



# UNIVERSITÀ DEGLI STUDI DI GENOVA

## DIBRIS

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY,  
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

## COMPUTER VISION

---

### Third Assignment

---

*Authors:*

Marmolino Giorgio s7721100  
Penna Lorenzo s4953984  
Petrosilli Marco s7812048

# 1 Discussion on the Eight-Point Algorithm Implementations

The code implements two versions of the Eight-Point Algorithm to compute the fundamental matrix  $F$ . These versions differ in whether they normalize the input point coordinates.

## 1.1 Basic Eight-Point Algorithm

The function `EightPointsAlgorithm` directly computes  $F$  using the input points by solving a system of equations derived from the epipolar constraint:

$$\mathbf{x}_2^\top F \mathbf{x}_1 = 0$$

where  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are corresponding points in the two images. To compute the solution for the fundamental matrix  $F$ , the following steps are performed:

1. Singular Value Decomposition of  $A$ :

The matrix  $A$ :

$$A = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2 x_2 & x'_2 y_2 & x'_2 & y'_2 x_2 & y'_2 y_2 & y'_2 & x_2 & y_2 & 1 \\ x'_3 x_3 & x'_3 y_3 & x'_3 & y'_3 x_3 & y'_3 y_3 & y'_3 & x_3 & y_3 & 1 \\ \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix}$$

is decomposed using Singular Value Decomposition (SVD):

$$[U, D, V] = \text{svd}(A)$$

where:

- $U$  and  $V$  are orthogonal matrices,
  - $D$  is a diagonal matrix containing the singular values of  $A$  in descending order.
2. The solution vector  $f$  is selected as the last column of  $V$ , which corresponds to the smallest singular value and reshaped into a  $3 \times 3$  matrix
  3. Since the fundamental matrix  $F$  must have rank 2, the rank is enforced as follows:

Compute the SVD of  $F$ :

$$[U, D, V] = \text{svd}(F)$$

Set the smallest singular value in the diagonal matrix  $D$  to zero:

$$D(3, 3) = 0$$

4. Recompute the Final Matrix: The final fundamental matrix  $F$  is recomputed by reconstructing it from the modified singular value decomposition:

$$F = U \cdot D \cdot V^T$$

This process ensures that the matrix  $F$  satisfies the fundamental matrix constraints, including having a rank of 2.

If the coordinates of  $P1$  and  $P2$  are large or unevenly distributed, the resulting matrix  $A$  can become inaccurate, leading to errors in  $F$ .

For the Rubik image the calculated  $F$  :

$$F = \begin{bmatrix} 7.3500 \times 10^{-8} & 1.2635 \times 10^{-6} & 0.0013 \\ 2.5270 \times 10^{-6} & -3.5624 \times 10^{-6} & 0.1180 \\ -0.0041 & -0.1190 & 0.9858 \end{bmatrix}$$

## 1.2 Normalized Eight-Point Algorithm

The function `EightPointsAlgorithmN` addresses the stability issues of the basic version by normalizing the input points before solving for  $F$ . The normalization ensures:

- The centroid of the points is shifted to the origin.

- The points are scaled such that their average distance from the origin is  $\sqrt{2}$ .

This process generates a more robust  $F$ , especially when the point coordinates vary significantly in magnitude or are not centered.

The resulting  $F$  for the Rubik image:

$$F = \begin{bmatrix} -2.0748 \times 10^{-9} & -4.7014 \times 10^{-7} & 7.4580 \times 10^{-5} \\ 3.8425 \times 10^{-7} & -2.9192 \times 10^{-8} & -0.0044 \\ -4.1294 \times 10^{-5} & 0.0044 & -0.0071 \end{bmatrix}$$

### 1.3 Impact of Normalization

If the same set of correspondences is used as input to both versions of the algorithm, the following differences arise:

- The basic algorithm's result depends heavily on the scale and distribution of the points. If the points are unbalanced (e.g., widely scattered in one direction or have large coordinate values),  $F$  may be inaccurate.
- The normalized version produces more consistent and reliable results, even with poorly distributed points. The scaling step ensures that the points contribute equally to the solution, avoiding dominance by points with larger coordinate values.
- For well-distributed points, the results from both versions may be similar. However, for extreme cases, normalization can significantly improve accuracy.

