CSE-483: Mobile Robotics **Monsoon 2019**

Epipolar geometry: 8 point algorithm and RANSAC

Prepared by: Chaitanya Kharyal and Ajay Shrihari

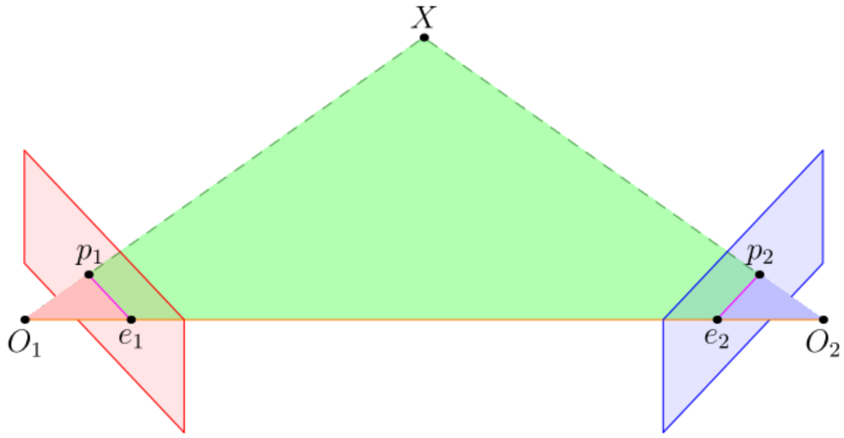

fig1: The epipolar plane. O_1 O_2 are the two camera centers, X is the 3D point of which p_1 and p_2 are images in respective image planes. e_1 and e_2 are the respective epipoles. e_1p_1 and e_2p_2 are epipolar lines

1 The Fundamental matrix (Recap)

The Fundamental matrix (F) has already been covered in previous notes but we should do a brief recap of F matrix and its properties.

From our prior knowledge,

$$\boldsymbol{F_{1,2} = K^{-T}[t_X]RK^{-1}}$$

where,

$$[t_X] = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$$

is skew symmetric cross product matrix

$F_{1,2}$ is a rank 2 (rank deficient) matrix and relates the points in image 1 to the corresponding points in image 2 as,

$$\boldsymbol{p_1^T F_{1,2} p_2 = 0} \text{ ——(coplanarity constraint)}$$

If we rip F of camera intrinsics, K, we get Essential matrix E.

ie.

$$\begin{aligned} \boldsymbol{E} &= \boldsymbol{K^T F K} \\ \implies \boldsymbol{x_1^T E X_2} &= \boldsymbol{0} \end{aligned}$$

Where, x_1 and x_2 are normalised image co-ordinates

2 Estimating F (The 8 point algorithm)

For estimating F, we use the coplanarity constraint.

We know,

$$\begin{aligned} \boldsymbol{p_1^T F_{1,2} p_2} &= \boldsymbol{0} \\ \implies \boldsymbol{(p_1 \otimes p_2)^T f} &= \boldsymbol{0} \end{aligned}$$

Where, $p_1 \otimes p_2$ is kronecker product of p_1 and p_2 and f is flattened F matrix.

ie.

$$\begin{aligned} f = [F[1,1], F[1,2], F[1,3], F[2,1], F[2,2], F[2,3], F[3,1], F[3,2], F[3,3]]^T \\ = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9]^T \end{aligned}$$

Statement: The fundamental F matrix has seven degrees of freedom. This is because F is homogeneous and singular, as the skew symmetric matrix is singular with rank two.

Explanation:

Let us start with 9 parameters of F since it is a (3×3) matrix. Now,
$\boldsymbol{(p_1 \otimes p_2)^T f = 0}$
where, $p_1 = [u, v, 1]^T$ and $p_2 = [u', v', 1]^T$

$\implies p_1 \otimes p_2 = [uu', uv', u, vu', vv', v, u', v', 1]^T$
$\implies \boldsymbol{(p_1 \otimes p_2)^T f = uu' f_1 + uv' f_2 + u f_3 + vu' f_4 + vv' f_5 + v f_6 + u' f_7 + v' f_8 + f_9 = 0}$

or,

$$\boldsymbol{f_9 = -(uu'f_1 + uv'f_2 + uf_3 + vu'f_4 + vv'f_5 + vf_6 + u'f_7 + v'f_8)}$$

ie. f_9 or $F[3,3]$ can be represented by all of the other parameters. This is due to **Homogenity**.

