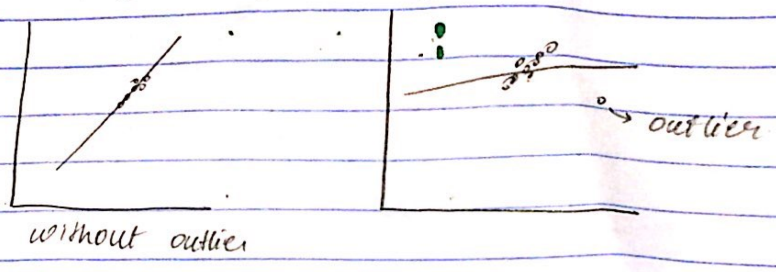
1 a) Outliers are points, which behave completely different with the existing points.

During line fitting, because of outlier it presented a completly wrong solution.



1b) Objective function for Robust Estimation:-

$$E(\theta) = \sum_{i=1}^{n} \rho_{\sigma} (d(x_i, \theta))^2$$

in standard least square → we use $\rho = x^2$

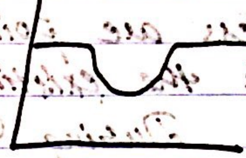
where individual error is squared

whereas, in robust estimation
we use $\rho_{\sigma} = \frac{x^2}{x^2 + \sigma^2}$ → called Geman-McClure function

So, in robust if error is big we give weight '1', whereas in least square, we square all the error & sum it

1c) Geman McClure function:-

$$\rho_\sigma = \frac{x^2}{x^2+\sigma^2}$$



The advantage here is that;

→ due to straight line before & after the curve, the maximum weight a outlier can get is 1.

if $x >> \sigma$
$\rho_\sigma = 1$

if $x << \sigma$
$\rho_\sigma = \frac{x^2}{\sigma^2}$

if σ is small, too less points are considered and σ is large, too many points are selected thus giving wrong answers, for both the cases.

~~1d) since large σ gives considers many~~

Since small σ considers less points & large sigma considers too many points. Right σ is estimated by considering a large sigma then iteratively reduce it. and

σ at iteration n → $\sigma_n = 1.5 * \text{median}(d(x_i, \theta))$ till parameters donot change.

1) Draw a large subset of points uniformly at random.
2) Fit the model using robust estimation. ⇒ θ_n

1 d) Principle of RANSAC Algorithm.

: We try to do many small experiments and hope one of them is a right model.

The number of points draon at each attempt should be small, possibility ~~to~~ get inliers and not outliers is high, ~~so we are~~ so high hope to get one good experiment.

1 e) Parameters:-

n = no. of points to draw at each evaluation
d = minimum no. of points needed.
K = no. of trials
t = distance to identify inliers.

K can be estimated as:-

$$K = \frac{\log(1-p)}{\log(1-w^n)}$$

where
w = Probability that a point is an inlier (Initial guess $w = 0.5$)
$1-p$ = Probability that experiment fail

1f) Objective of the segmentation:-

The main objective of segmentation, is to seperate object from background.

Merge approaches :- Here, we start with each pixel in seperate cluster iteratively merge clusters

split approaches :- start with all in one cluster then iteratively split clusters (by seeing distance)

1g) <u>K- means,</u>

- Select k.
- Start with initial guess of k means $\{m_i\}_{i=1}^{k}$
- repeat: $l_i = \underset{j \in [1,k]}{\operatorname{argmin}} ||f_i - m_j||$

for each pixel we chose cluster with closest centre.

- recompute means $m_j = \frac{\sum_{i=S_j} f_i}{\#\text{elements of } S_j}$ when $S_j \rightarrow$ all the pixels labelled l_j.

- Stop when m_j donot change.

K-means can be improved by using <u>mixture of gaussians</u>:-

This is similar to k-means, but here we replace

$d = ||f_i - m_j||^2$ with

with $d = (f_i - m_j)^T \Sigma_j^{-1} (f_i - m_j)$

where

$$m_j = \frac{\sum_{i=S_j} f_i}{\# S_j}, \quad \Sigma_j = \frac{\sum_{i \in S_j} (f_i - m_j)(f_i - m_j)^T}{\# S_j}$$

So a point in one (P_1) in C_1 will be included now in cluster C_1 than C_2 even if distance from centre of C_2 is small because variance in this direction (C_1) is more, thus this example will be success in mixture of Gaussian but fail with k-means.

1 h) **Mean Shift**

This algorithm is similar to 'K' means, with a minor change here:-

$$m_j = \frac{\sum_{i \in S_j} f_i}{\# S_j} \longrightarrow m_j = \frac{\sum_{i \in S_j} w(f_i - m_j) f_i}{\sum_{i \in S_j} w(f_i - m_j)}$$

in K means — in mean shift

Here → weight is given to each example based on distance of sample from m_j.

→ function 'w' - determines value of weight.
when far from center → low w value.

→ here cluster centre are peaks of histogram.

2a) Camera Calibration

we find ~~3D~~ 2D point for image from point in 3D-world point

equation ($\boxed{p = MP}$) — (1).

if we are given P (3D point) & M (projection matrix)

In reconstruction, → given 2D → we find 3D.

Reconstruction is more difficult, as we have to add an unknown dimension here whereas in other cases we simply remove it which is easy.

2(b) The inputs for camera calibration are:-

→ 2D image points (x_i, y_i)

→ and 3D ~~in~~ world points (X, Y, Z).

2(k) STEPS in the non-coplaner calibration algorithm:-.

