Digital Image Processing

Ch 9 Image Segmentation

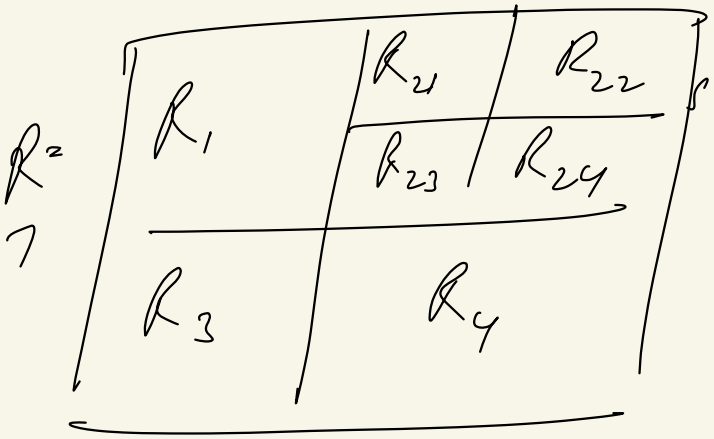
- Segmentation is the process of partitioning a digital image into multiple regions & extracting the meaningful region which is known as Region of interest
- RoI may vary with applications
- In fact no single universal segmentation algorithms exist for segmenting the RoI in all images
- Therefore many segmentation algos. need to apply & pick that algo which performs the best for given requirement

Image Segmentation Algos are based on:

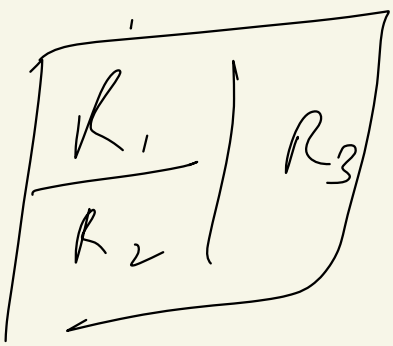
- Similarity Principle (Region Approach)
  - Objective is to group pixels based on common property to extract coherent regions
- Discontinuity Principle (Boundary Approach)
  - Objective is to extract regions that differ in properties like intensity, color, texture etc.

Defn of Image Segmentation

An image can be partioned into many regions $R_1, R_2, R_3 \ldots R_n$



Eg :



Characteristics of Segmentation Process

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

→ $\bigcup_{i=1}^{n} R_i = R$

→ R_i should be connected region :

