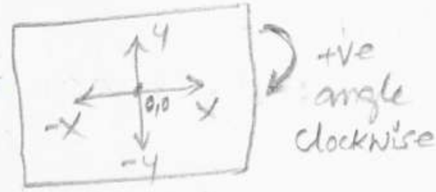
# Image Processing Labprogramming

1. I.COMET →
An Integrated IDE, program which can be used to create Image processing for Industrial products & analysis with the Imaged products.




⇒ Operators are executed by sequentially with 'control sequence' operator.

⇒ We can write code in control sequence to automate the operators.

## Lab 1a. Cables :-

Iconet Pin Board :

1. Iconet → Operators → Interfaces → Camera → avt GigE cam
2. Iconet → Operators → Image operator → Image input → Image from file (Load Image file)
3. Iconet → Operators → Image operators → Preprocessing → averaging
4. Iconet → Operators → image operators → Image changement → Image section → Rect. section V2

select portion the image

5. Iconet → Operators → Image Operators → Image changement → Color operators → color classifying
7. Iconet → Operators → mathematical operators → Counting operators → white ratio
6. Iconet → Operators → Image Operators → Image changement → color operators → color region

8. Iconet → Operators → Control operators → Control Sequence.

14/11/23

Lab - 3

Edge measurement / cylinder lengths - Lab (4)

| x | y | $\underline{x}$ | $\underline{y}$ |
|---|---|---|---|

x comp. of Unit vector ⊥r to edges

gradient unit vector +y

+y, -x, 0, +x, -y

1. control seq
2. Image from file
3. Rect. Mask V2
4. Color channel Edges V2

Binary shapes

Image from file → HMT (hit-miss SE) → structuring Element Dilation → Image multiplication

Seperate characters in Text

Image from file → Averaging → h minima → Binary fixed threshold Image → Reconstruction by Erosion → Watershed labelling → Binary fixed threshold Image → Image multiplication

Image from file → pixel format conversion → Image multiplication

Template matching

Image from file → Averaging → Down sizing → Rectangular mask V2 → figure matching

Down sizing → figure matching

23/10/23 

→ Luminous Flux $\phi_v = K_m \int_0^\infty V(\lambda) \cdot \frac{d\phi_e}{d\lambda} \cdot d\lambda$ (lm) (Lumen) → unit of luminous flux.

$V(\lambda)$ = luminosity function

$\frac{d\phi_e}{d\lambda}$ = Radiant flux density

$K_m = 683 \frac{lm}{W}$ = constant

⇒ Luminous Intensity $I_v = \frac{d\phi_v}{d\Omega}$, unit is $\frac{lm}{sr} = cd$

(∴ $\Omega$ is solid angle) [∴ sr - steradian, cd - candella]

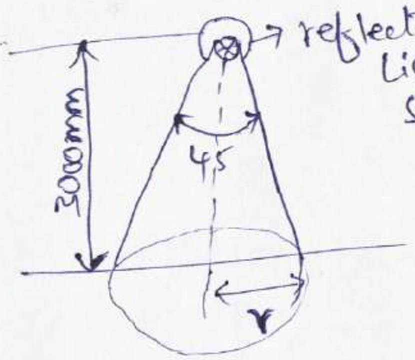
solid angle is the area in a unit sphere, i.e, sphere with radius = 1m, defined by a cone.


$$\Omega = \frac{A_t}{r^2}$$
$$= 2\pi\left[1 - \cos\left(\frac{\phi}{2}\right)\right]$$

→ Illuminance :- $E_v = \frac{d\phi_v}{dx}$, unit is lux (lx) **

Example:-

Given $I_v = 350$ cd, cal. Illuminance, $E_v$ = ?




1) $I_v = \frac{d\phi_v}{d\Omega} = \frac{\phi_v}{\Omega}$ (assumed homogeneous radiation)

2) $E_v = \frac{\phi_v}{A}$

$$\Omega = 2\pi\left(1 - \cos\left(\frac{\phi}{2}\right)\right) = 2\pi\left(1 - \cos\left(\frac{45}{2}\right)\right)$$
$$= 0.478 \text{ sr}$$

$\tan(22.5) = \frac{r}{3000mm}$

$r = 3 \cdot \tan(22.5)$

$= 1.243$ m

$A = \pi r^2 = \pi \cdot (1.243)^2$

$= 4.854 \ m^2$

$I_v = \frac{\phi_v}{\Omega} \Rightarrow \phi_v = I_v \cdot \Omega$

$= 350 \times 0.478$

$\phi_v = 167.4$ lm

$E_v = \frac{\phi_v}{A} = \frac{167.4}{4.854}$

$E_v = 34.51$ lx

⇒ Bigger lens have ~~more~~ less 'F' numbers & vice versa.

$$\text{F-number} = \frac{\text{fraction of image side focal length}}{\text{dia. of the entrance pupil}}$$

$$F = \frac{f'}{d_{ENP}}$$

30/10/23

⇒ Depth of Field :- ($\Delta s$)

⇒ Object side Telecentric Lenses :-

Objects are imaged in the same size even if they have different object distances 's'.

Depth of field, $\Delta s = \dfrac{d'}{\beta \sin\alpha} = d'$

In Telecentric lenses, object should be under lenses light.

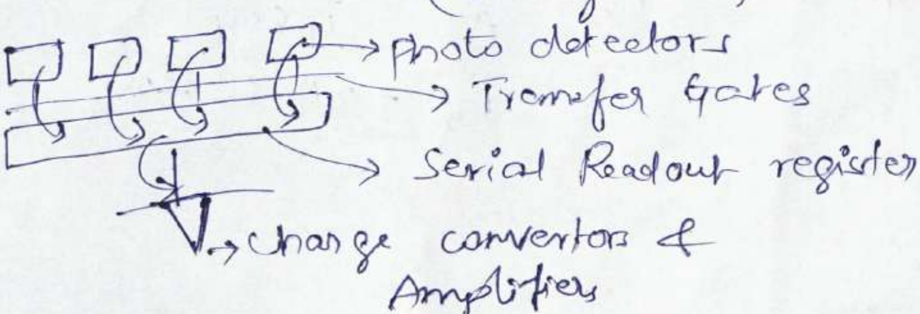
⇒ Bilateral Telecentric Lenses :-

To obtain a bilateral Telecentric lenses, a $2^{nd}$ lens is placed behind the $1^{st}$ lens in such a way that "$F_1' = F_2$".

⇒ Images with Distortions
- → w/o distortion
- → Pincushion "
- → barrel "

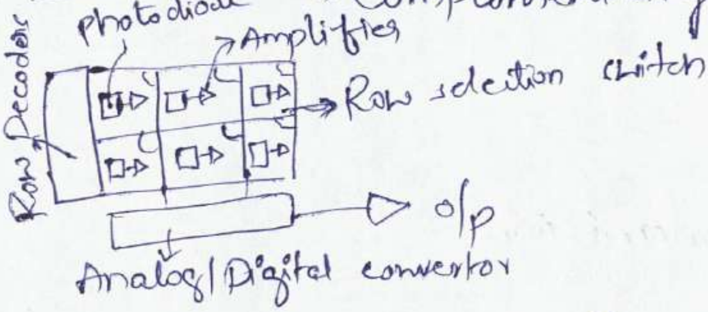
⇒ Chromatic Aberrations and Vigentting :-

⇒ Camera - Linear CCD Sensors :
(Charge Coupled Device)



fast moving object → Imagined in Interlaced mode
→ Imagined with progressive.

⇒ Camera - CMOS sensor
⇒ Complemmentary Metal Oxide Semiconductor (CMOS)

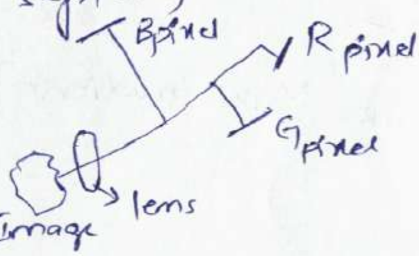



fast moving object → Imaged with a rolling shutter
→ " " " global "

6/11/23

Global shutter is good for fast moving objects

Color Camera :-
⇒ CFA (Colour filter Array) is used to get color images cheaply.
⇒ In Bayer Pixel, 'G' is double of 'R' & 'B'. (bcoz G is in wavelength of human vision system)
⇒ 3 chip cameras are available, one chip for each color, but they are expensive




⇒ for big sensor, lens is small size & vice versa.
S(pixel size)
When resolution of picture/image is high, sensor is also high.
⇒ Sensor sizes are calculated by 2/3 of diagonal of sensor (length & width). Why 2/3? bcoz in history, sensor size is 2/3 of (Vidicon video) outer tube diameter.

Camera Noise :-
1) Photon noise

$\sigma_p^2 = \mu_p$

further sources of noise: (4 NOISE Types)

- Amplifier noise
- reset noise
- dark current noise (caused by thermal excitation)

* SNR - Signature Noise Ratio.

$$SNR = \frac{Mean}{Variance} =$$

⇒ 4 Types of signal transmission

Analog Video Signals

Digital " "

- Camera Link
- IEEE 1394 (old & no longer in industrial app)
- USB 2.0 & USB 3.0 (max. wire length 3-5m)
- Gigabit Ethernet (unlimited wire length, very high speed, robust)

18/12/23

⇒ $\nabla_{max}$ (Max Gradient) $= \frac{\Delta f_{max}}{\Delta x} = \frac{255}{1} = 255$

# Image Processing for Automated Production WS 2023

- Rconnet software
- 50% written exam & 50% lab exam.

⇒ Scriptum - 1

→ Main application fields of IIP (Industrial Image Processing)

1. Object identification
2. 2D & 3D position detection
3. Completeness checking
4. Surface inspection
5. Shape & dimensional inspection.

→ Typical components of Machine Vision System:

Vedio - 2

→ Image Acquisition:

3. Light Properties

light spectrums

Visible 380-780 nm — Blue Purple, Blue, Green, Yellow, Red

- Visible & Infrared range are not dangerous for skin & eyes (even with high intensity)
- But UV rays & shorter wavelength (i.e higher frequencies) are invisible & more dangerous for skin, avoid expose to Ultravoilet ray.

$\boxed{\text{Visible} - 380 \text{ to } 780\text{nm} - \text{BBGYR RBG}}$

- Planck's law for the radiation of a black body

formula necessary not for exam

$$I(\lambda T) = \frac{2hc^2}{\lambda^5} \cdot \frac{1}{e^{hc/\lambda kT} - 1}$$

⇒ A black body is a body which appears black in all wavelength, so which doesn't emits electromagnetic radiation/light when it's cold. And also the black body absorbs all radiation which comes from environment & which hits black.

- When heats the black body, you get an intensity spectrum with different wavelength.

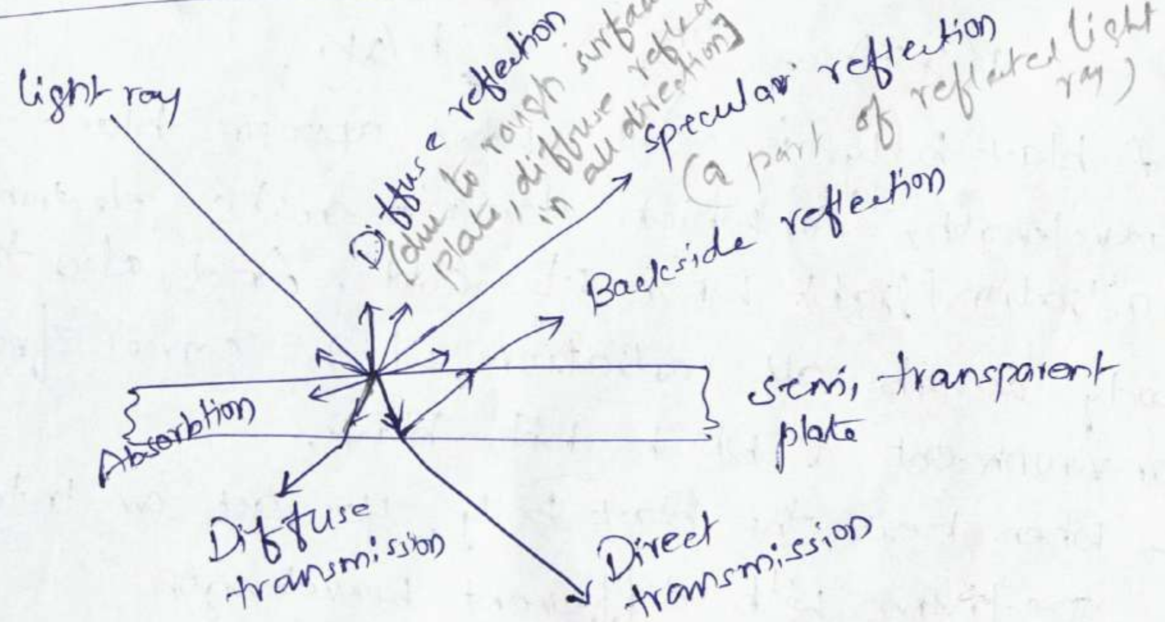
<u>Qstn:-</u> Why are room heaters painted white color?

- for invisible light like IR light, the white paint appears completely black.
- A black body has high absorbance property

Types of Light Sources Ved-4 4. ILLUMINATIONS

1) LED (Light Emitting Diode)
   - Best & most modern source. They have high energy eff. & can produce bright light. Best light system.
   - Monochromatic (red/blue).
2) Flourescents Lamps:-
   -
3) Xenon Lamps:-
   - Used for very bright flashes.
   - We'll use flashes if objects is moving (continuous process industry objects) & we need chop ~~objects~~ images from moving objects.
4) Incadescent Lamps:-
   - Out of ~~fashing~~ market i.e not using

- LED are normally monochromatic (red/blue)
- To get other color LED, need flourescent substances



$I = R + T + A$ → Absorbed light

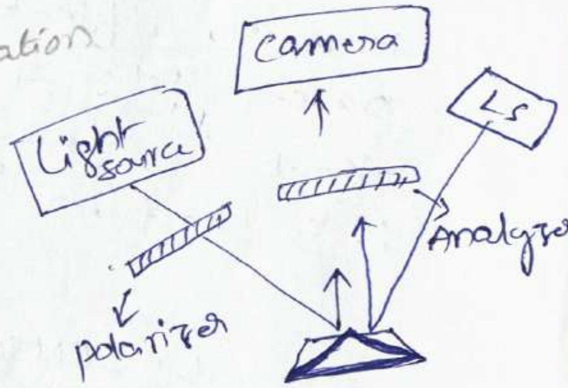
I → Incoming light
R → Reflected light
T → Transmitted light

Usig the spectral Composition of light:-

You can use section of the light spectrum by
- selecting & mixing Red-Green-Blue pixel values in color cameras (internal filter) or
- using external spectral camera filters

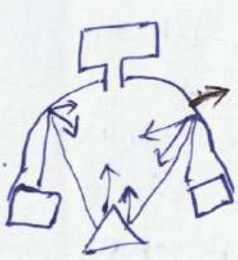
① ⇒ Polarizing filters: *

- Used this to improve illumination citeration and to highlight/emphasis somethings (like reduce reflection.....)
- Polarizing filter is a thin layer of transparent plastic part & the light passes the layer, osscilates or swinging in different direction (i.e, this polarizing layer allows one kind of oscillation/swing plane)



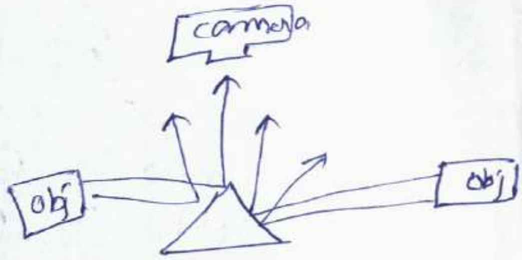

② ⇒ Diffuse light * Imp for examination

Used to reduce local reflections and shadows to enhance the visibility of this diffuse light source feature (i.e one type of illumination)
This can be acheived with a „dome" illumination.



③ ⇒ Dark field Light :- *

Edges as well as scratches can be highlighted by using this illumination type




④ ⇒ Bright field Illuminations

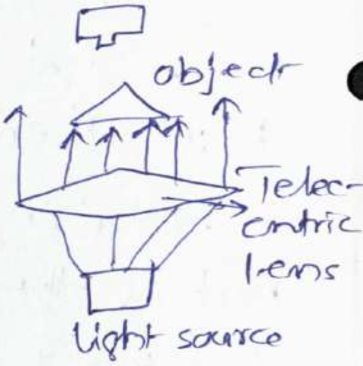
This can haighlight reflections from surfaces which are oriented ⊥r to camera axis.

⑤ ⇒ Diffuse Bright-field Backlight

- Used for sheet metal parts & sharp edges like threads in bolt


6 ⇒ Telecentric Bright-field Backlight

- To get high precision of thin sheet, sharp edges/objects, we can use Telecentric lens.



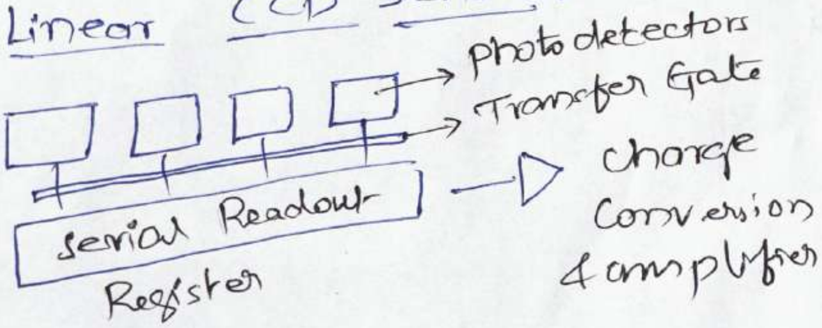

- Here, No reflection effect i.e rays passes ||el.
- object is always smaller than lens size. So it is costlier

Illuminations **
- Polarizing filters
- Diffuse light
- Dark field Light
- Bright field Illuminations
- Diffuse Bright-field Back
- Telecentric " " Backlight

5. SENSORS → array
Light sensitive in camera

- CCD sensor (Charge coupled Device)
- CMOS sensor Complementary Metal oxide semiconductor

⇒ Linear CCD sensors :-




- Line camera has linear photodetectors which converts photons to electrons.
- photodetector stores pixel of each one

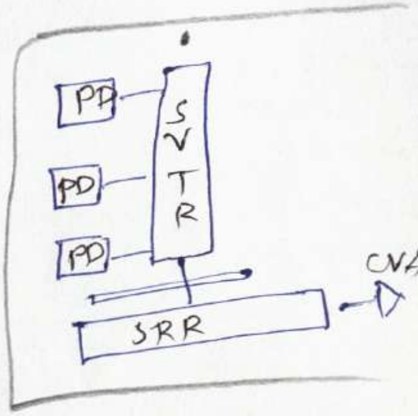
CCD sensors used in more expensive camera (or) old camera (or) industrial camera.

→ Area CCD sensors

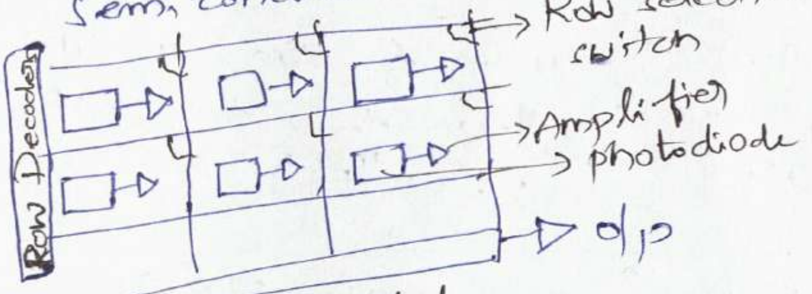
(1) photodetectors → shielded transfer register → serial Readout register → charge converter & Amplifier

(to avoid continuous storage of pixel)

(2) photo detectors → shielded Vertical Transfer Register → Serial Readout Register → Charge Converter & Amplifier




→ CMOS sensor
Complementary Metal Oxide Semi conductor




Here, Amplifier, AD convertor need more space

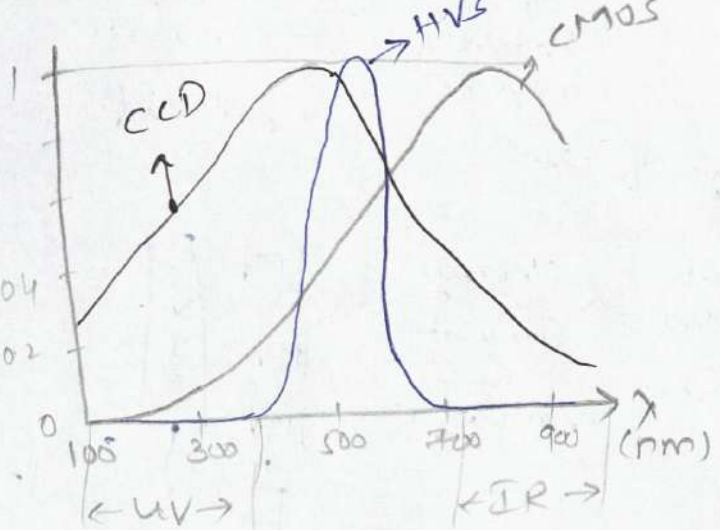
SHUTTER
→ Rolling shutter
→ Global "

→ shutter is element that determines exposure time that sensors exposed to incoming light

- Global shutter is prefered for fast moving object
- Rolling shutter is rolls over image & cause deformation of image whereas Global shutter causes blurring because capture all pixels but not deform the object.

| ** CCD | ** CMOS |
| --- | --- |
| - High fill factor<br>- High full well capacity<br>- Commonly large pixels (12 or 14 µm) with high sensitivity & dynamic range<br>- for moving object, Mechanical or flashlight shutter are necessary to avoid smearing | - each pixel is addressable<br>- high frame rates<br>- User-specific ROI (Region of Interest)<br>- global & Rolling shutter<br>- less full well capacity<br>- less uniformity in pixel sensitivity |

⇒ <u>Color</u> <u>Cameras</u> :- **




- CMOS have high sensitivity in IR region
- CCD has high sensitivity in blue (UV) region & HVS
- HVS is more senstive for Green colour, so

| | | | | | |
|---|---|---|---|---|---|
| R | G | R | G | R | G |
| G | B | G | B | G | B |
| R | G | R | G | R | G |
| G | B | G | B | G | B |

CFA

⇒ CFA (colour filter Array) Bayer

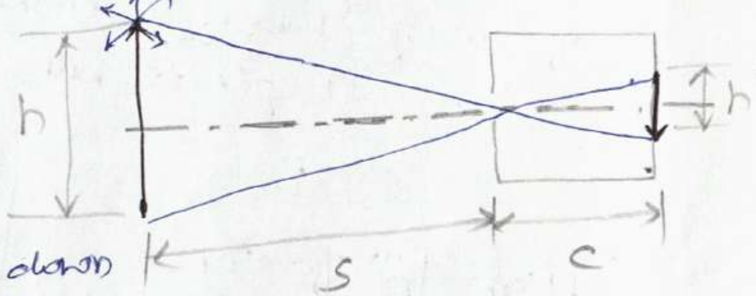
- Pixel values for each color channel (R G B) have to be interpolated by bilinear or bicubical interpolation algorithm
- more 'G' pixels in CFA filter (twice the R & B, [2G = R or B])
- In industry applications, grey scale camera is used because in color cameras, the resolution or accuracy due to interpolation of color pixels.

