14/3/18

MODULE - 5
IMAGE SEGMENTATION

1) Types of Segmentation
2) Detection of point, line, edge
v.v.imp 3) Thresholding.

Types of Segmentation

Dividing an image into different block based on predefined criteria is called segment. After segmentation, blocks of similar properties are obtained. Those segments are known as regions denoted by R.

Conditions/Properties to be satisfied by a region

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


U → union

Consider a big image. Divide it into 5 segments R_1, R_2, R_3, R_4, R_5.
The pixels of region R_1 has similar properties. So the union of 5 regions is the full image.

2) R_i should be a connected set

Connectivity:- If there exist a set of pixel with a cell, ie, R_1 should be a connected set, R_2 should be a connected set, etc... Individually all region must be connected set.

3) $R_i \cap R_j = \phi$

Ri intersection Rj is a null set. Two different regions should not contain same properties or pixels.

4) $Q(R_i) = \text{TRUE}$

Q(Ri) is that, all the pixel with a region should contain similar pixels.

5) $Q(R_i \cup R_j) = \text{FALSE}$

It doesn't satisfy similarity property.

Two types of Segmentation

1) Edge based segmentation
2) Region based segmentation.

Edge based segmentation

Eg Edge → Transition from smoothness (continuous) to sudden discontinuity.

Segmenting an image based on edge. It is done by gradient and laplacian operator.

Region based segmentation

Segmenting based on regions. ie, similar property region is segmented.

Eg:- red pixels from image.

Scanned with CamScanner

Techniques

1) Region growing (clubing various regions)
2) Region splitting. (removing various pixels)

Point, Line and Edge Detection

Point detection

Laplacian filter is used for point detection. ie, centre pixel must have large value than other pixel and full pixel must sum to zero.

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

$$g(x,y) = 1 \quad \overset{i/p}{f(x,y)} > T$$
$$= 0 \quad \text{otherwise}$$

T is threshold value

line detection

Standard size of filter is 3x3.

If it is a horizontal line, the filter made use must have high value at middle row.

vertical line → filter will have high value at middle column

+45 diagonal → filter will have high value at 45° diagonal.

-45 diagonal → filter will have hight value at -45° diagonal.

Scanned with CamScanner

Filter for horizontal line dection

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

vertical line detection

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

+45° diagonal line

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

-45° diagonal line

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

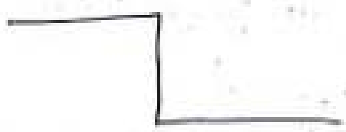
Edge Detection

Gradient operators are used

Different edges that appear in an image

1) Step edge

white	black



It has a sharp change high to low transition

Scanned with CamScanner

2) Ramp edge

white	→	→	black

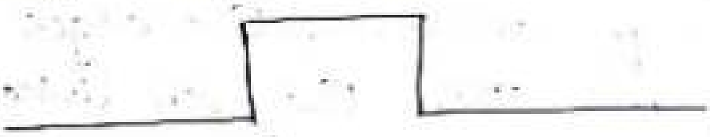
transition from white to black

3) Roof edge

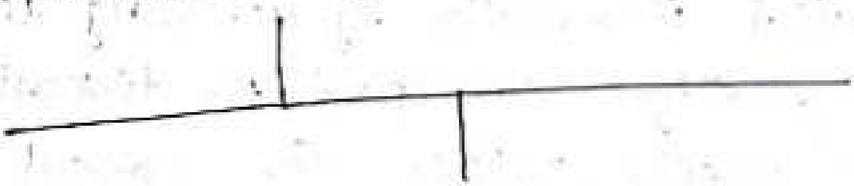
black	black to white transition	white to black transition	black

Consider the case of ramp edge,

According to characteristics of 1st order derivative (along ramp, non zero values)



Characteristics of 2nd order derivative



along the ramp → 0 value
onset and end → non zero
low to high high to low

Scanned with CamScanner

NOTE

Image Segmentation refers to the process of partitioning an image into group of pixels which homogenous with respect to some criterion. Different groups must not intersect with each other and adjacent groups must be heterogenous. Segmentation algorithms are area oriented instead of pixel oriented. The result of segmentation is splitting up of image into connected areas. Thus segmentation is concerned with dividing an image into meaningful region.

Segmentation can be broadly classified in 2 types:

* Edge Based Segmentation
* Region based segmentation

Edge based segmentation

In the edge approach, the edges are identified first, and then they are linked together to form required boundaries. Edge based segmentation exploits spatial information by detecting the edges in an image. Edges correspond to discontinuities in the homogeneity criterion for segment. Edge detection is usually done with local linear gradient operators such as Prewitt, Sobel and Laplacian filters

Scanned with CamScanner

These operations work well for images with sharp edges and low amount of noise.

Region Based Segmentation

Regions in an image are a group of connected pixels with similar properties. In region approach each pixel is assigned to a particular object or region. Based on the properties of region we segment out the region.

* Region growing

(look after edge detection)

It is a procedure that groups pixels or subregions into larger regions based on predefined criteria for growth. It requires a seed to begin with. Ideally the seed would be a region, but it could be a single pixel. A new segment is grown from the seed by assimilating as many neighbour pixels as possible that meet the homogeneity criterion. The resultant segment is then removed from the process. A new seed is choosen from remaining pixel. This continues until all pixels have been allocated to a segment.

* Region splitting

It is a top-down approach. It begins with a whole image and divides it up such that the segregated parts are more homogenous than the whole.

Scanned with CamScanner

Properties of region

Let 'R' represent the entire image region. We may view segmentation as a process that partitions R into 'n' subregions, $R_1, R_2 \dots R_n$ such that

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

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

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

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

e) $P(R_i \cup R_j) = FALSE$ for $i \neq j$

Here $P(R_i)$ is a logical prediction defined over the points in set R_i and ϕ is the null set.

- Condition (a) indicates that the segmentation must be complete, ie, every pixel must be in a region.
- Condition (b) requires that points in a region must be connected in some predefined sense.
- Condition (c) indicate that the region must be disjoint
- Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region - for Eg:- $P(R_i) = TRUE$ if all pixels in R_i have the same gray level.
- Condition (e) indicates that Regions R_i and R_j are different in the sense of predicate P.

Scanned with CamScanner

The derivatives of a digital function are defined in terms of differences. There are various ways to define these differences. The characteristics of the 1st order derivatives (gradient operator) are as follows:-

1) Must be zero in areas of constant intensity.
2) Must be non zero at the onset of a step or a ramp.
3) Must be non zero along the ramps.