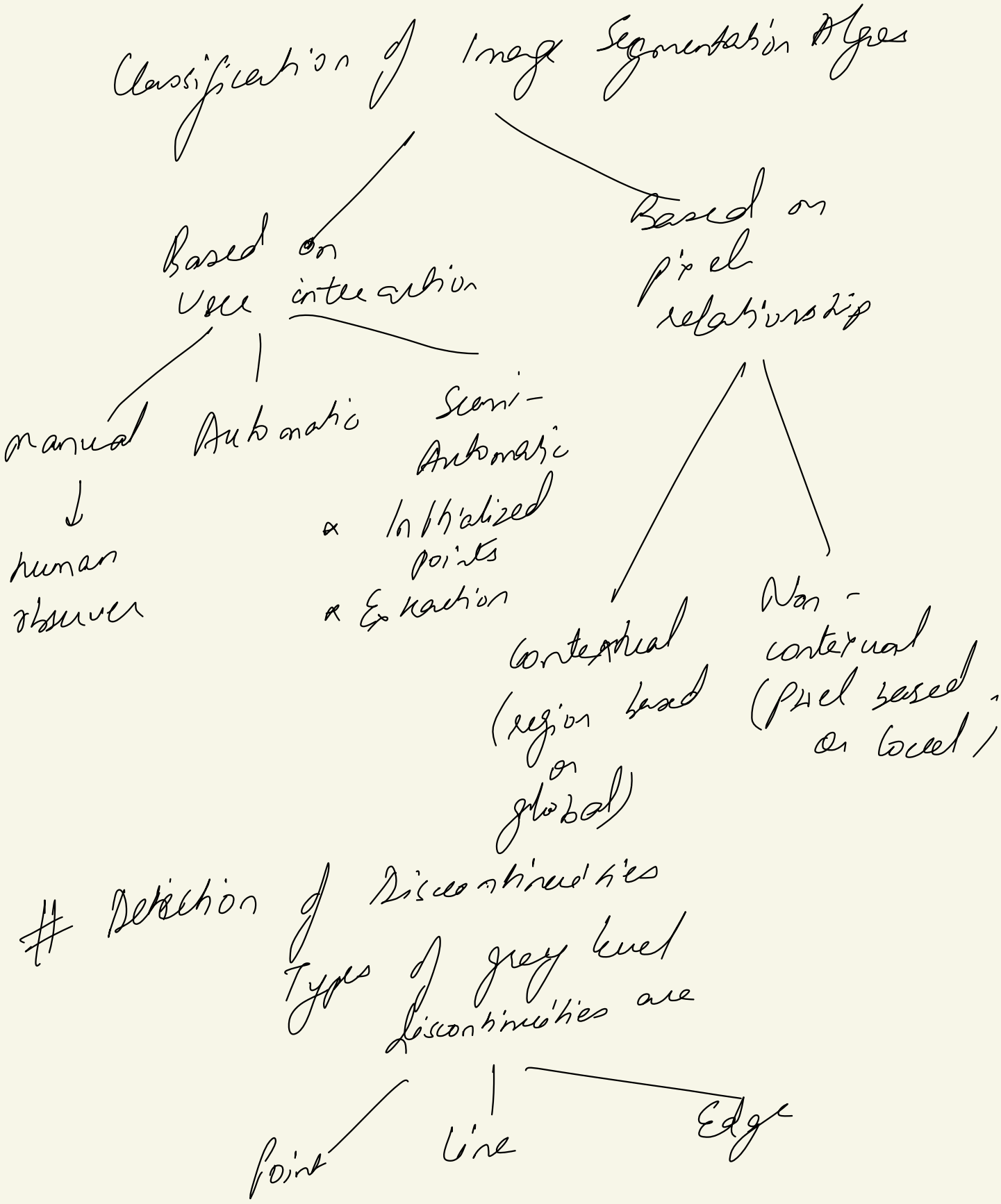
$i = 1, 2, 3 \ldots n$

→ $R_i \cap R_j = \phi \quad \forall\, i \& j : i \neq j$ (disjoint)

→ $P(R_i)$ = TRUE for $i = 1, 2, 3 \ldots n$

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

Here $P(R_i)$ is a predicate that indicates some property over the region


Classification of Image Segmentation Algos
Based on User interaction
Based on pixel relationship
manual
human observer
Automatic
Semi-Automatic
Initialized points
Extraction
Contextual (region based or global)
Non-contextual (Pixel based or local)
Detection of Discontinuities
Types of grey level discontinuities are
Point
Line
Edge

Point Detection

- An isolated point is a point whose grey level is segnificantly different from its background in a 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

Response of mask

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

if $|R| \geq T$ a point is detected

where T is a non negative integer

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

Sample mask for Point Detection

Line detection

4 types of masks used to get responses ie R_1, R_2, R_3 & R_4 for the directions vertical, horizontal, $+45$, -45 respectively

$$\begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix}$$

horizontal

$$\begin{matrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{matrix}$$

Vertical

$$\begin{matrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{matrix}$$

$+45^\circ$

$$\begin{matrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{matrix}$$

-45°

Response of mask

$$R_k = \sum_{k=1}^{4} w_k z_k$$

if at a certain point in the image, $|R_i| > |R_j|$ for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i

Edge Detection

- An edge is a set of connected pixels that lie on the boundary b/w 2 regions which differ in gray value. Pixels on edge is known as edge points
- Edges provide an outline of the object
- In physical plane, edge corresponds to the discontinuities in depth, surface orientation, change in material properties, light variations etc.
- It locates sharp changes in the intensity function
- Edges are pixels where brightness changes abruptly
- An edge can be extracted by computing - the derivative of the image function

