Kanishk Jain

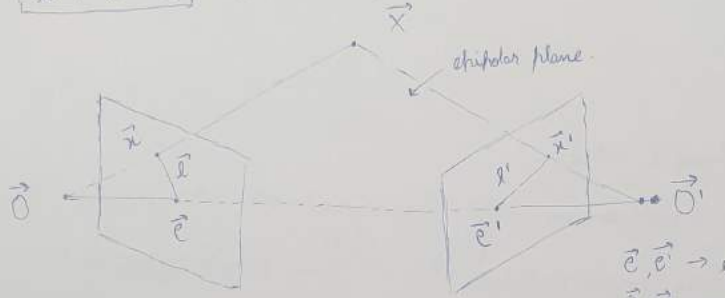
Mobile Robotics . End Sem

Q1

(a) Let $\vec{x}$ and $\vec{x}'$ be the pixel coordinates in two cameras for some world-point $\vec{X}$, and let $\vec{e}$ and $\vec{e}'$ be epipole of second camera seen first camera's image and epipole of first camera seen in second camera's image, respectively.

We have,

$\boxed{\vec{x}'^T F \vec{x} = 0} \rightarrow$ ① [F is the fundamental matrix between image in Camera 1, $\vec{x}$ and image in Camera 2, $\vec{x}'$]



$\vec{e}, \vec{e}' \rightarrow$ epipoles.
$\vec{O}, \vec{O}' \rightarrow$ Camera centers
$\vec{x}, \vec{x}' \rightarrow$ image ~~coordinates~~ coordinates of $\vec{X}$
$\vec{l}, \vec{l}' \rightarrow$ epipolar lines.
$\vec{X} \rightarrow$ world point

* Now, $\vec{x}'$ corresponding to $\vec{x}$ must lie on epipolar line $\vec{l}'$, ie:-

$\boxed{\vec{x}'^T \vec{l}' = 0} \rightarrow$ ②

from ① and ②, we have

$\boxed{\vec{l}' = F\vec{x}} \longrightarrow$ ③

* from figure, we can see that $\vec{e}'$ also lies on $\vec{l}'$.

$\therefore \ \vec{e}'^T \vec{l}' = 0$

$\Rightarrow \boxed{\vec{e}'^T F \vec{x} = 0} \rightarrow$ ④

from ③, $F\vec{x} = \vec{l}'$, therefore $F\vec{x} \neq 0$.

$\therefore \boxed{\vec{e}'^T F = 0}$

* 11^{ly}, we have $\vec{l}^T x = 0$ and $\vec{l}^T e = 0$

from ① $\Rightarrow \vec{l}^T = \vec{x}'^T F$

$\therefore \ \vec{l}^T e = \vec{x}'^T F e = 0$

Now, $\vec{x}'^T F = \vec{l}^T \neq 0$

$\therefore \boxed{Fe = 0}$

(b) Fundamental matrix F, b/w images I_1 (cam 1) and I_2 (cam 2) satisfies the following equation,

$$\boxed{I_1^T F I_2 = 0}$$

Taking transpose on both sides,

$$\boxed{(I_1^T F I_2)^T = I_2^T F^T I_1 = 0}$$

$\therefore$ F^T is the fundamental matrix between I_2 and I_1.

* Fundamental matrix depends on the order in which camera pairs are taken. Specifically,

  $F = [e']_\times P' P^+$, where P and P' are camera matrices with, P^+ being psuedo inverse of P.

  As a result fundamental matrix changes as the order of camera changes.

(c)

Essential Matrix	Fundamental Matrix
* Computed for calibrated cameras	* computed for uncalibrated cameras
* Contains 5 degree of freedoms	* Contains 7 degree of freedoms
* points are in normalized image coordinates	* points are in pixel coordinates

→ Essential Matrix E, and Fundamental Matrix F are related by $\boxed{E = \vec{K'} F \vec{K}}$, where K and K' are camera calibration matrices.

Q2

(a) Let x_{i_1} be the pixel location in I_1 frame and x_{i_2} be the pixel location in I_2 frame, both corresponding to point X in world frame

- Assume that (I_2 frame) is obtained by rotating (I_1 frame) by rotation matrix R, without any translation.

