4/2/25/ Tuesday.

Computer Vision:

Computer vision is a field of artificial intelligence (AI) that allows computers to see and understand the visual world.

Geometric Primitives:

2D Point:

- How to project a point from one space to another.
- Euclidean Space $\longrightarrow$ Projective space.

Image, $I = (x, y)$ → inhomogeneous $\qquad \tilde{I} = (\tilde{x}, \tilde{y}, \tilde{w})$ → homogeneous.

- 2D points represented using homogenous coordinates.

$$\tilde{X} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{X}$$

$\tilde{w}$ → scaling.

$(x, y, 1)$ → form of augmented matrix

Why do we use augmented matrix representation?

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & & \vdots \end{bmatrix}$$

$(x, y, 1)$ → we are trying to capture an image at point (x, y) with the distance b/w camera & img being 1 unit. If it is at 3 units, the whole thing is ~~ex~~ scaled by 3 ($\tilde{w} = 3$).
→ scaling.

classmate
Date ____
Page ____

2D Lines:

$\tilde{l} = (a, b, c)$

corresponding line equation:

$$\bar{x} \cdot \tilde{l} = ax + by + c$$

Intersection of two lines:

$$\tilde{x} = \tilde{l}_1 \times \tilde{l}_2$$

Line joining two points:

$$\tilde{l} = \tilde{x}_1 \times \tilde{x}_2$$

Cross product is used in both cases because we need vector.

★ $\tilde{a} \times \tilde{b} = [a] \times b$ → ~~to~~ the vector we get is the line intersecting the lines $\tilde{a}$ & $\tilde{b}$.

$\tilde{a}$ ↓ matrix, $\tilde{b}$ ↓ vector; $[a]$ ↓ skew symmetric matrix. b ↓ vector

Q. How to make ~~vector~~ line $\tilde{l}$ to skew symmetric matrix?

↳ $\tilde{l} = (x, y, z)$

$$[\tilde{l}] = \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix}$$

5/2/25 | Wednesday.

eg: $l_1 \rightarrow y=1$

$\hookrightarrow (0 \quad 1 \quad -1)$

$\downarrow$

$0 \times x + 1 \times y + -1 = 0$

$\Rightarrow y - 1 = 0$

$\underline{y = 1}$

$l_2 \rightarrow x = 2$

$\hookrightarrow (\cancel{2}\, 1 \quad 0 \quad -2)$

$\downarrow$

$1 \times x + 0 \times y + -2 = 0$

$x - 2 = 0$

$\underline{x = 2}$

$l_3 \rightarrow$ line intersecting l_1 & l_2 ?

$$\tilde{l}_1 \times \tilde{l}_2 = [\tilde{l}_1]_\times \tilde{l}_2$$

skew symmetric matrix form of l_1

$$= \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$$

$$= \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

$\rightarrow$ The point of intersection is $(2, 1)$

augmented part.

Q. What do we get from getting the ⊥cular vector to both the lines?

↳ We get the direction of intersection of both the lines. This is why we are doing cross product.

2D Transformations :

1. Translation (2 DoF)

↳ t_x + t_y.

- shifting in x-y plane.
- $x' = x + t$

OR

$$x' = [I \quad t]\, \bar{x}$$

$$\bar{x}' = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

I -> identity matrix

-> $(3\times3) \times (3\times1) = (3\times1)$

(3×3) ↓ transformation matrix; (3×1) ↓ original point $\bar{x}$; (3×1) ↓ transformed point $\bar{x}'$

Translation matrix

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

2. Euclidean: (3 DoF)

- Combining rotation (1 DoF) + translation (2 DoF) = 3 DoF
- $x' = Rx + t$

OR

$$x' = [R \quad t]\, \bar{x}$$

R -> orthonormal 2x2 rotation matrix.

$$\bar{x}' = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$|R| = 1$

$R^T R = R R^T = I$

↳ orthonormal.

$$\underset{3\times1}{\bar{x}'} = \underset{3\times3}{\begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}} \underset{3\times1}{\bar{x}}$$

3. Similarity (4 DoF)

2 DoF + 1 DoF + 1 DoF = 4 DoF

- Translation & then scaled rotation
- $x' = sRx + t$ ↳ we are doing uniform scaling so 1 DoF. If it were non-uniform then 2 DoF.

