Università degli studi di Genova

Computer Vision

Assignment 6: Fundamental Matrix Estimation .

Aurora Bertino, Chiara Saporetti, Silvana Andrea Civiletto

May 27, 2020

Introduction

0.1 The fundamental Matrix

The fundamental matrix is a 3x3 matrix which relates corresponding points (x and x), in a stereo image pair. Fx describes a line (an epipolar line) on which the corresponding point x on the other image must lie. That is valid, for all pairs of corresponding points.

$$(x')^T Fx = 0$$

The fundamental matrix is of rank 2 and its kernel defines the epipole.

0.2 The 8-points algorithm

The eight-point algorithm is an algorithm used to estimate the essential matrix or the fundamental matrix related to a stereo camera pair from a set of corresponding image points.
Characteristics of matrix F:

- *F is a 3x3 homogeneous matrix. Homogeneous means there is a scale ambiguity in the matrix, so the scale doesn't matter.*
- *F is a matrix with rank 2. It is not a full rank matrix, which means that it is singular and its determinant is zero.*
  The reason why F is a matrix with rank 2 is that it is mapping a 2D plane (the first image) to all the lines (in the second one) that pass through the epipole (of the second image).

0.3 Matlab Resolution

0.4 The 8-points algorithm

The goal of this assignment is to better understand and implement the 8-points algorithm, to estimate the fundamental matrix F. Two different versions of the implementation of this algorithm are requested. The

function which describes the first version of the algorithm is the following:

$$F1 = EightPointsAlgorithm(P1, P2) \tag{1}$$

The second function for the algorithm with normalization is the following:

$$F2 = EightPointsAlgorithmN(P1, P2) \tag{2}$$

The input of these functions are the same and they consist of a set of points' coordinates which will be normalized within the function.

$$x = (x, y, 1)^T and x' = (x', y', 1)^T$$

In the assignment' s text is requested to write the matrix A defined as following:

$$A = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \tag{3}$$

$$A = \begin{bmatrix} f_1 1 \\ f_2 2x_2 \\ \vdots \\ f_3 3x \end{bmatrix} = 0 \tag{4}$$

This is an homogeneous system which may be solved with the Singular Value Decomposition method, for which Matlab provides a function svd(). So in our function we apply the SDV on A, then we select as solution the last column of V and use it to reshape the column vector f. Then we apply again the SVD on f to recompute the fundamental matrix F and we force its rank to be equal to 2.
The only difference between the two functions lays in the last step of the second one, since it computes the final fundamental matrix F by de-normalizing the previously computed matrix F.

0.5 Evaluation of the results

To verify the implementation, it's required to check the epipolar constraint :

$$(x')^T F x = 0$$

for all points with the estimated F in the previous sections "Version 1" and "Version 2". In order to check the results for the "Version 1":

- we initialize a column vector of zeros with dimension [N,1].
- we fill through a loop (with N=15) each raw of the array with the formula that takes into account the constraint:

$$p2(:, i)' * F1 * p1(:, i); \tag{5}$$

In which p1 and p2 are the defined point in 3D in the two image planes.

- we display the epipolar constraint.

We make the same computations in order to check the results for the "Version 2", only the epipolar constraint changes because we use the Fundamental matrix F2. Subsequently, we used the given function "visualizeEpipolarLines" to show the the stereo pairs epipolarLines (in the images "img1" and "img2"):

$$visualizeEpipolarLines(img1, img2, F1, p1(1:2,:)', p2(1:2,:)') \tag{6}$$

$$visualizeEpipolarLines(img1, img2, F2, p1(1:2,:)', p2(1:2,:)') \tag{7}$$

Note: the function visualiseEpipolarLines has been slightly modified: we changed, in the code at line 13, figure(1) to figure in order to be able to plot all of the four images we wanted to use for the results analisys. The resto of the code remains unchanged since it was already provided.

In order to compute the left and right epipoles, we knew that they are, respectively, the right and left null space of previously computed matrix F.
We also knew that, given a generic mxn matrix S decomposed via SVD, the right null space of S is given by the columns of V corresponding to singular values equal to zero.
Similarly, the left null space of S is given by the columns of U transposed corresponding to singular values equal to zero. For this reason we decomposed F, selected the right columns of V and U transposed and displayed the results.

