CS6790: Geometry and Photometry based Computer Vision

Assignment # 3
Fundamental Matrix Calculation

Sourav Sahoo, EE17B040

April 7, 2020

Contents

List of Figures

List of Tables

1 Introduction

In computer vision, a fundamental matrix is 3×3 that relates corresponding points in a pair of stereo images [1]. In epipolar geometry, with homogeneous corresponding points $\mathbf{x}$ and $\mathbf{x}'$ in a pair of stereo images, this indicates that the corresponding point $\mathbf{x}'$ lies on the epipolar line $\mathtt{F}\mathbf{x}$ for all such pairs of points. So, mathematically, the fundamental matrix can be defined as given in Equation 1.

$$\mathbf{x}'^{\top}\mathtt{F}\mathbf{x} = 0 \tag{1}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are the corresponding points in the pair stereo images and $\mathtt{F}$ is the fundamental matrix. In this assignment, we estimate the $\mathtt{F}$ matrix in two different methods and then do an experiment on the number of trials/time taken for RANSAC [2]. All the codes for this assignment can be found here.

2 Eight-Point Algorithm

In this section, we first discuss about the theory of the eight-point algorithm, its normalized version and compute the fundamental matrix for multiple pairs of images.

2.1 Theory

The eight-point algorithm [3] is a well-known algorithm for estimating the fundamental matrix $\mathtt{F}$. Let $\mathbf{x} = (x, y, 1)^{\top}$ and $\mathbf{x}' = (x', y', 1)^{\top}$ and $\mathtt{F}$ be 3×3 matrix as shown in Equation 2.

$$\mathtt{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \tag{2}$$

If $\mathbf{x}$ and $\mathbf{x}'$ are known, then from Equation 1, we have the following expression.

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1)\mathbf{f} = 0 \tag{3}$$

where $\mathbf{f}$ is the 9-vector formed by the entries of $\mathtt{F}$ in row-major order. If we have 8 pairs of points in two stereo images, we can analytically calculate the $\mathtt{F}$ matrix. However, as mentioned by Hartley [4], this problem is often ill-conditioned because in most of the cases the x, y in $(x, y, 1)^{\top}$ is the order of 100s and 1000s (In this assignment, we have images of size 4000×6000). So, we need to normalize the coordinates before any computation.

In this assignment, we divide the coordinates by their corresponding dimension, so that they lie in the interval [0,1]. This normalization can be represented, as shown in Equation 4.

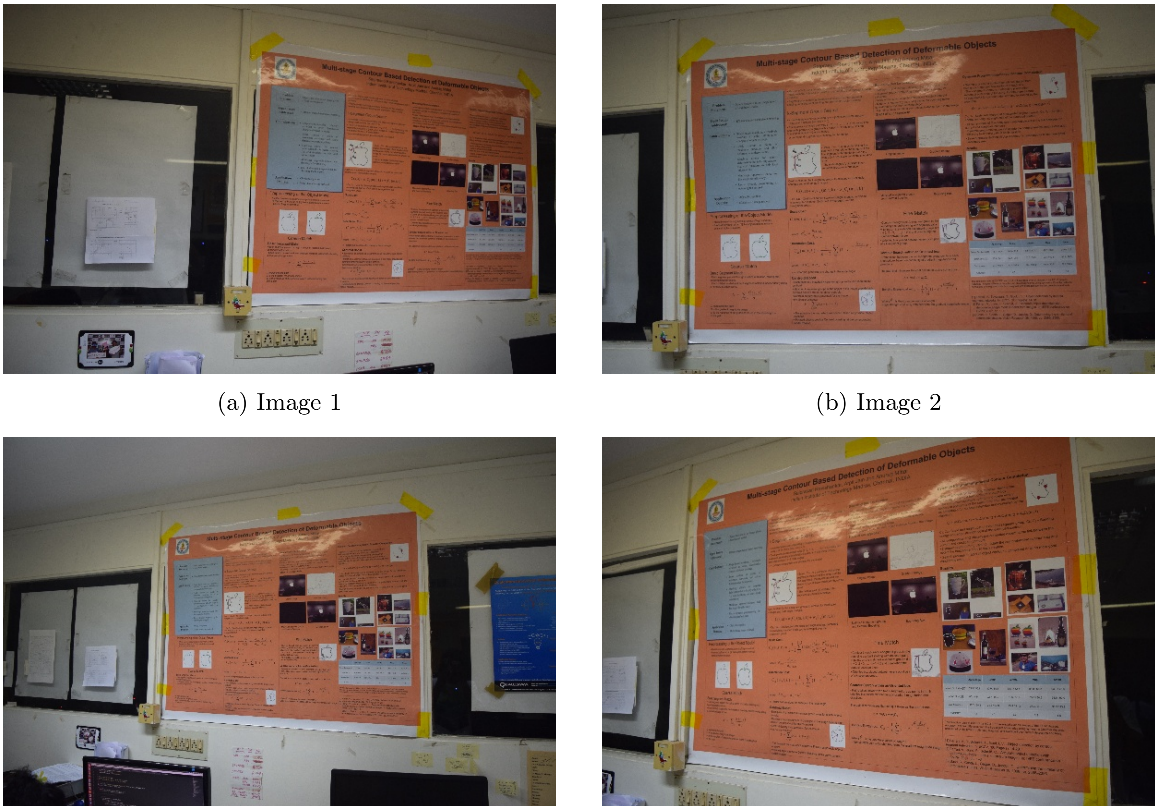
$$\hat{\mathbf{x}} = \mathtt{T}\mathbf{x}, \quad \hat{\mathbf{x}}' = \mathtt{T}\mathbf{x}' \tag{4}$$