⇒ Lens must be larger than sensor (otherwise dark edges are captured)

## 6. Gaussian Optics

1) <u>Pinhole</u> <u>Cameras</u> :

$$\Rightarrow \quad h' = h \frac{c}{s}$$




- Object is imaged upside down on the pinhole box backplane
- Only few light passes the pinhole & meets the light sensitive film. So, very long exposure time needed to snap an image

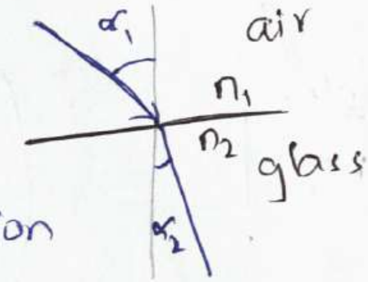
2) <u>Gaussian optics</u> :-

$$n = \frac{c}{V}$$

Refractive index, $n = \dfrac{\text{vaccum light speed}}{\text{medium light speed}}$

for air, $n = 1.00029 \simeq 1$

for glass, $n \simeq 1.45$ normal glass to 2.14




$\Rightarrow$ Principle of refraction :

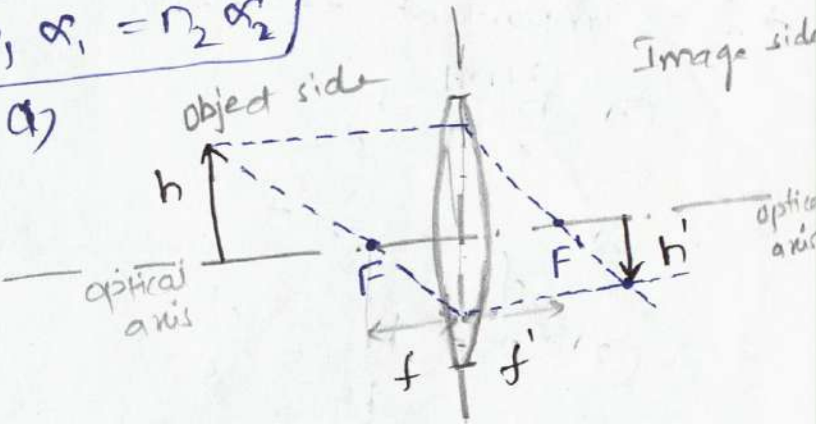
$$n_1 \sin\alpha_1 = n_2 \sin\alpha_2$$

Linearized principle of refraction by assuming small angles leads to Gaussian optics

i.e $\boxed{n_1 \alpha_1 = n_2 \alpha_2}$

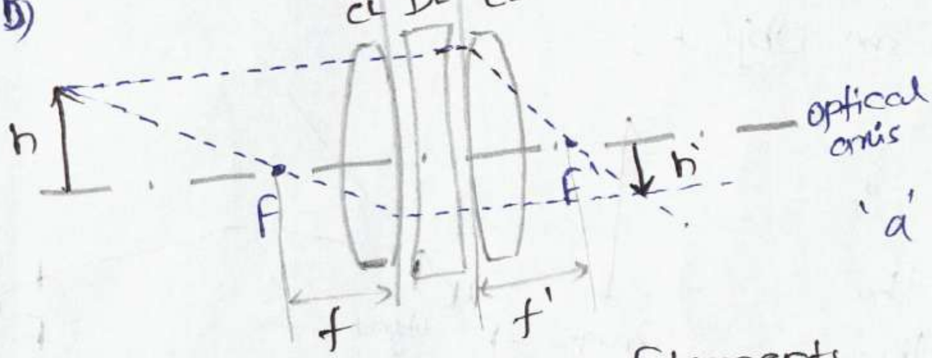
i) Condensing lens ii) Diverging lens

a)



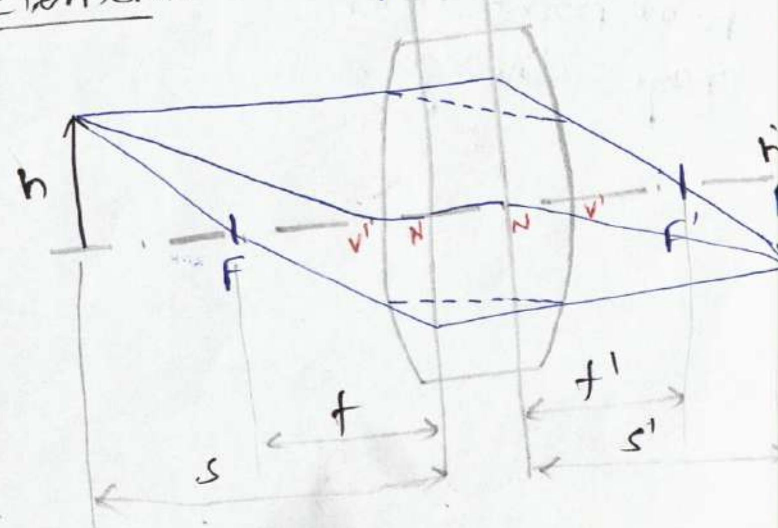

P lens
Lens

b)




'a' & 'b' are Gaussian optics concepts.

3) <u>Thick ~~ed~~ Lens</u> <u>Cardinal Elements</u>




i) $\dfrac{h}{s-f} = \dfrac{h'}{f} \rightarrow ①$

2) $\dfrac{h'}{s'-f'} = \dfrac{h}{f'} \rightarrow ②$

setting $f = f' \Rightarrow \dfrac{h}{s} = \dfrac{h'}{s'}$

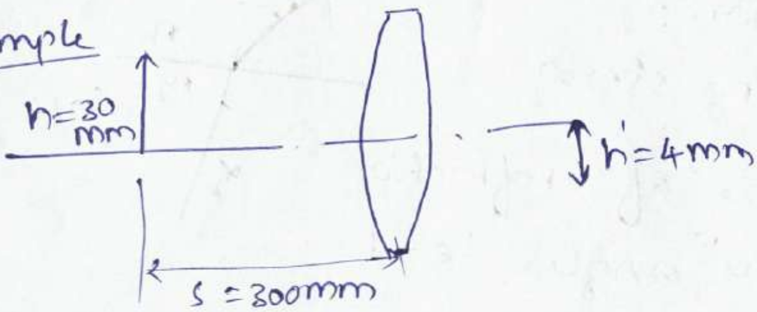
① ⇒ $hf = h'(s-f) \Rightarrow (h+h')f = h's$

② ⇒ $h'f = hs' - hf \Rightarrow (h+h')f = hs'$

$\therefore h's = hs' \Rightarrow$ ① $\frac{h}{s} = \frac{h'}{s'}$ ② $\frac{1}{f} = \frac{1}{s'} + \frac{1}{s}$

3) $\frac{h'}{h} = \frac{s'}{s} = \beta$ = magnification factor

<u>Example</u>




What is the focal length of lens?

Sol: $\frac{h}{s} = \frac{h'}{s'}$

$s' = \frac{h'}{h} \times s = \frac{4}{\cancel{30}} \times \overset{10}{\cancel{300}}$

$s' = 40$

$\frac{1}{f} = \frac{1}{s} + \frac{1}{s'} = \frac{1}{300} + \frac{1}{40}$

$= \frac{34\cancel{0}}{(300)(4\cancel{0})}$

$f = \frac{1200}{34} = 35.29\,mm$

** 3 Important equation

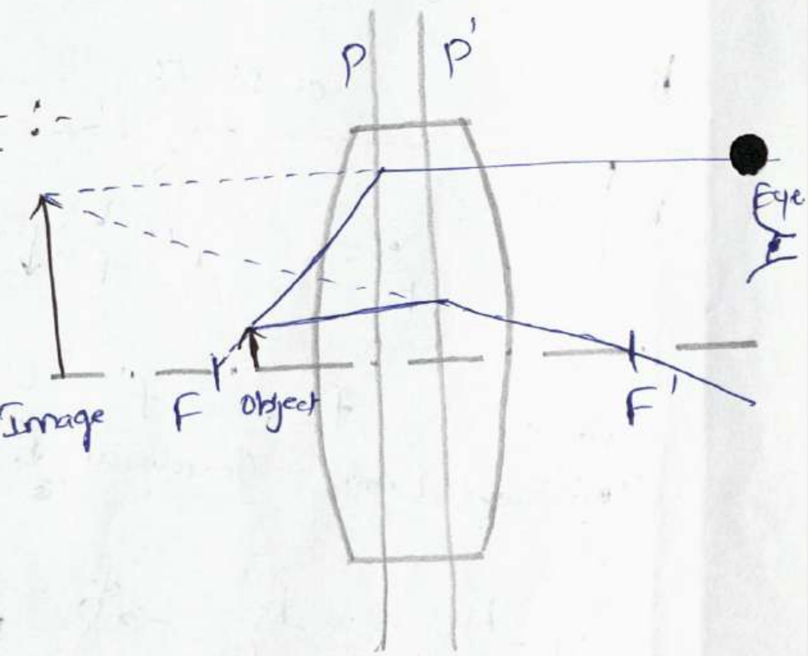
① $\frac{h}{s} = \frac{h'}{s'}$

② $\frac{1}{f} = \frac{1}{s'} + \frac{1}{s}$

③ $\beta = \frac{h'}{h} = \frac{s'}{s}$

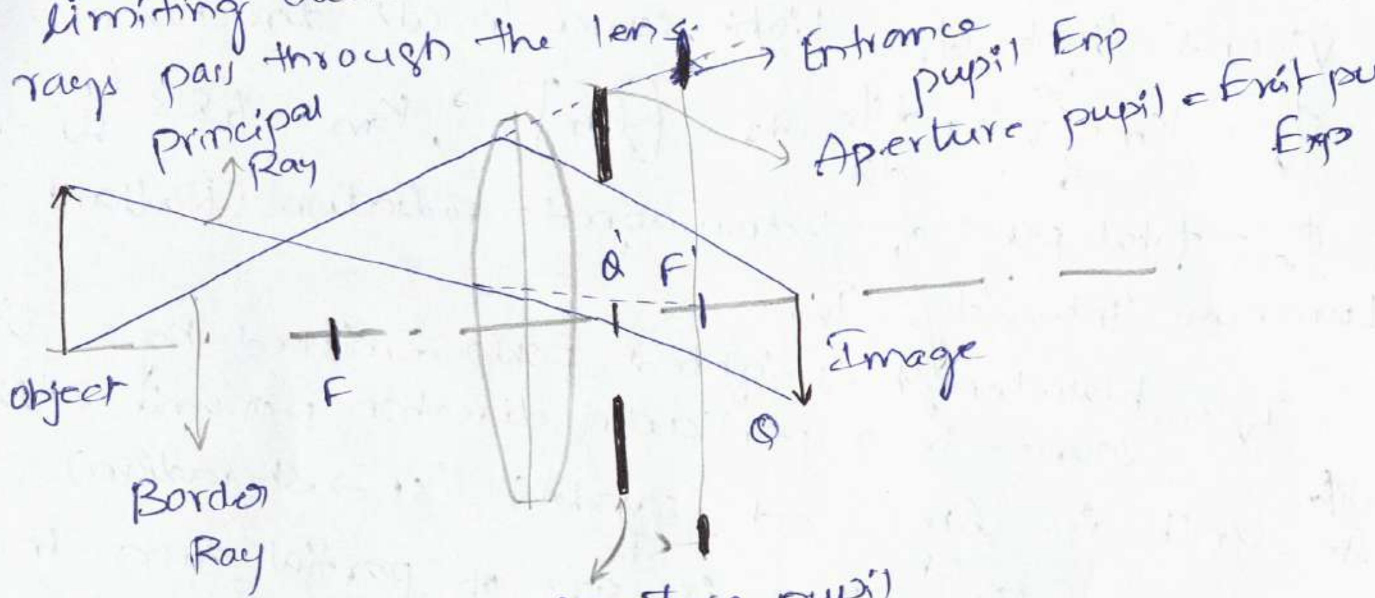
④ <u>Virtual image of an Object</u> :-

If the object is positioned behind the focal point, then the rays diverge.



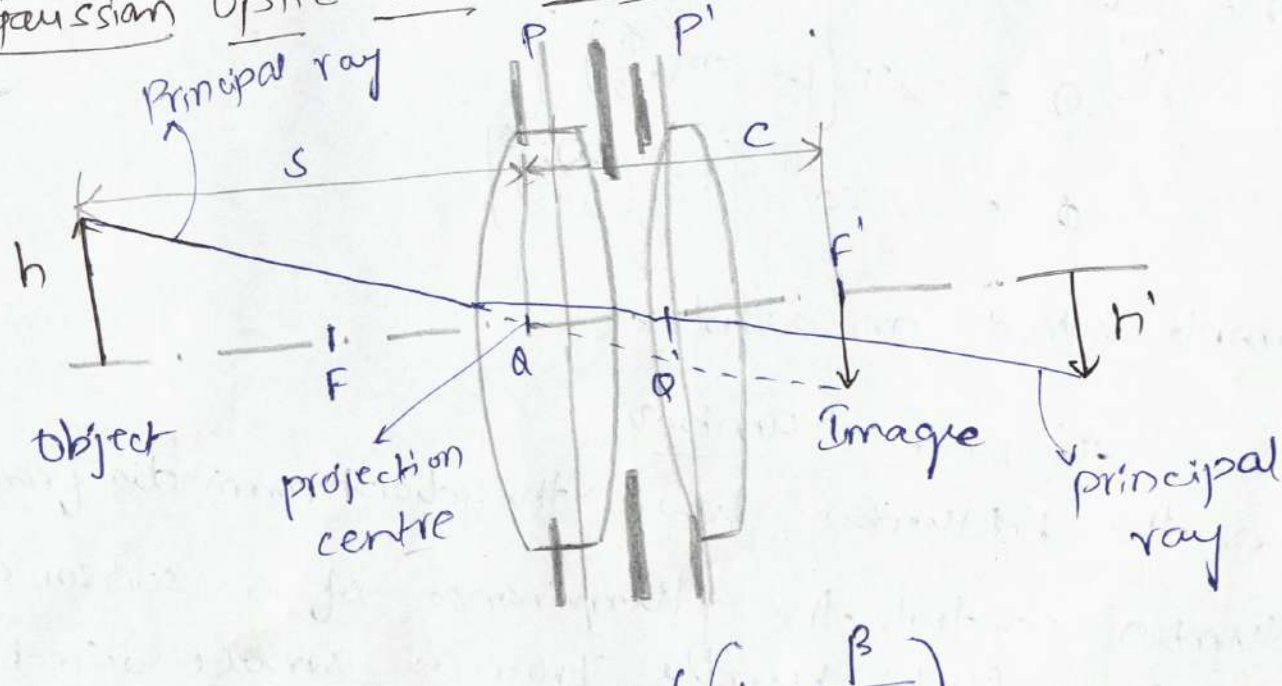
## pupils in a Lens Systems :-

limiting elements (lemit) the rays or bundles of rays pass through the lens.




- (Imp is) *** What is Entrance pupil and Exit pupil ?

## Gaussian Optic and the Pinhole Camera Model




$$c = \frac{h'}{h} \cdot s = \beta s = f\left(1 - \frac{\beta}{\beta_r}\right)$$

# 7 - Illumination Calculations

1) <u>Luminous</u> <u>flux</u>:- Measure of total perceived power of visual light of a light source in all directions

$$\phi_v = K_m \cdot \int_0^\infty V(\lambda)\, \frac{d\phi_e}{d\lambda} \cdot d\lambda \quad [lm] \quad , K_m = 683 \frac{lm}{W}$$

$\phi_e$ = total power of electromagnetic radiation (Radiant flux)

2) Luminous Intensity: $I_v$ :-

$I_v$ = Wavelength-weighted power emitted by a light source in a particular direction per unit solid angle

$I_v = \frac{d\phi_v}{d\Omega}$ Unit is $\frac{lm}{sr}$ = cd (candela), (sr → steradian)

($\Omega$) solid angle [steradian] is size of partial area in a unit sphere

3) Illuminance [$E_v$]: Total wavelength-weighted luminous flux incident on a surface per unit area

$$E_v = \frac{d\phi_v}{dA} \; (lx) \rightarrow \text{lux unit}$$

4) Solid angle ($\Omega$); $\Omega$ is the area in an unit sphere.

$$\Omega = \frac{A_t}{r^2}$$


$$\Omega = 2\pi \left[1 - \cos\left(\frac{\phi}{2}\right)\right]$$

$A_t =$

$$\phi = 2\cos^{-1}\left[1 - \left(\frac{\Omega}{2\pi}\right)\right]$$

* Example solved on 23/10/23

** 8. <u>F-Number</u>

⇒ Higher the F-Number, lower the (diaphragm) diaframe.

F-Number control, the illuminance of a sensor area, caused by a light bundle from a small object area is light energy per time and per sensor area

$$\therefore \text{Illuminance} \propto \frac{\text{area of the entry pupil}}{\text{square root of image side of focal length}}$$

ie $E_v \propto t \cdot A$

Illuminance, $E_V \propto \underset{(time)}{t} \cdot \left(\frac{d_{ENP}}{f'}\right)^2$ → dia of entry pupil (11)

** For F-Number $(F) = \frac{f'}{d_{ENP}} = \frac{\text{Image side focal length}}{\text{dia. of entrance pupil}}$

** $E_V \propto \frac{t}{F^2}$  $[\because E \propto t \cdot A]$

f-numbers are usually specified as the standard series as power of $\sqrt{(2)^n}$ namely : f/1.4, f/2, f/2.8, f/4

** How $\sqrt{2^n}$?

$\therefore \boxed{\frac{E_V}{2^n} = \frac{1}{F^2} \Rightarrow F^2 = 2^n \Rightarrow F = \sqrt{2^n}}$

| n | F |
|---|---|
| 1 | 1.4 |
| 2 | 2 |
| 3 | 2.8 |
| 4 | 4 |

→ f/2.8 can be represented as 1:2.8 means f-number of 2.8!

