

COMPUTER VISION LABORATORY REPORT N.6

FUNDAMENTAL MATRIX ESTIMATION

May 31, 2019

Claudio Curti ID: 4216203
Nicola De Carli ID: 4198668
Alberto Ghiotto ID: 4225586
University of Genoa

Contents

1	Introduction	2
1.1	Defining the objective	2
2	Implementation and Results	3
2.1	EightPointsAlgorithm	4
2.2	EightPointsAlgorithmN	4
2.3	checkEpipConstr	4
2.4	RansacF	4
2.5	Results	6

Chapter 1

Introduction

1.1 Defining the objective

The goal of this project is to estimate the fundamental matrix, given couples of corresponding points, using different variants of the same algorithm and in different condition (having only right corresponding points or having also fake correspondences). In order to perform this operation it's necessary to understand and implement the 8 point algorithm. In the interest of making sure that the implemented algorithms has produced the right results, some checks have been performed, among which the epipolar constraint, the position of the epipoles and the visualization of the epipolar lines.

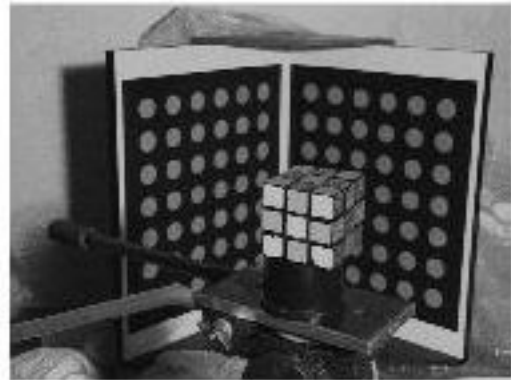
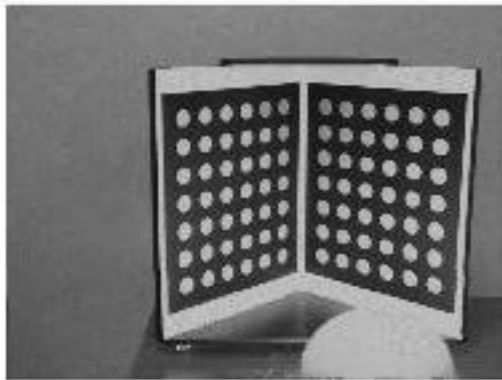


Figure 1.1: Main images used for the experimentation

Chapter 2

Implementation and Results

The code is organized using a script, which loads two sets of corresponding points and arranges them in two matrices P1 and P2 which will be used by another function called next to estimate the fundamental matrix. The whole images are also loaded in order to display the epipolar lines after that the fundamental matrix has been computed both in the case with the normalization of the points and without it.

In order to carry out all the required manipulation the following functions were developed:

- **EightPointsAlgorithm**: which simply implements the steps highlighted in the instructions text in order to perform the algorithm.
- **EightPointsAlgorithmN**: which implements the same algorithm as the previous function and adds the normalization of the points before filling the matrix A and the de-normalization of the resulting F at the very end.
- **checkEpiConstr**: which checks whether the epipolar constraint $x'^T F x = 0$ holds for all points with the estimated F.
- **RansacF**: which in case of erroneous correspondences, makes a distinction between inlier and outlier to compute the best model (the F matrix).

In addition to the ones listed above, two premade functions were provided in order to simplify the procedures:

- **normalise2dpt**: which is used in the function "*EightPointsAlgorithmN*" to normalize the points
- **visualizeEpipolarLines**: which allows to visualize the stereo pairs with epipolar lines of the corresponding points.

2.1 EightPointsAlgorithm

This function takes as inputs the two matrices of corresponding points organized by the main script in order to fill the matrix A with the relative data. It then performs the SVD decomposition in order to find the optimal solution of the homogeneous system in the least square sense. Since the rank of the estimated F, in general, will be three, but its rank should be two by definition, the F with rank two closer to the one estimated is obtained by setting the last singular value of the estimated one to zero. To do so it simply compute the SVD of the matrix F, set the last element of D to zero and recompute the final F as: $F' = U * D' * V^T$.