With 8 independent parameters remaining in F, we proceed,

We know,
$|\mathbf{F}| = \mathbf{0}$
ie.
$(f_1 f_5 f_8) + (f_2 f_6 f_7) + (f_3 f_4 f_8) - (f_3 f_5 f_7) - (f_2 f_4 f_9) - (f_1 f_6 f_8) = 0$

We already know that f_9 can be represented as all the other 8 parameters. Therefore, using the above equation, we can represent another parameter using the remaining 7 parameters. This is because F is **Singular**

Therefore, $DoF(F) = 7$

For a given set of corresponding points, p_i and p'_i in camera C and C' respectively, The equation $p_i^T F p'_i = 0$, or $(p_i \otimes p'_t)^T f = 0$, will yield only one equation, unlike Homography. Therefore, we need 7 points to estimate F using coplanarity constraints.

Since, $|F| = 0$ is a cubic equation, it would have at most 3 solutions, the correct one of which can't be found using only 7 correspondences. Therefore, we need 8 points to estimate F and hence we call it 8 point algorithm.

We, get the correspondences (SIFT, SURF etc.) between the images and use those as points in estimating F.

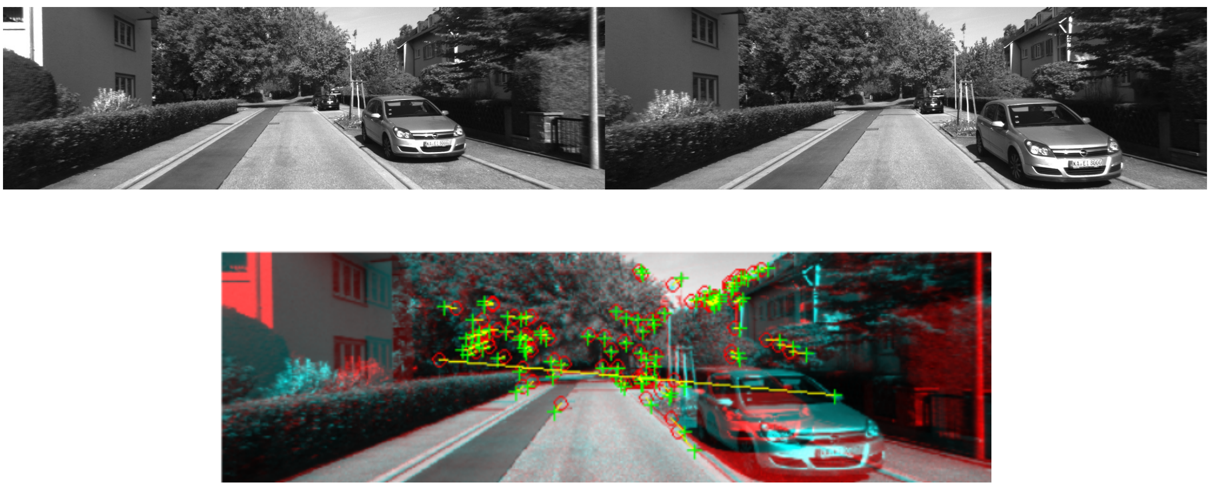

fig2: Images and their matched ORB features

We select 8 points from these matched features, find their Kronecker product and stack them up in a matrix, say $A_{(8\times 9)}$, such that,

$$\boldsymbol{A_{(8\times 9)} f_{(9\times 1)} = 0}$$

Statement: Now, we do **Singular Value Decomposition** of A and take the last column vector of V as f,

$$\boldsymbol{A = UDV^T}$$
$$\boldsymbol{f = V[:, -1]}$$

Since, recovered vector is a scaled version of original F because we can't recover the scale factor, we usually take f as a unit vector. *ie.* $||f|| = 1$

Explanation 1:

Since, our estimates of points is not truly precise, The readings will be noisy. Therefore, Af need not be 0. *ie.* $Af \sim 0$

Since $Af = 0$, ideal f lies in *Null Space* of A, we pick the vector which is closest to the Null space of A. This would be the vector corresponding to least singular value *ie.* last column vector of V.

