Computer Vision: Lab 6
Fundamental matrix estimation

Francesca Canale 4113133
Filippo Gandolfi 4112879
Marco Giordano 4034043

06/06/2019

1 Introduction

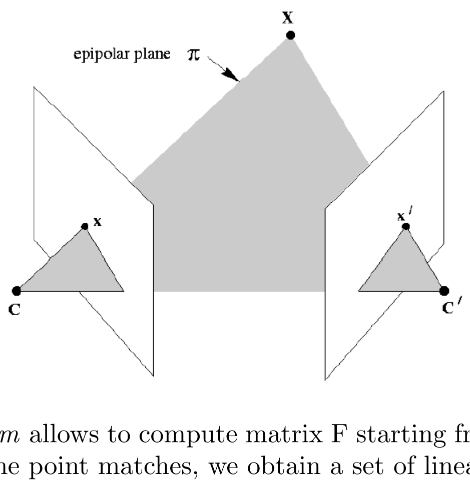
The goal of this lab is to understand and implement the 8 points algorithm, to estimate the fundamental matrix F. The fundamental matrix is a 33 matrix of rank 2 which relates corresponding points in stereo images. If a point in 3-space X is imaged as x in the first view, and x in the second, then the image points satisfy the relation

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

where:

$$\mathbf{x} = (x, y, 1)'$$

$$\mathbf{x'}\text{=(x' y' 1)}$$



The *Eight points algorithm* allows to compute matrix F starting from 8 points correspondences and knowing that from all the point matches, we obtain a set of linear equations of the form :

$$\mathbf{Af = 0}.$$

where f is a nine-vector containing the entries of the matrix F, and A is the equation matrix.

$$\begin{bmatrix} x1'x1 & x1'y1 & x1' & y1'x1 & y1'y1 & y1' & x1 & y1 & 1 \\ x2'x2 & x2'y2 & x2' & y2'x2 & y2'y2 & y2' & x2 & y2 & 1 \\ . & . & . & . & . & . & . & . & . \\ xn'xn & xn'yn & xn' & yn'xn & yn'yn & yn' & xn & yn & 1 \end{bmatrix} \begin{bmatrix} f11 \\ f12 \\ . \\ . \\ f33 \end{bmatrix} = 0$$