where $\mathtt{T}$ is a diagonal matrix with entries [1/W, 1/H, 1] and W and H are the width and height of the image respectively. We compute the fundamental matrix $\mathtt{F}$' for the normalized coordinates and prove that $\mathtt{F} = \mathtt{T}^{\top}\mathtt{F}'\mathtt{T}$.

Proof.

$$\begin{aligned} \hat{\mathbf{x}} &= \mathtt{T}\mathbf{x} \\ \hat{\mathbf{x}}' &= \mathtt{T}\mathbf{x}' \\ \hat{\mathbf{x}}'^{\top}\mathtt{F}'\hat{\mathbf{x}} &= 0 \\ \implies (\mathtt{T}\mathbf{x}')^{\top}\mathtt{F}'\mathtt{T}\mathbf{x} &= 0 \\ \implies \mathbf{x}'^{\top}\mathtt{T}^{\top}\mathtt{F}'\mathtt{T}\mathbf{x} = 0 \; and \; \mathbf{x}'^{\top}\mathtt{F}\mathbf{x} &= 0 && \text{(From Equation 1)} \\ \implies \mathtt{F} &= \mathtt{T}^{\top}\mathtt{F}'\mathtt{T} \end{aligned}$$

□



(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

Figure 1: Stereo images from different view points

2.2 Computing Fundamental Matrix $\mathtt{F}$

We compute the fundamental matrix between all possible pair of images as shown in Figure 1. The images are numbered as shown in the figure and $\mathtt{F}$ is computed such that $\mathbf{x}$ come from the lower numbered image and $\mathbf{x}'$ come from the higher numbered image for the sake of brevity as the fundamental matrix for the other order pair is the transpose of the first matrix. For example, if we choose image 1 and 3 for our purpose, $\mathbf{x}$ comes from image 1 and $\mathbf{x}'$ comes from image 3. The computed fundamental matrices are presented in Table 1.

3 Automatic Estimation of Fundamental Matrix

In this section, instead of manually choosing eight points for computing the fundamental matrix, we can use off-the-shelf feature descriptors like SIFT [5] or SURF [6] or ORB [7] for getting the corresponding matches in a pair of images. In this assignment, we use ORB feature descriptor from the OpenCV[1] library. We first obtain a sample of 500 features for both the images. Then the matches are found using a brute force approach taking the Hamming distance between a pair of descriptors into account. In this method, about $\sim 55\%$ of all the matches are retained.

After getting a list of correspondences between a pair of images, we use RANSAC [2] for obtaining the fundamental matrix $\mathtt{F}$. For deciding, if a point is inlier or not, we use the following criterion: $|\mathbf{x}'^{\top}\mathtt{F}\mathbf{x}| < \tau$, where τ is the threshold of error. After getting a set of n matched inliers, we use Equation 3 to build a $n \times 9$ matrix $\mathtt{A}$ and solve $\min_{\|\mathtt{f}\|_2=1} \|\mathtt{Af}\|_2$, which is given by *the right singular vector of* $\mathtt{A}$. The default values of the threshold of error is 0.05 and maximum number of trials is 1000 with which get $\sim 95\%$ matches between all the pairs of images. The fundamental matrices between pairs of images are given in Table 2.