## 9. Depth of Field

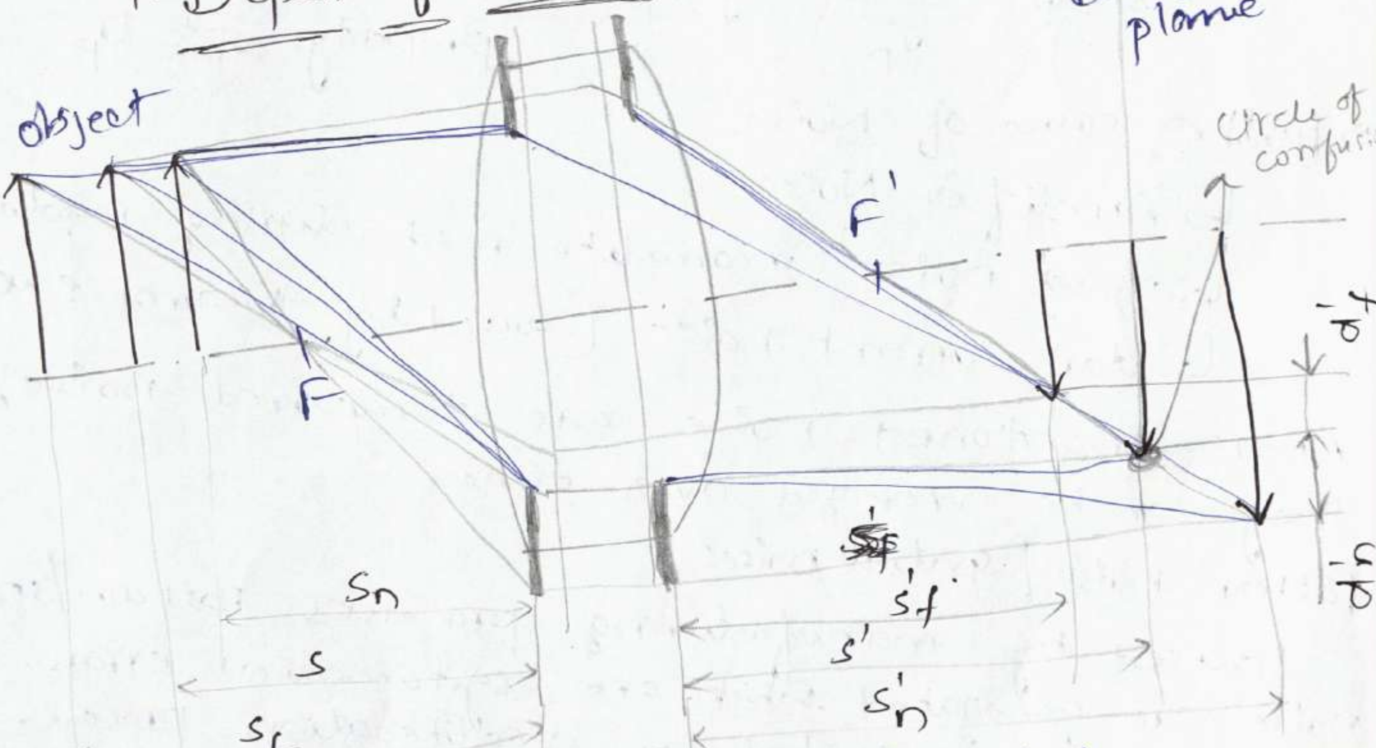



Fig: Imaging of different spaced object

** Depth of Field, $(\Delta s) = \frac{2s^2 F d'}{f'^2}$

$d'$ – Diameter of (circle of confusion) one pixel

$s_{fn}$ ? $s_f - s_n$ ⇒ Depth of Field

## 10. Camera-Noise

⇒ Camera noise, EMVA Standard 1288, char. parameters

1) Total quantum efficiency $\eta$ = ratio of electrons created per photon

2) Overall system gain $K = \frac{\text{gray value}}{\text{no. of electrons}}$

A no. of photons $n_p$ created no. of electrons $n_e$ & holes (CD) resp annihilates (CMOS) electron-hole-pairs

⇒ ∴ Photon Noise is Poisson distributed

$\sigma_p^2$ is variance,

$\mu_p$ is mean no. of photons

$\mu_e$ is " " " electrons

$$\boxed{\begin{aligned} \sigma_p^2 &= \mu_p \\ n_e &= \eta n_p \\ \mu_e &= \eta \mu_p \\ \sigma_e^2 &= \mu_e = \eta\mu_p = \eta\sigma_p^2 \\ \therefore \sigma_e^2 &= \eta\sigma_p^2 \end{aligned}}$$

⇒ $SNR_p$ (signal to Noise Ratio):

$$SNR_p = \frac{\mu_p}{\sigma_p} = \sqrt{\mu_p}$$ , with Poission distributed property $\sigma_p^2 = \mu_p$

⇒ Further source of Noise:

- Amplifier Noise
- reset noise (incomplete reset during readout)
- dark current noise (caused by thermal excitation)

Above mentioned noise are temporal noise, bco they can be averaged over time

⇒ Pattern noise / spatial noise:-

Caused by manufacturing processes, pattern/spatial noise are generated, which are systematical errors. They can be eliminated by a caliberation process.

Pattern noise subdivided into 2 components

- Offset noise, fixed pattern noise, dark signal non uniformity (DSNU) ⇒ $\sigma_0^2$
- Gain noise, photo response non-Uniformity (PSNU)

$$\sigma_g^2 = S_g^2 \cdot \eta^2 \cdot \mu_p^2$$

$s_g^2 \rightarrow$ variance co-eff of the gain noise

** CMOS sensors typically have a larger spatial noise due to the fact that each pixel has its own amplifier.

$\therefore$ Total noise $= \sigma_y^2 = K^2(\eta \mu_p + \sigma_d^2 + s_g^2 \eta^2 \mu_p^2 + \sigma_0^2)$

$SNR_{DB}$ (SNR is specified in DB) = 20 log SNR

$\Rightarrow$ The illumination for which SNR=1,

$$\mu_{p,min} = \frac{\sigma_d}{\eta}$$

$\Rightarrow$ Dynamic range of camera } DYN $= \frac{\mu_{p,sat}}{\mu_{p,min}}$ $\frac{\text{(saturated no. of photons)}}{\text{min. no. of photons}}$

CCD sensors have higher/better DYN & a better sensitivity for light than CMOS.

## 11. Data Transmission

Camera - Computer Interfaces:

- Analog video signals (quality of data is not good) due to noise in surroundings
- Digital " " (1)
  - Camera Link
  - IEEE1394 (firewire)
  - USB 2.0
  - USB 3.0
  - Gigabit Ethernet

$\left.\begin{matrix}\text{Camera Link} \\ \text{IEEE1394 (firewire)} \\ \text{USB 2.0} \\ \text{USB 3.0}\end{matrix}\right\}$ — not industrial usage

## 12. Telecentric Lenses

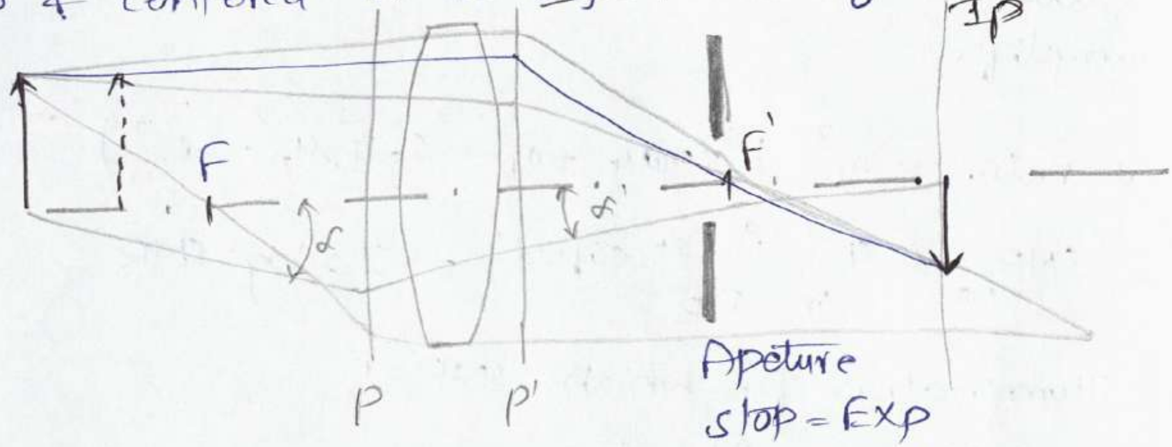
Standard lenses are homocentric lenses

In homocentric lenses, when we are placing/moving object closer to lens, the image will become larger (as the 'f' distance decreases).

This is not possible in Telecentric lenses.

$\Rightarrow$ The telecentric lens keeps the image of an object in constant size when the object move near/far to/from the lens.

So, Telecentric lens keeps the imaged object in constant size, with the help of small apeture stop & centered in the ~~object~~ Image side focal ~~length~~ point $f'$.

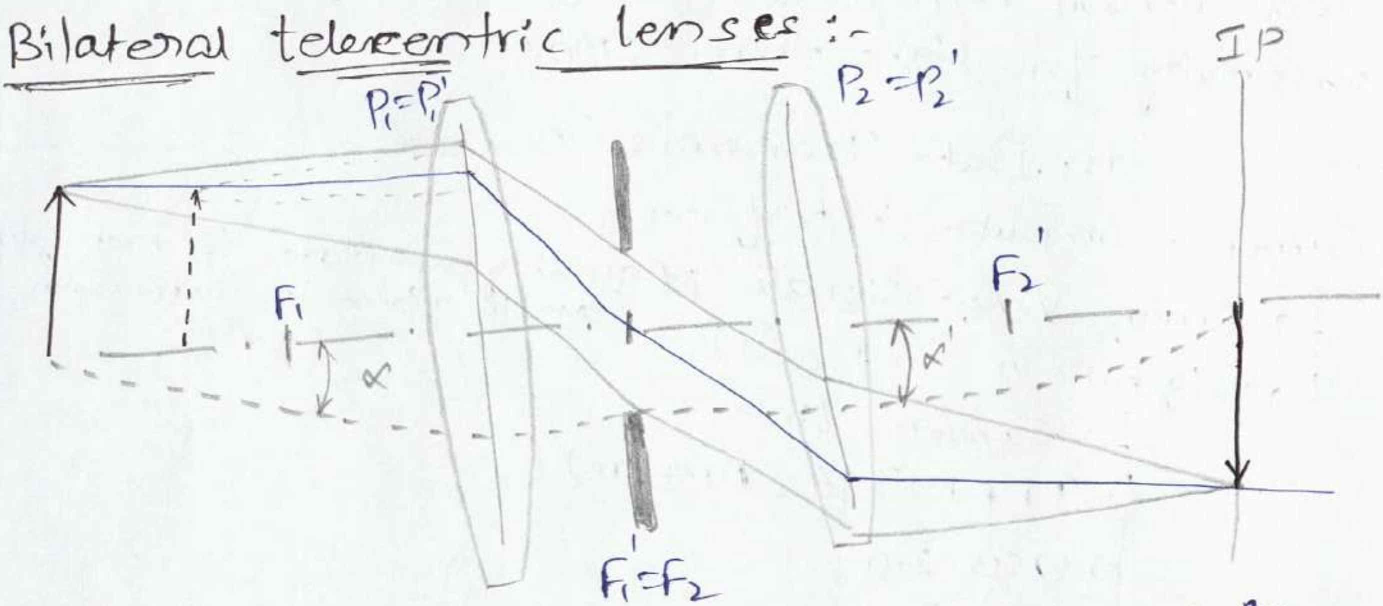



depth of field $(\Delta s) = \frac{-d'}{\beta \sin\alpha} = \frac{d'}{\beta^2 \sin\alpha'}$, $\beta = \frac{-\sin\alpha}{\sin\alpha'}$

is given by,

With $\alpha$, $\alpha'$ are the apeture angle.

⇒ Bilateral telecentric lenses :-




$$\Delta s = \frac{d'}{\beta \sin\alpha}$$

⇒ These lenses are used in measurement taske like screw dia, small objects, threads etc,

*⇒ Object may not be larger than lens

⇒ So, telecentric lenses, all lenses must be 20% larger than the object size

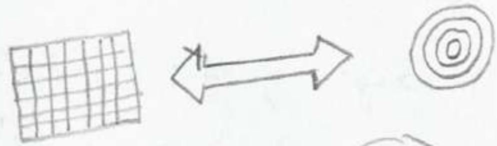
In Telecentric lenses, the lens must be (20%) larger than height of object

⇒ Spherical Abberations : They are caused by spherical shape of lenses.

⇒ Astigmatism: Tangential & sagittal light rays do not intersect at the same point.

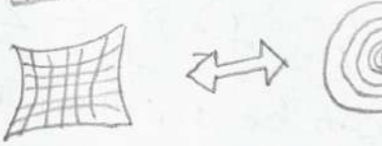
⇒ Images with Distorns:-

① Without distortion



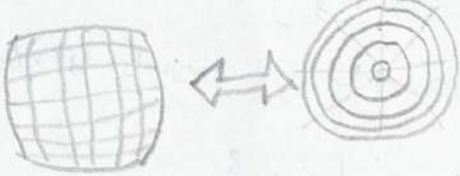
circles having same distance b/w them

② Pincushion distortion



inner circles gaps are less compared to outside

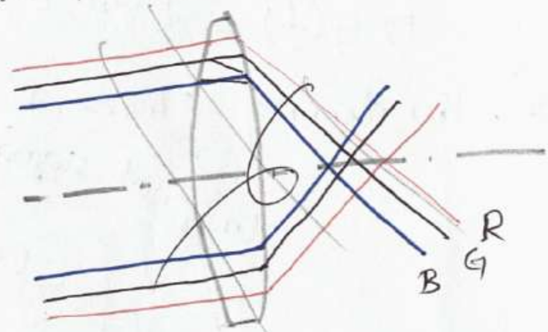
③ Barrel distortion.

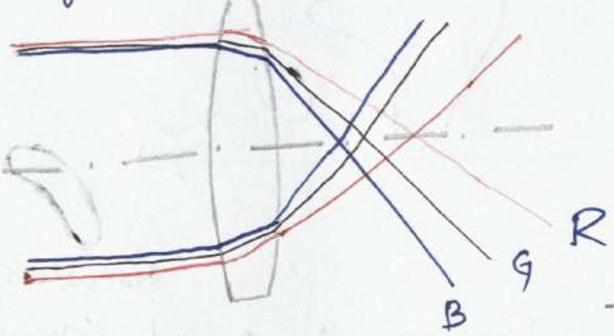


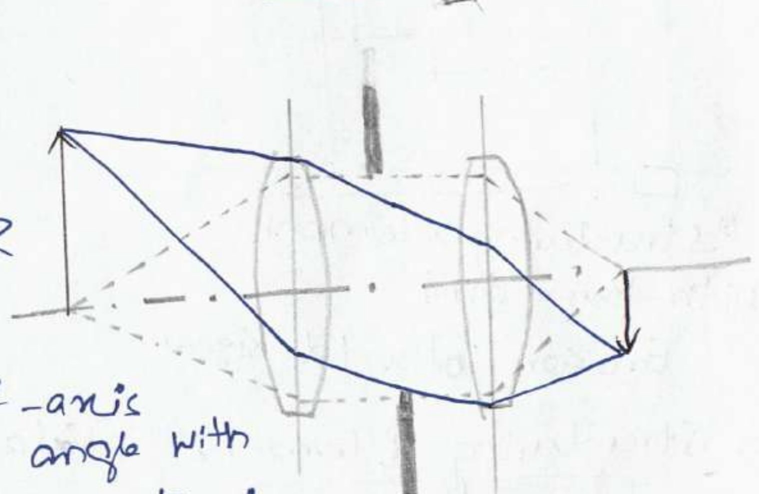
inner circle gaps are more (radius ↓ when moving →) & distance reduces when moving outside

⇒ Different diffraction for different color

① Chromatic abberrations:-
chromatic aberrations are caused by different bending angles of different wavelengths



② Vignetting:- For off-axis rays which form a large angle with the optical axis, the aperture stop is no longer the limiting element. Instead, the lens extension are limiting the amount of incoming light. Therefore, for those rays the free transit section is smaller and the transmitted energy is less. The object appears darker. This effect is called Vignetting

The angle of light cone from top of object is less compared to the angle of light cone from the axis point (foot of object). so cone is smaller & it collects few light energy &

this causes the object points far from axis & darker. (15)

## Chapter (3) 13. MORPHOLOGY

⇒ book

⇒ Morphology is the theory of pixel quantities & forms.

1) Binary image , $f: D_f \in Z^n \to \{0, 1\}$

2) gray image , $f: D_f \in Z^n \to \{0, 1, \ldots, t_{max}\}$

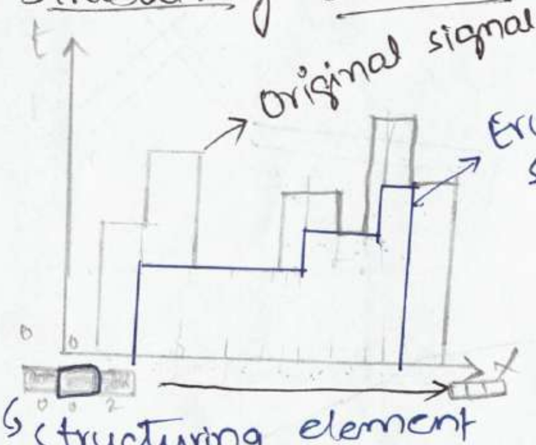
3) An image can be written as graph,

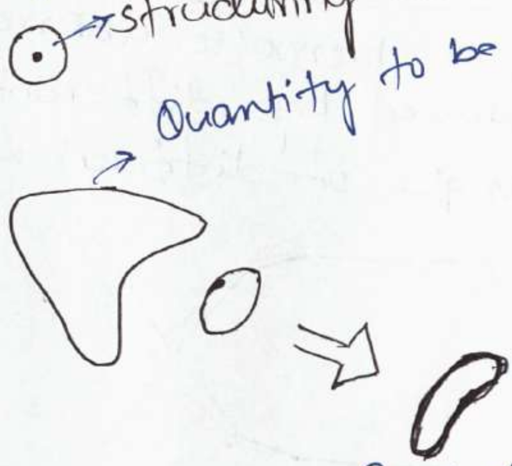
$$G(f) = \{(x, t) \in Z^n \times N_0 \mid t = f(x)\}$$

4) An under-graph is given as the quantity of points under the graph

$$UG(f) = \{(x, t) \in Z^n \times N_0 \mid 0 \le t \le f(x)\}$$

** → Structuring Element Erosion :- at consider minimum value

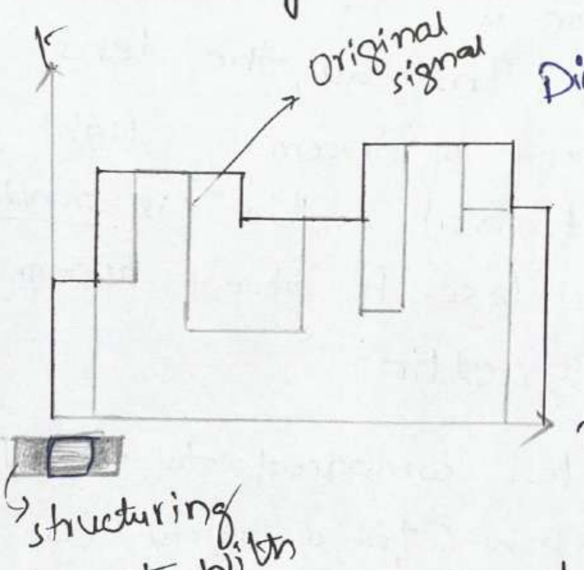


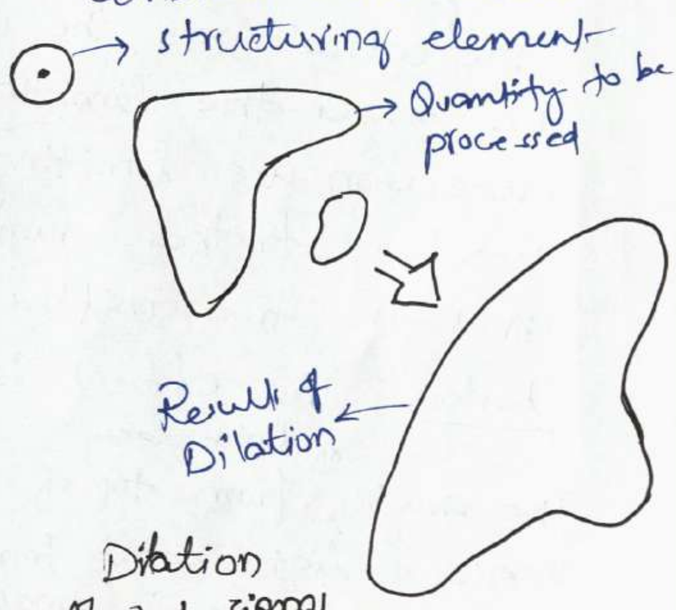




Erosion of a 1d signal

Erosion of a 2d signal

→ Structuring Element Dialation : Structuring element consider max value.




Dilation of 1D signal

Dilation of 2d signal

⇒ Erosion $\varepsilon_B(f)$ : The erosion is the point by point minimum of displacement of $f$ by a vector b!

$$[\varepsilon_B(f)](x) = \min_{b \in B_V}\{f(x+b) \cdot B_V(b)\}$$

⇒ Dilation $\delta_B(f)$ : It is the point by point maximum of the displacement of $f$ by a vector b

⇒ Gradients:- $\rho(f) = \delta_B(f) - \varepsilon_B(f)$

* The erosion and dilation operators are the two basic operators in image morphology technique. They can be combined in a various manner to fulfill image analysis tasks.

One example is the extraction of gradients by subtracting an erroded image from a dilated image.

i.e, $\rho(f) = \delta_B(f) - \varepsilon_B(f)$

14. open-close-with-asm

⇒ Internal and External Gradient

* Gradients are local changes in grayscale of an image.

Gradients can be determined by three combination

- Beucher Gradient ; $\rho_B = \delta_B - \varepsilon_B$
- Internal half Gradient ; $\rho_B^- = id - \varepsilon_B$
  Arithmetic difference b/w original image & eroded image
- External half Gradient ; $\rho_B^+ = \delta_B - id$
  $\rho_B^+$ = Dilated image − Original image

* ⇒ Morphologic Opening

original image → Erosion → Dilation (it is called opening)

To avoid noises in the original image, first apply the erosion ~~with~~ of structuring element & do the dilation of eroded image with same structuring element

$$\gamma_B(f) = \delta_B[\varepsilon_B(f)]$$

Note:- The main shape in the opened image has the same size in the original image

*⇒ <u>Morphologic Closing</u>:-

Original image → Dilation of Image → Erosion of dilated Image

Here, the original image is structured Element Dilation and then the dilated image is again do Erosion. It is called Morphologic closing

$$\phi_B(f) = \varepsilon_B[\delta_B(f)]$$

*⇒ <u>White Top Hat</u> (WTH):-

$$WTH(f) = f - \cancel{\gamma \beta \|}\ \gamma(f)$$

WTH = original image - opening operated image

With large structured Element, the noise at edges are more and corner/edges appears darker. To overcome this (avoid threshold), first opening operation of original image and do the difference from original image. i.e called White Top Hat $(= f - \gamma(f))$

⇒ <u>Thick Gradients</u>:- Thick gradients are given in case of blurred edges or smooth transitions from an object to its background. For the detection of thick gradients, we need large structuring element, but result thick gradient has width of more than 1 pixels.