### 1.4 Visualization of Epipolar Lines and Epipoles

To evaluate the fundamental matrix obtained, a visual method was first used, exploiting the function provided to visualize the epipolar lines, given two images, the two sets of corresponding points and the foundational matrix we want to use. This allows us to check how the epipolar lines are made at corresponding points, between the two images, to intuitively and quickly see if the calculated  $F$  works as expected.

We report the results obtained:

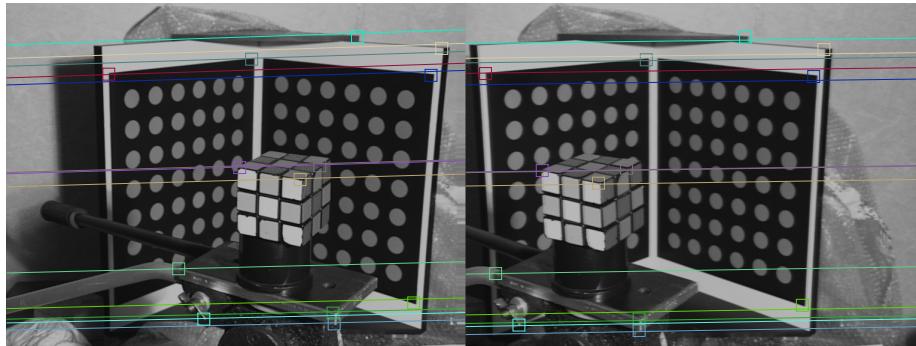


Figure 1: Epipolar lines of the basic  $F$

And the left and right epipoles, evaluated as: the right and left null space of  $F$  and implemented with the SVD decomposition of  $F$ ,

$$F = UWV^T$$

and then selecting the last columns of  $U$  and  $V$ .

The epipoles of the basic  $F$  are:

$$e_{\text{left}} = \begin{bmatrix} 91312.1794 \\ -1031.5242 \\ 1 \end{bmatrix} \quad e_{\text{right}} = \begin{bmatrix} -44520.6536 \\ 1543.3180 \\ 1 \end{bmatrix}$$

The epipoles for the normalized  $F$ :

$$e_{\text{left}} = \begin{bmatrix} 9420.219 \\ 158.332 \\ 1 \end{bmatrix} \quad e_{\text{right}} = \begin{bmatrix} 11438.681 \\ 108.153 \\ 1 \end{bmatrix}$$

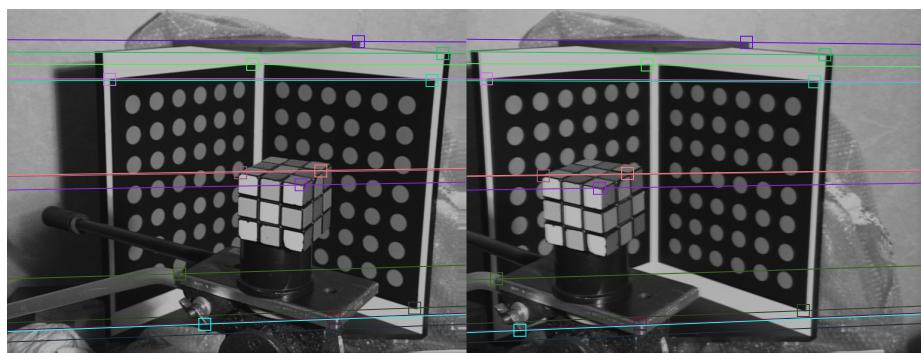


Figure 2: Epipolar lines of the normalized F

As we could expect, the two epipols are very far from appearing in the image.

