Unit 5

Image Segmentation & Representation

- In the previous units input and output in image processing methods are images.
- In segmentation, input are images but outputs are attributes extracted from those images.
- Segmentation subdivides an image into its constituent regions or objects until when the objects or regions of interest in an application have been detected.

OR

- Segmentation is the process of partitioning a digital image into multiple regions and extracting the meaningful region which is known as Region of Interest (ROI).
  (1) ROI vary with applications
  (2) No ~~signle~~ single segmentation algo exists for segmenting the ROI in all Images.
  (3) Perform many segmentation algorithms and get the best

Image Segmentation Algorithm are Based on

Similarity	Discontinuity
✓ Region Approach	✓ Boundary Approach (Edge)
- Objective is to group pixels based on common property to extract a coherent region	- Objective is to extract regions that differ in properties like intensity, color, texture etc.

Definition of Image Segmentation :-

Image (R)

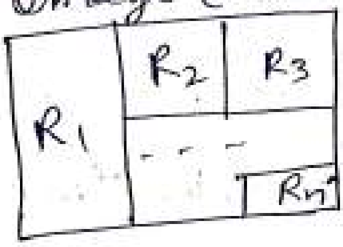


Image Segmentation as a process that partitions R into n subregions $R_1, R_2, \ldots R_n$

eg:-

10	10	20	20	20
10	10	20	20	20
10	10	20	20	20
30	30	30	20	20
30	30	30	20	20

Predicate
↓
Some property over the region

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

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

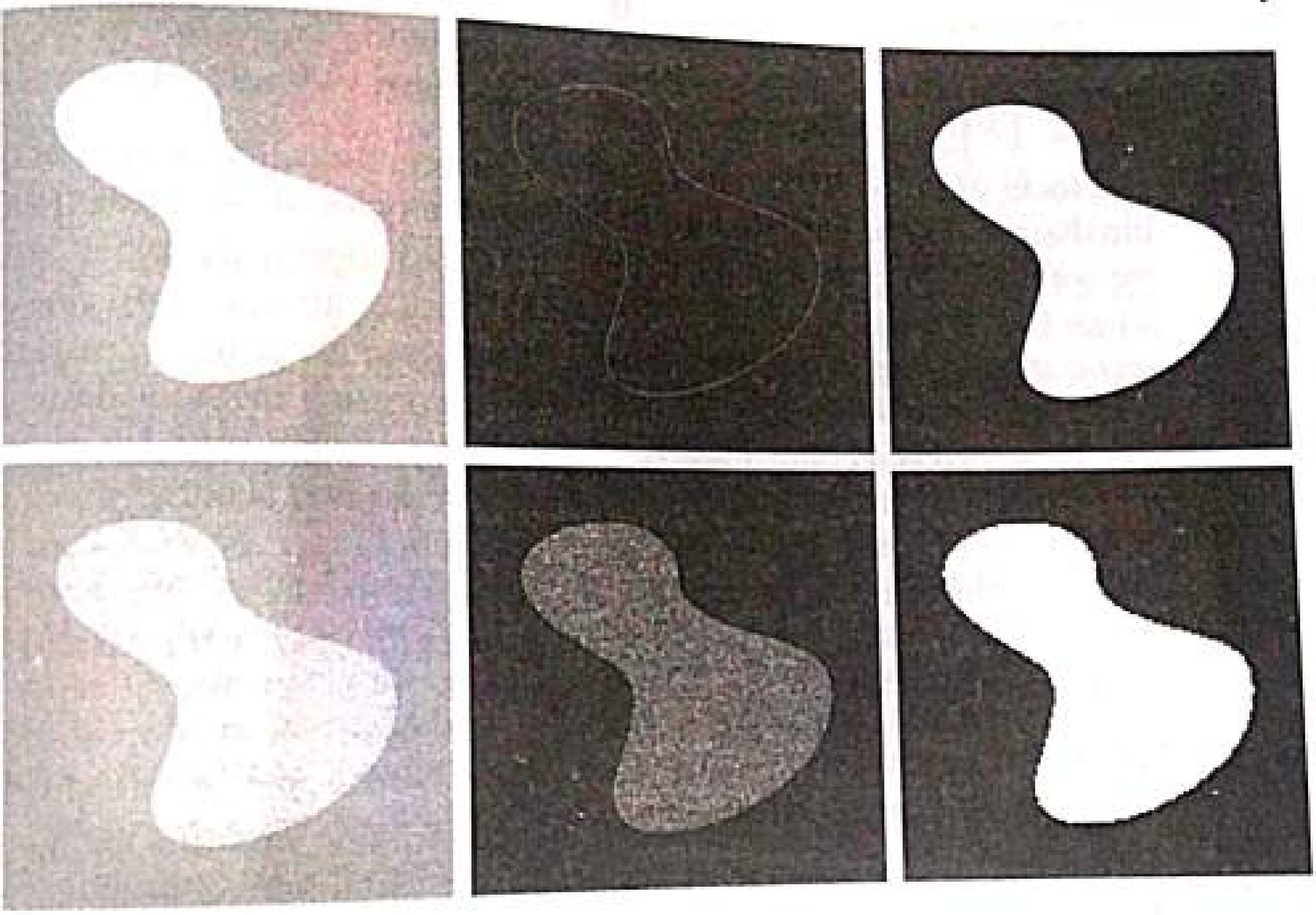
(3) $R_i \cap R_j = \phi$ for all i and j, $i \neq j$

(4) $Q(R_i) = \text{TRUE}$ for $i = 1, 2, \ldots n$

logical predicate

(5) $Q(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions R_i and R_j

...ginning of this section are satisfied by



a b c
d e f

FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Classification of Image Segmentation Algorithms

Based on User Interaction	Based on Pixel Relationship

Based on User Interaction

(1) Manual (Human observer and do manualy)

(2) Automatic (all done by algo)

(3) Semi-Automatic (Initial sed points by user).

Based on Pixel Relationship

(1) Contextual (Region based or global)
- grouped based on similarity like color, texture

(2) Non-textual (pixel based or local)
- Identify discontinuity like line, edge.

Types of Grey level discontinuities are:-

(1) Point - isolated point (equal to one pixel)

(2) Line - Edge segment

(3) Edge - pixel value change abruptly

Point Detection / Detection of Isolated Points:

In isolated point is a point whose grey level is significantly different from its background in a homogeneous area:

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Mask

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Image

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

With the help of threshold value, we can identify the point.

If $|R| \geq T$, a point is detected

Where,

T is a non negative Integer

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

eg.. sample mask for Point detection

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

Laplacian mask (Point detection)

Line Detection :-

In line detection, four type of masks are used to get the response i.e R_1, R_2, R_3 and R_4 for the directions vertical, horizontal, $+45°$ and $-45°$ respectively

Horizontal

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

+45°

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

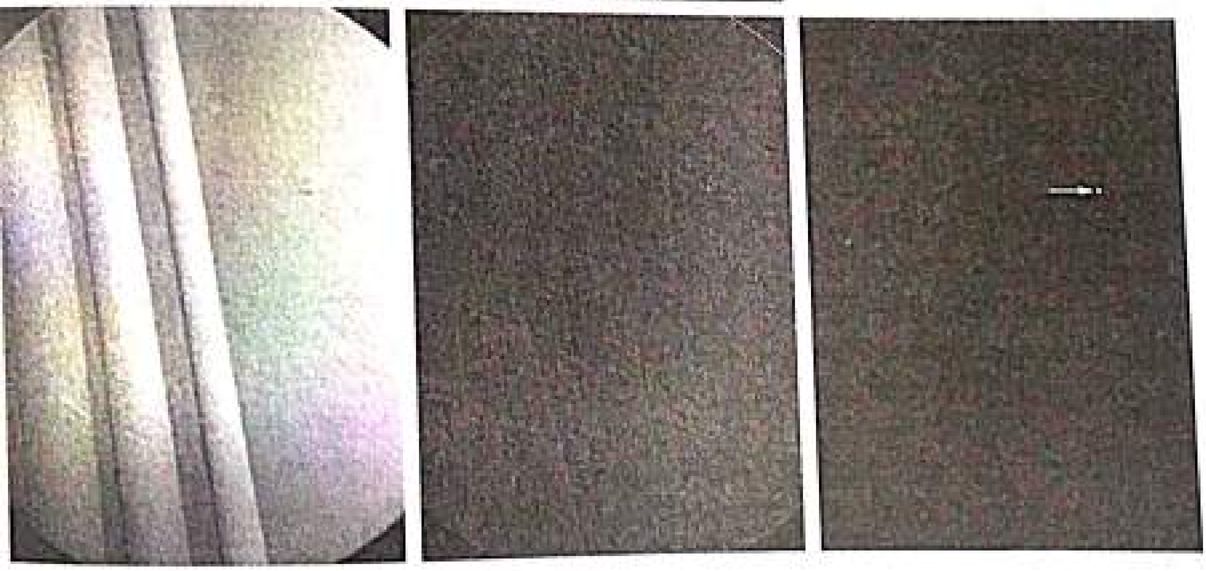
Vertical

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

-45°

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

1	1	1
1	−8	1
1	1	1



a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Response of the Mask

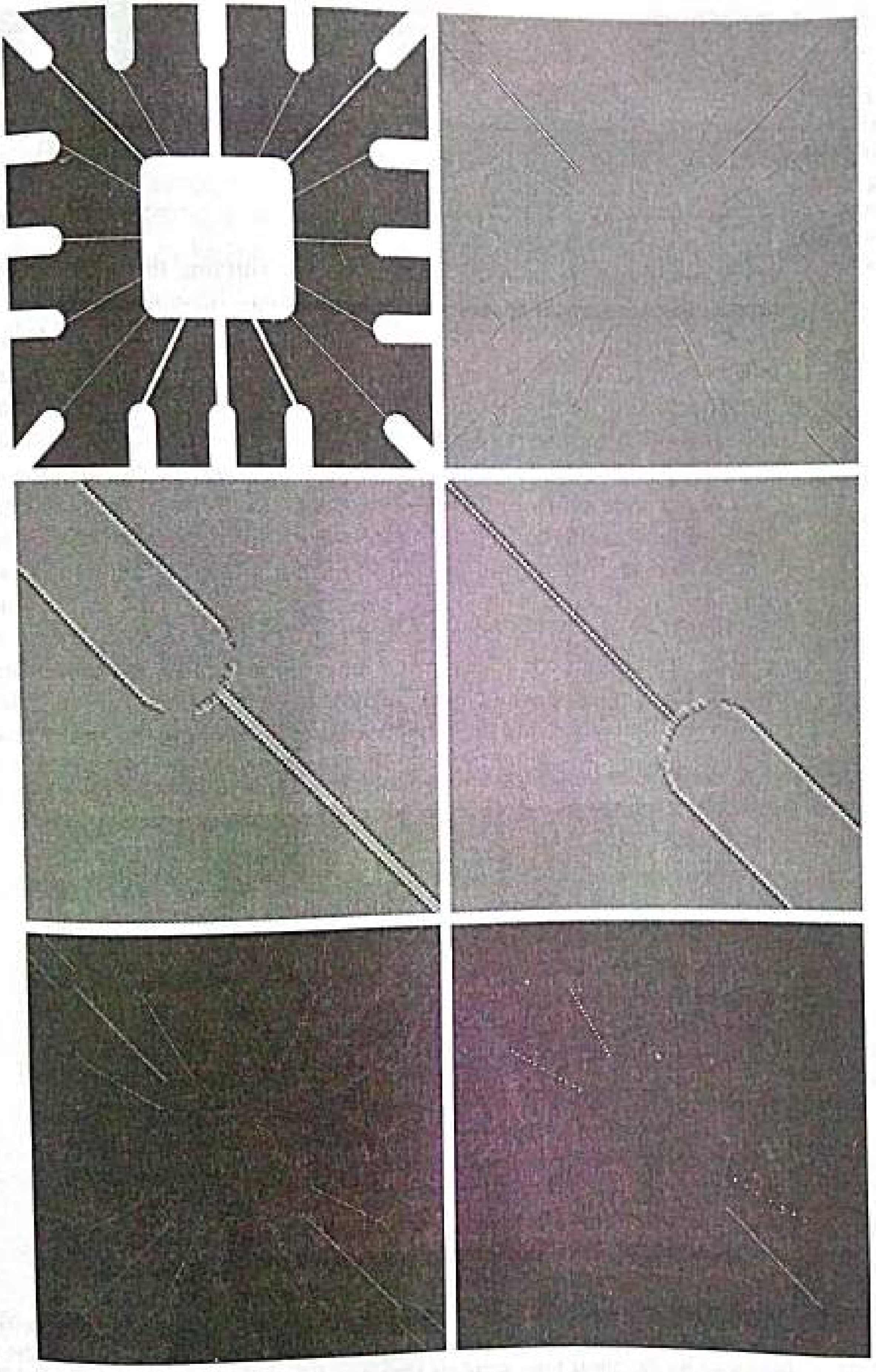
$$R_k = \sum_{k=1}^{4} w_k z_k$$

$R_1, R_2, R_3, R_4 \rightarrow$ but for eg: R_1 is more suitable $|R_1| > |R_1, R_2, R_3|$

If, at a certain point in the image, $|R_i| > |R_j|$ for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i.

Edge Detection:-

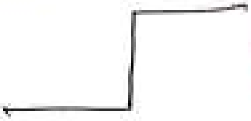
- An edge is a set of connected pixels that lies on the boundary between two regions which differ in grey value. Pixels on edge is known as edge points.
- Edge detection is the approach used most frequently for segmentaloong images based on abrupt (local) changes in intensity.
- Edges provide an outline of the object.
- It locate sharp changes in the intensity funᶜ.
- Edges are pixels where brightness changes abruptly
- An edge can be extracted by computing the derivation of the image funᶜ.
  (1) Magnitude of the derivation, indicates the strength of contrast of edge.
  (2) Direction of the derivative vector indicates the edge orientation



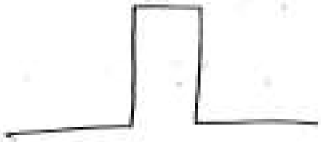
a b
c d
e f

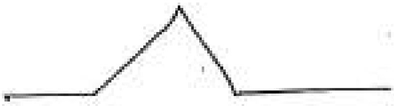
FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the $+45°$ line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

- Some of the commonly encountered edges in image processing are:

(1) Step Edge  abrupt change in intensity

(2) Ramp Edge  Slow and gradual change

(3) Spike Edge  Quick change

(4) Roof Edge  not instantaneous over short distance

Stages in Edge Detection

Start → Input image → filtering → Differentiation → localization → Display → End

filtering: Smoothing with the help of filters

localization: Determine the exact location of edge

Differentiation:- find the diff b/w two adjacent pixels

find strength & ~~location~~ direction at location (x,y) of image f.

