Computer Vision: Third Assignment

Students Giacomo Lugano (5400573), Claudio Tomaiuolo (5630055)

Abstract

In this work, we implement the Eight-points algorithm in two ways: normalising the points and without do that, to estimate the fundamental matrix F. The fundamental matrix is a basic tool in the analysis of scenes taken with two uncalibrated cameras, and the eight-point algorithm is a frequently cited method for computing the fundamental matrix. It has the advantage of simplicity of implementation [1].

Introduction

The Eight-point Algorithm is an algorithm used in computer vision to estimate the essential matrix or the fundamental matrix related to a stereo camera pair from a set of corresponding image points [2]. It was introduced by Christopher Longuet-Higgins in 1981 for the case of the essential matrix [5]. In theory, this algorithm can be used also for the fundamental matrix, but in practice the normalized eight-point algorithm, described by Richard Hartley in 1997, is better suited for this case.
The algorithm's name derives, obviously, from the fact that it estimates the essential matrix or the fundamental matrix from a set of eight (or more) corresponding image points. However, variations of the algorithm can be used for fewer than eight points [1].

Eight-point Algorithm

In order to obtain 3D information from cameras, it's necessary to overcome the scene depth ambiguity. To do that a stereo system is required.
A stereo system consists of two cameras, relatively fixed and oriented in space in such a way they have a large common covered area. The relative orientation and position of the two cameras is summarized through a rotational matrix R and a position vector T (extrinsic parameters), expressing the rotational relation between the two camera frames and the relative position of the origins of such frames [6][7].

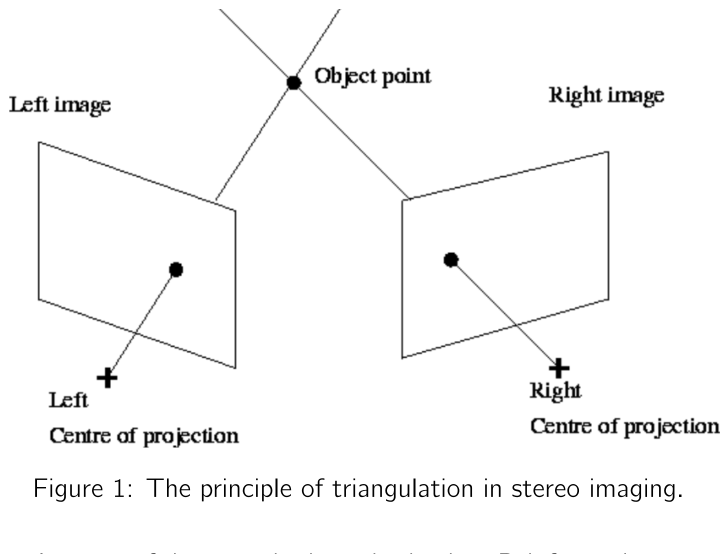

Figure 1: The principle of triangulation in stereo imaging.

Then, to fulfill the complete set of data required to obtain the 3D information, we need to consider also the intrinsic parameters that characterize the mapping of an image point x from camera to pixel coordinates in each of them, described inside two 3x3 triangular matrices (one for each camera) K_l, K_r.

Starting from that, we can consider how the same point in the 3D world is mapped in the two images I_l, I_r and which are the important relations between the two image points X_r, X_l.

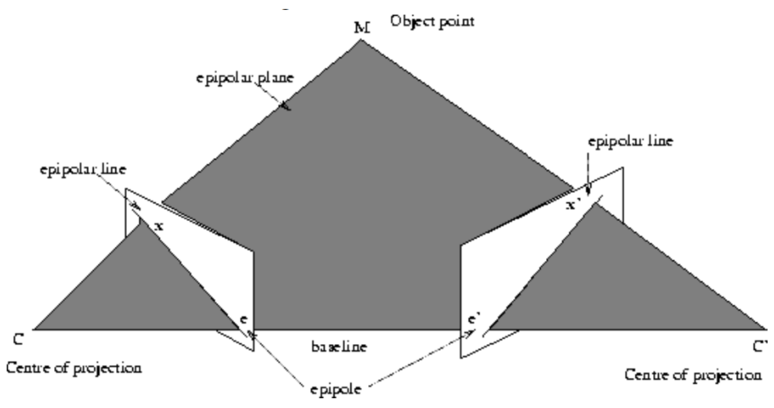

Figure 2: The epipolar constraint.

