Computer Vision Laboratory Report n.6

FUNDAMENTAL MATRIX ESTIMATION

June 7, 2020

Chiara Terrile ID: 4337786
Giulia Scorza Azzarà ID: 4376318
University of Genoa

Contents

1 Introduction **2**
1.1 Defining the objective 2

2 Implementation **3**
2.1 EightPointsAlgorithm 4
2.2 EightPointsAlgorithmN 4
2.3 checkEpConst 5

3 Results **6**

Chapter 1

Introduction

1.1 Defining the objective

The objective of this laboratory is to estimate the fundamental matrix, given couples of corresponding points, using different variants of the same algorithm.
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 algorithm 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.

Chapter 2

Implementation

For the implementation is used a **script** where are first loaded two sets of corresponding points. The first set of points is loaded as *P1i* and *P2i*. The second one is more precisely composed of two images, *I1* and *I2* (shown in Figure 2.1 and Figure 2.2). These two sets are then organized into two matrices *P1* and *P2* in order to apply the two functions **EightPointsAlgorithm** and **EightPointsAlgorithmN** to compute the respective fundamental matrices F1 and F2.
Subsequently it is called the function **checkEpConst** to check if the epipolar constraint for all points with the estimated fundamental matrix is verified.
After this check, all the epipolar lines are plotted in both images (I1 and I2), using again both F1 and F2 to highlight the differences between them. To perform this operation, we have used a premade function called **visualizeEpipolarLines**, which takes as input two images, the fundamental matrix and two sets of points, giving as result the plot of the epipolar lines on the original images.
Finally in the script are computed right and left epipoles of F1 and F2 in order to be able to make a comparison between them. To achieve this goal, it is sufficient to perform the SVD (Singular Value Decomposition) of the matrices F1 and F2 and then to extract the last columns of U and V, obtaining the right and left null space of the fundamental matrix. Above all the function that we have implemented are described:

- **EightPointsAlgorithm**: which simply implements the steps highlighted in the instructions text in order to perform the algorithm and to obtain the fundamental matrix.

- **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.

- **checkEpConst**: which checks whether the epipolar constraint $x'^T F x = 0$ holds for all points with the estimated F.

Note : The code has been initially tested with the "Mire" images and with the "Rubik" images later on.

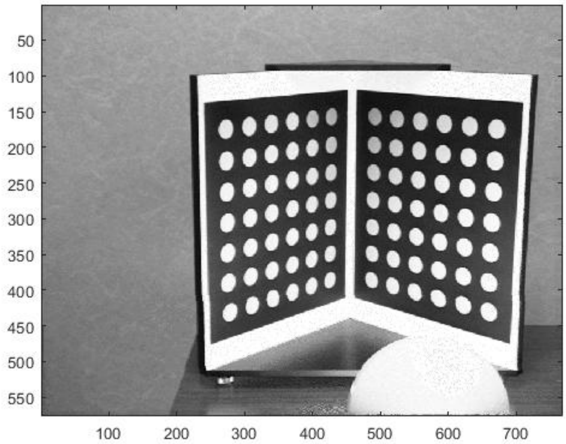

Figure 2.1: Mire1 (I1)

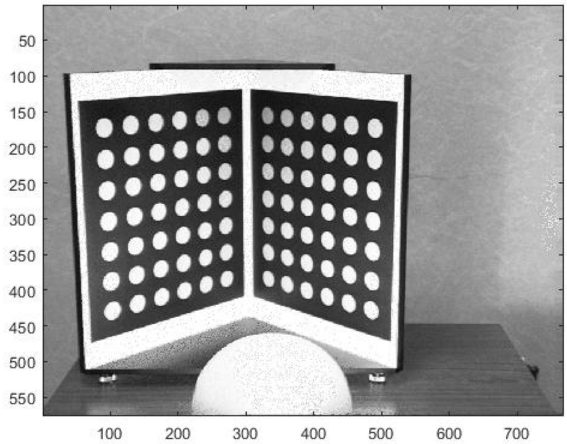

Figure 2.2: Mire2 (I2)

2.1 EightPointsAlgorithm

This function takes as input the two matrices of the corresponding points *P1* and *P2* and gives as output the fundamental matrix F.
P1 and P2 are used in order to fill the matrix A with the proper values. Then it is applied the SVD to the matrix A in order to select the last column of V (f in the code), which will be reshaped with the MATLAB function **reshape**.
The following step consists in forcing the rank of the matrix F (the one obtained after reshaping) to be equal to 2, and this can be obtained simply imposing that the element in position (3,3) of the diagonal matrix D (from SVD of F) is equal to zero.
At the end is computed the final value of F, obtained as follows: $F = U * D * V^T$.