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

$\boxed{\text{vector}}$

Edge strength

Magnitude of vector ∇f, denoted by $M(x,y)$

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

Direction of gradient

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

Edge Detection Algorithms

1. Derivate Types
   - It uses differentiation technique for edge detection (like diff b/w two adjacent pixel)

2. Template Matching
   - It uses templates that resembles the target shapes and match with image.

3. Gaussian Derivaties
   - Use gaussian filter concept + concept of derivatio
   - Very effective for real time images

4. Pattern fit Approach
   - Here surface is considered as topographic surface with pixel values representing altitude.

First Order Edge Detection Operator :-

- Edges play a important role in all image processing applications & show boundry of the object.

- Suppose you have a image and watching the intensity of the image, if there is difference in intensity means edge is there or local transitions among different image intensities constitute an edge.

- Therefore the objective is to measure the intensity gradient.

- Edge detectors can be viewed as gradient calculators have magnitude & direction both.

Gradient Operator → in 2D Image $f(x, y)$

(1) vector $\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$

(2) Magnitude $\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2}$

(3) Direction of gradient $\tan^{-1}\left(\frac{G_y}{G_x}\right)$

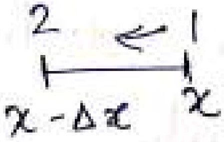
- An edge can be extracted by computing the derivative of the image func.

(1) Magnitude of derivative - show the strength or contrast of edge.

(2) Direction of derivative vector - show the edge orientation.

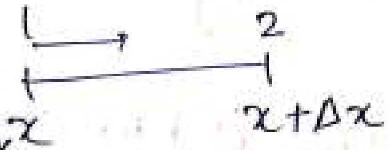
Now,

Backward Difference



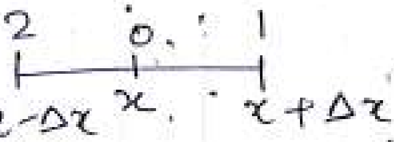
$= (f(x) - f(x-\Delta x)) / \Delta x$

forward Difference



$= (f(x+\Delta x) - f(x)) / \Delta x$

Central Difference



$= (f(x+\Delta x) - f(x-\Delta x)) / 2\Delta x$

- These differences can be obtained by applying the following masks, assuming $\Delta x = 1$

So,

Backward difference

$[f(x) - f(x-1)] = [1 \ -1]$

forward difference

$[f(x+1) - f(x)] = [-1 \ 1]$

With the help of these differences we can get the edges.

Robert Operator:-

- Robert kernels are derivatives with respect to the diagonal elements.
- When diagonal edge direction is of interest, need 2-D mask.
- Robert operator is known as
  Cross-gradient operator / cross diagonal difference

-1	0
0	1

0	-1
1	0

- Masks of size 2x2 are simple conceptually but they are not as useful for computing edge direction.

So, Generic gradient based algorithm can be

(1) Read the image and smooth it
(2) Convolve the image f with g_x $\hat{f}(x) = f * g_x$
(3) Convolve the image f with g_y $\hat{f}(y) = f * g_y$
(4) Compute the edge magnitude and edge orientation
(5) Compare the edge magnitude with a threshold value
  - If edge magnitude is higher, assign it as a possible edge point.

Prewitt Operator:-

The Prewitt method takes the central difference of the neighbouring pixels; This difference can be represented mathematically as:

$$g_x = \frac{\partial f}{\partial x} = [f(x+1) - f(x-1)]/2 \rightarrow 1D$$

$g_y / \frac{\partial f}{\partial y} /$ = for 2D (x, y)

$$[f(x+1, y) - f(x-1, y)]/2$$

Central difference is obtained by mask

$$[-1 \quad 0 \quad +1]$$

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Let we have an image

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

3x3 digital approximation of Perwitt operation is given by

$$G_x = \frac{\partial f}{\partial x} = (Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3)$$

$$G_y = \frac{\partial f}{\partial y} = (Z_3 + Z_6 + Z_9) - (Z_1 + Z_4 + Z_7)$$

$$\nabla f \cong |G_x| + |G_y|$$

Prewitt mask for detecting diagonal edges

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

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

Sobel Operator:-

- It provides both a differentiating and a smoothing effect
- Sobel operator relies on the central diff
- It can be viewed as an approximation of first Gaussian Derivative
- Here convolution is both commutative and associative.

$\rightarrow \frac{\partial}{\partial x}(f * G)$

$\rightarrow f * \frac{\partial}{\partial x}(G)$

G_x

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

G_y

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

- 3x3 digital approximation of sobel operator

$$G_x = \frac{\partial f}{\partial x} = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$$

$$G_y = \frac{\partial f}{\partial y} = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)$$

$$\nabla f \cong |G_x| + |G_y|$$

a b

FIGURE 10.13 One-dimensional masks used to implement Eqs. (10.2-12) and (10.2-13).

−1
1

−1	1

These two equations can be implemented for all pertinent values of x and y by filtering $f(x, y)$ with the 1-D masks in Fig. 10.13.

When diagonal edge direction is of interest, we need a 2-D mask. The Roberts *cross-gradient operators* (Roberts [1965]) are one of the earliest attempts to use 2-D masks with a diagonal preference. Consider the 3 × 3 region in Fig. 10.14(a). The Roberts operators are based on implementing the diagonal differences

In the remainder of this section we assume implicitly that f is a function of two variables, and omit the variables to simplify the notation.

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5) \tag{10.2-14}$$

and

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6) \tag{10.2-15}$$

a
b c
d e
f g

FIGURE 10.14 A 3 × 3 region of an image (the z's are intensity values) and various masks used to compute the gradient at the point labeled z_5.

Filter masks used to compute the derivatives needed for the gradient are often called *gradient operators, difference operators, edge operators,* or *edge detectors.*

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

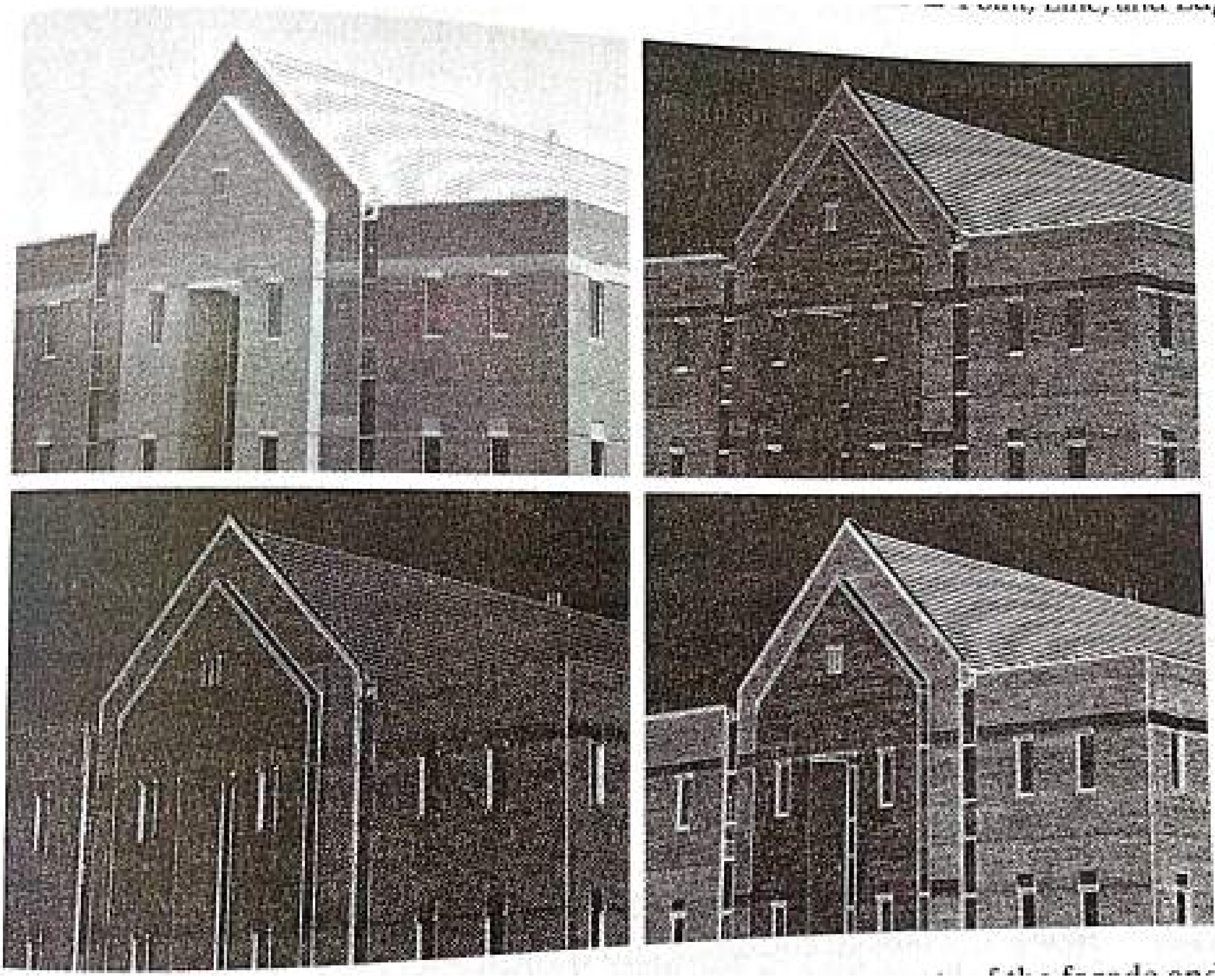
−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel



a b
c d

FIGURE 10.16
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

Sobel mask for detecting diagonal edges

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

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

Second Order Derivative filters

Laplacian Operator :-

- It is a Second order Derivation filter

Diff b/w FOD & SOD.

- In first-order-derivative, edges are considered to be present when edge magnitude is large compared to the threshold value.
- In SOD, edge is present at that location where the SOD is zero
- It's look like zero-crossing {means sign change}

eg: $\underbrace{20 \quad 40}_{20} \; \underbrace{\quad 20}_{-20}$

- Laplacian algo is zero-crossing algo
- Laplacian masks are very sensitive to noise because there is no magnitude checking, even a small ripple looks like edge
- So in this first image is filter than edge detection process is applied
- It's advantage is that its rotationally invariant.
- Laplacian operator ($\partial^2/\partial x^2 + \partial^2/\partial y^2$) for 2-D image

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

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

$x-1 \quad x \quad x+1$

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

$y-1$
y
$y+1$

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

So, Laplacian Operator applied and detect the zero crossing

0	-1	0
-1	4	-1
0	-1	0

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

↓ Inverse sign

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

5x5

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

- Now two differential operator capable of computing a digital image and give better o/p.

Laplacian of Gaussian (Marr-Hildreth) Operator

According to Marr & Hildreth

- Intensity of changes is not independent of image scale
- Sudden intensity change will cause a zero crossing of the SOD.

So, they conclude that

- LOG is capable to tuned at any scale
- LOG is capable to compute the FOD and SOD

- To minimize the noise susceptibility of the Laplacian operator, LOG operator is often preferred.

$\nabla^2 G$

$\nabla^2 \rightarrow (\partial^2/\partial x^2 + \partial^2/\partial y^2)$ — ②

$G(x,y) \rightarrow e^{-\frac{x^2+y^2}{2\sigma^2}}$ — ①

So,

$$\nabla [G_\sigma(x,y) * f(x,y)] = [\nabla G_\sigma(x,y)] * f(x,y)$$
$$= f(x,y) * LOG.$$

eq ① & ②

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

$$= \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] + \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\right]$$

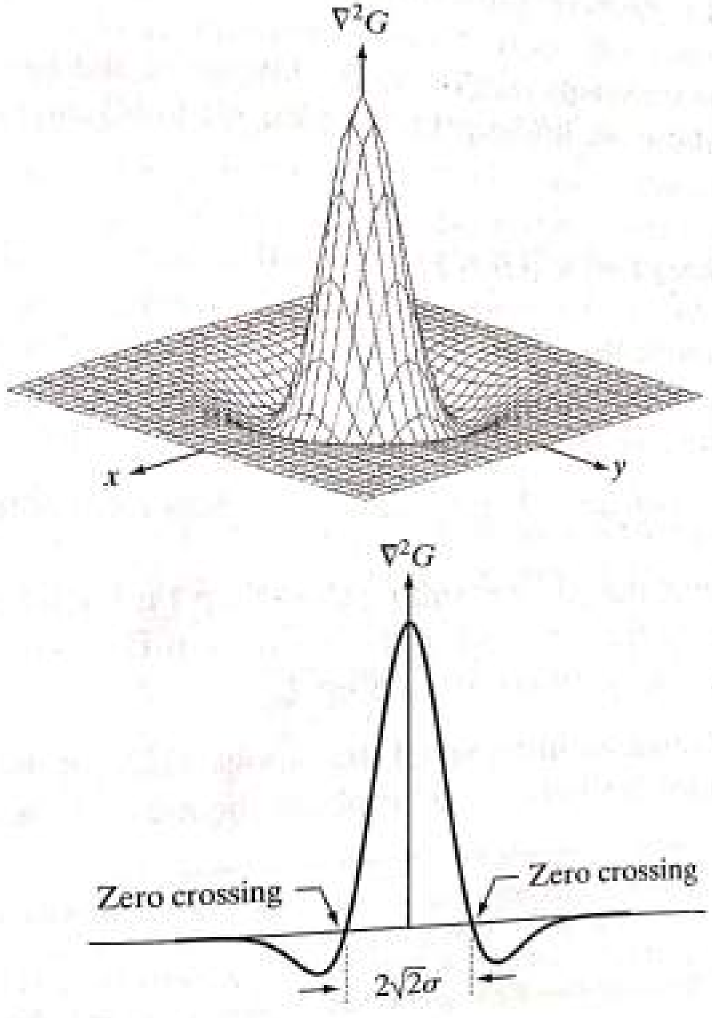
$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

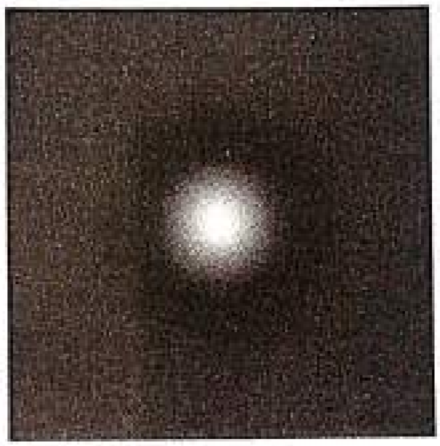
LoG kernel

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

The Marr-Hildreth edge detection algo may be summarized as

1. filter the i/p image with $n \times n$ Gaussian lowpass filter.
2. Compute the laplacian of the image on step I image
3. Find the zero crossing of the image.