We can see how the point in one camera, for example x_r, is constrained to lies on the line l_r obtained by the intersection between the epipolar plane and the right image plane. The same can be said for the left point. So, knowing one of the two point, we can bound our research to the corresponding line, saving computational power and time by moving from a 2D to a 1D problem.

We can formalise these constraints by knowing that the line connecting the origins of both cameras and the 3D point X_l ($X_r = X_l - T$) and the line connecting these two origins T, are coplanar. This can be expressed as:

$$(X_l - T)^T T \times X_l = 0 \tag{1}$$

Now, we can express the cross product in the screw symmetric form as:

$$(X_l - T)^T S X_l = 0 \tag{2}$$

where

$$S = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix} \tag{3}$$

Then, we can move all the elements into the same reference frame, in this case the left camera:

$$\begin{gathered} (R^T X_r)^T S X_l = 0 \\ (X_r)^T R S X_l = 0 \end{gathered} \tag{4}$$

where $E = SR$ takes the name of essential matrix and relates the image point of the same 3D point in the two camera images expressed in millimeters $(x_r^{mm})^T E x_l^{mm} = 0$. To move the relation in the pixel space, we can multiply each point vector for the related triangular matrix of the intrinsic parameters, and we can do the same for the relation above:

$$x_l^{mm} = K_l^{-1} x_l \qquad x_r^{mm} = K_r^{-1} x_r \qquad x_r^T K_r^{-T} E K_l^{-1} x_l = 0 \qquad x_r^T F x_l = 0 \tag{5}$$

Obtaining the fundamental matrix $F = K_r^{-T} E K_l^{-1}$, which maps each pixel point in a camera image plane to the corresponding one in the other one, it allows us to obtain the epipolar lines in this way:

$$l_r = F x_l \qquad\qquad l_l = F^T x_r \tag{6}$$

In the end, the matrix F has seven degrees of freedom: three are introduced by the relative orientation of the two cameras; two from the relative position in space (considering also the scale ambiguity); two introduced by the intrinsic parameters of the two cameras.

This procedure works under the assumption that we already know all the intrinsic and extrinsic parameters. It's also possible to obtain it starting from the two images obtained by the stereo system with at least 8 points correspondence since the fundamental matrix has 7 degrees of freedom.

In this way, we can start from the defined relation above and the set of corresponding points:

$$x_r^T F x_l = 0 \qquad\qquad x_r = (x_i^r, y_i^r, 1)^T \qquad\qquad x_l = (x_i^l, y_i^l, 1)^T \tag{7}$$

We can rearrange them to obtain the relation:

$$\begin{bmatrix} x_1^r x_1^l & x_1^r y_1^l & x_1^r & y_1^r x_1^l & y_1^r y_1^l & y_1^r & x_1^l & y_1^l & 1 \\ x_2^r x_2^l & x_2^r y_2^l & x_2^r & y_2^r x_2^l & y_2^r y_2^l & y_2^r & x_2^l & y_2^l & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_n^r x_n^l & x_n^r y_n^l & x_n^r & y_n^r x_n^l & y_n^r y_n^l & y_n^r & x_n^l & y_n^l & 1 \end{bmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ \cdot \\ f_{33} \end{pmatrix} = 0 \tag{8}$$

where we can call A the matrix containing the points and we can see it as an over-determined homogeneous system, which may be solved with the DLT algorithm, by adding the constraint that the rank of F should be equal to 2.

To do that, we can start by computing the SVD decomposition of the matrix A:

$$A = UDV^T \tag{9}$$

and choosing as solution the last column of V, which is a vector of shape 9x1.

We can reshape it creating a matrix 3x3 as follows:

$$v = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9) \qquad\qquad F^1 = \begin{bmatrix} v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 \\ v_7 & v_8 & v_9 \end{bmatrix} \tag{10}$$

Then, we can enforce the rank of F^1, which is our first estimation of the fundamental matrix, to be equal to 2, by locking at the closer matrix which has this rank. To do that, we can again rely on the SVD decomposition applied on the matrix F^1:

$$F^1 = UDV^T \tag{11}$$

Then, we can pick the obtained matrix D of the Engen values of F^1 and set the smallest one, which appear in the position 3x3, equal to 0:

$$\text{From: } D = diag(\sigma_1, \sigma_2, \sigma_3) \qquad\qquad \text{to } D^1 = diag(\sigma_1, \sigma_2, 0) \tag{12}$$

and then recompute F as:

$$F = UD^1V^T \tag{13}$$

Obtaining our estimated F.

