MODULE 5

<u>Segmentation</u>: Segmentation subdivides an image into its constituent regions or objects. The level of subdivision depends on the problem being solved.

* Segmentation accuracy determines the sucess or failure of computerized analysis procedure ∴ care should be taken to improve the probability of accurate segmentation

* Most of segmentation algorithms are based on the properties of intensity value ie discontinuity and similarity

* discontinuity → The approach is to partition an image based on abrupt changes in intensity such as, point, line and edges.

* similarity → The approach is based on partitioning an image into regions that are similar according to a set of predefined criteria such as Thresholding, region growing, and region splitting and merging.

* The performance of segmentation can be improved by combining the above methods

* other methods of segmentations are based on morphology

10.2

* **Point, line and Edge Detection:** → are based on detecting sharp, local changes in intensity.

* Edge detectors are local image processing methods designed to detect edge pixels.
  * edge pixels are pixels at which the intensity of an image function changes abruptly, and edges are sets of connected edge pixels.

* A line is an edge segment in which the intensity of the background on either side of the line is either much higher or much lower than the intensity of the line pixels.

* Isolated point is a line whose length and width are equal to one pixel.

* The local changes in intensity can be detected using derivatives. ie the first & second derivatives are used for this purpose.
  * Derivatives of a digital function are defined in terms of differences.

∴ first order derivative at point x of one dimensional func $f(x)$ is given by.

$$\boxed{\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)} \quad —(1)$$

* 2^{nd} order derivative at point x of 1D func $f(x)$ is given by

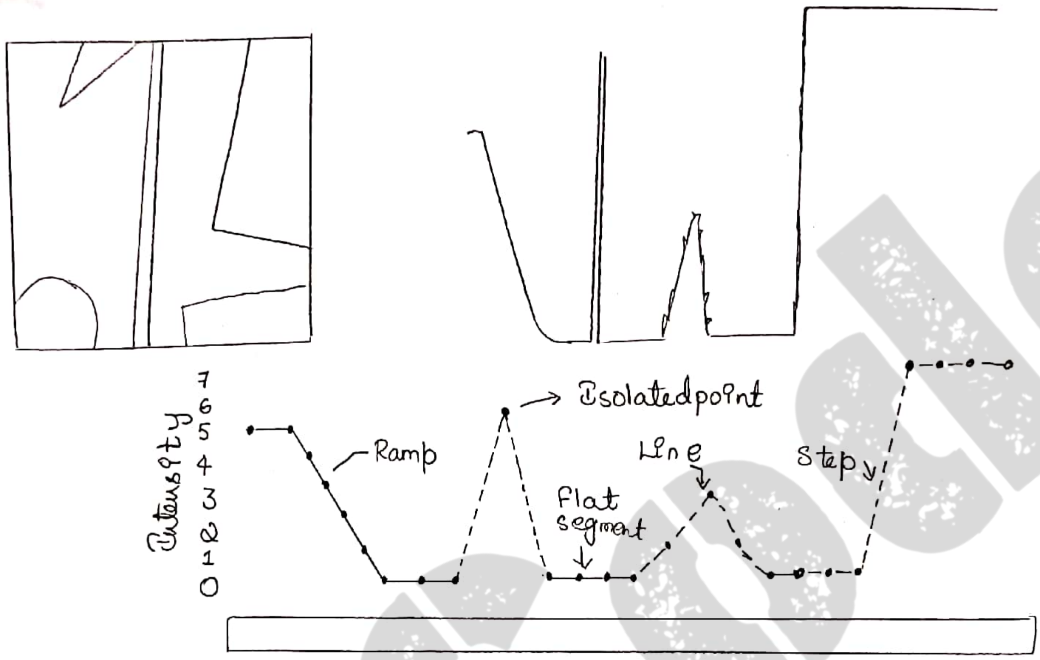
$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial f'(x)}{\partial x} = f'(x+1) - f'(x)$$
$$= f(x+2) - f(x+1) - [f(x+1) - f(x)]$$
$$= f(x+2) - f(x+1) - f(x+1) + f(x).$$
$$= f(x+2) - 2f(x+1) + f(x) \quad —(2)$$

eqn (2) is about point $x+1$ but we need at point x ∴ subt 1 from the argument of (2) & we get

$$\boxed{\partial^2/\partial x = f''(x) = f(x+1) + f(x-1) - 2f(x)} \quad -(3).$$

Example.