[1] https://opencv.org/

Table 1: Table of fundamental matrices between pairs of images using 8-point algorithm

Image I	**Image II**	**Fundamental Matrix** $\mathtt{F}$
1	2	$\begin{bmatrix} 7.059\times10^{-8} & -6.971\times10^{-9} & -3.312\times10^{-4} \\ 9.917\times10^{-9} & 2.163\times10^{-9} & -6.057\times10^{-5} \\ -2.141\times10^{-4} & 3.587\times10^{-5} & 1 \end{bmatrix}$
1	3	$\begin{bmatrix} 6.593\times10^{-8} & -1.036\times10^{-8} & -2.958\times10^{-4} \\ 1.418\times10^{-8} & 1.589\times10^{-9} & -7.324\times10^{-5} \\ -2.216\times10^{-4} & 3.521\times10^{-5} & 1 \end{bmatrix}$
1	4	$\begin{bmatrix} 7.436\times10^{-8} & -1.418\times10^{-8} & -3.246\times10^{-4} \\ 1.642\times10^{-8} & 1.793\times10^{-9} & -7.902\times10^{-5} \\ -2.282\times10^{-4} & 3.991\times10^{-5} & 1 \end{bmatrix}$
2	3	$\begin{bmatrix} 1.620\times10^{-9} & -1.582\times10^{-6} & 2.234\times10^{-3} \\ 1.517\times10^{-6} & -4.188\times10^{-8} & -5.223\times10^{-3} \\ -2.457\times10^{-3} & 5.339\times10^{-3} & 1 \end{bmatrix}$
2	4	$\begin{bmatrix} 1.128\times10^{-7} & -5.964\times10^{-8} & -1.072\times10^{-4} \\ 4.382\times10^{-8} & -4.474\times10^{-9} & -1.305\times10^{-4} \\ -5.478\times10^{-4} & 1.786\times10^{-4} & 1 \end{bmatrix}$
3	4	$\begin{bmatrix} -9.838\times10^{-9} & 5.693\times10^{-6} & -1.070\times10^{-2} \\ -5.458\times10^{-6} & -5.273\times10^{-8} & 1.718\times10^{-2} \\ 9.465\times10^{-3} & -1.624\times10^{-2} & 1 \end{bmatrix}$

Table 2: Table of fundamental matrices between pairs of images using RANSAC

Image I	**Image II**	**Fundamental Matrix** $\mathtt{F}$
1	2	$\begin{bmatrix} 1.683\times10^{-8} & 2.772\times10^{-8} & -1.238\times10^{-4} \\ 2.789\times10^{-8} & 4.889\times10^{-8} & -2.133\times10^{-4} \\ -1.337\times10^{-4} & -2.259\times10^{-4} & 1 \end{bmatrix}$
1	3	$\begin{bmatrix} 1.212\times10^{-8} & 2.496\times10^{-8} & -1.011\times10^{-4} \\ 3.553\times10^{-8} & 6.998\times10^{-8} & -2.905\times10^{-4} \\ -1.218\times10^{-4} & -2.424\times10^{-4} & 1 \end{bmatrix}$
1	4	$\begin{bmatrix} 1.226\times10^{-8} & 2.314\times10^{-8} & -9.802\times10^{-5} \\ 3.453\times10^{-8} & 6.518\times10^{-8} & -2.827\times10^{-4} \\ -1.232\times10^{-4} & -2.337\times10^{-4} & 1 \end{bmatrix}$
2	3	$\begin{bmatrix} 1.662\times10^{-8} & 3.513\times10^{-8} & -1.478\times10^{-4} \\ 2.871\times10^{-8} & 6.607\times10^{-8} & -2.646\times10^{-4} \\ -1.090\times10^{-4} & -2.473\times10^{-4} & 1 \end{bmatrix}$
2	4	$\begin{bmatrix} -2.834\times10^{-8} & 2.041\times10^{-7} & -2.747\times10^{-4} \\ -1.435\times10^{-7} & 2.763\times10^{-7} & 7.311\times10^{-5} \\ 3.853\times10^{-4} & -1.372\times10^{-3} & 1 \end{bmatrix}$
3	4	$\begin{bmatrix} 1.169\times10^{-8} & 1.927\times10^{-8} & -7.515\times10^{-5} \\ 4.012\times10^{-8} & 9.106\times10^{-8} & -3.360\times10^{-4} \\ -1.332\times10^{-4} & -2.648\times10^{-4} & 1 \end{bmatrix}$