The same has been applied to the Mire images, here the epipolar lines representation:

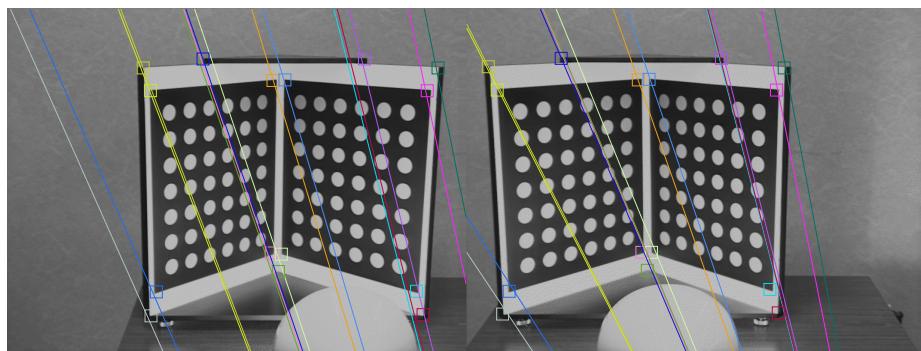


Figure 3: Epipolar lines of the basic F

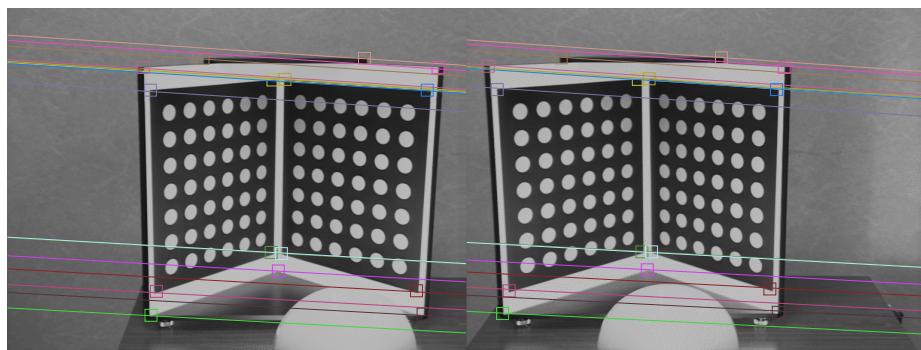


Figure 4: Epipolar lines of the normalized F

## 2 Acquisition and matching of a pair of stereo images

### 2.1 Images acquisition

In order to perform the acquisition and matching of a pair of stereo images, two pictures of the same objects have been acquired using a phone with a resolution of  $900 \times 1600$  pixels; this was done by capturing an image of the same object first from the left and then moving slightly to the right.



Figure 5: The acquired stereo images

### 2.2 Features detection

A SURF (Speeded-Up Robust Features) descriptor was used instead of SIFT (Scale-Invariant Feature Transform) primarily due to its higher computational efficiency and robustness. Furthermore, SURF provides superior robustness against variations in scale, rotation, noise, and changes in illumination or contrast, which are essential features for ensuring reliable performance in complex scenarios or with images captured under uncontrolled conditions. After using the image matching function, X correspondences were found between the provided images.

### 2.3 Estimation of the Fundamental Matrix F with RANSAC

The RANSAC function used accepts two lists of corresponding points between two images and a threshold value as parameters. The threshold defines the maximum tolerance for considering a point as an inlier, meaning it is consistent with the estimated geometric model. The function returns the model that maximizes the number of inliers, ensuring a robust estimate even in the presence of outliers among the correspondences. X inliers and Y outliers have been found using the function.

### 2.4 Threshold choice for SURF and RANSAC algorithms

In the research for common features between the two images a threshold between 0.5 and 0.6 was used on the SURF, this choice has been made 'cause (assuming that the pairs of points are "correct enough") working with a larger dataset of points then allows RANSAC to better estimate the desired F for following reasons:

- RANSAC is designed to perform well in the presence of outliers (noisy or bad points). However, if the total number of points is too low, even a small number of outliers can significantly influence the result. With more points, there is a higher probability that the randomly selected subset for each iteration contains only inliers, improving model estimation.
- In many applications, match points may be unevenly distributed across the image. A large point list increases the chances of including matches that cover the entire scene, improving the accuracy of model estimation.
- Once a model is estimated in one iteration of RANSAC, it is verified against the entire dataset. Having a large number of points allows you to better evaluate the model and separate inliers and outliers more precisely.