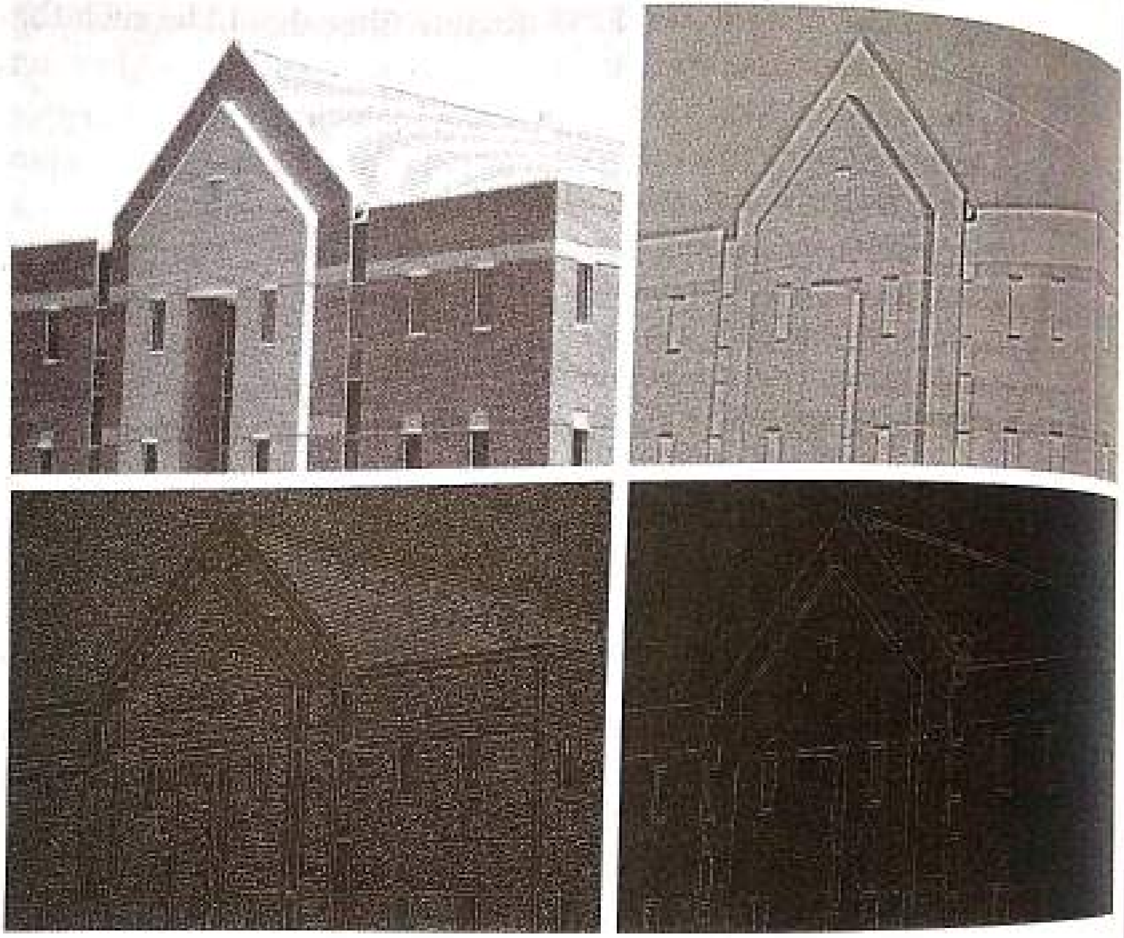




0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.



a b
c d

FIGURE 10.22
(a) Original image of size 834×1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Difference of Gaussian (DOG) filters

The DOG filter can be written as

$$G_{\sigma_1}(x,y) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{x^2+y^2}{2\sigma_1^2}\right)$$

with gaussian width σ_1

The width of the Gaussian is changed and new kernel is

$$G_{\sigma_2}(x,y) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{x^2+y^2}{2\sigma_2^2}\right) \text{ for } \sigma_2 \text{ width}$$

The DOG is diff. b/w these two Gaussian kernel

$$G_{\sigma_1}(x,y) - G_{\sigma_2}(x,y) = (G_{\sigma_1} - G_{\sigma_2}) * f(x,y)$$
$$= DoG * f(x,y)$$

DoG Kernel

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}}$$

with $\sigma_1 > \sigma_2$

if $\frac{\sigma_1}{\sigma_2}$ is b/w 1 & 2 → its a good

DOG algorithm:-

1) Generate mask and apply DOG to image.
2) Detect the zero crossing and apply the threshold to suppress the weak zero-crossing
3) Display

Canny Edge Detector:-

- The canny edge decetor is an edge detection operator that uses a multistage algorithm to detect a wide range of edges in images.
- This algorithm was developed by John F. canny in 1986.
- Canny approach is based on optimization the trade off
- Few factors are used in canny are
  (1) Good edge detection:- It must be capable to detect only real edge points and discard all false edge points
  (2) Good edge localization:- It should have the ability to produce edge points which are close to real edges.
  (3) Only one response to each edge:- It should not produce any false, double, or spurious edges.

Canny Edge Detection Algorithm:→

Step 1 :- Convolve the image with Gaussian filter and compute the gradient of resultant smooth image.

- Store the edge magnitude and edge orientation in two separate arrays.

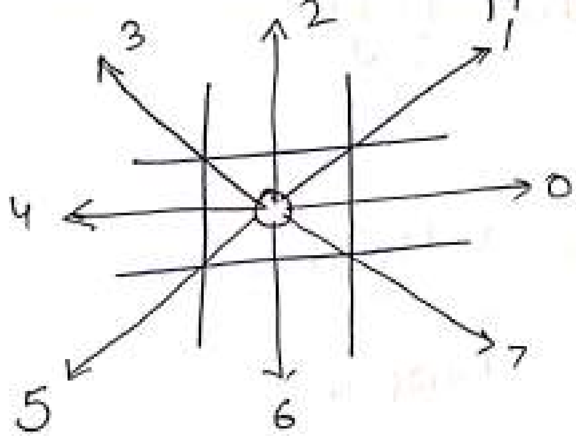
$$S = G_\sigma * I \quad , \quad G_\sigma = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla S = \left[\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}\right] = [S_x \;\; S_y]^T$$

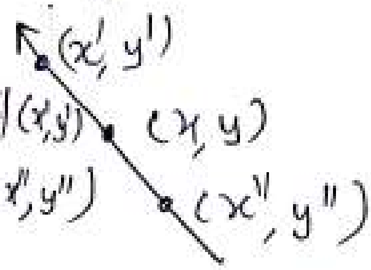
$$|\nabla S| = M(x,y) = \sqrt{S_x^2 + S_y^2}$$

$$\alpha(x,y) = \tan^{-1}\frac{S_y}{S_x}$$

Step 2 :- Thinning of edges by a process known as Non-maxima suppression



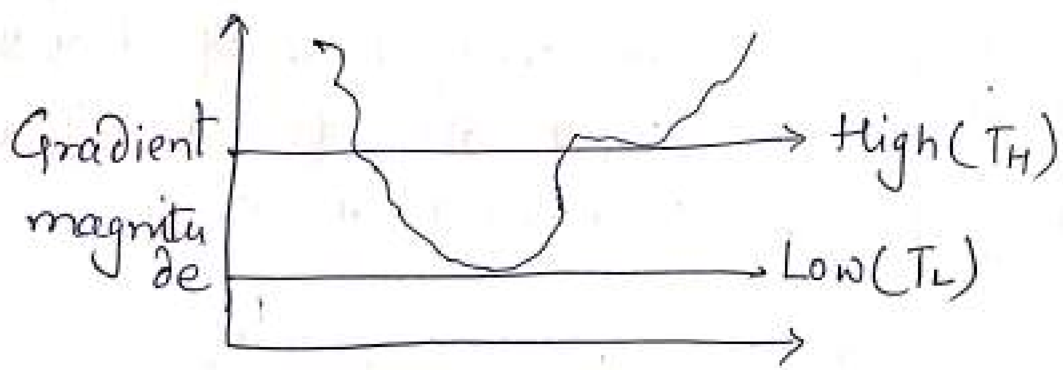
Let (x',y') and (x'',y'') are the neighbors of $f(x,y)$. So $|\nabla S|$ along the direction normal to an edge



$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\nabla S|(x',y') \;\&\; |\nabla S|(x,y) > |\nabla S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$

Step3:- Apply Hysteresis thresholding (Reduce false edge points)
(Used because only a large amount of change is gradient magnitude)

- In general, we use only one threshold value, but in canny we are using two threshold value Low(T_L) and high(T_H) and follow below three condition.
  - If the gradient at a pixel is above 'High' declare it an edge pixel.
  - If the gradient at a pixel is below 'Low', declare it a non-edge pixel
  - If the gradient at a pixel is b/w 'Low' and 'High' then declare it an 'edge pixel' if and only if it is connected to an 'edge pixel' directly or via pixels b/w 'Low' and 'High'.



a b
c d

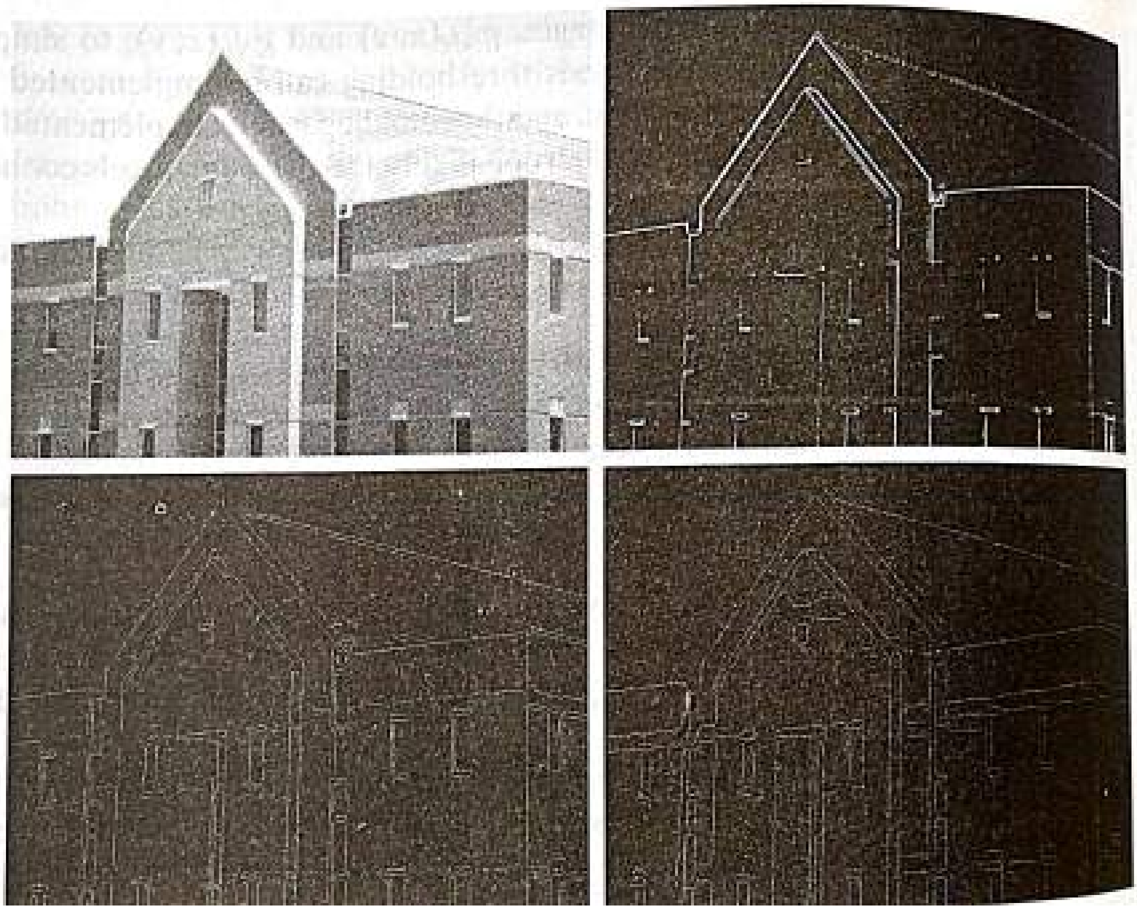
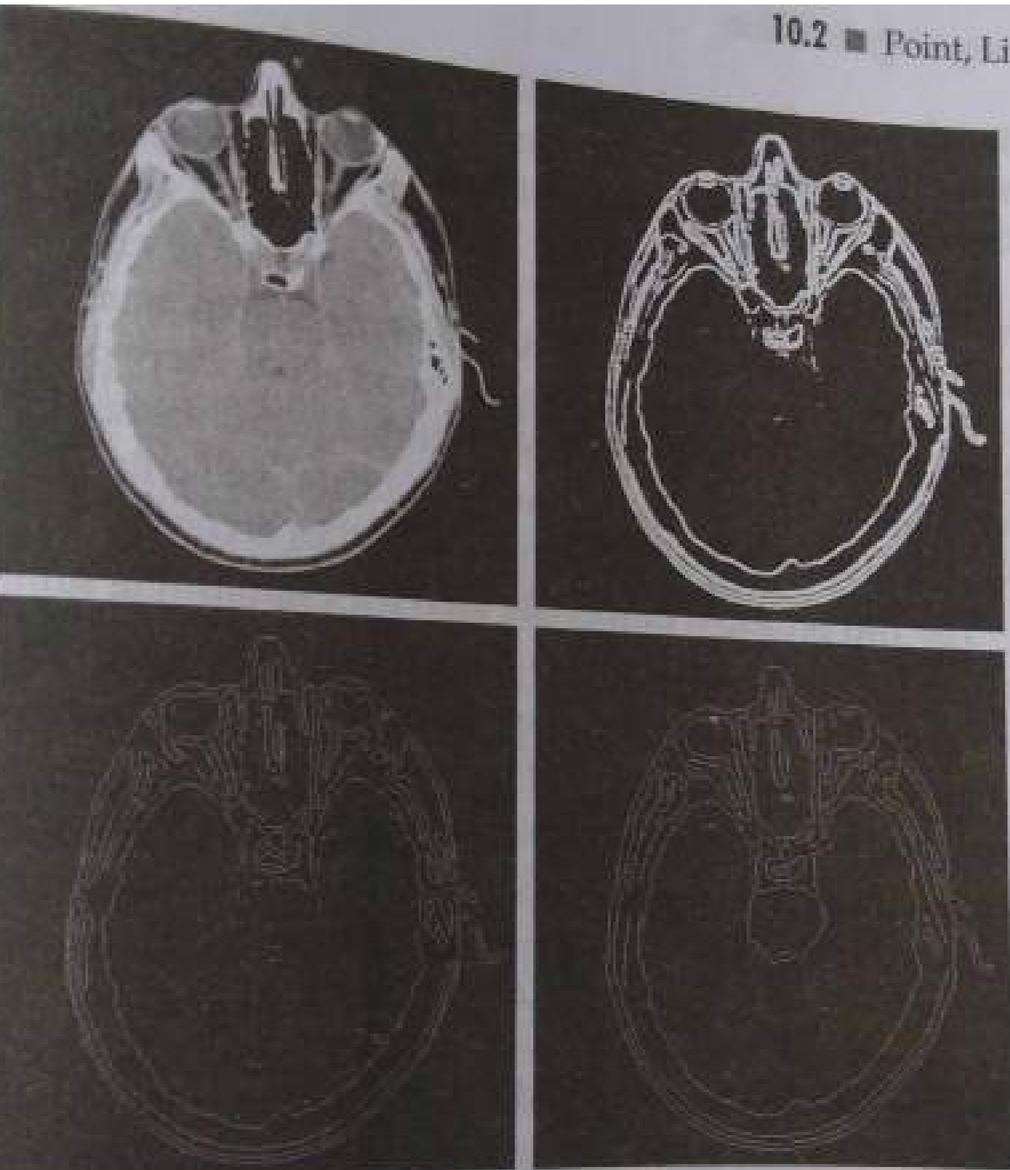