Fig. shows ~~the~~ an image that contain various solid objects, a line and a isolated point.



~~Summar~~ Summary: 1) 1st order derivatives produces thicker edges in an image. 2) second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points and noise. 3) second order derivatives produce a double-edge response at ramp & step transitions in intensity. 4) The sign of second derivative can be used to ~~detect~~ determine whether a transition into an edge is from light to dark (-ve 2nd derivative) or dark to light (+ve 2nd derivative)

∇^2 The computation of the derivatives are by using spatial masks called as spatial filters.

For a 3×3 mask, the response 'R' of the mask at the center point of the region is given by.

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

$$= \sum_{k=1}^{9} w_k z_k$$

Detection of Isolated points

Point detection is based on 2nd derivative as seen from the previous example - fig(1).

∴ using Laplacian.

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

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

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

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

ie mask

0	1	0
1	-4	1
0	1	0

&

1	1	1
1	-8	1
1	1	1

→ including the diagonal terms.

* The Point is detected at locn (x,y) on which the mask is centered if the absolute value of the response of the mask at that point exceeds a specified threshold, given by.

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

$g(x,y) \rightarrow$ o/p pixel value ; T is non negative threshold.

$$R = \sum_K w_K z_K$$

* The derivative mask, the co-efficients sums to zero, indicating that the mask response will be zero in areas of constant intensity.

Line detection:

* The 2^{nd} derivative laplacian mask can be used for line detection, But the double-line effect of 2^{nd} derivative must be handled properly.

* Since laplacian masks are isotropic [independent of direction], for detecting Lines in specified directions, the following line detection masks are used.

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°

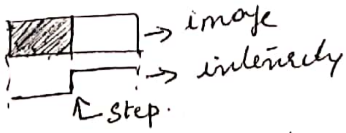
-1	-1	2
-1	2	-1
2	-1	-1

* The preffered direction of each mask is weighted with a larger co-efficient (ie 2) than other possible directions.

* If the image is filtered with the four masks individually then their responses are R_1, R_2, R_3, R_4 from left to right. where $R = \sum_{k=1}^{4} w_k z_k$ Then at a given point in the image $|R_k| > |R_j|$ for all $j \neq k$, that point is said to be more likely associated with a line in the direction of mask k.

Edge models: Edge models are classified according to their intensity profiles.

1) A step edge: It involves a transition between two intensity levels occuring ideally over the distance of one pixel.



2) Ramp edge: The digital images have edges that are blurred and noisy, ~~In such~~ due to image acquistion (lenses) & by electronic components of imaging system. In such cases edges are modelled as having intensity ramp profile

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

Edge detection:

W.K.T. Edge is nothing but set of points that change in intensity.

∴ & we can use 1^{st} & 2^{nd} derivatives to find discontinuities.

* 1^{st} derivative tells us where an edge is
* 2^{nd} " can be used to show edge direction

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

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \text{--- (1)}$$

→ This gradient vector points in the direction of the greatest rate of change of 'f' at location (x,y).

* The magnitude (length) of vector ∇f is given by

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

* The direction of the gradient vector is given by.

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] \quad \text{--- (3)}$$

∴ For a given 3×3 region of an image the following edge detection filters (masks) of gradient operators are used.

* 3 x 3 region of an image [z's are intensity values].

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

* The masks are used to compute the gradient at point z_5.

* The Roberts cross-gradient operators are given by.