OR

$$x' = [sR \quad t]\bar{x}$$

$$\bar{x}' = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

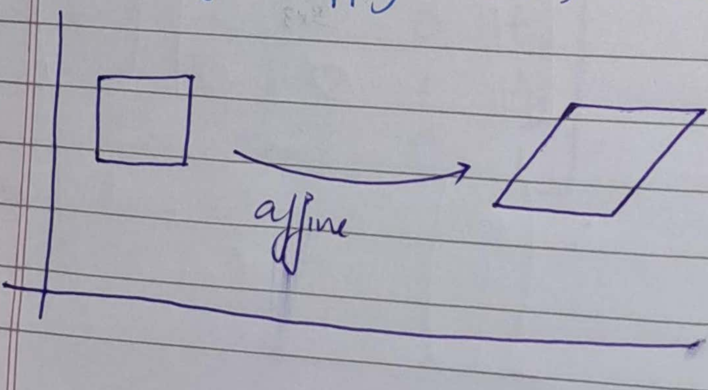
s - arbitrary scale factor.

4. Affine transformation (6 DoF)

- $x' = Ax$
- Affinity matrix → has six elements → 6 DoF

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix}$$

- When you apply shear, it causes affine transformation



- Parallelism is maintained. Orientation may be changed but lines which were parallel ~~f~~ before would remain parallel after the transformation.

- $$\bar{x}' = \cancel{A} \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{x}$$

5. Projective transformation / Homography (8 DoF)

- operates on homogeneous coordinates.

- $\tilde{x}' = \tilde{H} \tilde{x}$

  ↳ arbitrary 3x3 homogeneous matrix. (The last element (h_{33}) is 1 so the DoF is 8 ~~3~~ & not 9).

but lines which were ...
parallel after the transformation.

- $$\bar{x}' = \cancel{A}\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{x}$$

5. Projective transformation / Homography (8 DoF)

- operates on homogeneous coordinates.

- $$\tilde{x}' = \tilde{H}\,\tilde{x}$$
  ↳ arbitrary 3x3 homogeneous matrix. (the last element (h_{33}) is 1 so the DoF is 8 ~~&~~ not 9).

12/02/25 / Wednesday.

3D Transformations

Transformation	Matrix	DoF	Preserves	Icon
translation	$[I\ t]_{3\times4}$	3	orientation	
Euclidean	$[R\ t]_{3\times4}$	6	lengths	
Similarity	$[sR \cdot t]_{3\times4}$	7	angles	
Affine	$[A]_{3\times4}$	12	parallelism	
Projective	$[\tilde{H}]_{4\times4}$	15	straight lines.	

Photometric Image Formation

Sensor Functions

1. Photoelectric conversion
2. Charge accumulation
3. Transfer Signal
4. Signal Detection
5. Analog to Digital conversion

Functions done in order.

Lens → Image Sensor → ISP → Interface I/O → Image data

$\underbrace{\text{Lens} \rightarrow \text{Image Sensor} \rightarrow \text{ISP} \rightarrow \text{Interface I/O} \rightarrow \text{Image data}}_{\text{CMOS Camera}}$

Q. How to convert analog to digital?
1. Sampling
2. Quantization
3. Encoding

1. Sampling:

~~Digitalizing~~
Digitizing along (x, y) coordinates
or
Making grids.

2. Quantization:

Digitizing along the amplitude
↳ intensity of the image.

Q. How many greyscale levels?
↳ 8 bit → 2^8 grayscale level quantization.
excluding white & black there are 254 grey scale levels.
white ↳ 255, black ↳ 0

2. Quantization:

Digitizing along the amplitude.
↳ intensity of the image.