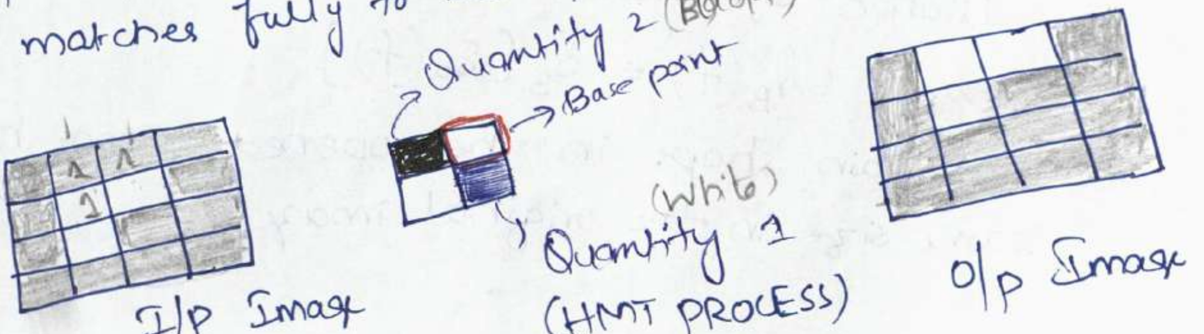
15. HMT_ Thinning - Debearding

<u>Hit</u> or <u>Miss</u> <u>Transform</u>:-

$$HMT_B(X) = \varepsilon_{B1}(X) \cap \varepsilon_{B2}(X^c)$$

**

A composed structuring element consists of 2 quantities Quantity 1 matching with foreground & the complementary quantity 2 matching with background. The structuring element is applied to binary input image pixel by pixel. The output pixel is set to high if the structuring element matches fully to the input image in its current position



* ⇒ **Thinning** :- $X \circ B = X \setminus HMT_B(X)$ ⇒ Original image w/o $HMT_B(X)$

For a binary image, thinning is the difference of an image X minus the hit-or-miss transformed image.

⇒ A homotopic thinning is realized by a sequential thinning, using two structural elements in four 90° rotation.

Original Image → thinning 1 (0°) → (Mask) thinning 1 (90°) → thinning 1 (180°) → thinning 1 (270°) → (Mask) thinning 2 (0°) → thinning 2 (90°) → thinning 2 (180°) → thinning 2 (270°) → skeleton Image

Sequential thinning to get skeleton Image

● ⇒ **Pruning & De-Bearding** :-

$$\text{Pruning} = (X \circ B)^n$$

$$\text{Debearding} = (X \circ B)^n$$

Sequential thinning for Pruning and de-bearding

→ Thinning is apply (4 or 8) thinning mask to remove pixels from both sides (top, bottom, 45°) by one pixel

→ skeletonizing is applying thinning continuously until the output image become stability. (The same procedure for pruning & de-bearding.

●

V-16 Geodesic - Erosion - Dilation

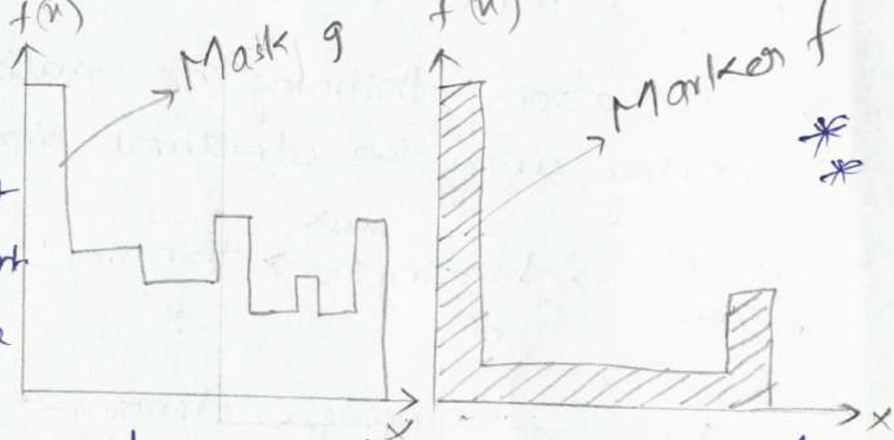
⇒ **Geodesic Transformations** :

Marker Image, Mask Image → Geodesic transformation → Output Image

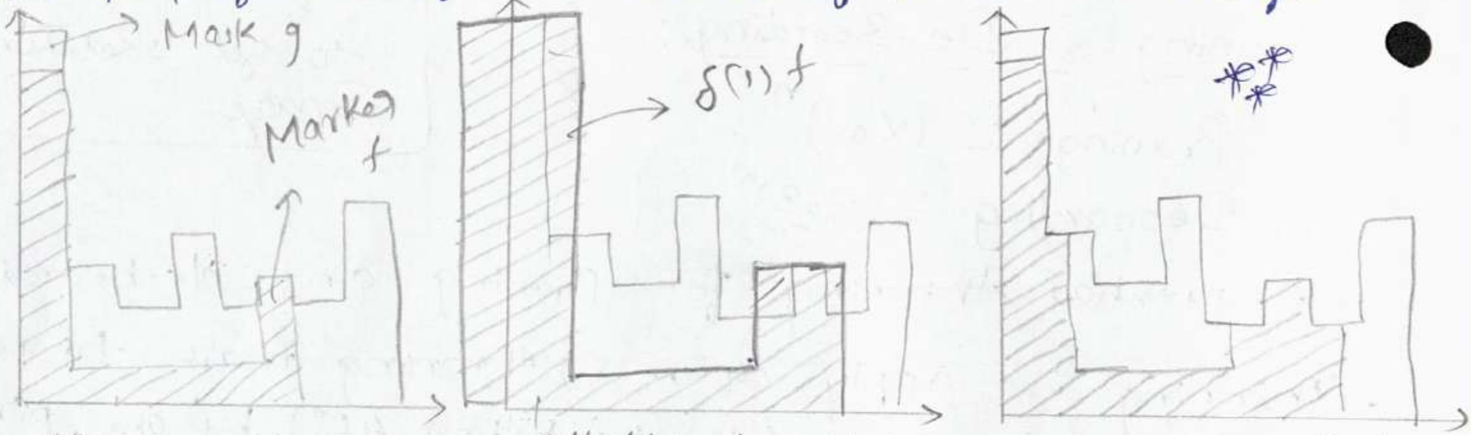
Geodesic transformations use two images. A first marker image to which the morphologic transformation is applied to, and a second image acts as a mask in a way that resulting image is forced to remain either above or below the mask image.

Marker & mask images can be binary as well as gray scale images. Mask image size $\geq$ (larger / same as) size of marker image

⇒ <u>Geodesic</u> <u>Dilation</u> :-




The marker image is first dilated by the element isotropic structuring element. The resulted dilated image is then forced to remain below the mask image. Hence, the mask image acts as a limiter to the propogation of the dilation of the marker image.




Marker over mask

Dilation of marker w.r.t

Marker can't be higher than mask. So remove dilated marker

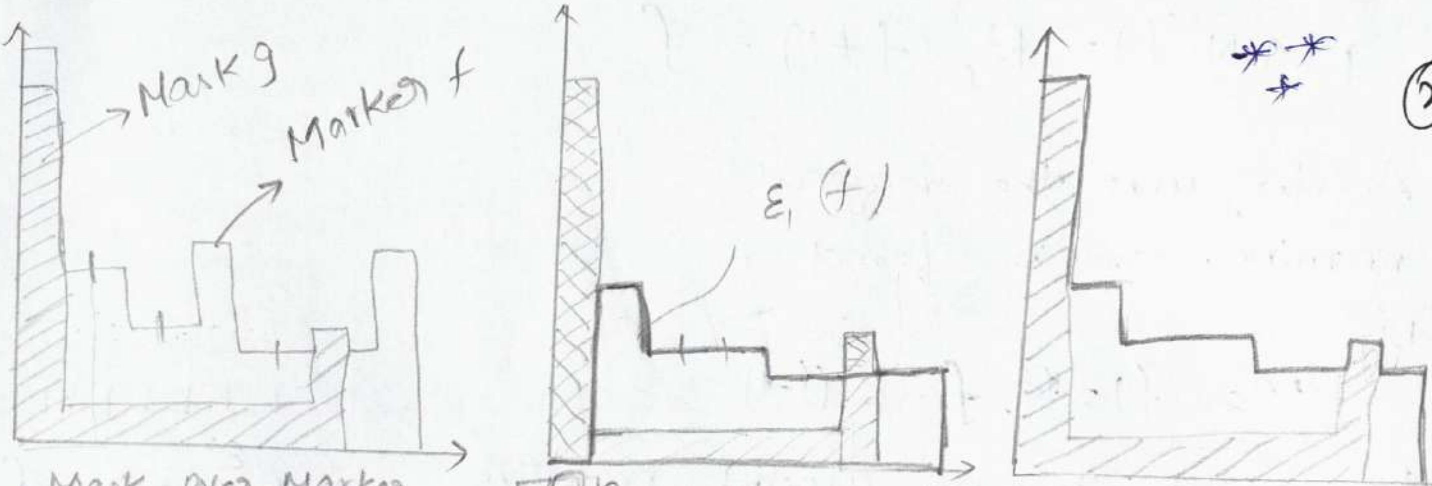
Let $f$ denotes the marker image and $g$ denotes the mask image. $\underline{D_f = D_g \text{ and } f \leq g}$.

Geodesic dilation is represented by $\delta_g^{(1)} \cdot f$

⇒ Geodesic dilation of size $n$ of a marker image $f$ w.r.t a mask image '$g$' is computed by performing $n$ steps

$$\delta_g^{n}(f) = \delta_g^{(1)}\left[\delta_g^{n-1}(f)\right]$$

⇒ <u>Geodesic</u> <u>Erosion</u> :- Geodesic erosion is the dual complementary transformation of geodesic dilation. The marker image image is first eroded by elementary isotropic structuring element. The resulted eroded image is then forced to remain above the mask image. There fore the mask image acts as a minimum operator to the propogation of erosion of marker image




Mask over Marker Image

Erosion of marker with structuring element

pull pixel value of marker to the mask

$f$ denotes the marker image and

$g$ denotes the mask image, $D_f = D_g$ and $f \geq g$

- $\varepsilon_g^{n}(f) = \varepsilon_g^{(1)}\left[\varepsilon_g^{n-1}(f)\right]$

V17 Geodesic Reconstruction.

⇒ Reconstruction by Dilation :-

Reconstruction by Dilation means continue the geodesic dilation process until the stability. It is also known as Morphological reconstructions.

Morphological reconstructions by dilation is denoted by

$$R_g^{\delta}(f) = \delta_g^{i}(f) \Rightarrow \delta_g^{i}(f) = \delta_g^{i+1}(f)$$

→ • Reconstruction by Erosion:-

- Continue the geodesic erosion process until the stability

$$R_g^{\varepsilon}(f) = \varepsilon_g^{i}(f) \Rightarrow \varepsilon_g^{i}(f) = \varepsilon_g^{i+1}(f)$$

V18 : Geodesic_Min_Max_Imposition

⇒ Regional Minima:-

The morphological reconstruction operators can be used to detect regional minima and maxima

- Regional minima can be found by subtracting of the original image from the reconstruction by erosion which is applied to the original image which is incremented by "1"

$$RMIN(f) = R_f^{\varepsilon}(f+1) - f$$


Analoguous, the regional maxima can be found by

$$RMax(f) = f - R_f^{\delta}(f-1)$$

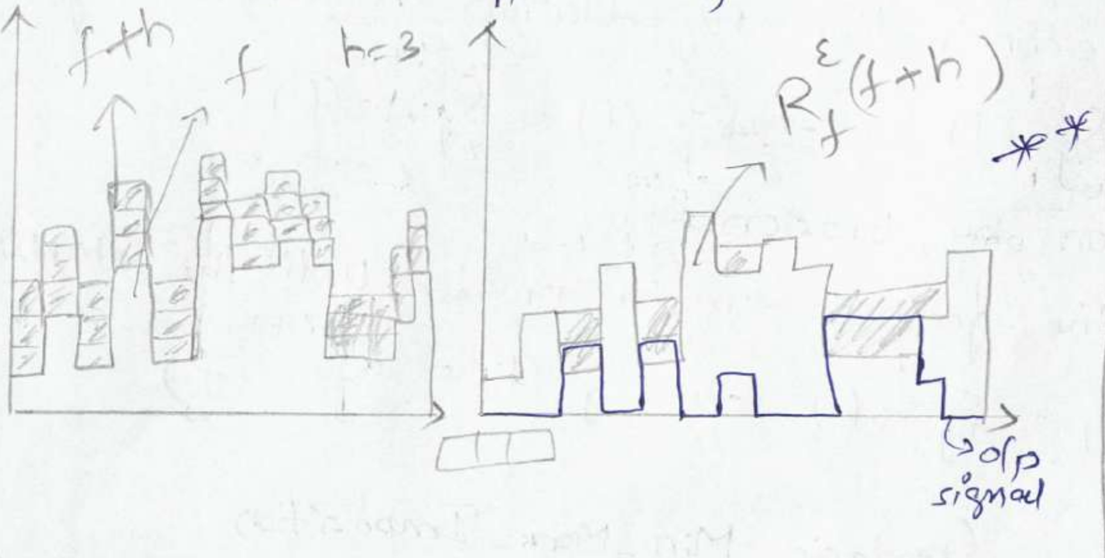
Note: Image data does not support negative values, then use $RMAX(f) = (f+1) - R_{(f+1)}^{\delta} f$

⇒ **h-minima:-**

The regional minima operator detects all minima, even the smallest, if they have a local difference of 1 w.r.t their neighbours. Noise often yields an overdetection of regional minima.

To avoid that overdetection, we can offset the original image by more than one gray value, let say h to get marker image.

$$HCONCAVE_h(f) = R_f^{\varepsilon}(f+h) - f$$




(i) Add original image by 'h' pixels to avoid overdetect from neighbourhood
(ii) Reconstruction by erosion to the original image with 'h' pixels
(iii) Subtract original image from reconstruction f+h to get h minima

⇒ **h-maxima:-**

H-maxima are complementary to h-minima i.e

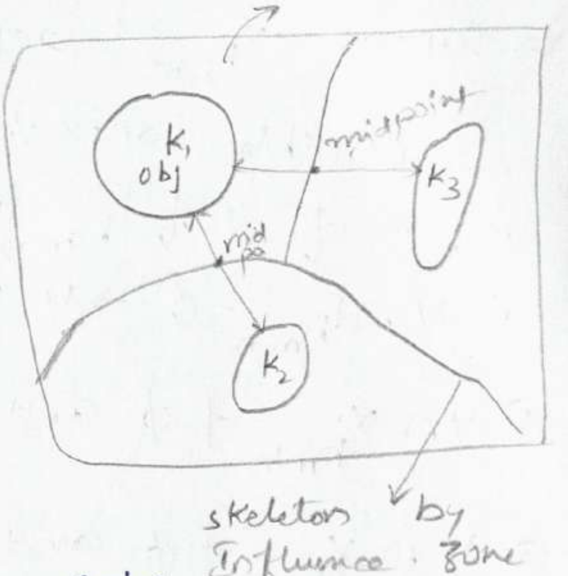
It is obtained by subtracting an image reconstruction of dilated by from the original image

$$HCONVEX_h(f) = f - R_f^{\delta}(f-h)$$

⇒ Minima Imposition:-
This technique is used to change image pixels to strongly ascending ones from all imposed minima

V.19 Influence zone, Watershed Transformation

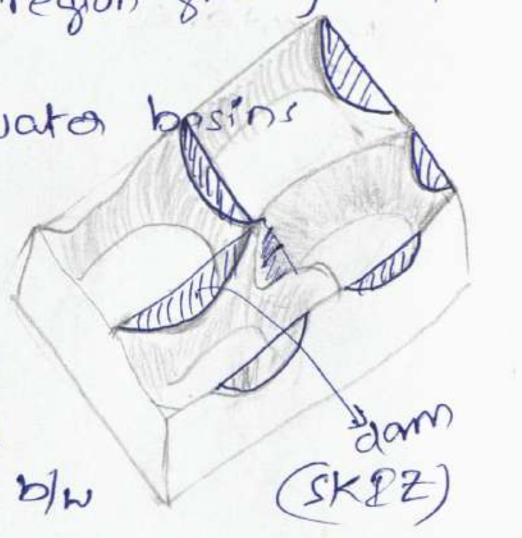
⇒ Influence zone:
The set of pixels which are closer to a given connected object than to any other object is called the "Influence Zone (IZ). The boundaries of the influence zones defines the "skeleton by Influence zone".




⇒ skeleton by Influence zones:-
It is defined as the quantity of points that do not belong to any influence zone.
The skeleton by influence zones (SKIZ) can be approximate by thickening until idempotence (stability) (or) watershed transformation


* 8 connected objects are wider than 4 connected objects

⇒ Watershed Transformation: It is a better approach to get skeleton by Influence zones
** - A very mighty algorithm, used both region growing and edge detection
- Working principle is like filling of water basins
We punch holes into the surfaces minima points. Then we immerge the surface, which is like a shell slowly into water. The water fills into bain through the minima holes and forms a watershed b/w basins.



When the water level has reached the highest point of the topographological surface, then the dams represent watersheds and regions are completely surrounded by the dams.

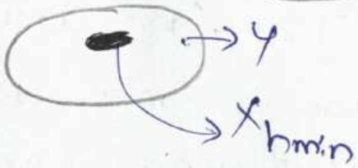
"Each segment is associated to exactly one local minimum"

⇒ Suppose $X_{h_{min}+1}$ represents regional minima at $h_{min}+1$ level

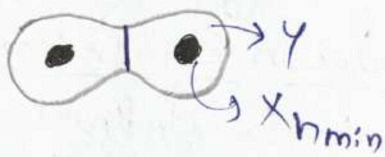
3 possible (~~correction~~) situation between connected region of $Y$ of $T_{h_{min}}(f)$ and intersection b/w $Y$ and $X_{h_{min}}$

① $Y \cap X_{h_{min}} = \phi \Rightarrow Y$ is a new regional minimum


② $Y \cap X_{h_{min}} \neq \phi$ and is connected




③ $Y \cap X_{h_{min}} \neq \phi$ and is not connected




⇒ Image Segmentation Techniques:

Segmentation is the classification of each image pixel into set of classes, here $w_0$ = background & $w_1$ = object

Approaches:

1) Constant value thresholding - This will fail if the illumination changes or noise is high

2) Region growing: Detect homogeneous regions first and grow those regions based on similarity measurements of spatial & spectral attributes

3) Edge Detection: Use high gradient locations as object boundaries.

V20 Image Enhancement Chapter 4

Averaging, Recursive, Seperable

**

**Temporal Averaging**:- → One of method to reduce noise

- We need operators for noise reduction and for equalizing inhomogeneous illuminations.

⇒ Noise is assumed as a statistical random variable with mean '0' & standard deviation 'σ'. The measured grey value $\hat{g}_{r,c}$ is given as $\hat{g}_{r,c} = g_{r,c} + n_{r,c}$

$g_{r,c}$ → arithmetic average, calculated over no. of sensor data i.e, $g_{r,c} = \frac{1}{n}\sum_{i=1}^{n} \hat{g}_{r,c;i}$

$n_{r,c}$ → gaussian distributed noise

⇒ for noise reduction, we calculate $g_{r,c}$

⇒ The standard deviation is reduced by the factor of $\sqrt{n}$ since the mean variance is $\sigma_m^2 = \frac{\sigma^2}{n}$

⇒ **Spatial Arithmetic Averaging**:

for time-critical applications or in case of moving objects or dynamic scenes, we can calculate the spatial arithmetic averaging value over a $(2m+1)\cdot(2n+1)$ mask is

** $$\boxed{g_{r,c} = \frac{1}{(2m+1)(2n+1)} \sum_{i=-n}^{n} \sum_{j=-m}^{m} \hat{g}_{r-i,\,c-j}}$$

| Arithmetic averaging | Gaussian averaging |
|---|---|
| - can be computed / programmed very fast because it is seperable or recursive | - little bit slower as it use seperable program not recursive |
| - very fast algorithm | - time taking algorithm but the object can view properly |
| - drawback is the production of direction-dependent artefacts i.e | - Independent of direction so, ~~no~~ no artefacts |
| - Objects not view clearly | |

⇒ Seperable filters **

The arithmetic averaging operator is known as a Mean filter.

A filter is an operation that takes input function and produces a functions as an output. Image can be a function of grey values with rows & columns i.e $g(r,c)$.