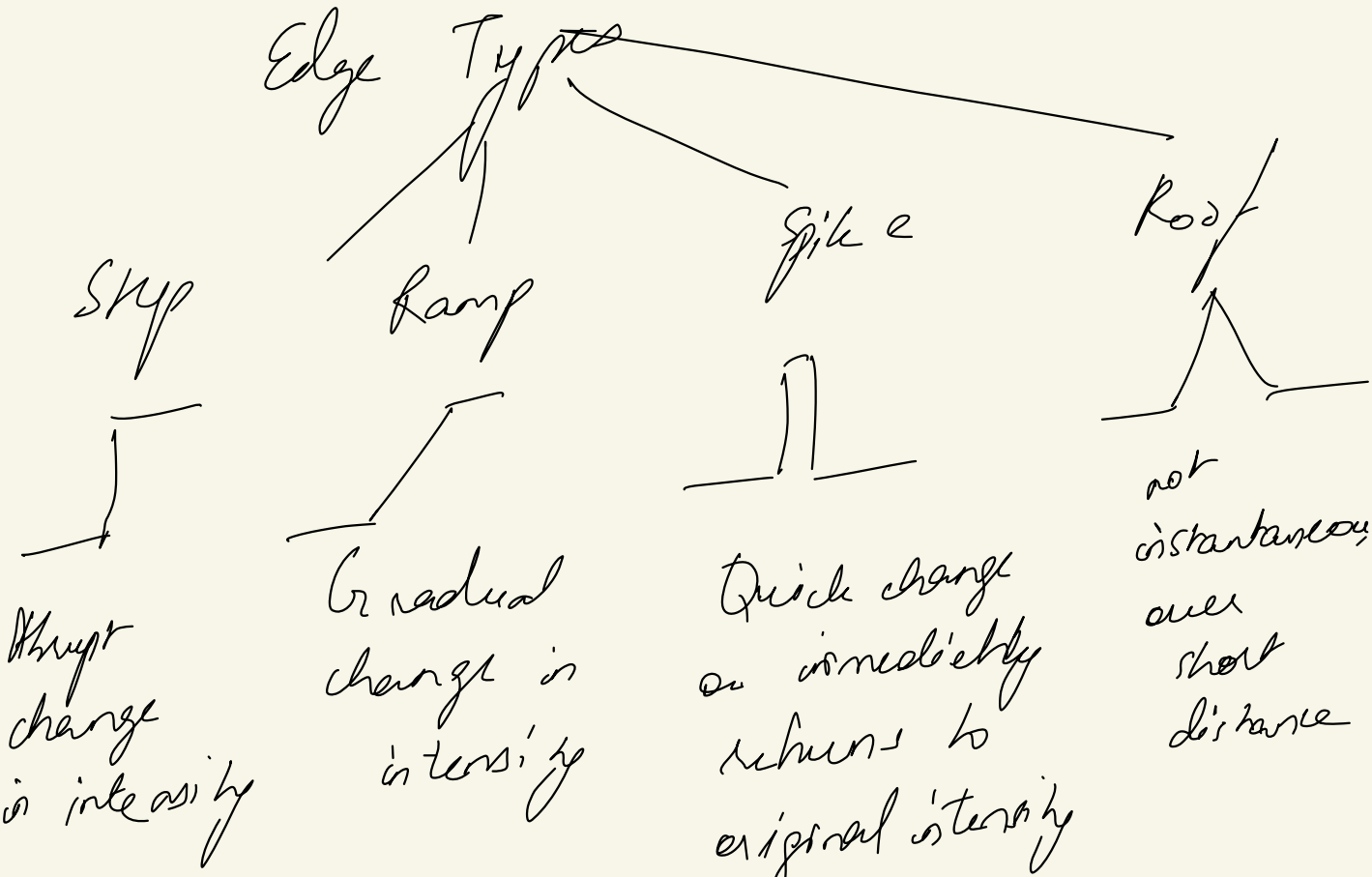
1) Magnitude of the derivative indicating strength/contrast of edge

Edge pixels = Edgels

Magnitude & Orientation

4) Direction of derivative vector, indicating edge orientation

Edge Types



Step: Abrupt change in intensity

Ramp: Gradual change in intensity

Spike: Quick change or immediately returns to original intensity level

Roof: not instantaneous over short distance

Stages in edge detection

Start → Input image → Filtering → Differentiation → localization → Display → End

Filtering: It involves smoothing. This stage may be performed implicitly/explicitly

It distinguishes the edge pixels from others

↓

Differentiation

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

$\text{Magnitude}(\nabla f) = \sqrt{G_x^2 + G_y^2}$

$\approx |G_x| + |G_y|$

Direction of gradient $= \tan^{-1} \frac{G_y}{G_x}$

localization ⇒ determine the exact location of edge

Edge Detection Algos

- Derivative Types: It uses differentiation technique for edge detection
- Template Matching - It uses templates that resembles the target shape and match with image