2.2 EightPointsAlgorithmN

This function is entirely similar to the previous one except for the point normalization part. Before filling the matrix A with the points taken as inputs from the main script it normalizes them using the function *"normalise2dpt"*. Then after having recomputed the final F as outlined in the previous section, it denormalizes the matrix as follows: $F = T'^T F' T$. This tweak is necessary in order to solve a problem due to numerical instability, i.e. if more than eight corresponding points are used to construct A, where the coordinates are only approximately correct, there may not be a well-defined singular value which can be identified as approximately zero. Therefore, the solution of the homogeneous linear system of equations may not be sufficiently accurate to be useful. In this case, one singular value of A could be very large with respect to the other. The criticality lies in the fact that in homogeneous coordinates the first two values of the point coordinates (x and y) vary in a range containing values much bigger than the last value which is always one. Therefore, the upside of using points normalization lies in the fact that the coordinates are evenly scaled so that the mean distance from the origin to a point equals $\sqrt{2}$. The A matrix obtained in this way, in general, has a better condition number than the previous one had.

2.3 checkEpipConstr

This function simply checks if the epipolar constraint $x'^T F x = 0$ holds (with a threshold of tolerance) for each point with the estimated F. The parameter *"maxerror"* allows to set a reasonable threshold value such that the constraint holds when correct points correspondences are used. The boolean output *"answer"* of this function is used afterwards to print on the command windows the results obtained.

2.4 RansacF

It is possible to notice how, by adding erroneous random correspondences to the real ones, the estimated fundamental matrix becomes wrong since all the points are used in

the estimation process. This function implements a solution to this problem which is to use the Ransac algorithm. This procedure is performed by taking a random permutation of 8 points correspondences, computing a model (F matrix) on this data and then checking how many correspondences between all the data fit to the obtained model. This process is iterated a certain number of times which depends also on the obtained results and eventually the best estimated model is saved. In this case to evaluate if a certain point is to be considered an inlier, the distance between a point and the corresponding epipolar line is compared with a certain threshold. By using the provided and user-defined evaluation functions (respectively `visualizeEpipolarLines` and `checkEpipConstr`), it is possible to notice that the model obtained in this way has to be considered reliable till the number of inlier is sufficiently greater than the number of outlier.

2.5 Results

The code has been initially tested with the "Mire" images and with the "Rubik" images later on. With the first set of images, a threshold of 0.1 has been applied in order to check the epipolar constraint (since a perfect 0 is almost impossible to obtain). As it can be seen in the following images the epipolar constraint is satisfied both with and without normalization to compute the fundamental matrix. However it is possible to notice a big difference in the slope of the epipolar lines (and the corresponding positions of the epipoles). By adding some random points to the correspondences (3), as requested in the optional task, the estimated F becomes very unreliable and the epipolar constraint is no longer satisfied, it is also possible to notice a very different position of the epipoles with respect to the previous cases (in this case the epipoles are almost always in the image plane). To overcome this problem the Ransac algorithm is used together with the algorithm containing the points normalization. The algorithm without points normalization is abandoned since it is less precise and adding random points would have worsened his results. By using the Ransac algorithm, the F matrix becomes again reliable (up until the number of inliers is considerably greater than the number of outliers), in fact, it is possible to notice a certain similarity in the slope of the epipolar lines with respect to the case in which only real correspondences and point normalization were used. Last but not least, the epipolar constraint is again satisfied.

```

EPIPOLAR CONSTRAINT CHECKING WITH ORIGINAL POINTS
The epipolar constrain (without norm) is respected
The epipolar constraint (with norm) is respected
EPIPOLAR CONSTRAINT CHECKING WITH INVENTED POINTS:
The epipolar constraint (with norm, no Ransac) is NOT respected
The epipolar constrain (with norm + Ransac) is respected