Q. How many greyscale levels?
↳ 8 bit → 2^8 grayscale level quantization.
excluding white & black there are 254 grey scale levels.
white ↓ 255, black ↳ 0

17/2/25 | Mon

1. Camera Calibration:

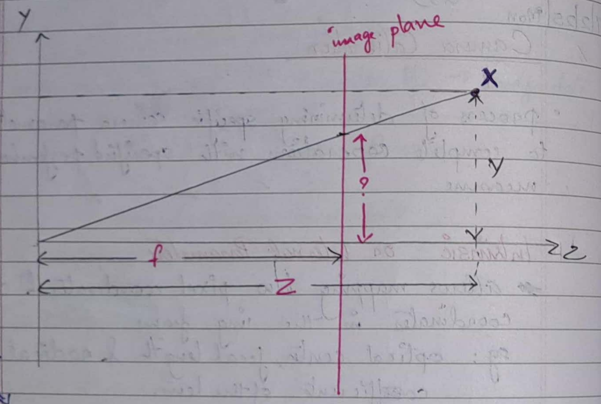
- process of determining specific camera parameters in order to complete operations with specified performance measure.

1. Intrinsic or Internal Parameters
   - allows mapping b/w pixel coordinates & camera coordinates in the img frame.
   eg: optical center, focal length & radical distortion coefficients of the lens.

2. Extrinsic or External parameters
describes orientation & location of the camera.
Refers to rotation & translation of camera

Pinhole camera: • black box white screen & a pin hole.
• get inverted image.
• projecting object (3D) to a 2D plane.
• Thus we get a ~~a~~ transformed 2D vector.

focal length: distance from camera center to the image plane.



$$[X \quad Y \quad Z]^T = \left[f\cdot\frac{X}{Z} \quad f\cdot\frac{Y}{Z}\right]^T$$

$$\frac{f}{z} = \frac{y}{\cancel{Y}} \qquad \cancel{\frac{y}{f}} = \frac{Y}{Z}$$

• The division by Z captures perspective projection, meaning objects further from the camera appear smaller.

Camera Matrix:

A camera is a mapping b/w the 3D world & a 2D image.

$$\underbrace{x}_{\text{2D image point}} = \underbrace{P}_{\text{camera matrix}} \underbrace{X}_{\text{3D world point}}$$

$$\underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\text{homogeneous matrix}} = \underbrace{\begin{bmatrix} P_1 & P_2 & P_3 & P_4 \\ P_5 & P_6 & P_7 & P_8 \\ P_9 & P_{10} & P_{11} & P_{12} \end{bmatrix}_{3\times4}}_{\text{camera matrix}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\text{homogeneous world point}}$$

Camera Matrix decomposed into 2 matrices

principal points coordinates — External calibration matrix

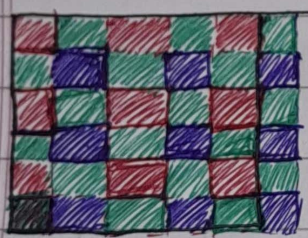
$$P = \underbrace{\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}_{3\times3}}_{\text{Internal calibration matrix}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{3\times4} \Rightarrow P = K[I|0]$$

☆ The camera is already aligned with the real world coordinates hence why the external calibration matrix is always $[I;0]$.
If it is not aligned, we can apply rotation and translation to align it.

☆ The camera is already aligned with the real world coordinates hence why the external calibration matrix is always [I¦0].
If it is not aligned, we can apply rotation & translation to align it.

18/02/25/Tuesday.

- A digital camera separates R, G & B components using Bayer filter placed over the image sensor. Each pixel only captures one color (R, G or B).



→ Bayer filter

But you notice that some information is lost. So what to do?

↓

Interpolation: used to estimate unknown values b/w ~~data~~ known data points.

Types:

1. Nearest neighbour interpolation
2. Bilinear interpolation
3. Bicubic interpolation

etc.

classmate

Date ______

Page ______

1. Nearest Neighbour Interpolation: (NNI).

-

3	4
5	6

2×2 → how to scale it by a factor of 2 using interpolation?

