Hough transform (Global processing)

- one powerful global method for detecting edge is called the Hough transform.

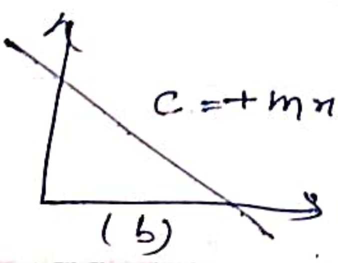
- Let us suppose that we are looking for straight line in an image

- If we take a point (x', y') in the image all the lines which pass through that pixel have the form

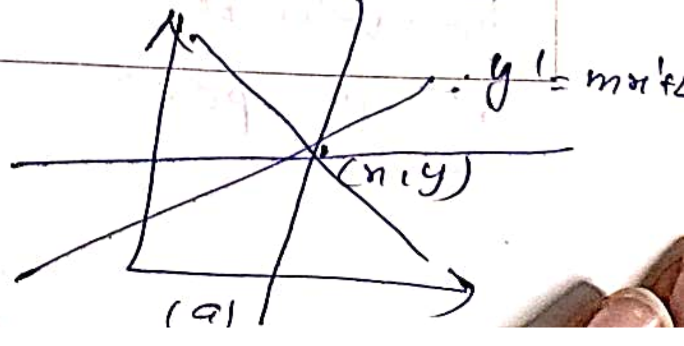
$$y' = mx' + c \quad \text{for varying values of } m \text{ and } c.$$

$$c = -x'm + y'$$

where now we consider x' and y' to be constants and m and c are varying.



(b)

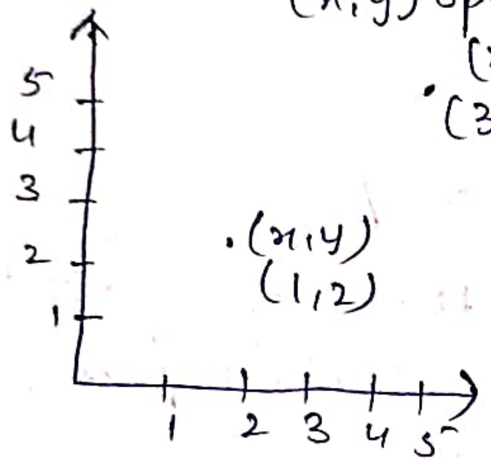


(a)

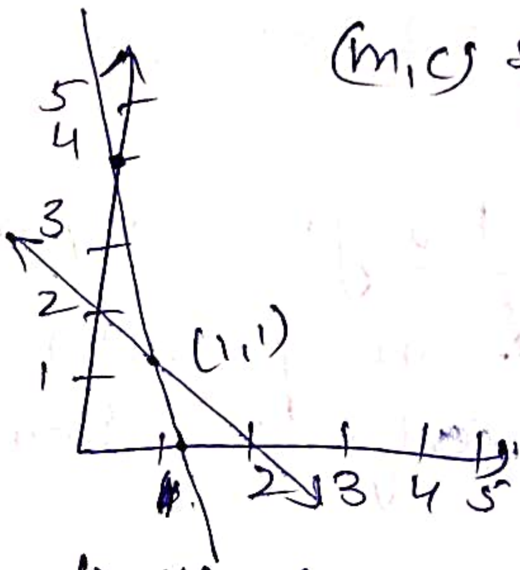
in fig (b), each diff. line through the (x', y') corresponds to one of the points on the line in (m, c) space.

Ex

(x, y) space



(m, c) space



$(x, y) = (1, 2)$

$y = mx + c$

$c = -mx + y$

$c = -m(1) + 2$

$c = -m + 2$

if $c = 0$ $m = 2$

$m = 0$ $c = 2$

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

$(x, y) = (3, 4)$

$y = mx + c$

$c = -mx + y$

$c = -m3 + 4$

$c = -3m + 4$

if $c = 0$ $m = \frac{4}{3} = 1.33$

$m = 0$ $c = 4$

$(m, c) = (1.33, 4)$

If point (A and B are two points connected by a line in spatial domain / (x, y) space they will be intersecting lines in the Hough space.