Eight-point Algorithm with normalization

The previously seen method to obtain the Fundamental Matrix leads to a fundamental matrix which is not invariant to point transformations. For this reason, it's advisable to normalize the matrix obtained above following these steps:

- translate points in such a way that the image coordinates center corresponds to the points centroid.
- scale points so that their average distance from the origin is $\sqrt{2}$.

These steps can be easily done by multiply the fundamental matrix by the transformation matrix T equal to:

$$T = \begin{bmatrix} s & 0 & -sc_x \\ 0 & s & -sc_y \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Results

First Implementation. Eight-point algorithm.
Starting from the given sets of points, is possible to compute the matrix A as shown above. Then, it can be decomposed using the SVD MATLAB function from which we can extract the last column of the matrix V, which will be reshaped by the MATLAB function reshape() (the obtained vector needs to be transposed in order to obtain our desired F^1). After that, we can perform again the SVD decomposition using the same MATLAB function as before, extract the matrix D and set the 3x3 term equal to 0 and finally recompute F as a matrix multiplication.

Having done so, it is possible to load the given pictures with the relative sets of point and evaluate the epipolar lines with the function visualizeEpipolarlines(), obtaining:

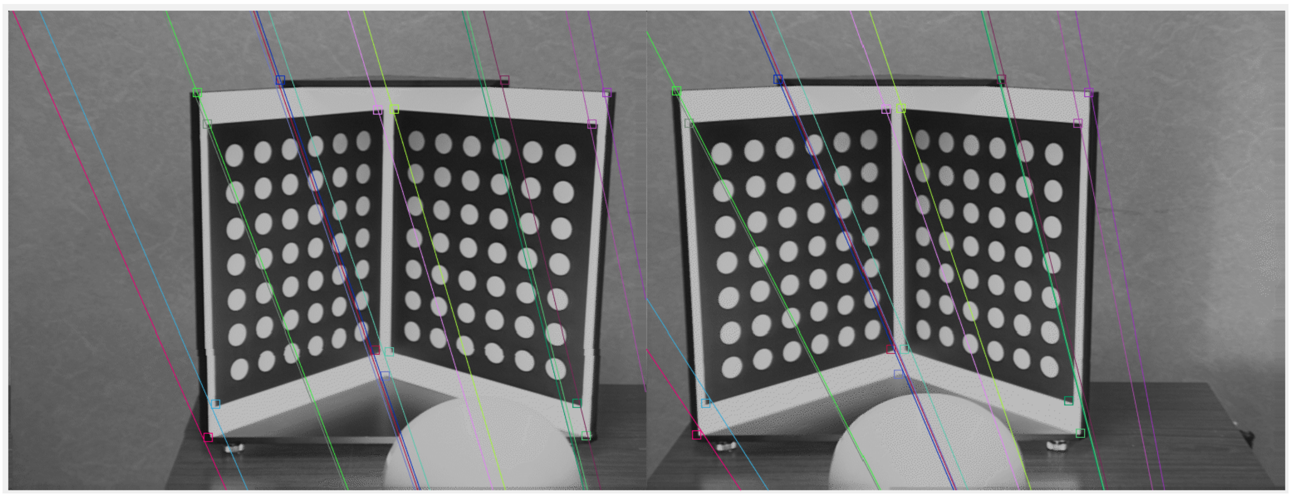
Figure 3: Mire result without normalization.