- Mean filter is a linear filter
  - ↳ seperable
  - ↳ Recursive

i) Seperable : It is not necessary to compute $m \cdot n$ operations for each pixel of the input image.

for example, an input image with size $w \cdot h$, then in the case of non-seperable $(2m+1)(2n+1)$ filter, we have to compute $(2m+1)\cdot(2n+1)\cdot w \cdot h$ operation. But, if we can seperate the operation in rows & columns, then only $(2m+1)\cdot h \cdot w + (2n+1)\cdot w \cdot h = (2m+2n+2)\cdot w \cdot h$ operations are to be computed

Example : An image with $w = 1000$, $h = 800$, $n = m = 50$ (mask size)

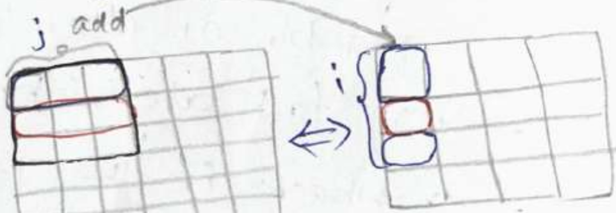
non-seperable / naive $\Rightarrow (2m+1)(2n+1)\cdot w \cdot h$
$= (101)(101)(1000)(800) = 8.16 \times 10^9$

seperable $\Rightarrow (101+101+2)\, 1000 \times 800 = 1.63 \times 10^8$

ii) In the case of the mean filter

$$g_{r,c} = \frac{1}{(2n+1)(2m+1)} \sum_{i=-n}^{n} \left( \sum_{j=-m}^{m} \hat{g}_{r-i,c-j} \right)$$

Here, first compute the inner sum over j, one for each row & store the result temporarily in a lookup table (LUT). In the second run, the sum over i can be computed using the values stored in LUT




$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

In fig, 3x3 mask, firstly 3 rows add & stored in LUT & next run, 3 columns are computed.

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

ii) * Recursive filters :-

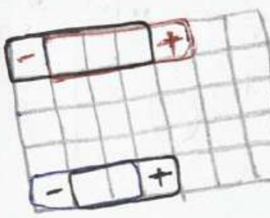
In this type of filter, the next pixel can be calculated from the result of previous filter by adding the result of the next boundary pixel and subtracting the result of the previous boundary pixel.

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 1 | 2 | 1 |
| 1 | 2 | 1 |

Recursive in y direction

$$t_{r,c} = \sum_{j=-m}^{m} \hat{g}_{r,c-j} = t_{r,c-1} + \hat{g}_{r,c+m} - \hat{g}_{r,c-m-1}$$

Here, only two operations, one for each column and one for each row have to be executed.



| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 5 | 7 | 1 | 3 |

| | | | |
|---|---|---|---|
| 8 | 8+7 −1 | 14+1 −2 | 13+3 −5 |
| | 14 | 13 | 11 |

Instead of adding 3 pixels in every step, we can add next pixel value to the result & subtract the previous pixel value

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 1 | 1 | 1 |

Recursive in x direction

- This decreasing the processing time with an increasing filter size

V.2 . Convolutional_ Filters :-

⇒ Linear filters :-

Linear means, that applying a filter 'h' to a linear combination of two input images f(p) and g(p) yields the same result as applying the filter h to two images seperately and then computing the linear combination

- $$h\{a f(p) + b g(p)\} = a h\{f(p)\} + b h\{g(p)\}$$

For 2D, a linear filter can be computed as convolution

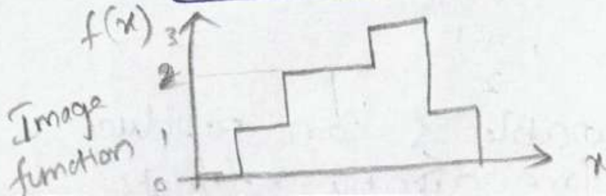
** $$\boxed{f * h = (f * h)(r,c) = \int_{v=-\infty}^{v=\infty} \int_{u=-\infty}^{u=\infty} f(u,v)\, h(r-u, c-v)\, du\, dv}$$

for discrete image, this can be written as

** $$\boxed{f * h = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f_{i,j}\, h_{r-i,\, c-j}}$$

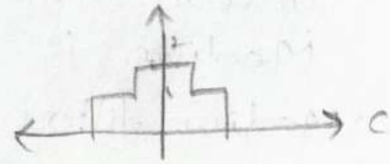
Note that the mask 'h' is mirrored

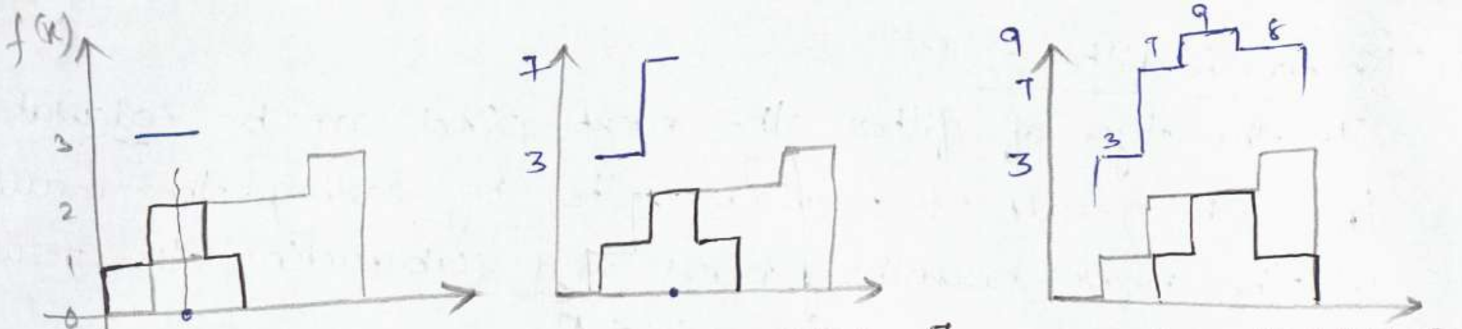
**1 D example**




f(x) is gray value of image x

simple mean convolution mask



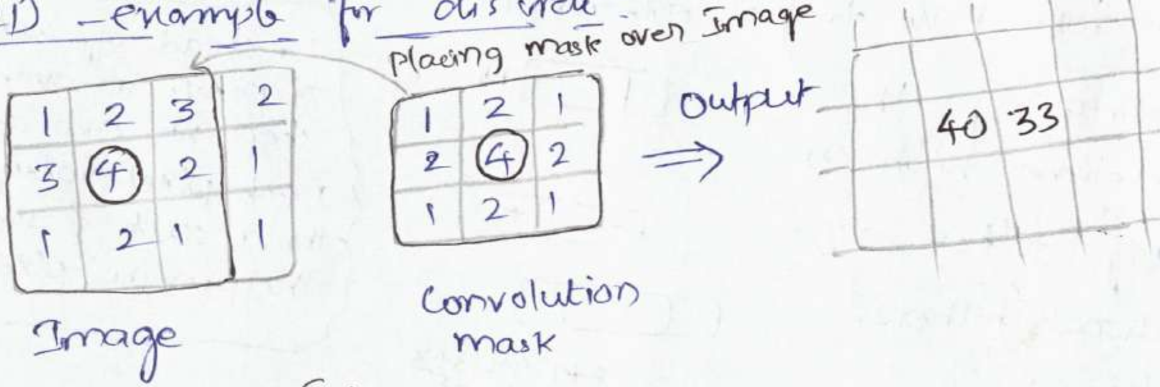



$0\times1 + 1\times2 + 1\times1 = 3$ | $1\times1 + 2\times2 + 2\times1 = 7$ | $2\times1 + 2\times2 + 3\times1 = 9$

Placing convolution mask on image & calculate gray value

shift mask to next pixel and continues.

2D - example for discrete:




$1\times1 + 2\times2$
$+ 3\times1 + 3\times2$
$+ 4\times4 + 2\times2$
$+ 1\times1 + 2\times2 +$
$+ 1\times1$
$=$

⇒ Gaussian filter :-

The Gaussian filter is given by

1D - $h_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{(-x^2/2\sigma^2)}$

2D - $\boxed{h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{-u^2-v^2}{2\sigma^2}\right)}}$ **

⇒ Gaussian filter 2D version is seperable but not recursive (so there are no artefacts & image seen clearly comparted to arithmetic filters)

* So, The Gaussian filter gives a better smoothing since it is direction independent.

⇒ Median filter: A very simple filter for image smoothing.

The median filter is defined as the value for which 50% of the values in the probability distribution are smaller and 50% are larger.

Median is calculated by $\frac{n}{2}$ or $\frac{n+1}{2}$

→ Median filters are nonlinear, not seperable & can produce unforecastable effects.

V22. Example Convolutional filters :-

Gaussian function in 2D $h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{u^2-v^2}{2\sigma^2}\right)}$

| | | |
|---|---|---|
| 0.51 | 1.78 | 0.51 |
| 0 | 0 | 0 |
| -0.51 | -1.78 | -0.51 |

Differentiate with respect v gives

$$\frac{\partial}{\partial v} h_\sigma(u,v) = \frac{-v}{2\pi\sigma^4} \cdot e^{\left(\frac{u^2-v^2}{2\sigma^2}\right)}$$

⇒ Sobel-y filter: $\frac{\partial}{\partial v}(h_\sigma(u,v))$ with $\sigma = 0.4$ and rounding the floating points to integer.

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

Sobel-x filter: $\frac{\partial}{\partial c} h_\sigma(u,v)$ i.e differentiate 2D Gaussian filter with respect to c and setting $\sigma = 0.4$. The table rounded to integers from floating point yield Sobel-x filter

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

## Chapter 5 Feature Extraction

V.23

⇒ Template Matching: It is used to find and localize one or more objects which are not well shaped and which always have the same orientation.

1) Translation transformation
2) Rotational & translation
3) " " " & scaled

⇒ Color-Based Template Matching :-

A simple way to calculate the matching of a template with a background image is to compute the similarity as sum of absolute or squared gray value differences SAD / SSD.

$$** \quad sad(r,c) = \frac{1}{n} \sum_{(u,v) \in T} | t(u,v) - f(r+u, c+v) |$$

$$** \quad ssd(r,c) = \frac{1}{n} \sum_{u,v \in T} (t(u,v) - f(r+u, c+v))^2$$

with template quantity $t(u,v)$ and the image quantity $f(r,c)$

- To accelerate template matching, usually AOI (area of interest) or ROI (region of interest) should be defined.

⇒ Because of high brightness sensitivity of the SAD and SSD algorithm, an advanced algorithm has developed, that is normalized cross correlation (NCC).

$$ncc(r,c) = \frac{1}{n} \sum_{u,v \in T} \left( \frac{t(u,v) - m_t}{\sqrt{s_t^2}} \cdot \frac{f(u+r, c+v) - m_f(r,c)}{\sqrt{s_f^2}} \right)$$

where, $m_t$ - mean gray value of template

$m_t = \frac{1}{n} \sum_{u,v \in T} t(u,v)$ ; $s_t^2$ is variance of gray value of the template.

$$s_t^2 = \frac{1}{n} \sum_{u,v \in T} (t(u,v) - m_t)^2$$

lly $m_f$ - mean gray value of image & $s_f$ - variance

$$m_f(r,c) = \frac{1}{n} \sum_{u,v \in T} f(r+u, c+v)$$

$$s_f^2(r,c) = \frac{1}{n} \sum_{u,v \in T} (f(r+u, c+v) - m_f(r,c))^2$$

⇒ <u>Extracting Edges</u> - <u>1D Gradients</u>:-

- To detect edges, the best method is to detect the local maxima of the <u>first derivates</u> of a signal. Note that, the noise in the averaged signal has been significantly reduced

* Average signal then do derivation

- for a discrete signal, the first symmetric derivation can be calculated by, $f'_i = \frac{1}{2}(f_{i+1} - f_{i-1})$

**

- $\boxed{(f * h)' = f * h'}$

$(f*h)'$ - convolution (smoothing) and subsequent derivation

$f*h'$ - Convolution of f with a derived filter mask

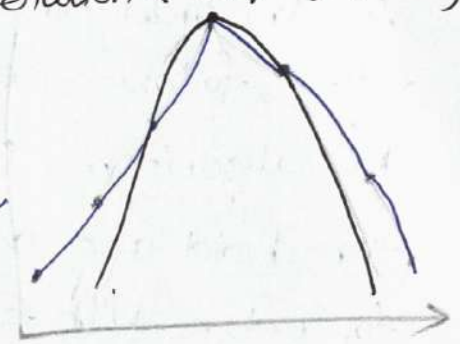
$(f*h)'$ ⇒ first convolute f with h (mask) and derived the convolution image - 2 operations are involved

$f*h'$ ⇒ Convolution of f with derived mask ($h'$) so, only one operation is required for smoothing and derivation.

⇒ Gaussian filter is the best for edge detection in Image processing, according to Canny research in 1988.
The first derivative of the 1D gaussian filter is,

$$g_\sigma(x) = \frac{-x}{\sigma^3\sqrt{2\pi}} \cdot e^{\left(-\frac{x^2}{2\sigma^2}\right)}$$

$\sigma = [0.4, 1.0]$ range is good choice for smoothing operation & edge filtering



⇒ <u>Sub-pixel-precision</u> :-

To find the maximum in first derivation's peak with sub-pixel precision, we approximate the peak by a parabola using 3 points and calculate the maximum of the parabola.

2D Gradients

↓

● ⇒ <u>Extracting</u> Edges – 2D Gradients :-

⇒ <u>Sobel-y-filter</u> & -<u>Sobel-x-filter</u>

Deriving Gaussian (2D) filter with respect to 'r' and setting $\sigma = 0.4$ then the table is resulted as follow (y-direction)

$\frac{\partial}{\partial r} h_\sigma(u,v)$ with $\sigma = 0.4$ ; $h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{-u^2+v^2}{2\sigma^2}\right)}$

| | | |
|---|---|---|
| 0.51 | 1.78 | 0.51 |
| 0 | 0 | 0 |
| -0.51 | -1.78 | -0.51 |

Sobel-y filter for a vertical filtering direction is rounding the floating values

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

● If we derive 2D Gaussian filter with respect to c with $\sigma = 0.4$, we result the table as

| | | |
|---|---|---|
| 0.51 | 0 | -0.51 |
| 1.78 | 0 | -1.78 |
| 0.51 | 0 | -0.51 |

Sobel-x-filter for horizontal filtering direction is

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

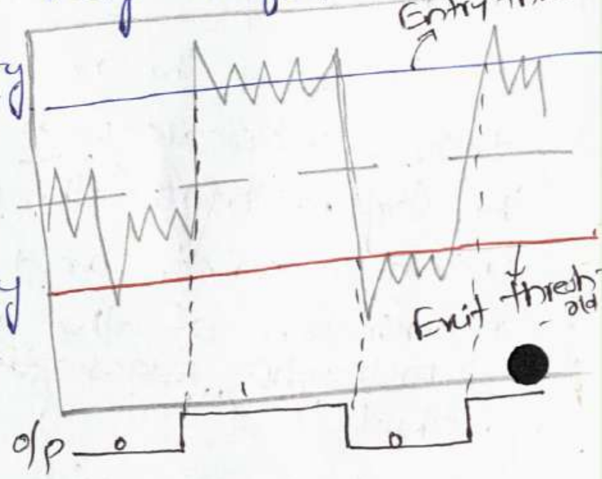
– A fast algorithm to find edges with pixel accuracy is to apply the sobel-x-filter and the sobel-y-filter to the input image. The result have to combined by pixel by pixel Euclidean distance operator

$$\text{Euclidean distance} = \sqrt{\Delta x^2 + \Delta y^2}$$

⇒ Edge follower :- To make the edge extractor more robust, an edge follower algorithm can be combined.

- Sometimes the edge gradients are weak or blurred along the edge lines. This causes breaks in edge lines.
So, in order to get better edge results, Canny developed edge follower algorithm, which is very useful.

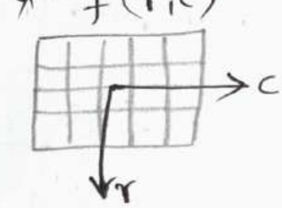
The algorithm is, define an entry threshold and an exit threshold for edges with a high noise level.
If the signal passes over the entry threshold, the edge following is set to active until the signal passes under the exit threshold




In Iconet, upper Gradient Threshold → Entry threshold,
lower " " → Exit threshold.

V-25 Template Matching

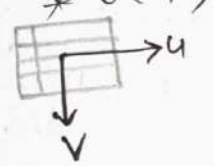
⇒ Transformations :-
① Translational transformation (object only shifted)
② Translational & rotational " (object is rotated & shifted)
③ Translational, rotational & scaled " (object is scaled, rotated & shifted)
④ Translational, rotational, scaled & projective transformation in 3D




⇒ SAD - sum of Absolute gray value difference

$$sad(r,c) = \frac{1}{n}\sum_{(u,v)\in T} |t(u,v) - f(r+u, c+v)|$$

⇒ SSD - sum of squared gray value differences




$$ssd(r,c) = \frac{1}{n}\sum_{u,v\in T} \left(t(u,v) - f(r+u, c+v)\right)^2$$

where $t(u,v)$ → template & $f(r,c)$ → Image

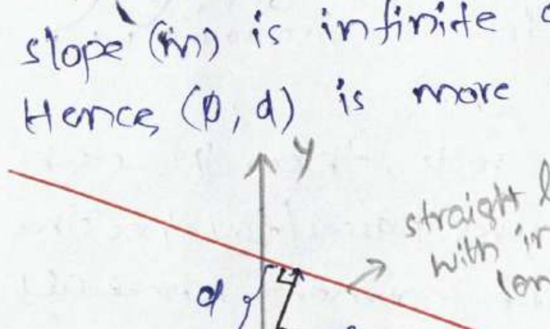
## * V27 5.3 Hough Transform for Extracting lines

Since edges are a very strong description for figures, they are often used to detect and analyze and to localize their poses (a "pose" is nothing but position of object in x, y, $\alpha$)

- so, to detect edges and their poses of object, a robust and powerful tool is available, that is GHT (Generalized Hough Transform). With this algorithm, we can detect not only x, y position but also rotational angle of objects.
- This algorithm can be extended to higher order object coordinates like (3D, 4D, 5D ...), but higher orders (5D/6D) are time consuming and not suitable in industrial applications. So, we deal with 2D.
- A 2D straight line can be defined by two parameters.
  i) The $\perp r$ distance from the origin to the line (d) which is the length of lead
  ii) The angle of the lead against the x-axis ($\phi$)

Why this two parameters?

- In programming, we can't handle with infinite variables suppose, if the straight line parallel to y axis, then slope (m) is infinite of line $y = mx + c$.
  Hence ($\phi$, d) is more appropriate for subsequent deductions



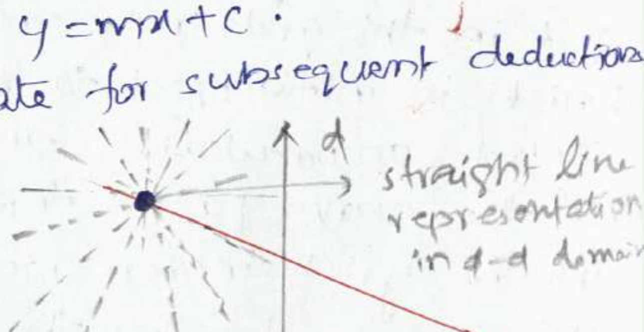
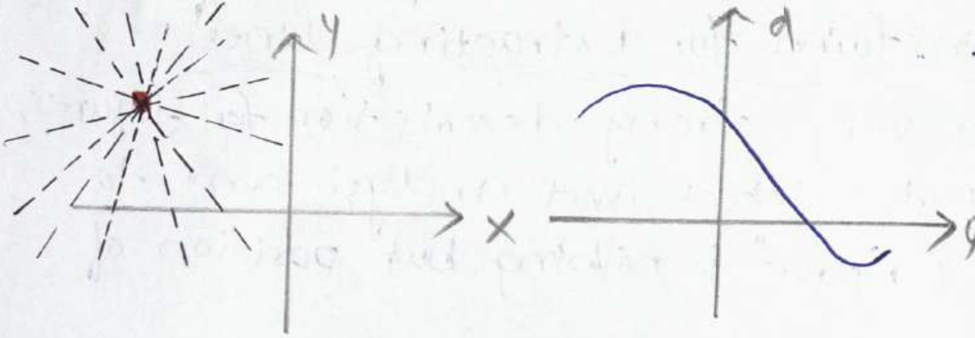



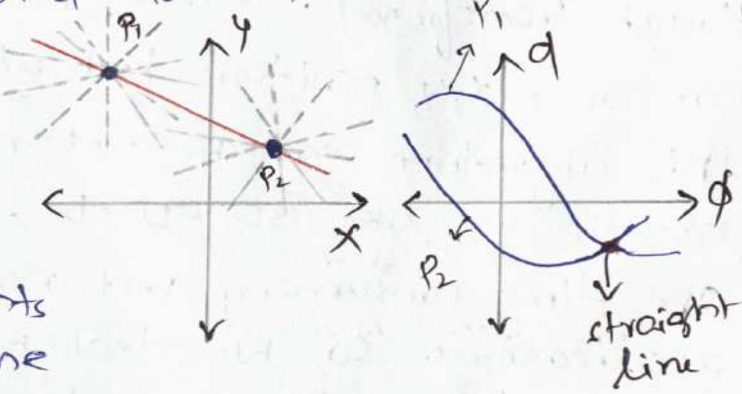

fig: line representation in spacial domain

fig: Line representation (i.e, point) in $\phi$-d domain

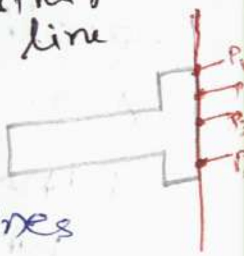
** so, "In the Hough transformed, a straight line with infinite length in x-y plane is transformed by a single point in $\phi$-d domain (with lead length d and angle $\phi$)"



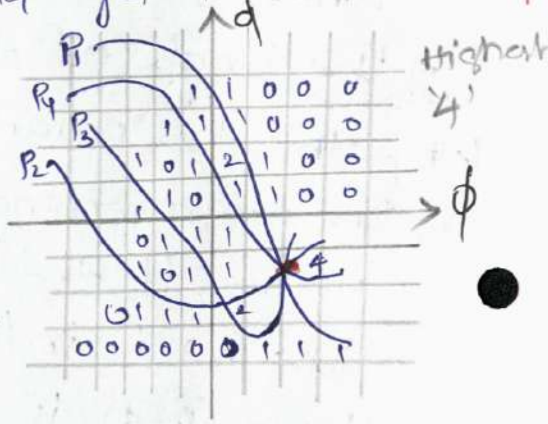

→ A point in the X-Y domain is represented as curve in ϕ-d domain, which represents all straight line of XY domain running through the point.
i.e. all (points) lines passing through the point in X-Y domain is curve in the ϕ-d domain.




