* Image segmentation -

→ segmentation is the process of Partitioning a digital image into multiple Reigons and extracting the meaningful region which is Known as Region of Interest.

Algorithm are based on

1.) Similarity Principle (Region Approach)

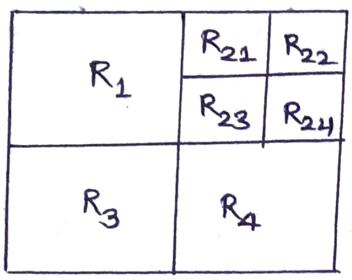
⇓

objective is to group Pixels based on Common Property to extract a Coherent Region.

2) Discontinuity Principle (Boundary Approach)

⇓

objective ~~is~~ is to extract regions that differ in properties like intensity, Color, texture etc.



Image

Let R represent the entire image region and segmentation is Partitioning R into n subgroups R_i

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

2) R_i should be Connected Region

3) $R_i \cap R_j = \phi$

4) $P(R_i) = \text{True}$

5) $P(R_i \cup R_j) = \text{False}$

$P(R_i)$: predicate that indicates some Property over the Region

Segmentation Algo

1) Based on User Interaction

- A) manual
- B) Automatic
- C) semi-Automatic

2) Based on Pixel Relationship

- A) Contextual (Region based or global)
- B) Non-Contextual (Pixel based or local)

* Detection of Discontinuities -

Discontinuities

1) Point 2) Line 3) Edge

1) Point :-

An isolated Point is a Point whose grey level is significantly different from its ~~bea~~ background in homogeneous Area.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image

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

if $|R| \geq T$: Point is detected

$T \geq 0$

$$\text{Mask} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

2) Line detection -

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

Horizontal   vertical

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

+45°   -45°

$$R_K = \sum_{K=1}^{4} W_K Z_K$$

if At certain Point in the image, $|R_i| > |R_j|$ for all $j \neq i$ ⇒ Line detected

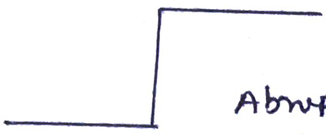
3) Edge detection -

→ An edge is a set of Connected pixels that lies on the boundry b/w 2 Regions Which differ in gray Value.

→ Locates sharp changes in the intensity function

→ Commonly Encountered Edges -

1) Step Edge -

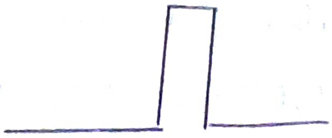


Abrupt change in intensity

2) Ramp Edge -

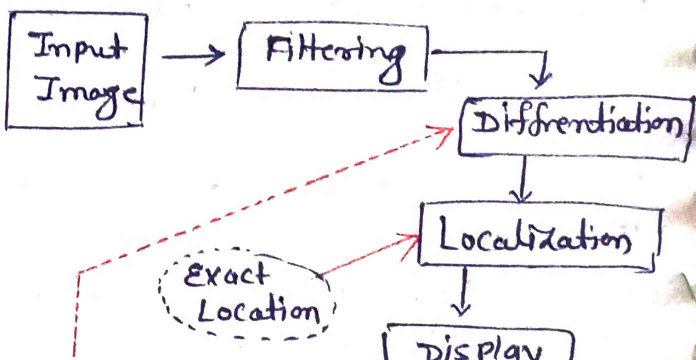
slow and gradual change

3) spike Edge -



4) Roof Edge -

• Steps -



distinguise the edge pixel from other pixel

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

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

$$\text{Direction} = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

(A)
* First order Edge detection operators -

different Image Intensity

↳ Constitute Edge

↓

measure Intensity

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

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

$$\approx |G_x| + |G_y|$$

$$\text{direction of gradient} = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

↳ An edge can be extracted by magnitude and direction.

- Backword difference = $f(x) - f(x-1)$

  ↳ mask = $[1 \quad -1]$

- forward diff = $f(x+1) - f(x)$

  ↳ mask = $[-1 \quad +1]$

- **Robert operator** -

  ↳ Cross - diagonal difference

  Known as - Cross gradient operators

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

- **Prewitt operator**

  ↳ takes the central difference of neighboring pixels.