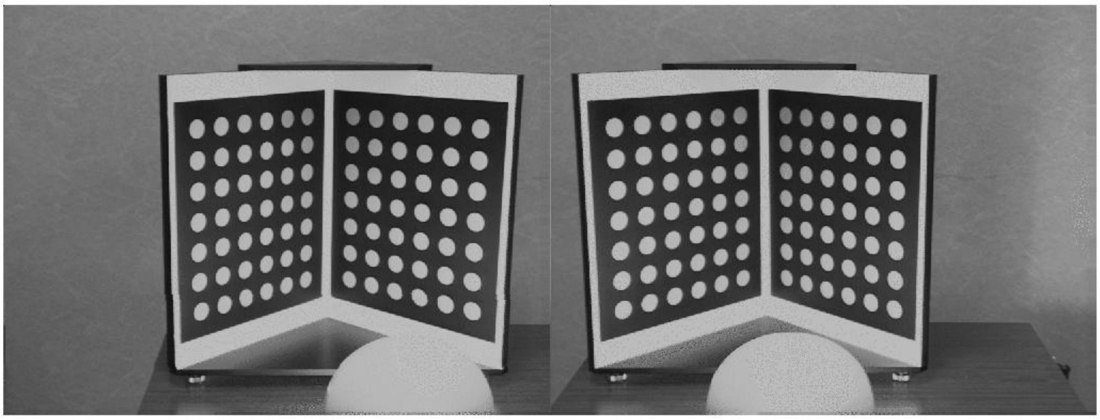


Figure 1: Mire image

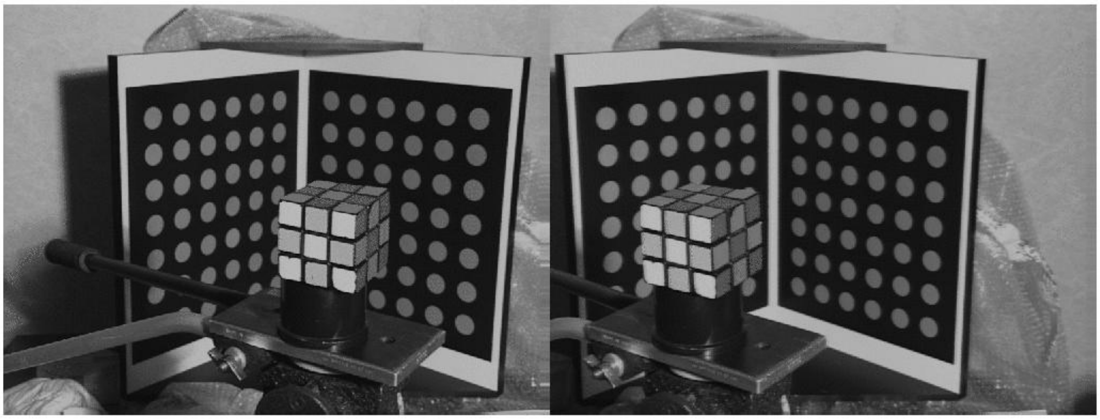


Figure 2: Rubik image

2 Eight points algorithm function, version 1

We implemented a function $EightPointsAlgorithm(P1, P2)$ that:

- Creates matrix A from the points correspondences given by P1 and P2;
- Computes the singular value decomposition of A;

$$[U, D, V] = svd(A)$$

- Selects as solution f of the equation Af=0 the last coloumn of V;
- Reshapes the column vector f in order to obtain the 3x3 matrix F;
- Since Af = 0 is a overdetermined problem, which means it has more solutions, the algorithm finds the one that minimizes a cost function: it forces the rank of F to be two, it computes

the svd of F, the obtained diagonal matrix D has values in decreasing order, so a solution which minimize the cost is the one with D(3,3) = 0;

- It calculates the final F as:

$$F = U * D * V^{\mathrm{T}}$$

3 Eight points algorithm function, version 2

This second version is exactly as the previous one with the only differences that the points are normalized with the given function $[nP, T] = normalise2dpts(P)$ and at the end the resulting F is "de-normalized" to obtain the final F:

$$F = T2^{\mathrm{T}} * F * T1$$

4 Evaluation of the results

The calculation of F is good if $x'^{\mathrm{T}} * F * x$ is near to zero.