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5) \; ; \; g_y = \frac{\partial f}{\partial y} = (z_8 - z_6).$$

-1	0
0	1

0	-1
1	0

* The Prewitts mask are given by.

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

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

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

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

* The Sobel operator are given by.

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

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

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

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

* The Prewitt mask for detecting diagonal edges.

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

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

* The Sobel mask for detecting diagonal edges.

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

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

Advanced Techniques for edge detection.

The edge detection methods discussed so far ~~in~~ are based on simply filtering an image with one or more masks, without considering the noise content & the nature of the edge.

The Marr-Hildreth edge detector or Laplacian of Gaussian (LoG) or Mexican hat operator.

ie The operator or filter uses the Gaussian for noise removal and the Laplacian for edge detection given by $\nabla^2 G$. where ∇^2 is Laplacian operator & G is 2-D Gaussian function.

$$\therefore G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{--- (1)}$$

$\sigma \rightarrow$ is standard deviation called as space constant.

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

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

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

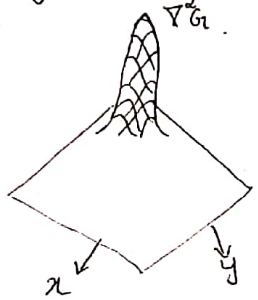
$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{--- (3)}$$

by simplifying - 3 we get

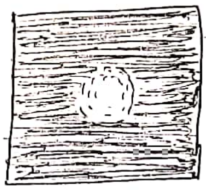
$$\therefore \nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \quad -(4)$$

The above expression is called the <u>Laplacian of a Gaussian</u> (LOG).

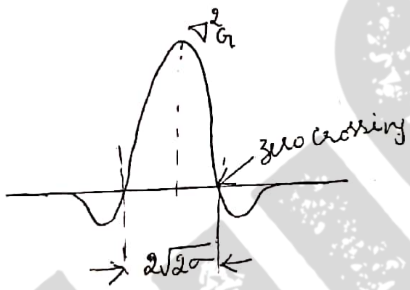
~~fig. below shows 3-D plot~~



a) 3-D plot of the -ve of LOG



b) -ve of the LOG as an image



c) cross section of (a) showing zero crossing.

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

d) 5X5 mask approximation to (a). * The -ve of this mask will be used in practice.

* zero crossing of the LoG occur at $x^2+y^2=2\sigma^2$.

* This LoG filter is convolved with an input image $f(x,y)$ to the output image given by

$$g(x,y) = \left[\nabla^2 G(x,y)\right] * f(x,y) \quad -(5)$$ and then finding

the zero crossing of $g(x,y)$ the <u>location of edges in $f(x,y)$ is determined</u>

eqn (5) is linear $\therefore$ can be written as

$$g(x,y) = \nabla^2\left[G(x,y) * f(x,y)\right] \quad -(6)$$

∴ From eq^n (6) The Marr-Hildreth edge-detection algorithm is as follows.

1) Filter the input image with an $n \times n$ Gaussian lowpass filter obtained by sampling. $G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$

2) Compute the Laplacian of the image resulting from step 1 using the 3 x 3 mask. $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ ie $\nabla^2 G(x,y)$.

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

3) Find the zero crossing of the image $g(x,y)$. to ~~detect~~ determine the edge.

* <u>The Canny edge detector</u>: This edge detector is superior than other edge detectors. The detecting edge is based on three basic objectives.

1) <u>Low error rate</u>: All edges should be found, ie the edges detected must be as close as possible to the true edges.

2) <u>Edge points should be well localized</u>. The edges located must be as close as possible to the true edges. ie. The distance between a point marked as an edge by the detector and the center of the true edge should be minimum.

3) <u>Single edge point response</u>: The detector should return only one point for each true edge point. ie. The No of local maxima around the true edge should be minimum. meaning that the detector should not detect multiple edge pixels for a single edge point pixel

⑥

* ~~Canny edge detector mask are~~

* The above objectives are achieved by following steps [Algorithm of canny edge detection].