FIGURE 10.25 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the



a b
c d

FIGURE 10.26 (a) Original head CT image of size 512 × 512 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

and a mask of size 13 × 13, which, as in the

Thresholding :-

- Thresholding is an important technique for image Segmentation.
- It produces uniform regions based on the threshold criteria, T
- Key parameter of thresholding process is the choice of threshold value
- There are different type of thresholding

① Global Thresholding :-

If thresholding operation depends upon only grey scale value, it is known as GT or constant applicable over an entire image.

② Variable Thresholding :/Local / regional thresholding

When thresholding value change over an image. or In case neighborhood properties (or some local properties) is also taken into account.

③ Dynamic / Adaptive thresholding

If case T depends on pixel coordinates also. or on spatial coordinates (x,y)

So, from these three type of thersholding, we cas say that

(1) Thresholding is a func of spatial coordinats i.e (x,y)

(2) It is a grey level of the pixel ie. $f(x,y)$

(3) Some local property of the image i.e $A(x,y)$

Therefore, thresholding operation can be expressed as:

$$Th = T[x, y, f(x,y), A(x,y)]$$

Histogram & Thresholding :-

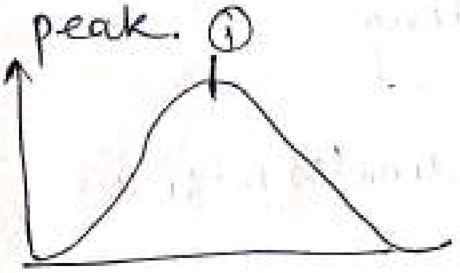
- The quality of thresholding algorithm depends on the selection of a suitable threshold.
- Tool that helps to find the threshold is histogram

Types of Histograms

→ Unimodel → Bimodal → Multimodel

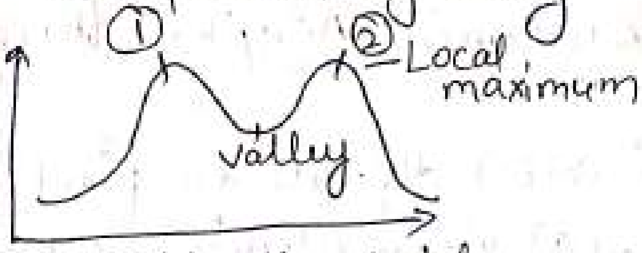
Unimodel

- if a Histogram have one central peak.



Bimodal

- if a Histogram have two peaks & seprated by valley

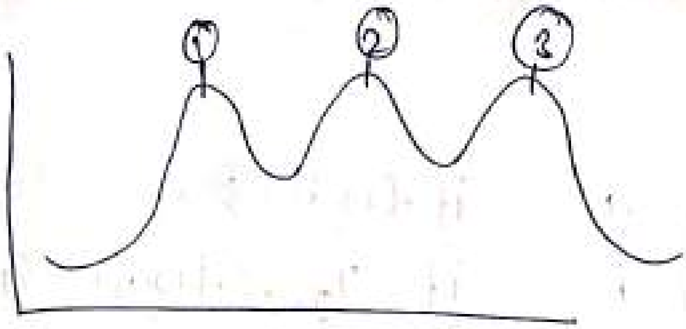


suitable threshold value with the help of valley separate the two peaks

→ very easy and because valley can be identified easily.

Multimodal Histogram

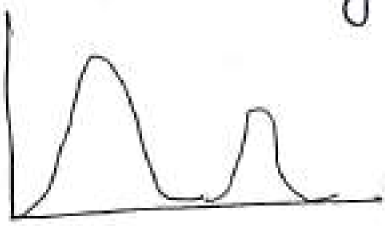
- if a histogram have ~~two~~ ~~peaks~~ more than two peaks.



- Suppose we have image $f(x,y)$ is composed of light objects on a dark background, and the histogram of the image is shown

① Single level thresholding

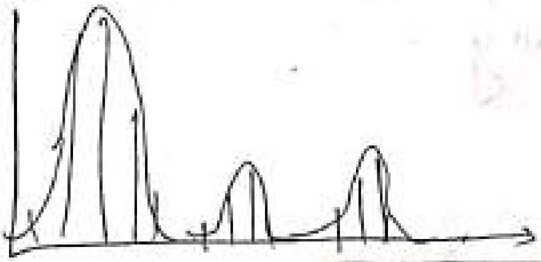
- The objects can be extracted by comparing pixel values with a threshold T.
- for any particular point (x,y) (object point) the segmented image $g(x,y)$ is (b.g point)



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

② Multi level Thresholding

- It is also possible to extract objects that have a specific intensity range using multiple thresholds.



Histogram shows that three peaks are there. Two types of light objects on a dark background.

- Here multiple thresholding ~~catas~~ classifies a point (x, y)

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) > T_2 \quad \text{— second object.} \\ b & \text{if } T_1 < f(x,y) \leq T_2 \quad \text{— one object} \\ c & \text{if } f(x,y) \leq T_1 \quad \text{— background} \end{cases}$$

a, b, c are three distinct intensity values.

- So we can say that the sucess of intensity thresholding is directly related to the width and depth of the valley(s) seprating the histogram modes.

- factors affecting the properties of the valley(s) are

(1) Sepration b/w peaks

(2) the noise content in the image.

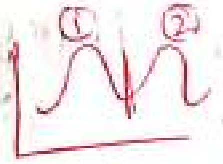
(3) Relative size of objects and background.

(4) Uniformity of the illumination source

(5) Uniformity of the reflectance properties of the image.

Global Thresholding:-

- In bimodal images, histogram have two distinct peaks seperated by a valley b/w them.
- So, valley point is chosen as threshold (T)
- Then pixels of image are compared with threshold value.



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

The basic global threshold (T) is calculated using following algorithm:-

(1) Randomly select an initial threshold T.

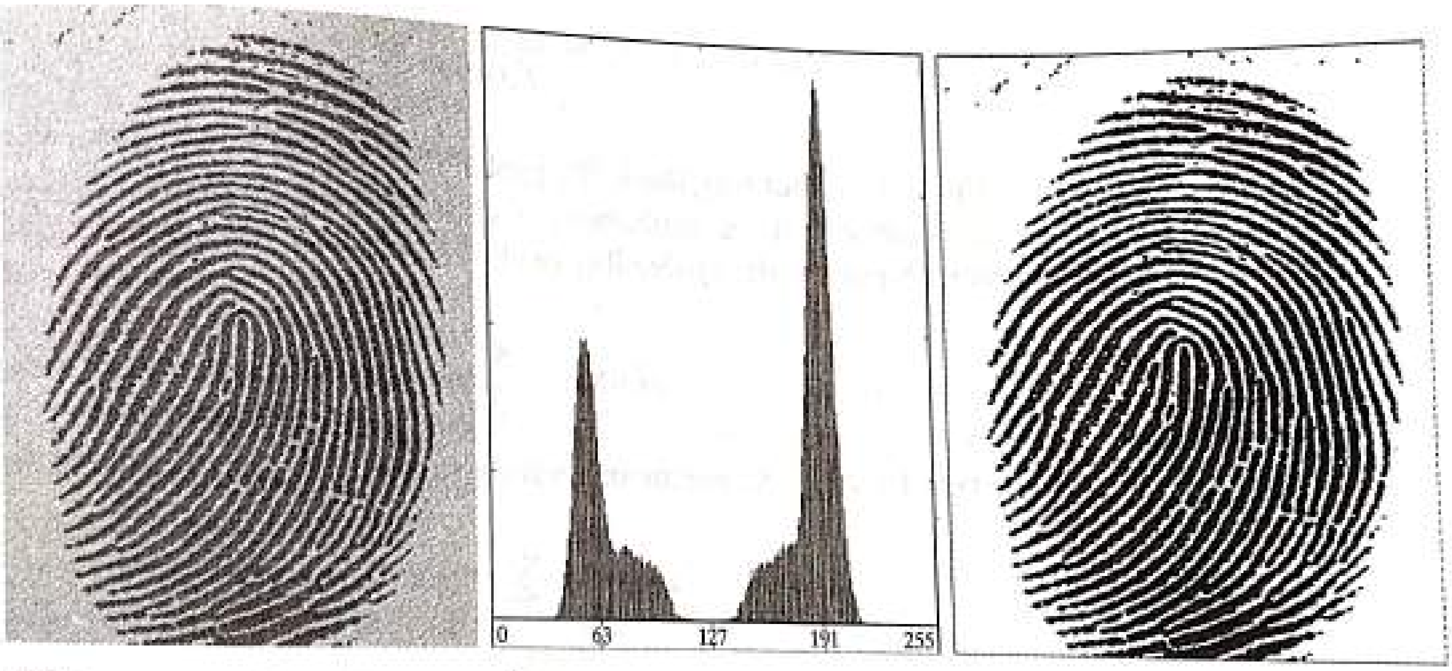
(2) Segment the image using equation to produce two groups $G_1 (>T)$ and $G_2 (\leq T)$.

(3) Compute the average (mean) intensity values m_1 and m_2 for the pixels in G_1 and G_2 groups respectively.

(4) Compute a new threshold value

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

(5) Repeat steps 2 to 4 until the difference b/w values of T in sucessive iterations is smaller than a predefined parameter ΔT.



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Multiple Thresholding:-

- Multilevel thresholding is a process that segments a gray level images into several distinct regions.
- It is the extension of simple thresholding technique.
- Let f is an image and f_i - are input image pixel value
- $t_1, t_2 \ldots t_n \rightarrow$ multiple threshold values
- $g_1, g_2 \ldots g_n \rightarrow$ values of o/p images

$$\text{output images} = \begin{cases} g_1 & \text{if } 0 \le f_i \le t_1 \\ g_2 & \text{if } t_1 < f_i \le t_2 \\ \vdots & \vdots \quad \vdots \\ g_n & \text{if } t_{n-1} < f_i < 255 \end{cases}$$

Adaptive Thresholding:-

- Adaptive thresholding changes the threshold dynamically over the image.
- It is also known as Dynamic thresholding algorithm
- Adaptive thresholding works well in situations where image is affected by non-uniform illumination problem.

- Many thresholding algorithm are applied for the segmentation
- But there are two techniques to apply thresholding algorithm

① Divide image into many overlapping sub-images
- draw histogram of all sub images
- determine local minimum threshold

- So, it is complex process

② Split the image into many sub regions
- no overlapping in subimages
- some useful image statistics is applied
  - Mean + C   where C → constant
  - Median + C
  - (Min + Max)/2

Optimal Otsu Thresholding Algorithm :- (when boundries are not clear)

- Optimal thresholding is used where there is a considerable overlapping of the histogram.
- It uses a merit fun^c of either maximaization or minimization to determine the optimal threshold.
- Optimal threshold take the value from the image (in b/w class variance)

eg. Let us assume that there are two classes:-

	Class 1	Class 2
weight	$w_1(t)$	$w_2(t)$
variance	$\sigma_1^2(t)$	$\sigma_2^2(t)$

- So weighted sum of variance is

$$\sigma_w^2(t) = w_1(t)\,\sigma_1^2(t) + w_2(t)\cdot\sigma_2^2(t)$$

- Threshold with the maximum between class variance also has minimum within class variance.

- Therefore equation can be written as

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t)$$

Using the fact that

$$\mu = w_b \mu_b + w_f \mu_f$$

- Relation can be reduced to

$$w_1(t)\, w_2(t)\, [\mu_1(t) - \mu_2(t)]^2$$

- It leads to faster computation.

Let $f(x,y)$ is a grey-scale image, intensity value $(0, L-1)$

- $\{0,1,2,\ldots L-1\}$ — distinct intensity levels $M \times N$ pixels.
- n_i - denotes the no. of pixels with intensity i,
- calculate PDF $P_i = n_i / M.N$
- Now suppose select threshold $T(k) = k$

$$0 < k < L-1$$

and threshold the input image into two classes C_1 and C_2

C_1 — consist of all pixels $[0, k]$

C_2 - consist of all pixels $[k+1, L-1]$

class C_1 $P_1(k) = \sum_{i=0}^{k} P_i$ class C_2 $P_2(k) = \sum_{i=k+1}^{L-1} P_i = 1 - P_1(k)$

- mean intensity of class C_1

$$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^{k} i P_i \qquad m_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i P_i$$

- average intensity of the entire image (global mean)

$$m_G = \sum_{i=0}^{L-1} i P_i \,/\, P_1 m_1 + P_2 m_2$$

(Global variance)

- calculate $\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 P_i$

between class variance $\sigma_B^2 = P_1 (m_1 - m_G)^2 + P_2 (m_2 - m_G)^2$

Optimal Otsu Thresholding Algorithm

① Load the image.

