

4/2/25 | Tuesday.

GAYATHRI B NAI
AM.EN.U4AIE22117

classmate

Date _____

Page _____

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 \rightarrow Projective Space.

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

- 2D points represented using homogenous coordinates.

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

↓
scaling.)

Why do we use augmented matrix repeatedly?

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

form of augmented matrix

$(x, y, 1) \rightarrow$ 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 8 units, the whole thing is scaled by 3 ($\tilde{w}=3$).
↳ scaling.

2D Lines :

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

corresponding line equation:

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

Intersection of two lines:

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

Line joining two points:

$$\bar{x} = \tilde{n}_1 \times \tilde{n}_2$$

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
 another vector ↓ ↓ the line intersecting the
 skew vector symmetric lines \tilde{a} & \tilde{b} .

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

Ans

$$\tilde{l} = (x, y, z)$$

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

5/2/25 | Wednesday.

$$\text{eg: } l_1 \rightarrow y = 1 \\ \downarrow (0 \ 1 \ -1)$$

$$0x + 1xy + -1 = 0 \\ \Rightarrow y - 1 = 0 \\ \underline{y = 1}$$

$$l_2 \rightarrow n = 2 \\ \downarrow (1 \ 0 \ -2)$$

$$1x + 0xy + -2 = 0$$

$$n - 2 = 0$$

$$\underline{n = 2}$$

$\tilde{l}_3 \rightarrow$ 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

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

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

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

augmented
part.

- Q. What do we get from getting the scalar vectors 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)

$$\hookrightarrow t_x + t_y$$

- shifting in x-y plane
- $\bar{x}' = \bar{x} + t$

OR

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

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

$I \rightarrow$ identity matrix

$$\rightarrow (3 \times 3) \times (3 \times 1) = (3 \times 1)$$

\downarrow transformation matrix. \downarrow original point \bar{x} \downarrow 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)

$$1 \text{ DoF} + 2 \text{ DoF} = 3 \text{ DoF}$$

Combining rotation and translation

$$\bar{x}' = Rx + t$$

or

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

$R \rightarrow$ orthonormal 3×2 notation matrix.

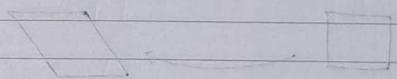
$$\bar{x}' = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

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

$$|R| = 1 \quad R^T R = R R^T = I$$

\hookrightarrow orthonormal.

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



3. Similarity (4 DoF)

$$2 \text{DoF} + 1 \text{DoF} + 1 \text{DoF} = 4 \text{DoF}$$

- Translation & then scaled rotation

- $\bar{x}' = SRx + t$
- ↳ we are doing uniform scaling
so 1 DoF. If it were non-uniform
- OR
- $$\bar{x}' = \begin{bmatrix} SR & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$
- then 2 DoF.

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

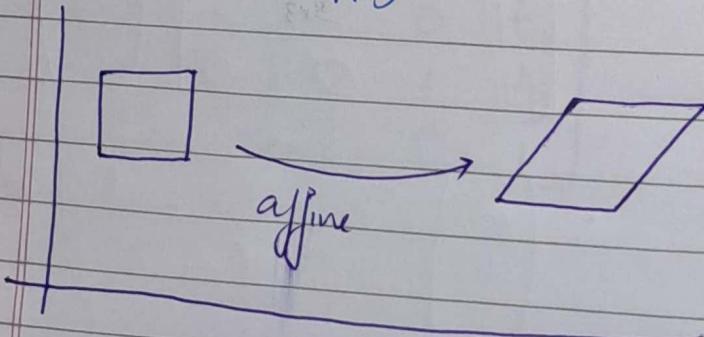
s - arbitrary scale factor.

4. Affine transformation (6 DoF)

- $x' = Ax$
- Affinity matrix \downarrow has six elements $\rightarrow 6 \text{DoF}$

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

- When you apply shear, it causes affine transformation



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

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

5. Projective transformation / Homography (8 DoF)

- operates on homogeneous coordinates.

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

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

but lines remain parallel after the transformation.

- $\tilde{\mathbf{r}}' = \mathbf{A} \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \tilde{\mathbf{r}}$

5. Projective transformation / Homography (8 DoF)

- operates on homogeneous coordinates.

- $\tilde{\mathbf{r}}' = \tilde{\mathbf{H}} \tilde{\mathbf{r}}$
 \hookrightarrow arbitrary 8×3 homogeneous matrix.
 (The last element (h_{33}) is 1 so the
 DoF is 8 if not 9).

12/02/25 | Wednesday.

3D Transformations

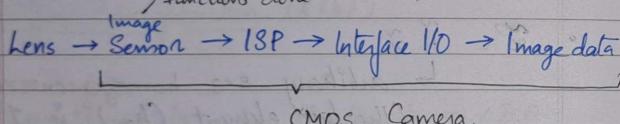
Transformation	Matrix	DoF	Preserves	Icon
translation	$[I \ t]_{3 \times 4}$	3	orientation	<input type="checkbox"/>
Euclidean	$[R \ t]_{3 \times 4}$	6	lengths	<input checked="" type="checkbox"/>
Similarity	$[sR \cdot t]_{3 \times 4}$	7	angles	
Affine	$[A]_{3 \times 4}$	12	parallelism	<input type="checkbox"/>
Projective	$[\bar{H}]_{4 \times 4}$	15	straight lines.	<input checked="" type="checkbox"/>

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.



Q How to convert analog to digital?

- ↳ 1. Sampling
- 2. Quantization
- 3. Encoding

1. Sampling.

Digitalizing

Digitizing along (x, y) coordinates

Marking grids

Quantization:

Digitizing along the amplitude.

↳ Intensity of the image.

Q How many grayscale levels?

↳ 8 bit → 2^8 grayscale level quantization.
excluding white & black there are 254 grayscale levels.

255 ↳ 0

2. Quantization:

Digitizing along the amplitude
 ↳ intensity of the image.

Q How many grayscale levels?

↳ 8 bit $\rightarrow 2^8$ grayscale level quantization.

excluding white & black there are 254 grayscale levels.

$$\begin{matrix} 255 \\ \downarrow \\ 0 \end{matrix}$$

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.

e.g.: optical center, focal length & radial distortion coefficients of the lens.

