Mobile Robotics – Final Exam

- Question 1.

1. **Questions on Fundamental Matrix [5 points]**
   a. Derive Fe = 0, where e is the epipole of the 1st camera seen in the second image **[2 points]**
   b. If the fundamental matrix between images I1 and I2 is F, what is the fundamental matrix between images I2 and I1? Why are they different? **[2 points]**
   c. What is the difference between a fundamental matrix and an essential matrix? How are they related? **[1 point]**

a) By definition of the fundamental matrix,

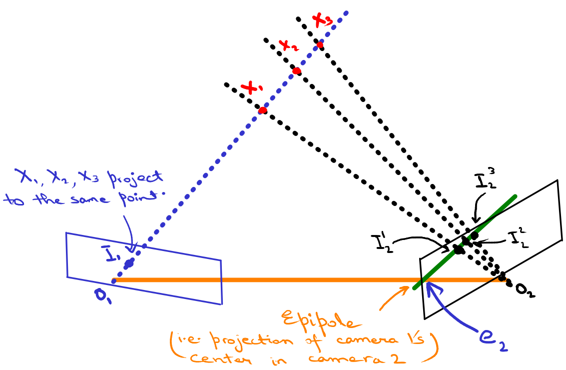
$$I_2^T F I_1 = 0$$

If I_2 and I_1 are images of the same point in the two frames. Such a matrix F is possible since the position of the corresponding point along a line, encapsulated by the nullspace of F, shown below

For any Image point in 1, its corresponding point in 2 lies on its epipolar line, which also has the epipole. Proof shown below.

$$\Rightarrow e_2^T F i_1 = 0 \qquad \forall i_1 \leftarrow \text{point in image}_1$$

$$\& \; i_1^T F e_2 = 0 \qquad \forall i_1$$



I_1, X_1, X_2, X_3, O_1 are colinear. So, all the points put together, i.e. $I_1, I_2^1, I_2^2, I_2^3, X_1, X_2, X_3, O_1, O_2$ are coplanar, this plane is called the Epipolar Plane.

Image plane of the second camera intersects the Epipolar plane, the intersection of 2 planes is a line, called the Epipolar line.

∴ Any point on the Epipolar plane will lie on this epipolar line.

Given that

$$i_1^T F e = 0 \quad \forall i_1$$

We can conclude that

$$Fe = 0$$

b) The definition of the Fundamental Matrix is that:

$$i_1^T F_{12} i_2 = 0 \quad \text{—— } 1$$

for corresponding points i_1 and i_2

Using the same definition for to opposite transform (from 2 to 1), we get:

$$i_2^T F_{21} i_1 = 0 \quad \text{—— } 2$$

Taking transpose of 1

$$i_2^T F_{12}^T i_1 = 0 \quad \text{—— } 1^T$$

Comparing 1^T and 2, we get the equality between the two matrices.

$$F_{21} = F_{12}^T$$

They have to be different because F_{12} takes a full line of image pixels in image 2 to its null space, while F_{21} takes those in image 1.

These projective relations are not symmetric hence the difference.

c) Fundametal Matrix captures Epipolar geometry in pixel space, for an uncalibrated camera

Essential Matrix captures the same Epipolar geometry in the normalized Image space.
This can only be used with a calibrated camera.

Information about camera intrinsics is encapsulated in the Fundamental matrix F, but not in Essential Matrix

Fundamental matrix has 7 degrees of freedom whereas Essential matrix has 5 degrees.