② form the histogram and normalize it, $P_i = i = 0, \ldots L-1$

③ Compute the cumulative sums $P_1(k)$ and $P_2(k)$ by calculating the probability of every level.

④ Compute the cumulative means $m(k)$

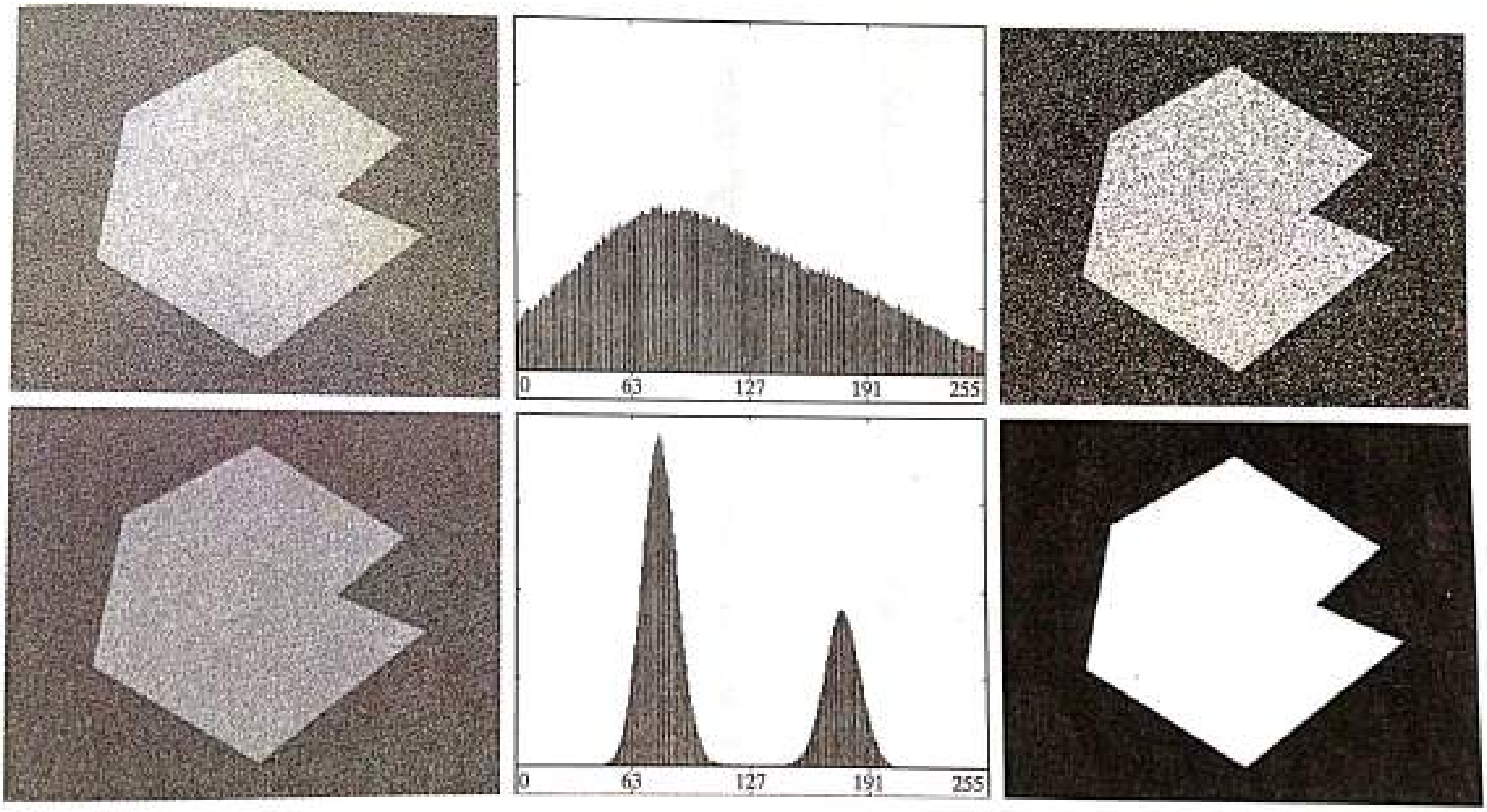
⑤ Compute the global intensity mean, m_G

⑥ Compute the b/w-class variance $\sigma^2_B(k)$ $k = 0, 1, 2 \ldots L-1$

⑦ obtain the Otsu threshold k^*, as the value of k for which $\sigma^2_B(k)$ is maximum.

$$\eta(k) = \frac{\sigma^2_B(k)}{\sigma^2_G}$$

⑧ obtain the separability measure, η^* at $k = k^*$.



a b c
d e f

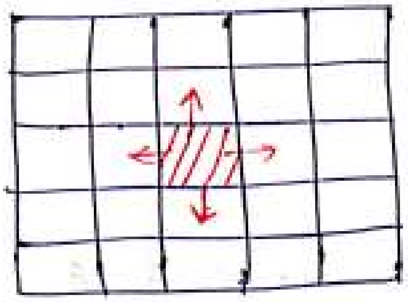
FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Region-Based Segmentation :-

Region Growing :-

- Region growing is a procedure that groups pixels or sub regions into large regions based on predefined criteria for growth.
- logic behind region growing is the principle of similarity.
- Principle of similarity states that a region is coherent if all pixels of that region are homogeneous
- Homogeneity of regions is used as the main segmentation criterion in region growing
- Characteristic (criteria) on which homogeneity depends are gray level, color, texture, shape, model etc.
- Choice of criteria affects segmentation results dramatically.
- Majors steps of the region growing algorithm are.
  (1) Selection of the intial seed
  (2) Seed growing criteria
  (3) Termination of the segmentation process

eg:



5x5 (up, down, left, right)

→ flow of region growing algorithm

- Groups pixels into large regions
- Starts with a seed regions
- Grows regions by merging neighboring pixels.

Iterative process
- How to start?
- How to iterate?
- When to stop?

Example:- Lets have image

1	0	7	8	7
0	1	8	9	8
0	0	7	9	8
0	1	8	8	9
1	2	8	8	9

- Take two seeds points S_1 & S_2
- Threshold value, $T = \leq 4$

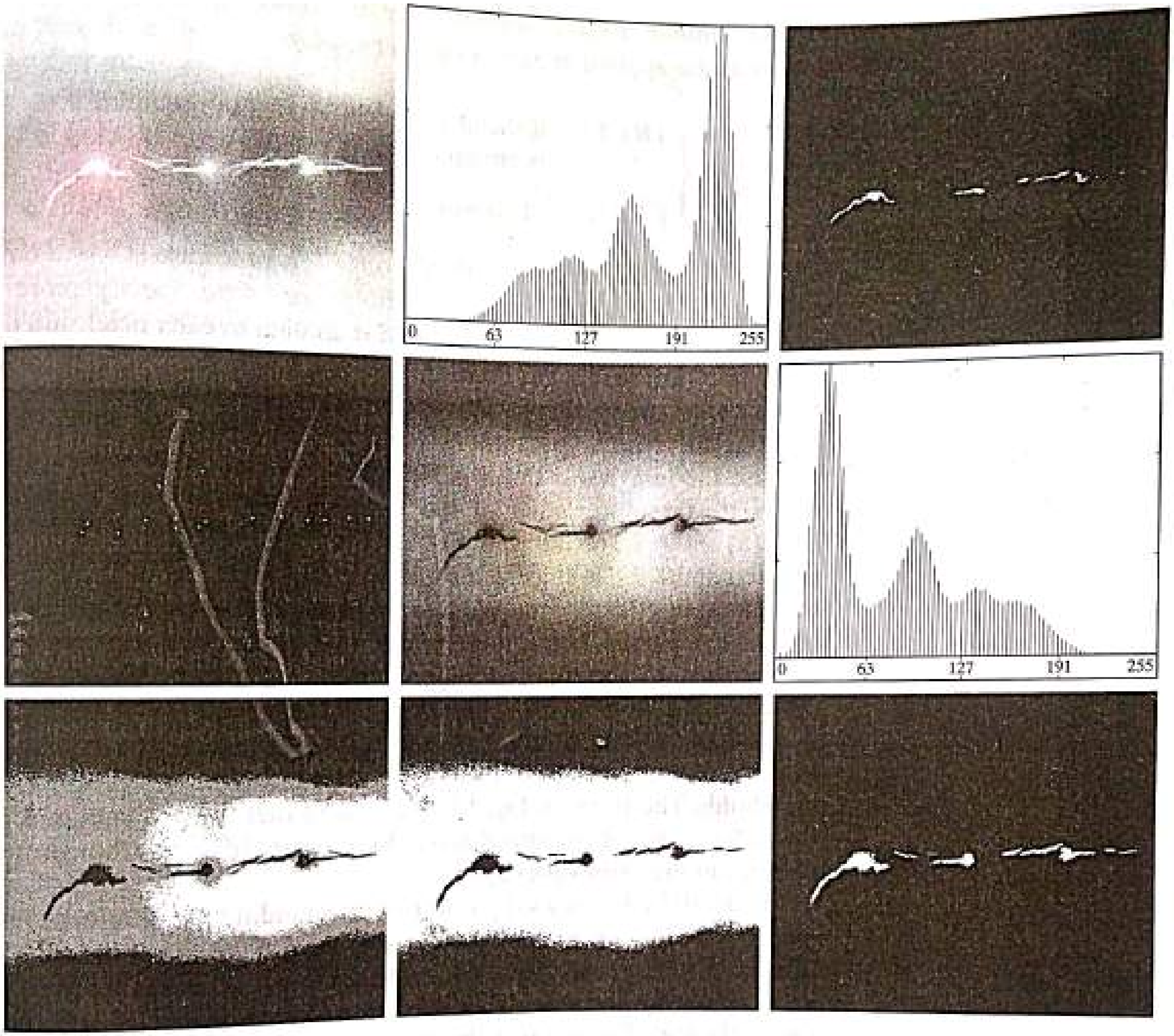
for $S_1 = 9$ $|f(x,y) - f(x',y')| \leq 4$

$|f(x,y) - 9| \leq 4$

$\hookrightarrow \{5, 6, 7, 8, 9\}$ — (A)

for $S_2 = 1$

$|f(x,y) - 1| \leq 4 \rightarrow \{1, 2, 0, 3, 4, 5\}$ — (B)



a b c
d e f
g h i

FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference image thresholded with the smallest of the dual thresholds. (i) Segmentation result obtained by region growing. (Original image courtesy of X-TEK Systems, Ltd.)

Region - Based Segmentation. -

- Segmentation techniques are based on finding the regions directly.
- It is an alternative method of image segmentation
- An image is sub-divided into arbitrary disjoined region.
- Arbitrary regions can be split and merged in order to satisfy the condition.
- There are two phases

Split — if homogenetic is there, no need to Split, else split

Merge — After spiliting regions are merged if they share same properties

OR

Phase 1:- $P(R_i) = \text{false}$

- Split & Continue subdivision process until some stopping criteria is fullfilled
- Often it is stopped when no further spitting is possible

Phase II:- $P(R_i \cup R_j) = \text{True}$

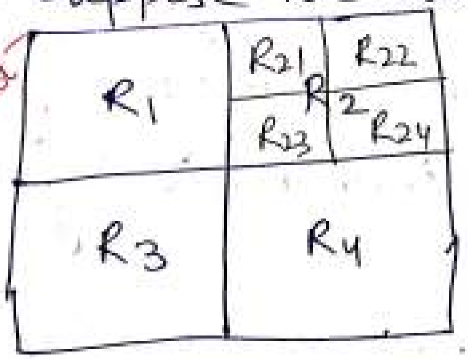
- Merge adjacent regions if the regions share any common criteria
- stop the process when no further merging is possible

- It is known as <u>Divide & Conquer</u> or <u>Top-down</u> approach

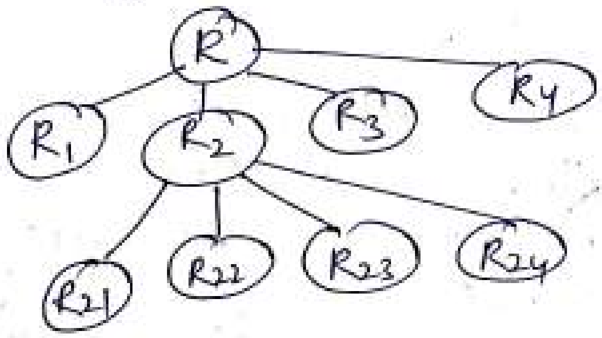
- Suppose we have an image, divide it into four quadrants R_1, R_2, R_3, R_4

Subdivide R_2 in four quadrants again

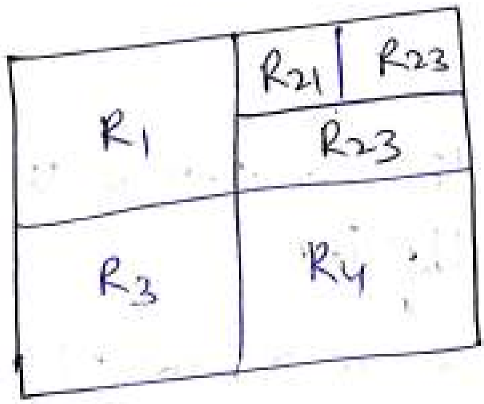
Split



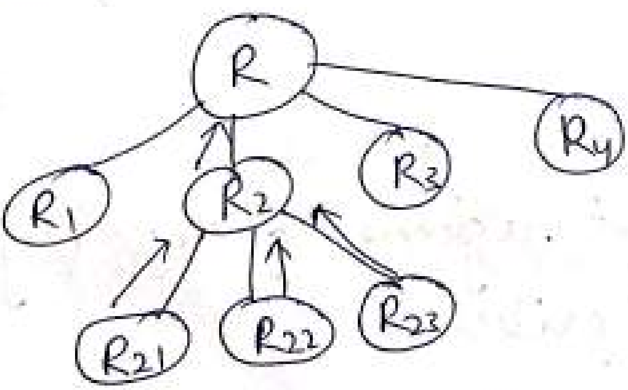
- Corresponding quadtree is generated



Merge



- Region R_{23} & R_{24} are merged due to the similar criteria



Eg:- Apply split and merge algo.