On the other hand for the RANSAC a threshold equal to  $10^{-4}$  has been used, this threshold should be very small since is used after the fundamental relationship is calculated from a specific function on 8 pairs of points

already taken to estimate an F, subsequently the values obtained (which ideally should be close to 0) are added, and checking that this sum is less than the threshold the algorithm proceeds with the identification of inliers and outliers.

## 2.5 Matching result

Generally, we get matches like the one pictured below:

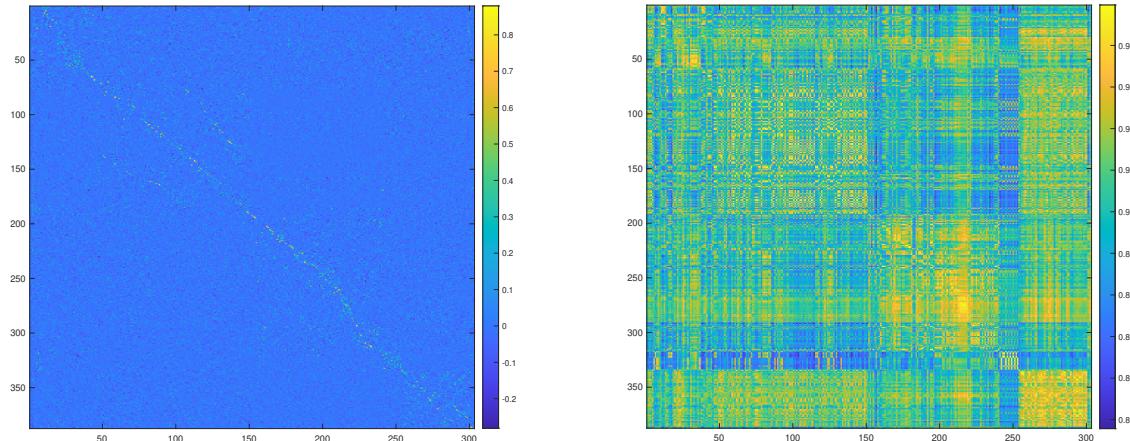


Figure 6: Match map between pixels of both images (left figure) and enhanced match map (right figure).

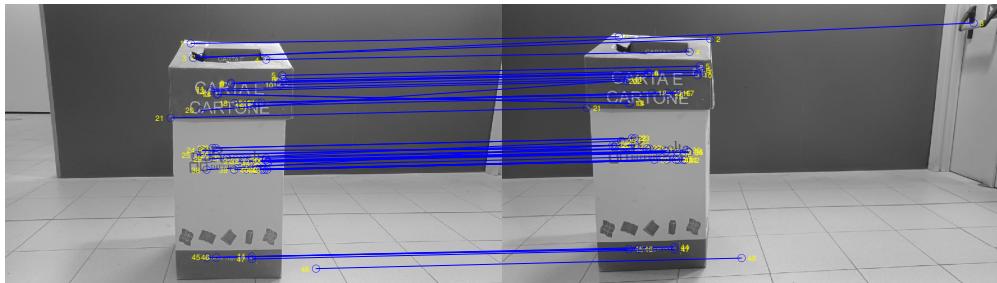


Figure 7: Match between two images

## 2.6 Evaluation of foundamental matrix and (FORSE) EPIPOLES

Evaluating F matrices obtained performing the RANSAC algorithm we've noticed that sometimes happen that if this procedure takes points very close to each other to determine the bestF, when you print the epipolar lines what happens is that you have degenerate configurations with the epipolar poles printed inside the image rather than outside as they should be:

The matrix F used is as follows:

$$\mathbf{F} = \begin{bmatrix} -4.0 \times 10^{-7} & -9.1 \times 10^{-6} & 1.0863 \times 10^{-3} \\ 8.9 \times 10^{-6} & -2.0 \times 10^{-7} & -3.2503 \times 10^{-3} \\ -8.502 \times 10^{-4} & 5.6941 \times 10^{-3} & -2.7413 \times 10^{-1} \end{bmatrix}$$

The epipols computed using just shown F are:

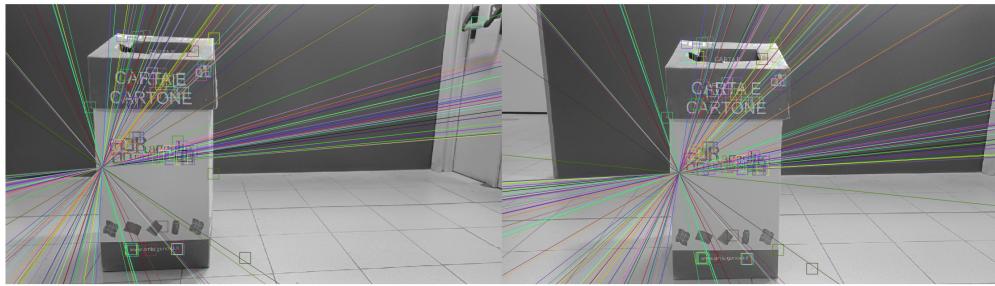


Figure 8: An example of epipolar lines passing through degenerate epipolar poles

$$e_{\text{left}} = \begin{bmatrix} 623.01359 \\ 123.88362 \\ 1.0 \end{bmatrix} \quad e_{\text{right}} = \begin{bmatrix} 367.10107 \\ 102.95326 \\ 1.0 \end{bmatrix}$$

Instead of this, other times the RANSAC builds an F matrix taking couples of points in such a way that they cover a bigger portion of the surface of the image; in those cases the F performs better results when we visualize epipolar lines, as show below:

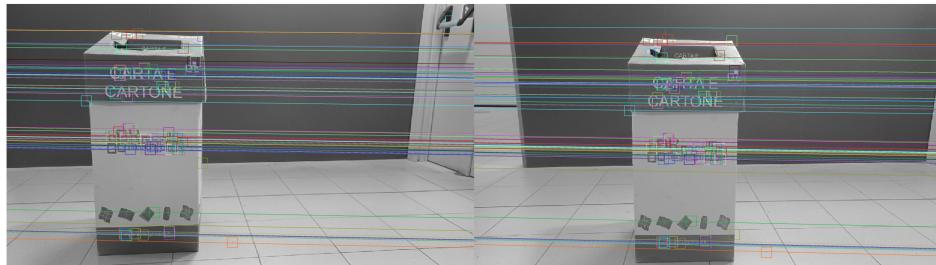


Figure 9: An example of epipolar lines using a good F obtained

The matrix F used is as follows:

$$\mathbf{F} = \begin{bmatrix} 0.0 & 9.0 \times 10^{-7} & -3.515 \times 10^{-4} \\ -7.0 \times 10^{-7} & -2.0 \times 10^{-7} & 6.5559 \times 10^{-3} \\ 2.175 \times 10^{-4} & -6.3348 \times 10^{-3} & 1.713381 \times 10^{-1} \end{bmatrix}$$

And the computed epipols are:

$$e_{\text{left}} = \begin{bmatrix} 6822.6619 \\ 339.7056 \\ 1.0 \end{bmatrix} \quad e_{\text{right}} = \begin{bmatrix} 8890.3360 \\ 332.3444 \\ 1.0 \end{bmatrix}$$

## 2.7 Manual matching to check RANSAC issue

We tried to get around this ransac problem by passing to the algorithm a list of matches taken by hand so as to be rather distributed across the entire surface of the image, as shown below.

