MODULE V

IMAGE SEGMENTATION

↳ Image Segmentation refers to the process of partioning an image into groups of pixels which are homogeneous w.r.t some criterion.

↳ Different groups must not intersect with each other and adjacent groups must be heterogenous.

↳ Segment^n algms are area oriented instead of pixel oriented.

↳ The result of Segmentation is the splitting up of the image into connected areas.

Classification of Image Segment^n techq:

↳ I.S can be broadly classified into 2.

- local segmentation.
- global segmentation.

Local Segmentation.

↳ it deals with segmenting sub images which are small windows on a whole image.

↳ No. of pixels available to local segm^n is much lower than global segment^n.

Global Segmentation

↳ Global. Seg^n is concerned with segmenting a whole image. G.S deals mostly with segments consistg of large No. of pixels. ~~available to~~

↳ Estimated parameter values for global segments more robust.

↳ I.S can be approached from 3 different perspectives.

1. Region approach.
2. Boundary "
3. Edge "

↳ $Segment^n$ subdivides an image into its constituent regions or objects.

↳ level of detail to which the subdivision is carried depends on the pblm being solved. ie, $segm^n$ should stop when the objects or region of interest in an $appl^n$ have been detected.

Applications

- Industrial inspection $appl^ns$, atleast some measure of control over the environment typically possible.

↳ Autonomous target acquisition, slm designer has no control over the operating environment

↳ Use of infrared Imaging by the military to detect objects with strong heat signatures.

such as equipment & troops in motion.

FUNDAMENTALS OF SEGMENTATION

↳ Let R represent the entire spatial region occupied by an image.

↳ I.S as a process that partitions R into n subregions $R_1, R_2, \ldots R_n$, such that:

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

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

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

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

e) $Q(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions R_i & R_j.

↳ $Q(R_K)$ is a logical predicate defined over the points in set R_K & ϕ is the null set.

↳ Symbol $\cup$ & $\cap$ represent set union & intersⁿ.

↳ (Two regions R_i & R_j are said to be adjacent if their union forms a connected set.)

↳ Condition (a) indicates that the segmentⁿ must be complete; ie, every pixel must be in a region

↳ Condition (b) requires that pts in a region be connected in some predefined sense. (4- or 8-connected)

↳ condition (c) indicates that the regions must be disjoint.

↳ Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region.

↳ condition (e) indicates that two adjacent regions R_i & R_j must be different ~~in the~~

↳ Segment^n algms for monochrome images generally are based on one of basic categories dealing with properties of intensity values: discontinuity & similarity.

↳ Edge based segmentation: Boundaries of regions are sufficiently different from each other & from the bkgd to allow boundary detection based on local discontinuities in intensity.

↳ Region based segm^n: Based on partitiong an image into regions that are ||^r according to ~~the~~ a set of predefined criteria.

Scanned by CamScanner

REGION APPROACH TO IMAGE SEGMENTATION

↳ Regions in an image are a group of connected pixels with similar properties.

↳ In the region approach, each pixel is assigned to a particular object or region.

↳ In the boundary approach, the attempt is to locate directly the boundaries that exist b/w the regions.

↳ In the edge approach, the edges are identified first and then they are linked together to form required boundaries.

1. Region Growing: R.G is a procedure that gps pixels or subregions into larger region based on predefined criteria for growth.

↳ R.G is an region based approach to Image Segm^n in which neighbouring pixels are examined and added to a region class if no edges are detected.

↳ This process is iterated for each boundary pixel in the region.

↳ if adjacent regions are found, a region-merging algorithm is used in which weak edges are dissolved & strong edges are left intact.

↳ Region growing 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 neighbouring pixels as possible that meet the homogeneity criterion.

↳ The resultant segment is then removed from the process.

↳ A new seed is chosen from the remaining pixels. This continues until all pixels have been allocated to a segment.

↳ As pixels are aggregrated, the parameters for each segment have to be updated.

↳ The resulting segmentation depend on the initial seed chosen and the order in which the neighbouring pixels are examined.

↳ The selection of homogenity criteria in image growing depends not only on the pblm under consideration but also on the type of image to be segmented.

↳ Region growing algms vary depending on the criteria used to decide whether a pixel should be included in the region or not, the connectivity type used to determine neighbours, & the strategy used to visit neighbouring pixels.

Scanned by CamScanner

Advantages over conventional segm^n techqs.

↳ The borders of regions found by region growing are perfectly thin & connected.