Step 1: Write down the corner points. We have 4 known values. Now we find remaining using NNI.

Step 2: Now to a point consider the nearest point.

↓

3 nearest 3	3	4	nearest 4
3	3	4	4
5	5	6	6
5	5	6	6

4×4

- Simplest method.

$\frac{11}{3} \times \frac{1}{3} +$

2. Bilinear Interpolation:

1 → distance.

3	4
5	6

2×2

Task is the same. Again we know only the corner points.

1 → assume dist remains same even after scaling.

↓

3	3.33	3.66	4
3.66	4	4.32	4.66
3.32	4.64	4.9[illegible]	5.33
5	5.32	5.66	6

4×4 (spacing $\frac{1}{3}$ between rows and columns)

Step 1: Find distance between known pixels in original matrix. Assume same dist in new matrix.

Step 2: Find dist b/w known & unknown pixels.

Step 3: Multiply & Add. (Consider either vertically or horizontally)

$$\frac{10}{3} \times \frac{2}{3} + \frac{16}{3} \times \frac{1}{3}$$

$$= \frac{20}{9} + \frac{16}{9} = \frac{36}{9} = 4$$

$$3 \times (1 - \tfrac{1}{3}) + 4 \times (1 - \tfrac{2}{3})$$

$$= 3 \times \frac{2}{3} + 4 \times \frac{1}{3} = 2 + 1.33$$

$$= 3.33$$

24|2|25 | Monday.

~~25|2~~ ~~19|2|25 | Wednesday~~

3. Bicubic interpolation:

- Instead of 4 total points (as in bilinear), we'll be considering 16 neighbouring points.
- What if initially we only have 4 known points like in previous questions?

→ We pad the values to make it a 4x4 matrix. We can use bilinear ~~to~~ or any other method to make it 4x4 & then do bicubic interpolation.

$$\begin{bmatrix} P_1 & P_2 & P_{unknown} & P_3 & P_4 \\ P_5 & P_6 & & P_7 & P_8 \\ P_9 & P_{10} & & P_{11} & P_{12} \\ P_{13} & P_{14} & & P_{15} & P_{16} \end{bmatrix}$$

$$P_{unknown} = P_1 q_1 + P_2 q_2 + P_3 q_3 + P_4 q_4$$

In bilinear $P_{unknown} = P_1 q_1 + P_2 q_2$

where q_1 & q_2 ~~was~~ were distance.

But in bicubic q_1, q_2, q_3, q_4 are polynomial functions. ~~based on~~

Based on Herman interpolation.

Derivation:

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$P'(t) = a_1 + 2a_2 t + 3a_3 t^2$$

→ $P(0) = f_0 = a_0$ — (1)

→ $P(1) = f_1 = a_0 + a_1 + a_2 + a_3$ — (2)

→ $P'(0) = f_0' = a_1$ — (3)

→ $P'(1) = f_1' = a_1 + 2a_2 + 3a_3$ — (4)

$$f_1 = a_0 + a_1 + a_2 + a_3$$
$$= f_0 + f_0' + a_2 + a_3$$

$$a_2 + a_3 = f_1 - f_0 - f_0'$$

$$f_1' - f_1 = -a_0 + a_2 + 2a_3$$

$$f_1' = a_1 + 2a_2 + 3a_3$$

$$3f_1 = f_1' + 3a_0 + 2a_1 + a_2$$

$$3f_1 = f_1' + 3f_0 + 2f_0' + a_2$$

$$f_1 - f_0 - f_0' + f_1' - f_1 = -a_0 + 2a_2 + 3a_3$$

$$f_1 - f_0 - f_0' + f_1' - f_1 = -f_0 + f_1' - a_1$$

$$f_1 \cancel{- f_0} - f_0' + f_1' - f_1 = \cancel{-f_0} + f_1' - \cancel{f_0'}$$

$$0 = 0$$

$$a_0 = f_0$$
$$a_1 = f_0'$$

$$f_1' - 2f_1 = a_1 + \cancel{2a_2} + 3a_3 - (2a_0 + 2a_1 + \cancel{2a_2} + 2a_3)$$
$$= -2a_0 - a_1 + a_3$$

$$f_1' - 2f_1 = -2f_0 - f_0' + a_3$$

$$a_3 = f_1' - 2f_1 + 2f_0 - f_0'$$

Similarly find a_4.