(i) Estimate projection matrix, M

(ii) Find intrinsic & extrinsic parameters (K^*, R^*, T^*)

$$\underset{2D}{p_i} = M P_i$$

$$\begin{bmatrix} x_i \\ y_i \\ w'_i \end{bmatrix} = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} P_i \rightarrow x_i = \frac{x_i}{w'_i} = \frac{m_1^T P_i}{m_3^T P_i} \quad -(1)$$

$$y_i = \frac{y_i}{w'_i} = \frac{m_2^T P_i}{m_3^T P_i} \quad -(2)$$

To solve these we need min '6' points.

$$\underbrace{\begin{bmatrix} P_i^T & 0 & -x_i P_i^T \\ 0 & P_i^T & -y_i P_i^T \\ \vdots & \vdots & \vdots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}}_{\text{this is } M} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

we can also find M by SVD of A →

→ so, now

$$K^*[R^* | T^*] = \rho \hat{M}$$

$$M = \begin{bmatrix} a_1^T & \\ a_2^T & b \\ a_3^T & \end{bmatrix}$$

so $\underset{3\times3}{K^* T^*} = \rho \underset{3\times3}{\begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}}$

$$\underset{3\times1}{K^* T^*} = \rho \underset{3\times1}{b}$$

→ solving we get - $|\rho| = \frac{1}{|a_3^T|}$

$$u_0 = \rho^2 a_1 \cdot a_3$$

$$v_0 = \rho^2 a_2 \cdot a_3.$$

2(a) - Projection Matrix = $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

$P_i = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} \Rightarrow$ homogenous form.

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

$= \begin{bmatrix} 18 \\ 14 \\ 7 \end{bmatrix} \Rightarrow$ so $2D = \begin{bmatrix} 18/7 \\ 14/7 \end{bmatrix} = \begin{bmatrix} 18/7 \\ 2 \end{bmatrix}$

2DH

2 e) world point = (1, 2, 3) ⇒ (X, Y, Z)
image point = (100, 200) ⇒ (x_i, y_i)

so for matrix A :-

$$\begin{bmatrix} X, & Y, & Z, & 1, & 0, & 0, & 0, & 0, & -x_iX, & -x_iY, & -x_iZ, & -x_i \\ 0, & 0, & 0, & 0, & X, & Y, & Z, & 1, & -y_iX, & -y_iY, & -y_iZ, & -y_i \end{bmatrix}$$

$$= \begin{bmatrix} 1, & 2, & 3, & 1, & 0, & 0, & 0, & 0, & -100, & -200 & -300 & -100 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 & -200 & -400 & -600 & -200 \end{bmatrix}$$

2 f) The minimal number of points required is '6' to able to find unique solution.

For to obtain solution,

$$Mx = 0$$

where

$$M = \underbrace{\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & -x_iX, & -x_iY & -x_iZ & -x_i \\ 0 & 0 & 0 & 0 & X & Y & Z & -y_iX & -y_iX & -y_iZ & -y_i \end{bmatrix}}_{\text{for 1 point}}$$

for all points this is find

$$Mx = 0$$

now take svd for M.
then obtain vector V → which is 12x12 matrix.

V^T → last column will provide 12 values
↳ will provide reshape provide projection vector.

as:-

$$\begin{bmatrix} p^T & 0 & -x\ p_i^T \\ \vdots & & \\ 0^T & p_n^T & -y_i p_i^T \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

12×1 $\quad$ $2n \times 1$

this ~~will~~ will provide projection matrix after reshaping

g) To find the camera parameters from ~~projection~~ "M" matrix:-
we will find SVD of projection matrix ($Mx = 0$)
Then for $V \rightarrow$ matrix $\rightarrow V^T \rightarrow$ the last column will provide projection matrix.

$$P = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{44} \end{bmatrix}$$

'P' also divided as:-

$$P = K^* \times [R^* | t^*]$$

where $K^* R^* = \begin{matrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{matrix}$, $K^* t^* = \begin{matrix} m_{14} \\ m_{24} \\ m_{44} \end{matrix}$

Q h) To compute the quality of projection matrix 'M'.

$$\{p_i\}_{i=1}^{n} \longrightarrow \{P_i\}$$

image point (2DH) — world point (3DH)

For the camera calibration, we will compute parameters K^*, R^*, T^*, & will find error, with the correct one. like,

$$E_m(K^*, R^*, T^*) = \sum_{i=1}^{n} \left(x_i - \frac{m_1^T P_i}{m_3^T P_i}\right)^2 + \left(y_i - \frac{m_2^T P_i}{m_3^T P_i}\right)^2$$

smaller the error, → shows better quality of projection matrix
smaller the distance → smaller the error.

Q i) Principal of Planer Camera Calibration :-

Here, ~~to~~ in order to perform camera calibration minimum of three times we have show the views, for processing.

→ Estimation of 2D homography, between calibration target and image

→ Calculate intrinsic parameters from different views & then calculate extrinsic parameters.

Non-coplaner	Planer
→ use 3D image for calibration target, which is in more than one plane. we calculate pixel coordinate (image point) & based on these camera calibration was done	→ Here a single image (planer points) is used with different views, to be used for camera calibration.

2j) Homography (H) → 2D projective map is used for transformation from one plane to other, it is (3×3)
whereas Projection matrix (M) is used to project 3DH points to get image 2DH points it is (3×4)

→ $x' = H x$

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \underset{(3\times3)}{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

(Homography)

→ $p_i = M P_i$

$$\underset{2DH}{\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}} = \underset{(3\times4)}{\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}} \underset{3DH}{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}$$

→ $P_i = K^*[R^*|T^*]\underline{P}_i$

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = K^*[\lambda_1\,\lambda_2\,\lambda_3\,T^*]\begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix}$$

$$= \underbrace{K^*[\lambda_1\,\lambda_2\,T^*]}_{\text{2D Homography}} \underset{2DH}{\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}}$$