The 2nd order derivative (laplacian operator) has the following characteristics.

1) Must be zero in constant areas.
2) Must be non zero at the onset and end of a step or a ramp.
3) Must be zero along the ramps of constant slope.

A basic definition of 1st order derivative of a 1-D function f(x) is the difference,

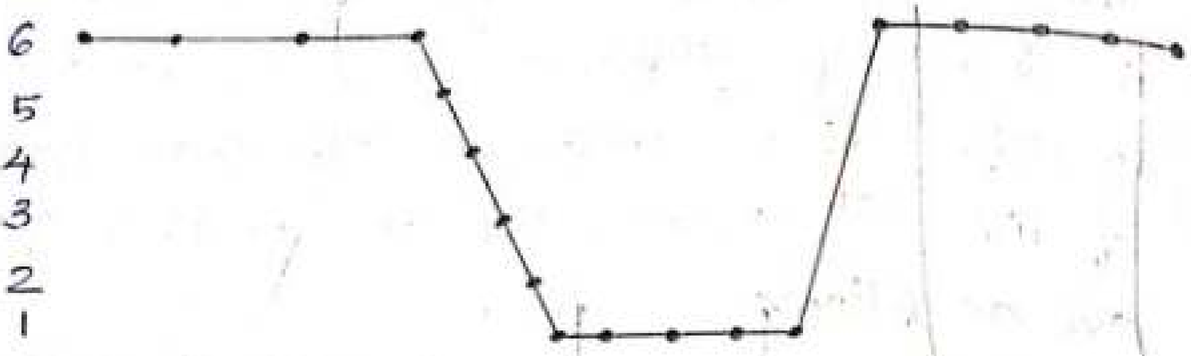
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

and 2nd order derivative has

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

To explain the characteristics of 1st order and 2nd order derivatives, consider a scan line as shown below.

Scanned with CamScanner



6 6 6 6 5 4 3 2 1 1 1 1 1 6 6 6 6 6 ...

1st derivative 0 0 0 -1 -1 -1 -1 -1 0 0 0 0 5 0 0 0 0

2nd derivative 0 0 -1 0 0 0 0 1 0 0 0 5 -5 0 0 0

2nd order derivatives ~~for~~ (Laplacian operator)

[illegible] for edge detection

The 2nd order derivative operator (Laplacian operator) for the image $f(x,y)$ of two variables is defined as

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

In x-direction,

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

In y-direction,

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

$$\therefore \nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$
$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Scanned with CamScanner

Because, the laplacian is a derivative operator, its use highlights intensity discontinuities in an image and de-emphasises regions with slowly varying intensity levels.

The basic way to use laplacian ~~for imag~~ is

$$g(x,y) = f(x,y) + c\,\nabla^2 f(x,y)$$

Examples of laplacian operator filters are:-

0	1	0
1	-4	1
0	1	0

or

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

1	1	1
1	-8	1
1	1	1

or

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

1st order derivative (Gradient operator)

write for edge detection

The 1st order derivatives in image processing are implemented using magnitude of the gradient. For a function $f(x,y)$, the gradient of f at coordinates (x,y) is defined as a two dimensional column vector.

$$\nabla f = \text{grad}(f) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$$

$$= \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$$

Scanned with CamScanner

The magnitude of the vector ∇f denoted as

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

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

Sometimes it may be approximated as

$$M(x,y) = |g_x| + |g_y|$$

Various 1^{st} order derivative filters are

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

-1	0
0	1

0	-1
1	0

The above filters are called Robert's Cross gradient filters.

The cross differences can be represented as

$g_x = x_9 - x_5$ and $g_y = x_8 - x_6$

$$M(x,y) = \sqrt{(x_9 - x_5)^2 + (x_8 - x_6)^2}$$

$$\approx |x_9 - x_5| + |x_8 - x_6|$$

Scanned with CamScanner

Another type of gradient operator is sobel operator

(Eg. for Prewitt operator)

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

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

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

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

Prewitt operator

$$g_x = \frac{\partial f}{\partial x} = z_7 + 2z_8 + z_9 - [z_1 + 2z_2 + z_3]$$

$$g_y = \frac{\partial f}{\partial y} = z_3 + 2z_6 + z_9 - [z_1 + 2z_4 + z_7]$$

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

$$= \sqrt{\left[z_7 + 2z_8 + z_9 - [z_1 + 2z_2 + z_3]\right]^2 + \left[z_3 + 2z_6 + z_9 - [z_1 + 2z_4 + z_7]\right]^2}$$

$$\approx \left| z_7 + 2z_8 + z_9 - [z_1 + 2z_2 + z_3] \right| + \left| z_3 + 2z_6 + z_9 - [z_1 + 2z_4 + z_7] \right|$$

Point, Line and Edge Detection

Point detection

The detection of isolated points in an image is straight forward in principle. Using the mask shown below, we say that the point has been detected at the location on which the mask is centered if $|R| \geq T$. where $T \rightarrow$ non negative threshold and

$R = w_1 z_1 + w_2 z_2 + \dots\dots + w_9 z_9 = \sum_{i=1} w_i z_i$

where z_i is the gray level of pixel associated with mask coefficient w_i.

Basically this formulation measures the weighted differences between the center point and its neighbours. The idea is that an isolated point (a point whose gray level is significantly different from its background and which is located in a homogeneous or nearly homogeneous area) will be quite different from its surroundings and thus be easily detectable by this type of mask.

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

The Emphasis here is strictly on the detection of points. The mask coefficients sum to zero, indicating that the mask response will be zero in areas of constant gray level.

Scanned with CamScanner

Line Detection

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

Horizontal

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

+45°

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

vertical

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

-45°

If the 1st mask were moved around an image, it would respond more strongly to lines (one pixel thick) oriented horizontally with a constant background; the maximum response would result when the lines passed through the middle row of the mask.

2nd mask in figure responds to line oriented at +45°, the third mask to vertical lines, fourth mask to lines in -45° direction. These directions can be established also by noting that the preferred direction of each mask is weighted with a larger coefficient (i.e, 2) than other possible directions. Coefficients in each mask sum to zero, indicating a zero response from the masks in areas of constant gray level.

Scanned with CamScanner

Let R_1, R_2, R_3 and R_4 denote the responses of the mask shown above where the R_i's are given by

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

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

Suppose that the four mask are run individually through an image. If at a certain point in the image $|R_i| > |R_j|$, for all $j \neq i$, that the point is said to be more likely associated with a line in the direction of mask i. For detecting lines in a specified direction, we use the mask associated with that direction and thresholding its output. For detecting all the lines in an image in the direction defined by a given mask, we simply run the mask through the image and threshold the absolute value of the result.