↳ Algm is also very stable w.r.t noise

A basic region-growing Algm based on 8-connectivity may be stated as follows:

Let: $f(x,y)$, denote an i/p image array

$S(x,y)$, denote a seed array containing 1's at the locns of seed points & 0's elsewhere.

Q, denote a predicate to be applied at each locn (x,y).

Arrays f & S are assumed to be of same size.

1. Find all connected components in $S(x,y)$ & erode each connected component to one pixel; label all such pixels found as 1. All other pixels in S are labeled 0.

2. 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, at those coordinates; otherwise, let $f_Q(x,y) = 0$.

3. Let g be an image formed by appending

to each seed point in S, all the 1-valued pts in f_Q that are 8-connected to that seed point.

4. Label each connected component in g with a different region label (eg: 1, 2, 3, ...). This is the segmented image obtd by region growing.

↳ To specify a predicate, appending to each seed all the pixels that are 8-connected to that seed are similar to it.

- Using intensity differences as a measure of similarity, Predicate applied at each locn (x,y) is

$$Q = \begin{cases} \text{TRUE}, & \text{if the absolute difference of the intensities b/w the seed \& the pixel at } (x,y) \text{ is } \leq T. \\ \text{FALSE}, & \text{otherwise.} \end{cases}$$

where T is a specified Threshold.

2. Region Splitting

↳ R.S is a top down approach.

↳ It begins with a whole image & divides it up such that the segregated parts are more homogenous than the whole.

↳ Splitting alone is insufficient for reasonable segmentation, as it severely limits the shapes of segments.

↳ A Merging phase after the splitting phase is always desirable which is termed as the split & merge algm.

3. Region Splitting & Merging

↳ Region splitting and merging is an I.S techq that takes spatial inform^n into considerat^n.

↳ The region-splitting &-merging is as follows.

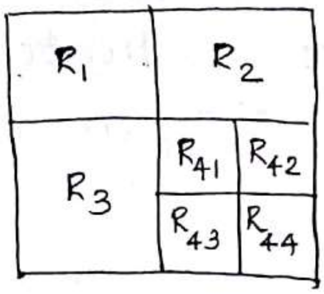
a) Splitting

1. Let R represent the entire image. Select a predicate P.

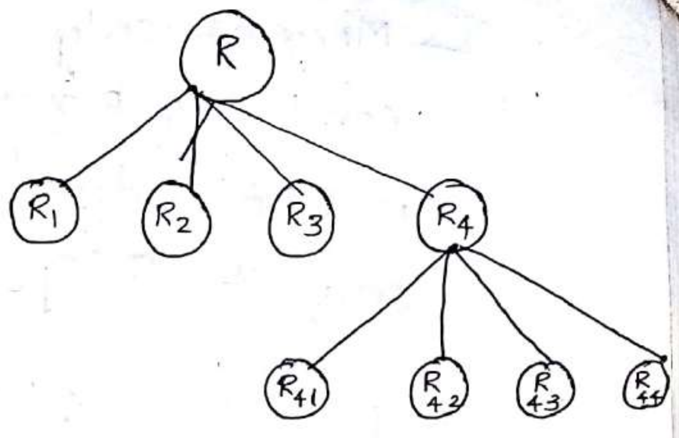
2. Split or subdivide the image successively into smaller & smaller quadrant regions.

- The splitting techq has a convenient represn in the form of a structure called a quadtree.

- In a quad tree, the root of the tree corresponds to the entire image and each node corresponds to a subdivision.



a) Splitting of an image.



b) Representation by a quadtree

↳ The final partition is likely to contain adjacent regions with identical properties. This drawback may be fixed by applying merging, & merging only adj. regions whose combined pixels satisfy the predicate P.

b) <u>Merging</u>

↳ Merge any adjacent regions that are similar enough.

↳ The procedure for <u>split & merge algm</u> is given below:

1. Start with the whole image.
2. If the variance is too large, break it into quadrants.
3. Merge any adjacent regions that are similar enough.
4. Repeat steps (2) & (3) iteratively until no more splitting or merging occurs.

↳ Merging only adjacent regions whose combined pixels satisfy the predicate Q.

↳ ie, two adjacent regions R_j & R_K are merged only if $Q(R_j \cup R_K) = TRUE$

1. Split into 4 disjoint quadrants any region R_i for which $Q(R_i) = FALSE$.
2. When no further splitting is possible, merge any adjacent regions R_j & R_K for which $Q(R_j \cup R_K) = TRUE$.
3. Stop when no further merging is possible.