6	5	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

Ans:- first chose threshold value. ≤ 3

max Pixel = 7
min Pixel = 0] diff = $7 \nleq 3$

- divide it into fours quadrants

6	5	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

① (top-left), ② (top-right) ✓ Split, ③ (bottom-left), ④ (bottom-right) ✓ Spilit

① Quadrant = max = 7, min = 4 = $7-4 = 3 \leq 3$ True - No further Spitiling

② Quadrant = max = 7, min = 2 $7-2 = 5 \leq 3$ false

→ More Spiliting

③ Quadrants $max = 3$, $min = 0$ · $3-0=3 \leq 3$ No more Spliting

④ Quadrants $max = 7$, $min = 0$ $7-0=7 \leq 3$ more Spliting

6	5	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

$7-5=2 \leq 3$ ✓

$7-4 \leq 3 \leq 3$ ✓

$6-4=2 \leq 3$ ✓

$3-2=1 \leq 3$ ✓

$3-2=1 \leq 3$

$7-4=3 \leq 3$

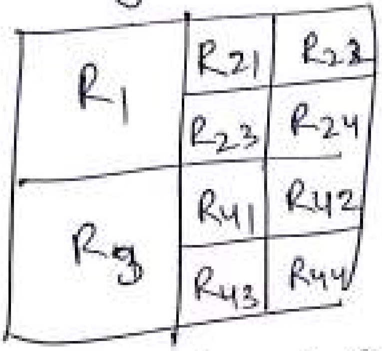
$5-4=1 \leq 3$

$3-0=3 \leq 3$

all are satisfying the condition so further Splitting

Spliting

Merge



in Region ① & ③

$$\left.\begin{aligned} max(1) - min(3) &\Rightarrow 7-0=7 \\ max(3) - min(1) &\Rightarrow 3-7=5 \end{aligned}\right\} \leq 3 \rightarrow \text{not true}$$

So, no merging

Now, R_1 & $R_{21} \rightarrow \left.\begin{aligned} 7-4 &= 3 \leq 3 \\ 7-5 &= 2 \leq 3 \end{aligned}\right\}$ True

merge R_1 & R_{21}

Now R_{21} & $R_{22} \rightarrow$ merge

Now R_1 & $R_{23} \rightarrow$ no merging

Now R_{23} & R_{24} → not satisfying → no merging

R_3 & R_{41} → merged

R_3 & R_{43} → merged

R_{41} & R_{42} → not merged

R_{44} & R_{43} → not merged

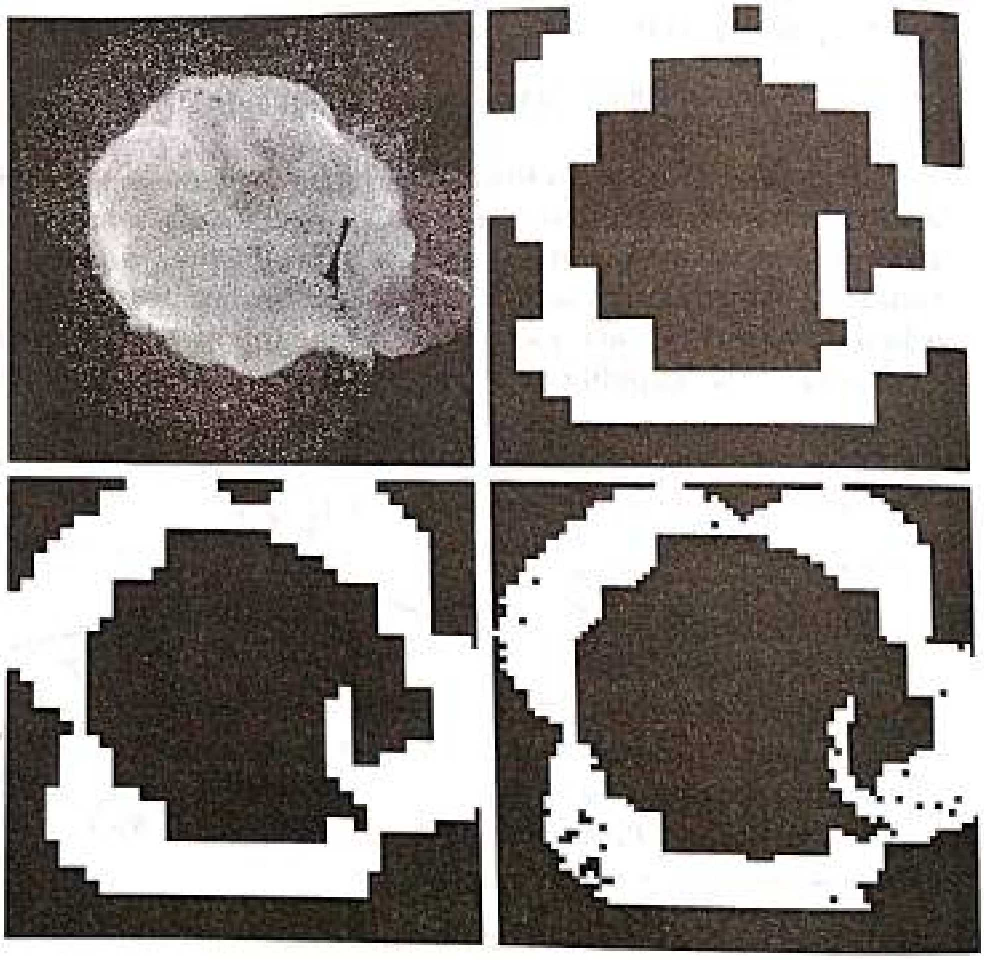
R_{42} & R_{44} → merged.

6	5	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

Getting two regions after splitting and merging.

So, steps in Split & merging algo

1. Split into four disjoint quadrants for which $Q(R_i) = FALSE$.
2. When no more splitting is possible, merge any adjacent regions R_j and R_i for which $Q(R_i \cup R_j) = $ True
3. Stop when no further merging is possible



a b
c d

FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

Hough Transform :-

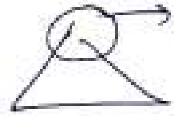
- Hough transform is a feature extraction method for detecting simple shapes such as circles, lines etc. in an image.

- The objective of hough transform is to connect the disjoint edges

or

Hough transform takes the images created by edge detection operators but most of the line, edge map is disconnected.

- for example if triangle is detected and this



point is missing. With the help of hough transform we can draw the edge.

- Therefore hough transform is used to connect the disjoined edge points.

→ Equation of line

$$y = mx + c$$

m - slope

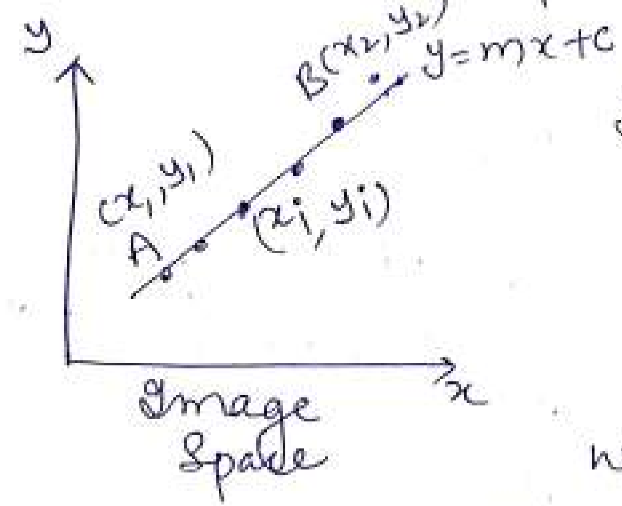
c - intercept of the line

- So with the help of (x, y) and line equation we can draw many lines for different values of m & c.

now

$$c = -mx + y$$

and consider c-m plane (parameter space)



So for point (x_i, y_i) we can write

$$y_i = mx_i + c$$

or

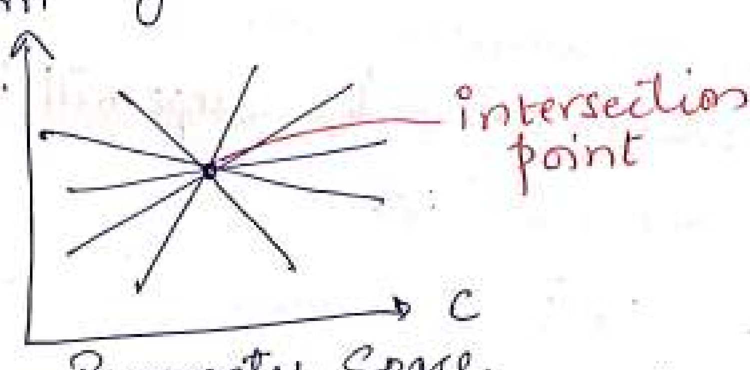
$$c = -x_i m + y_i$$

we will draw c-m plane

Now points $A(x_1, y_1)$ and $B(x_2, y_2)$ can also fit in c-m plane

→ So if we will draw A, B in c-m plane than A, B, (x_i, y_i) points will intersect with each other in same plane.

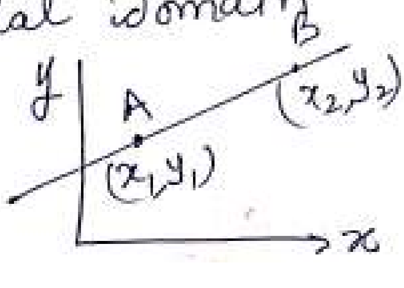
- It means they are points of a single line. (same)

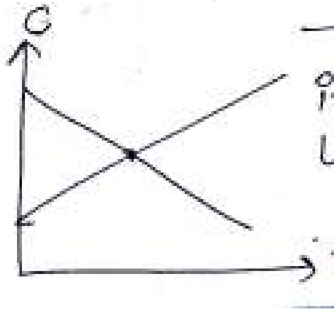


Parameter Space

- Parameter Space is known as hough space

- If A & B are two points connected by a line in spatial Domain





They will be intersecting lines in Hough Space

Hough Transform Algorithm:-

(1) Load image
(2) Determine the image edges using any edge Detector (gradient, canny, LOG etc.)
(3) Quantize the Parameter Space, P
(4) Repeat the process for all pixels of image
If the pixel is an edge pixel, then

$$c = -xm + y$$

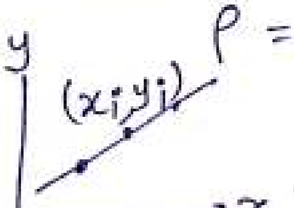
$$P(c,m) = P(c,m) + 1$$

(5) Show the Hough Space
(6) find the local maxima in parameter space
(7) Draw the line using local maxima

- There is a limitation with this algorithm, if there is a vertical line its slope will be infinity and this algorithm will not work
- Therefore this line must be converted into polar cordinates

$$\rho = x \cos\theta + y \sin\theta$$

Where θ = angle b/w the line
ρ = diameter



Hough Space

For this modified Hough transform, all steps are same except (4)

all steps are same

(4) Repeat the process for all pixels of image

If the pixel is an edge pixel, then for all θ

- Calculate ρ for pixel (x,y) and θ
- Increment the position (ρ,θ) in accumulator array.

Eg:- Using the hough transform show that the points (1,1) (2,2) and (3,3) are collinear and find the line equation.

<u>Solu:-</u> line eq: $y = mx + c$

- Convert (x,y) plane into $(m-c)$ plane

$$C = -mx + y$$

1. for (1,1) $\quad C = -m + 1$

if $c = 0$, $m = 1$

if $m = 0$ $\quad c = 1$

$\therefore$ Thus $(m,c) = (1,1)$

(2) for (2,2)

$$c = -2m + 2$$

if $c = 0$ $\quad 2m = 2$

$m = 1$

$m = 0$ $\quad c = 2$

Thus $(m, c) = (1, 2)$

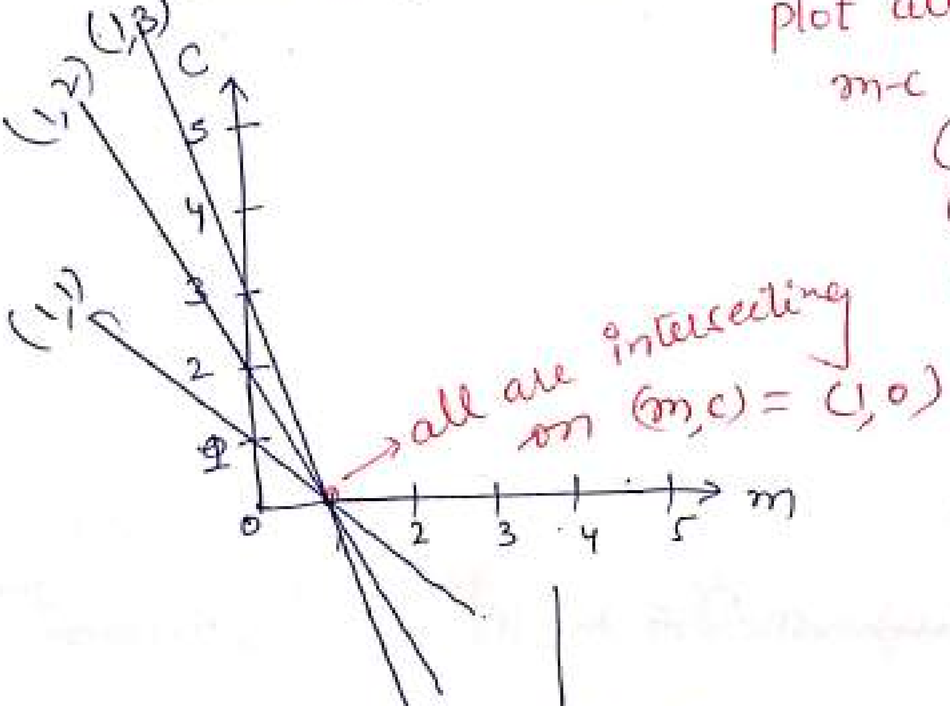
(3) for (3,3)

$$c = -m3 + 3$$

if $c = 0$ $\quad m = 1$

if $m = 0$ $\quad c = 3$

Thus $(m, c) = (1, 3)$



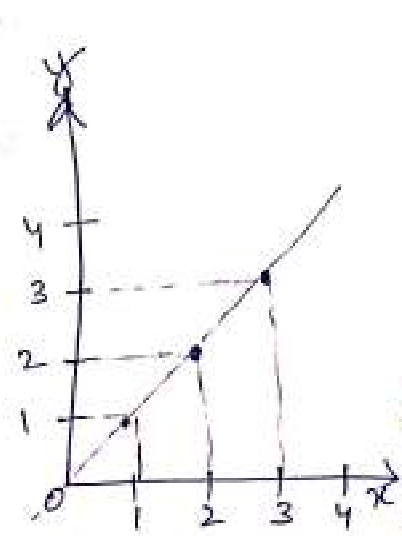
plot all values in m-c plain
(1,1)
(1,2)
(1,3)