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

$$\text{mask} = [-1 \quad 0 \quad +1]$$

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

$$G_x = \frac{\partial f}{\partial x} \qquad G_y = \frac{\partial f}{\partial y}$$

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

- **Sobel operator** -

  ↳ provides both a diffrentiating and smoothing effect

$$\underset{G_x}{\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}} \qquad \underset{G_y}{\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}}$$

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

* B) Template matching masks

→ Gradient masks are isotropic and insensitive to direction

→ These are direction sensitive.

• **Kirsch mask -**

↳ Known as Compass mask

→ These are obtained by taking one mask and rotating it to the 8 major directions.

Directions : East, N-E, N, N-W, W, S-W, S, S-E

$$\underset{m_0}{\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}} \quad \underset{m_1}{\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}} \quad \cdots\cdots \quad m_7$$

• **Robinson Compass mask -**

$$\underset{m_0}{\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}} \quad \underset{m_1}{\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}} \quad \cdots\cdots \quad m_7$$

• **Frei-Chen mask**

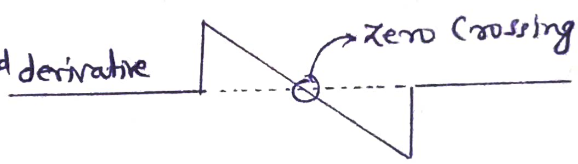
*c) Second order Derivative filters -

→ In case of second derivative, edge is present at that location where the second derivative is zero.

→ It is like zero crossing. Which can be observed as a sign change.

1^{st} derivative

2^{nd} derivative



• **Laplacian operator -**

→ very sensitive to Noise

→ Image must be filter first then edge detection process is applied

→ ⊕ : Rotationaly Invariant

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

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

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

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

for p&q : $|\nabla^2 f(p)| \leq |\nabla^2 f(q)|$

* **Canny Edge detection-**

→ detect wide range of images using multistage algorithm.

→ It is based on optimizing the trade off between two performance criteria and can be described as:

(i) Good Edge detection

(ii) Good Edge Localization

↳ should have the ability to produce edge points which are close to real edges points.

(iii) Good Edge Localization :

↳ should have the ability to produce edge points which are close to real edges.

• Algorithm-

Step-1 :- Smoothing Image and Computing Coefficient

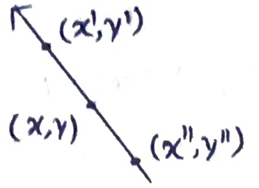
$$S = G_\sigma * I$$

$$G_\sigma = \frac{1}{\sqrt{2}\,\pi\sigma} e^{\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla S = \left[\frac{\partial}{\partial x}S \cdot \frac{\partial}{\partial y}S\right]^T$$

$$= (S_x \;\; S_y)^T$$

Step-2 : Non-maxima suppression



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

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

⇒ blurred Edges ⟶ Sharp edges.

Step-3 :- Apply Hysteresis Thresholding

→ if the gradient at a pixel is above 'High', declare it an 'edge pixel'.

→ if the gradient at a pixel is below 'Low', declare it a Non Edge pixel

→ if the gradient at a pixel is between Low and High then ~~deto~~ declare it an edge pixel, if and only if it is connected to an 'edge pixel' directly or via pixels between Low and High.

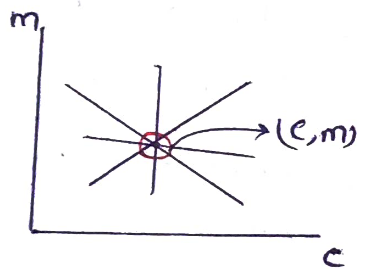
* Hough Transform -

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

→ Hough Transform takes the images created be edge detection operators but most of the time, edge map is dis connected. Therefore Hough Transform is used to connect the dis joined edge points.

- Equation of Line : $y = mx + c$

$$\Rightarrow c = -xm + y$$



Parameter space also called Hough Space.

Load the Image

↓

Determine the image edges using any edge detector

↓

Quantize the Parameter space, P

↓

Repeat the process for all pixels of Image, if the pixel is an edge pixel, then -

$$c = (-x)m + y$$

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

↓

Show the Hough Space

↓

find the local maxima in Parameter space

↓

Draw the line using local maxima.