Edge Detection

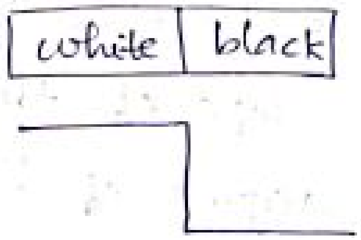
Edge detection is the most common approach for detecting meaningful discontinuities in gray level. An edge is a set of connected pixels that lie on

Scanned with CamScanner

the boundary between two regions. A reasonable definition of edge requires the ability to measure gray-level transitions in a meaningful way.

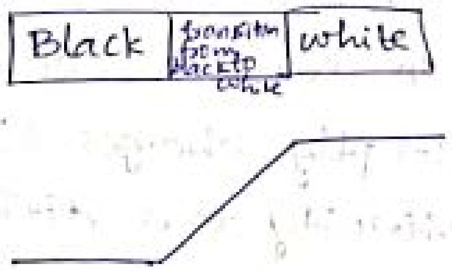
Different types of edge that appears in an image

1) step Edge



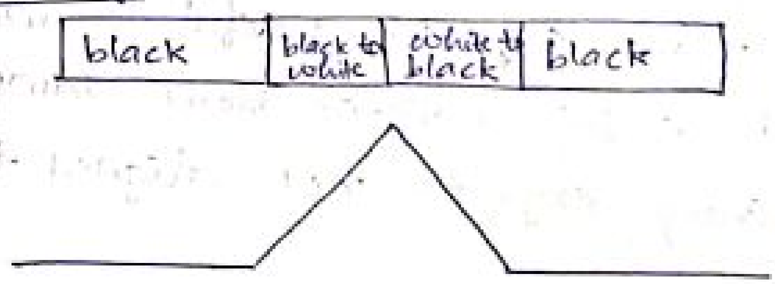
(a)

2) Ramp edge



(b)

3) Roof Edge



(c)

An ideal ~~eg~~ edge (fig a) is a set of connected pixels which is located at an orthogonal step transition in gray level.

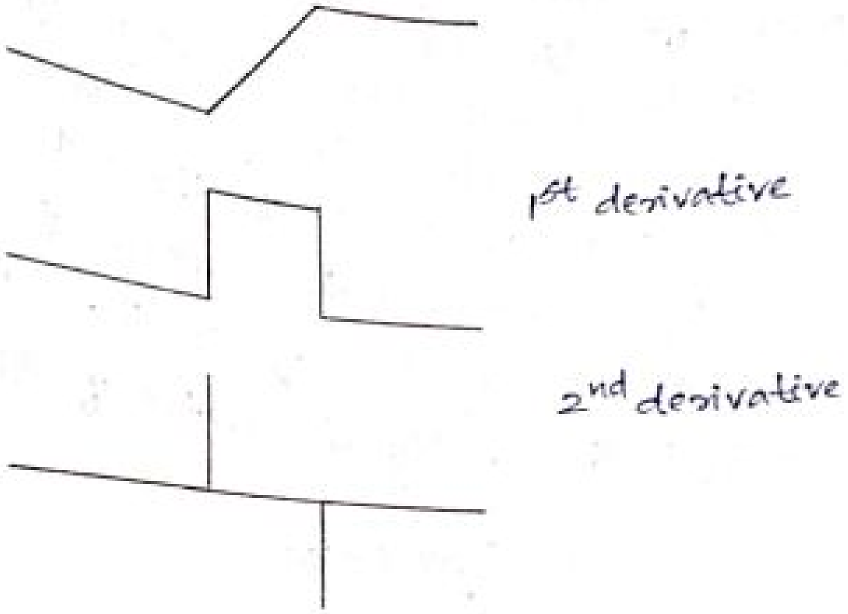
Scanned with CamScanner

optics, sampling and other image aquisition imperfections yield edges that are blurred, with the degree of blurring being determined by the factors such as quality of the image aquisition system, the sampling rate and illumination conditions under which an image is acquired. So the Edges are modeled as having a "ramplike" profile (fig b). The slope of the ramp is inversely proportional to the degree of blurring in the edge.

* Step Edge defines a perfect transition from one segment to another. In the case of step edge, the image intensity abruptly changes from one value to one side of discontinuity to a different value on the opposite side.
* A ramp allows for a smoother transition between segments. A ramp edge is useful for modelling the blurred edges created from sampling a scene containing objects not aligned to the pixel grid.
* Two nearby ramp edges result in a line structure called a roof.

Scanned with CamScanner

nsider the case of a ramp edge,



The first derivative is positive at the points of transition into and out of the ramp, as we move from left to right along the profile. It is constant for points in the ramp and is zero in areas of constant gray level.

The second derivative is positive at the transition associated with the dark side of the edge, negative at the transition associated with the light side of the edge and zero along the ramp and in areas of constant gray level.

The magnitude of the first derivative can be used to detect the presence of an edge at a point in an image. The sign of the 2nd derivative can be used to determine whether an edge pixel lies on the

Scanned with CamScanner

dark or light side of an edge. 2nd
Two additional properties of the 2nd in an im
around an edge are:-

* It produces two values for every edge in extreme
* An imaginary straight line joining the extreme positive and -ve values of the 2nd derivative would cross zero near the midpoint of the edge. This zero-crossing property of the 2nd derivative is useful for locating the centres of thick edges.

21/3/18 Region Based Segmentation

1) Region growing
2) Region splitting and merging.

Region Growing

Region growing is a procedure that grows pixel or subregions into larger regions based on predefined criteria for growth. The basic approach is to start with a set of seed points and from these grow regions by appending to each seed, those neighbouring pixel that have predefined properties similar to the seed. When a priory information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign

Scanned with CamScanner

pixels to regions during the growing process. Let $f(x,y)$ denote an i/p image array, $s(x,y)$ denote a seed array and Q denote a predicate to be applied at each location (x,y).

Algorithm

step1 :- Find all connected components in $s(x,y)$ and erode each connected component to one pixel. Label all such pixels found as one. All other pixels in s are labelled zero.

step2 :- Form an image f_Q such that at a pair of coordinates (x,y), let $f_Q(x,y) = 1$, if the i/p image satisfies the given predicate Q otherwise let $f_Q(x,y) = 0$.

step3 :- Let G be an image formed by appending to seed points in s, all the one valued points in f_Q that are eight connected to the seed point. This is a segmented image obtained by region growing.

