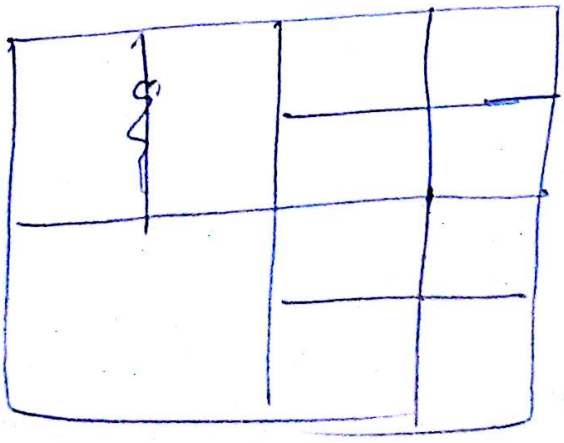
On this, it is applicable only to images whose number of rows & columns are an integer power of 2.

Here each pixel will be considered as a homogeneous region. For merging we check if the 4 adjacent homogeneous regions arranged in 2×2 fashion together satisfy the homogenity property.

If yes they are merged to form a bigger region otherwise the regions are left as they are.

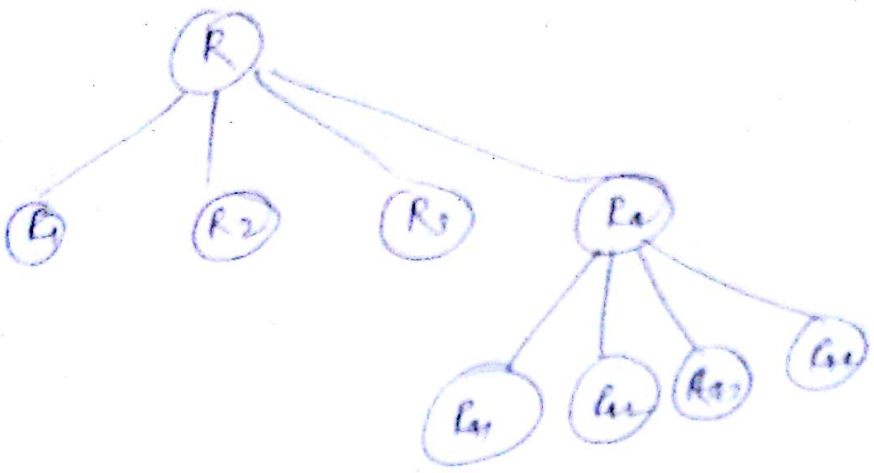
$P_i \Rightarrow \max\{g(x,y)\} - \min\{g(x,y)\} \leq 3$

5 6 6 7	6 6 6 7	7 7 5 5	6 6 4 7
6 6 5 4	4 4 5 4	3 2 2 3	5 6 4 6
0 3 0 0	2 3 0 0	3 2 2 2	4 7 5 6
1 1 1 0	0 1 1 0	0 3 2 3	4 4 5 4



Scanned by CamScanner

Region splitting :-

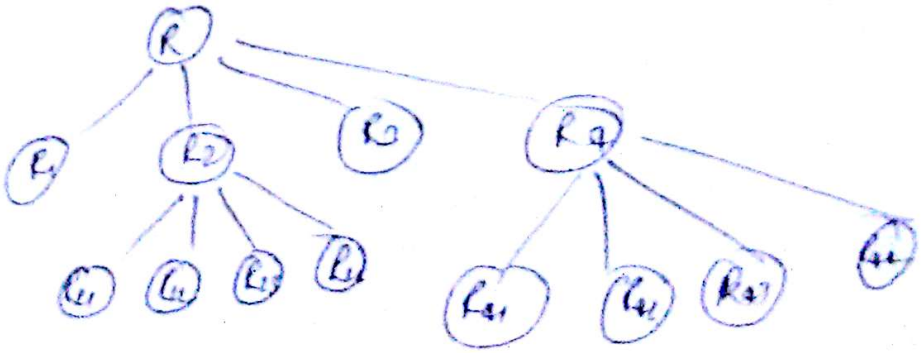


Quad Tree representation

This method is applicable to the image where number of rows and number of columns are some integer power of 2.

Pi {max g(xy) - min g(xy) ≤ 3

5	6	6	6	7	7	6	6
6	7	6	7	5	5	4	7
6	6	4	4	3	2	5	6
5	4	5	4	2	3	4	6
0	3	2	3	3	2	4	7
0	0	0	0	2	2	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	4



Scanned by CamScanner