We have,

$$x_{i_1} = K[I|0]X \quad \Big\} \text{ K is camera calibration matrix}$$

$$x_{i_2} = K[R|0]X$$

$$= KR[I|0]X$$

$$= KRK^{-1}K[I|0]X$$

$$\boxed{x_{i_2} = KRK^{-1}x_{i_1}}$$

$$\boxed{x_{i_2} = Hx_{i_1}}, \quad \boxed{H = KRK^{-1}} \rightarrow \text{Homography relation for pure Rotation}$$

* Homography relation doesn't hold when camera translation is involved.
Suppose, I_2 frame is rotated by R and translated by t w.r.t I_1 frame.
We have, $x_{i_2} = KR[I|-t]X$ which cannot be written as multiple of $K[I|0]X$ because of additional $-t$, translation term.

(b) Following ways are used to relate pixel x_{i_1} in frame I_1 to pixel x_{i_2} in frame I_2,

① moving frame I_2 along principal axis of I_1

$$x_{i_1} = K[I|0]X$$

$$x_{i_2} = K'[I|0]X = K'K^{-1}K[I|0]X = K'K^{-1}x_{i_1}$$

$$\therefore \boxed{x_{i_1} = K'K^{-1}x_{i_1}}$$

② I_1 and I_2 correspond to two calibrated cameras

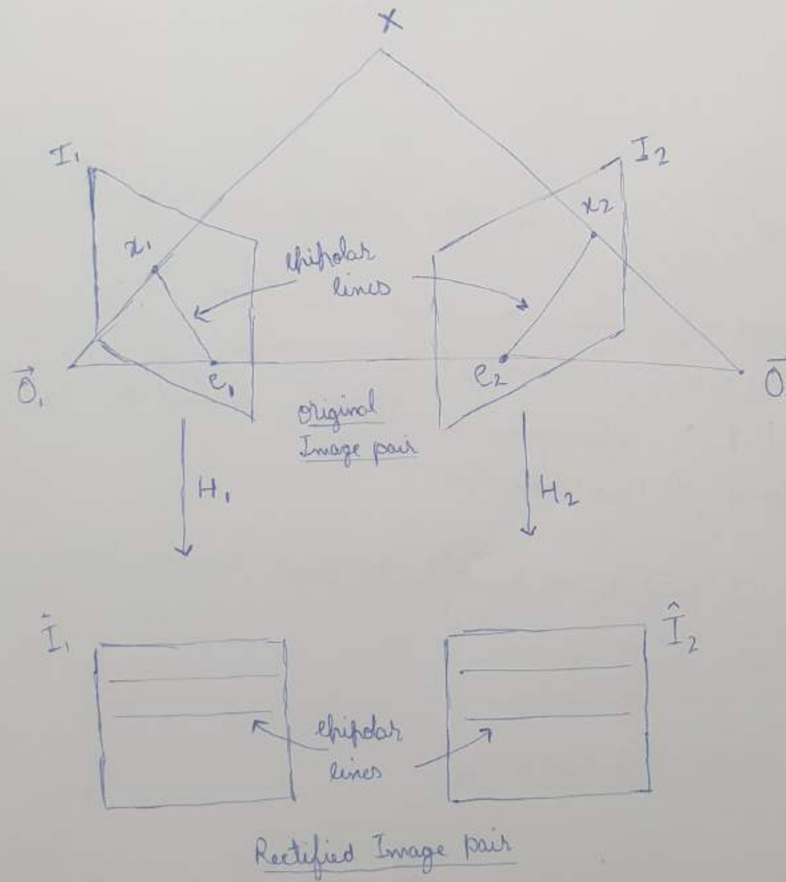
$$\boxed{x_{i_1}^T E x_{i_2} = 0}$$

③ I_1 and I_2 correspond to two uncalibrated cameras

$$\boxed{x_{i_1}^T F x_{i_2} = 0}$$

(c) * Two homographies involved in Stereo Rectification are the two rectification homographies, one for each input image. Specifically, given images I and I', we have homography $H: I \rightarrow \hat{I}$ and $H': I' \rightarrow \hat{I}'$, where $\hat{I}$ and $\hat{I}'$ are rectified images synthesized from rotationally aligned cameras with co-linear/co-incident principal axis.