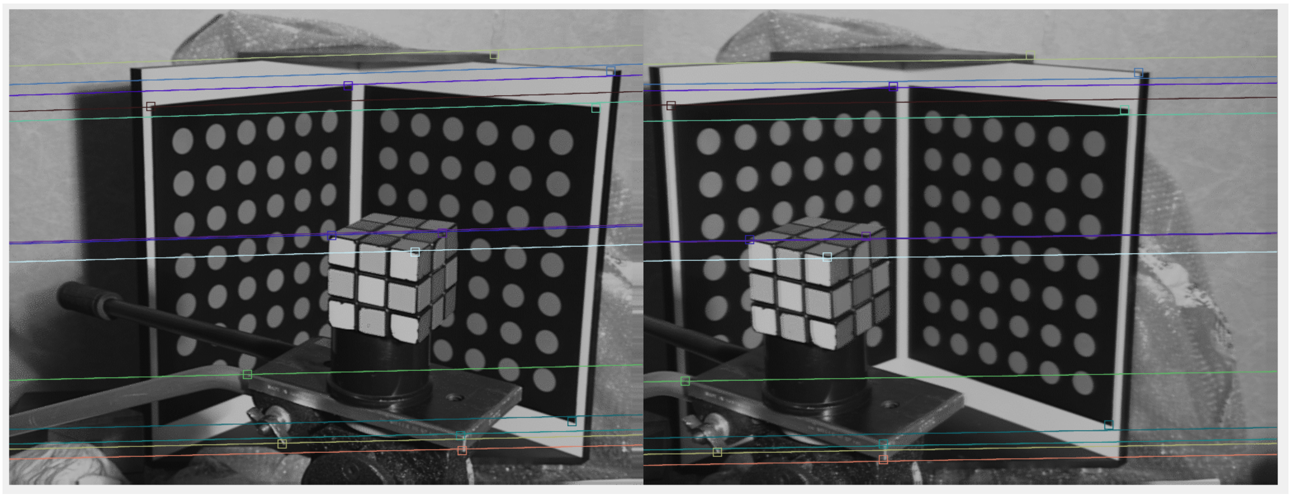
Figure 4: Rubik result without normalization.

As we can see in both cases, the epipole lines pass almost through the corresponding point in both the images of the stereo system, we can also observe how the slope of the epipole lines change between the right and left picture.

We can address these errors in part to the computational process and in part to MATLAB itself, which during the evaluation can introduce a certain error. We can have a deeper understanding of that by looking at the verification of the epipolar constraint $x_r^T F x_l = 0$ and the relative produced error for both the cases:

Mire epipolar constraint error = 17.9813 px

Rubik epipolar constraint error = 33.3598 px

We can note also that the slope of the Mire picture epipolar lines are bigger than the Rubik image case, but this will be discussed at the end of the second implementation.

Second Implementation. Eight-point algorithm with normalization.
Like in the theory, even here we can keep the defined algorithm above, processing before the normalized points thanks to the function normalise2dpts() and using them as input for the algorithm. Then we need to de-normalize the obtained matrix F multiplying it for the two transformation matrix given as output from the normalise2dpts() function:

$$F = T_2^T F' T_1 \tag{15}$$

Having done so, it is possible to load the given pictures with the relative sets of point and evaluate the epipolar lines with the visualizeEpipolarlines() function, obtaining:

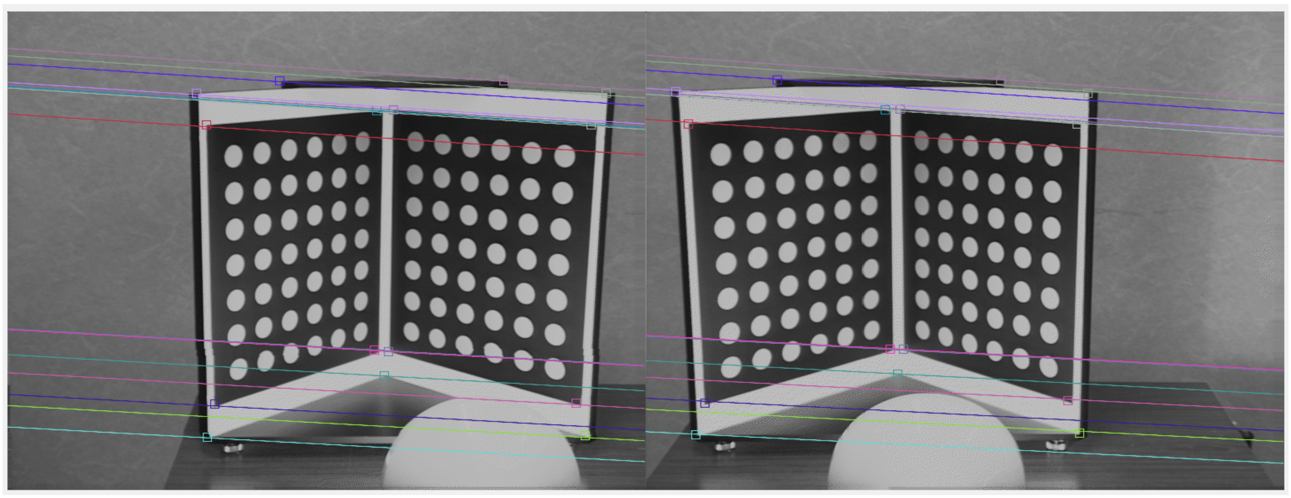

Figure 5: Mire result with normalization.