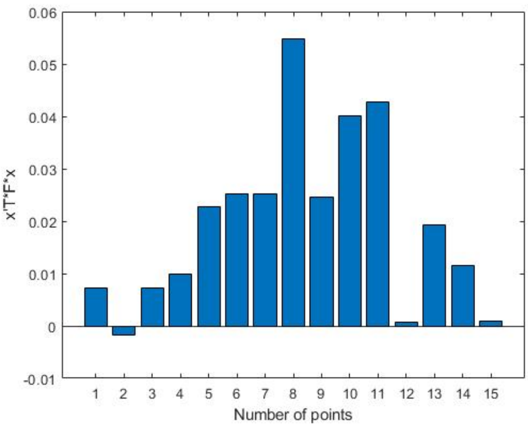


Figure 3: Result on Mire with version 1

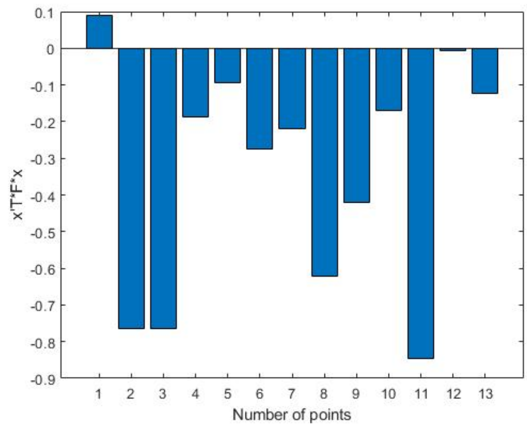


Figure 4: Result on Rubik with version 1

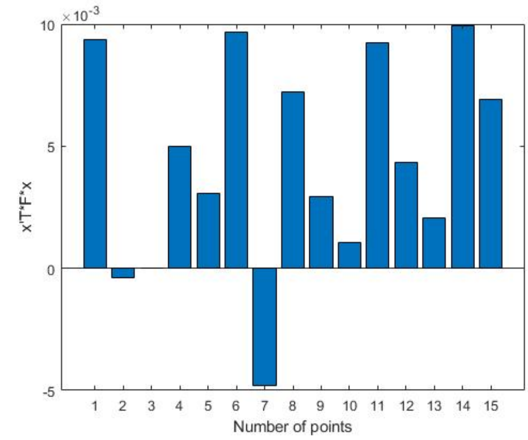


Figure 5: Result on Mire with version 2

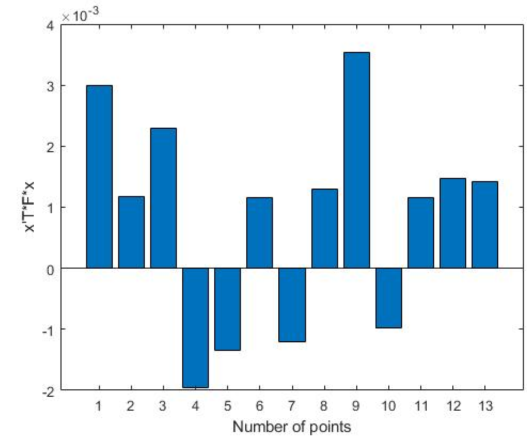


Figure 6: Result on Rubik with version 2

We can clearly notice that the results obtained with the second version, the normalized one, of the algorithm are way better than the first one.

4.1 Epipolar Lines

Thanks to the given function $VisualizeEpipolarLines$ we are able to visualize the stereo pairs of images with the epipolar lines of the corresponding points.