The measure of similarity is based on the predicate Q applied at each location (x,y). i.e,

$$Q = \begin{cases} \text{TRUE}, & \text{if the absolute difference of the intensities between the seed and the pixel is less than or equal to } T \text{ (threshold)} \\ = \text{FALSE}, & \text{otherwise} \end{cases}$$

Scanned with CamScanner

Region Splitting and Merging.

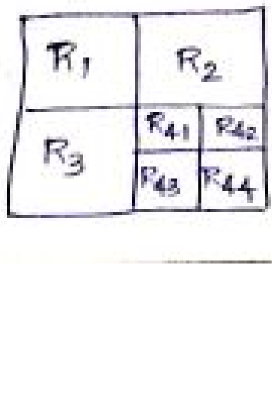
Let R represent the entire image regi[on] and select a predicate Q. One approach for segmenting R is to subdivide it successively smaller and smaller quadrant regions so that for any region R_i, $Q(R_i) = TRUE$. If $Q(R_i) = FAL[SE]$ we divide that region to quadrants, if $Q = FALSE$ for any quadrant, we subdivide th[e] quadrants into subquadrants and so on. This splitting techniques results in Quad-trees or Quad regions or Quad images. Two adjacent regions R_j and R_K are merged only if $Q(R_j \cup R_K) = TRUE$

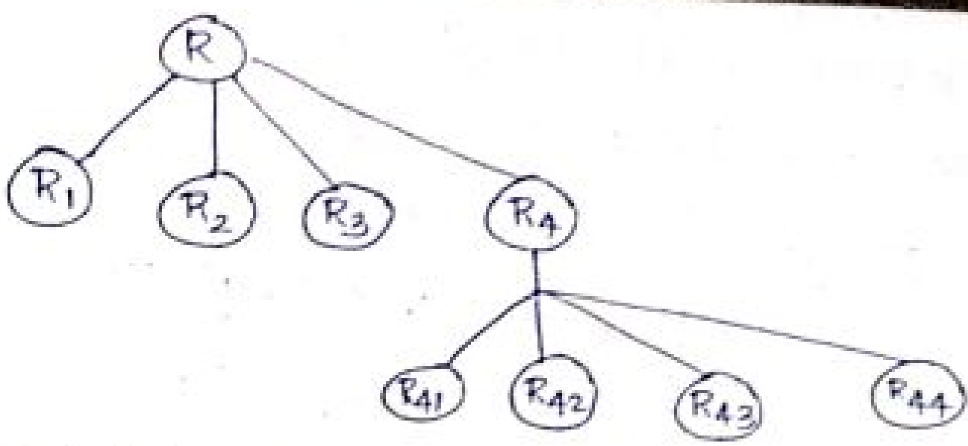
Algorithm

Step1:- Split into 4 disjoint quadrants. Any regin R_i for which $Q(R_i) = FALSE$.

Step2:- when no further splitting is possible, merge any adjacent regions R_j and R_K for which $Q(R_j \cup R_K) = TRUE$

Step3:- Stop when no forther merging is possible.





imp **Segmentation using thresholding.**

Suppose an image $f(x,y)$ composed of light objects on a dark background in such a way that the object and the background pixels have the intensity values grouped into 2 dominant modes. One way to extract objects from the background is to select a threshold T if $f(x,y) > T$, it is an object point else a background point.

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

When T is a constant, applicable over an entire image, then this process is known as global thresholding. If T at any point (x,y) in an image depends on properties of a neighbourhood of (x,y), then it is local thresholding or variable thresholding.

Multiple thresholding classifies a point (x,y) as belonging to the background if $f(x,y) \leq T_1$, to one object class if $T_1 < f(x,y) \leq T_2$ and to another

Scanned with CamScanner

object class if $f(x,y) > T_2$.

The segmented image is given by,

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

Basic Global Thresholding.

step:- 1) Select an initial estimate for the global threshold T.

2) Segment the image using T to produce two groups of pixels C_1 and C_2. C_1 consisting of all pixels with intensity values less than T and C_2 consisting of pixels with intensity values greater than or equal to T.

3) Compute the mean of the intensity values m_1 and m_2 for the pixels in C_1 and C_2 respectively.

4) Compute the new threshold value $T = \frac{1}{2}[m_1 + m_2]$

5) Repeat the steps until the difference between the values of T in successive iterations is very small.

Scanned with CamScanner

Optimum Thresholding Using Otsu's method

Otsu's method of classification maximises the between class variance and minimises the in class variance. Let L denotes intensity levels in a digital image of size MXN in such a way that $\sum_{i=0}^{L-1} P_i = 1$.

Select a threshold k to classify the image into two classes C_1 and C_2 where C_1 consists of all the pixels in the image with intensity values in the range 0 to k and C_2 consists of pixels with values in the range k+1 to L-1.

The probability $P_1(k)$ in class C_1 is given by $P_1(k) = \sum_{i=0}^{k} P_i$.

Similarly probability of class C_2 is given by

$$P_2(k) = \sum_{i=k+1}^{L-1} P_i$$

$$P_2(k) = 1 - P_1(k)$$

The mean intensity values of the pixel assigned to class C_1 is

$$m_1(k) = \sum_{i=0}^{k} i \cdot P(i/C_1) = \sum_{i=0}^{k} i \cdot \frac{P(C_1/i) \cdot P(i)}{P(C_1)}$$

$$= \frac{\sum_{i=0}^{k} i P_i}{P_1(k)} = \frac{1}{P_1(k)} \sum_{i=0}^{k} i P_i$$

Scanned with CamScanner

The mean intensity values of the pixel assigned to class C_2 is

$$m_2(k) = \sum_{i=k+1}^{L-1} i \cdot P(i/C_2)$$

$$= \sum_{i=k+1}^{L-1} i \frac{P(C_2/i)\, P(i)}{P(C_2)}$$

Bay's theorem
$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$

$$= \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i P_i$$

The global mean is given by,

$$m_G = \sum_{i=0}^{k} i P_i + \sum_{i=k+1}^{L-1} i P_i$$

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

$$= m_1 P_1 + m_2 P_2$$

Inorder to evaluate the goodness of the threshold level, we use a dimensionless matrix η,

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$ where σ_G^2, the global variance
σ_B^2, the b/w class variance

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

$$\sigma_B^2 = P_1 P_2 (m_1 - m_2)^2 = \frac{(m_G P_1 - m)^2}{P_1 (1 - P_1)}$$

Scanned with CamScanner

Re-introducing k, we can write $\eta(k)$,

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \quad \text{and}$$

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

To find k^*, we select a value of k that yield maximum $\sigma_B^2(k)$

$$\text{ie, } \sigma_B^2(k^*) = \max_{0 \le k \le L-1} \sigma_B^2(k)$$

Once k^* has been obtained, the i/p image $f(x,y)$ is segmented as

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > k^* \\ 0 & \text{if } f(x,y) < k^* \end{cases}$$

Otsu's algorithm may be summarised as:-

1) Compute the normalised histogram of the i/p image p_i.
2) Compute the cumulative sums $P_1(k)$ for $k = 0, 1, \ldots, L-1$
3) Compute the cumulative mean $m(k)$.
4) Compute the global intensity mean m_G.
5) Compute the between class variance $\sigma_B^2(k)$.
6) Obtain otsu's threshold k^*, the value of k for which $\sigma_B^2(k)$ is maximum.
7) Obtain the seperability measure η^*.