2.2 EightPointsAlgorithmN

This function works in the same way of the previous one but with some differences.
The inputs and the output are the same, but in this case is used the normalization.
First of all, the matrix A is no more filled with the matrices *P1* and *P2*, but with two new normalized matrices *nP1* and *nP2*. These two matrices are obtained with a premade function called **normalise2dpts**, which takes as input a set of points and gives as output a new set of normalized points and a matrix T, which is the one defining the normalization.
The following steps are exactly the same as before, but at the end instead of obtaining directly the matrix F, it's necessary to de-normalize it applying the following formula:

$$F = T2^T * F * T1$$

where *T1* and *T2* are the outputs of the function "normalise2dpts" previously defined.

2.3 checkEpConst

This function takes as input two sets of points *P1* and *P2* and the fundamental matrix F and gives as output a boolean variable (*answer*), which remains true if the condition of the epipolar constraint is verified.
In this function it's simply checked if the constraint $x'^T F x = 0$ (with a threshold of tolerance) holds for all the points belonging to P1 and P2 with a certain F.

Chapter 3

Results

After running the code the following images (Figure 3.1 and Figure 3.2) are obtained.

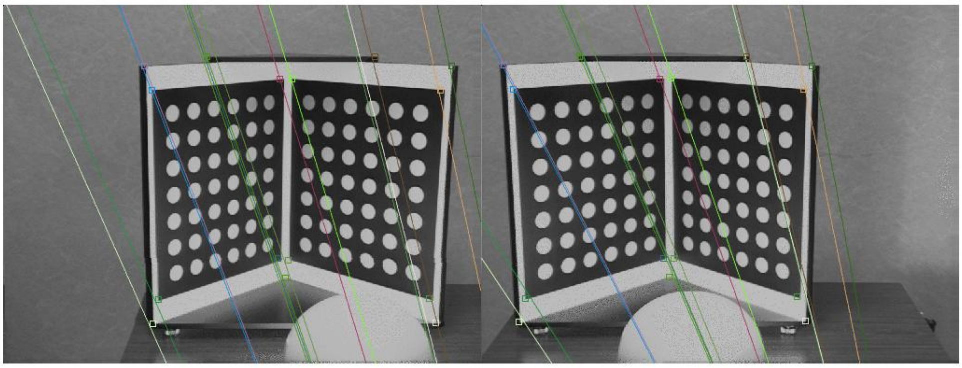

Figure 3.1: Epipolar lines without normalization

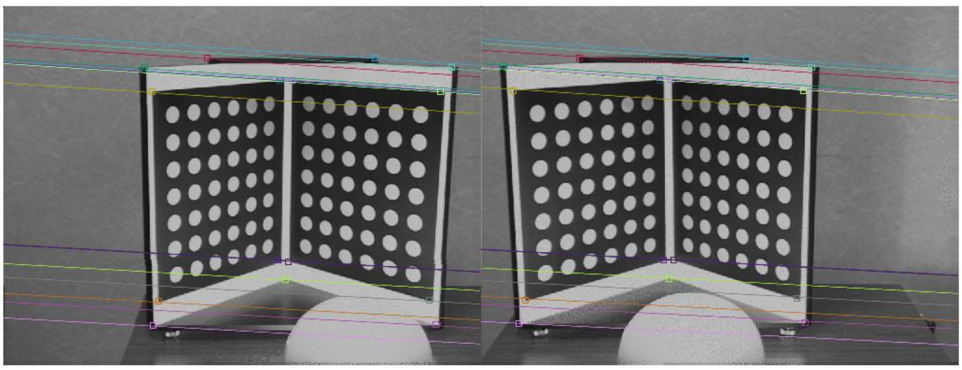

Figure 3.2: Epipolar lines with normalization

As emerges from the images, there is an evident variation in the slope of the epipolar lines from the Figure 3.1 to the Figure 3.2 due to the normalization process which implies translation and rotation.

What is obtained from the check of the epipolar constraint and from the computation of the epipoles is the following:

The epipolar constraint (without normalization) is respected
The epipolar constraint (with normalization) is respected
The right epipoles (without normalization) are: 0.44869, 0.89369, 0.00054724
The left epipoles (without normalization) are: 0.40907, 0.9125, 0.00032371
The right epipoles (with normalization) are: 0.99747, 0.071135, 4.277e-05
The left epipoles (with normalization) are: 0.99774, 0.067244, 2.557e-05

As far as it concerns the epipolar constraint, it is respected in both cases, but it has been necessary the addition of a threshold of tolerance equal to 0.1, since a perfect 0 is almost impossible to obtain.
Regarding the epipoles, due to the change of slope of the epipolar lines, also the position of the right and left epipoles has changed from the case without normalization to the normalized one.