→ Let $f(x,y)$ denote the i/p image

$G(x,y)$ " the Gaussian function

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

1) → Find the smoothed image $f_s(x,y)$ by convolving G & f.

$$\therefore f_s(x,y) = G(x,y) * f(x,y) \quad \text{— (2)}$$

2) → Compute the gradient magnitude & direction (angle).

as $$M(x,y) = \sqrt{g_x^2 + g_y^2} \quad \text{— (3)}$$

& $$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] \quad \text{— (4)}$$

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

3) → The $M(x,y)$ contains wide ridges around local maxima

∴ To thin these ridges use non maxima suppression, to get an image $g_N(x,y)$ with thinned edges.

4) → Then Threshold $g_N(x,y)$ to reduce false edge points.

* Edge linking and Boundary detection:

In practice there will be breaks in the edges due to noise or due to nonuniform illumination and other effect. ∴ Edge detection is followed by linking algorithms designed to assemble edge pixels into meaningful edges or region boundaries.

* There are three fundamental approaches to edge linking.

1) <u>Local processing</u>: It requires the knowledge about edge points in a local region (eg in a neighborhood) about every point (x, y), that has been declared an edge point. & all points that are similar according to predefined criteria are linked forming an edge of pixels.

2) <u>Regional processing</u>: It requires that points on the boundary of a region be known. Then link pixels on a regional basis. which results in a boundary of a region.

3) <u>Global Processing using Hough Transform</u>.

For <u>global processing</u> all the pixels are candidates for linking and thus have to be accepted & eliminated based on predefined <u>global properties</u>. ie linking edges is based on whether set of pixels lie on curves of a specified shape. once detected, these curve form the edges & region boundaries of interest.

<u>Hough Transform</u>.

next page.

10.3 Thresholding

* Thresholding is the simplest segmentation method
* The pixels are partitioned (classified) based on their intensity value.
* In an image $f(x,y)$ with an object (light) and background (dark), that separates these modes. Select a threshold T,
* ie Any point (x,y) in the image at which $f(x,y) > T$ is called an object. otherwise it is an background.
* ∴ The segmented image $g(x,y)$ is given by.

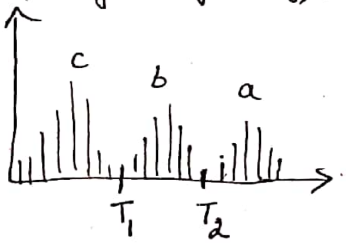
$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \rightarrow \text{object} \\ 0 & \text{if } f(x,y) \leq T \rightarrow \text{Background} \end{cases}$$

* Global Thresholding → when T is constant over an entire Image
* Variable Thresholding → when T can change over the Image
  - → Local or Regional Thresholding if T depends on a neighborhood of (x,y)
  - → Adaptive thresholding is T is a function of (x,y).

* Multiple thresholding

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

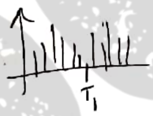
* histogram of $f(x,y)$.



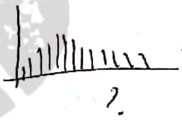
* The factors that affect the choice of threshold based on Valleys are
1) The ~~sp~~ separation between peaks
2) The noise content in the image
3) The relative size of objects and background
4) The uniformity of the illumination.
5) The uniformity of the reflectance.

* <u>The Role of Noise</u>, ~~and~~ <u>illumination and reflection</u>.

* if the image is corrupted with 10% of noise, still a Thresholding can be applied & if the image is corrupted by 50% noise then we do not get the peaks & valley hence Thresholding is not possible ie



10% noise



50% noise

* with improper illumination & Reflection

* <u>Basic Algorithm for Global thresholding</u>.

This algorithm works well when there is a reasonably clear Valley between two modes.