- Adv of Split & Merge Algm

- Simpler & Faster algm, becoz testing of predicate is limited to individual quadregions.
- Simplification is still capable of yielding good segmentⁿ results.

- Segment the region of interest using the following predicate.

$$Q = \begin{cases} TRUE & \text{if } \sigma > a \text{ AND } 0 < m < b \\ FALSE & \text{otherwise} \end{cases}$$

where m & σ are the mean & std deviation of pixels in a quadregion & a & b are constants.

Scanned by CamScanner

↳ The segmentⁿ effectively partitioned the image into 3 distinct areas that correspond to the 3 principal features in the image.

- background
- dense "
- sparse regions.

↳ Properties based on the mean & std deviatⁿ of pixel intensities in a region attempt to quantify the texture of the region.

↳ The concept of texture segmⁿ is based on using measures of texture in the predicates.

IMAGE SEGMENTATION BASED ON THRESHOLDING

↳ //Thresholding techq produce segments having pixels with similar intensities //

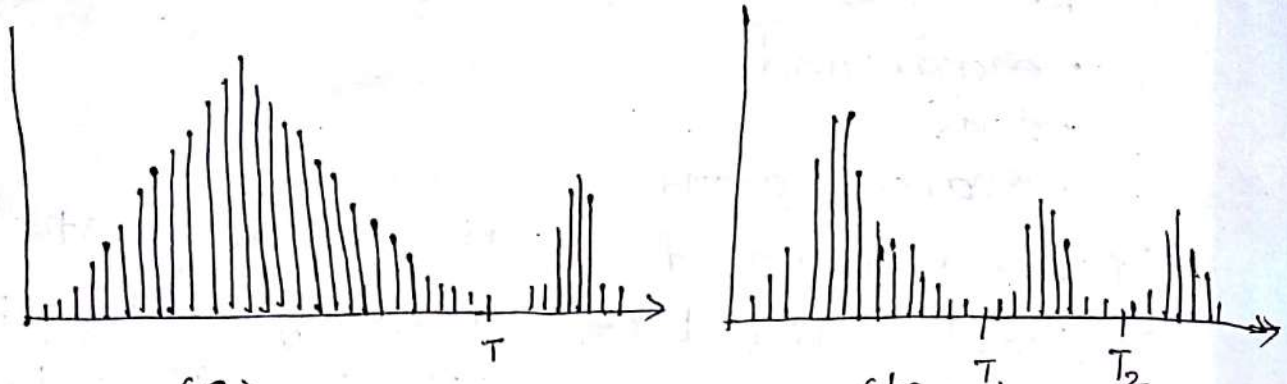
↳ //Thresholding is a useful techq for establishing boundaries in images that contain solid objects resting on a contrasting background //

↳ There excist a large no. of gray-level based segmentⁿ methods using either global or local image information.

↳ //The thresholding techq requires that an object has homogenous intensity & a bkgd with different intensity level. Such an image can be segmented into 2 regions by simple thresholdg //

↳ //Regions were first finding edge segments & then attemptg to link the segments into boundaries In thresholding, techqs for partitioning images direct into regions based on intensity values & properties of these values.

- The basics of intensity thresholding.



↳ The intensity histogram in (a) corresponds to an image $f(x,y)$, composed of light object on a dark bkgd, in such a way that obj & bkgd pixels have intensity values grouped into 2 dominant modes.

↳ Any pt (x,y) in the image at which $f(x,y) > T$ is called an object point; otherwise, the point is called a backgd pt.

↳ Segmented image $g(x,y)$ is given by:

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

↳ when T is a constant applicable over an entire image, the process given in (1) is referred to as global thresholding.

↳ When the value of T changes over an image, we use the term variable thresholding.

↳ The term local or regional thresholding is used sometimes to denote variable thresholding in which the value of T at any point (x,y) in an image depends on properties of a neighborhood of (x,y).

↳ If T depends on the spatial coordinates (x,y) themselves, then variable thresholding is often referred to as dynamic or adaptive thresholdg.

↳ Fig (b) shows a more difficult thresholdg problem involving a histogram with 3 dominant modes.

↳ Multiple thresholding classifies a point (x,y) as belonging to the bkgd if $f(x,y) \le T_1$, to one object class if $T_1 < f(x,y) \le T_2$ & to other object class if $f(x,y) > T_2$

Segmented image is given by :

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

a, b & c are any 3 distinct intensity values.