- Gaussian Derivatives : Very effective for real time images
- Pattern fit Approach : Here surface is considered as topographic surface with pixel value representing altitude

First order Edge detection Operators (Image Segmentation)

- Local transitions among different image intensities constitute an edge
- Therefore Objective is to measure the intensity gradient
- Edge detectors can be viewed as gradient calculators

Gradient operator represented as

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

$\text{Magnitude}(\nabla f) = \sqrt{G_x^2 + G_y^2}$

$\approx |G_x| + |G_y|$

Direction gradient $= \tan^{-1} G_y / G_x$

- Backward difference $\begin{pmatrix} 1 & -1 \end{pmatrix}$

$= f_x - f(x - \Delta x) / \Delta x$

$x - \Delta x \quad x$

- Forward difference $\begin{pmatrix} -1 & +1 \end{pmatrix}$

$f(x + \Delta x) - f(x) / \Delta x$

- Central difference

$$\frac{f(x + \Delta x) - f(x - \Delta x)}{2 \Delta x}$$

Robert Operator

Robert kernels are derivatives cost diagonal elements

- Cross gradient operators / Cross diagonal differences

Let $f(x,y)$ & $f(x+1,y)$

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

Robert masks

$$G_x = \begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix} \qquad G_y = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$$

1) Read image & smooth it
2) Convolve f with g_x & g_y
3) Compute edge magnitude & direction $\sqrt{G_x^2 + G_y^2}$ $\tan^{-1}\frac{G_y}{G_x}$
4) If mag > threshold
   then it is edge point

Prewitt operator

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

Central difference mask $[-1 \quad 0 \quad +1]$

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

3X3 approximation of Prewitt operator

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

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

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

Sobel operator

- It provides both differentiating & a smoothing effect
- approximation of first Gaussian derivative
- Convolution is both commutative & associative

$$G_x = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \qquad G_y = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

$$G_x = \frac{\partial f}{\partial x} = (z_2 + 2z_4 + z_9) - (z_1 + 2z_2 + z_3)$$

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

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

Another mask for diagonal edges
& Sobel

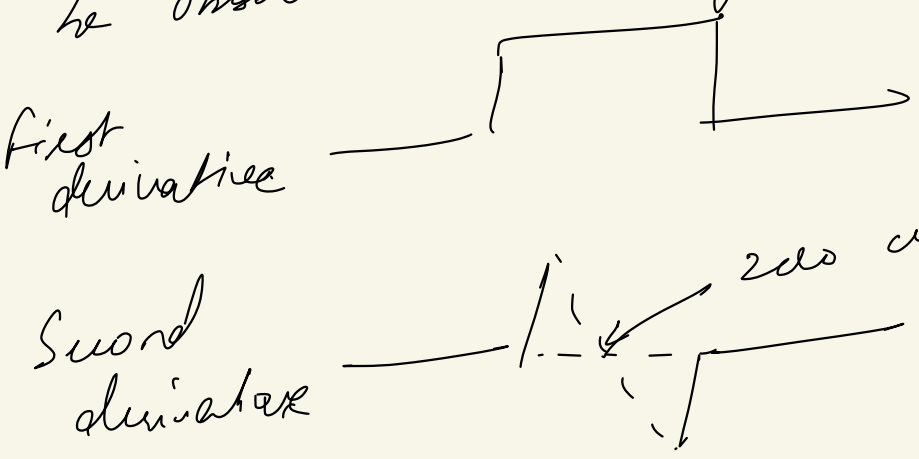
G_x

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

G_y

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

Second order Derivative filters
- In first derivative, edges are considered to be present if magnitude large than threshold value
- In case of second derivative, edge is present at location where second derivative is zero
- It is like zero crossing, which can be observed as a sign change



Laplacian operator

- Zero crossing algo
- Laplacian masks are very sensitive to noise because there is no magnitude checking, even a small ripple looks like edge

Therefore, image must be filter first & then edge detection process is applied

- Advantage - rotationally invariant

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

$$f(x+1,y) - 2f(x,y) + f(x-1,y)$$

$$f(x,y+1) - 2f(x,y) + f(x,y-1)$$

1) Generate mask
2) Apply mask
3) Detect zero crossing - is a situation where pixels in neighborhood differ from each other in sign