```

Figure 2.1: Epipolar constraint results using "Mire"

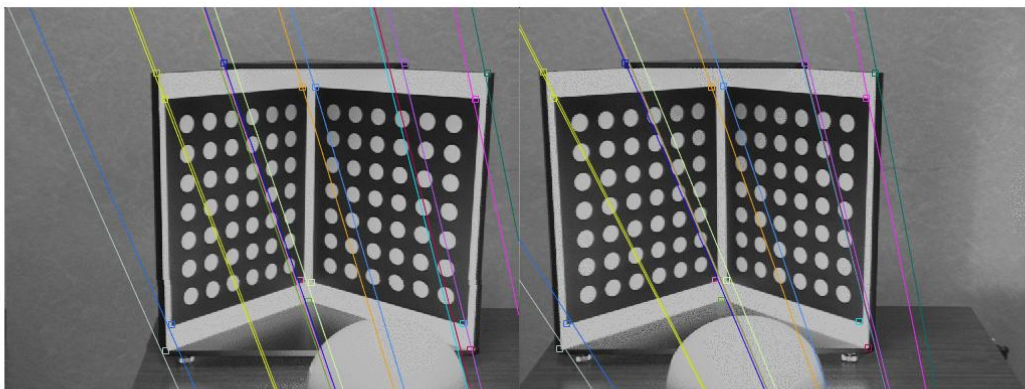


Figure 2.2: Epipolar lines without points normalization

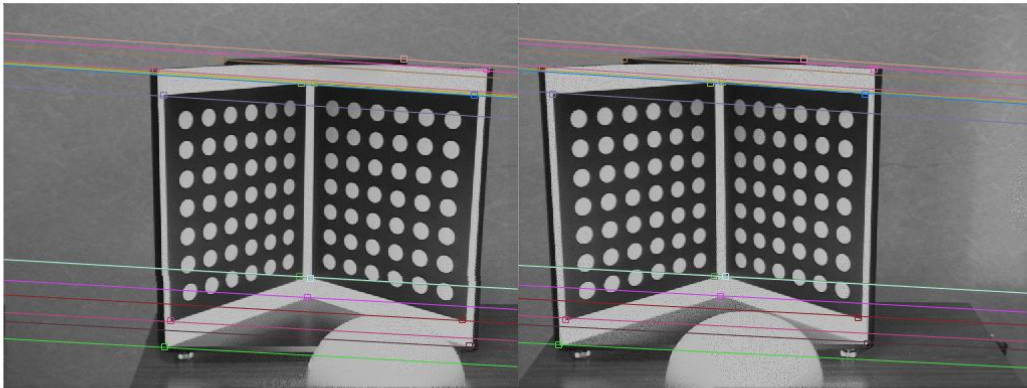


Figure 2.3: Epipolar lines with points normalization

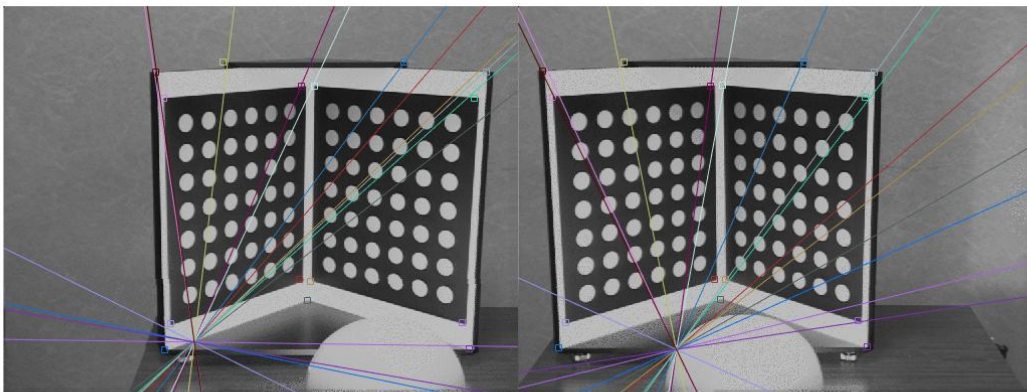


Figure 2.4: Epipolar lines without Ransac (with random points)