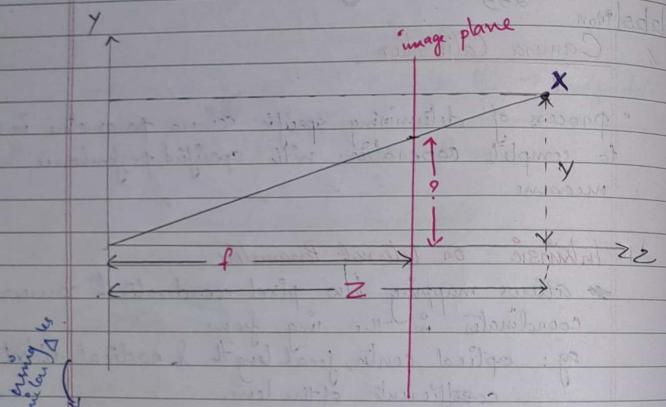
2. Extrinsic or External parameters

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 transformed 2D vector.

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



$$[x \ y \ z]^T = [f \cdot \frac{x}{z} \ f \cdot \frac{y}{z}]^T$$

$$\frac{f}{z} = \frac{y}{x}$$

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

Camera Matrix:

A camera is a mapping from the 3D world to a 2D image.

$$\begin{matrix} x \\ y \\ z \end{matrix} = \begin{matrix} P \\ \downarrow \\ \text{camera matrix} \end{matrix} \begin{matrix} X \\ Y \\ Z \end{matrix} \quad \begin{matrix} \rightarrow \\ \text{2D image point} \end{matrix} \quad \begin{matrix} \rightarrow \\ \text{3D world point} \end{matrix}$$

$$\begin{matrix} x \\ y \\ z \\ 1 \end{matrix} = \begin{matrix} P_1 & P_2 & P_3 & P_4 \\ P_5 & P_6 & P_7 & P_8 \\ P_9 & P_{10} & P_{11} & P_{12} \end{matrix}_{3 \times 4} \begin{matrix} X \\ Y \\ Z \\ 1 \end{matrix} \quad \begin{matrix} \rightarrow \\ \text{homogeneous world point} \end{matrix}$$

Camera Matrix

decomposed into 3 matrices
principal point coordinates \rightarrow External calibration matrix

$$P = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{3 \times 4} \Rightarrow P = K[I|0]$$

\hookrightarrow Internal calibration matrix.

classmate

Date _____

Page _____

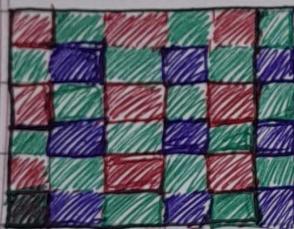
- ★ The camera is already aligned with the real world coordinates hence why the external calibration matrix is always $[I|0]$.
If it is not aligned, we can apply rotation & translation to align it.

★ 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.

$$\frac{10}{3} \times \frac{2}{3}$$

$$= \frac{20}{9}$$

1. Nearest Neighbors Interpolation (NNI).

3	4
5	6

2x2

• 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.



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

4x4

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

• Simplest method.

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

2. Bilinear Interpolation:

→ distance.

3	4
5	6

2x2

Task is the same. Again we know only

the corner points → assume dist remains same even after scaling.

3	3.33	3.66	4
3.66	4	4.32	4.66
3.32	4.64	4.99	5.33
5	5.32	5.66	6

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)

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

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

$$3 \times \left(1 - \frac{2}{3}\right) + 4 \times \left(1 - \frac{1}{3}\right)$$

$$= 3 \times \frac{2}{3} + 4 \times \frac{1}{3} = 2 + 1.33 = 3.33$$

24/12/2021 Monday.

~~25/12/2021 Tuesday~~

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 4×4 matrix.
We can use bilinear or any other method to make it 4×4 & then do bicubic interpolation.

P ₁	P ₂	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

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

In bilinear $P_{\text{unknown}} = P_1 q_1 + P_2 q_2$
where q_1, q_2 were distance.

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

Based on Hermann interpolation.

classmate
Date _____
Page _____

classmate
Date _____
Page _____

Derivation:

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

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

$$\rightarrow P(0) = f_0 = a_0 \quad \dots \quad (1)$$

$$\rightarrow P(1) = f_1 = a_0 + a_1 + a_2 + a_3 \quad \dots \quad (2)$$

$$\rightarrow P'(0) = f'_0 = a_1 \quad \dots \quad (3)$$

$$\rightarrow P'(1) = f'_1 = a_1 + 2a_2 + 3a_3 \quad \dots \quad (4)$$

$$\begin{aligned} f'_1 &= a_0 + a_1 + a_2 + a_3 \\ &= f_0 + f'_0 + a_2 + a_3 \end{aligned}$$

$$f'_1 - f_1 = -a_0 + a_2 + 2a_3$$

$$f'_1 = a_1 + 2a_2 + 3a_3 \quad \rightarrow a_2 + a_3 = f'_1 - f_0 - f'_0$$

$$3f_1 = f'_1 + 3a_0 + 2a_1 + a_2$$

$$3f_1 = f'_1 + 3f_0 + 2f'_0 + a_2$$

$$f'_1 - f_0 - f'_0 + f'_1 - f_1 = -a_0 + 2a_2 + 3a_3$$

$$f_1 - f_0 - f'_0 + f'_1 - f_1 = -f_0 + f'_1 - a_1$$

$$f_1 - f_0 - f'_0 + f'_1 - f_1 = -f_0 + f'_1 - f'_0$$

$$0 = 0 \quad (2 -)$$

$$q_0 = f_0$$

$$q_1 = f'_0$$

$$\begin{aligned} f'_1 - 2f_1 &= q_1 + 2q_2 + 3q_3 - (q_0 + 2q_1 + 2q_2 + 2q_3) \\ &= -2q_0 - q_1 + q_3 \end{aligned}$$

$$f'_2 - 2f_2 = -2f_0 - f'_1 + q_3$$

$$q_3 = f'_1 - 2f_1 + 2f_0 - f'_2$$

Similarly find q_4 .

Edge Detection :

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

$$a_0 = b_0$$

$$a_1 = b_1$$

$$f'_1 - 2f_1 = a_1 + 2a_2 + 3a_3 - (a_0 + a_1 + a_2 + a_3)$$

$$= -2a_0 - a_1 + a_3$$

$$f'_2 - 2f_2 = -2b_0 - b_1 + b_3$$

$$a_3 = f'_1 - 2f_1 + 2f_2 - f_0$$

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.