Scanned with CamScanner

Multiple thresholds.

If there are N classes, then we can make use of multiple thresholding techniques.

If $N=3$, then $P_1+P_2+P_3=1$ where,

$$P_1(k) = \sum_{i=0}^{k_1} P_i$$

$$P_2(k) = \sum_{i=k_1+1}^{k_2} P_i \quad \text{and} \quad P_3(k) = \sum_{i=k_2+1}^{L-1} P_i$$

The mean of intensities within each class is given by,

$$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^{k_1} i P_i$$

$$m_2(k) = \frac{1}{P_2(k)} \sum_{i=k_1+1}^{k_2} i P_i \quad \text{and} \quad m_3(k) = \frac{1}{P_3(k)} \sum_{i=k_2+1}^{L-1} i P_i$$

Also $m_1 P_1 + m_2 P_2 + m_3 P_3 = m_G$

$$\sigma_B^2 = P_1(m_1-m_G)^2 + P_2(m_2-m_G)^2 + P_3(m_3-m_G)^2$$

Similarly,

$$\sigma_B^2(k_1^*, k_2^*) = \max_{0 \le k_1, k_2 \le L_2} \sigma_B^2(k_1, k_2)$$

The segmented image can be represented as

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) < k_1^* \\ b & \text{if } k_1^* < f(x,y) < k_2^* \\ c & \text{if } f(x,y) > k_2^* \end{cases}$$

Scanned with CamScanner

Variable Thresholding.

In variable thresholding, main approach is to compute a threshold value at every point (x,y) in an image based on one or more specified properties computed in a neighbourhood of (x,y). Let σ_{xy} and m_{xy} denotes the standard deviation and mean value of a set of pixels contained in a neighbourhood centered at coordinates (x,y). The local threshold can be defined as

$T_{xy} = a\sigma_{xy} + bm_{xy}$ where a and b are non negative constants. The segmented image is computed as

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

In other words,

$$g(x,y) = 1 \quad \text{if } Q(\text{local parameters}) = \text{TRUE}$$

$$= 0 \quad \text{if } Q(\text{local parameters}) = \text{FALSE}$$

i.e, $Q(\sigma_{xy}, m_{xy}) = \text{TRUE}$ if $f(x,y) > a\sigma_{xy}$ and $f(x,y) > bm_{xy}$

$Q(\sigma_{xy}, m_{xy}) = \text{FALSE}$, otherwise

Scanned with CamScanner

Clustering Techniques.

The clustering technique attempts to access the relationships among patterns of the data by organising the patterns into groups or clusters such that patterns within a cluster are more similar to each other than patterns belonging to different clusters.

ie, clustering refers to the classification of objects into groups according to certain properties of these objects.

A standard procedure for clustering is to assign each pixel to the class of the nearest cluster mean.

1) Hierarchical clustering.

Hierarchical clustering techniques are based on the use of a proximity matrix indicating the similarity between every pair of data points to be clustered. The end result is a tree of clusters representing the nested group of patterns and similarity levels at which groupings change. The resulting clusters are always produced as the internal nodes of the tree, while the root node is reserved for the entire dataset and leaf nodes are for individual data samples.

Scanned with CamScanner

The two main categories of algorithms used in the hierarchical clustering framework are agglomerative and divisive.

* Agglomerative algorithms seek to merge clusters to be larger and larger by starting with N single point clusters. The algorithm can be divided into 3 classes :
1) single-link algorithm
2) complete-link algorithm
3) minimum variance algorithm.

→ The single-link algorithm merges two clusters according to the minimum distance between the data samples from two clusters. The algorithm allows for a tendancy to produce clusters with elongated shapes.

→ The complete-link algorithm incorporates the maximum distance between data samples in clusters, but its application always results in compact clusters. The quality of hierarchical clustering depends on how the dissimilarity measurement between two clusters is defined.

→ The minimum-variance algorithm combines two clusters in the sense of minimising the cost function, namely to form a new cluster with the minimum increase of the cost function. This algorithm has attracted

Scanned with CamScanner

considerable interest in vector quantisation, when it is termed pairwise-nearest-neighbourhood algorithm.

* Divisive clustering begins with the entire dataset in the same cluster followed by iterative splitting of the dataset until the single-point clusters are attained on leaf nodes. It follows a reverse clustering strategy against agglomerative clustering. On each node, the divisive algorithm conducts a full search for ~~for~~ all possible pairs of clusters for data samples on the node.

Some of the hierarchical algorithms include COBWEB, CURE and CHAMELEON.

2) Partitional clustering

Partition-based clustering uses an iterative optimisation procedure that aims at minimising an objective function f, which measures the goodness of clustering. Partition based clustering are composed of two learning steps - the partitioning of each pattern to its closest cluster and the computation of the cluster centroids.

A common feature of the partition-based clusterings is that clustering procedure starts

Scanned with CamScanner

from an initial solution with a known number of clusters. The cluster centroids are usually computed based on the optimality criterion such that the objective function is minimised.

Partitioning algorithms are categorised into partitioning relocation algorithms and density-based partitioning. Algorithms of the first type are further categorised into probabilistic clustering k-medoids and k-means. Density-based partitioning include algorithms such as DBSCAN, OPTICS DBCLAS, DENCLUE.

3) K-means clustering.

The k-means method is the simplest method in unsupervised classification. The clustering algorithms do not require training data. K-means clustering is an iterative procedure. the k-means clustering algorithm clusters data by iteratively computing a mean intensity for each class and segmenting the image by classifying each pixel in the class with the closest mean.