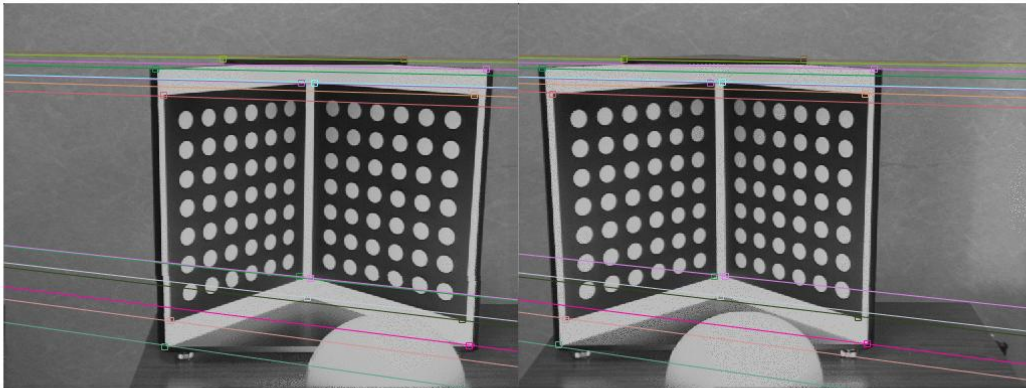


Figure 2.5: Epipolar lines with Ransac (with random points)

In the second part of the experimentations, the "Rubik" set of images has been used. Using the same threshold as before, it is possible to see that the epipolar constraint in the case without normalization isn't satisfied; in fact the result is significantly different from zero compared to the algorithm which uses the points normalization (the difference is about one order of magnitude). The latter passes the epipolar constraint test both in the case with the original points and the random points. Regarding the epipolar lines, as in the "Mire" set of images, it is clear that the Ransac algorithm allows to obtain a better precision even in the case in which some random points are added to the correspondences, as it can be seen in the following images.

```
EPIPOLAR CONSTRAINT CHECKING WITH ORIGINAL POINTS
The epipolar constrain (without norm) is NOT respected
The epipolar constraint (with norm) is respected
EPIPOLAR CONSTRAINT CHECKING WITH INVENTED POINTS:
The epipolar constraint (with norm, no Ransac) is NOT respected
The epipolar constrain (with norm + Ransac) is respected
The epipolar constrain (with norm) is respected
```

Figure 2.6: Epipolar constraint results using "Rubik"