Edge Linking - An edge detection algorithm (Roberts, sobel, Prewitt, Log etc.) enhance the edges. When implemented, there are normally breaks in lines. Due to this reason, these are generally followed by linking procedures to assemble edge pixels into meaningful edges.

There are two basic approaches for edge linking

1) Local processing - This is a simplest approach for linking pixels in a small neighborhood.

2) Global processing via the Hough Transform
In this, we attempt to link edge pixels that lie on specified curves. The Hough transform is designed to detect lines using the parametric representation of a line.

Input image → Gradient operator → Edge Linking → A good edge output image.

Local processing

Analyze the characteristics of pixels in a small neighborhood S_{xy} (say 3×3, 5×5) about every edge pixel (x, y) in an image that have undergone edge detection.

All points that share some common properties are linked together. These are

a) strength / Magnitude of the gradient
b) Direction of Gradient

The Magnitude of gradient

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

$$= \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial y}{\partial y}\right)^2}$$

The direction of the gradient vector

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

Pixels (s,t) and (x,y) are similar and linked if

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

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

where E is a positive threshold
where A is a positive angle threshold.

Algo

1) compute ∇f, $M(x,y)$, $\alpha(x,y)$

2) Form a binary image $g(x,y) = \begin{cases} 1 & M(x,y) \geq T_M \text{ and } \alpha(x,y) = A \pm T_A \\ 0 & \text{otherwise.} \end{cases}$

3) Scan the row of g and fill all gaps in each row that do not exceed a specified length K

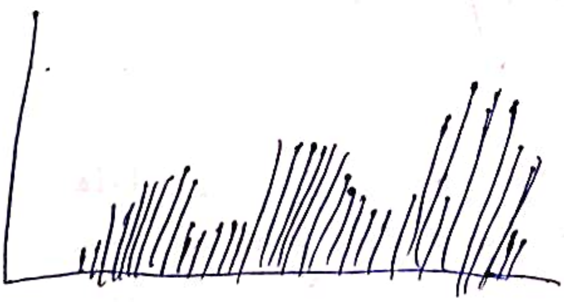
4) Detect the gap in any direction θ, rotate g by this angle and apply horizontal scanning procedure in step 3. Rotate the result by $-\theta$

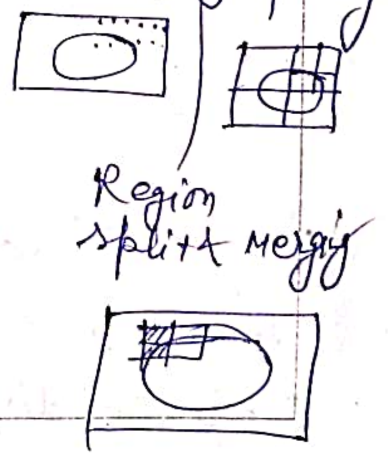
This local processing is expensive. A record has to be kept of all linked points by, eg - assigning a different label to every set of linked points.

Region Based Segmentation

Image Segmentation

- Discontinuity
  - Edge based (Point, Line & Edge)
    - FOD
      - Gradient based
        - Sobel
        - Robert
        - Prewitt
    - SOD
      - Gaussian based
        - Canny edge detection
        - Laplacian of Gaussian
- Similarity
  - Cluster based
  - Thresholding
    - Global
    - Local
  - Region based
    - Region Growing
    - Region splitting & merging
      - Region Merging
      - Region splitting
      - Region split & merging





It is difficult to segment this by thresholding methods.

Region based segmentation - Basic formulation
- Let R represent the entire image region.
- segmentation is a process that partitions R into subregions $R_1, R_2 \ldots R_n$

such that

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

b) R_i is a connected region $i = 1, 2 \ldots n$

c) $R_i \cap R_j = \phi$ for all i & j, $i \neq j$

d) $P(R_i) = TRUE$ for $i = 1, 2 \ldots n$.

e) $P(R_i \cup R_j) = FALSE$ for any adjacent regions R_i & R_j

where $P(R_K)$: a logical predicate defined over the points in set R_K.

eg- $P(R_K) = TRUE$ if all pixels in R_K have the same gray level.