Results

0.6 The 8-points algorithm

From the 8-points algorithm we obtain the Fundamental matrix: In the "Version 1" we obtain:

$$F1 = \begin{bmatrix} -2.0804.. & -2.0242.. & 0.0050.. \\ 2.1436.. & 8.2236.. & -0.0018.. \\ -0.0034.. & 0.0023.. & -0.9999.. \end{bmatrix} \tag{8}$$

In the "Version 2" we obtain:

$$F2 = \begin{bmatrix} -4.2038.. & 1.7559.. & -2.9774.. \\ -1.1244.. & 7.2694.. & 0.0041.. \\ 2.8505.. & -0.0042.. & -0.0355.. \end{bmatrix} \tag{9}$$

0.7 Evaluation of the results

Results for the epipolar constraint with simple Eight Points Algorithm:
-1.4661, -1.2436, -1.2512, -1.4339, -0.9702, -1.1203, -0.9687, -1.0842, -1.0974, -0.9512, -1.0488, -1.3611, -1.3599, -1.5029

Results for the epipolar constraint with Eight Points Algorithm with normalization:
0.0094, -0.0004, -0.0000, 0.0050, 0.0031, 0.0097, -0.0048, 0.0072, 0.0030, 0.0010, 0.0092, 0.0043, 0.0021, 0.0099, 0.0069

As we can notice, the second version gives us the best result, because all the values are closer to the zero and this is what we want in order to satisfy the epipolar constraint:

$$(x')^T F x = 0$$

The estimated F is not invariant to point transformations, for this reason it is advisable to normalize points. As a matter of fact, from these results we notice that the 8-points algorithm with normalization is more robust than the other one.
We use the provided function "visualizeEpipolarLines" to visualize the stereo pairs with epipolar lines of the corresponding points and the results are shown in the Figure 1-2-3-4.

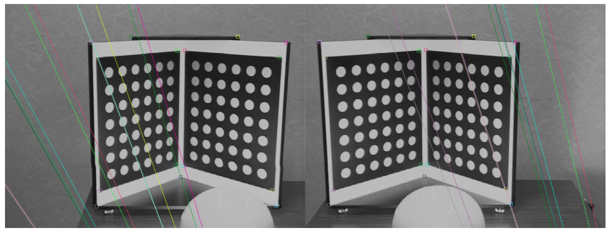

Figure 1: Eight points algorithm not normalized for Mire

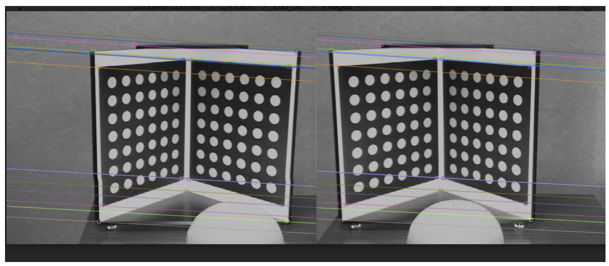

Figure 2: Eight points algorithm normalized for Mire

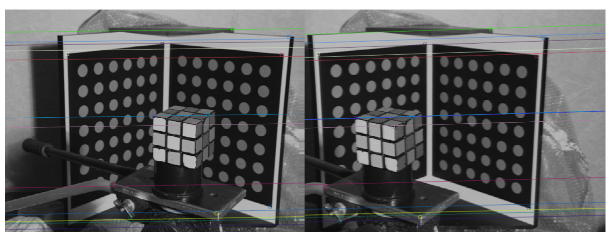

Figure 3: Eight point algorithm not normalize for Rubik

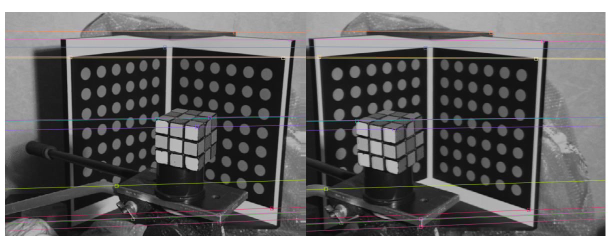

Figure 4: Eight point algorithm not normalize for Rubik

As last result we have:
The left and the right epipoles of the simple algorithm are:

$$[-0.4487; -0.8937; -0.0005] and [-0.4091; -0.9125; -0.0003].$$

The left and the right epipoles of the normalized algorithm are:

$$[-0.9977; -0.0672; -0.0000] and [0.9975; 0.0711; 0.0000]$$