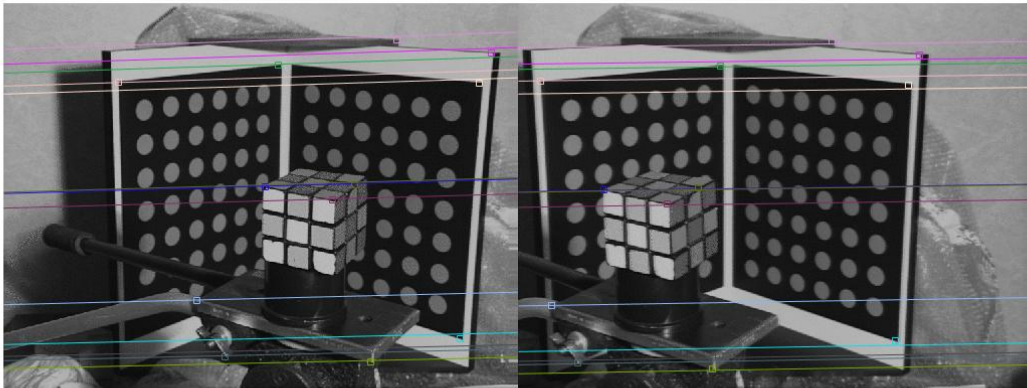


Figure 2.7: Epipolar lines without points normalization

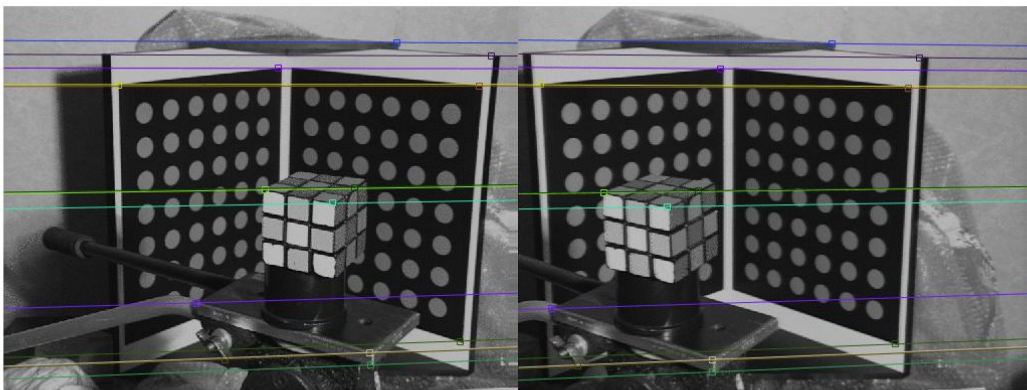


Figure 2.8: Epipolar lines with points normalization

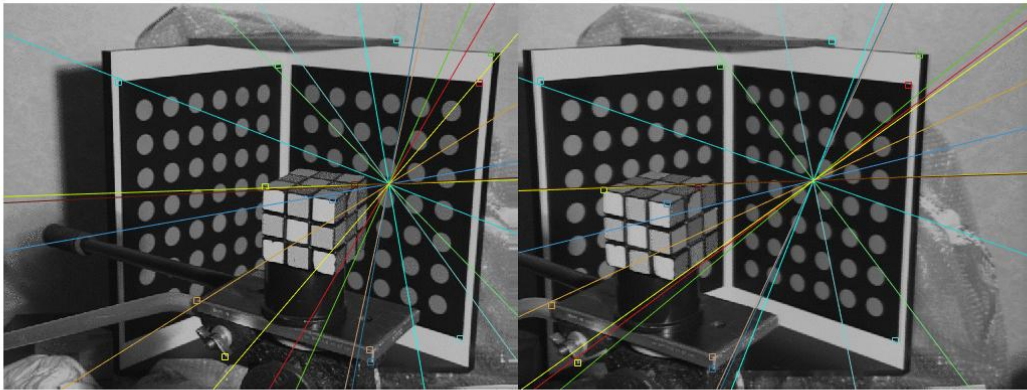


Figure 2.9: Epipolar lines without Ransac (with random points)