Table 3: Time taken and fraction of matches vs number of trials

Image I	**Image II**	**N**	**Time taken in seconds**	**Fraction of matches**
1	2	10^2	0.082 ± 0.022	0.921 ± 0.015
		10^3	0.466 ± 0.045	0.953 ± 0.004
		10^4	4.291 ± 0.428	0.959 ± 0.001
		10^5	37.755 ± 0.769	0.965 ± 0.001
1	3	10^2	0.116 ± 0.012	0.922 ± 0.019
		10^3	0.600 ± 0.045	0.942 ± 0.003
		10^4	4.641 ± 0.267	0.948 ± 0.001
		10^5	45.630 ± 3.161	0.954 ± 0.002
1	4	10^2	0.087 ± 0.019	0.918 ± 0.015
		10^3	0.546 ± 0.047	0.945 ± 0.006
		10^4	4.195 ± 0.357	0.951 ± 0.002
		10^5	39.968 ± 2.135	0.955 ± 0.001
2	3	10^2	0.081 ± 0.011	0.910 ± 0.032
		10^3	0.575 ± 0.047	0.944 ± 0.009
		10^4	4.532 ± 0.418	0.963 ± 0.009
		10^5	42.064 ± 1.412	0.974 ± 0.002
2	4	10^2	0.115 ± 0.039	0.881 ± 0.012
		10^3	0.866 ± 0.086	0.918 ± 0.006
		10^4	8.104 ± 0.385	0.932 ± 0.004
		10^5	79.796 ± 3.257	0.944 ± 0.005
3	4	10^2	0.077 ± 0.009	0.900 ± 0.029
		10^3	0.546 ± 0.060	0.942 ± 0.003
		10^4	4.077 ± 0.181	0.954 ± 0.004
		10^5	39.269 ± 1.629	0.955 ± 0.002

It is to be noted that the default value of $N = 10^3$ which generates the `F` matrices presented in Table 2.

4 Time taken vs Number of Trials in RANSAC

In the RANSAC algorithm, the maximum number of allowed trials, N plays a vital role in deciding the cost and performance of the algorithm. So, we experiment with a wide range of N and gather inferences. We run each experiment for a pair of images for five times for a fixed value of N and note the mean and standard deviation of the time taken and the fraction of matches. The detailed observations are presented in Table 3.

5 Discussion and Conclusion

In this assignment, we calculated the fundamental matrix using two different approaches namely the eight-point algorithm and using ORB feature descriptor along with RANSAC. After that, we did an experiment by varying the maximum number of trials, N and observed its effects on the time taken as well as the performance. The main inferences can be listed as follows:

- It is observed that the results of the *eight-point algorithm is very sensitive* to the choice of the points. Even a small change, either intentional or not, can result in a significantly different fundamental matrix `F`. This is the primary reason that the corresponding `F` in Table 1 and Table 3 are quite different.
- On the contrary, the *RANSAC method provides quite stable results.* The main attributing factors for the same is taking a much higher number of putative correspondences (∼280 compared to only 8 in the eight-point algorithm) and multiple random sampling.

- In the last part of the assignment, we run the RANSAC for a wide range of N and note the results. As expected, the time taken for a pair of images grows almost *linearly* for all the pairs of images.

- It is also worth mentioning the time taken also *depends on the number of putative correspondences* taken as input for the RANSAC algorithm. For example, in the fifth row, we can see the time taken for 10^5 trials is almost double compared to the corresponding time taken for the other pairs. This result can be explained because the fifth pair had 310 correspondences compared to the other five, which had $\sim$270 matches.

- The performance increases as N increases, which is expected. Another important point to be noted is that the standard deviation when $N = 10^5$ is quite less as compared to when $N = 10^2$. A possible explanation for the same can be as follows: as the N increases, more "sets" of eight points are sampled and hence more stable results are obtained.

References

[1] Fundamental matrix (computer vision) — Wikipedia, the free encyclopedia. [Online; 01-April-2020].

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[3] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[4] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[7] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.