The steps involved in k-means clustering algorithm

1) choose k initial ~~the~~ clusters $z_1(1), z_2(1), \ldots z_k(1)$

2) At the k^{th} iterative step, distibute the samples

Scanned with CamScanner

x among the k clusters using the relation

$$x \in C_j(k) \text{ if } \|x - z_j(k)\| < \|x - z_i(k)\|$$

for $i = 1, 2, \dots k$, $i \neq j$ where $C_j(k)$ denotes the set of samples whose cluster centre is $z_j(k)$.

3) Compute the new cluster centres $z_j(k+1)$, $j = 1, 2, \dots k$, such that the sum of the squared distance from all points in $C_j(k)$ to the new cluster ~~TR~~ is minimised. The measure which minimises this is simply the sample mean of $C_j(k)$. Therefore, the new cluster centre is given by

$$z_j(k+1) = \frac{1}{N_j} \sum_{x \in C_j(k)} x \quad , j = 1, 2, \dots k$$

wher N is the number of samples in $C_j(k)$.

4) If $z_j(k+1)$, $j = 1, 2, \dots k$, the algorithm has converged and the procedure is terminated. Otherwise go to step 2.

The drawback of the k-means algorithm is that the number of clusters is fixed, once k is chosen and it always returns k cluster centres.

26/3/18

Canny Edge Detector

Canny's Edge detector is based on 3 basic objectives.

1) Low error rate :-

All edges should be found and those edges detected must be as close as possible to the true edges.

2) Edge points should be well localised:-

The distance between a point marked as an edge by the detector and the centre of the true edge should be minimum.

3) Single edge point response :-

The detector should return only one point for each true edge point.

The optimal step edge detector is the first derivative of the Gaussian filter. Let $f(x,y)$ denote the i/p image and $G(x,y)$ denote the gaussian function.

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

We form a smoothed image $f_s(x,y)$.

$$f_s(x,y) = f(x,y) * G(x,y)$$

Scanned with CamScanner

This operation is followed by computing the gradient, magnitude and direction

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

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

where $g_x = \frac{\partial f_s}{\partial x}$ and $g_y = \frac{\partial f_s}{\partial y}$

Since it is generated using gradient, $M(x,y)$ typically contains white ridges. (thick ridges)

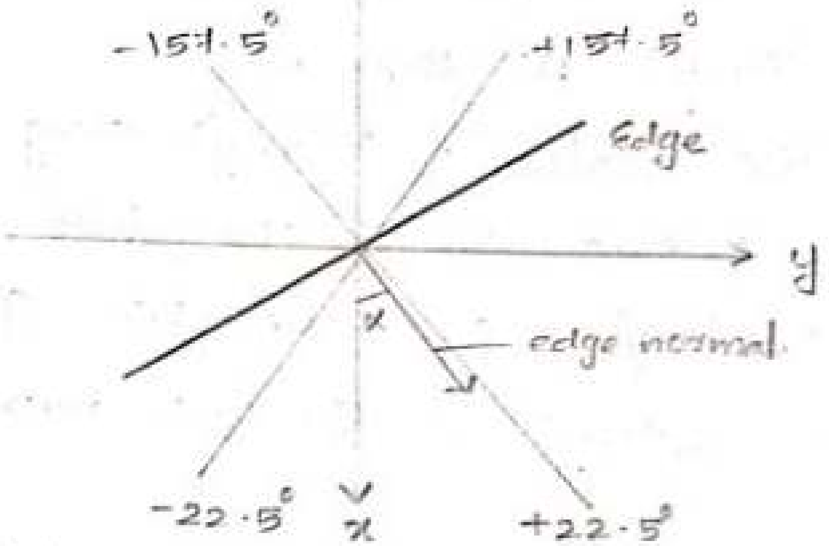
The next step is to thin those ridges. The process is known as Non maxima suppression.

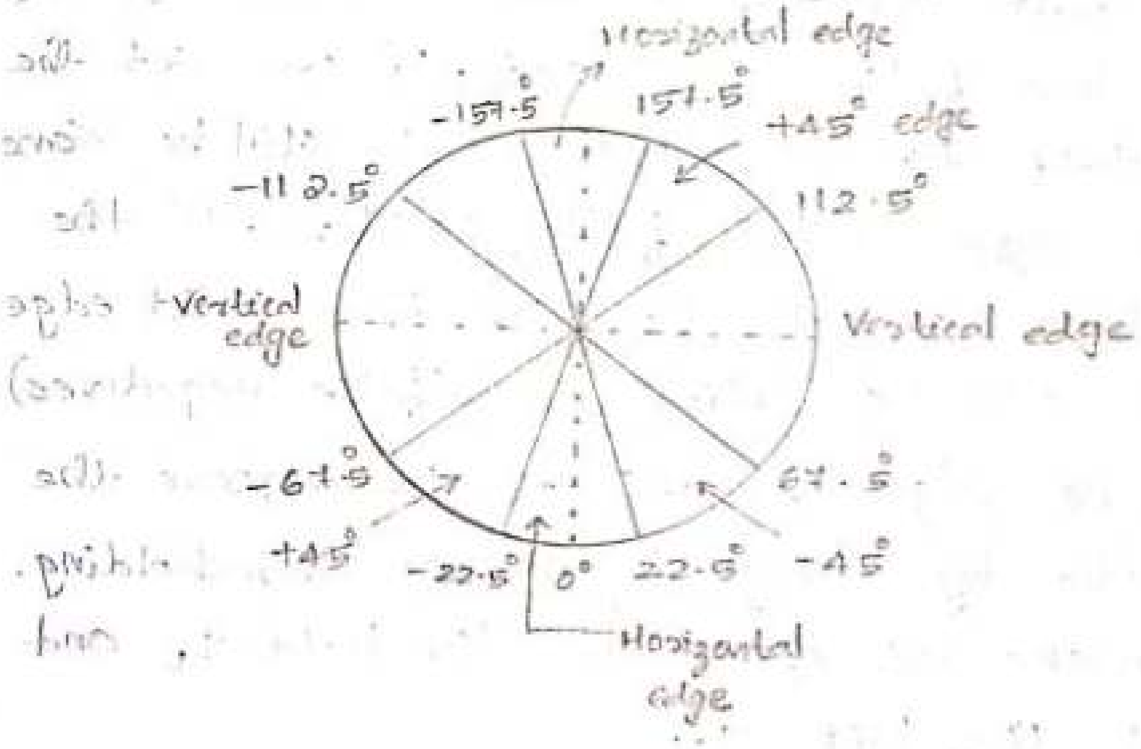
Consider a 3×3 region in which we can define 4 orientations for an edge passing through the centre point of the region.