Edge Detection:

Edges: abrupt changes in intensity, discontinuity in image brightness or contrast; usually edges occurs on the boundary of two regions.

$a_0 = f_0$
$a_1 = f_0'$

$$f_1' - 2f_1 = a_1 + 2a_2 + 3a_3 - (2a_0 + 2a_1 + 2a_2 + 2a_3)$$
$$= -2a_0 - a_1 + a_3$$

$$f_1' - 2f_1 = -2f_0 - f_0' + a_3$$

$$a_3 = f_1' - 2f_1 + 2f_0 - f_0'$$

Similarly find a_4.

Edge Detection:

Edges: abrupt changes in intensity, discontinuity in image brightness or contrast; usually edges occur on the boundary of two regions.

25/2/25 | Tue

Masks for edge detection:

1. Prewitt operator
2. Sobel operator
3. Robinson Compass masks
4. Kirsch Compass masks
5. Laplacian operator

1. Prewitt Operator:

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

vertical edges — horizontal edges

- can detect only horizontal & vertical edges.

2. Sobel Mask:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

vertical — horizontal

- Very similar to Prewitt.
- We can change -2 & 2. We could even give more weightage unlike Prewitt where it is a constant.

3. Robinson Compass Mask:

- 8 Masks each corresponding to: North, South, East, West, ~~No.~~ NE, NW, SE & SW.

rotate

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \xrightarrow{\text{rotate}} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

NW N NE

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

You rotate towards right.

W S E

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

SW SE

4. Kirsch Compass:

Same as Robinson except more weightage.

5. Laplacian

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

Negative Laplacian Positive Laplacian.

Canny's Edge Detection:

- Multi-step process that detects edges in images.

Img in Gray Level → Gaussian Smoothening → Take 1st derivative → Find Magnitude & Direction → Non-Maximal Suppression → Hysteresis Thresholding → Edge detected image

(Take 1st derivative: application of sobel filter over x & y separately.)

Gaussian Filter (Gaussian Smoothening)

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

You apply this formula to get each value in the kernel given that you know x, y & σ.

coordinate is (0,0)

$$\underbrace{\frac{1}{16}}_{\text{sum of all elements in}} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

As σ increases, degree of smoothening also increases

From Take 1st Derivative step you get G_x + G_y using sobel filter on horizontal & vertical done separately.

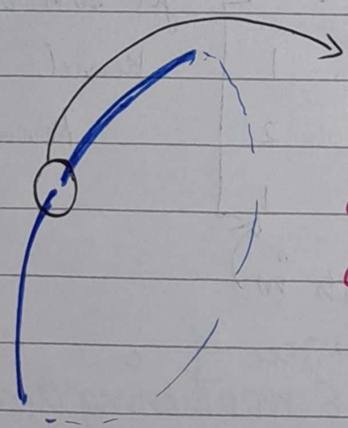
Now in nxt step

$$|G| = \sqrt{G_x^2 + G_y^2} \rightarrow \text{magnitude}$$

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

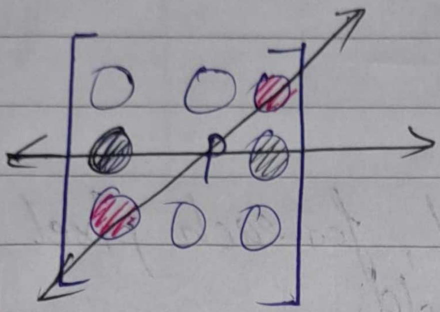
to show how strong the edge is.

