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

Epipolar Geometry II

Prepared by: Anirudha Ramesh, Aryan Sakaria

1 F-matrix estimation

Motivation

From the previous notes, we saw how the coplanarity constraint led to the derivation of the fundamental matrix. This constraint is captured in the equation:

$$x'^T \mathbf{F} x'' = 0 \tag{1}$$

where x' and x'' are image points in two different images that correspond to the same 3-D point X such that

$$x = P_1 X \tag{2}$$
$$x'' = P_2 X \tag{3}$$

The information regarding the orientation of the cameras is stored in the projection matrices. However, we do not always have access to these matrices. Fortunately for us, the fundamental matrix contains information about the relative orientation of two images. This relative orientation is useful for many applications(ex. odometry). This is the primary motivation behind estimating the F matrix.

Figure 1: Set of corresponding points (taken from Stachniss Slides)

As you can see in figure 1, the points correspond to the same 3-D locations in the world. A set of corresponding points can be manually labeled or identified based on some standard models like SIFT although these can be noisy.

Linear Dependency
Expanding on eq(1), we get

$$\begin{bmatrix} x'_1 & x'_2 & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x''_1 \\ x''_2 \\ 1 \end{bmatrix} = 0$$

This equation has the unknown variables $\mathbf{F}_{ij}$ in the middle and the known variables x' and x" on the sides. We will re-formulate this equation as an equation in the variables of F

Consider F_i to be the i_{th} column vector of F
Re-writing eq(1)

$$\begin{bmatrix} x'^T \end{bmatrix} \begin{bmatrix} F_1 & F_2 & F_3 \end{bmatrix} \begin{bmatrix} x'' \end{bmatrix} = 0$$

$$\begin{bmatrix} x'^T F_1 & x'^T F_2 & x'^T F_3 \end{bmatrix} \begin{bmatrix} x'' \end{bmatrix} = 0$$

$$x'^T F_1 x''_1 + x'^T F_2 x''_2 + x'^T F_3 x''_3 = 0$$

We can do the following as x''_i is a scalar quantity.

$$\begin{bmatrix} x'^T x''_1 & x'^T x''_2 & x'^T x''_3 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = 0$$

The dimensions of the matrices are 1x9 and 9x1 respectively

$$\begin{bmatrix} x'_1x''_1 & x'_2x''_1 & x'_3x''_1 & x'_1x''_2 & x'_2x''_2 & x'_3x''_2 & x'_1x''_3 & x'_2x''_3 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{21} \\ F_{31} \\ F_{12} \\ F_{22} \\ F_{32} \\ F_{13} \\ F_{23} \\ F_{33} \end{bmatrix} = 0$$

This is a linear equation in the variables of F for one correspondence. Note that $x'_3x''_3 = 1$ as both x' and x'' are homogenous normalized pixel co-ordinates. The matrix on the right is the kronecker product of x'' and x', which is written as $x'' \bigotimes x'$. The matrix on the right is characterised as Vec(F)
We define a^n is the kronecker product of the n^{th} set of corresponding points **Problem Formulation** The last element of any a^n correspondence will always be 1. From the co-planarity constraint, we know that

$$a^n f = 0 \tag{4}$$

This implies that

$$f_9 + \sum_{i=1}^{8} a_i^n f_i = 0$$

Therefore, f_9 can be written as a linear combination of the other 8 points.

Since F is a singular matrix, another parameter can be expressed in terms of other parameters. Therefore, f has 7 Degrees of Freedom

We stack up our n-correspondences in a matrix called A where each row is of the form a^{iT} A is a matrix of dimensions Nx9, where each row is a kronecker product of a correspondence. Due to co-planarity constraint and eq(4):

$$Af = 0 \tag{5}$$

The 8-point algorithm
We first explain the motivation behind taking only 8 points. Consider the matrix A. Since the last column of A can be represented as a linear combination of the other 8 columns. The intuition for this is fairly simple. This result is captured by the summation mentioned earlier. As a result, we consider 8 points. Considering more points is not a good idea, as more noise will be incorporated into the solution and the data and the matrix A will become a regular matrix.
We have to find the null space of the matrix A to solve Af = 0. In practise, this does not happen as measurements are not precise and data is noisy. So we settle with a solution that gives us a solution close to 0. This is achieved by taking the right singular vector of A.
Taking the singular valued decomposition of A, we get:

$$A = UDV^T \tag{6}$$

We take the last column vector of V as this is the vector that corresponds to the least singular value. Since V is orthogonal, $\|f\|$ is a unit vector. The information regarding scale transformation is lost

Explanation for the algorithm

We have to minimize $\|Af\|$ s.t. $\|f\| = 1$
Taking SVD of A, we get

$$A = UDV^T Af = UDV^T f \tag{7}$$

Our minimization problem is now min($\|UDV^T f\|$

$$\|UDV^T f\| = \|DV^T f\| \tag{8}$$

as U is orthogonal

Let

$$y = V^T f$$
$$\|y\| = \|V^T f\|$$

But since V is an orthogonal matrix, and $\|f\| = 1$

$$\|y\| = 1$$

To minimize $\|DV^T f\|$ or $\|Dy\|$, we take

$$y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \ldots.. & 1 \end{bmatrix} \tag{9}$$

This implies that the best solution is the last column of V. This column vector is reshaped column-wise to form the Fundamental Matrix