1) Select the intial ~~value~~ estimate of T, Typically the average intensity of the Image.
2) Segment the image using T ~~then~~ using

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

∴ $G_1 \Rightarrow$ pixels grater than T

$G_2 \Rightarrow$ Pixels less than or equal to T.

3) compute the average (mean) intensity values m_1 & m_2 of G_1 and G_2 respectively
4) compute the new Threshold $T = \frac{1}{2}(m_1 + m_2)$

~~5) T_new~~

5) Repeat steps 2 through 4 until the difference between values of T in ~~successi~~ successive iteration is smaller than a predefined value ΔT.

* optimum Global Thresholding using Otsu's method

→ This method is used to find the optimal value for the global thresholding.

* It is based on PDF of the intensity levels of each class and probability of occurance of each class.

Algorithm:

1) compute the normalized histogram of the input image.

i.e for MXN image with 'L' intensity levels [0, --- L-1]

where n_i is the No pixels of intensity i.

then normalized histogram is given by

$$P_i = \frac{n_i}{MN}$$

$$\therefore \sum_{i=0}^{L-1} P_i = 1 ;\ P_i \geq 0$$

2) compute the cumulative sums, $P_1(k)$ for $K = 0, 1, 2 --- L-1$ using

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

3) compute the cumulative means, $m(k)$ for $K=0,1,2,---L-1$ using

$$m(k) = \sum_{i=0}^{K} i P_i$$

4) compute the global intensity mean M_G using

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

5) compute the between class Variance $\sigma_B^2(k)$ for $k=0,1,2 \ldots L-1$ using

$\cancel{\sigma_B^2(k)} = \cancel{\max}$

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

6) obtain the otsu threshold k^*, as the value of k for which $\sigma_B^2(k)$ is maximum; If the max is not unique, obtain k^* by averaging the values of k corresponding to the various maxima detected.

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

7) obtain the separability measure 'η' by using

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

Then

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

* using Image Smoothing to improve Global Thresholding

* when noise cannot be reduced at the source,
& thresholding is the segmentation choice, then
Smooth the image prior to thresholding.

i.e. otsu's method may not work in the presence of noise.
→ Smoothing can produce a histogram with separated peaks.

* If the distribution is not balanced, no information can be extracted from the histogram
i.e Smoothing cannot help.

*** ∴ Edge extraction techniques (Laplacian) can be used for selecting the region that carry the Valuable information.

∴ Algorithm using Edges to Improve Global Thresholding.

1) compute an edge image as either the magnitude of the gradient, or absolute value of the Laplacian of $f(x,y)$
2) specify the threshold value, T.
3) Threshold the image from step 1, using T of step 2. to produce a binary image $g_T(x,y)$. This image is used as a mask image to select strong edge pixels.
4) compute a histogram using only the pixels in $f(x,y)$ that correspond to the location of the 1-valued pixels in $g_T(x,y)$.
5) use the histogram from step 4 to segment $f(x,y)$ globally. using the otsu's method.