Using these data you suppress or do not suppress using non-maximal suppression. How?



for that point we select 2 nearest neighbours along the gradient then compare the magnitude.

3/3/25 | Monday.



θ

↑

say angle is 15° then you consider these two points.

If it is 45° then you consider these two points

4/3/25 | Tuesday.

Segmentation:

- partitioning a digital image into multiple segments

Techniques:

1. Thresholding
2. Region growing
3. Edge-based segmentation
4. Clustering
5. Watershed
6. Graph based
7. Deep Learning based.

classmate
Date
Page

1. Thresholding

1. Simple Thresholding

- ~~For ea~~ Given a threshold, for each pixel compare the pixel value & threshold.

  If pixel value > threshold
  → pixel value = 1
  elif pixel value < threshold
  → pixel value = 0
  pixel value = threshold
  → pixel value = 0 or 1 → You can decide.

- We have to manually assign threshold value.

2. Otsu Thresholding:

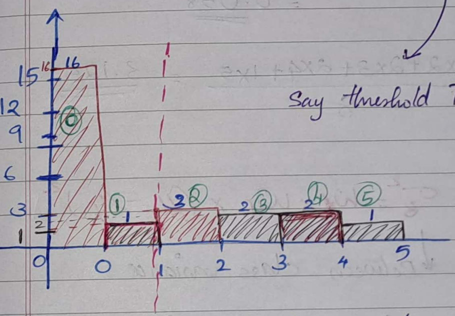
- selects a threshold that minimizes the within class variance.
- We need to plot a histogram
  ↳ plots frequencies.
- Plot frequencies of pixel intensities using histogram.

classmate
Date
Page

eg.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 3 & 0 \\ 0 & 2 & 3 & 4 & 0 \\ 0 & 1 & 4 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

histogram plot.



Say threshold $T = 2$.

$T \leq$ value
↳ foreground class

$T >$ value.
↳ background class.

$$W_b = \frac{\text{~~b class~~ background class elements}}{\text{total elements}}$$

$$= \frac{16+1}{25} = \frac{18}{25} \; \frac{17}{25} = \frac{72}{100} = 0.72$$

$$= \frac{68}{25 \times 4} = \underline{0.68}$$

$$W_f = \frac{\text{foreground class elements}}{\text{total}} = \frac{3+2+2+1}{25}$$

$$= \frac{8}{25} = \underline{\underline{0.32}}$$

$$\mu_b = \frac{\sum_i freq_i \times value_i}{\text{total freq of background}}$$

$$= \frac{16\times0 + 1\times1}{17} = 1/17$$

$$= \underline{\underline{0.058}}$$

$$\mu_f = \frac{3\times2 + 2\times3 + 2\times4 + 1\times5}{8} = \underline{\underline{3.125}}$$

$$\sigma_B^2 = W_b \times W_f \times (\mu_b - \mu_f)^2$$

↓ Between class variance

$$= 0.68 \times 0.32 \times (0.058 - 3.125)^2$$

$$= \underline{\underline{2.04}}$$

The σ_B^2 should be maximum.

Such a threshold must be chosen!

What if $T = 3$?

value	freq
0	16
1	1
2	3
3	2
4	2
5	1

$$W_b = \frac{16+1+3}{25} = \frac{20}{25} = \frac{80}{100} = \underline{\underline{0.8}}$$

$$W_f = \frac{2+2+1}{25} = \frac{5}{25} = \frac{1}{5} = \underline{\underline{0.2}}$$

$$\mu_b = \frac{16\times0 + 1\times1 + 3\times2}{20} = \frac{7}{20} = \frac{35}{100} = \underline{\underline{0.35}}$$

$$\mu_f = \frac{3\times2 + 4\times2 + 5\times1}{5} = \frac{6+8+5}{5} = \frac{19}{5} = \underline{\underline{3.8}}$$

$$\sigma_B^2 = \underline{\underline{1.9044}}$$

5/3/25 | Wednesday.