- Key factors affecting the properties of valleys are
  - o Separation b/w peaks.
  - o noise content in the image.
  - o relative sizes of objs & bkgd
  - o uniformity of the illumination source.
  - o uniformity of the reflectance properties of the image.

The role of noise in image thresholding

(Refer Gonzalez Page 740 fig(a))

↳ In Fig. (a) & (d) illustration of how noise affects the histogram of an image

↳ Segmenting the image into two regions is a trivial task involving a threshold placed anywhere b/w the 2 modes.

↳ A threshold placed midway b/w the two peaks would do segmenting the image.

↳ Eg: Page 740 using image & its correspondg histogram.

The role of illumination & reflectance.

↳ illuminⁿ & reflectance play a central role in the success of I.S using thresholding or other segⁿ techqs.

↳ Therefore, controlling these factors when it is possible to do so should be the first step considered in the solution of segmⁿ pblm.

↳ 3 basic approaches:

- To correct the shading pattern directly.
  (Eg: Nonuniform illuminⁿ can be corrected by multiplying the image by the inverse of pattern, which is obtd by imaging a flat surface of const intensity.)

- To attempt to correct the global shading pattern via processing (Eg: top-hat transformⁿ)

- "work around" nonuniformities using variable thresholding.

Scanned by CamScanner

LOCAL THRESHOLDING

- A single threshold will not work well when we have uneven illumination due to shadows or due to the direction of illumination.
- To partition the image $m \times m$ sub images & then choose a threshold.

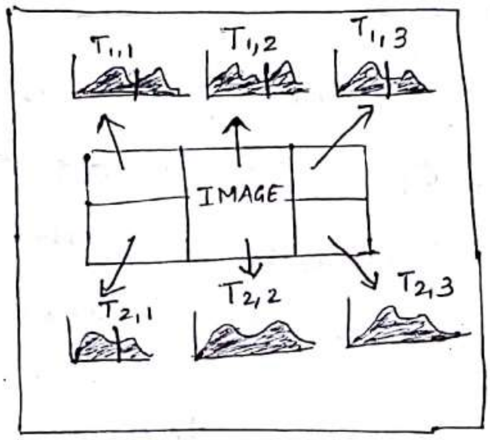


Fig: subimages of an image.

- T_{ij} for each sub image.
- Local Thresholdg can be used effectively when the gradient effect is small w.r.t the chosen sub image size.
- L.T techq, the threshold value T depends on gray levels of $f(x,y)$ & some local image properties of neighboring pixels such as mean or variance.
- Threshold function $T(x,y)$ is given by:

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

where $T(x,y) = f(x,y) + T$

↳ First the gray level histogram for a sub image is approximated by a sum of two Gaussian distributions, then the threshold is obtd by minimizing the classif^n error w.r.t the threshold value.

MULTILEVEL THRESHOLDING

↳ Thresholding method can be extended to an arbitrary no. of thresholds, because the separability measure on which it is based to an arbitrary no. of classes.

↳ Let $\{0, 1, 2, \ldots L-1\}$ denote L distinct intensity levels in a digital image of size $M \times N$ pixels.

n_i: denote the no. of pixels with intensity i.

↳ Total no, MN, of pixels in the image as:

$$MN = n_0 + n_1 + n_2 + \cdots + n_{L-1}$$

↳ The normalized histogram has components

$$P_i = n_i / MN,$$

$$\sum_{i=0}^{L-1} P_i = 1 \quad , \quad P_i \geq 0 .$$

↳ In the case of K classes, $C_1, C_2, \ldots C_K$, the b/w class variance generalizes to the expression

$$\sigma_B^2 = \sum_{K=1}^{K} P_K (m_K - m_G)^2 . \quad \text{—— (1)}$$

where $$P_K = \sum_{i=C_K} P_i$$

$$m_K = \frac{1}{P_K} \sum_{i \in C_K} i P_i$$

m_G is the global mean given as

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

↳ The K classes are separated by K-1 thresholds whose values, $k_1^*, k_2^*, \ldots, k_{K-1}^*$, are the values that maximize ①

$$\sigma_B^2(k_1^*, k_2^*, \ldots, k_{K-1}^*) = \max_{0<k_1<k_2<\ldots k_{n-1}<L-1} \sigma_B^2(k_1, k_2 \ldots k-1)$$

↳ For three classes consisting of 3 intensity intervals the between class variance is given by :

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

where,
$$P_1 = \sum_{i=0}^{k_1} P_i$$

$$P_2 = \sum_{i=k_1+1}^{k_2} P_i$$

$$P_3 = \sum_{i=k_2+1}^{L-1} P_i$$

EDGE DETECTION

↳ Edge detection is the process of finding meaningful transitions in an image.