<u>Region Growing</u> - Edge & thresholds sometimes do not give good results for segmentation. Thresholding still produces isolated image.

- Region growing algorithm works on principle of similarity.

- It states that a region is coherent if all the pixels of that region are homogenous w.r.t some characteristics such as color intensity, texture or other statistical properties.

- Thus idea is to pick a pixel inside a region of interest as a starting point (also known as a seed point) and allowing it to grow.

- Seed point is compared with its neighbours and if the properties match they are merged together.

- This process is repeated till the regions converge to an extent that no further merging is possible.

Region-Growing Algorithm

- It is a process of grouping the pixels or subregions to get a bigger region present in an image.

*** selection of the initial seed -**

- Initial seed given by the user
- can be chosen automatically
- seeds can be either single or multiple

* **seed growing criteria** - Illarity criterion denotes the minimum diff. in grey levels or the average of the set of pixels.

- Thus initial seed grows by adding the neighbours if they share the same properties as the initial seed.

* **Terminate process**: If further growing is not possible then terminate region growing process.

Ex

Consider image shown in figure

1^C	0^C	7^D	8^D	7^D
0^C	1^C	8^D	9^P	8^D
0^C	0^C	7^D	9^D	8^P
0^C	1^C	8^P	8^D	9^D
1^C	2^C	8^D	8^D	9^D

Black   White

① Assume seed point indicated by underlines.
let the seed pixels 1 and 9 represents
the regions C and D respectively.

② - Subtract pixel from seed value

Abs| seed value - pixel value| ≤ Threshold

③ If the difference is less than or equal to 4
i.e (T=4) merge the pixel with that
region.

otherwise merge the pixel with
the other region.

Region splitting

Entire image is assumed as a single region. Then the homogeneity (similarity) test is applied, where pixels that are similar are grouped together. If the conditions are not met, then the regions are split into 4 quadrants, else ~~leve~~ leave the region as it is.

- split & continue the subdivision process until some stopping criteria is fulfilled. The stopping criteria often occur at a stage where no further splitting is possible.

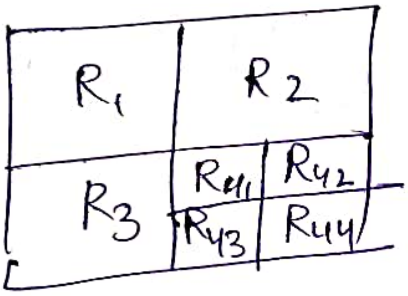
- This process is repeated for each quadrant until all the regions meet the required homogeneity criteria. If the regions are too small then division process is stopped.

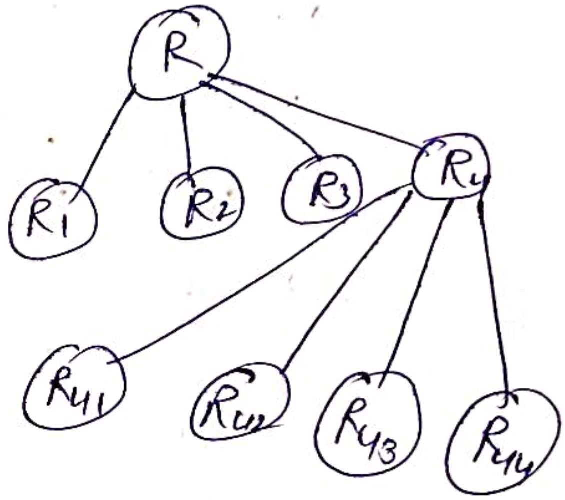
Region splitting

- To explain this in the terms of graph theory, we call each region a node.