25/01/25/Tue

Masks for edge detection:

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

Prewitt Operator:

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

vertical edges horizontal edges

- can detect only horizontal & vertical edges.

Sobel Mask:

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

vertical

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

horizontal

- Very similar to Prewitt

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

Robinson Compass Mask:

- 8 Masks each corresponding to: North, South, East, West, NE, NW, SE & SW.

CLASSMATE
Date _____
Page _____

rotate

rotate

NW N NE

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

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & -2 & -1 \end{bmatrix} \quad \text{You rotate towards right.} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

W S E

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

SW SE

4. Kirsch Compass

Same as Robinson except more weightage.

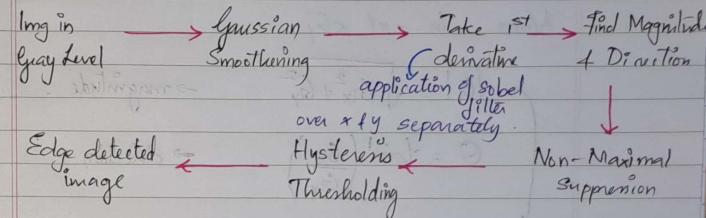
5. Laplacian

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

Negative Laplacian Positive Laplacian

Canny's Edge Detection:

- Multi-step process that detects edges in images.



Gaussian Filter (Gaussian Smoothening)

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

You apply this formula to get each value in the coordinate is $(0, 0)$

kernel gives that you know x, y to

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

sum of all elements in

As σ increases, degree of smoothing also increases

From 1st Derivative step you get
 $G_{x,y}$ using sobel filters on horizontal
 & vertical done separately.

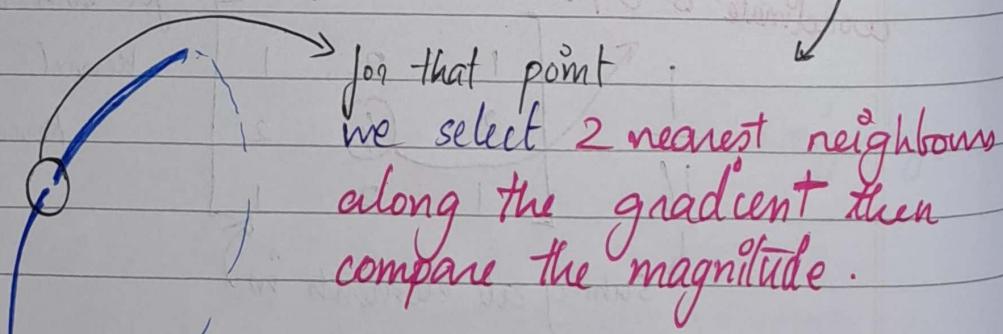
Now in next step

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

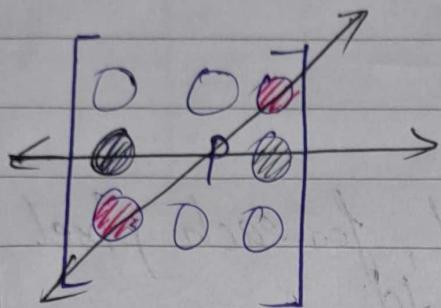
$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \rightarrow \text{direction}$$

to show how strong the edge is.

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



3/3/25 Monday.



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.

1. Thresholding

1. Simple Thresholding

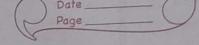
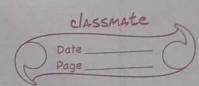
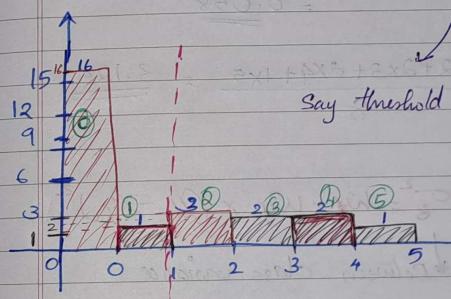
- Given a threshold, for each pixel compare the pixel value & threshold.
 - If pixel value $>$ threshold
 \rightarrow pixel value = 1
 - elif pixel value $<$ threshold
 \rightarrow pixel value = 0
 - pixel value = threshold
 \rightarrow pixel value = 0 or 1 → You can decide.
- We have to manually assign threshold value.

2. Otsu Thresholding:

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

e.g.

0	0	0	0	0
0	2	2	3	0
0	2	3	4	0
0	1	4	5	0
0	0	0	0	10



$$w_b = \frac{\text{background class elements}}{\text{total elements}}$$

$$= \frac{16+1}{25} = \frac{17}{25} = \frac{2}{25} = 0.22$$

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

$$w_f = \frac{\text{foreground class elements}}{\text{total}} = \frac{3+2+2+1}{25} = \frac{8}{25} = \underline{\underline{0.32}}$$

$$\mu_b = \frac{\sum \text{freq} \times \text{value}}{\text{total freq of background}}$$

$$= \frac{16 \times 0 + 1 \times 1}{17} = \frac{1}{17} \\ = 0.058$$

$$\mu_f = \frac{3 \times 2 + 2 \times 3 + 2 \times 4 + 1 \times 5}{8} = 3.125$$

$$\sigma_B^2 = w_f \times w_b \times (\mu_f - \mu_b)^2$$

↓ Between class variance

$$= 0.68 \times 0.32 \times (0.058 - 3.125)^2 \\ = 2.04$$

The σ_B^2 should be maximum.

Such a threshold must be chosen!

What if $T=3$? value freq.

$w_b = \frac{16+1+3}{25}$	1 - 1
$= \frac{20}{25} = 0.8$	2 - 3
	3 - 2
	4 - 2
	5 - 1

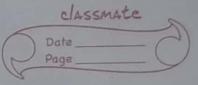
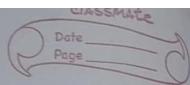
$$w_f = \frac{2+2+1}{25} = \frac{5}{25} = \frac{1}{5} = 0.2$$

$$\mu_b = \frac{16 \times 0 + 1 \times 1 + 3 \times 2}{20} = \frac{7}{20} = \frac{35}{100} = 0.35$$

$$\mu_f = \frac{3 \times 2 + 4 \times 2 + 5 \times 1}{5} = \frac{19}{5} = 3.8$$

$$\sigma_B^2 = 1.9044$$

5/3/25 Wednesday.