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \qquad \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

Different laplacian masks

Difference of Gaussian Operator

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

Gaussian width σ_1

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

$$\underbrace{\left(G_{\sigma_1}(x,y) - G_{\sigma_2}(x,y)\right)}_{DoG} * f(x,y)$$

Canny Edge Detection

- It uses a multistage algo to detect wide range of edges in images
- John F. Canny in 1986

Canny approach is based on finding the trade off b/w 2 performance criteria :

- Good edge detection : should be capable to detect only real edge points & discard all false edge points
- Good edge localization : should have ability to produce edge points which are close to real edges
- Only one response to each edge : should not produce any false, double or spurious edges

Step 1

1) Convolve image with Gaussian filter
2) Compute gradients of resultant smooth image
3) Store edge magnitude & orientations in 2 arrays

Smoothing image & computing coefficients

$$S = G_\sigma * I$$

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

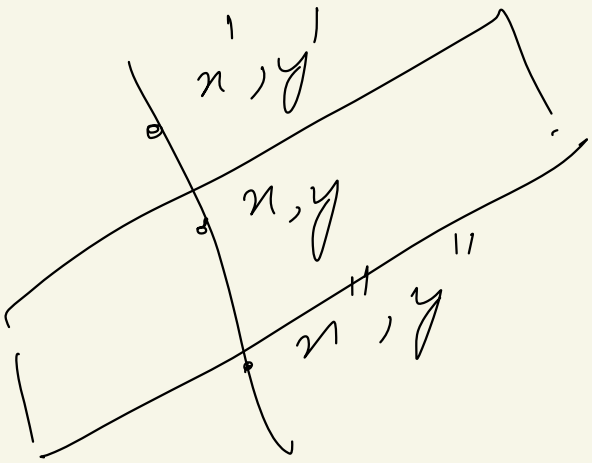
$$\nabla S = \begin{bmatrix} \frac{\partial S}{\partial x} & \frac{\partial S}{\partial y} \end{bmatrix}^T$$

$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

$$\theta = \tan^{-1} \frac{S_y}{S_x}$$

Step 2

Thinning edges by non maxima suppression

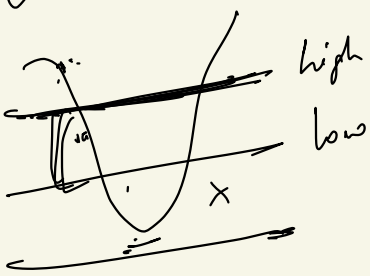


x', y' & x'', y'' are neighbors of x, y in $|\nabla S|$ along the direction normal to an edge

spurious ≈ false ≈ irrelevant

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

Step 3 Apply hysteresis thresholding



- if gradient at a pixel is above 'high', declare it as an edge pixel
- if gradient is below low → non edge pixels
- if b/w low & high → edge pixel if it is connected by another edge pixel

Hough Transform

- Feature extraction method for detecting shapes such as circles, lines etc
- Hough transform takes the images created by edge detection operators might be disconnected so Hough transform is used