Enforcing Rank 2

The F matrix is supposed to be a matrix of rank 2. However this may not be the case with our recovered matrix. So to ensure rank 2, we take the SVD of our calculated matrix. Out of the three singular values, we only consider the highest two and their corresponding vectors in U and V^T. The D_{new} matrix with D_{33} zeroed out is multiplied as follows:

$$F_{new} = UD_{new}V^T \tag{10}$$

Concluding Notes

Oftentimes, feature detection is not done manually and instead modules such as SIFT and orb are used. As a result, we get many noisy data points that can skew calculations of our F-matrix. To combat this, we use robust schemes like RANSAC, which is discussed in the next section.
SVD of a matrix with large variance in its elements will not give accurate results. In practice, the points are usually de-normalized and the F-matrix is calculated. This matrix is then denormalized to get the required F-matrix

2 RANSAC

Motivation

Before going on to connect RANSAC to the 8 point algorithm, lets understand what RANSAC is and what it does.
RANSAC stands for Random Sample Consensus. It is an iterative approach to find the points that would best represent the data for a given model. It enables a robust estimation of model parameters, i.e. it can estimate the parameters with a high degree of accuracy even when a significant number of outliers are present in the data set.
Let us start with the example of fitting a line through a cluster of points.

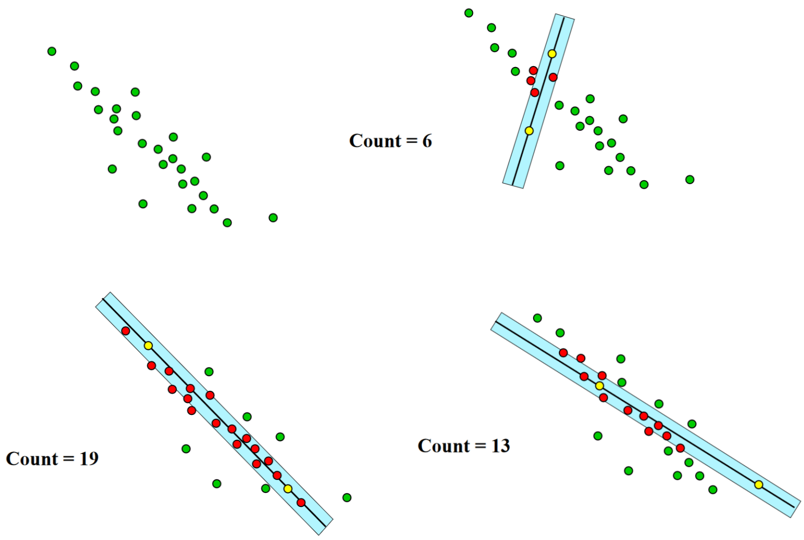

Figure 2: Set of corresponding points (taken from Penn State Slides)

The best fitting data points for the model are the points chosen in the image where count = 19 as when we fit our model (in this case, line) through it, most of our points can be rightly represented.

This is preferred in many cases over minimizing square distance error and other such error functions as outliers can significantly change the best fit line/model in them, whereas RANSAC tries to fit our model best to inliers.

RANSAC is a 2 stage algorithm:

1) Classify data points as inliers or outliers

2) Fit model to inliers

RANSAC for the 8 point algorithm

Now that we have seen the general case, we can take a look at how we can use RANSAC to find a good F matrix.

Steps :

1) Run the process of finding F matrix (8 point algorithm) N number of times. Each time choose 8 corresponding points from the set of all corresponding points randomly.

2) Run the algorithm to find F matrix with chosen 8 corresponding points.

3) Now for every pair of corresponding points in the dataset check if the points follow the coplanarity constraint x'Fx = 0. Setting a hard equal to zero will not give good results, and we should give room for noises induced. So test for x'Fx < d, where d is a small value used as a threshold.

4) Keep a count of how many such data points fit in with the F calculated.

5) Repeat steps 1-4 N times.

6) Choose F matrix corresponding to which maximum points fit the coplanarity constraint, since this is the F that best represents the data.

Selection of N

Q) How many samples to iterate over to surely find sample with only inliers ?

e = probability that a point is an outlier
s = number of points in a sample
N = number of samples (we want to compute this)
p = desired probability that we get a good sample

=> $1 - e$ = Probability of choosing one point, and the point is an inlier
=> $(1 - e)^s$ = Probability of choosing s points, and they are all inliers
=> $1 - (1 - e)^s$ = Probability of choosing s points, and not all of them are inliers
=> $(1-(1-e)^s)^N$ = Probability of choosing N samples of size s, and none of the samples have all inliers
=> $1 - (1 - (1 - e)^s)^N$ = Probability of choosing N samples of size s, and at least one of the samples has all inliers

Therefore,
$1 - (1 - (1 - e)^s)^N = p$

Deriving N from this,
$(1 - (1 - e)^s)^N = 1 - p$
$N * log(1 - (1 - e)^s) = log(1 - p)$

$N = log(1 - p)/log(1 - (1 - e)^s)$

Which is the number of times we should pick samples in RANSAC to ensure that at least one sample has only inliers.

Early Termination:
Instead of running N times to get all inliers, we can also terminate early to get a good estimate usually. We can terminate when the inlier ratio given when testing how many data points fit into our model is equal to or greater than the expected inlier ratio, i.e. 1-e.

Selection of d

Value of threshold d is usually chosen empirically depending on data being worked with, but for certain distributions of data we can follow some formulae to choose required d. Eg:- Gaussian distribution with standard deviation σ and mean 0, $d^2 = 3.84\sigma^2$ gives p = 0.95 that the point is an inlier when the model has 1 DOF like a line.
Limitations If the inlier ratio is very small, the number of iterations can become very large.
It usually performs badly when the number of inliers is less than 50%.
It requires problem-specific setting of threshold.
RANSAC can only estimate one model for a particular data set. As for any one-model approach when two (or more) model instances exist, RANSAC may fail to find either one.

References

[1] Slides by Cyrill Stachniss: Download here

[2] GMU Slides: Download Here

[3] Wikipedia: Link

[4] Multiple View Geometry, Hartley-Zisserman