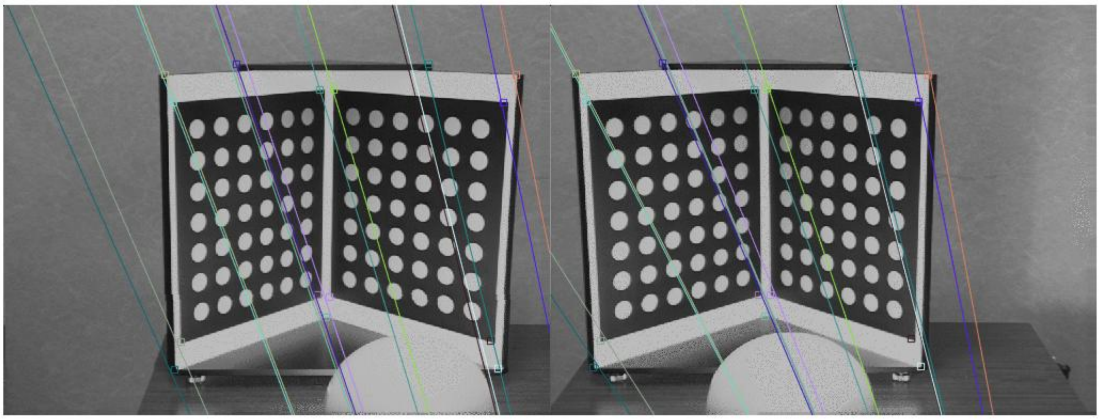


Figure 7: Mire: epipolar lines version 1

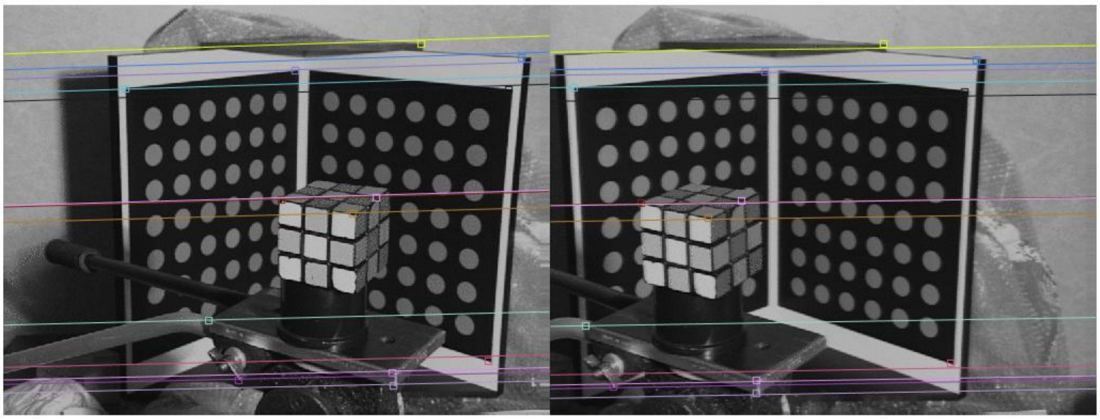


Figure 8: Rubik: epipolar lines version 1

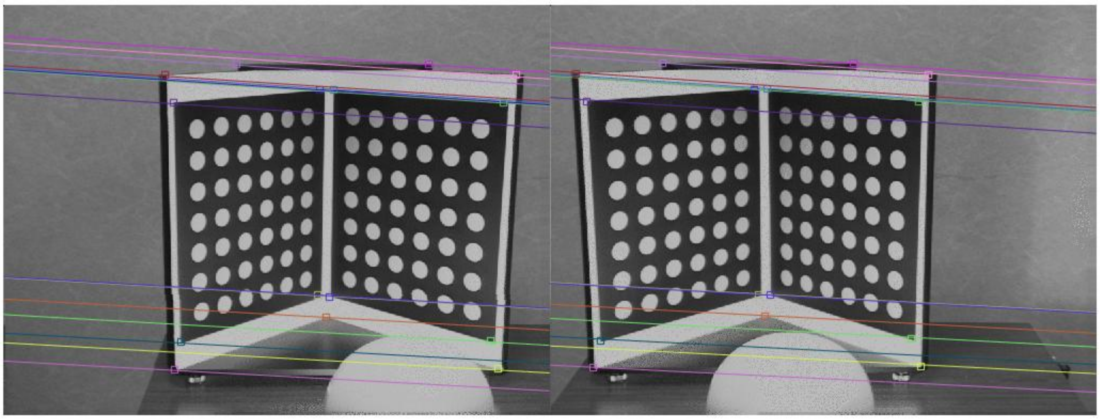


Figure 9: Mire: epipolar lines version 2

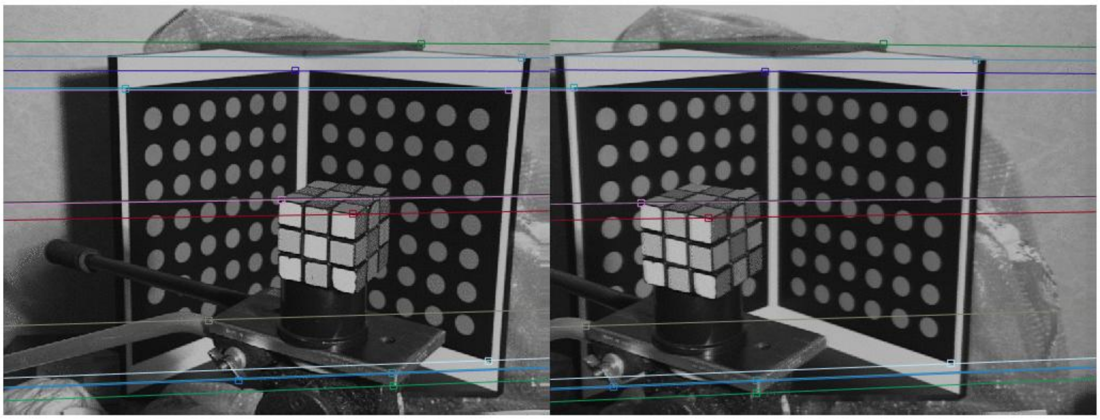


Figure 10: Rubik: epipolar lines version 2

With the first version of the algorithm the result for the Rubik image is acceptable; but it is not for the Mire's one: some line are far from their points. This means that it is a more "difficult" pair and it needs a better algorithm: the normalized version. Indeed the results with the second version are better.

Then we computed the left and the right epipoles that are, respectively the last column of V and U, because they are the right and left null space of F.