* Each homography maps the epipolar lines in the original image pairs to horizontally aligned epipolar lines in transformed image pairs.
* H and H' are structure preserving mappings that modifies the orientation of epipolar lines, ie:- H & H' are isomorphism of image spaces, hence they are homographies.



$$\boxed{\begin{aligned} H_1 e_1 &= [1, 0, 0]^T \\ H_2 e_2 &= [1, 0, 0]^T \end{aligned}}$$

Q3

In DLT algorithm, we are given 3D-2D correspondences, and the goal is to estimate the Camera intrinsic and extrinsic parameters or Camera transformation matrix P.

$$\vec{x} = P\vec{X} = KR[I \mid -X_0]\vec{X}$$

$K \rightarrow$ intrinsic camera matrix
$R \rightarrow$ rotation matrix
$X_0 \rightarrow$ camera optical center location
$P \rightarrow$ Camera transformation matrix.

$$\vec{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow (1)$$

Here, P is a 3×4 matrix with 11 unknown parameters because of homogeneous coordinates.

→ Now, each $\vec{x}-\vec{X}$ correspondence results in two equations,

$$\vec{x}_i = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \vec{X}_i$$

$A^T = (P_{11}\ P_{12}\ P_{13}\ P_{14})$
$B^T = (P_{21}\ P_{22}\ P_{23}\ P_{24})$
$C^T = (P_{31}\ P_{32}\ P_{33}\ P_{34})$

$$\vec{x}_i = \begin{bmatrix} A^T \\ B^T \\ C^T \end{bmatrix} \vec{X}_i$$

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \begin{bmatrix} A^T\vec{X}_i \\ B^T\vec{X}_i \\ C^T\vec{X}_i \end{bmatrix} \xrightarrow[\text{coordinates}]{\text{Homogeneous}} \begin{bmatrix} u_i/w_i \\ v_i/w_i \\ 1 \end{bmatrix} = \begin{bmatrix} A^T\vec{X}_i / C^T\vec{X}_i \\ B^TX_i / C^TX_i \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{A^TX_i}{C^TX_i} \\ \frac{B^TX_i}{C^TX_i} \\ 1 \end{bmatrix} \Rightarrow \left.\begin{matrix} x_iC^TX_i - A^TX_i = 0 \\ y_i^TC^TX_i - B^TX_i = 0 \end{matrix}\right\} \rightarrow \text{~~Taking transpose on both sides~~}$$

↓ Taking transpose on both sides.

$$-X_i^TA + x_iX_i^TC = 0$$
$$-X_i^TB + y_iX_i^TC = 0$$

We have,

$$(-X_i^T,\ 0^T,\ x_iX_i^T)\begin{pmatrix} A \\ B \\ C \end{pmatrix} = 0 \rightarrow (2)$$

$$(0^T,\ -X_i^T,\ y_iX_i^T)\begin{pmatrix} A \\ B \\ C \end{pmatrix} = 0 \rightarrow (3)$$

from (2) and (3),

$$\underbrace{\begin{bmatrix} -X_i & -Y_i & -Z_i & -1 & 0 & 0 & 0 & 0 & x_iX_i & x_iY_i & x_iZ_i & x_i \\ 0 & 0 & 0 & 0 & -X_i & -Y_i & -Z_i & -1 & y_iX_i & y_iY_i & y_iZ_i & y_i \end{bmatrix}}_{2\times12} \underbrace{\begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{24} \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix}}_{12\times1} = 0 \rightarrow (4)$$

* Given at least 6 observation points, we will have 12 equations with 11 P_{ij} unknown parameters.

We have, $A_{2M\times 12}\, P_{12\times 1} = 0$, for M observation points,

$$A_{2M\times 12} = \begin{pmatrix} -X_1^T & 0 & x_1 X_1^T \\ 0 & -X_1^T & y_1 X_1^T \\ & \vdots & \\ -X_M^T & 0 & x_M X_M^T \\ 0 & -X_M^T & y_M X_M^T \end{pmatrix}$$

* Solution to $A_{2M\times 12}\, P_{12\times 1} = 0$ is given by applying SVD to A and taking the eigenvector corresponding to smallest eigen value as the solution for P.

$$SVD(A) = U S V^T, \quad \boxed{P = \text{last column of } V^T}$$

* If all correspondences lie on a plane, then $rank(A) < 11$.