↳ Edge detection is one of the central tasks of the lower levels of IP.

↳ The points where sharp changes in the brightness occur typically form the border between different objects.

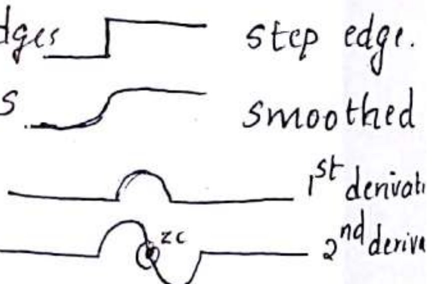
↳ These points can be detected by computing intensity differences in local image regions.

Importance of Edge Detection.

↳ ED is a pblm of fundamental importance in image analysis.

↳ Purpose of ED is to identify areas of an image where a large change in intensity occurs.

↳ Imp features can be extracted from the edges of an image (lines, corners, curves).



EDGE DETECTION - Edge Operators

1. Gradient Operator

↳ A gradient is a 2D vector that points to the direction in which the image intensity grows fastest.

↳ The gradient operator ∇ is given by:

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

If the operator ∇ is applied to the function f then :

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

↳ The two functions that can be expressed in terms of the directional derivaties are gradient magnitude & gradient orientation.

↳ To compute the magnitude $\|\nabla f\|$ of gradient and the orientation $\phi(\nabla f)$.

↳ The gradient magnitude gives the amount of the difference b/w pixels in the neighbourhood which gives the strength of the edge.

↳ The gradient magnitude is defined by :

$$|\nabla f| = \left\| \begin{bmatrix} G_x \\ G_y \end{bmatrix} \right\| = \left\| \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \right\| = \left[G_x^2 + G_y^2 \right]^{1/2}$$

↳ The magnitude of the gradient gives the maximum rate of increase of f(x,y) per unit distance in the gradient orientation of $|\nabla f|$

↳ The gradient orientation gives the direction of the greatest change based on intensity.
The gradient orientation is given by :

$$\phi(\nabla f) = \tan^{-1}\left[\frac{G_y}{G_x}\right]$$ where the angle is measured w.r.t the x-axis.

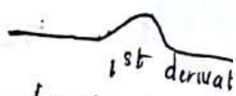
↳ An edge pixel is described using two imp. features :

(i) Edge strength, which is equal to the magnitude of the gradient.

(ii) Edge direction, which is equal to the ~~equal~~ angle of the gradient.

2. <u>Edge detection using first-order derivatives.</u>

↳ The derivative of a digital pixel grid can be defined in terms of differences.



↳ The first derivative of an image containing gray value pixels must fulfill the conditions.

- it must be <u>zero</u> in <u>flat segments</u> ie, in area of const gray-level values
- it must be <u>non-zero</u> at the <u>beginning of a gray-level step or ramp</u>

↳ The first order derivative of a 1D function f(x) can be obtd using

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

↳ An image is a funⁿ of two variable $f(x,y)$.

↳ Pixel discontinuity can be determined along 8 possible directions such as up, down, left, right and along 4 diagonals.

↳ Other method of calculating the first-order derivative is given by estimating the finite difference : 2D.

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

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

The finite difference can be approximated as :

$$\frac{\partial f}{\partial x} = \frac{f(x+h, y) - f(x,y)}{h_x} = f(x+1, y) - f(x,y) \quad [h_x = 1]$$

$$\frac{\partial f}{\partial y} = \frac{f(x, y+h) - f(x,y)}{h_y} = f(x, y+1) - f(x,y) \quad [h_y = 1]$$

3. Robert's Kernel

↳ objective is to determine the differences b/w adjacent pixels, one way to find an edge is to explicitly use {+1, -1} that calculates the difference b/w adj. pixels.

↳ Mathematically, these are called forward differences.

↳ R.K are too small to reliably find edges in the presence of noise.

↳ Simplest way to implement the first order partial derivative is by using Roberts cross gradient operator. (Gonzalez TB)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

— ①

Masks used to compute the gradient at pt z_5 (z's are graylevel values)

3x3 region of an image.