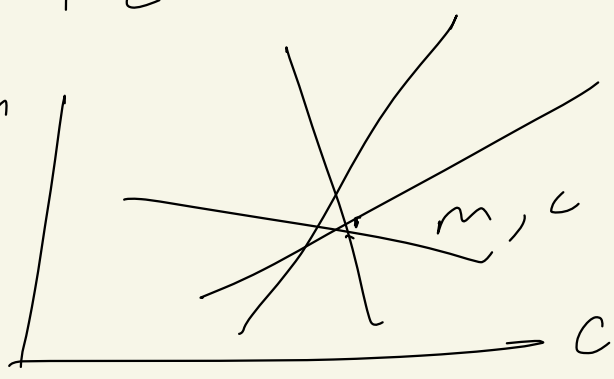
Eq: $y = mx + c$

Problem: infinite lines can be drawn connecting these points

Therefore an edge point in the x-y plane is transformed to the c-m plane

x_i, y_i

$y_i = m x_i + c$ or $c = -x_i m + y_i$

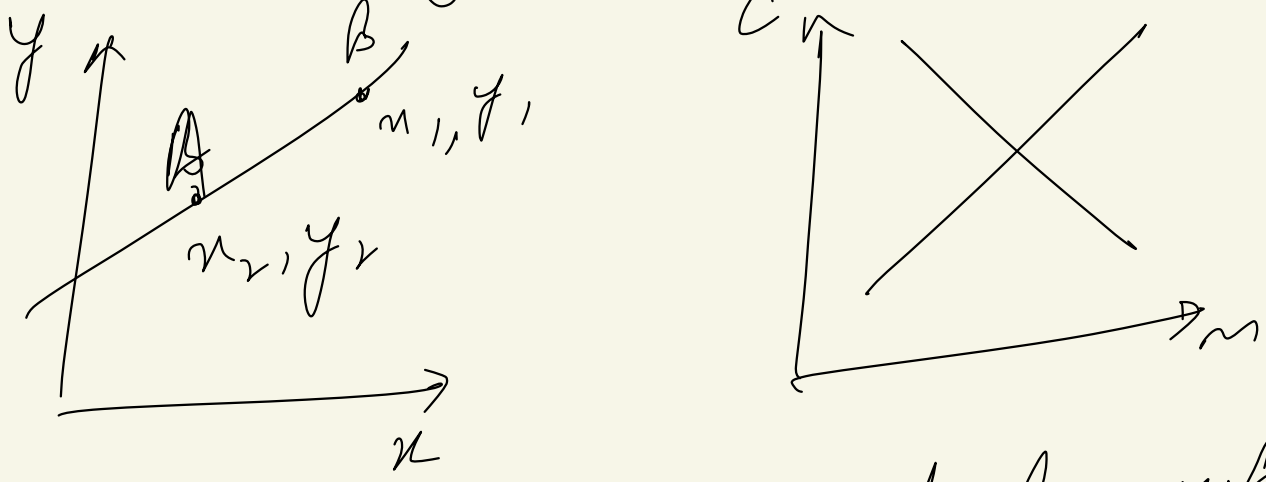


Parameter space / Hough space

If A & B are 2 points connected by a line in spatial domain

↓

They will be intersecting lines in hough space



Limitation: It doesn't work for vertical lines as they have infinity slope

→ This line must be converted into polar coordinate

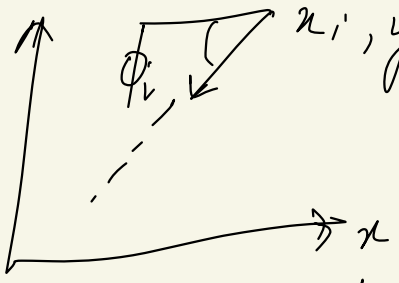
Eqn in polar: $\rho = x\cos\theta + y\sin\theta$

Circle:

$(x_i - a)^2 + (y_i - b)^2 = r^2$

Edge location: x_i, y_i

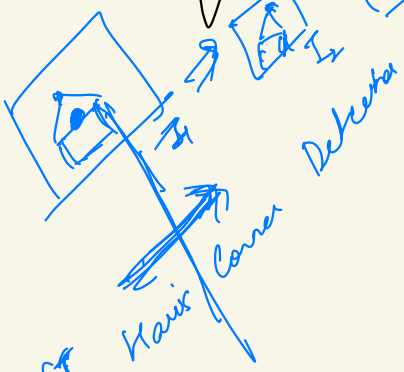
Edge direction: ϕ_i



$a = x - r\cos\phi$

$b = y - r\sin\phi$

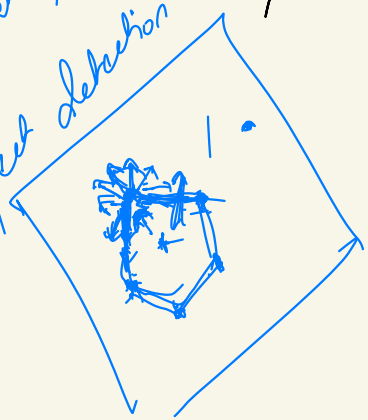
1) Load image
2) Determine image edges using any edge detector
3) Quantize parameter space A
4) Repeat process for all pixels of image if pixel is an edge pixel then $a = x - r\cos\phi$, $b = y - r\sin\phi$, $A(a,b,r) = A(a,b,r) + 1$
5) Show hough space
6) Find local maxima in parameter space & draw circle using it.