*

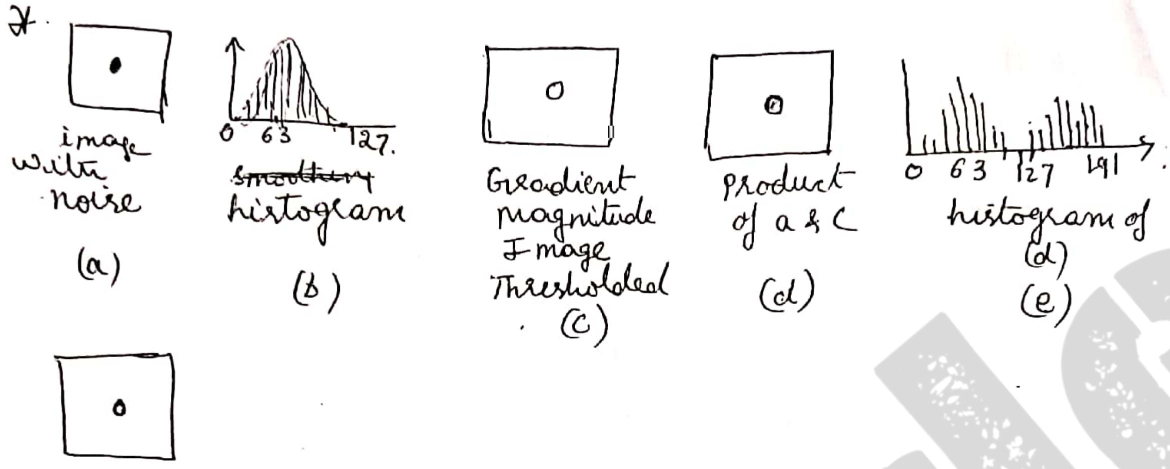


image with noise (a) — ~~smoothing~~ histogram (b) — Gradient magnitude Image Thresholded (c) — product of a & c (d) — histogram of (d) (e)

segmenting Img (a) with otsu's method with $T = 134$. (f).

* Multiple Thresholding using otsu's method

Algorithm:

1) for K classes, $C_1, C_2, \ldots C_K$ the between-class variance is given by

$$\sigma_B^2(K_1 K_2) = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + \cdots$$

$$= \sum_{k=1}^{K} P_k (m_k - m_G)^2 .$$

where $P_k = \sum_{i=C_i} P_i$ & $m_k = \frac{1}{P_k} \sum_{i=C_i} i P_i$

for Three classes the between class variance is given by [separated by 2 thresholds]

$$\sigma_B^2(K_1 K_2) = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2$$

2) The optimal thresholds K_1^* & K_2^* can be computed as

$$\sigma_B^2(K_1^* K_2^*) = \max_{0 < K_1 < K_2 < L-1} \sigma_B^2(K_1 K_2)$$

3) The separability degree can be measured as

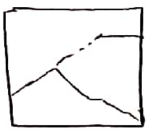
$$\eta(K_1^*, K_2^*) = \frac{\sigma_B^2(K_1^*, K_2^*)}{\sigma_G^2}$$

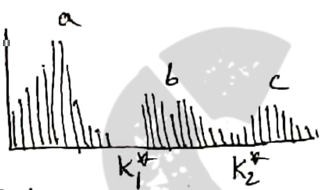
4) The Thresholded image is given by

$$g(x,y) = \begin{cases} a \ ; \text{ if } f(x,y) \le K_1^* \\ b \ ; \text{ if } K_1^* < f(x,y) \le K_2^* \\ c \ ; \text{ if } f(x,y) > K_2^* \end{cases}$$

where a, b, c are any three valid intensity values.

eg





** Variable Thresholding:

1) * Variable Thresholding using Image Partitioning

In order to overcome the problem of non uniform illumination and reflection, the image is partitioned and the thresholding is operated on each partition.

ie

2) * Variable Thresholding using local image properties.

Local properties (eg statistics) can be used for adapting the threshold

* Let σ_{xy} & m_{xy} denote the standard deviation and mean value of the set of pixels contained in a neighborhood S_{xy}, centered at co-ordinates (x,y) in an image

∴ The local thresholds are given by

$$T_{xy} = a\sigma_{xy} + b m_{xy} .$$

$$T_{xy} = a\sigma_{xy} + b m_G$$

where a & b are nonnegative constants
& m_G is the global image mean.

Then the segmented image is computed as

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

* By using local predicates Q. then

$$g(x,y) = \begin{cases} 1 ; & \text{if } Q(\text{local parameters}) \text{ is true} \\ 0 ; & \text{if } Q(\quad " \quad) \text{ is false} . \end{cases}$$

3) <u>Variable Threshold using moving averages</u>

* Pixels are visited following a zig zag path and the statistics are computed [used in document processing]
* Let z_{K+1} denote the intensity of the point in a scanning sequence at step $K+1$, then the moving average (mean intensity) at this point is given by.

$$m(K+1) = \frac{1}{n} \sum_{i=K+2-n}^{K+1} z_i$$

$$= m(K) + \frac{1}{n}\left(z_{K+1} - z_{K-n}\right)$$

where $n \rightarrow$ No of points used in computing the average

$$\therefore \boxed{T_{xy} = b m_{xy}}$$

where $b \rightarrow$ constant
& $m_{xy} \rightarrow$ moving average from above eqn at point (x,y) in the input image.