✓ $\frac{df}{dx} = f(i,j) - f(i+1, j+1)$; $G_x = (z_9 - z_5)$ &

& $\frac{df}{dy} = f(i+1, j) - f(i, j+1)$ $G_y = (z_8 - z_6)$

↳ Roberts operator masks are given by

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

↳ Masks of size 2x2 are awkward to implement becoz they donot have a clear center.

Adv

- Filters have the shortest support, thus the position of the edges is more accurate
- Vulnerability to noise.

4. Prewitt Kernel

↳ P.K are based on the idea of central difference.

↳ P.K edge detector is a much better operator than the Roberts Operators.

↳ Consider the arrangements of pixels about the central pixel [i, j] as shown below:

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & [i,j] & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

↳ An approach using masks of size 3x3 is given by. from ① ✓

$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$

$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

or

↳ Partial derivates of Prewitt operator are calculated as:

$$G_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$\& \quad G_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)$$

↳ The constant c in above expression implies the emphasis given to pixels closer to the centre of the mask, G_x & G_y are the approximat^ns at [i, j].

↳ Setting c=1, the Prewitt operator mask is obtd as,

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

↳ Prewitt masks have longer support.

↳ Prewitt mask differentiates in one direction & avg's in other direction.

↳ E.D is less vulnerable to noise.

5. Sobel Kernel

↳ Sobel kernel relies on central differences, but gives greater weight to the central pixels when avging.

↳ S.K can be 3×3 approximations to first derivates of Gaussian kernels.

↳ Partial derivates of the Sobel operator are calculated as:

from ① $G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$ (Gonzlz)

$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$ or

$$G_x = (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6)$$

$$G_y = (a_6 + 2a_5 + a_4) - (a_0 + 2a_1 + a_2)$$

↳ Sobel masks in matrix form are given as:

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(smoothing)

↳ Noise-suppression characteristics of a Sobel mask is better than that of a Prewitt mask.

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

6. Second derivative method of detecting edges in an Image.

↳ Finding the ideal edge is equivalent to finding the point where the derivative is maximum or minimum.

↳ Maximum or minimum value of a function can be computed by differentiating the given function & finding places where derivative is zero.

↳ Finding the optimal edges is equivalent to finding places where the second derivative is zero.

↳ The differential operators can be applied to images; the zeros rarely fall exactly on a pixel.

↳ Typically, the fall b/w pixels. The zeros can be isolated by finding the zero crossings.

↳ Zero crossing (ZC) is the place where one pixel is +ve & a neighbouring pixel is -ve.

↳ Pblms with ZC methods are:

- ZC methods produce two pixel thick edges.
- ZC methods are extremely sensitive to noise

↳ For images, ‖r to gradient magnitude, that measures the second derivative obtd by taking dot product of ∇.

$$\nabla \cdot \nabla = \begin{bmatrix} \partial/\partial x \\ \partial/\partial y \end{bmatrix} \cdot \begin{bmatrix} \partial/\partial x \\ \partial/\partial y \end{bmatrix} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

✓

↳ The operator $\nabla \cdot \nabla = \nabla^2$ is called Laplacian operator

↳ Laplacian operator applied to the funⁿ f:

$$\nabla^2 f = \left[\begin{bmatrix} \partial/\partial x \\ \partial/\partial y \end{bmatrix} \cdot \begin{bmatrix} \partial/\partial x \\ \partial/\partial y \end{bmatrix}\right] f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

✓

↳ Laplacian operation can be expressed in terms of difference eqns as below:

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y) \quad \text{\&}$$

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

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

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

↳ 3x3 Laplacian operator is given by:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

↳ Laplacian operator substracts the brightness values of each of the neighbouring pixels from the central pixel.

↳ When a discontinuity is present within the neighbourhood in the form of a pt, line or edge the result of Laplacian is a non-zero value.

↳ it may be either +ve or -ve dependg where the central pt lies w.r.t the edge.

↳ Laplacian operator is rotationally invarient.

↳ The Lapln. operator doesnot depend on direction as long as they are orthogonal.

Disadv

1. Useful directional information is not available by the use of a Laplacian operator.

2. The Laplacian, being an approximation to second derivative, doubly enhances any noise in the image.

The Laplacian : (Gonzalz TB)

↳ 2D Fun^n $f(x,y)$ is a 2^{nd} order derivative defined as:

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

↳ For 3x3 region,

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \quad \text{--- (2)}$$

↳ A digital approximation including the diagonal neighbors is given by:

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9) \quad \text{—} \; (3)$$

↳ Laplacian mask used to implement eqn (2) & (3)

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

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

MODULE V Continued...