Suppose, we have $Z_i = 0$ for all correspondences,

$$\text{from } (4) \longrightarrow \begin{bmatrix} -X_i & -Y_i & 0 & -1 & 0 & 0 & 0 & 0 & x_i X_i & x_i Y_i & 0 & x_i \\ 0 & 0 & 0 & 0 & -X_i & -Y_i & 0 & -1 & y_i X_i & y_i Y_i & 0 & y_i \end{bmatrix}$$

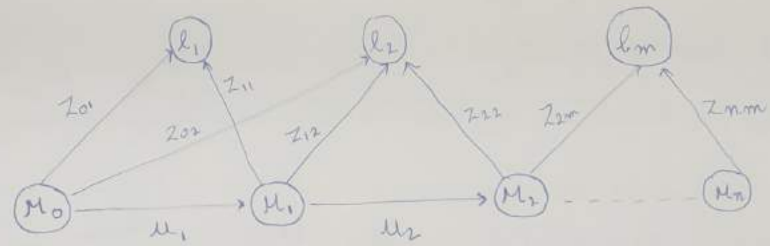
So, three of the columns are 0, we won't be able to solve for the 11 parameters of P.

* most of the times we have more equations than number of unknowns, as a result $Ap = b \neq 0$.

We want to keep b as small as possible, this is achieved by choosing P as the eigenvector of A corresponding to the smallest eigenvalue of A.

Q4

(a)



$\mu_i \rightarrow$ robot's ~~pose~~ state in $SE(2)$, i.e. $\mu_i = [\mu_{x_i}, \mu_{y_i}, \mu_{\theta_i}]^T$

$\{l_1, l_2, \ldots l_m\} \rightarrow$ Landmarks

$\{u_1, u_2 \ldots u_{n-1}\} \rightarrow$ Control inputs

$z_{ik} \rightarrow$ observation of landmark l_k from μ_i robot state.

$\rightarrow$ The goal of Smoothing and Mapping (SAM) is to estimate the robot state $\hat{\mu}_i$ and landmark states $\hat{l}_j$ given the observations $\{z_{ik}\}$ and control inputs $\{u_i\}$

* We can obtain the estimate of robot's state using a motion model by utilizing the control inputs,

$$\boxed{\hat{\mu}_{i+1} = f(\hat{\mu}_i, u_{i+1})} \rightarrow ①$$

- μ_{i+1} is the observation from odometry

[f is the motion model, $\hat{\mu}_{i+1}$ is the estimate of robot's state,
u_{i+1} is the control input for i^{th} state
& can be odometry info

* We have the measurement model which gives the ~~predicted location~~ observation measurement from estimated robot state and estimated landmark location.

$$\boxed{\hat{z}_{ik} = h(\hat{\mu}_i, \hat{l}_k)} \rightarrow ②$$

- z_{ik} is the actual observation.

[h is the measurement model,
$\hat{\mu}_i \rightarrow$ estimate of robot's state
$\hat{l}_k \rightarrow$ estimate of landmark's location
$\hat{z}_{ik} \rightarrow$ predicted observation of $\hat{l}_k$ from $\hat{\mu}_i$

* SAM optimizes the following objective function,

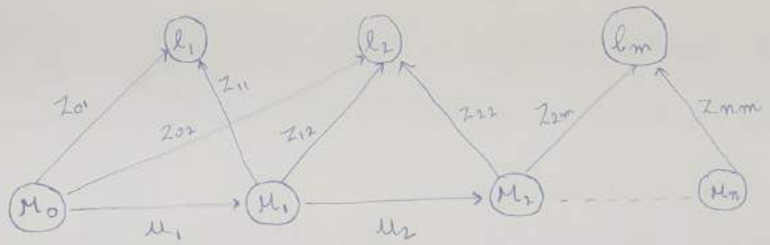
$$\sum_{i=0}^{n-1} \|\hat{\mu}_{i+1} - \mu_{i+1}\|_2^2 + \sum_{i=1}^{n} \sum_{k=1}^{m} \|\hat{z}_{ik} - z_{ik}\|_2^2$$

from ① & ②,

$$\boxed{\underbrace{\sum_{i=0}^{n-1} \|f(\hat{\mu}_i, u_{i+1}) - \mu_{i+1}\|_2^2}_{\text{odometry term}} + \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{m} \|h(\hat{\mu}_i, \hat{l}_k) - z_{ik}\|_2^2}_{\text{measurement term}}} \rightarrow ③$$

Q4

(a)



$\mu_i \to$ robot's ~~pose~~ state in $SE(2)$, i.e. $\mu_i = [\mu_{x_i}, \mu_{y_i}, \mu_{\theta_i}]^T$

$\{l_1, l_2, \ldots l_m\} \longrightarrow$ Landmarks

$\{u_1, u_2 \ldots u_{n-1}\} \longrightarrow$ Control inputs

$z_{ik} \to$ observation of landmark l_k from μ_i robot state.