SIFT, Harris Corner Detector, Separability, distinctiveness, detectability, Corner score

keypoint, Corner, (image feature)

good VP



Corner Detection

- Corner is very imp feature of an image, it indicates some imp points of image - Landmark points / interest point
- Corner is formed at the boundary when 2 bright regions met where boundary curvature is very high
- Corner point exhibits the characteristics of having large variations in all directions while edges have variation in only 1 direction.

Moravec Corner Detector

- It is one of the earliest corner detection algo
- It calculates edge strength in all 4 directions
- Minimum of 4 directional responses is selected & threshold process is applied
- Then non-maximal suppression is applied to extract a set of corner points

* Moravec corner detection is not rotationally invariant & it is sensitive to even small variations along the edge

Harris corner detection

- It is very popular corner detection algo.
- It uses auto correlation fn. to find out the differential of corner score with direction directly
- Advantage is corner detection irrespective of different sampling quantization, small changes in scales & affine transformations

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

↓

Auto correlation is performed

* If no variation - Region is a patch
* If large variation - Pixel is marked as an interest point

1) If original point is x, y & shift of window is Δx & Δy

4 Auto correlation fn. or corner score

$$C(x,y) = \sum_{w} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2$$

(I: image; I: shifted image)

Shifted image can be approximated by Taylor expansion as

$$I(x_i + \Delta x, y_i + \Delta y) \cong I(x_i, y_i) + [I_x(x_i, y_i), I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

where $I_x(x,y) = \frac{\partial I}{\partial x}(x,y)$ & $I_y(x,y) = \frac{\partial I}{\partial y}(x,y)$

$$C(x,y) = \sum_{w} [I(x_i, y_i) - [I(x_i, y_i) + I(x_i + \Delta x, y_i + \Delta y)]]^2$$

Any quadration equation of form
$F(x,y) = Ax^2 + 2Bx + Cy^2$ can be
expressed as in matrix form:
$F = u M u^T$ where $u = [x, y]$ &
$M = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$

Autocorrelation fn. $C(x,y)$ is given as:
$$C(x,y) = (\delta x \ \delta y) \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$$

where
$$A(x,y) = \sum_w I_x^2(x,y)$$
$$C(x,y) = \sum_w I_y^2(x,y)$$
$$B(x,y) = \sum_w I_x(x,y) I_y(x,y)$$

Autocorrelation fn. "C" captures
intensity structure of the neighborhood
& it is smoothed by linear
Gaussian filter

Gaussian smoothed matrix R is represented by: $R = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$

- Eigen values of this matrix characterize edge strength & Eigen vectors represent edge orientation
- Let us assume λ_1, & λ_2 ⇒ Eigen values
  - If λ_1 & λ_2 are too small
    ↳ window is flat
  - If λ_1 is high & λ_2 is small
    ↳ it is edge
  - If λ_1 & λ_2 are high - corner is present

Corner strength is given as

$$Q(u,v) = \det(R) - k(\text{trace}(R))^2$$

$$\det(R) = \prod_{i=1}^{n} \lambda_i$$

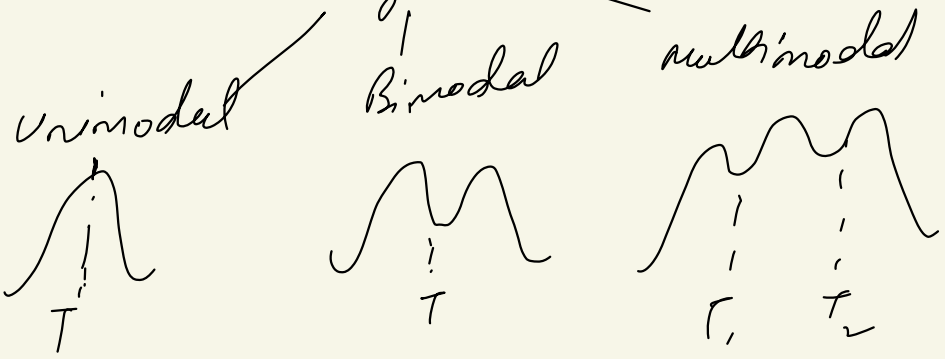
$$\text{trace}(R) = \sum_{i=1}^{n} \lambda_i$$

1) Select an image location (x, y)
2) Identify it as a corner pt. where corner strength
   $Q(u, v) >$ threshold
   Define a window & apply Autocorrelation fn.
3) Insert into a list of corner pts. sorted by corner strength
4) Eliminate false corner when a weaker corner pt lies near a stronger corner
5) Display all corner pts.