2. Region Growing:

$$i/p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 2 & 2 & 1 \\ 9 & 7 & 5 & 5 \\ 1 & 6 & 7 & 1 \end{bmatrix}$$

For this also we need a threshold value, let it be T.

Let $T = 3$.

let this be the initial seed point.

Step1: For the given img choose an initial seed point.

initial seed pixel intensity = 5

2. Look into neighbours. It can be 4-way (4 neighbours) → , 8-way (8 neighbours).
   We choose 4-way ↓
   neighbours → 2, 7, 7, 5

3. If absolute diff. b/w the neighbouring points & the initial seed point is ≤ threshold, then segment it into same segment (where seed point is present).

eg: $2 \rightarrow |5-2| = 3 \le 3$ → 2 is added to 5.

$7 \rightarrow |5-7| = 2 \le 3$ → 7 is added to 2,5

$5 \rightarrow |5-5| = 0 \le 3$ → 5 is added to 2,5,7,7

Thus it becomes:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1^* & ^*2 & ^*2 & ^*1 \\ 9 & 7^* & 5 & ^*5 \\ 1 & 6 & 7 & 1 \end{bmatrix} \qquad T = 3.$$

Now go to nxt point, initial seed = 7.

$|7-2| = 5 > 3$ No

$|7-9| = 2 \le 3$ Yes

$|7-6| = 1 \le 3$ Yes.

initial seed = 5

$|5-1| = 4 > 3$ No

$|5-1| = 4 > 3$ No

initial seed = 2

$|2-0| = 2 \le 3$ Yes

$|2-1| = 1 \le 3$ Yes

$|2-2| = 0 \le 3$ Yes

initial seed = 1

$|1-1| = 0 \le 3$ Yes

initial seed = 2

$|2-0| = 2 \le 3$ Yes

$|2-1| = 1 \le 3$ Yes

initial seed = 1

$|1-0| = 1 \le 3$ Yes.

T=2.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 2 & 2 & 1 \\ 9 & 7^* & 5 & 5 \\ 1 & 6 & 7 & 1 \end{bmatrix}$$

$S=5$

$|5-5| = 0 \leq 2$ (Y)

$|5-7| = 2 \leq 2$ (Y)

$|5-2| = 3 > 2$ (N)

$S=7$

$|7-9| = 2 \leq 2$ (Y)

$|7-6| = 1 \leq 2$ (Y)

$|7-2| = 5 > 2$ (N)

Now make everything in segment to 1 & others to zero,

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

T=2

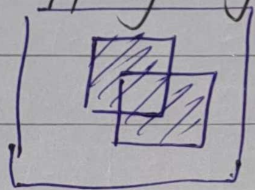
classmate
Date
Page

10/3/25 | Monday.

3. ~~Watershed~~

Disadv. of region growing:

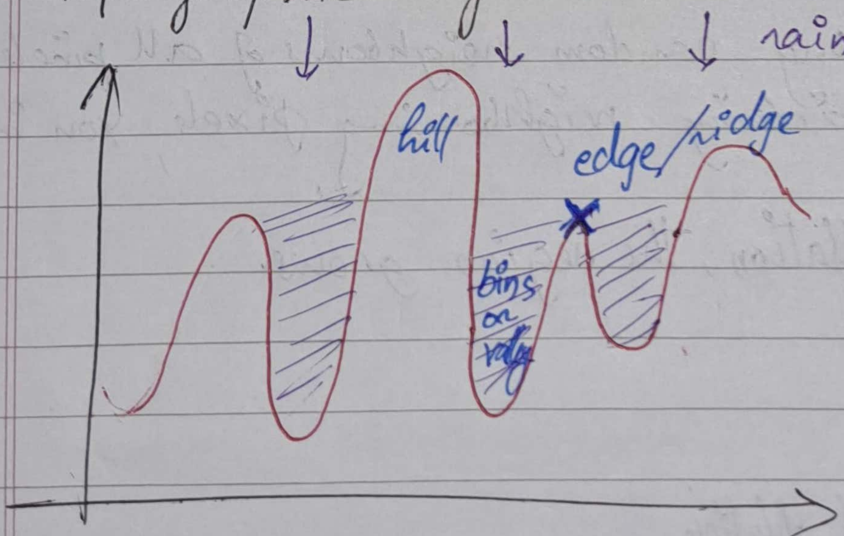
1. Computational complex.
   pixels that we have already marked as foreground might be checked again.
2. Overlapping objects.