- This example or tech. has a convenient representation in the form of a quadtree structure.

quadtree - A tree in which nodes have exactly 4 descendants





Region Merging

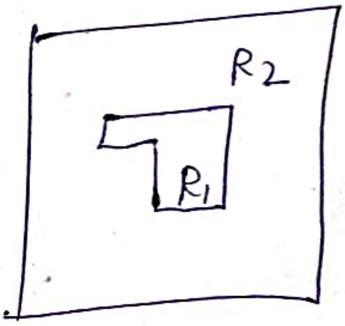
- It is opposite to region splitting
- we start from pixel level & consider each of them as a homogenous region.

- At any level of merging, we check if the 4 adjacent regions satisfy the homogeneity property. if yes, they are merged to form a bigger region,

otherwise the regions are left as they are.
- This is repeated until no further region exists that requires merging

0	0	0	0
0	1	1	0
0	0	1	0
0	0	0	0

Merge →



Region splitting & Merging.

splitting or merging might not produce good results when applied separately.
Better results can be obtained by interleaving merge & split operations.

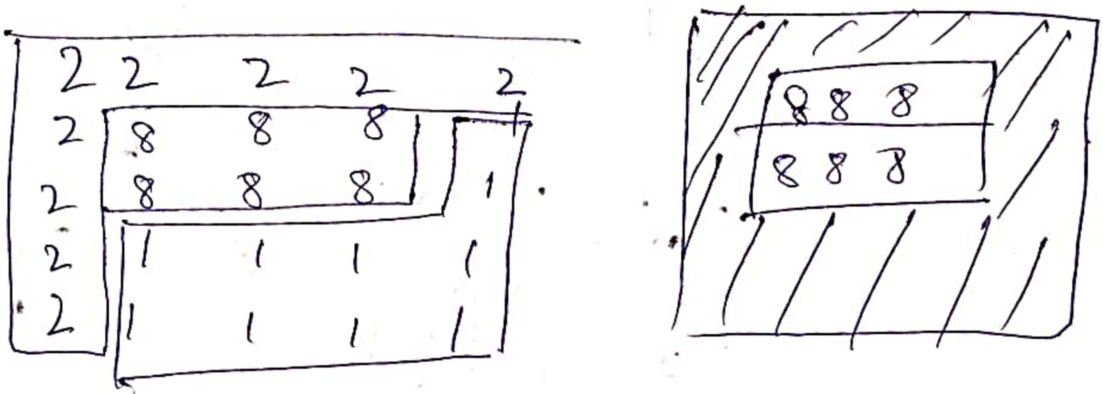
- The split & merge ~~produ~~ procedure is as follows.
- First there is a large region (possible the entire image)
- ~~first~~ Split into 4 disjoint quadrants any region R_i for which $P(R_i) = FALSE$

b) Merge any adjacent regions R_j and R_k for which $P(R_j \cup R_k) = TRUE$. (The quadtree structure may not be preserved).

c) Stop when no further merging or splitting is possible.

Ex

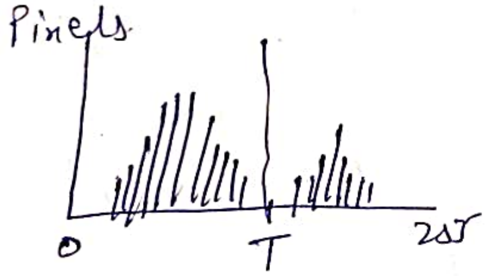
$T = 3$



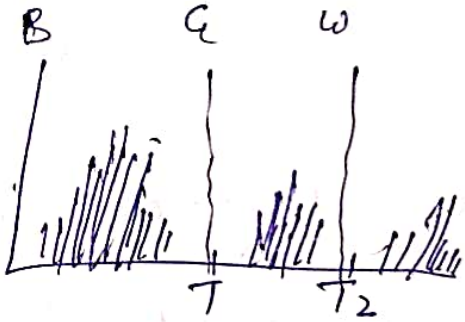
Thresholding – Global
– Local or adaptive

It is the simplest segmentation method. It is used to produce regions of similarity within the given image, based on some threshold criteria. Hence it partitions/segments an image into diff. objects.