7. LAPLACIAN OF GAUSSIAN (LOG)

- A prominent source of performance degradation in Laplacian operator is noise in the i/p image.
- The noise effects can be minimised by smoothing the image prior to edge enhancement.
- The LOG Operator smooths the image through convolution with a Gaussian-shaped kernel followed by applying the Laplacian operator.
- Sequence of operation involved in an LOG Operator is given below:

step 1 : Smoothing of the i/p image $f(x,y)$.

$$g(x,y) = f(x,y) \otimes h(x,y)$$

i/p image $f(x,y)$ is smoothed by convolving it with the Gaussian mask $h(x,y)$ to get the resultant smooth image $g(x,y)$.

step 2 : Laplacian operator is applied to the result in step 1.

$$g'(x,y) = \nabla^2(g(x,y))$$
$$= \nabla^2(f(x,y) \otimes h(x,y)).$$

∇^2 : Laplacian operator

$f(x,y)$: i/p image

$h(x,y)$: Gaussian mask..

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

σ : S.D (sometimes σ is called space const).

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

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

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

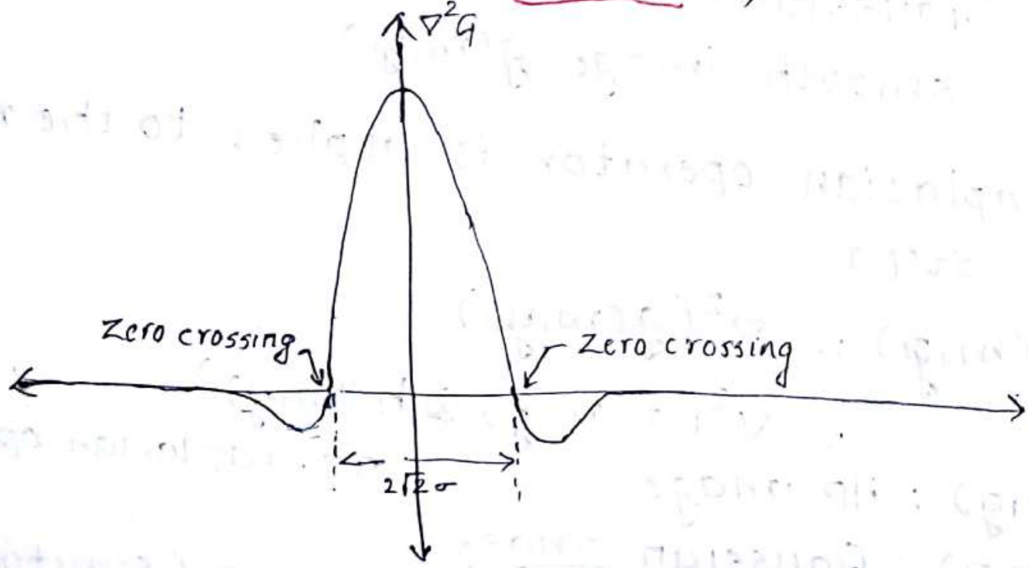
$$= \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

This expression is called the Laplacian of a Gaussian (LoG)

↳ The LoG function sometimes is called the Mexican hat operator.

Disadv

↳ The LoG operator being a second-derivative, the influence of noise is considerable. it always generates closed contours (doesn't intersect the edge of the map area), which is not realistic.



8. DIFFERENCE OF GAUSSIANS FILER (DOG)

↳ The DoG filter is obtd by taking the difference of two Gaussian functions.

↳ The expxn of a DoG filter is given by:

$$h(x,y) = h_1(x,y) - h_2(x,y)$$

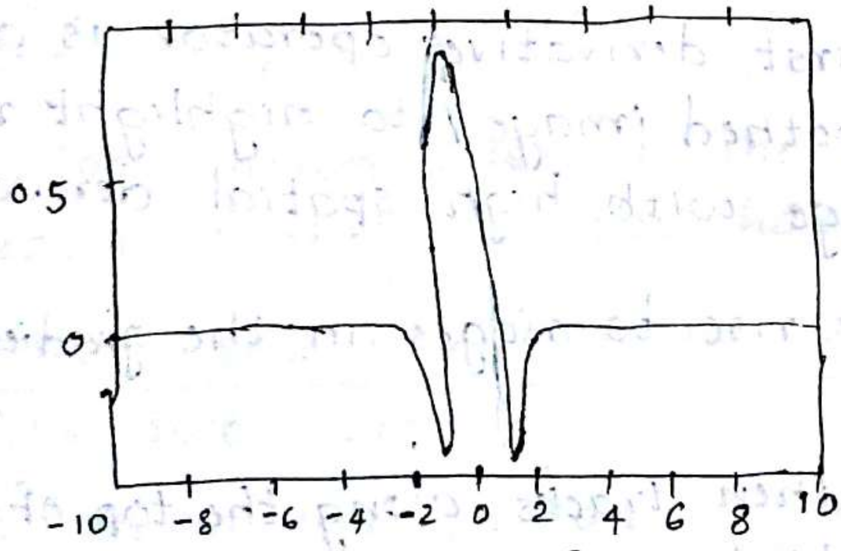
$h_1(x,y)$ & $h_2(x,y)$ are two Gaussian functions. which are given by:

$$h_1(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}}$$

$$h_2(x,y) = \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

$$h(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \quad \sigma_1 > \sigma_2.$$

↳



Shape of DoG filter:

↳ it is clear that the DoG filter function resembl a Mexican-hat wavelet. Therefore, a Mexican-hat wavelet is obtd by taking the difference of two Gaussian functions.

CANNY EDGE DETECTOR

↳ One pblm with a Laplacian zero crossing as an edge detector is that it simply adds the principal curvatures together.

ie, it doesn't really determine the maximum of the gradient magnitude.

↳ The Canny edge detector defines edges as zcs of second derivatives in the direction of the greatest first derivative.

↳ The canny operator works in a multi-stage process.

1. The image is smoothed by a Gaussian convolution.
2. A 2D first derivative operator is applied to the smoothed image to highlight regions of the image with high spatial derivatives.

↳ Edges give rise to ridges in the gradient magnitude image.

↳ The algm then tracks along the top of these ridges & sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the o/p, a process known as non-maximal suppression.

↳ The tracking process exhibits hysteresis controlled by two thresholds T_1 & T_2, with $T_1 > T_2$.

↳ Tracking can only begin at a pt on a ridge higher than T_1.

↳ Tracking then continues in both directions out from that pt until the hgt of the ridge falls below T_2.

↳ This hystersis helps to ensure that noisy edges are not broken into multiple edge fragments.

↳ The effectiveness of a Canny edge detector is determined by 3 parameters.

(i) width of the Gaussian kernel.

(ii) upper threshold.

(iii) lower threshold used by the tracker.

↳ Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise, at the expense of losing some of the finer details in the image.

↳ The localisation error in the detected edges also increases slightly as the Gaussian width increases

↳ Gaussian smoothing in the Canny edge detector fulfills two purposes

1. It can be used to control the amount of detail that appears in the edge image
2. used to suppress noise.

↳ The upper tracking threshold is usually set quite high and lower threshold value is set quite low for good results.

↳ ~~Setting the lower threshold is usually set quite high~~

↳ Setting the lower threshold too high will cause noisy edges to break up.

↳ Setting the upper threshold too low increases the no. of spurious and undesirable edge fragments appearing in the o/p.

Canny's approach is based on 3 basic objectives.

- Low error rate
- Edge points should be well localized.
- Single edge point response.

↳ Let $f(x,y)$ denote the i/p image & $G(x,y)$ denote the Gaussian function:

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

We form a smoothed image, $f_s(x,y)$, by convolving G & f.

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

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

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

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

↳ Let d_1, d_2, d_3 & d_4 denote the four basic edge dir^n for a 3x3 region;

We can formulate the following nonmaxima suppression scheme for a 3x3 region centered at every point (x,y) in $\alpha(x,y)$.

1. Find the direction d_K that is closest to $\alpha(x,y)$
2. If the value of $M(x,y)$ is less than atleast one of its two neighbors along d_K, let $g_N(x,y) = 0$ (suppression); otherwise, let $g_N(x,y) = M(x,y)$.

CANNY EDGE DETECTION ALGORITHM

1. Smooth the i/p image with a Gaussian Filter.
2. Compute the gradient magnitude & angle images
3. Apply nonmaxima suppression to the gradient magnitude image.
4. Use double thresholding and connectivity analysis to detect and link edges.