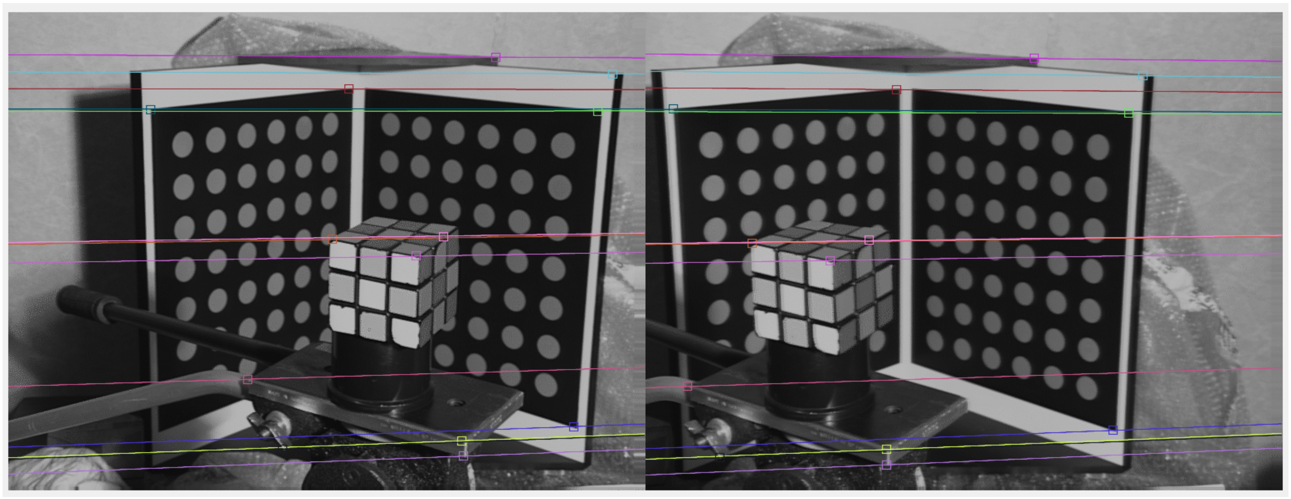

Figure 6: Rubik result with normalization.

Also in this case we've obtained the desired epipolar lines. As we can see, they pass in a more precise way through the relative points and that is also denotated by the epipolar constraint error.

Right epipolar constraint error = 0.4741 px

Left epipolar constraint error = 0.3285 px

We can also observe that the epipolar lines in the Mire picture are more horizontal respect the results obtained in the first version of the eight-point algorithm. This is due to the point transformation between the left and right picture and the invariance of the new obtained fundamental matrix F.

Conclusions

After evaluating the results obtained from the tests, it is possible to deduce that the normalized eight-point algorithm is much more efficient than the one without normalization in terms of accuracy and precision.

Furthermore, with coordinate normalization to improve the condition of the noise problem, the normalized eight-point algorithm performs almost quite well.

References

[1] R. I. Hartley, *In defense of the eight-point algorithm*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, pp. 580-593, June 1997, doi: 10.1109/34.601246.

[2] H. C. Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, Nature, vol. 293, no. 5828, pp. 133–135, 1981.

[3] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*, MIT Press, 1993.

[4] 4] B. D. Lucas, T. Kanade, et al., *An iterative image registration technique with an application to stereo vision*.

[5] H.C. Longuet-Higgins, (Sept 1981). A computer algorithm for reconstructing a scene from two projections. Nature, 293:133–135.

[6] O. Faugeras, Q.-T. Luong, and T. Papadopoulou, (2001). The Geometry of Multiple Images. MIT Press.

[7] R. Hartley and A. Zisserman., (2000). Multiple View Geometry in Computer Vision. Cambridge University Press.