The pixels are partitioned depending on their intensity value.



single Threshold



multiple threshold.

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

0 – 255
T = 125

$\boxed{\begin{array}{l} 0 - 125 \rightarrow 0 \\ 125 - 255 \rightarrow 1 \end{array}}$ Black
White.

Multiple Thresholding.

$$g(x,y) = a \quad \text{if } f(x,y) > T2$$
$$= b \quad \text{if } T1 \leq f(x,y) \leq T2$$
$$= c \quad \text{if } f(x,y) \leq T1$$

Thresholding types:-

Thresholding operation can be thought of as an operation, such that

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

where $f(x,y) \rightarrow$ gray level of input pixel at (x,y) and ~~P~~

$P(x,y) \rightarrow$ denotes some local property of this point (x,y)

eg - avg. level of a neighbor centered on (x,y)

Global Thresholding $T = T[f(x,y)]$

If the thresholding operation depends only on the gray scale value, it is called global thresholding.

Local Thresholding- If the neighborhood property is also taken into account, it is called local thresholding.

dynamic / adaptive thresholding

If T depends on pixel coordinates also, T is called dynamic / adaptive thresholding. Thresholding is called adaptive when a diff. threshold is used for different regions in the image.

$$T = T[P(x,y)\ f(x,y)]$$

Procedure or steps

1) select an initial estimate for T
2) segment the image using T. This will produce two group of pixels

$G_1 \rightarrow$ consisting of all pixels with gray level values $> T$

$G_2 \rightarrow$ consisting of pixels with gray level values $\leq T$

3) compute the average gray level values μ_1 and μ_2 for the pixels in regions G_1 and G_2.
4) compute a new threshold value
5) $T = \mu_1 + \mu_2 / 2$

6) Repeat steps 2 through 4 until the difference b/w the values of T in successive iterations is smaller than a predefined parameter ΔT.

Example

$$\begin{bmatrix} 5 & 3 & 9 \\ 2 & 1 & 7 \\ 8 & 4 & 2 \end{bmatrix}$$

Let $T = \frac{5+3+9+2+1+7+8+4+2}{9}$

$= \frac{41}{9} = 4.55 \approx 5$

segmenting the image using T, we would get

$G_1 = \{9, 7, 8\}$

$\mu_1 = \frac{9+7+8}{3} = \frac{24}{3} = 8$

$G_2 = \{5, 3, 2, 1, 4, 2\}$

$\mu_2 = \frac{5+3+2+1+4+2}{6}$

$= \frac{17}{6} = 2.83 \approx 3$

$T = \frac{1}{2}(8+3)$

$= \frac{1}{2}(11) = 5.5 \approx 5$ (stop the process)

If not equal you need to repeat the step till it is equal.

Image representation & description

Boundary representation & description

While image representation involves converting an image into a suitable format or representation that can be used for further analysis or processing.

Image description refers to extracting meaningful information or features from the image for various tasks like object recognition, classification or retrieval.

Boundary descriptors such as boundary length, diameter, curvature etc

- simple (curve length, curvature)
- shape (first diff. of smallest mag.)
- fourier (DFT)

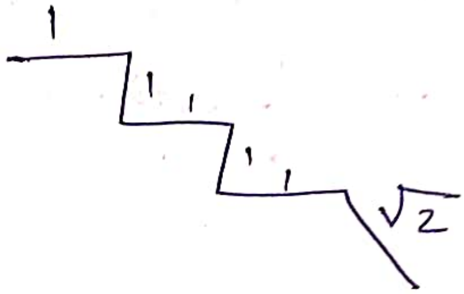
Regional Descriptors such as area, perimeter, compactness etc.

1) simple boundary descriptors

a) Length (Perimeter)

No. of pixels along a boundary gives its length.