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.

Step 1: For the given image choose an initial seed point.

initial seed pixel intensity = 5

2. Look into neighbours. It can be 4-way (4 neighbours) \rightarrow , 8-way (8 neighbours). We choose 4-way \rightarrow neighbours \rightarrow 2, 7, 1, 5

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

$$\text{eg: } 2 \rightarrow |5-2| = 3 \leq 3 \rightarrow 2 \text{ is added to } 5$$

$$7 \rightarrow |5-7| = 2 \leq 3 \rightarrow 7 \text{ is added to } 5$$

$$5 \rightarrow |5-5| = 0 \leq 3 \rightarrow 5 \text{ is added to } 2, 5, 7$$

Thus it becomes:

0	0	10	
*	2	2	*
9	7*	7*	5
1	6	7	1

$$T=3$$

Now go to next point, initial seed = 7.

$$|7-2| = 5 \geq 3 \quad \text{No}$$

$$|7-9| = 2 \leq 3 \quad \text{Yes}$$

$$|7-6| = 1 \leq 3 \quad \text{Yes}$$

initial seed = 5

$$|5-1| = 4 \geq 3 \quad \text{No}$$

$$|5-1| = 4 \geq 3 \quad \text{No}$$

initial seed = 2

$$|2-0| = 2 \leq 3 \quad \text{Yes}$$

$$|2-1| = 1 \leq 3 \quad \text{Yes}$$

$$|2-2| = 0 \leq 3 \quad \text{Yes}$$

initial seed = 1

$$|1-1| = 0 \leq 3 \quad \text{Yes}$$

initial seed = 2

$$|2-0| \leq 2 \leq 3 \quad \text{Yes}$$

$$|2-1| = 1 \leq 3 \quad \text{Yes}$$

initial seed = 1

$$|1-0| = 1 \leq 3 \quad \text{Yes}$$

T₂₂.

0	0	0	1	
1	2	2	1	
9	7*	1/8	1/5	
1	6	0	7	1

$s = 5$

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

$|5-1| = 2 \leq 2 \text{ (Y)}$

$|5-2| = 3 > 2 \text{ (N)}$

$s = 7$

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

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

$|7-2| = 5 > 2 \text{ (N)}$

Now make everything in segment to 14
 others to zero,

0	0	0	0	0
0	0	0	0	0
1	1	1	1	1

$T=2$

10/3/25 | Monday.

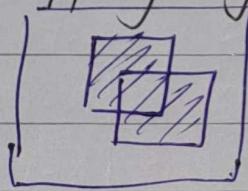
3. Watershed

Disadv. of region growing:

1. Computational complex.

pixels that we have already marked as foreground might be checked again.

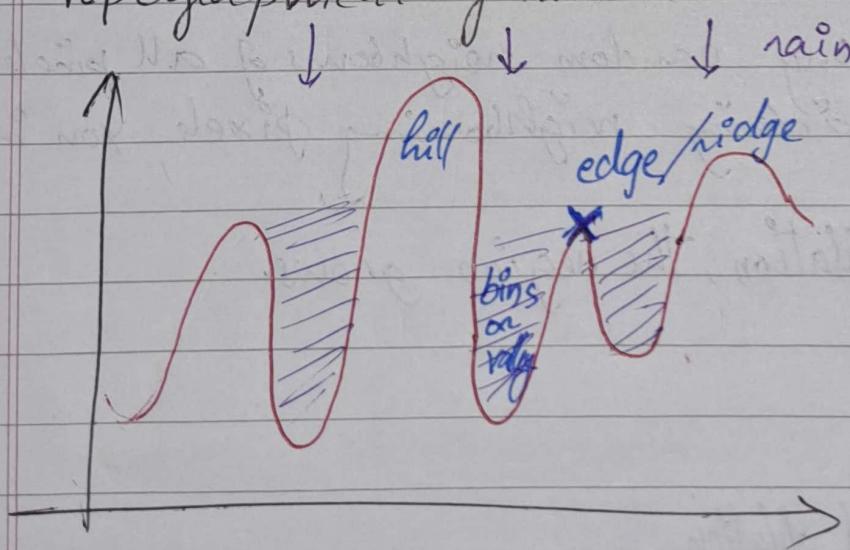
2. Overlapping objects.



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

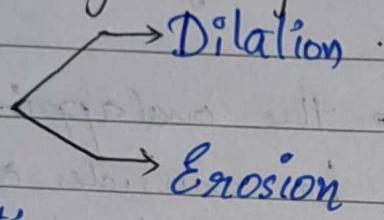
3. Watershed

- Topographical features.



Steps:

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



Dilation:

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

Erosion:

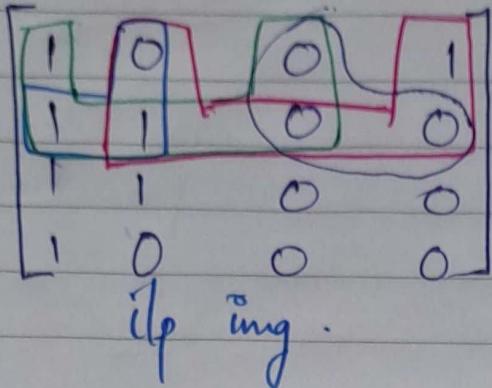
- Opposite of dilation
- shrinkage of region

11/3/25 Tuesday.

classmate
Date _____
Page _____

signifies all neighbours are to be considered

eg



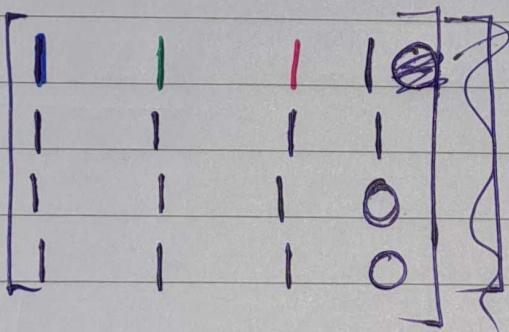
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

structuring element.

if img.