With the couples shown, we obtain almost correct epipolar lines as reported on the following image. Obviously the presence of some cases of degenerate epipolar poles still persist, since as already said everything then depends on which points RANSAC chooses to compute de F matrix, but in this cases more times couple of points are taken in a way such that F fit quite good with all test couples of points.

Computed F:

$$\mathbf{F} = \begin{bmatrix} 1.7843 \cdot 10^{-8} & -4.2632 \cdot 10^{-7} & -4.1523 \cdot 10^{-4} \\ 3.2815 \cdot 10^{-7} & 5.9450 \cdot 10^{-8} & 2.0421 \cdot 10^{-3} \\ 4.1123 \cdot 10^{-4} & -1.9696 \cdot 10^{-3} & 5.4783 \cdot 10^{-2} \end{bmatrix}$$

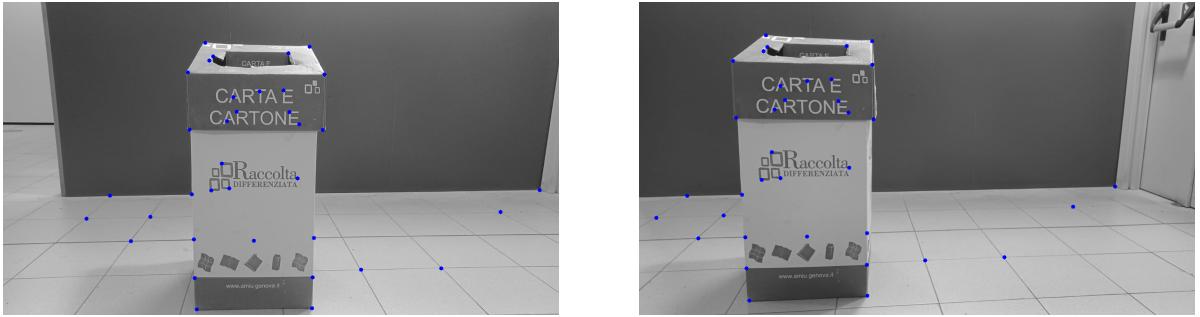


Figure 10: Points selected manually from the first image (right figure) and from the second image (left figure).

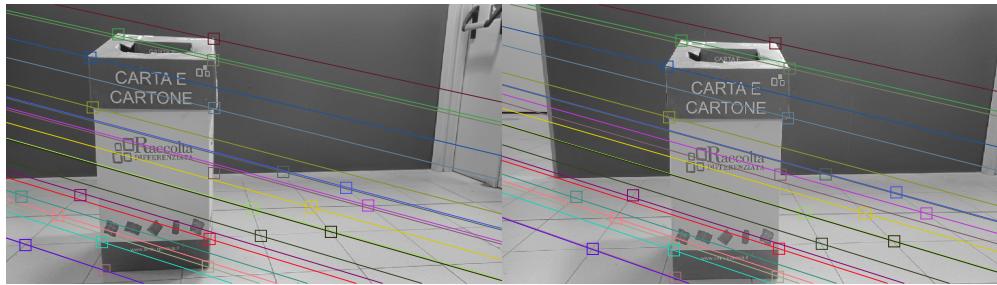


Figure 11: Epipolar lines obtained starting from manually matched points

Epipolar poles coordinates:

$$e_{\text{left}} = \begin{bmatrix} -4758.61735 \\ -994.435245 \\ 1.0 \end{bmatrix} \quad e_{\text{right}} = \begin{bmatrix} -6001.1683499 \\ -1225.1597192 \\ 1.0 \end{bmatrix}$$

## 2.8 Function check\_F()

Following the requests, we also implemented a function that verified the correctness of  $F$  on all the points provided between the matches. The verification is carried out by taking each pair of points and calculating the fundamental equation using the coordinates of the points as input. Obviously for numerical reasons there are no clean zeros among the results, but these are fully acceptable. We've obtained values of  $10^{-5}$  for the best points, while the values calculated with noisy (or at least not entirely precise) matches give values of  $10^{-2}, 10^{-1}$  in worst cases.