- for a chain coded curve with unit spacing in both directions
- The No. of verticle and horizontal components plus $\sqrt{2}$ times the No. of diagonal components gives its exact length

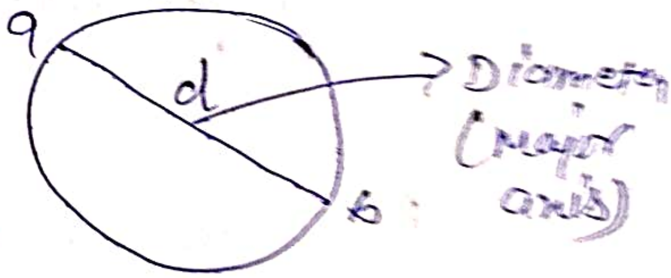


$1+1+1+1+1+\sqrt{2}$

b) Diameter $= \max_{i,j} [D(P_i + P_j)]$ = Major axis

D is the distance Measure (euclidian, cityblock or chessboard)

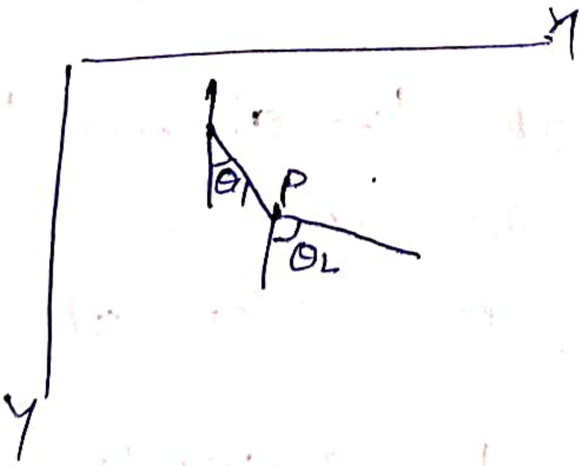
D = Diameter (Major axis)



6) Curvature - It is defined as the rate of change of slope.

Curvature at the point of intersection is defined as the difference between slopes of adjacent boundary segments.

To calculate curvature, the boundary is transversed in the clockwise direction and a vertex point P belongs to a convex segment, if the change in slope at P is non-negative (i.e +ve) otherwise P is said to belong to concave segment.



$d = \theta_2 - \theta_1$

$= +ve$ $\quad \theta_2 > \theta_1$

$\therefore$ Convex segment

or $\theta_2 < \theta_1$

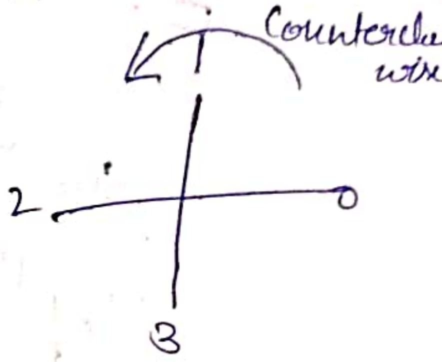
$= -ve$

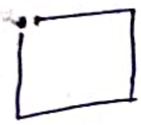
Concave Segment

2) Shape Number

The shape No. is defined as the first difference (difference code) of smallest magnitude.
↓
Chain

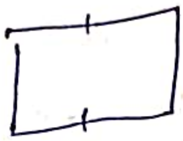
order (n) of a Shape No. is the No. of digits in the representation, n is even for a closed boundary.





Chain code : 0 3 2 1
diff. : 3 3 3 3
LFNo.
Shape No. : 3 3 3 3

order = 4



0 0 3 2 2 1
3 0 3 3 0 3
0 3 3 0 3 3

order = 6

3) Fourier descriptors

- Represent the boundary as a seq. of coordinates
- They convert the objects boundary into a set of Nos in the frequency domain, capturing its essential shape features.
- Treat each co-ordinate pair as a complex number.

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

$$S(k) = x(k) + i y(k)$$

- from the DFT of the complex No. we get the ~~fourier~~ fourier descriptors (the complex coefficients, $a(u)$)

$$\boxed{a(u) = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K}}$$

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

Now IDFT

$$\boxed{S(k) = \frac{1}{K} \sum_{u=0}^{K-1} a(u) e^{j2\pi uk/K} \qquad k = 0, 1, 2 \ldots K-1}$$