→ The goal of Smoothing and Mapping (SAM) is to estimate the robot state $\hat{\mu}_i$ and landmark states $\hat{l}_j$ given the observations $\{z_{ik}\}$ and control inputs $\{u_i\}$

* We can obtain the estimate of robot's state using a motion model by utilizing the control inputs,

$$\boxed{\hat{\mu}_{i+1} = f(\hat{\mu}_i, u_{i+1})} \to (1)$$

- μ_{i+1} is the observation from odometery

f is the motion model, $\hat{\mu}_{i+1}$ is the estimate of robot's state,
u_{i+1} is the control input for i^{th} state
& can be odometery info

* We have the measurement model which gives the ~~predicted location~~ observation measurement from estimated robot state and estimated landmark location.

$$\boxed{\hat{z}_{ik} = h(\hat{\mu}_i, \hat{l}_k)} \to (2)$$

- z_{ik} is the actual observation.

h is the measurement model,
$\hat{\mu}_i \to$ estimate of robot's state
$\hat{l}_k \to$ estimate of landmark's location
$\hat{z}_{ik} \to$ predicted observation of $\hat{l}_k$ from $\hat{\mu}_i$

* SAM optimizes the following objective function,

$$\sum_{i=0}^{n-1} \| \hat{\mu}_{i+1} - \mu_{i+1} \|_2^2 + \sum_{i=1}^{m} \sum_{k=1}^{m} \| \hat{z}_{ik} - z_{ik} \|_2^2$$

from (1) & (2),

$$\boxed{\underbrace{\sum_{i=0}^{n-1} \| f(\hat{\mu}_i, u_{i+1}) - \mu_{i+1} \|_2^2}_{\text{odometery term}} + \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{m} \| h(\hat{\mu}_i, \hat{l}_k) - z_{ik} \|_2^2}_{\text{measurement term}}} \to (3)$$

→ Linearizing the odometry term, $f(\hat{x}_i^{\circ}, u_{i+1}) \approx f(x_i^{\circ}, u_{i+1}) + F_i \delta \hat{x}_i$.

Here, F_i is 3×3 Jacobian for motion.

$$\therefore \sum_{i=0}^{n-1} \| f(\hat{x}_i, u_{i+1}) - x_{i+1} \|_2^2 = \sum_{i=0}^{n-1} \| f(x_i^{\circ}, u_{i+1}) + F_i \delta \hat{x}_i - x_{i+1} \|_2^2 \rightarrow (4)$$

→ Linearizing the measurement term as,

$$\hat{z}_{ik} = h(\hat{x}_i^{\circ}, \hat{l}_k^{\circ}) + \frac{\partial h}{\partial x_i} \delta \hat{x}_i \Big|_{x_i^{\circ}} + \frac{\partial h}{\partial l_k} \delta \hat{l}_k \Big|_{l_k^{\circ}}$$

$$= h(\hat{x}_i^{\circ}, \hat{l}_k^{\circ}) + H_{ik} \delta \hat{x}_i + J_{ik} \delta \hat{l}_k$$

$\delta \hat{x}_i$ → change in Robot state
$\delta \hat{l}_k$ → change in landmark state

Now, $\hat{z}_{ik} - z_{ik} = H_{ik} \delta \hat{x}_i + J_{ik} \delta \hat{l}_k - (z_{ik} - h(\hat{x}_i^{\circ}, \hat{l}_k^{\circ}))$

$$= H_{ik} \delta \hat{x}_i + J_{ik} \delta \hat{l}_k - c_{ik}$$

c_{ik} → measurement residual.

$$\therefore \sum_{i=1}^{n} \sum_{k=1}^{m} \| \hat{z}_{ik} - z_{ik} \|_2^2 = \sum_{i=1}^{n} \sum_{k=1}^{m} \| H_{ik} \delta \hat{x}_i + J_{ik} \delta \hat{l}_k - c_{ik} \|_2^2 \longrightarrow (5)$$