They are related by the following equation (this is by definition and needs no proof)

$$E = K' \, F \, K$$

Intrinsic Matrix of Camera 2

Essential Matrix

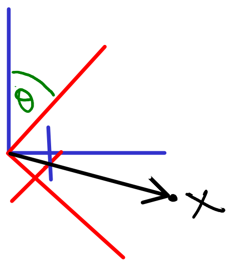
Fundamental matrix

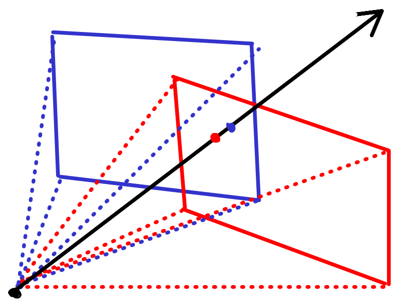
Intrinsic matrics of Camera 1

• Question 2

2. **Questions on relationships between image planes [13 points]**
   a. Derive the homography relation for pure rotation that relates a pixel location x_{i1} in frame I1 to the pixel x_{j2} in frame I2. Does such a homography relation hold when there is a camera translation involved? Explain mathematically your answer **[2 + 2 = 4 points]**
   b. What are multiple ways you can relate a pixel x_{i1} in frame I1 to the pixel x_{j2} in frame I2? Be sure to state your assumptions and any additional information used. Write down the equations for those relations **[4 points]**
   c. What are the two homographies involved in Stereo Rectification? What does each such homography accomplish? Why would you call them as homographies? Explain with figures and equations **[1.5 + 1 + 0.5 + 2 = 5 points]**

a)





Given the image point-world point relations of both cameras, we can write:

$$x_B = K_B [I_{3\times3} \mid 0_{3\times1}] X$$

$$x_R = K_R [R_{3\times3} \mid 0_{3\times1}] X$$

Rearranging and factoring out R, we get these equivalencies

$$K_B^{-1} x_B = [I_{3\times3} \mid 0_{3\times1}] X$$

$$K_R^{-1} x_R = [I_{3\times3} \mid 0_{3\times1}] R_{3\times3} X$$

Equating them both

$$\Rightarrow \quad R_{3\times3} K_B^{-1} x_B = K_R^{-1} x_R$$

$$\Rightarrow \quad \boxed{x_R = (K_R \, R_{3\times3} \, K_B^{-1}) \, x_B}$$

This is the target homography for the camera rotation

No, the relation does not hold when a camera translation is involved.
This is because our equations become:

$$x_B = K_B \left[I_{3\times3} \mid 0_{3\times1} \right] X$$
$$x_R = K_R \left[R_{3\times3} \mid t_{3\times1} \right] X$$

Now to equate these, we need to invert the projective 3×4 matrix, to recover the 3D point and then equate; only a pseudo-left-inverse will be possible.

So, due to lack of invertibility of $[R_{3\times3} \mid t_{3\times1}]$ we cannot easily form a homography, and it cannot be written as a simple equation if one exists since we are going to 3-D 4-length vectors.

b) There are 2 classes of relations which relate points in 2 different Images

i) Homography: One to One relation of points in one image to those in another.

$$\underset{\text{point in image 1}}{x'} = \underset{\text{Homography}}{M}\ \underset{\text{point in image 2}}{x}$$

This is possible if:
i) The two cameras only have a rotation & no translation
OR ii) Both images are viewing the same plane from different angles.

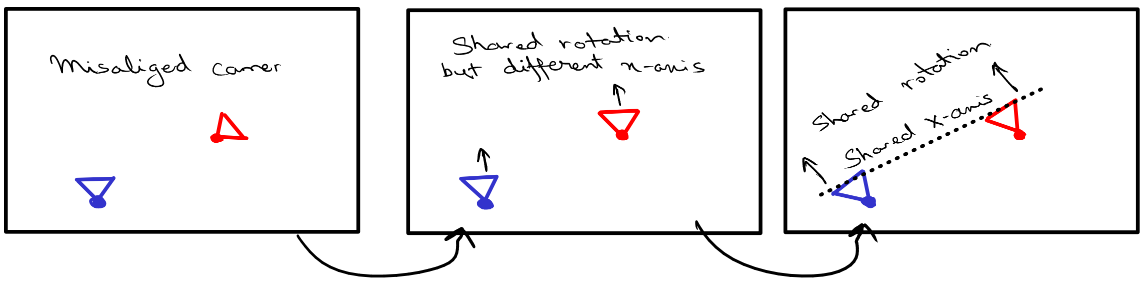
ii) Epipolar geometry: When exact relations don't exist, an image pixel in one image can be mapped to somewhere along a line in the other.