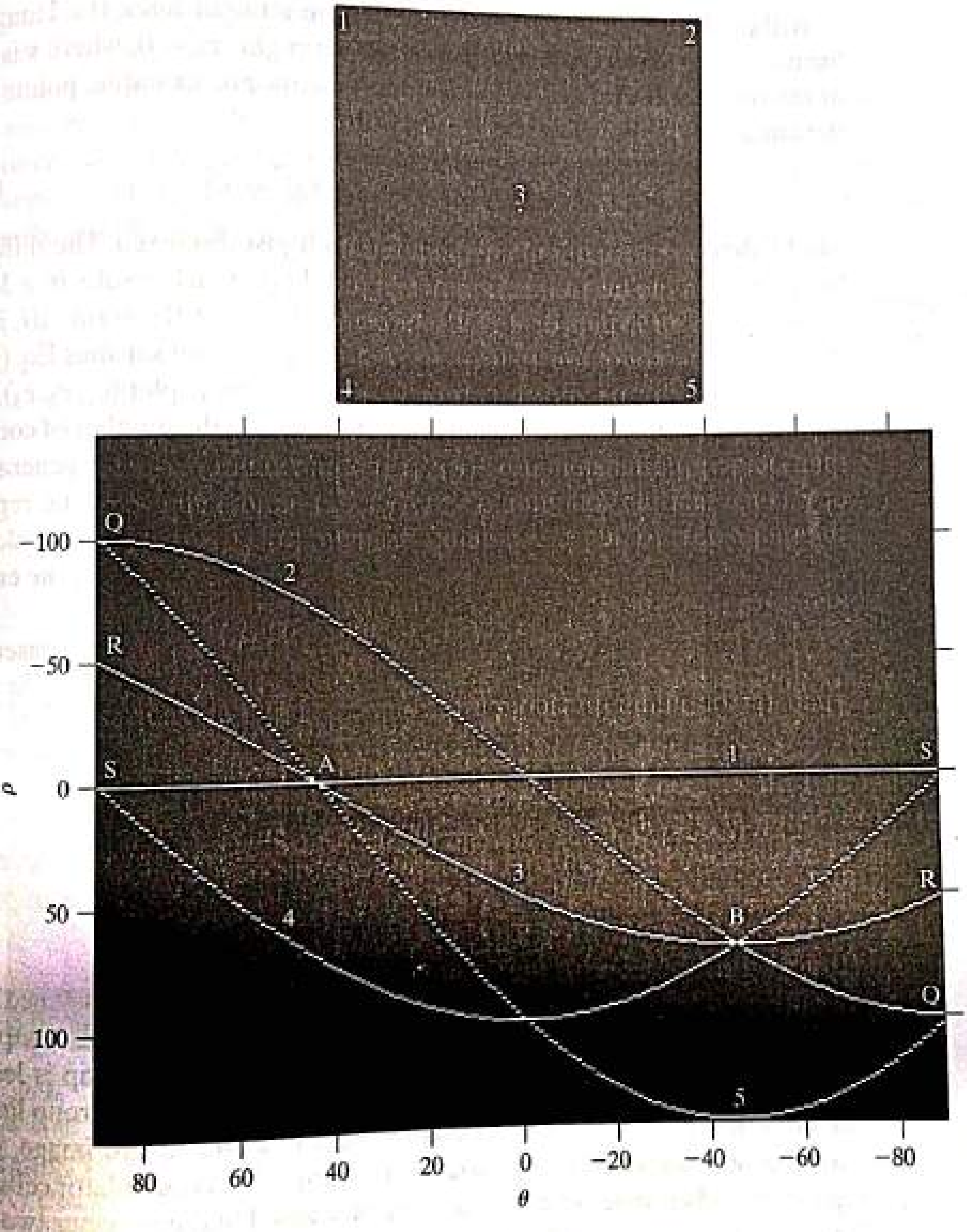
original equation of line

$y = mx + c$ —— put (1,0)

equation of line $\boxed{y = x}$

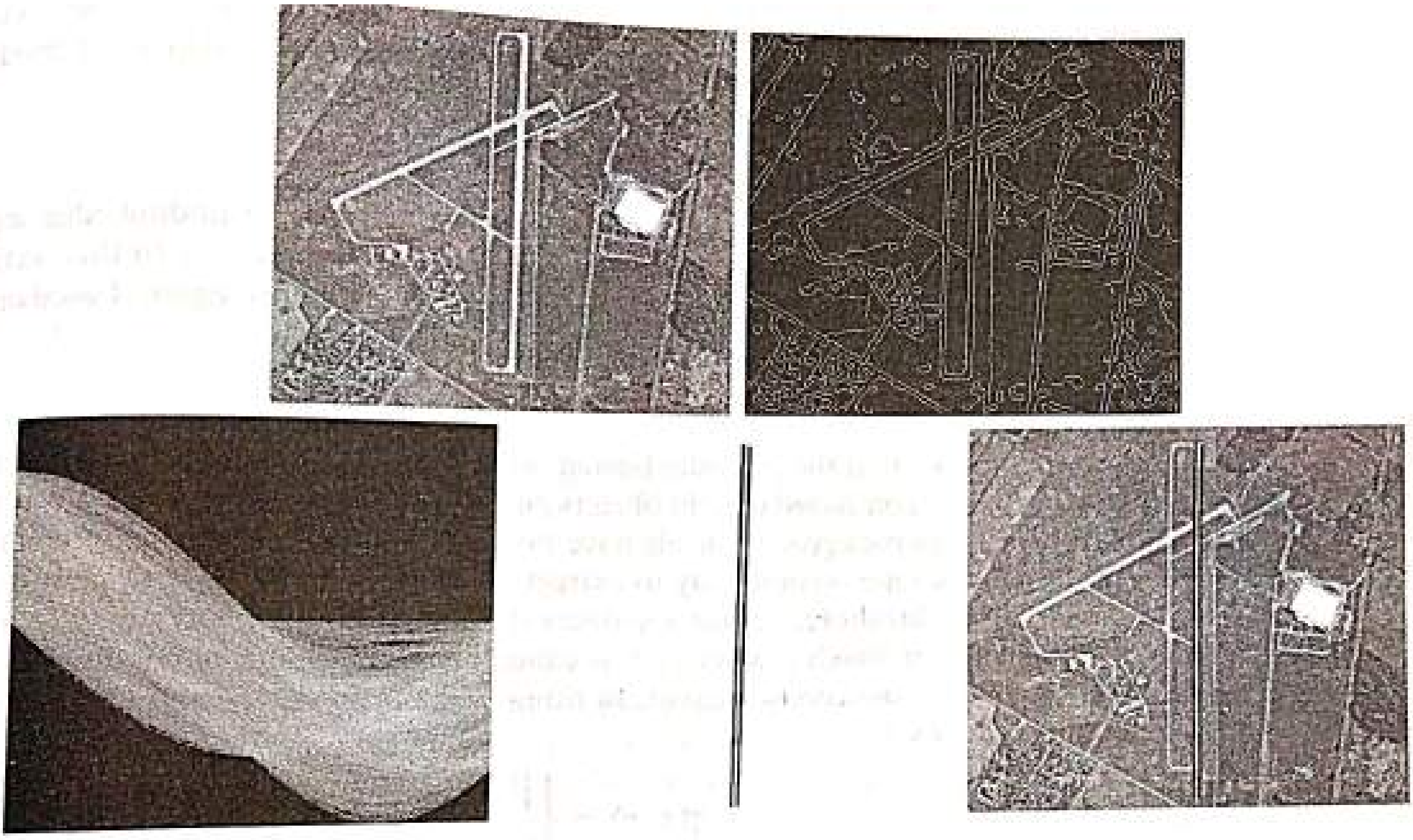
All points are ~~collinear~~





a
b

FIGURE 10.33
(a) Image of size 101 × 101 pixels, containing five points.
(b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)



a b
c d e

FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed on the original image.

Edge Linking & Boundary Detection :-

In edge detection methods we find that edges are detected after these methods.

In few cases pixels are not identified as edges due to noise, breaks in edges due to non uniform illumination and discontinuitie in intensity values due to other effects.

- Edge linking algorithms are designed to assemble edge points (pixels) into meaningful edges or region boundaries.

- Three fundamental approaches are there
  (1) Local processing (3x3 neighborhood)
  (2) Boundary of region \ Regional Processing
  (3) Global approach (entire edge image)

① Local Processing :-

- For linking edge, analyze the characteristics of pixels in a small neighborhood
- Two principal properties are used to establish similarity of edge pixels
  ① Strength (Magnitude)
  (2) Direction of gradient vector

Edge Linking & Boundary Detection :-

In edge detection methods we find that edges are detected after these methods.

In few cases pixels are not identified as edges due to noise, breaks in edges due to non uniform illumination and discontinuitie in intensity values due to other effects.

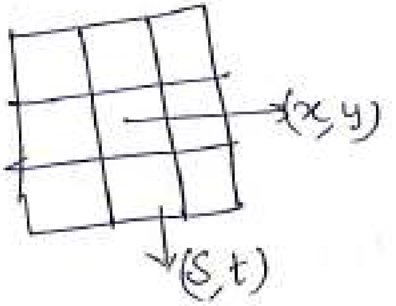
- Edge linking algorithms are designed to assemble edge points (pixels) into meaningful edges or region boundaries.

- Three fundamental approaches are there
  (1) Local processing (3x3 neighborhood)
  (2) Boundary of region \ Regional Processing
  (3) Global approach (entire edge image)

① Local Processing :-

- for linking edge, analyze the characteristics of pixels in a small neighborhood
- Two principal properties are used to establish similarity of edge pixels
  ① Strength (Magnitude)
  (2) Direction of gradient vector

Let



(x, y) is centered point &
(S,t) is neighborhood coordinates

So, they will be similar if $|M(s,t) - M(x,y)| \leq E$

E is positive threshold.

- for angle $|\alpha(s,t) - \alpha(x,y)| \leq A$, A positive angle threshold.
- So, a pixel (s,t) is linked to the pixel at (x, y) if both magnitude and direction criteria are satisfied.
- It's computationally expensive because all neighbors of every point have to be examined.

So steps of this process are

(1) Compute the gradient magnitude and angle array $M(x,y)$ & $\alpha(x,y)$ of the input image $f(x,y)$.

(2) form a binary image g whose value at any pair of coordinates (x,y) is given by

$$g(x,y) = \begin{cases} 1 & \text{if } M(x,y) > T_m \ \& \ \alpha(x,y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

(3) Scan the rows of g and fill (1) all gaps (0) in each row

(4) Detect gap in θ direction, rotate g by θ and apply horizontal scan and, rotate back by −θ for result.

Representation & Description :-

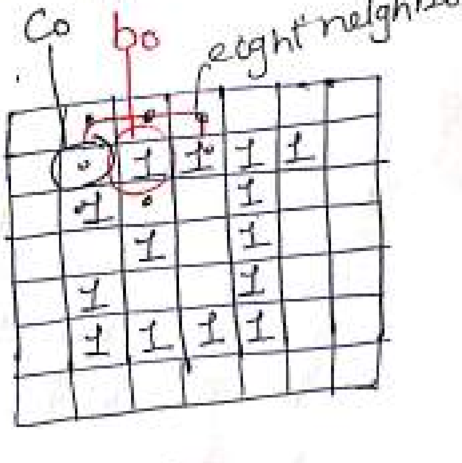
- After an image has bee Segmented into regions by thresholding, edge detection or region growing.
- We can represent the regions in term of
  - (1) External characterstics (boundary)
  - (2) Internal characterstic (region)

Shape characteristics -
- length
- orientation of the straight line

regional properties
(1) color, texture

(1) Boundary following :- (o/p is an ordered seq. of points.)
~~(Moore Boundary tracking~~ Algorithm)

- We are working with binary image, object → 1, background - 0
- Image is padded with a border of 0's
- Binary image R, steps are



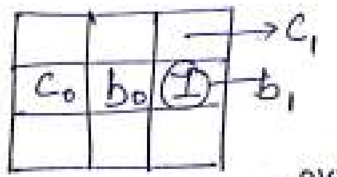
(1). Define starting point (b_0) → 1 uppermost, left most

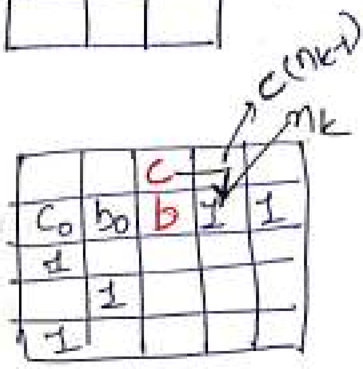
- West point of b_0 → c_0 (always background point)
- Examine 8neighbors of b_0, starting from c_0 in clockwise direction.

- Now trace the neighbors of b_0 → start from c_0 and assign b_1 to first 1 value pixel

- c_1 is the point (background) immediately preceding b_1 in the sequence

- Store the location of b_0 and b_1 for step 5.





2. Let $b = b_1$ and $c = c_1$
3. Let the 8-neighbors of b, starting at c and proceeding in a clockwise direction,
   - find first 1
4. Let $b = n_k$ and $c = n_{k-1}$
5. Repeat steps 3 & 4 until $b = b_0$ (all boundary pixel covered) & next boundary point found is b_1.

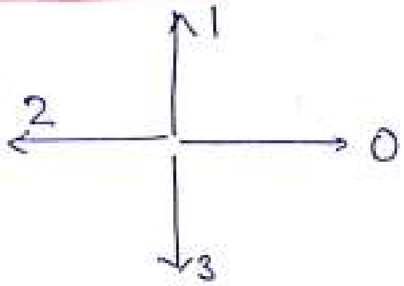
O/p is a sequence of <u>b</u>-points found (ordered boundary points)

2. Chain Codes :-

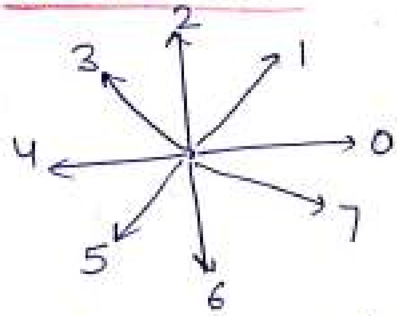
- Chain codes are used to represent a boundary by a connected sequence of straight-line segment of specified length and direction
- This is based on 4 or 8 connectivity of the segments.