H_{ik} → Jacobian of h w.r.t robot pose.
J_{ik} → Jacobian of h w.r.t landmark state

from (4) & (5)

∴ Final SAM objective,

$$\boxed{\sum_{i=0}^{n-1} \| f(x_i^{\circ}, u_{i+1}) + F_i \delta \hat{x}_i - x_{i+1} \|_2^2 + \sum_{i=1}^{n} \sum_{k=1}^{m} \| H_{ik} \delta \hat{x}_i + J_{ik} \delta \hat{l}_k - c_{ik} \|_2^2}$$

Q4

(b) Given a series of images across a trajectory, the goal to calculate the relative pose between images and map the observed environment.

Assume, $\{x_i\}_{i=1,2,\ldots n}$ are ordered sequence of images taken along a trajectory. Additionally, assume that images are taken from a single camera moving in the scene with known calibration matrix K.

I Steps to Estimate Relative Pose b/w Images

* First we use two consecutive images $\{x_k, x_{k-1}\}$ to compute the fundamental matrix of the camera using the following equation,

$$\boxed{x_k^T F x_{k-1} = 0}$$

, F in 3x3 matrix and $x_k = \begin{bmatrix} u_k \\ v_k \\ 1 \end{bmatrix} \rightarrow$ pixel coordinates

- To solve for F, we need 8-point co-correspondences between the two images. (8-point algorithm)

We can use SIFT features to find point correspondence b/w the images.

* Now, from the computed fundamental matrix F, we can compute the essential Matrix E using,

$$\boxed{E = K^T F K}$$

E is 3x3 matrix, K is 3x3 matrix

* With the Essential Matrix, we can compute the relative poses, R (rotation matrix) and t (translation vector) between the cameras in the following way,

$$\boxed{E = [t]_\times R}$$

R is 3x3 matrix, t is 3x1 vector [$[t]_\times$ is skew-symmetric matrix]

- Solving the above equation will result in 4 configurations of t and R. We only choose the configuration for which the world points are in-front of cameras.

II Steps to estimate Structure (map the environment)

* from the above steps, we can compute the Camera Projection matrix

$P_i = K R_i [I \mid t_i]$ for the i^{th} time step, where P_i is 3x3 matrix.

* With P_i's known, we can compute reprojection error for the estimated 3D world point $\hat{X}_j$ in the following way,

N → world points
M → camera views

$$\boxed{\min_{\hat{X}_j} \sum_{j=1}^{N} \sum_{i=1}^{M} \| P_i \hat{X}_j - x_{ij} \|^2} \rightarrow (1)$$

$_{3\times1}x_{ij} \rightarrow$ observed projection j^{th} 3D point X_j at i^{th} time-step (i^{th} camera)

$_{4\times1}\hat{X}_j \rightarrow$ estimated 3D point location of X_j

$\boxed{\hat{x}_{ij} = P_i \hat{X}_j} \rightarrow$ predicted projection of reconstructed j^{th} 3D point $\hat{X}_j$ in i^{th} camera view

↳ (2)

* for each world point we have $\underline{2}$ equations - one for x-coordinate in the image and the other one for y-coordinate. And each world point has $\underline{3}$ unknowns, X, Y and Z.

$\therefore$ Jacobian corresponding to objective function (1) for N world points is of shape $\underline{\underline{2N \times 3}}$.

→ The objective function in (1) can be minimized through LM algorithm to solve for 3D point estimates.

* The initial guess of 3D points can be made using triangulation method. Specifically, $x_{ij} = P_i \hat{X}_j^0$, we can use the fact that $\boxed{x_{ij} \times P_i \hat{X}_j^0 = 0}$
and $x_{i'j} = P_{i'} \hat{X}_j^0 \Rightarrow \boxed{x_{i'j} \times P_{i'} \hat{X}_j^0}$ to solve for initial estimate of $\hat{X}_j^0$.

* Odometery information can be used to obtain initial guess of Camera poses.

* Bundle Adjustment is used to optimise for the trajectory. In Bundle adjustment, both the 3D-points and Camera relative poses are optimized using an objective function of the form,

$$\boxed{\min_{\hat{P}_i, \hat{X}_j} \sum_{j=1}^{N} \sum_{i=1}^{M} d(\hat{P}_i \hat{X}_j, x_{ij})^2}$$

$d \rightarrow$ distance metric.