Region-Based Segmentation

The segmentation technique are based on finding the regions directly.

Region Growing:

Region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria for growth.

* The basic approach is to start with a set of "seed" points & from these grow regions by appending to each seed
* This growing is based on connectivity [4-connectivity, 8- "]

* A basic region-growing algorithm based on 8-connectivity is as follows.

Let $f(x,y)$ be the image to be segmented & $S(x,y)$ denote a seed array containing 1s at the location of seed point. & 0s elsewhere. & Q denote a predicate to be applied at each location (x,y)

Then
1) Erode all the connected components of 'S' until they are only one pixel wide.
2) Generate the binary image f_Q such that $f_Q(x,y) = 1$ if $Q(x,y)$ is true.
3) Create & grow the binary image 'g' where $g(x,y) = 1$ and if $f_Q(x,y) = 1$ and (x,y) is 8-connected to a seed in S.
4) The resulting connected components in 'g' are the segmented regions.

using 8-connectivity grow

eg.

1	1	9	9
5	1	9	9
5	1	1	3
5	5	3	3

	1	9	
5			3

Seeds.

1	1	9	9
5	1	9	9
5	1	1	3
5	5	3	3

	1	9	
5	1	9	9
5	1		3
5		3	3

4-connectivity

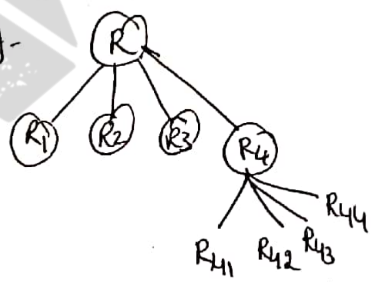
4. Region Splitting and Merging

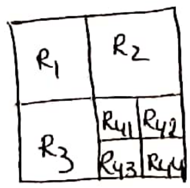
* Region splitting is the opposite of region growing
* Subdivide an image (split) into a set of arbitrary disjoint regions and then merge and/or split the regions ~~in~~ based on condition of segmentation.

Algorithm:

* Let R represent the entire image region & select a predicate Q.
  1) Split into four disjoint Quadrants such that any region R_i for which $Q(R_i)$ = False.
  2) When no further splitting is possible, merge any adjacent regions R_j and R_k for which $Q(R_j \cup R_k)$ = TRUE. [i.e weak edges are dissolved and strong edges are retained].
  3) Stop when no further merging is possible.

eg.

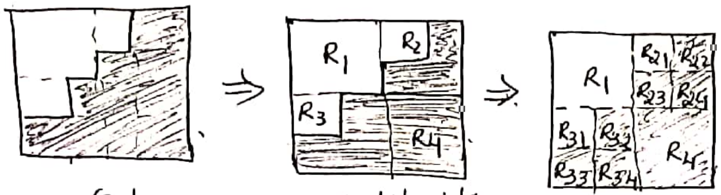




↳ This is called as Quadtree

* Segmentation using morphological watersheds.
  * Segmentation is based on watershed techniques.
  * watershed techniques is based on a topological interpretation of the image.
    → ie the intensity levels represents the height of the terrain that describe mountains and basins.
* For each basin a hole (morphology) in its minimum is supposed to be realized, from which the rising underground water spills and fills the basins.
* As the water rises, the level reach the border of the basin and two or more adjacent basins tend to merge together.
* Dams are required for maintaing a separation between basins
* These dams are the border of the region of the segmentation
* The dams can be build using morphological dilation.
  * The dams can be realized setting the pixels at L levels of intensity in [0, L-1].
* The watershed algorithm is applied to the gradient of the image to be segmented.