- Limitation - does not Work for vertical lines, because they have infinity slope.

Therefore this line must be converted into Polar Co-ordinates.

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

* Corner detection –

- Corners having large variations in all directions, while edges have variation in only one direction.

→ Harris Corner Detection –

- It uses auto corelation function to find out the diffrential of corner score unit w.r.t direction directly.

A small square window is placed around each pixel
&
window is shifted in all 8 directions

⇓

Auto correlation is performed

if no variation : region is patch

if large variation : Pixel is marked as an corner point

Let Image : I
Patch : W of size m×m centered in (x_o, y_o)

We want to calculate the intensity variation occured if the window W is shifted by small amount (u,v), such variation can be estimated by sum of square differences (SSD)

$$SSD(u,v) = \sum_{(x,y)\in W} g(x,y)\left[I(x,y) - I(x+u, y+v)\right]^2$$

$g(x,y)$: window function

Using the first order Taylor Expansion

$$I(x+u, y+v) \approx I(x,y) + uI_x(x,y) + vI_y(x,y)$$

I_x : Partial derivative of I in x

I_y : Partial derivative of I in y

$$SSD(u,v) = \sum_{(x,y)\in W} g(x,y)\left[u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2\right]$$

$$\Rightarrow SSD(u,v) \approx [u \cdot v] \; m \begin{bmatrix} u \\ v \end{bmatrix}$$

$$m = \sum_{(x,y)\in W} g(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

m is called structure tensor

Eigen values of m characterize edge strength and eigen vectors represent the edge orientation.

Harris detector uses a Response fn that scores the presense of corners

$$R = det(m) - K\,tr(m)^2$$

$$K \in [0.04, 0.06]$$

$$\det(m) = \lambda_1 \lambda_2$$

$$tr(m) = \lambda_1 + \lambda_2$$

λ_1, λ_2 are Eigen Valves

$$R = \lambda_1 \lambda_2 - K(\lambda_1 + \lambda_2)^2$$

if λ_1 and λ_2 are Small $\Rightarrow$ $|R|$ is Small
the Reigon is flat

if $\lambda_1 >> \lambda_2$ or vice versa
then $R<0$ and Reigon is an edge

if $\lambda_1 \approx \lambda_2$ → and both are larger
then R is large and Reigon is a Corner

* Texture Analysis

(A) Local Binary Patterns (LBP):-

→ Popular technique used for Image/face representation and classification.

→ It describes the local features/structure of an Image, which is invariant to the changes in the illumination.

→ Computation of LBP

- Window of Image = (3x3)

$$LBP = \sum_{n=0}^{7} s(i_n - i_c)\, 2^n$$

i_n: Neighbor Pixel Valve
i_c: Center pixel Valve

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

i_0	i_1	i_2
i_7	i_c	i_3
i_6	i_5	i_4

(Assuming top-left as LSB)

Exa-

$$\begin{bmatrix} 5 & 12 & 3 \\ 6 & 2 & 4 \\ 8 & 3 & 7 \end{bmatrix}$$

$$s(i_0 - i_c) * 2^n = s(5-2)\, 2^0$$
$$= s(3)$$
$$= 1$$

$$s(i_1 - i_c) * 2^n = s(12-2) \cdot 2$$
$$= s(10) \cdot 2$$
$$= 2$$

$$s(i_2 - i_c) * 2^2 = s(3-2) \cdot 2^2$$
$$= 4$$

$$s(i_3 - i_c) * 2^3 = s(4-2) \cdot 8 = 8$$

$\vdots$

$$LBP = \sum s(i_n - i_c)\, 2^n$$
$$= 1+2+4+8+16+32+64+128$$
$$= \underline{255}$$

Binary Pattern

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

* SIFT Detector-

Scale Invariant Feature Transform

↳ used for Image Alignment and 2D object ~~Reo~~ Recognition.

- SIFT finds Key Points (Interesting Points) in an Image and generate descriptors that uniquely identify those points.

Steps:-

(A) Scale-space Construction

Apply Gaussian blur with different σ