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.



the neighbors of that is 1 was all zeroes but originally it was one.

We only change zeroes in dilation

Q. Do erosion:

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

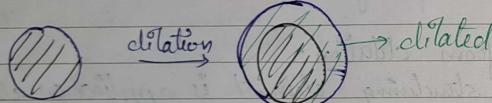
If even one neighbour is zero make the one to zero.

13/03/25 Thursday.

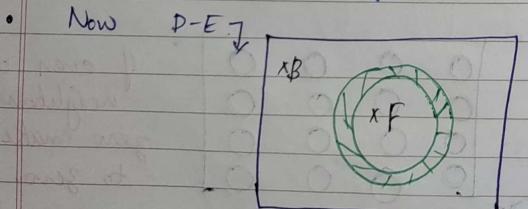
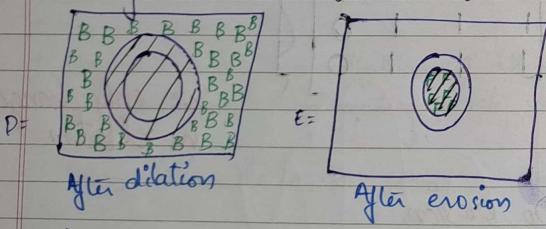
- By doing dilation we get some background pixels.
- By doing erosion we get some foreground pixels.

4. ~~Markers~~ Markers:

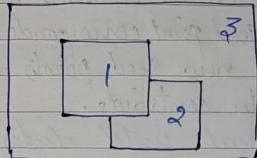
- We do dilation - erosion. How? ?



- Now you mark them - markers.



- 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 find boundary?
- The region where the two colours intersect.

A. 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.

Step 3: Assign each data point to a segment based on the distance to the seed point.

(Each seed point corresponds to one segment).

Step 4: Take new seed points ~~base~~ by finding cluster centroids.

Step 5: Repeat until cluster no longer changes.

Q 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.

- Step 3: Assign each data point to a segment based on the distance to the seed point.
 (Each seed point corresponds to one segment)
- Step 4: Take new seed points by finding cluster centroids.
- Step 5: Repeat until cluster no longer change.

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.

17/3/2021 UNIT-2

Feature Detection:

Currently we have segregated the foreground objects.
 Now we need to identify the features of the foreground objects.

1. Harris Corner Detection Algorithm:

- to detect corners in the given images.
- How?

0	0	0	1	5
0	1	1	4	5
1	2	3	4	4
1	2	2	3	4
0	2	3	3	3

Step 1:
 Find the gradients using sobel or prewitt or etc.

~~for gradient computation:~~

$$\text{Prewitt } P_x \rightarrow \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\text{Prewitt } P_y \rightarrow \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Assuming that we are doing 1D convolution.

$$\begin{bmatrix} 1 & 1 & 4 \\ 2 & 3 & 4 \\ 2 & 2 & 3 \end{bmatrix} \rightarrow \begin{array}{l} \text{Using } P_x \\ \begin{bmatrix} -1 & 0 & 4 \\ -2 & 0 & 4 \\ -2 & 0 & 3 \end{bmatrix} \end{array}$$

$$\begin{bmatrix} 1 & 1 & 4 \\ 2 & 3 & 4 \\ 2 & 2 & 3 \end{bmatrix} \rightarrow \begin{array}{l} \text{Using } P_y \\ \begin{bmatrix} -1 & -1 & -4 \\ 0 & 0 & 0 \\ 2 & 2 & 3 \end{bmatrix} \end{array}$$

Now make it 1D:

$$\left\{ \begin{array}{l} P_x = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \\ P_y = \begin{bmatrix} -6 \\ 0 \\ 7 \end{bmatrix} \end{array} \right.$$

Gradient Computation:

(Just consider the part outside $\boxed{}$ as padding) (Don't do this in exam. In exam if 5×5 matrix is given, then give enough padding to get 5×5 matrix as $\mathbf{L}_x + \mathbf{L}_y$).

$$\mathbf{P}_x \rightarrow \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & -4 & 5 \\ 0 & 1 & 1 & 4 & 5 \\ 1 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 0 & 2 & 3 & 3 & 3 \end{bmatrix} \quad \text{padding to get } 5 \times 5 \text{ matrix as } \mathbf{L}_x + \mathbf{L}_y.$$

$$\begin{bmatrix} 1 & 3 & 4 \\ 2 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \text{ox} = 1 + 1 \times 0 + 1 \times 1$$

$$\mathbf{P}_y \rightarrow \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{L}_y \rightarrow \begin{bmatrix} 2 & 3 & 0 \\ 1 & 1 & -1 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{ox} = 1 + 1 \times 0 + 2 \times 1$$

Step 2: Harris corner matrix, \mathbf{H}

$$\text{calculate: } H_1 = l_x^2 = l_{x_1}^2 + l_{x_2}^2 + l_{x_3}^2 + \dots + l_{x_9}^2$$

$$H_1 = 1^2 + 3^2 + 4^2 + 2^2 + 2^2 + 1^2 + 1^2 + 2^2 = 1 + 9 + 16 + 4 + 4 + 3 + 4 = 41$$

$$\mathbf{H} = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix}$$

$$\text{calculate } H_4 = l_y^2 = l_{y_1}^2 + l_{y_2}^2 + \dots + l_{y_9}^2$$

$$H_2 = H_3 = l_{xy} = l_{x_1} l_{y_1} + l_{x_2} l_{y_2} + \dots + l_{x_9} l_{y_9}$$

$$H_2 = 2^2 + 3^2 + 0^2 + 1^2 + 1^2 + (-1)^2 + 0^2 + 0^2 + (-1)^2 = 4 + 9 + 1 + 1 + 1 + 0 + 1 = 17$$

$$H_3 = H_4 = 2 \times 9 + 0 + 0 + 2 + (-1) + 0 + 0 + (-2) = 12$$

$$\mathbf{H} = \begin{bmatrix} 41 & 12 \\ 12 & 17 \end{bmatrix}$$

6/5/25/Tuesday.

- How to distinguish blob corner edges?
- When two edges intersect, it becomes a corner.
- Variation/intensity change occurs in multiple directions in the case of a corner.