1) Horizontal
2) Vertical
3) $+45°$
4) $-45°$

We determine the edge direction from the direction of the edge normal, which we obtain directly from the image data.

Scanned with CamScanner





Let d_1, d_2, d_3 and d_4 denotes the 4 basic edge directions for a 3x3 region: Horizontal, $-45°$, vertical and $+45°$ respectively. The non maxima suppression scheme for a 3x3 region centred at every point (x,y) in $\alpha(x,y)$ can be formulated as:-

Scanned with CamScanner

1) Find the suppression d_K that is closest to $\alpha(x,y)$. ~~If the va~~

2) If the value of $M(x,y)$ along d_K is less than atleast one of its two neighbours, let $g_N(x,y) = 0$ otherwise $g_N(x,y) = M(x,y)$ where $g_N(x,y)$ is the non maxima suppressed image.

The final operation is to threshold $g_N(x,y)$ to reduce false edge points. If we set the threshold too low there will still be some false edges called false positives. If the threshold is set too high, then valid edge points will be eliminated (false negatives). Canny's algorithm attempts to improve the situation by using hysteresis thresholding. i.e, make use of a low threshold T_L and high threshold T_H.

The thresholding operations can be visualised as

$$g_{NH}(x,y) = g_N(x,y) \geq T_H$$

$$g_{NL}(x,y) = g_N(x,y) \geq T_L.$$

After thresholding $g_{NH}(x,y)$ will have non-zero

Scanned with CamScanner

pixels which represents valid edge points.
we eliminate from $g_{NL}(x,y)$, all the non-zero pixels in $g_{NH}(x,y)$.

i.e, $g_{NL}(x,y) = g_{NL}(x,y) - g_{NH}(x,y)$

From the new $g_{NL}(x,y)$, valid edge points will be selected and added with $g_{NH}(x,y)$ which is the edge detected image.

~~Hough Transform.~~

27/3/18

Active Contour model (Snake).

* snakes are flexible. They can be used to describe the shape of an object in an image.
* It can be defined by splines (It is a type of curve defined by a set of points)
* A snake is defined by an energy function.
* To find the best fit b/w snake and object's shape, we minimise the energy.
* Snake can be considered as a rubber band, which when placed near an image will move towards the image and take the shape of the object in the image by minimising the energy associated with them.

Scanned with CamScanner

Snake's can be defined by
1) A set of 'n' points
2) Energy.

Snakes can be defined by a set of n points

$$V_i = (x_i, y_i) \quad \text{where } i = 0, 1, \ldots n-1$$

The energy of a snake is given by,

$$E_{snake} = \int_0^{n-1} \left(E_{internal}(x_i, y_i) + E_{image}(x_i, y_i) + E_{constraint}(x_i, y_i)\right)$$

$$= \int_0^{n-1} \left(E_{int.} V_i + E_{image} V_i + E_{constraint} V_i\right) di$$

where $E_{internal}$ is the internal energy caused by stretching or bending of splines.

E_{image} is the image energy which is the function of image features.

* $E_{constraint}$ → some systems allow the user interaction to guide the snake. So $E_{constraint}$ can be used to interactively guide the snake towards or away from a particular feature.

* $E_{internal}$:- Internal energy is defined as

$$E_{internal} = \alpha(i) \left|\frac{dv}{di}\right|^2 + \beta(i) \left|\frac{d^2v}{di^2}\right|^2$$

(elastic) energy where $\alpha(i)$ is the measure of the elasticity of the snake

(Bending energy) $\beta(i)$ is the measure of the stiffness of the snake

The first order term $\frac{dv}{di}$ make the snake act like a membrane.

The 2nd order term $\frac{d^2v}{di^2}$ make the snake to act like a thin plate.

* Image Energy (E_{image})

$$E_{image} = \omega_{line} E_{line} + \omega_{edge} E_{edge} + \omega_{term} E_{term}$$

where,

- E_{line} is defined by, $E_{line} = f(x,y)$ — intensity of image at (x,y). If some smoothing function is applied on the image, then $E_{line} = \text{filter}(x,y)$.
- E_{edge} is based on the image gradient:

$$E_{edge} = \text{mag.}(\nabla f(x,y))^2 = |\nabla f(x,y)|^2$$

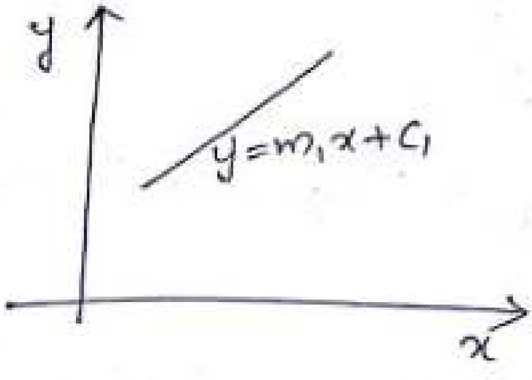
Scanned with CamScanner

- $E_{termination}$ is the energy related to the orientation and angles of termination or corners.

Hough Transform

The hough transform is the mapping from spatial domain (x-y plane) to parameter space (m-c plane).

Case 1 :- Suppose there is a single straight line in (x, y) coordinate system.

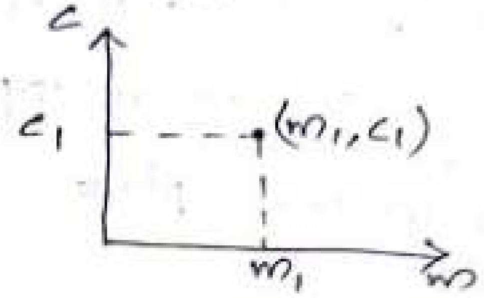


In the slope intercept form, the equation of a line is given as

$$y = mx + c$$

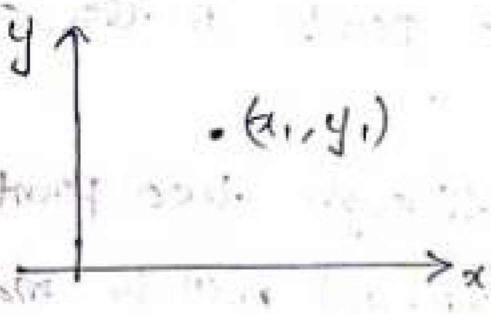
where m is the slope and c is the intercept.

For a particular line, the value of slope and intercept may be unique. Then the equation of the line is $y = m_1x + c_1$. ie, the value of slope and intercept are constant for a particular straight line in x-y plane. Since the line is specified by two parameters m_1 and c_1, the parameter space will be two dimensional. Now we can map this straight line to a parameter space.