- Mire:
  Left epipole: Version 1 [0.4 0.91 $3.23*10^{-04}$] Version 2 [0.99 0.06 $2.55*10^{-05}$]
  Right epipole: Version 1 [0.44 0.89 $5.47*10^{-04}$] Version 2 [0.99 0.07 4.27^{-05}]

- Rubik:
  Left epipole: Version 1 [0.99 -0.03 $-2.24*10^{-05}$] Version 2 [-1.0 -0.009 $-8.7*10^{-05}$]
  Right epipole: Version 1 [0.99 -0.01 $1.09*10^{-05}$] Version 2 [-0.99 -0.01 -1.0610^{-04}]

We can notice that the third component is almost zero, which means that point are almost at infinite: this is reasonable because epipolar lines are almost parallel.

5 Some erroneus points

We added a very erroneous correspondence point ([600,600] [0, 0]) to the ones provided for the images in order to test what happen with MATLAB function *estimateFundamentalMatrix*. Giving as input the RANSAC method the results are good: this statistic method finds the outlier couples and not count them for the estimation of fundamental matrix.

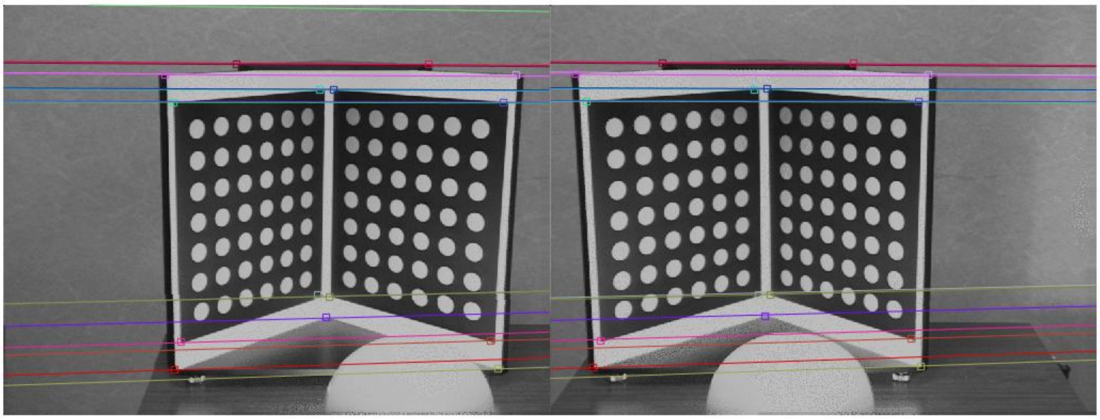


Figure 11: Mire: epipolar lines RANSAC

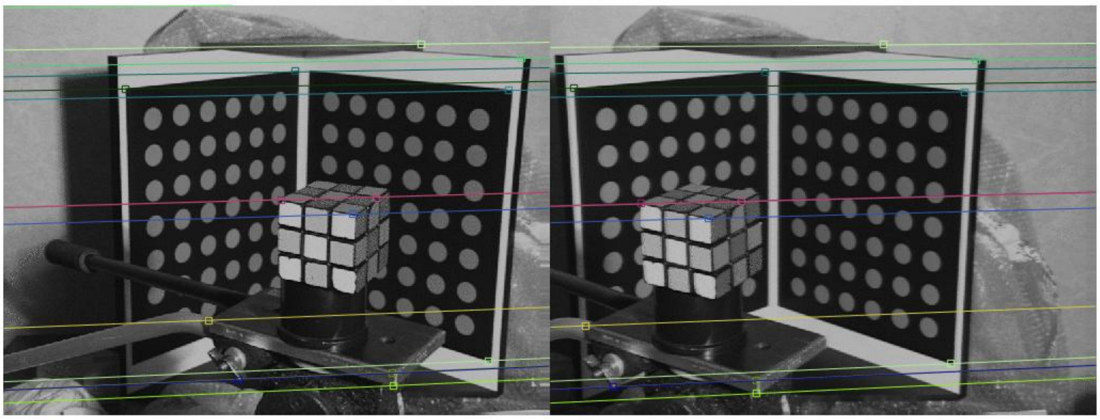


Figure 12: Rubik: epipolar lines RANSAC