→ In ϕ-d domain, two points $P_1, P_2$ (in X-Y domain) are represented by two curves. The curves intersection points represents the straight line which is passing through two points.



How this algorithm will implement in software?
This requires an "accumulator cell" and get the lines of edges by "voting algorithm."



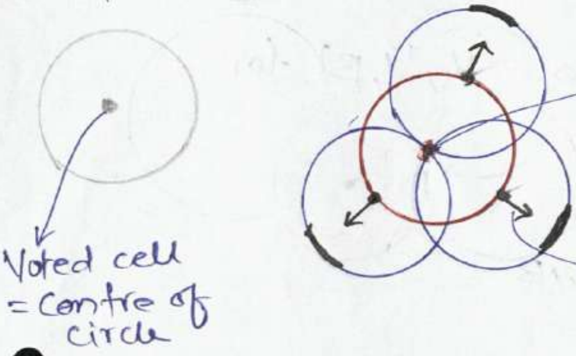

⇒ The ϕ-d domain space is subdivided into an orthogonal grids of cells. Each cell in this grids represents a counter which is initialized to zero. It is called accumulator cell.

⇒ If the curve passes through the grids, then the each cell in the grid incremented if curves passes/touches the cell. This increment gives the maximum counter where all curves intersects which represents the straight line that is fitting the edge.

⇒ This algorithm is called Voting algorithm because each extracted edge point votes for each straight which probably runs through this point.

## 5.4 Hough Transform for Extracting Circle :-

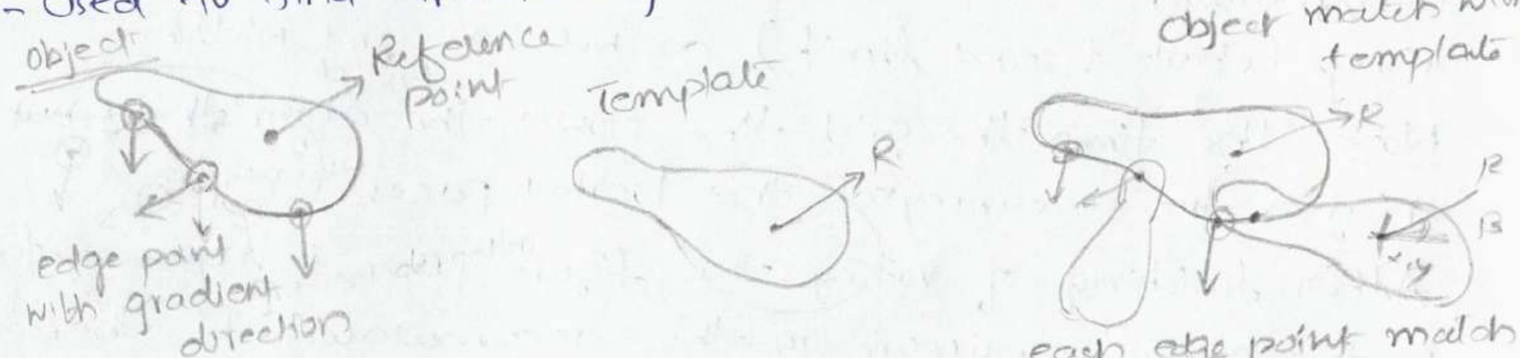
Detecting circles with Hough Transform and voting accumulator is very effective.
Each point which has extracted from an input image votes the probable centre point of all circles which run through this point. The voting curves are also circles with dia of same.



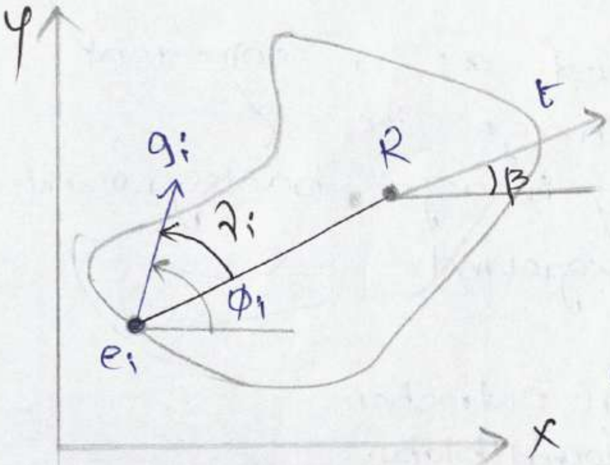

- 

## 5.5 Generalized Hough Transform :-

- More advanced algorithm.
- Used to find arbitrarily shaped objects in an image.



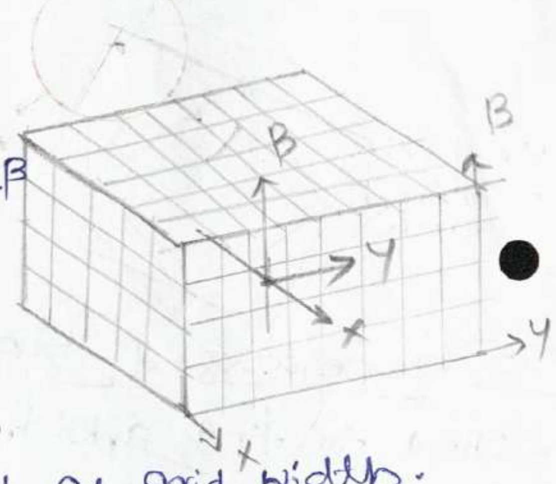

- The Generalized Hough transform is designed to match arbitrary edges with contours. In combination with an accumulator voting procedure, this is very powerful operator because objects are matched with template and compute (x, y, angle).




An arbitrary image with base/ reference point and direction vector 't'. Some edge points $e_i$ with edge gradient direction $g_i$ are extracted from the input image.
The extracted points with their gradient directions are used to calculate 2D pose $(x, y, \beta)$ of ref. point ('R').

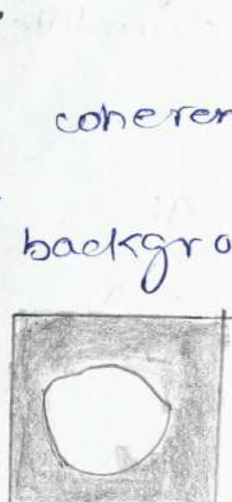
So, Image has 'n' edge points (passer point) with a reference point and template has similarly 'n' edge points. Hence, each edge point in template match with each edge point in Image, compute $(X, Y, B)$ pose of image.

<u>Algorithm</u>: for each extracted point
{ for each possible passer point on template
{ vote each possible pose $(X, Y, B)$ for the figure
}
}




- To represent all pose parameters X, Y, B a three dimensional accumulator must be provided.
Due to memory restrictions, an accumulator cell is defined with range (start & end limit) as well as grid width.
<u>Note</u>: The finer the grid, the slower the algorithm, but higher the accuracy of the located poses.
- After finishing of voting, the figure pose is represented by the global maximum in the accumulator. If there are more figure, all of the corresponding local maxima have to be found. This can be done within two steps.
1. A recursive algorithm of finding all regional maxima
2. Based on centre of gravity of each regional maxima, output pose $(X_{ci}, Y_{ci}, B_{ci})$ can be computed.

## <u>5.6 Blob Extraction</u>:-
<u>What is Blob?</u> A blob is defined as a coherent object in the image i.e.
white objects in a black binary image's background or black objects in a white background can be treated as Blobs.



- Basic task is localize blobs by feature extraction for each of the found blobs.

As blob is a coherent object in the image.
Two types of coherences a) 4-neighbourhood and
b) 8-neighbourhood coherences.

**

| | | |
|---|---|---|
| | P2 | |
| P3 | P0 | P1 |
| | P4 | |

4-neighbourhood of pixels

| | | |
|---|---|---|
| P4 | P3 | P2 |
| P5 | P0 | P1 |
| P6 | P7 | P8 |

8-neighbourhood of pixels

## Detection & labelling of Blob Extraction:-

A naive approach for detection and labelling is to
search white pixels row by row (or) column by column
the pixel number is incremented. Then start a recursive
loop over all neighbours (4 or 8), if white pixel is
found, copy the same number. If no pixel of neighbour
is found, then current label number is incremented.

## ** Algorithm in Pseudocode:-

```
function RecursiveLoopOverAllNeighbours(Col, Row)
{ for (iNgb = 0 ; iNgb < 4 (or) ; iNgb ++)
  { compute NgbColRow ( Col, Row, iNgb, & Ngbcol, & NgbRow)
    if (NeighbourPixel [Ngb col][NgbRow] is white and not
         labeled)
    { Label Pixel [NgbRow] [Ngbcol] with current label number
      RecursiveLoopOverAllNeighbours(NgCol, NgbRow)
    }
  }
}
```

Since the above algorithm is recursive, it takes more
time for computation. So, a better and faster approach
is there, which is based on matrix operation and
done in three steps.

1. Label in a single straightforward run over all columns
and rows. Set labels to label matrix which has
same extent as image. Set the elements in coexistlrtable

2. Sort coexist table by setting minimum values to rows and columns altering
3. Re-label label matrix.

## 6. Thresholding

⇒ fixed threshold in a gray scale Image

Setting a threshold for a Gray scale image with high contrast and a homogeneous illumination is very simple, which can be done by defining a fixed value that can be found from a histogram distribution

⇒ fixed Threshold in a color image:-

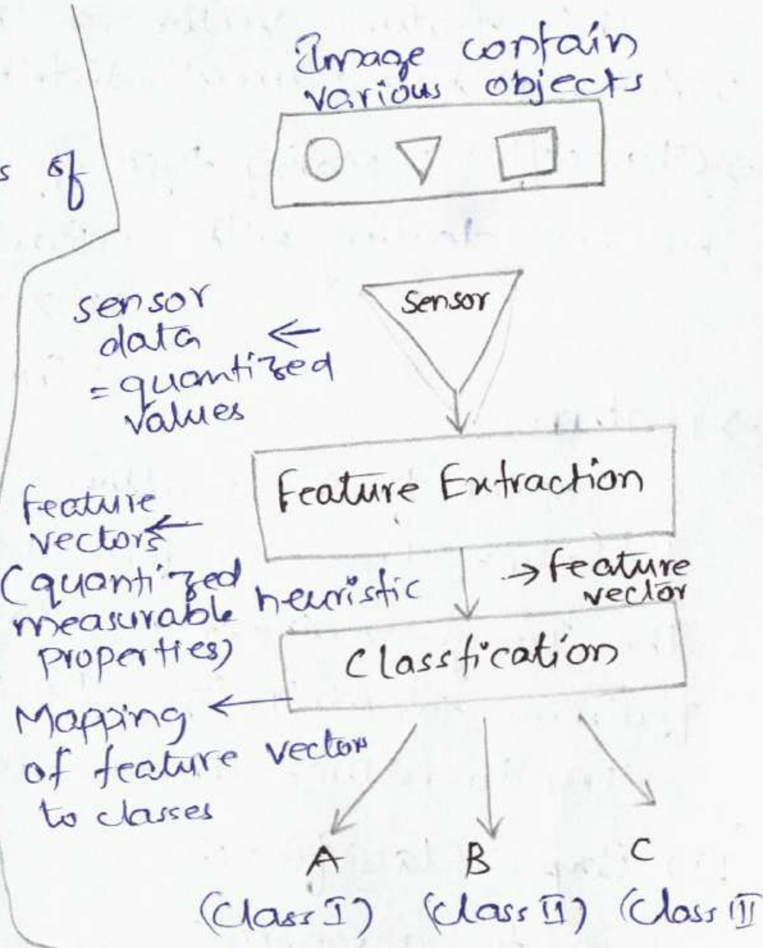
In a color image, fixing a threshold is more difficult since the color values in the blobs may have (for eg. same euclidean) brightness as in the background. So we can define a difference vector (blob - Background) and a separation plane which is located in the centre of the difference vector. The separation plane is ⊥r to the difference vector.

# 6. Classification

## 6.1 Introduction:-

1. Image contain various types of objects (shape, size)
2. sensor, camera capture the image and gives quantized values for different objects
3. To extract the quantized values from sensors, we use certain features (like, text, shape ...)

- feature vectors are used to define certain features of the object

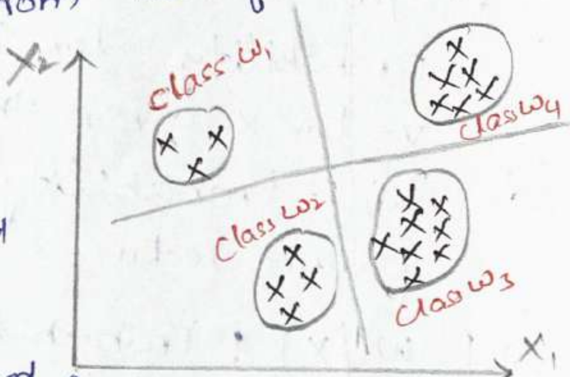



4. The feature vector is fed into classifier. Then the classifier do the classification of object based on the feature vector.

⇒ Principle of classification

Classification is the mapping from the feature space to a set of possible classes.

- Based on the feature vector, we classify the quantized values captured with the help of sensor.




"Example of 2 element feature vector"

As shown in fig, 4 classes are classified based on 2 Element feature vector $E(x_1, x_2)$

class $\omega_1$ : $x_1$ low ; $x_2$ High

class $\omega_2$ : $x_1$ low & $x_2$ low

class $\omega_3$ : $x_1$ High & $x_2$ low ; class 4 : $x_1$ High & $x_2$ High

So from the image, a n-element feature vectors $x_1, x_2 \cdots x_n$ can be obtained by various feature extraction algorithms. Then, we can observe this vector for each image, section of an image or object which has to be classified, This feature vector can be assumed as a statistical variable with a normal distribution

⇒ Classifiers: Many types of classifiers are there. But we are dealing with 1) Bayesian classifier
2) 2 and 3 layer Perception
3) Convolutional Neural Networks

⇒ Features :-

Good features allow feature vectors belonging to different classes to form compact & disjoint regions. The Image engineer has to identify those certain features to minimize classification error rate.

⇒ Try to reduce dimensionality by combination of features.

6.2
ID Baye's Classifier :-

Probability acronyms:-

1) $P(\omega_i)$ - Probability of the occurances of class $\omega_i$ $(i=1,2 \dots N)$
** It is also called "Prior-probability" because it can be calculated from the training pattern.

2) $P(X/\omega_i)$ - Conditional probability i.e
** probability density of occurances of the feature vector x given the class $\omega_i$

** ③ $P(X)$ :- Probability density of occurances of the feature vector x. This value has an analogous codomain

** 4) $P(\omega_i/X)$ : Probability that a given feature vector x belongs to the class $\omega_i$. This is also called "Posterior - Probability".

Consider a dataset contain feature vector, class as tuple
i.e, $(x_0, \omega_0)$ $(x_1, \omega_0)$, $(x_2, \omega_1)$ , $(x_3, \omega_0)$ - . . . . . .

With trained data-set, we can compute $P(\omega_i)$, $P(X/\omega_i)$

$$P(\omega_i) = \frac{\text{no. of occurances of } \omega_i}{\text{No. of data set tuples}}$$

$$P(X/\omega_i) = \frac{1}{\sqrt{2\pi}\cdot\sigma_i} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma_i^2}\right)$$

- Because $P(\omega_i)$ and $P(X/\omega_i)$ can be directly estimated from the training data set, which is given before the classifier runtime, they are called <u>Priori probabilities</u>
- Because $P(\omega_i/x)$ is computed at runtime, it is called a <u>Posteriori probability</u>.

$$\boxed{P(\omega_i/x) = \frac{P(x/\omega_i)\,P(\omega_i)}{\sum_{j=1}^{n} P(x/\omega_j)\,P(\omega_j)}}$$

∴ We know that, $P(A) = \sum_{i=0}^{N-1} P(A/E_i)\,P(E_i)$ and

$$P(E_i/A) = \frac{P(A/E_i)\cdot P(E_i)}{P(A)} = \frac{P(A/E_i)\cdot P(E_i)}{\sum_{i=0}^{N-1} P(A/E_i)\cdot P(E_i)}$$

lly,

$$P(\omega_i/x) = \frac{P(x/\omega_i)\,P(\omega_i)}{\sum_{k=0}^{N-1} P(x/\omega_k)\,P(\omega_k)}$$

This is the Baye's theorem for discrete classes $\omega_i$ with an one-dimensional analogous feature value $x$.

* <u>Common Bayes Decision Rule</u> :-

In the above eqn, the denominator is independent from class $\omega_i$, the decision rule can omit the denominator as follows

$$x \in \omega_i \Leftrightarrow P(x/\omega_i)\,P(\omega_i) > P(x/\omega_k)\,P(\omega_k) \quad \forall\, k \neq i$$

- In this decision rule, the probabilities are equal weighted, hence the cost of an erroneous decision for each class.

If weight are unequal (i.e $\omega_o > \omega_i$), then

$$x \in \omega_i \Leftrightarrow \sum_{l=0}^{N-1} \lambda_{il}\, P(x/\omega_l)\,P(\omega_l) < \sum_{l=0}^{N-1} \lambda_{kl}\, P(x/\omega_l)\,P(\omega_l) \quad \forall\, k \neq l$$

This is called "common Baye's Decision rule."

6.3 <u>Multivariate Baye's Classifier</u> :- **

<u>Multivariate feature Vectors</u>

In real, not only one feature value is used for the characterization of an image object.

for example, in OCR [optical characterstic Recognition], the characters are characterized by two feature values which are composed of feature vector $\vec{x}$:

$$\vec{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} \text{Height/Width ratio } (r_{hw}) \\ \text{No. of black pixels } (N_b) \end{bmatrix}$$

Convert 1D Bayes theorem to multidimensional in order to get an appropriate decision ru

**N-D Gaussian Normal Distribution function:-**

In N-dimensional problem, the gaussian normal distribution is given by

$$p(\vec{x}/\omega_i) = \frac{1}{(2\pi)^{N/2} \cdot |\overline{K}_i|^{1/2}} \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T \overline{K}_i^{-1} (\vec{x}-\vec{\mu}_i)\right]$$

where, $\vec{\mu}_i$ - the expected mean vector

$\vec{x}$ - the feature vector

$(|\overline{K}_i|)$ - The determinant of the symmetric $N \times N$ co-variance matrix.

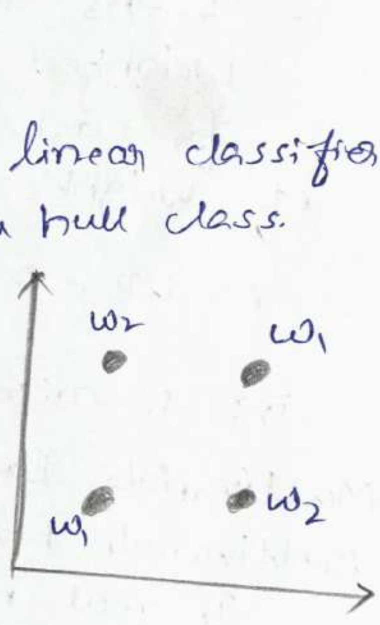
**6.4 K-Nearest-Neighbour Classifier:-**

Instead of estimating the k-nearest neighbours of a particular class and compute the volume $V(x_i, \omega_i)$, the k-nearest neighbours for the feature vector in the training set of any class are determined. The feature vector is then assigned to the class with the largest number of samples among the k nearest neighbours. This classifier is called k-nearest classifier (KNN).

**6.5 Linear classifiers:**

XOR problems:- Unfortunately, simple linear classifiers cannot solve problems which have a convex hull class.

Another example for a problem which is not classified by linear classifier is the XOR problem.



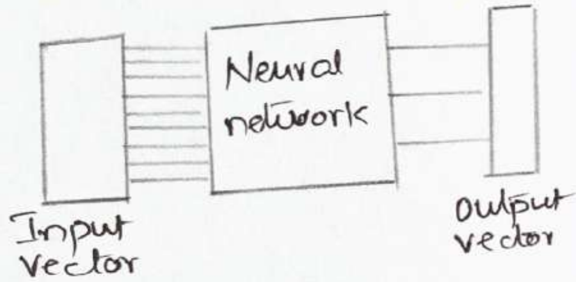

Here two classes can't be seperated by a simple straight line. Therefore a non linear classifier is needed.

## 6.6. Nonlinear Classifiers

**Neural Perception Network :-**

A neural perception network is the emulation of animal's or human brain cells in the computer. Like the brain, a perception network can learn a certain behaviour. In image processing, it is used as a classifier, which can be regarded as a black box.

The black box can be switched to either teaching mode or a working mode.




In teaching mode, a set of teaching data is shown to the neural network very often. The teaching data set consist of i/p vectors and corresponding output vectors which have the wanted behaviour of the neural net.