Thresholding
- Global
- Local
- Adaptive

- Tool for finding threshold - histogram

Histogram



Suppose $f(x,y)$ ⇒ composed of light objects on dark background

Single level thresholding:

any pt. (x,y) : if $f(x,y) > T \Rightarrow$ object FG

$f(x,y) < T \Rightarrow BG$

Multi level thresholding: $f(x,y) < T_1 \Rightarrow$ 1 object

$T_1 < f(x,y) < T_2 \rightarrow$ 2 object

$f(x,y) > T_2 \rightarrow$ other object class

Adaptive threshold

↳ Divide image into many overlapping sub images

↳ Split image into many sub regions

* Mean + C
* Median + C
* (Min + Max)/2

C ⇒ constant

Region Growing

- Procedure to group pixels or subregions into larger regions using similarity

homogeneity of regions is used as criterion of region growing

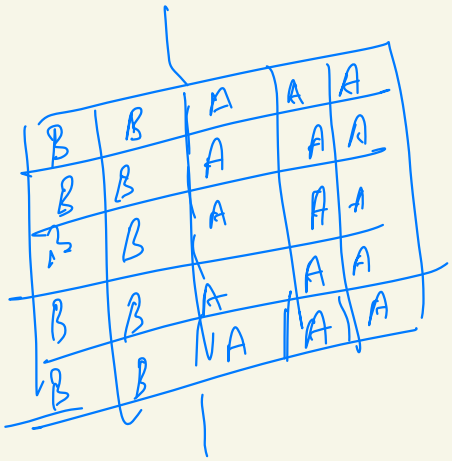
- gray level
- color
- texture
- shape
- model etc

Steps: 1) Selection of initial seed
2) Seed growing criteria
3) Termination of segmentation process

- Group pixels into larger regions
- Start with seed region
- Grow region by merging neighboring pixels

Eg:

1	0	7	8	7
0	1	8	9	8
0	0	7	9	8
0	1	8	8	9
1	2	8	8	9



Seed points → S_1 S_2

9 1

Threshold T $T \leq 4$

Federer's

* $S_1 = 9$

$|f(x,y) - f(x',y')| \leq 4$

$|f(x,y) - 9| \leq 4$

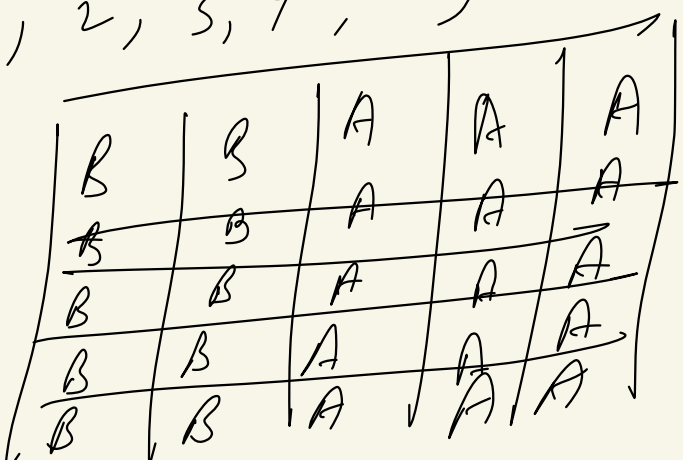
↳ $\{5, 6, 7, 8, 9\}$ → A

* $S_2 = 1$

$|f(x,y) - f(x',y')| \leq 4$

$|f(x,y) - 1| \leq 4$

↳ $\{0, 1, 2, 3, 4, 5\}$ → B



Split & Merge Algo.

- Alternative method of image segmentation
- Image is subdivided into arbitary disjoined region
- Arbitary regions can be merged in order to satisfy condition

Algo has 2 phases

Phase I

- Split & continue subdivision process till some stopping criteria is fulfilled
- Stopped when no splitting possible further

Phase II

- merge adjacent regions if the regions share any common criteria