Step 3: After finding this matrix, H find corner scores.

$$C = \det(H) - k(\text{trace}(H))^2$$

↓ ↓
corner score. Sensitivity
 $\begin{matrix} \text{sum of diagonal} \\ \text{elements} \end{matrix}$
 $\begin{matrix} \text{(nd sum)} \end{matrix}$

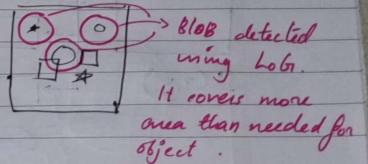
Step 4: Assign thresholds according to C.

- * BLOBs → Binary Large Objects
 - A group of pixels which is sharing a set of features; can be considered as a collective component
 - e.g. each cell is considered as a BLOB
↳ human body cell.

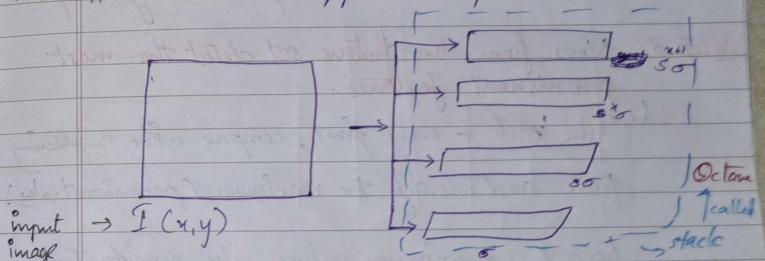
- * LoG → Laplacian of Gaussian.
 - We can detect BLOBs using LoG.

2: SIFT Algorithm → for det.

- for detecting BLOBs.
- Scale Invariant Feature Transform
- In LoG operators, all BLOBs are of same scale.
↳ disadvantage of LoG.

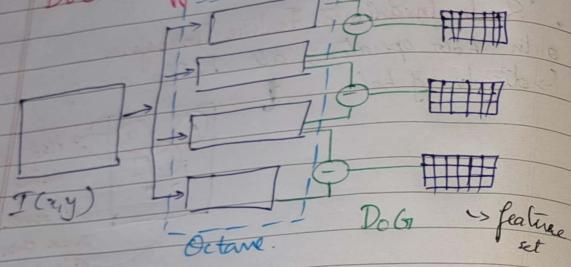


- In SIFT this disadvantage is removed.
- Here Gaussian is applied multiple times unlike LoG.



Step 1: Gaussian is applied with different values of σ . The top image at the top of the stack is the most smoothed (max value of σ). This stack is called Octave.

Step 2: Now after you have Octave, compute the DoG (Difference of Gaussian).

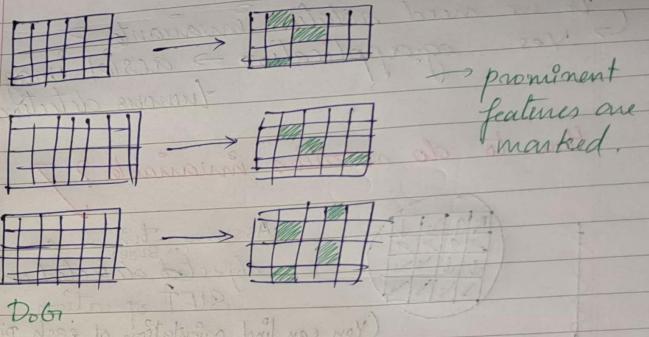


* We could have computed LoG instead of DoG but LoG is more computationally complex.

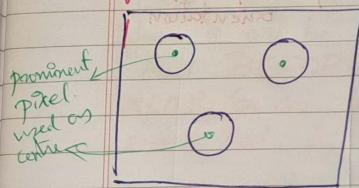
Step 3: Now from the feature set detect the most prominent features.

Take each & every pixel, compare with the neighbouring pixels and identify the maximum (prominent value).

Keep the maximum values else consider two points as weak pixels.



Step 4: Now use these marked prominent features are centres to the BLOBS.

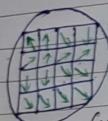


But here the BLOBS are of same size.
How to make it diff size?

Mate different octaves. Change the scale for each octave and then you'll get the diff. size BLOBS.

Applies
Do we need rotation invariant?
yes e.g. application → aerial images,
tumour detection etc.

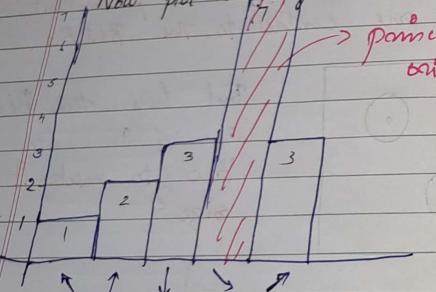
How to do rotation invariant?



Assuming this is an object detected using SIFT operation

(You can find orientation of each pixel by calculating gradient)

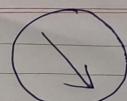
Now plot a histogram.



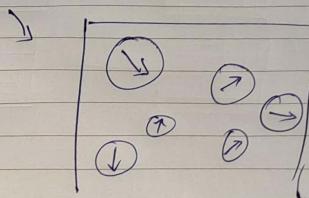
Now the highest frequency is taken as principal orientation.

Now assign the orientation of the

whole BLOB
as the principal orientation.



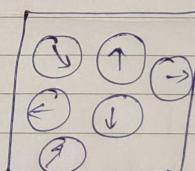
Now find orientation of all the BLOBS like this.



(Randomly assigned).

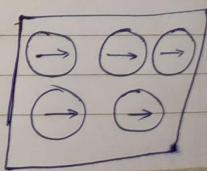
Now as we can't have these BLOBS to have diff. orientation. We need it to align for it to be & rotation invariant.

But BEFORE ALIGNING normalize the BLOBS as in make them of ~~size~~ same size. (That is; normalize the Octave part)



(All BLOBS of same size)

Now rotate the BLOBS and align the orientation



13/5/25 Tuesday.

Why create octaves?

We are taking different levels of smoothing to create octaves. To detect the object at different scales.

n : no. of prominent feature points.

SIFT will be giving us a $n \times m$ matrix.

m : each point which we use to describe each point in the image. Now after this we do feature mapping.

m : dimensions → each point represented using m dimensions.

To do feature mapping b/w two points?