- $x_1^T F x_2 = 0$ as Fundamental matrix
- $x_1^T E x_2 = 0$ as Essential matrix

point in Image 1   point in Image 2

This hold only if there is non-zero translation between the 2 cameras.

c)



$H_1 = K R K^{-1}$ $H_2 = K R_{rect} K^{-1}$

The two homographies are

① Rotational homography $H = K R K^{-1}$

We need both cameras to be in the same rotational frame, but fixed at their initial translation points.

→ Derivation:

$$x = K[I \mid 0]X$$

$$x' = K[I \mid 0]RX$$

$$\Rightarrow x' = K R K^{-1} x$$

② Rotational Homography to set Epipoles to infinity.

We need to compute the rotation matrix to set Epipole E_2 to $[1\ 0\ 0]^T$.

The rotation is generated by construction
$x' = K R_{rect} K^{-1} x$

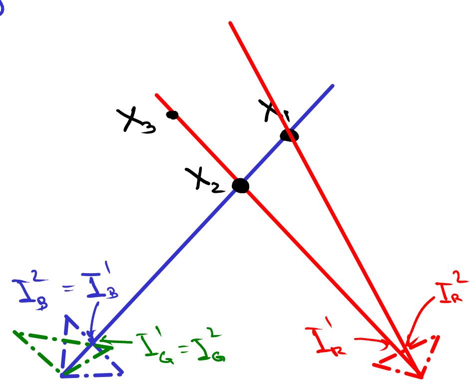
where $R_{rect} = [r_1 \; r_2 \; r_3]^T$, $\vec{r}_1 = \hat{K e_2} = \vec{O}_2 - \vec{O}_1$

$\vec{r}_2 = \vec{r}_1 \times [0 \; 0 \; 1]^T$ and $\vec{r}_3 = \vec{r}_1 \times \vec{r}_2$

and this takes the epipole e^2 to
$K^{-1} R_{rect} K e_2 = [1 \; 0 \; 0]^T$.

c) A Homography is a isomorphism of 2 projective spaces which corresponds to an underlying isomorphism of the vector space that underlies it.

2 points in the real world map to the same point if and only if they are collinear with the projection center.



Any point I_B can be mapped to its equivalent I_G in the rotated camera frame, ∴

∴ this is an isomorphism.

With translation:

$I_B^1 \longrightarrow I_R^1, I_R^2$ This is not 1-1 so

$I_R^1 \longrightarrow I_B^1, I_R^2$ not an isomorphism.

When space undergoes an isomorphism, i.e. structure & point preserving transform, we call it a homomorphism.

- Question 3

3. Questions on Camera Calibration [4 points]

a. Write down, in detail, the proof of the DLT algorithm elucidating upon each step. Be sure to highlight the steps in the algorithm that would fail in their purpose if all the correspondences taken lie on a plane. Why is the eigenvector corresponding to the least eigenvalue taken? **[2 + 1 + 1 = 4 points]**

Direct Linear Transform or DLT helps us estimate the calibration parameters of an uncalibrated camera.

To do so we need to know the 3-D real world coordinates of a few points, and have them identifyable in the image so that we can get the corresponding image coordinates

We know that a projective transform does this mapping

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} P_{11} X + P_{12} Y + P_{13} Z + P_{14} \\ P_{21} X + P_{22} Y + P_{23} Z + P_{24} \\ P_{31} X + P_{32} Y + P_{33} Z + P_{34} \end{bmatrix}$$

This is a homogenous equation, we can get 2-equations equating the left and the right:

$$x = \frac{P_{11} X + P_{12} Y + P_{13} Z + P_{14}}{P_{31} X + P_{32} Y + P_{33} Z + P_{34}}$$

$$y = \frac{P_{21} X + P_{22} Y + P_{23} Z + P_{24}}{P_{31} X + P_{32} Y + P_{33} Z + P_{34}}$$

we can rearrange the terms to write this as a system of linear equations.