eg:



(a) Given image f(x)

Split into $R_1, R_2, R_3 R_4$

R_1 & R_4 are homogeneous. regions hence do not distrub.

split R_2 & R_3

→ R_{21}, R_{31} are undistrubed

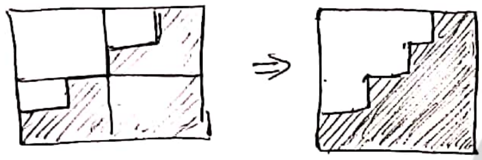
R_{22}, R_{23} & R_{24} → merge

R_{32}, R_{33} & R_{34} → merge

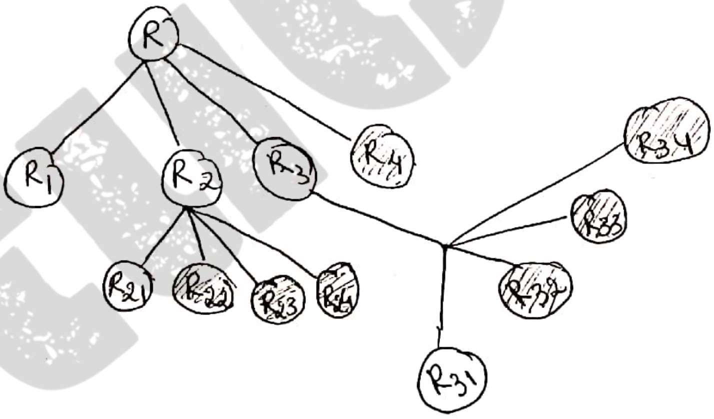
↳ no further splitting is Possible

∴

* Segmented Image



* Quadtree Representation



(5)

* The wavelet function $\psi(x)$ can be expressed as.

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\,\varphi(2x-n). \quad \text{--- (9)}$$

||ler to scaling func eqn (7).

$h_\psi(n) \Rightarrow$ wavelet function co-efficients

$h_\psi \Rightarrow$ wavelet vector

* The relation betn $h_\psi(n) \rightarrow$ wavelet function & $h_\varphi(n) \rightarrow$ scaling function is given by.

$$\boxed{h_\psi(n) = (-1)^n h_\varphi(1-n)} \quad \text{--- 10}$$

This relation is used in subband coding & decoding filters.

eg refer example 7.6

Representation and Description:

After an image is segmented into regions, the resulting aggregate of segmented pixels is represented and described for further computing processing.

* [Regions may be represented by its boundary, & the boundary is described by its features such as its length, the orientation of the straight line joining its extreme points & the NO of concavities in the boundary.]

* The segmented region is represented as
  - → In terms of its external characteristics (boundary)
  - → In terms of its internal characteristics (Pixels comprising the region).

* Description of the region based on the chosen representation eg. boundary → length, orientation of straight line joining the extreme points, NO of concavities

∴ Descriptors should be insensitive to changes in size, translation, rotation.

Representation of Schemes:

1) Boundary (Border) following or Moore boundary tracking Algorithm.

* For a given binary region 'R' & its boundary,

1) Let the starting point b_0, be the uppermost, leftmost point, in the image is labelled as 1. Denote c_0 the west neighbor of b_0 as shown in fig. where c_0 → background point. Examine the 8-neighbors of b_0 starting at c_0 and proceeding in a clockwise direction. Let b_1 denote the first neighbor

(7)

* watershed with marker:

Applying watershed technique in real image is difficult as it consists of noise and irrelevant details which leads to oversegmentation.

*

* The above problem can be handled limiting the flooding through markers

→ External markers: Points along the watershed associated with to the background

→ Internal markers: Points in region are connected components with same intensity associated to the object of interest