Just do Euclidean distance of the two vectors.

So if we want to find match of a point P_A in img_A in img_B , just find Euclidean distance b/w point P_A & all points in img_B .

Brute force Method. → not efficient. So,

We could do KNN instead.

$img_1 \rightarrow SIFT \rightarrow [n \times 128]$ points selected. → no. of features

$img_2 \rightarrow SIFT \rightarrow [m \times 128]$

→ n can be greater than or less than m .

- Do KNN for each of the vectors (rows) in the matrices.

within a matrix

How do you know which vectors are neighbours?

By taking Euclidean distance. Then you compare points vectors with the two matrices. Then compare the neighbours as well. If they are similar then you can select that point as the point matched (feature mapping).

discussed

done

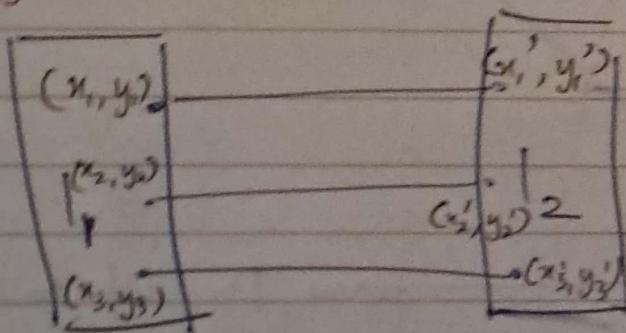
topic

- Do KNN for each of the vectors (rows) in the matrices.

How do you know which vectors are neighbors?
By taking Euclidean distance. Then you compare ~~points~~ vectors with the two matrices. Then compare the neighbors as well. If they are similar then you can select that point as the ~~point~~ point matched (feature mapping).

28/5/25 Wednesday.

Image Alignment



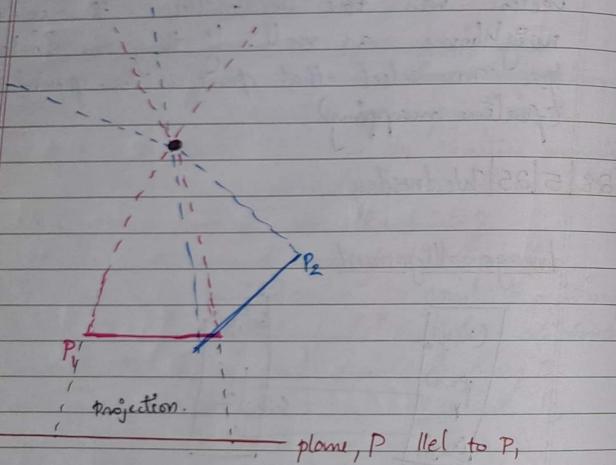
paired feature points which were matched using KNN.

Now you need to stitch it together. How?
By doing transformation.

$T \rightarrow$ transformation matrix.

$x \rightarrow x'$

H1 → Homogeneous transformation matrix
Projective transformation
→ 8 D.o.F.



Project all these viewpoints onto another plane.

CLASSMATE
Date _____
Page _____

CLASSMATE
Date _____
Page _____

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix}$$

(8x3)

2nd img points distincting points
 $(x_s, y_s) \rightarrow (x_d, y_d)$

$$x_d = (h_{11} \times x_s) + (h_{12} \times y_s) + (h_{13} \times 1)$$

$$y_d = (h_{21} \times x_s) + (h_{22} \times y_s) + (h_{23} \times 1)$$

$$z_d = (h_{31} \times x_s) + (h_{32} \times y_s) + (h_{33} \times 1)$$

$$x_d = \frac{x_s}{z_d} = \frac{(h_{11} \times x_s) + (h_{12} \times y_s) + (h_{13} \times 1)}{(h_{31} \times x_s) + (h_{32} \times y_s) + (h_{33} \times 1)}$$

$$y_d = \frac{y_s}{z_d} = \frac{(h_{21} \times x_s) + (h_{22} \times y_s) + (h_{23} \times 1)}{(h_{31} \times x_s) + (h_{32} \times y_s) + (h_{33} \times 1)}$$

After rearranging ↴

$$A = \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \text{ becomes}$$

$$\begin{bmatrix} x_3 & y_3 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_5 & y_5 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_3 & y_3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_5 & y_5 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

A

$$AH = 0$$

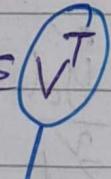
Now solve this & you will get
H. (Use Eigen ~~value~~ vectors)

$$\|A - I\| = 0$$

we will have a set of Eigen Vectors
with us.

Now how to find H?

$$\text{Use SVD. } A = U \Sigma V^T$$



Eigen vectors of $A^T A$ in
decreasing order.

In order to find H, find a minimum

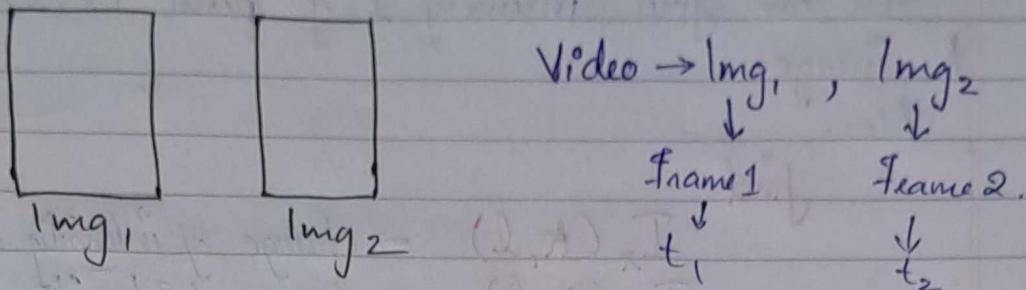
of Pairs of feature match on 8 feature
points. (8 unknowns in H so 8 eqns required.)

2/6/25 Monday.

classmate

Date _____
Page _____

Optical Flow:



Intensity change over time can be measured.