$$\underbrace{\begin{bmatrix} -X_i & -Y_i & -Z_i & -1 & 0 & 0 & 0 & 0 & x_iX_i & x_iY_i & x_iZ_i & x_i \\ 0 & 0 & 0 & 0 & -X_i & -Y_i & -Z_i & -1 & y_iX_i & y_iY_i & y_iZ_i & y_i \\ & & & & & \text{for all points } i & & & & & & \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ \vdots \\ \text{all 12 terms} \end{bmatrix}}_{\vec{p}} = 0 \quad \text{—①}$$

Since there are 11 parameters in the P vector (12 terms − 1 homogenity scalar), we need atleast 6 points for this to be fully determined. (2 eqn's per point)

However, we have noise in the real world due to which we take an overdetermined set of equations with more than 6 points and try to minimize the residuals.

Writing equation ① again: $\hat{p} = \min_{\vec{p}} A\vec{p}$

We can do SVD on A to get
$A = UDV^T$,

The last column vector in V is the best solution to $AP = 0$, since its the eigenvector corresponding to the smallest eigenvalue.

This is done because other than the trivial solution $P = 0$, all other values of $\vec{p}$ vector can be normalized.

Let $\vec{v}_1, \vec{v}_2 \ldots \vec{v}_n$ be the eigenvectors of A which are the columns of V^T or rows of U.

So any $\vec{p} = a_1\vec{v}_1 + a_2\vec{v}_2 + \cdots + a_n\vec{v}_n$

$\Rightarrow$ $A\vec{p} = \lambda_1 a_1 \vec{v}_1 + \lambda_2 a_2 \vec{v}_2 + \cdots + \lambda_n a_n \vec{v}_n$

where $\lambda_1 > \lambda_2 > \cdots > \lambda_n$, λs are eigenvalues

$\Rightarrow$ $\|A\vec{p}\|$ is smallest if $a_1 = a_2 = \cdots = a_{n-1} = 0$ and $a_n \neq 0$, since λ_n is the smallest.

$\therefore$ The smallest eigenvector is used as the solution to $\vec{p}$, to minimize the norm residuals $\|A\vec{p}\|$.

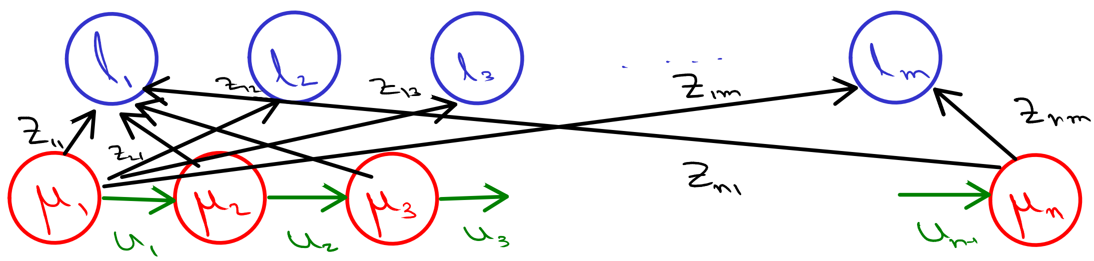
This algorithm will fail to produce a unique non-0 result because the matrix A would not be full rank. If points are in a plane, we can choose a coordinate frame where $\forall Z_i = 0$. $\therefore$ A has atmost 9 independent non-0 column vectors, so removing homogeneity P lies somewhere in a 2-D subspace.

Only 1 point in the subspace is valid for the full image, but all points in the subspace correctly map that one plane to the image.

