CS6790
Geometry and Photometry in Computer Vision
Assignment 3

Divya K Raman
EE15B085

15 April 2018

1 Introduction

This assignment deals with computing the fundamental matrix between two images using a point matcher like SIFT and RANSAC using the seven point and eight point algorithms as base. We then evaluate the performance of both these algorithms in terms of both running time and number of iterations needed for convergence.

2 8 point algorithm

The first task is to find corresponding points between the images. For this, we use MATLAB's inbuilt functions for detecting SURF features and matching them. We then normalise all the points before proceeding with the computation of the fundamental matrix.

$x'^T Fx = 0$ is the fundamental matrix relation. F has 8 free parameters due to the scale factor. Further, it is a rank 2 matrix and hence effectively has only 7 free parameters. Using 8 points, we use DLT to compute F. Now, this F obtained in general will not be of rank 2. What we do is then,

compute F = UDV'(SVD)

Say the diagonal elements of D are p,q,r. Ideally, r should be far far lesser than p and q. We then recompute F as

F = U[p,0,0;0,q,0;0,0,0]V'

This constrains F to be a rank 2 matrix. We finally denormalise F.

3 7 point algorithm

In this algorithm, we use only 7 points to compute F. Again, the $x'^T Fx = 0$ is used and a 2 dimensional null space is obtained- say f1, f2. Fundamental matrix is an element of this nullspace and is therefore of the form F = f1 + λ f2. We then enforce the condition det(F) = 0. This gives us a cubic equation in λ and hence has 3 solutions- 1 real solution or 3 real solutions. In the 3 real solutions case, we select the F whoch gives us maximum number of inliers.

4 RANSAC

We now use the 8 point and 7 point algorithms as base to calculate the fundamental matrix by an iterative algorithm. We use SURF to compute features in an image and use the matchFeatures function to find the corresponding points. Number of inliers are computed using a threshold on the algebraic error.
The RANSAC Algorithm
Initialise iter = 0 and Niter = 10000.
1. Select a random sample of s = 8(or 7) correspondences and compute F.
2. Compute the number of inliers consistent with this F.
3. Set ϵ =1(number of inliers)/(total number of points).
4. Compute Niter = log(1-p)/log(1-(1-ϵ)^s) with p = 0.99.
5. Increment iter by 1.
Perform the above steps while iter¡Niter and then terminate.
RANSAC improves performance by making the algorithm less sensitive to outliers.

5 Results

General observation:

The 8 point algorithm is faster than the 7 point algorithm. The number of iterations that both take to converge seems to be similar.

Image 1 and 2:

8 point algorithm - 10 iterations, 9.692 sec

-0.000000003540, -0.00000011723, 0.0002447;
0.00000014239, 0.00000000229455,-0.00077751860953867;
-0.0003361,0.00041274,1.0

7 point algorithm - 13 iterations, 84.318 sec

-0.0000004071755, -0.000000985965, 0.003857580;
0.00000073690, 0.000000466, -0.003593;

-0.0008207644, 0.00064869, 1.0

Image 1 and 3:

8 point algorithm - 9 iterations, 10.616 sec

-0.0000000102032,0.000000509227,-0.00121496582869;
-0.0000002379558, -0.00000000010645131, 0.00029760916368;
0.00058381992857903,-0.000728612882085144,1.0

7 point algorithm - 12 iterations, 88.089 sec

-0.000000016766619,0.0000029053496, -0.00698780;
-0.000001731, 0.000000070947019, 0.001368054;
0.0041960113509, -0.0019622514705, 1.0

Image 2 and 4:

8 point algorithm - 11 iterations, 11.44 sec

-0.0000000312379420,-0.00000157877400397, 0.0041980;
0.000001463248, 0.000000087012580, -0.0039985231;
-0.00381359597, 0.003394834,1.0

7 point algorithm - 7 iterations, 66.289 sec

0.000000018875178,0.0000005059131, -0.0014182790;
-0.000000411130, 0.00000000952953, 0.0013803938965503;
0.000920451854,-0.0015500,1.0

Image 3 and 4:

8 point algorithm - 7 iterations, 9.15 sec

-0.00000026602359,0.000034929267712, -0.0840505138039588892;
-0.000036824174458, 0.0000001928555093400063924491, 0.13468059897422790;
0.0967517048120498657, -0.1463133245706558227,1.0

7 point algorithm - 5 iterations, 39 sec

0.0000000017682526595286,-0.0000015223778, 0.003627731378;
0.00000178073146, 0.0000000251780, -0.0066688;
-0.0044593655,0.0064085792181,1.0

6 Conclusion

Fundamental matrix values obtained using both algorithms are of the same order and the values are also similar in most cases. There is a deviation of the value obtained when the matlab inbuilt function estimateFundamentalMatrix is used. This suggests the need for an improvisation of our implementation. One method would be to re estimate our fundamental matrix using only the inliers. Better thresholds can be placed. Sampson error can be used instead of algebraic error to compute the number of inliers.