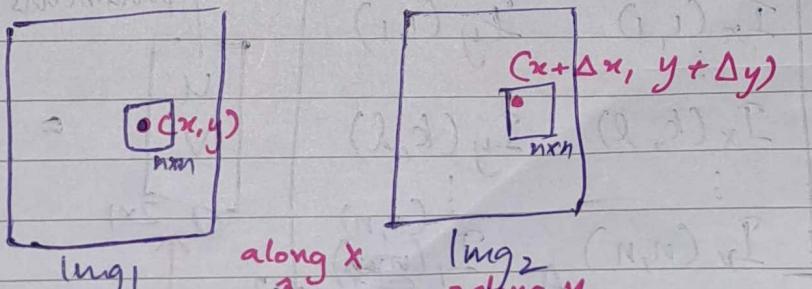
Must calculate change in intensity over time between frames. \rightarrow Optical Flow

\hookrightarrow why calculate?
 \hookrightarrow This will help us estimate motion.

Optical constraint

Optical flow constraint \rightarrow to calculate this, we'll be considering patches in images. The patch must be in the same location in the images.

Then we check how a particular point from the patch changed



Displacement : $(\Delta x, \Delta y)$

Optical flow = $\left(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t} \right)$ \rightarrow that is, velocity.

Let (k, l) be a point in the image.
 You can find intensity I of point (k, l) from the image.

Now, find

- $I_x(k, l)$ → change of intensity along x
- $I_y(k, l)$ → change of intensity along y
- $I_t(k, l)$ → change of intensity over time.

Now there are $n \times n$ pixels / points in the patch.
For thus \Rightarrow $\hookrightarrow R$.

$$\rightarrow \begin{bmatrix} I_x(1,1) & I_y(1,1) \\ \vdots & \vdots \\ I_x(k,1) & I_y(k,1) \\ \vdots & \vdots \\ I_x(n,n) & I_y(n,n) \end{bmatrix}_{n \times 2}$$

$$\begin{bmatrix} I_x(1,1) & I_y(1,1) \\ \vdots & \vdots \\ I_x(k, l) & I_y(k, l) \\ \vdots & \vdots \\ I_x(n, n) & I_y(n, n) \end{bmatrix} \xrightarrow{\text{k} \times 2 \text{ known}} \begin{bmatrix} u \\ v \\ 2x_1 \end{bmatrix} \xrightarrow{\text{Optical Flow parameters}} = \begin{bmatrix} I_t(1,1) \\ \vdots \\ I_t(k, l) \\ \vdots \\ I_t(n, n) \end{bmatrix} \xrightarrow{\text{n} \times 2 \text{ unknown}}$$

$$A \cdot q = B$$

left multiply A^T on both sides

$$A^T A \cdot u = A^T B$$

$$u = (A^T A)^{-1} A^T B$$

↳ This can be solved using least sq.s.

Finally you get $A^T A = \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \dots$

Based on these L values you get to know whether the patch you selected is good or not.

Good patch → There should be a variation but it should not be too much.

If the difference b/w two λ values are too high then that means variation is high; bad patch.

Also $A^T A$ should be invertible for this whole thing to work!

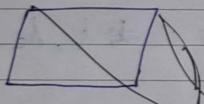
3/6/25 Tuesday.

o Jallat

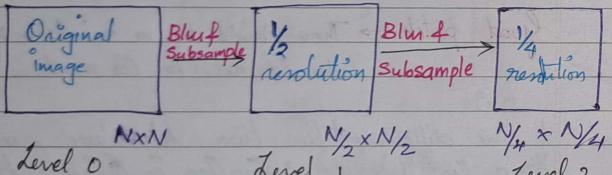
① What if optical flow blur ~~too~~ frames is too high?
→ Use Resolution pyramids.

Resolution pyramids

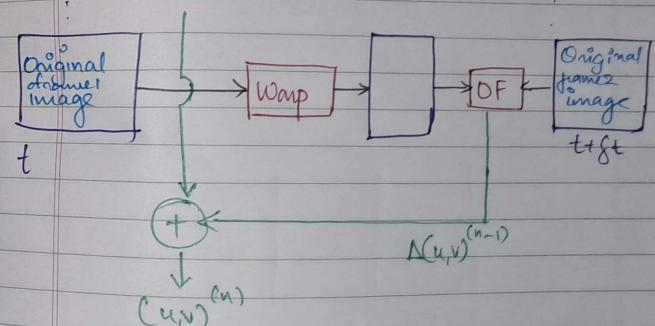
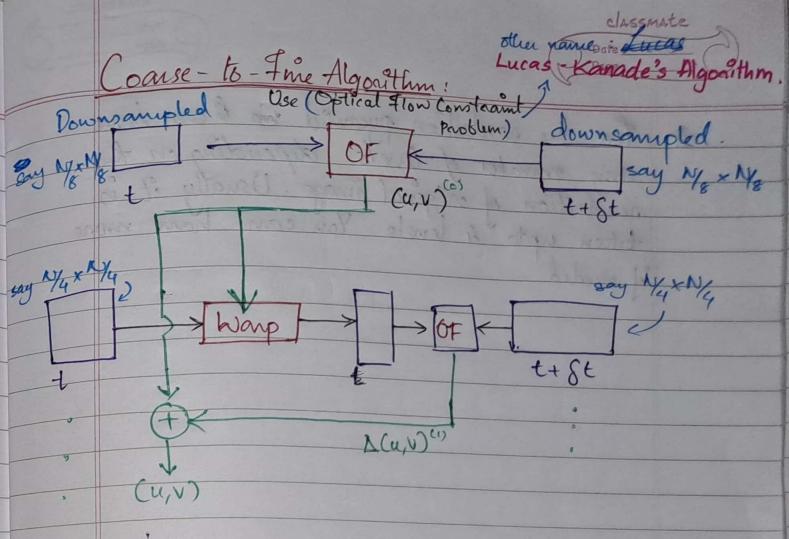
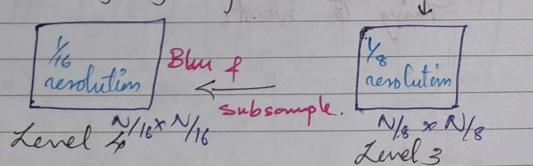
- also known as image pyramids.



- Downsampling ↴ multiple times to get the pyramid.



* Blin & Subsample \rightarrow down sampling $\xrightarrow{\text{using}}$ Pooling.



Q. What is warping?
→ Upsampling done using interpolation

- * For the resolution pyramid we can have any number of levels depending on the resolution of original image. Usually it is taken upto 4 levels. You can have more if needed.