- Direction of each segment is coded by using a numebring scheme

4-direction

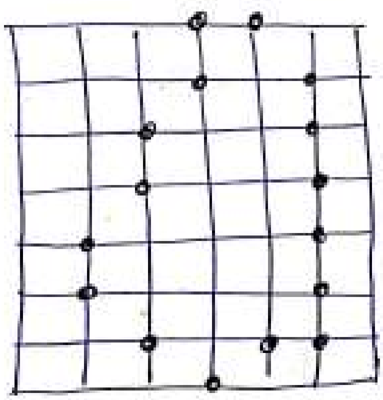


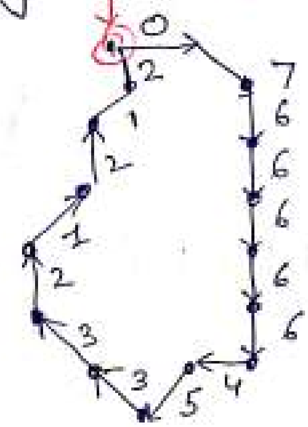
8-direction



- chain code generated by following a boundary in clockwise direction and assigning a direction to the segments connecting every pair of pixels.
- Topmost & leftmost point is choosen as a starting point of the boundary. starting point

eg.





So, chain code is: 0 7 6 6 6 6 6 4 5 3 3 2 1 2 1 2

- Chain code of a boundary depends on the Starting point (limitation) if starting point changed, chain code also changed.
- If object is reflected or rotated its chain code also changed.

- To solve this problem, normalization is performed on the object.

Steps of normalization

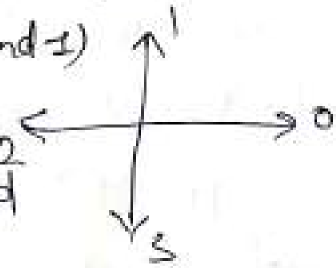
(1) Calculate chain code

(2) Calculate first difference of the chain code (difference is counting the no. of directions changes in a counterclock wise direction.)

(3) Calculate shape no (start the code from 0)

eg:

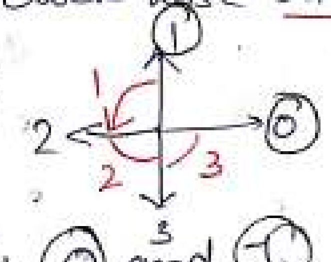
Let chain code is (using 4 direction)



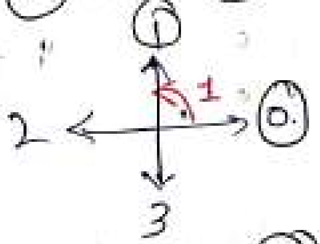
1 0 1 0 3 3 2 2 (diff of 2 and 1)

3 3 1 3 3 0 3 0

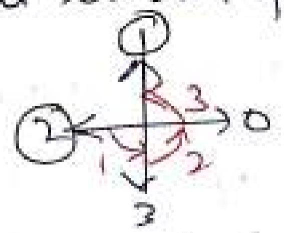
difference: first point is 1 & second is 0 so calculate counterclock wise direction



- Next point is 0 and 1



→ Now last number is 2 and first no is 1 find the difference and put it on first position of difference



Difference: - 3 3 1 3 3 0 3 0

Shape no: - 0 3 0 3 3 1 3 3

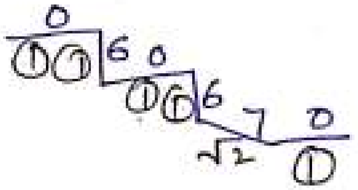
Boundary Descriptors

(1) Simple Descriptor

(i) **length:-** length of boundary is one of its simplest descriptors.

- The no. of pixels along a boundary gives a rough approximation of its length. For a chain coded curve with unit spacing in both directions, the no. of vertical and horizontal components plus $\sqrt{2}$ times the no of diagonal component.

eg:



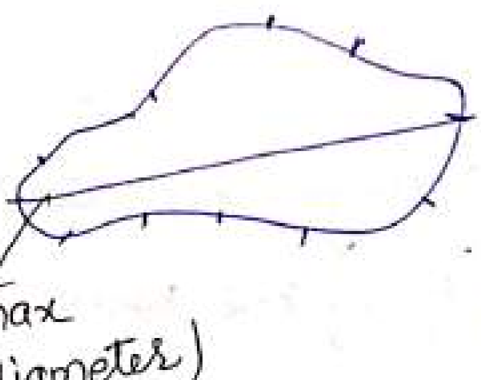
lenght $= 1+1+1+1+\sqrt{2}+1$
$= 5+\sqrt{2}$

(ii) **Diameter:-** The Diameter of a boundary B is defined as

$$\mathrm{Diam}(B) = \max_{i,j} [\underbrace{D}_{\text{Distance}}(p_i, p_j)]$$

p_i & p_j are points on the boundary.

e.g:

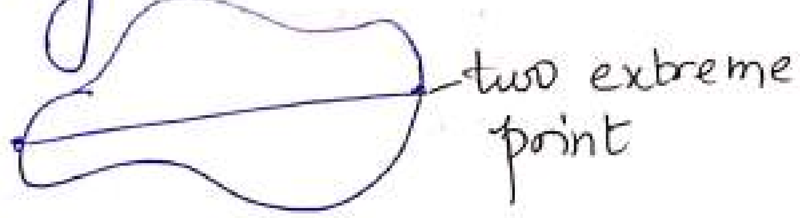


Points of boundary and calculate the distance b/w all pixels, max will be diameter

(iii) Major Axis :-

Line segment connecting the two extreme point that comprise the diameter is called major axis of the boundary.

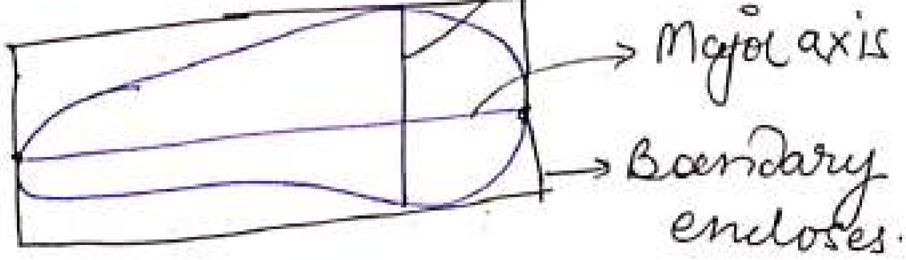
eg..



(iv) Minor Axis :-

- The minor axis of a boundary is defined as the line perpendicular to the major axis, and
- such length that a box passing through the outer four points of intersection of the boundary with the two axis completely encloses the boundary.

eg..



(v) Eccentricity :-

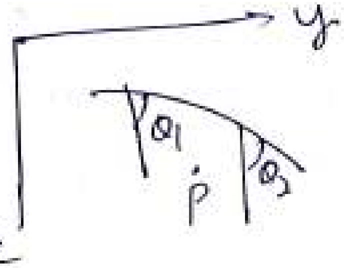
Ratio of the major to minor axis is called the eccentricity of the boundary.

(vi) Curvature :-

Curvature is Defined as the rate of change of slope.

As the boundary in the clockwise Direction, a vertex point P is said to be part of a convex segment, if the change in slope at P is nonnegative (+ve) otherwise, P is said to belong to a segment is concave.

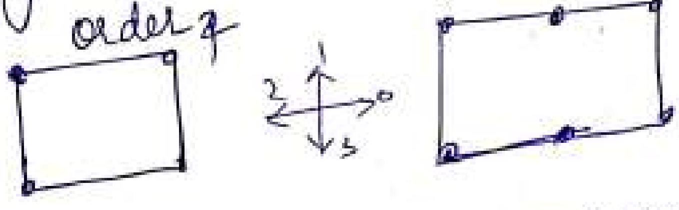
eg:



$\theta_2 - \theta_1 = +ve \rightarrow$ convex
else concave

(vii) Shape number :→

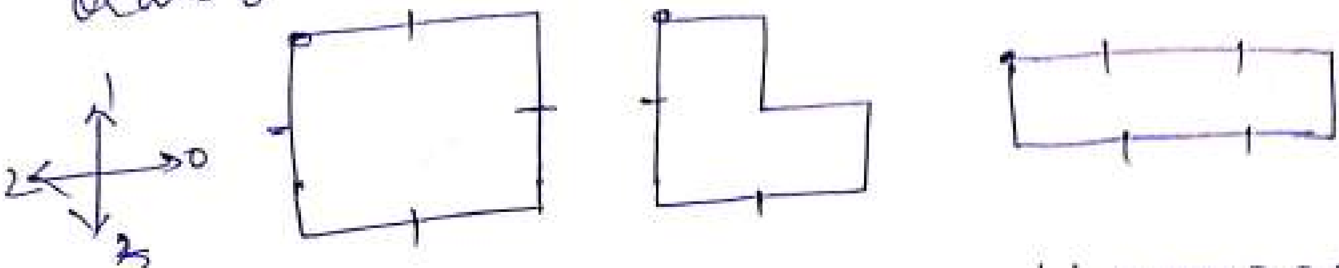
Shape no. of a boundary can be calculated with the help of difference in chain code. The order n of a shape no. is defined as the no. of digits in its representation.



order six

Chain code :	0321	003221
Difference :	3333	3033033
Shape no :	3333	033033

Order 8



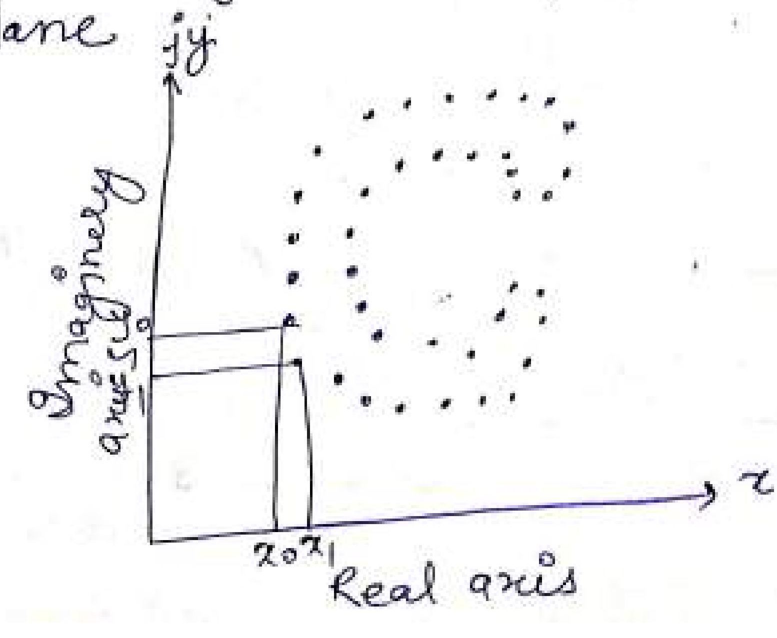
chain code :	00332211	03032211	00032221
Difference :	30303030	33133030	30033003
Shape no :	03030303	03033133	00330033

So, shape no. of a boundary is defined as the first difference (difference code) of smallest magnitude.

Order (n) of a shape no. is number of digits in the representation. n is even for a closed boundary.

Fourier Descriptors :-

- It's a boundary descriptor
- Convert the image boundary in frequency domain
- In below diagram K-point digital boundary in xy-plane



Point of Boundary.

$(x_0, y_0), (x_1, y_1) \ldots \ldots (x_{K-1}, y_{K-1})$ — K points.

$x(k) = x_k$, $y(k) = y_k$

Boundary $s(k) = [x(k), y(k)]$ $k = 0, 1, 2 \ldots K-1$

It can be written as

$$s(k) = x(k) + jy(k) \quad \text{——} \quad ①$$

$k = 0, 1, 2 \ldots K-1$

Convert $s(k)$ in fourier transform

$$a(u) = \sum_{k=0}^{K-1} s(k)\, e^{-j2\pi uk/K} \quad \text{—} ② \quad u = 0, 1, \ldots K-1$$

fourier descriptor of boundary

Apply IDFT to get $s(k)$

$$s(k) = \frac{1}{K} \sum_{u=0}^{K-1} a(u)\, e^{j2\pi uk/K} \quad \text{—} ③ \quad k = 0, 1, 2 \ldots K-1$$

- Now instant of taking all fourier coefficients only first P coefficients are used.

means $a(u) = 0$ for $u > P-1$ for eq ③

Approximation

$$\hat{s}(k) = \frac{1}{P} \sum_{u=0}^{P-1} a(u)\, e^{j2\pi \frac{ku}{P}} \quad \text{—} ④ \quad k = 0, 1, 2 \ldots K-1$$

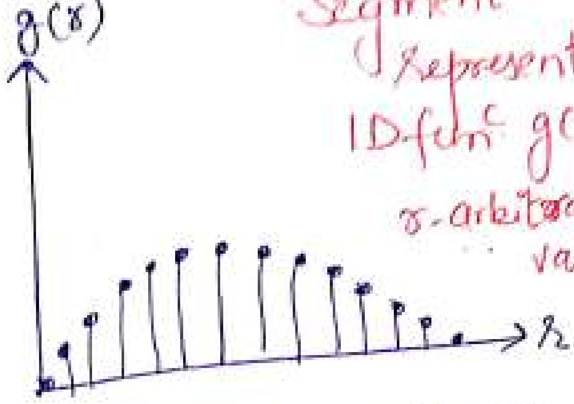
- High frequency components are eliminated.

9) Statistical Moments :-

- Boundary can be described by quantitatively measures like mean, variance, higher-order moments.

eg:

Segment of boundary



Segment represent as 1D fcn $g(r)$

r-arbitrary variable

- amplitude of g – discrete random variable v
- Generate amplitude histogram $p(v_i)$, $i = 1, 2 \ldots A-1$

  A-no. of discrete amplitude

- nth moment of v about its mean

$$\mu_n(v) = \sum_{i=0}^{A-1} (v_i - m)^n p(v_i) \quad \text{— (1)}$$

where

$$m = \sum_{i=0}^{A-1} v_i \, p(v_i) \quad \text{— (2)}$$

m is mean or average value of v and μ_2 is variance.