Explanation 2:

We need to minimize $||Af||$ *st.* $||f|| = 1$,

$$\boldsymbol{A = UDV^T}$$
$$\implies \boldsymbol{minimize(||UDV^T f||)}$$

Since U is orthogonal, $||Ux|| = ||x||$:

$$\implies \boldsymbol{minimize(||DV^T f||)}$$

Let, $\boldsymbol{y = V^T f}$, $||y|| = ||f|| = 1$
Therefore, for $||Dy||$ to be minimized,

$$y = [0, 0, 0, 0, ..., 0, 1]^T$$
$$\boldsymbol{f = Vy} \implies \text{last column of } V$$

Therefore, f is the last column of V.

We know that F is a rank 2 matrix but the recovered matrix need not be rank 2. Therefore, we force the rank 2 condition by making the last singular value of F as 0.

$$\boldsymbol{F_{recovered} = UDV^T}$$
$$\boldsymbol{F_{recoveredNew} = U(diag(D[1,1], D[2,2], 0))V^T}$$

Practical problem: SVD doesn't work well with large variance in matrix values, Therefore, normalise Image co-ordinates beforehand and then denormalize F after recovering it.

3 RANSAC as an algorithm

Random Sample Consensus is an iterative algorithm to estimate parameters of a mathematical model. In this algorithm, the outliers have no effect on the final estimated parameters. It is a non deterministic algorithm which has some probability of success which increases with number of iterations.

The algo: We iteratively sample random samples from our sample and try to fit our model to that data. We find the number of inliers (with distance $< \epsilon$) with our fitted model. We pick the model which has maximum number of inliers.

Statement: The RANSAC algorithm will converge in

$$\boldsymbol{N = \frac{log(1-p)}{log(1-(1-e)^s)}}$$

Steps. Where, p is the probability that at least one sample is free from outliers, s is the number of data points in a sample that we have picked and e is the probability of a data point being an outlier.

Explanation:

We say that the following statement is true,

$$\boldsymbol{(1-(1-e)^s)^N = 1-p}$$

RHS:

If p is the probability that at least one sample is free from outliers, $1-p$ is the probability that no sample that we have picked is free from outliers or we have at least one outlier in all samples.

LHS:

If e is the probability of a data point being an outlier, $1-e$ is the probability of a data point being an inlier and $(1-e)^s$ is the probability of all data points in a sample to be inliers. Therefore, $1-(1-e)^s$ is the probability that at least one data point in our sample is outlier and $(1-(1-e)^s)^N$ extends this idea to all the samples in N iterations *ie.* At least one data point in all samples is an outlier.

Therefore, **LHS** is same as **RHS**.

And, our statement equation is a direct result of our explanation equation.

4 RANSAC as used in F matrix estimation

Why RANSAC?

We pick 8 correspondences between two images for F matrix estimation as explained earlier. These correspondences are usually some sort of features which are extracted from the images and then matched. There can be some outliers (long line in *fig2*) present in these correspondences. We don't want to pick an outlier as one of the eight points and it to have any influence on the parameters (F) that we want to estimate.

How RANSAC?

Similar to conventional RANSAC, we iteratively calculate our "model" F by picking up random 8 points, we find the number of inliers for every iteration using the coplanatity constraint.

$$\boldsymbol{p_1^T F_{1,2} p_2} < \boldsymbol{\epsilon}$$

Then we take F which maximizes the number of inliers. And then we recalculate F by taking all the inliers into consideration for a better estimate of F.