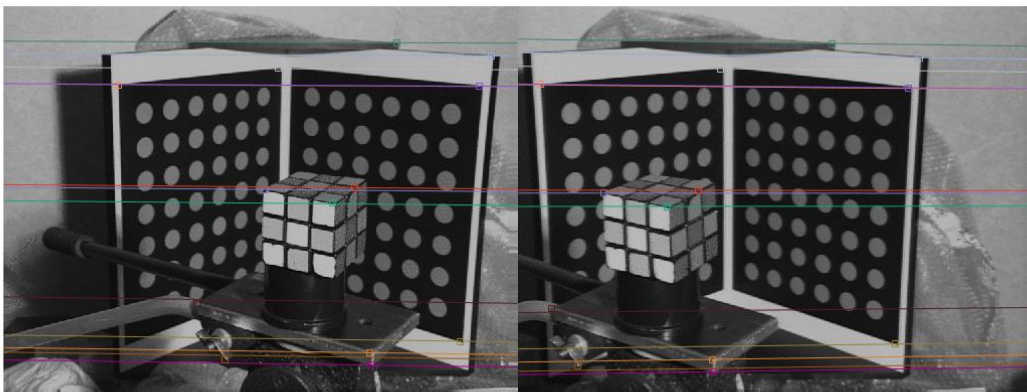


Figure 2.10: Epipolar lines with Ransac (with random points)

Eventually, the predefined Matlab function "estimateFundamentalMatrix" has been tested in both the set of images only in the case with the adjoint of random points. Since this function uses the Ransac algorithm to exclude the outliers, it provided results pretty similar to the ones obtained with the algorithms described previously ("EightPointsAlgorithmN" and "RansacF"), though they are not identical. This is probably due to the intrinsic randomness of Ransac, in fact, even if the same function is performed twice it won't give identical results. Moreover, the epipolar constraint has been satisfied and the epipolar lines are shown in the following images.

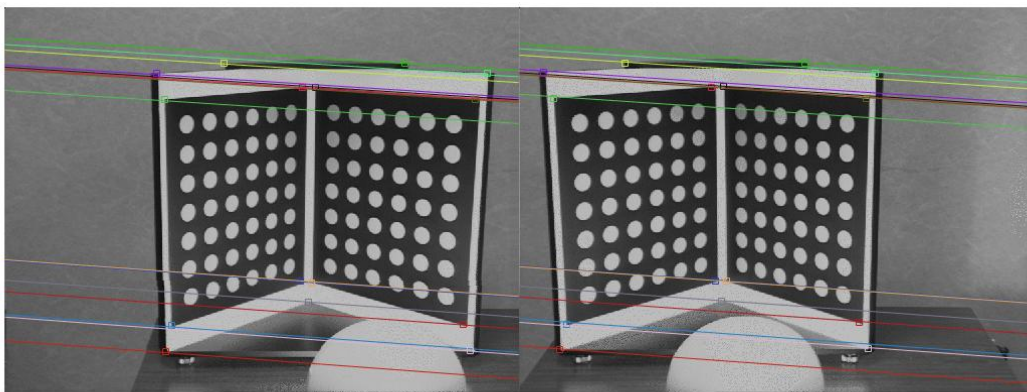


Figure 2.11: Epipolar lines with estimateFundamentalMatrix

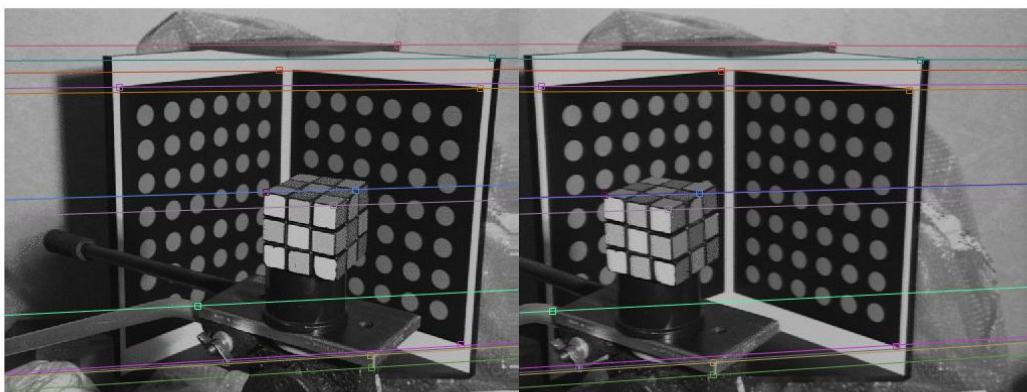


Figure 2.12: Epipolar lines with estimateFundamentalMatrix