Train Data set: $\left\{ \begin{bmatrix} .3 \\ .7 \\ .2 \\ .1 \\ -.25 \end{bmatrix} \begin{bmatrix} .9 \\ .1 \end{bmatrix}, \begin{bmatrix} .2 \\ -.15 \\ .1 \\ .09 \\ .5 \end{bmatrix} \begin{bmatrix} .7 \\ .15 \end{bmatrix}, \ldots \right\}$

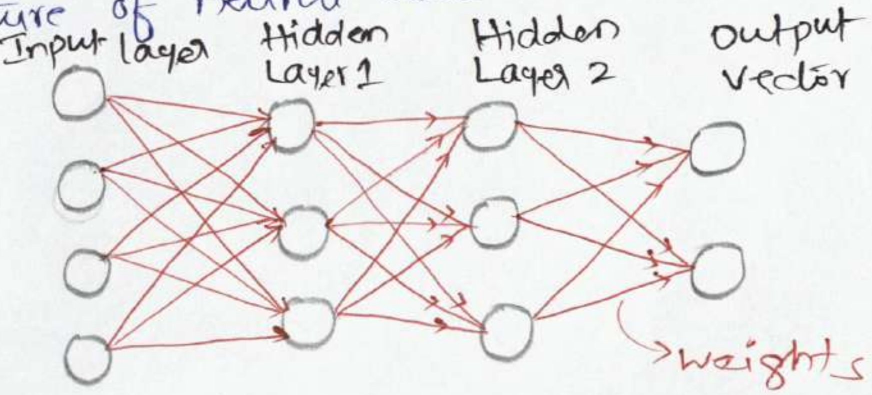
i/p vector → o/p vector

- In working mode, an i/p vector is given to the neural network is classified by calculating the o/p vector

Working data set : $\left\{ \begin{bmatrix} .5 \\ .6 \\ .1 \\ -.05 \end{bmatrix} \begin{bmatrix} ? \\ ? \end{bmatrix} \right\}$

- If training phase was successful, then the neural network classifies all i/p vector in right way.

- The structure of a neural network is similar to the structure of neural cells in a brain:

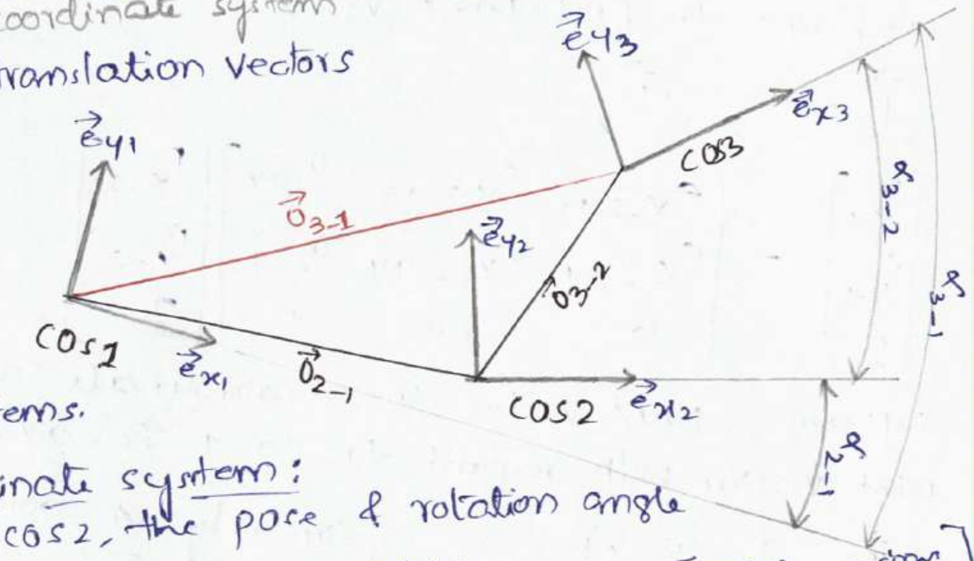


# 8. Geometric Transformations: 8.1 2D coordinate systems

We can consider three 2D coordinate systems COS1, COS2 and COS3. cos = coordinate system

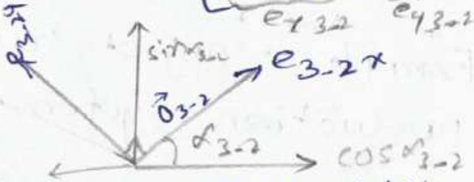
poses are defined by translation vectors $\vec{O}_{3-2}, \vec{O}_{2-1}, \vec{O}_{3-1}$ and rotation angles $\alpha_{3-2}, \alpha_{2-1}, \alpha_{3-1}$

The $\vec{e}_{ij}$ are unit vector of the coordinate systems.




<u>Pose of a 2D-coordinate system:</u>

COS3 with respect to COS2, the pose & rotation angle is $\vec{O}_{3-2} = \begin{bmatrix} O_{3-2x} \\ O_{3-2y} \end{bmatrix}$ and rotation matrix from $\alpha_{3-2}$ is $R_{3-2} = \begin{bmatrix} \underbrace{\begin{matrix} \cos\alpha_{3-2} \\ \sin\alpha_{3-2} \end{matrix}}_{e_{x3-2}} & \underbrace{\begin{matrix} -\sin\alpha_{3-2} \\ \cos\alpha_{3-2} \end{matrix}}_{e_{y3-2}} \end{bmatrix}$




<u>Transformation of a 2D coordinate system:</u>

To get the pose of COS3 with respect to COS1,

1) Translation: Transform the vector $\vec{O}_{3-2}$ into cos 1 and add it to $\vec{O}_{2-1}$ i.e, $\vec{O}_{3-1} = \vec{O}_{3-2} + \vec{O}_{2-1}$

2) Rotation: The rotation angles $\alpha_{3-1} = \alpha_{3-2} + \alpha_{2-1}$ (or) transform the unit vectors $\vec{e}_{x3-2}$ and $\vec{e}_{y3-2}$ which are given in COS2 to COS1

The vector $\vec{O}_{3-2}$ can be considered as a linear combination of its components, which are multiples of unit vectors

$$\vec{O}_{3-2-\cos 1} = \begin{bmatrix} O_{3-2x-\cos 1} \\ O_{3-2y-\cos 1} \end{bmatrix} = O_{3-2x-\cos 2}\,\vec{e}_{x2-\cos 1} + O_{3-2y\cos 2}\,\vec{e}_{y2-\cos 1}$$

with that, $R_{2-1} = \begin{bmatrix} \underbrace{\begin{matrix} \cos\alpha_{2-1} \\ \sin\alpha_{2-1} \end{matrix}}_{\vec{e}_{x2-\cos 1}} & \underbrace{\begin{matrix} -\sin\alpha_{2-1} \\ \cos\alpha_{2-1} \end{matrix}}_{\vec{e}_{y2-\cos 1}} \end{bmatrix}$

<u>Generalized 2D Transformation Matrix</u>

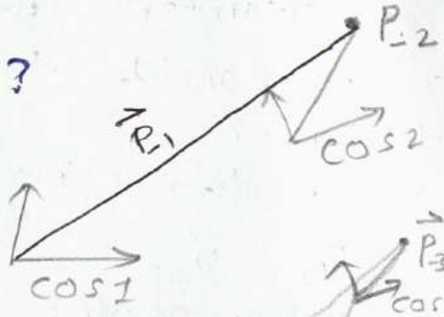
translation vector

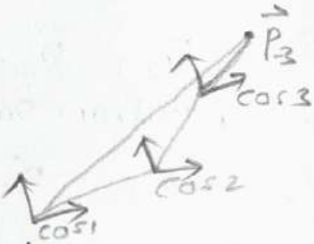
$$\overline{\overline{T}}_{3-1} = \overline{\overline{T}}_{2-1} \cdot \overline{\overline{T}}_{3-2} \quad \text{with} \quad \overline{\overline{T}}_{2-1} = \begin{bmatrix} \cos\alpha_{2-1} & -\sin\alpha_{2-1} & O_{3-2x-\cos 2} \\ \sin\alpha_{2-1} & \cos\alpha_{2-1} & O_{3-2y-\cos 2} \\ 0 & 0 & 1 \end{bmatrix}$$

rotation matrix

$$\overline{\overline{T}}_{3-2} = \begin{bmatrix} \cos\alpha_{3-2} & -\sin\alpha_{3-2} & O_{3-1x-\cos 3} \\ \sin\alpha_{3-2} & \cos\alpha_{3-2} & O_{1-3y-\cos 3} \\ 0 & 0 & 1 \end{bmatrix}$$

Given $\overline{\overline{T}}_{2-1}$ transformation matrix is known and point vector is cos2 ($\vec{P}_{-2}$). But we have to find point vector in cos1 ($\vec{P}_{-1}$)?




$$\vec{P}_{-1} = \overline{\overline{T}}_{2-1} \cdot \vec{P}_{-2}$$

$$\begin{bmatrix} P_{x-1} \\ P_{y-1} \\ 1 \end{bmatrix} = \begin{bmatrix} e_{x-2-1x} & e_{y2-1x} & 0_{2-1x} \\ e_{x-2-1y} & e_{y-2-1y} & 0_{2-1y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{x-2} \\ P_{y-2} \\ 1 \end{bmatrix}$$

Suppose, point vector in 3 coordinate systems, the point vector with respect to cos 1 is given by

$\vec{P}_{-1} = \overline{\overline{T}}_{2-1} \overline{\overline{T}}_{3-2} \cdot \vec{P}_{-3}$

$\vec{P}_{-1} = \overline{\overline{T}}_{3-1} \cdot \vec{P}_{-3}$

/ similarly 4 coordinates system, then

$\overline{\overline{T}}_{4-1} = \overline{\overline{T}}_{2-1} \cdot \overline{\overline{T}}_{3-2} \cdot \overline{\overline{T}}_{4-3}$

$\vec{P}_{-1} = \overline{\overline{T}}_{4-1} \cdot \vec{P}_{-4}$

** Example : object inspection / part inspection in automotive production system:

$\overline{\overline{T}}_{mask\_img-teach}$ ⇒ transformation matrix of mask image with respect to image coordinate system in teach situation

$\overline{\overline{T}}_{obj-img-teach}$ ⇒ Transformation matrix of object image w.r.t image cos in teach situation.

In current/runtime ($\overline{\overline{T}}_{obj-img-current}$) situation, we can get object image w.r.t image by using Hough transform/template matching operator.

but, we want mask image in current situation to inspect the product at that masked position

wanted ; $\overline{\overline{T}}_{mask-img-current}$

(1) $\overline{\overline{T}}_{mask\_img-teach} = \overline{\overline{T}}_{obj-img-teach} * \overline{\overline{T}}_{mask-obj-teach}$

(2) $\overline{\overline{T}}_{mask\_img-current} = \underbrace{\overline{\overline{T}}_{obj-img-current}}_{\text{known from template matching operator}} * \overline{\overline{T}}_{mask-obj-current}$

both are unknown but equal

$\overline{\overline{T}}_{mask\_obj-teach}$ is equal to $\overline{\overline{T}}_{mask-obj-current}$ because of rigid product (body).

$\overline{\overline{T}}_{mask-obj-teach} = \overline{\overline{T}}_{mask-obj-current}$

eqn(1) ⇒ $\overline{\overline{T}}_{mask-obj-teach} = (\overline{\overline{T}}_{obj-img-teach})^{-1} \cdot (\overline{\overline{T}}_{mask-img-teach})$.

eqn(2) ⇒ $\overline{\overline{T}}_{mask\_img-current} = \overline{\overline{T}}_{obj-img-current} * (\overline{\overline{T}}_{obj-img-teach})^{-1} * \overline{\overline{T}}_{mask\_img\_teach}$

$\bar{\bar{T}}_{2-0Teach} = \bar{\bar{T}}_{1-0\,Teach} * \bar{\bar{T}}_{2-1} \rightarrow (1) \quad \Rightarrow \bar{\bar{T}}_{2-1} = (\bar{\bar{T}}_{1-0Teach})^{-1} * \bar{\bar{T}}_{2-0Teach}$

$\bar{\bar{T}}_{2-0\,runtime} = \bar{\bar{T}}_{1-0\,runtime} * \bar{\bar{T}}_{2-1} \rightarrow (2)$

$\bar{\bar{T}}_{2-0runtime} = (\bar{\bar{T}}_{1-0\,runtime}) * (\bar{\bar{T}}_{1-0Teach})^{-1} * (\bar{\bar{T}}_{2-0Teach})$

**3D coordinate systems:-**

The pose of a 3D coordinate system cos 3 with respect to another 3D coordinate system cos 2 is

$\vec{O}_{3-2} = \begin{bmatrix} O_{3-2x} \\ O_{3-2y} \\ O_{3-2z} \end{bmatrix}$ and three rotational components $\alpha, \beta, \gamma$ which are the rotational angles around $x, y, z$ axes.

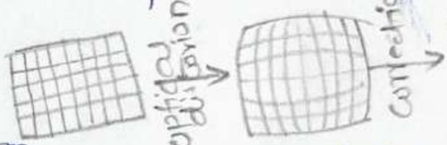
To transform a pose cos3 from one coordinate system cos2 to another coordinate system cos1, a simple matrix multiplication has to be executed

$\bar{\bar{T}}_{3-1} = \bar{\bar{T}}_{2-1} * \bar{\bar{T}}_{3-2}$ with $\bar{\bar{T}}_{2-1} = \begin{bmatrix} c\beta c\gamma & c\beta s\gamma & -s\beta & O_{2-1x} \\ (s\alpha s\beta c\gamma - c\alpha s\gamma) & (c\alpha c\gamma + s\alpha s\beta s\gamma) & s\alpha c\beta & O_{2-1y} \\ (s\alpha s\gamma + c\alpha s\beta c\gamma) & (c\alpha s\beta s\gamma - s\alpha c\gamma) & c\alpha c\beta & O_{2-1z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$

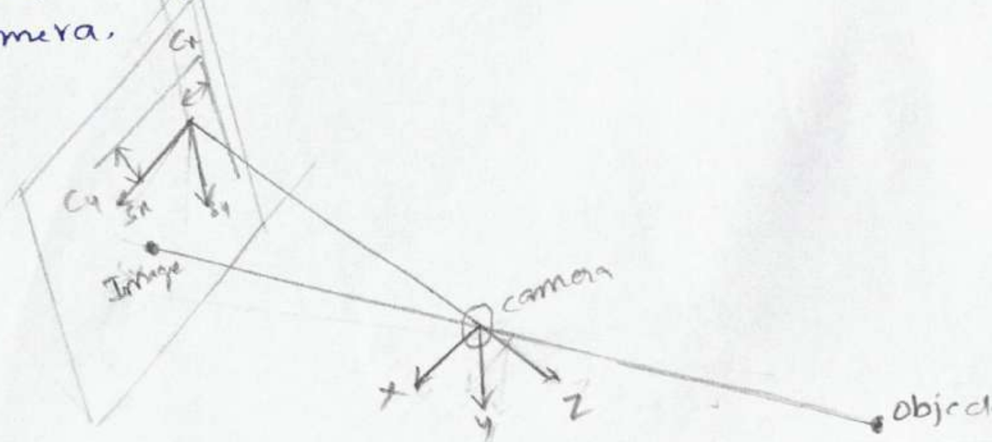
## 9. Geometric Camera Calibration

**Distortion of Lenses:-**

Generally, optical lenses cause geometric distortion of the image. The real image differs from the ideal image which would be obtained from a pinhole camera. The distortions have to be corrected for precise measuring tasks, so that the corrected image is nearly same as the image from an ideal pinhole camera.




Therefore, we have to find a correction algorithm which transform the distorted image into an image which we obtain from an ideal pinhole camera.



Template Matching :-


Averaging - for noise reduction, do averaging with radius of mask by either Arithmetic or Gaussian averaging.

Downsizing - Reduce the pixel size to compute object detection faster. With the scale value of 0 to 1, the pixel of original image is reduced.

Rectangular Mask - Take (or) crop the template image to identify that image in original image.

figure Matching : in Search Modes.

General - Direct pattern matching
Keep Maximum detection similarity threshold value as 10000 initially &
use ssp algorithm

Masking - Import the cropped (template) image from the rectangular mask. Mark reference point & brush the part of image which you would like to detect in original image.

Pattermatching - Press recalculate similarity field & measure the value of similarity at cursor position. Note the maximum value and change the threshold value in General to the noted max. value.

finally, execute the figure matching operator to detect the number of objects.

# 7. Morphology

1) Analyse Binary shapes:

Image from file → Hit-miss structuring Element → structuring Element Dilation → Image Difference

- After load the image in Image from file. Measure the blobs width & Height of pixels which you would like to detect.
- In Hit-miss-structuring element, Enter the grid extension element width and Height.

  Eg: suppose the blob object contain 4×4 (width × height) then Grid size is 6×6.

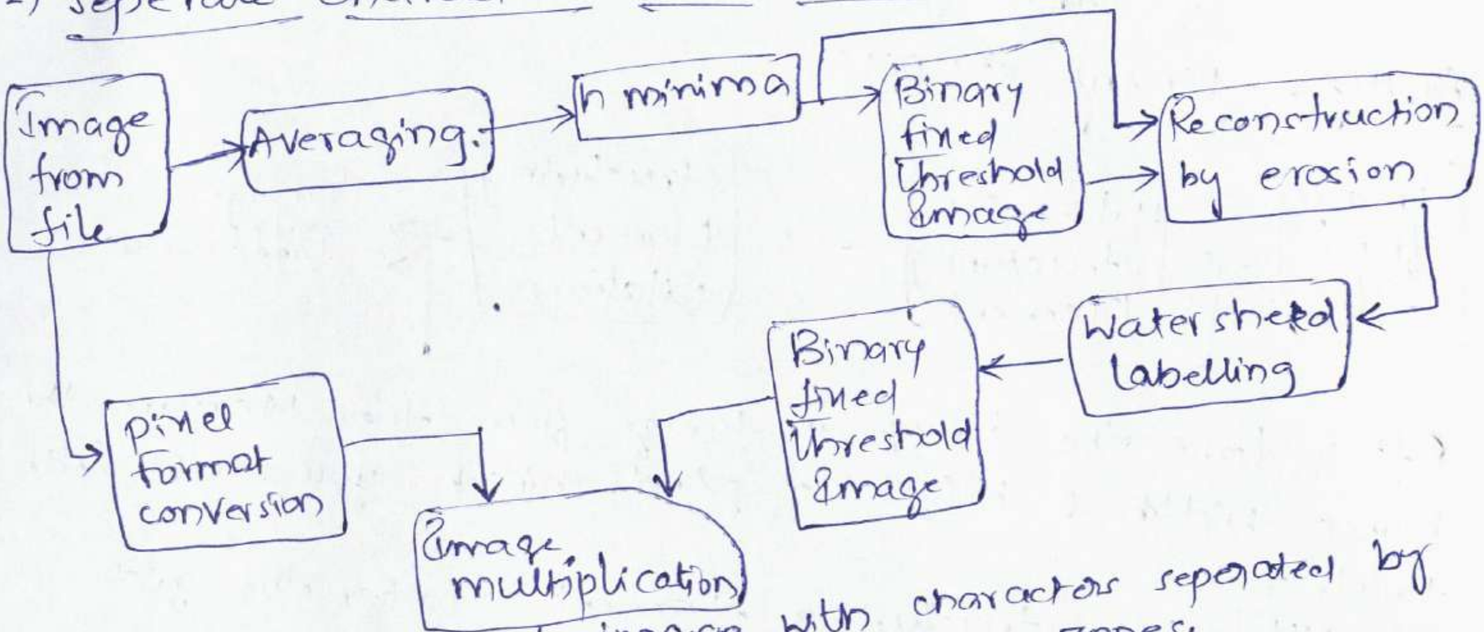
  → Mark Quantity 1 Pixels type (i.e required shape to detect in blob image) with centre/base point. Blue Color

  → Mark Quantity 2 type pixels in grey color that the background image should contain.

  the output image of HMSE operator contain less number of pixels of required identified image. To increase do dilation.
- In structuring element dilation, Give the grid size as same (opposite to/image of) detect/identified image. In this example 4×4 and brush the pixel to the image (mirror image) of original image to detect. And mark the base point
- Compare your dilated image with original image in Image difference operator by subtracting original image from dilated image.

  If you are n't able to (detect)/find the required blob to detect, then your process is correct orelse change the dilation grid base point & check.

2) Seperate characters in a Text sheet :-




o/p image with characters separated by influence zones.

Averaging → Enter radius of mask, choose required filter type & color channel for averaging (i.e reduce noise)

h-minima → Enter H - Threshold & local minimum detection threshold values such that clear visualization of characters.
H-Threshold - maximum threshold to detect characters in clear pixels.

Binary fixed threshold Image - gray colour to white & dark gray to black conversion

Reconstruction by erosion → mask image (output image of h-minima operator)
→ Marker image
Binary fixed thresholded image.