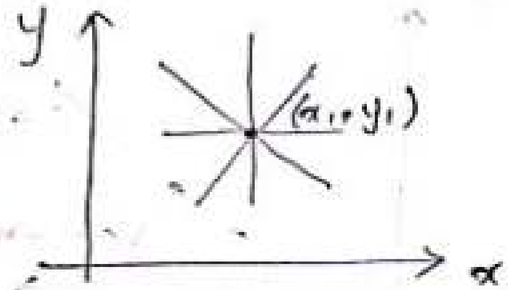


i.e, a straight line in x-y plane is mapped into a point in parameter space.

Case 2 :- Mapping a point in x-y plane to parameter space.



If a straight line $y = mx + c$ passes through a point (x_1, y_1), then the equation of the line is $y_1 = mx_1 + c$. Ideally there will be infinite no. of straight lines passing through the given point (x_1, y_1).



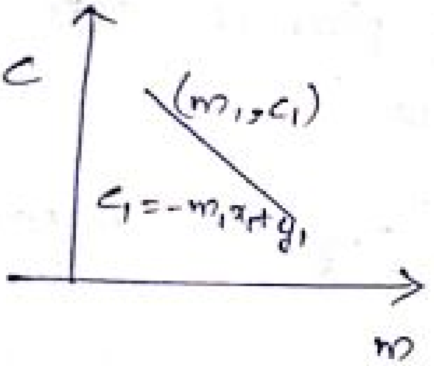
For each of the straight lines, the value of m and c will be different. i.e,

$$y_1 = m_1 x_1 + c_1$$
$$y_1 = m_2 x_1 + c_2$$
$$y_1 = m_3 x_1 + c_3 \; \ldots\ldots$$

Scanned with CamScanner

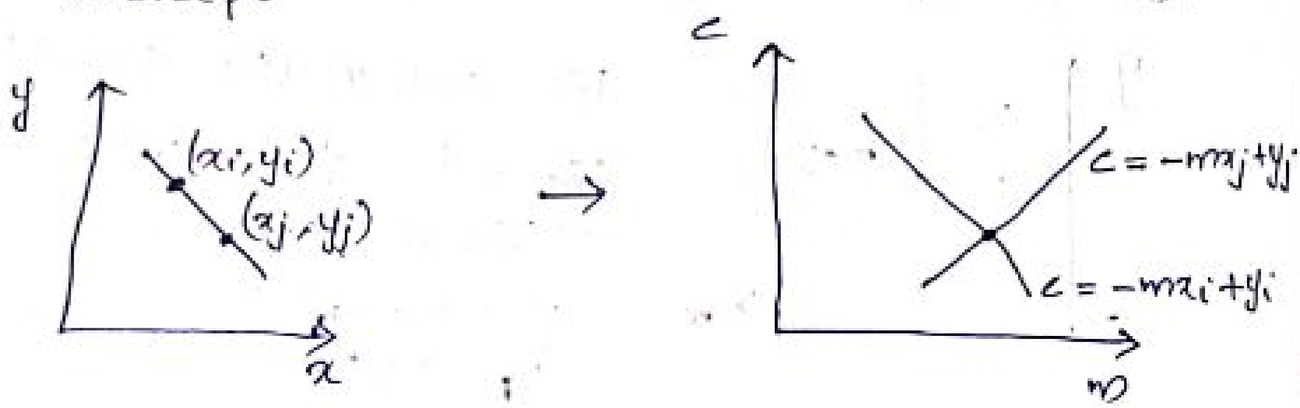
Thus m and c are varying and y's and x's are constant. For a particular straight line, $y_1 = m_1 x_1 + c_1$, $c_1 = -m_1 x_1 + y_1$

Thus a point in x-y plane is mapped to a single straight line in m-c plane.



Case 3 :- Mapping of two points in the spatial domain.

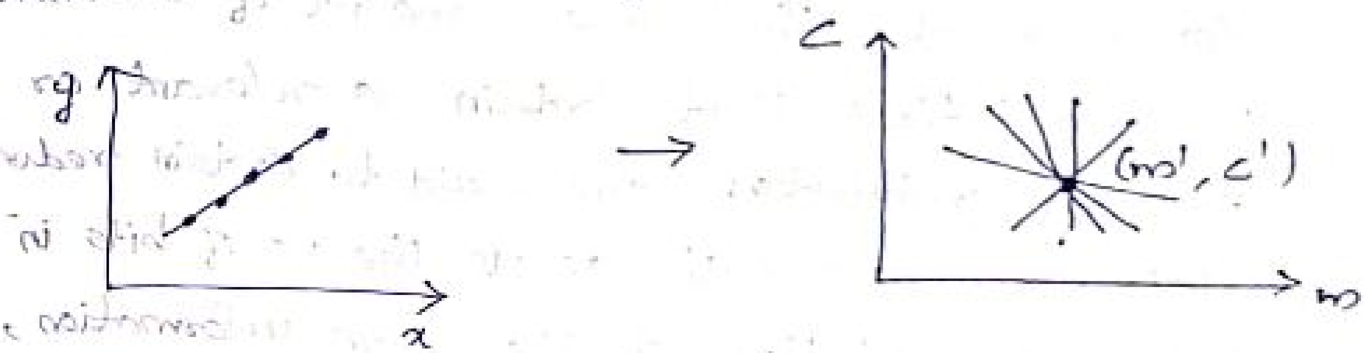
A straight line passing through two points in x-y plane (x_i, y_i) and (x_j, y_j) will be mapped to two straight lines in m-c plane that will intercept.



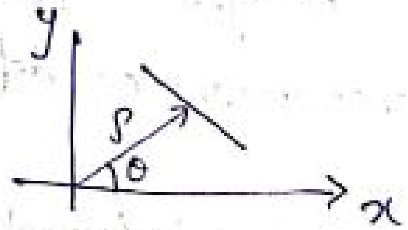
Scanned with CamScanner

Case 4:- Mapping infinite no. of points lying in the same straight line.

Each of these points will be mapped to a particular line in m-c plane. Thus there are infinite no. of collinear points in the x-y plane. ~~All this~~ which will be mapped to a no. of straight lines in m-c plane. All these straight lines intersect at a single point (m', c').



A particular difficulty with this approach is when a line approaches a ~~far~~ vertical direction, the slope of line becomes infinity. One way to get around this problem is to use the normal or polar representation of a line. i.e,



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

where ρ is the $\perp$ distance from the origin of the line

θ is the angle between x-axis and the $\perp$.

Scanned with CamScanner