- Question 4

4. **Questions on SLAM/SfM:** For the following questions, clearly write down the variables being solved for and the shape/size of all vectors/jacobians. **[14 points]**
   a. Given 2D points observed in *m* observations and the relative pose between any two observations (in the 2D plane - SE(2)), write down the optimization formulation for SAM (Smoothing and Mapping). **[4 points]**
   b. You are given a series of RGB images across a trajectory (no additional information) and have to estimate the relative pose between images and map the observed environment. In a systematic and concise way, write down the steps you would take to perform such a monocular SLAM with their mathematical equations. Describe how you obtain your initial estimates and optimise for the trajectory. **[10 points]**

a)



m landmarks (observations per frame)
n timesteps of motion and observation

$$\mu_i = [x_i \quad y_i \quad \theta_i]$$

We need to be given:

a) A motion model. $f(\vec{\mu}_i, u_i) \rightarrow \mu_{i+1}$

eg. $\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ \theta_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \begin{bmatrix} T\cos(\theta_i + \Delta\theta_i) \\ T\sin(\theta_i + \Delta\theta_i) \\ \Delta\theta_i \end{bmatrix}$

b) An observation model $Z_{ik} = h(\vec{\mu}_i, \vec{l}_k)$

eg. $Z_{ij} = \begin{bmatrix} \| \vec{\mu}_i - \vec{l}_j \| \\ \tan^{-1}\left({x_i - l_{jx}}/{y_i - l_{jy}} \right) \end{bmatrix}$

Now we can formulate our SAM objective and a linear approximation thereof

$$\mathcal{L} = \sum_{i=0}^{n} \left\| f(\vec{\mu}_i, u_i) - \mu_{i+1} \right\|_2^2 + \sum_{i=0}^{n} \sum_{k=0}^{m} \left\| \hat{z}_{ik} - h(l_k, \vec{\mu}_i) \right\|_2^2$$

Movement Error   Landmark error

After linearizing $h(l_k, \vec{\mu}_i) \longrightarrow H_{ik}\,\delta\hat{\mu}_i + J_k\,\delta\mu_{mk} - C_{ik}$.

Given this loss term we can compute our residuals $\therefore$ our Jacobian.

Jacobian

$$\begin{bmatrix} F_1 & 0 & \cdots & J_{11} & 0 & \cdots & 0 \\ H_{11} & 0 & \cdots & 0 & J_{12} & \cdots & 0 \\ H_{12} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & & & & \vdots \\ H_{1m} & 0 & \cdots & & & \cdots & J_{1m} \\ 0 & F_2 & & & & & \ddots \\ 0 & H_{21} & \cdots & 0 & J_{21} & & \vdots \\ 0 & H_{22} & \cdots & 0 & 0 & J_{22} \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & H_{NM} & 0 & \cdots & J_{NM} \end{bmatrix} \begin{bmatrix} \delta\hat{\mu}_1 \\ \delta\hat{\mu}_2 \\ \vdots \\ \delta\hat{\mu}_n \\ \delta\hat{\mu}_{m1} \\ \delta\mu_{m2} \\ \vdots \\ \delta\hat{\mu}_{nm} \end{bmatrix} = \begin{bmatrix} a_1 \\ C_{11} \\ C_{12} \\ \vdots \\ C_{1m} \\ a_2 \\ C_{21} \\ \vdots \\ C_{2m} \\ \vdots \\ C_{nm} \end{bmatrix}$$

Shape of Parameter vector $\longrightarrow (3N + 2M,\ 1)$

$\because$ N locations of robot with x, y, θ

M landmark locations with x, y

Shape of Residuals vector $\longrightarrow (2N + 2NM + 1,\ 1)$

$\because$ $(N-1)$ Motion constraints u_i with T and $\Delta\theta$

NM observations of points with error in T & $\Delta\theta$

3 anchor constraints $x_0 = y_0 = \theta_0 = 0$

$\therefore$ Jacobian has shape $(3N + 2M,\ 2N + 2NM + 1)$

b) The steps in the pipeline of computing 3D model from an image trajectory looks as follows.

Input Images → SIFT for Point descriptors

↓

Matching Algorithm → RANSAC to reduce bad correspondence errors.