6: Location of parts :

Image from file → Averaging → Downsizing → { Rect mask / Figure matching }

(i) screw :
Averaging - radius of mask - 3 ( to avoid noises at edges)
Downsizing - scale - 0.2 to 0.3 ( to reduce pixels)
figure matching - General - search modes - Edge Matching $x, y, \alpha$
Threshold type - Absolute

Contour Matching - Parameter for Edge gradients
Trail & error - get values { upper canny Edge Threshold = UT / lower " " " = LT }
Radius for Gaussian smoothing to get smooth edges in image

- change UT & LT ⇒ by $\frac{UT+LT}{4}$ (25%) for lower Threshold
$\frac{3(UT+LT)}{4}$ (75%) for upper threshold

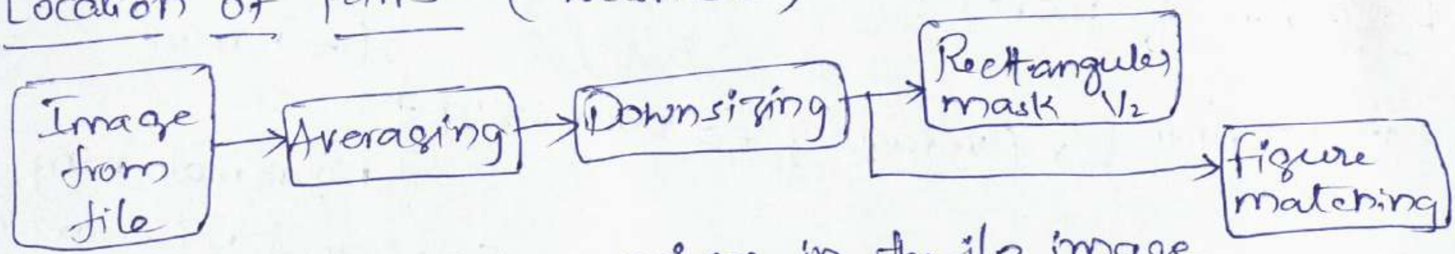
After Importing the mask image in Masking & mark Ref. point & detect edges.
To detect proper edges, change the threshold values (lower threshold & upper threshold)

- Contour Matching - parameters for pose-accumulation
change parameters to get max. voting cells (by using Generalized Hough Transform algorithm)

- Recalculate accumulator space
- calculate threshold avg. of accumulated value (High value + minimum value/2)
with the help of cursor (max. detection similarity elevation threshold in General block)

Eg:- $\frac{5.9+0.6}{2} = 3.125$

## Location of Parts (washers)




Averaging - To reduce noises in the i/p image
Use radius of mask, to reduce the noise

Downsizing - To reduce pixels size of averaged image.
Use scale value of 0 to 1, to reduce pixels.
less value gives less pixel image. (Recom. 0.2 to 0.35)

Rectangular Mask - crop the template image from the input image.

figure matching -

- Image Acquisition - Hardware
- Algorithms ⇒ Morphology, enhance Image, feature extractions

Application:-
- To localize image / object identification & mark on direction by Hough transform
- → Pose detection by using coordinate transformation
- → Surface inspection

→ Components of computer vision: screen, monitor, belt/conveyor

→ Longer $\lambda$ with less frequency

780 nm = Red

380-780 nm → Visible

Less frequency ie longer wavelength (IR, visible not dangerous)

But shorter wavelength (UV rays) of contain high freq. are dangerous to skin & cause illness

($\lambda \propto \frac{1}{freq}$)

→ A black body is body which appears black in all $\lambda$ & does not emit any ~~ad~~ electromagnetic radiation when it is cold. A black body absorbs all radiation from environment

→ Room heaters are white? why? for visible rays, white paint appear white but, for Infrared light white color appears black. A black body has 2 properties, high absorbance & emit radiations when body heated up.

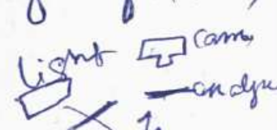
→ LED - High energy efficiency - very bright light for flash most modern source of light - Monochromatic - fluorescent substance to get other colours

for Bright flashes - uses Xenon lamps

We use flashes for moving objects.

→ Band pass filter - Red pass filter, blue pass filter & Green

→ Polarizing filter is thin transparent plastic layer of filter which allow only certain oscillations of light.




Polarizer ⇒ light source → polarizer → object

Analyzer ⇒ Object → Analyzer → Camera sensor

→ polarizer removes 50% of light intensity of diffused reflection light

→ Object appears darker at sometimes when observe through a polarizer for analysing the object

→ Diffuse light → dome shaped of white coated - light source are encapsulated on sides - reflecting light source in all direction when light rays hit dome shaped
Dome light illumination produce diffuse light to detect defect on surface if object has reflection

→ Highlight irregularities and elevations & edges in a flat region (not concave but convex body - use Dark field light (illumination)

① Soldering points, highlight screws - use Bright field illum:

② Coaxial telecentric illuminations : Sphere part highlight light passes to semi transparent mirror plane & 50% light reflect to object & 50% pass

Types of Bright field illumination
① Foccused ring light ② Coaxial telecentric illumination
→ light ray come out parallel to camera axis/optical axis

→ Diffuse Bright-field Backlight:-

→ Telecentric Bright field Backlight : Precise measurement of Telecentric Bright field illuminations - object is always silhouettes. Expensive lens illuminations - object is always smaller than lens

→ CCD used in more expensive camera, older camera & in industrial camera. Line camera with light sensitive PD which convert photons into electrons and charges are transferred to readout register. More exposure time because during exposure we can't read charges. Area CCD camera contain rows photodetector with single readout register where charges flow photodetectors to photodetectors & finally read out register. But during readout initial charges appear brighter the next coming charges appear darker. CCD

→ CMOS has each photodetector contain

[ [ ] ] Bnt each
CMOS
dynamic range

{ CMOS have high sensitivity in IR
{ CCD has " " " UV

CFA - Color filter Array - 2 G pixels but only one Red, blue pixels because Green color is in Visible system.

$\boxed{\text{2 Green pixels > double than Red \& blue pixels}}$

→ The sensor size is outer diameter of the Vidicon video tubes which is nearly 2/3 rd of outer tube diameter.

Width ≃ half the sensor size [i.e, 1 inch sensor has 12.8 mm width]

Eg:- $\frac{1}{4}$ inch size sensor has width of 3.175 mm

→ Pinhole camera : Imaged upside down on pinhole box backplane

$$\frac{h}{s} = \frac{h'}{c} \Rightarrow h' = h \cdot \left(\frac{c}{s}\right)$$

⇒ n = light speed in vacuum / light speed in medium

$n_1 \sin\alpha_1 = n_2 \sin\alpha_2 \xrightarrow{\text{linearize}} n_1 \alpha_1 = n_2 \alpha_2$ ⇒ Principle of refraction lead to Gaussian optics

→ In Gaussian optics, principal ray passes through Nodal points

$$\frac{1}{f} = \frac{1}{s'} + \frac{1}{s} \; ; \; \frac{h}{s} = \frac{h'}{s'} \; ; \; \beta = \frac{h'}{h} = \frac{s'}{s} \; ; \; \frac{h}{s-f} = \frac{h'}{f}$$

→ If Object is placed behind the focal point, then rays diverge. so virtual image appears for observer

→ Limiting elements of lens system which limits the bundle of rays pass through lens system

Aperture pupil = Aperture stop

Entrace pupil : If one look lens from the object side, we see an image of apeture pupil, it is entrance pupil

Exit pupil :- If you directly look lens from the image side, we can directly see the apeture pupil, it is called Exit pupil that image of apeture pupil can be seen from image side

→ Principal ray :

→ Border ray :

→ Gaussian optic & pinhole camera model :- Virtual shifting of image plance along the optical axis until section point of image plance & principal ray produces same image size

→ F-number : $\frac{d_{ENP}}{f'}$ = ~~diameter of entrance pupil / Image side focal lengths~~

$$\text{F-Number} = \frac{f'}{d_{ENP}} = \frac{\text{Image side focal lengths}}{\text{Diameter of entrance pupil}}$$

$$\text{Illuminance} \propto \frac{\text{d. entry area}}{\text{Image side focal lengths}} \Rightarrow \simeq E \sim \frac{E}{F^2} = E \cdot A$$

$\Rightarrow \frac{E}{2^n} \sim \frac{1}{F^2} \Rightarrow F^2 = 2^n \Rightarrow F = \sqrt{2^n}$

⇒ Image of different spaced object
Object closer to lens → Image far from lens system, Vice versa
↳ Object far to lens produce Image near to lens system
along the movement of Image plane on optical axis
If image plane is fixed, then we can't get sharp image
if you move object far or closer to lens system. This produced
as blurred image because of circle of confusion.
How much we can move the object to get circle of confusion
→ diameter of confusion is normally chose as size of one pixel

$S_f - S_n \Rightarrow S_{far} - S_{near}$ = Depth of field $= \Delta s = \frac{2s^2 f d'}{f'^2}$

→ By increase the F number, depth of field increases
(Big F number ⇒ smaller diaphragm → energy falling per time per sensor area is smaller, so we need higher exposure time)
Smaller the F-Number, bigger the dia-frame so less exposure time need so depth of field reduces

→ Photon noise – Poisson distributed
- Amplifier noise
- Reset noise (incomplete reset)
- dark current noise (due to thermal excitation)

SNR – Signal to noise ratio
SNR ∝ light intensity

↳ temporal noise → depend on time & related to one pixel
Spatial/pattern noise – caused by manufacturing process
which cause systematical error – eliminated by calibration process
→ offset noise
↳ Gain noise

→ Standard lens – Homocentric lens i.e. image size changes when move the object closer or far away from the lens.
→ Telecentric lens – Image size is same even if we move the object closer or farer from the lens system with the help of small aperture stop centered in image side focal (length) point F'. We can acheive this.
→ Bilateral telecentric: $F_1' = F_2$ in two lens system with 1 telecentric lens to get sharp image of same size even if object is move far/closer to lens.

** → In telecentric lens system, lens must be (20%) larger than object size. (diameter of lens size > object size)

→ Spherical Aberrations: Caused by spherical shape of lens

→ Astigmatism: Tangential & sagittal light rays do not intersect

→ Chromatic Aberration: Different bending angles from diff. $\lambda$

→ Vignetting: Image appears darker due to less light energy travel from top of object compared to axis position.

→ Morphology - Theory of pixel quantities and forms.

→ Structuring element Erosion - Consider minimum pixel value with Base point

→ structuring element Dilation - Consider maximum pixel value

→ Gradients = Dilation (of original) image - Eroded original image
↳ simple edge extractor algorithm.
Beucher Gradient = Dilated image - Eroded image
Internal half " = Original image - Eroded "
External " " = Dilated " - Original image

→ Morphological opening ⇒ $\gamma_B(f)$ = Dilation of eroded image
i.e, first erode original then dilate it

→ Morphologic closing ⇒ $\phi_B(f)$ = Erode ~~an~~ dilated image
i.e dilate original image and then erode the o/p dilated image

→ White Top Hat = Original image - Morphologic opening

→ HMT (Hit or Miss transform): Two quantities strulating elements ~~at cutter~~
output is set to high or 1 if two quantities matches fully at current pixel position. $HMT_B(X) = \varepsilon_{B1}(X) \cap \varepsilon_{B2}(X^c)$

→ Thinning ⇒ original image - Hit or miss transformed image
Homotopic thinning - sequential thinning with 2 structuring element in 90° rotational angle.

→ Pruning - Sequential thinning till idempotence (stability) is called skeletonizing

→ Debearding $(X \circ B)^\infty$ ; Pruning $(X \circ B)^n$

→ Geodesic transformation: Marker Image - to which morphologic trans. applied
mask image that transformed image is placed above/below the mask
↳ acts as a mask
& mask image is forced

→ Geodesic dilation: Marker dilated below of it.
the dilated image under/remain
Mask size > Marker size

→ Geodesic Erosion: Marker image is eroded & result eroded image is forced to above mask image. Here, Marker size > Mask size
**

→ Reconstruction by dilation: Continue Geodesic dilation process untill / till it reaches stability.

→ Reconstruction by erosion: continue a Geodesic erosion operation to an original image until it reaches stability

→ Regional minima: Reconstruction by erosion operation to an original image by incremented one pixel & subtract the result from original image i.e. $RMIN(f) = R_f^{\varepsilon}(f+1) - f$

Regional maxima: $RMAX(f) = f+1 - R_f^{\delta}(f)$

→ H-minima: Original image is added by h pixels and do Geodesic erosion operation for h added image and get the ~~o/p~~ output image is difference with original image to get h minima. i.e. $R_f^{\varepsilon}(f+h) - f$

→ Watershed Transformation:- Filling water in water basin through not in basin ~~8 connected pixels are wider than~~ 4 connected pixels

→ Temporal Averaging:- $g_{r,c} = \frac{1}{n}\sum_{i=1}^{n}\hat{g}_{r,c}$

→ Spatial Arithmetic Averaging $= g_{r,c} = \frac{1}{(2n+1)(2m+1)}\sum_{i=-n}^{n}\sum_{j=-m}^{m}\hat{g}_{r-i,c-j}$

→ Arithmetic Averaging - can compute faster because it is seperable & recursive. But this is directional dependent, so artefact resulted image. (not readable)

→ Gaussian Averaging - Causes rotational invariant result i.e. directional independent, so result image is clear & readable. But time consuming algorithm because it is seperable not recursive

→ Seperable filter:- Instead of $(2m+1)(2n+1)\,\boxed{h}$ → image size operations, we can do $(2m+2n+2)$ with operations only if filter is seperable.

→ Recursive filter:- Next pixel can be calculated from result of previous pixel by adding the result of next boundary pixel and subtract the result of previous boundary pixel

→ Linear filter: Apply filter to linear combination of 2 i/p images yields same result as apply filter to two images seperately & compute the linear combination. $h(f+g) = h(f) + h(g)$

→ Convolution filter:- $(f * h)(r,c) = \int_{r=-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)\, h(u-r, v-c)\, du\, dv$

$f * h = \sum_{j=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} f_{i,j} \cdot h_{i-r,j-c}$

→ Gaussian filter: $h_{\sigma}(u,v) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(\frac{-u^2-v^2}{2\sigma^2}\right)$

Gaussian filter gives better smoothing since it is direction independent; [frequency response is similar] whereas in arithmetic case, frequency response is not similar

→ Convolutional filter (Kernel) for detecting edges / highlighting gradients

3x3 Kernel is

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

sobel y-filter

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

sobel x-filter

Dark (upside) edge on left side +ve value gives dark edges

→ smoothing filter

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$ 3x3

Multi factor = 0.11

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

5x5 mask

multiplication factor is $\frac{1}{25} = 0.04$

→ To highlight diagonal gradients, use diagonal filter

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

Here multiplication factor is = 0.25

→ Convolution (ie, smoothing) & subsequent derivation is given as $(f*h)'$ which is identical to $f*h'$ (only one operation for smoothing & derivation)

→ Gaussian filter is best suited one for edge detection.

→ Subpixel precision

→ Edge follower for Extracting Edges (Canny Edge follower) - More robust algorithm. Define an entry & exit threshold and if signal passes over entry threshold, the edge following is set to active until the signal pass under exit threshold.

Entry threshold > Exit threshold

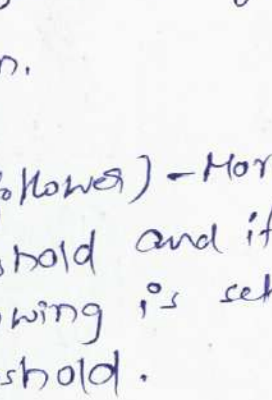
→ Coordinate Transformation:

COS 3 w.r.t COS 2 ⇒ translation vector $\vec{O}_{3-2} = \begin{bmatrix} O_{3-2x} \\ O_{3-2y} \end{bmatrix}$

Rotational vector $\alpha_{3-2}$

Rotational matrix $= \begin{bmatrix} e_{x_{3-2}} & e_{y_{3-2}} \end{bmatrix}$

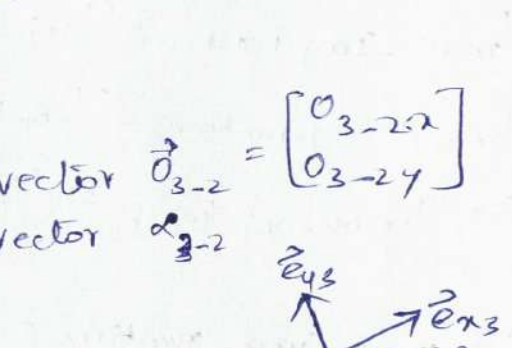
$$= \begin{bmatrix} \cos\alpha_{3-2} & -\sin\alpha_{3-2} \\ \sin\alpha_{3-2} & \cos\alpha_{3-2} \end{bmatrix}$$



→ $\vec{O}_{3-1} = \vec{O}_{3-2} + \vec{O}_{2-1}$ & $\alpha_{3-1} = \alpha_{3-2} + \alpha_{2-1}$

$\vec{O}_{3-2} = O_{3-2x}\,\vec{e}_{x2} + \vec{O}_{3-2y} \cdot \vec{e}_{y2}$

$= \vec{O}_{3-2x}(e_{x2-1x} + e_{x2-1y}) + \vec{O}_{3-2y}(e_{y2-1x} + e_{y2-1y})$

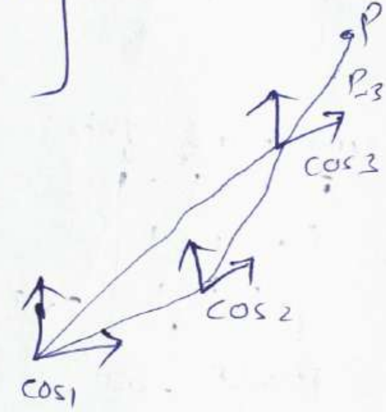


Given: $\overset{=}{T}_{2-1}$

Wanted: $\vec{P}_{-1}$

$$\vec{P}_{-1} = \overset{=}{T}_{2-1} \cdot \vec{P}_{-2} = \begin{bmatrix} \cos\alpha_{2-1} & -\sin\alpha_{2-1} & \vec{O}_{2-1x} \\ \sin\alpha_{2-1} & \cos\alpha_{21} & O_{2-1y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{x-2} \\ P_{y-1} \\ 1 \end{bmatrix}$$

$$\vec{P}_{-1} = \begin{bmatrix} P_{x-1} \\ P_{y-1} \\ 1 \end{bmatrix} = \begin{bmatrix} e_{x2-1x} & e_{y2-1x} & O_{2-1x} \\ e_{x2-1y} & e_{y2-1y} & O_{2-1y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{x-2} \\ P_{y-2} \\ 1 \end{bmatrix}$$




Given, $\bar{\bar{T}}_{3-2}, \bar{\bar{T}}_{2-1}, \vec{P}_{-3}$

$\vec{P}_{-1}$ wanted ?

$$\vec{P}_{-1} = \bar{\bar{T}}_{2-1} \cdot \underbrace{\bar{\bar{T}}_{3-2}\, \vec{P}_{-3}}_{\vec{P}_{-2}} \rightarrow \bar{\bar{T}}_{2-1}\, \vec{P}_{2}$$

$$\vec{P}_{-1} = \bar{\bar{T}}_{3-1} \cdot \vec{P}_{-3}$$

Suppose ; Given $\bar{\bar{T}}_{3-2}, \bar{\bar{P}}_{2-1}, \vec{P}_{-2}$ what $\vec{P}_{-1}$ and $\vec{P}_{-3}$

$\bar{\bar{T}}_{4-1} = \bar{\bar{T}}_{2-1} \cdot \bar{\bar{T}}_{3-2} \cdot \bar{\bar{T}}_{4-3}$

$T_{2-1} = T_{4-1} \cdot T_{4-3}^{-1} \cdot T_{3-2}^{-1}$

$\vec{P}_{-1} = \bar{\bar{T}}_{2-1} \cdot \vec{P}_{-2}$ ; $\vec{P}_{-2} = \bar{\bar{T}}_{3-2} \cdot \vec{P}_{-3}$

** $\vec{P}_{3} = \bar{\bar{T}}_{3-2}^{-1} \cdot P_{-2}$

$T_{4-1} = T_{2-1} \cdot T_{3-2} \cdot T_{4-3}$

$T_{2-1} = T_{4-1} \cdot T_{4-3}^{-1} \cdot T_{3-2}^{-1}$ and $T_{3-2} = T_{2-1}^{-1} \cdot T_{4-1} \cdot T_{4-3}^{-1}$

⇒ $T_{mask-Img-Teach} = T_{obj-img-Teach} \cdot T_{mask-obj-Teach}$ → (1)

is equal

$T_{mask-img-runtime} = T_{obj-img-runtime} \cdot T_{mask-obj-runtime}$ → (2)

(1) ⇒ $T_{mask-obj-Teach} = (T_{obj-img-Teach})^{-1} \cdot T_{mask-img-Teach}$

(2) ⇒ $T_{mask-img-runtime} = T_{obj-img-runtime} (T_{obj-img-Teach})^{-1} (T_{mask-img-Teach})$

⇒ In ~~B~~ 3D coordinate transformation, axes must be rotated about original coordinate systems.

→ Template matching, Hough transform, blob detection are used to find & localize the objects

→ Template matching algorithms is SAD, SSD, NCC

$SAD(r,c) = \frac{1}{n} \Sigma |t(u,v) - f(u+r, c+v)|$ ; $SSD(r,c) = \frac{1}{n} \Sigma [t(u,v) - f(r+u, v+c)]^2$

→ In template matching, Region of interest is used to reduce the computation time (with original image)

→ SSD & SAD is not good for bad illumination (or high brightness sensitivity

→ $ncc(r,c) = \frac{1}{n} \Sigma \left( \frac{t(u,v) - m_t}{\sqrt{s_t^2}} * \frac{f(r+u, c+v) - m_f(r,c)}{\sqrt{s_f^2}} \right)$

→ Generalized Hough transformed is used to detect edge, line, circle

Template matching / Pattern Matching is only X, Y

GHT is used to detect X, Y, $\alpha$ (rotational angle)

→ We use $\phi$-d coordinate system to represent a line.

i.e A line in X-Y plane is a point in $\phi$-d domain