- the overlapping object may be considered as a single object in this.

3. Watershed

- Topographical features



Steps:

1. RGB → binary image
   - we can use thresholding technique
2. Noise reduction
   - Use Gaussian filter, we can do noise reduction.
3. Morphological operators
   - → Dilation
   - → Erosion

Dilation:

- structuring element
  - ↳ something similar to Sobel filter, Laplacian etc.
- You do convolution
- Considering random neighbours of all pixels.
- By considering neighboring pixels, you increase width.
- After dilation, the region grows.

Erosion:

- Opposite of dilation
- shrinkage of region.

11/3/25 Tuesday.

signifies all neighbours are to be considered.

eg:

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

i/p img. structuring element.

Perform dilation.

A: The structuring element is applied on all pixels. If any of the neighbours is one then take 1. Corner elements has only 3 neighbours.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

→ the neighbours of that 1 was all zeroes but originally it was one.

We only change zeroes in dilation

Q. Do erosion.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

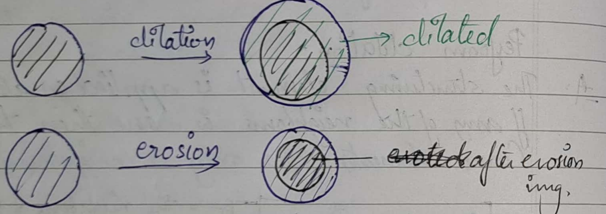
If even one neighbour is zero make the one to zero.

13|03|25| Thursday.

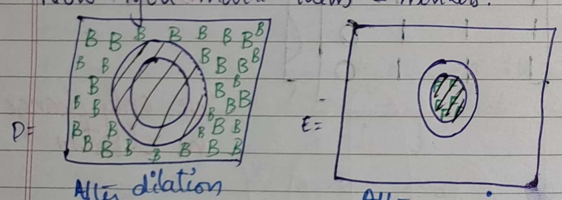
- By doing dilation we get sure background pixels.
- By doing erosion we get sure foreground pixels.

4. ~~Markers~~ Markers:

- We do dilation - erosion. How? ↓
-



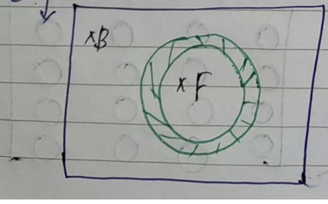
- Now you mark them - markers.



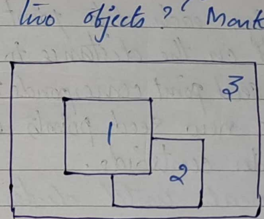
After dilation

After erosion

- Now D-E ↓



- What if two objects? Markers will be of 3 classes.



- Now you have marked the regions.

5. Flooding:

- We are doing flooding, so that each segment of the image has a different colour.
- flooding starts from the markers.
- ~~How to fin~~
- How to find boundary? ↓
  The region where the two colours intersect.

4. K-Means Clustering:

- K - no: of segments / clusters.
  $k=3 \Rightarrow 2$ objects + 1 background.
- Step 1: Take random k seedpoints where k is number of clusters.
  Step 2: Compute Euclidean dist b/w each point & a seed point.

Step3: Assign each data point to a segment based on the distance to the seed point. (Each seed point corresponds to one segment).

Step4: Take new seed points ~~base~~ by finding cluster centroids.

Step5: Repeat until clusters no longer changes.

How to do it in an image?

Based on intensity of image pixels.

You do not consider coordinates instead just take absolute value of difference of the two pixel intensities as distance.