↓

8-point algorithm to compute F.

↓

Structure from motion to get 4 possible poses.

↓

Triangulation to filter to right pose.

↓

PnP to get poses (for other cameras) → Bundle Adjustments

Now the entire pipeline in more detail.

i) SIFT to generate image descriptors.
Random filters are convolved with the image to generate point descriptors a vector of convolution outputs

$$I * C_i \longrightarrow D_i^{N\cdot M}$$

ii) An all pair point descriptor matching algorithm
One example (Rather inefficient) is the Brute Force matcher. We get $f(I_1) = I_2$ where $I_1 \sim I_2 \sim x$ i.e. images of the same point.

iii) Use the 8 point algorithm to compute Fundamental matrix.

We know that for every correspondence $x_1 \sim x_2$

$x_1^T F x_2 = 0$

we can take F as unknown vector and phrase linear equation

$$A \vec{F} = \vec{0}$$

9 parameter homogeneous matrix

$$A = \otimes x_1, x_2 \quad \text{(Kronecker product)}$$

The smallest eigenvector is an approximation to $\vec{F}$, so let $UDV^T = A$, we let

$$F_{approx} = V^T[-1].\text{reshape}(3,3)$$

$\because$ F has to be rank deficient,

we compute SVD again $UDV^T = F$.

We drop the lowest eigenvalue to 0 and reconstruct a rank deficient F.

$$\hat{F} = U \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

iv) We do RANSAC over correspondences to improve estimate of F.

v) Given our matrix F (or E if we know camera calibration, we use Structure from motion (SfM)'s. We get the following estimates of camera pose from E.

Let $E = UDV^T$, $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

So possible configurations are

$$P_1 = [UWV^T \mid U[:,3]]$$
$$P_2 = [UWV^T \mid -U[:,3]]$$
$$P_3 = [UW^TV^T \mid U[:,3]]$$
$$P_4 = [UW^TV^T \mid -U[:,3]]$$

So now we have an ambiguous camera pose.

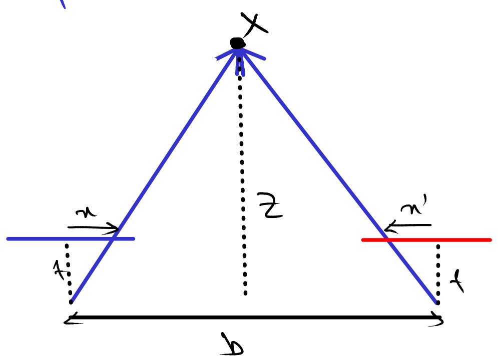
vi) Stereo Rectification is now performed using the two homographies.

$H = KRK^{-1}$ on right cameras

$H = KR_{rect}K^{-1}$ on both camera.

Now they share scan lines allowing easy triangulation

vi) Disparity map is generated to get 3D position (i.e. depth information) of points.



$$d = x - x' = \frac{bf}{Z}$$

$$\frac{X}{Z} = \frac{x}{f} \qquad \frac{b-X}{Z} = \frac{x'}{f}$$

Using chirality conditions, that camera pose of the 4 poses is selected which has camera seeing the points.

Now we have a point positions as well given the depth information

vii) Perspective from n-points helps us improve our camera pose estimates given the actual 3-D information about those points.

We know correspondences since we generated the 3-D points from 2D.

$\hat{x}_{3\times1} = P_{3\times4} \, X_{4\times1}$ Residual $= x - \hat{x}$

Loss $= \|x - PX\|_2^2$

We optimize the 11 free parameters of P using Levenberg-Marquardt (LM) scheme.

viii) Bundle adjustments: We have the point estimates and camera pose estimates, we need to refine them.

We are simultaneously refining P_i poses & X_j world map by minimizing Reprojection error

$$\|\lambda_{ij} P_i \bar{X}_j - x_{ij}\|^2$$

by computing Jacobian & using LM algorithm.