Module 4 - Report week 3

Hicham El Muhandiz[1[0000−1111−2222−3333]], Julia Ariadna Blanco Arnaus[1[0000−1111−2222−3333]], Marcos Muñoz González[1[0000−1111−2222−3333]], Rosana Valero Martínez[1[0000−1111−2222−3333]]

Universitat Autònoma de Barcelona, Campus UAB, Edifici O, 08193 Cerdanyola del Vallès, Barcelona

1 Introduction

This project aims to develop a robust method for estimating the fundamental matrix, which relates two different views of the same scene. The core component of the method is the normalized 8-point algorithm, which is used to estimate the fundamental matrix. Additionally, the RANSAC (Random Sample Consensus) algorithm is employed to improve the robustness of the method. The inliers are obtained using a threshold on the first order approximation of the geometric error, known as the Sampson distance. Furthermore, the project computes also the epipolar lines of the matching points in both images and applies the theoretical concepts to do photo-sequencing.

2 Implementation

2.1 Estimation of the fundamental matrix

8-point normalized algorithm The 8-point algorithm is a method for estimating the fundamental matrix, which relates two images captured by cameras with corresponding camera matrices $P = [I|0]$, and $P = [R|t]$.

The first step of the algorithm is to normalize the points correspondences, which refers to the process of transforming the input points such that the new origin is the centroid of points and the mean distance between the origin and the data points is 2 pixels. To do so, we must do the following:

Find scale and translation values The translation values to center the set of points will be the ones that cancel out the mean of each axis ($-mean\_points_x, -mean\_points_y$). The scaling value **s** can be obtained by dividing by the average distance and multiplying by $\sqrt{2}$ as it can be seen in Equation 1.

$$s = \frac{(\sqrt{2}n)}{\sum_{i=1}^{n} \sqrt{(x_i - mean\_points_x)^2 + (y_i - mean\_points_y)^2}} \tag{1}$$

Generate normalizing matrix T Once found the amount of translation and scaling needed to normalize the points, they can be transformed by using a similarity transformation that has s and $-mean\_points_x, -mean\_points_y$ as scaling and translation factors respectively (Equation 2).

$$T = s \begin{pmatrix} 1 & 0 & -mean\_points_x \\ 0 & 1 & -mean\_points_y \\ 0 & 0 & 1/s \end{pmatrix} \tag{2}$$

After implementing these two steps, the normalizing matrix for a pair of correspondences, p_i and p'_i, can be generated. This matrix can be used to transform the correspondences, allowing the application of the 8-point normalized algorithm.

The next step of the algorithm is to create a matrix W from pi and $p'i$ correspondences, by taking in account that for each point correspondence $pi = [uv]^T$ and $p'i = [u'v']^T$ the fundamental matrix constrains their relation to:

$$p'^T F\ p = 0 \rightarrow \begin{pmatrix} u' & v' & 1 \end{pmatrix} \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \tag{3}$$

Which translates to the fact that the following constraint must be verified:

$$\begin{pmatrix} uu' & vu' & u' & uv' & vv' & v' & u & v & 1 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0 \tag{4}$$

We can solve this system of equations by taking at least 8 corresponding points and using the singular value decomposition, resulting on $W = UDV^T$, where U and V are orthogonal matrices and D is a diagonal matrix with non-negative entries in descending order. The last column of V contains the least-squares solution to the linear system of equations represented by matrix W. From the last column of V it can be composed the fundamental matrix F by reshaping this column into a 3x3 matrix. However, this fundamental matrix F is of rank 3, meaning that the epipolar lines will not coincide into the epipole. For this reason F must be forced to be of rank 2.

To enforce a rank-2 constraint, the SVD is applied to the fundamental matrix F of rank 3, and the last singular value of D is removed to create $\tilde{D}$.

Finally, the algorithm re-computes the matrix $F = U\tilde{D}V^T$ and denormalizes it. This final matrix is the rank-2 approximation of the fundamental matrix, which relates the two images with corresponding camera matrices.

To test if the implementation works correctly, we compute a ground truth F_{gt} from two camera matrices $P = [I|0]$ and $P' = [R|t]$. If we know that we can obtain K from $P = K[R|t]$, and that $F = K'^{-T}[T'_x]RK^{-1}$, we can deduce that $K = I$ and $F = [T'_x]R$, which can be used to construct F_{gt}. We use 8 points to estimate F_{es} and apply the 8-point normalized algorithm, and we compare it to F_{gt}. We observe that the implementation is correct and the algorithm is capable of estimating the fundamental matrix as the difference between them is 1.0186658322130744e-13.

$$F_{gt} = \begin{pmatrix} 0.0978244 & 0.36508565 & -0.18898224 \\ -0.36508565 & 0.0978244 & 0.56694671 \\ 0.03580622 & -0.59654067 & 0 \end{pmatrix} F_{es} = \begin{pmatrix} 0.0978244 & 0.36508565 & -0.18898224 \\ -0.36508565 & 0.0978244 & 0.56694671 \\ 0.03580622 & -0.59654067 & 0 \end{pmatrix} \tag{5}$$

2.2 Robust 8-point normalize algorithm

Similarly to the previous assignment, we can find a way to obtain a more accurate estimation of the fundamental matrix F by using an iterative method, RANSAC. The objective is to keep the F estimate with the lowest error.

In each iteration, we sample 8 points of randomly picked correspondences between our two images. This means that in each iteration we will compute a different fundamental matrix. After obtaining the matrix using the procedure described in section 2.1, we have to quantify how much error it has. This is done by counting the amount of inliers using this F over the entire set of points. The criteria to determine whether or not a correspondence is an inlier is done using the Sampson distance (Equation 6), which is the first order approximation of the geometric error. When implementing it, we have to ensure that both points x_i and x'_i are in euclidean coordinates, since they were in homogeneous coordinates when we used them to compute the fundamental matrix. If this error is below a user-defined threshold, this set of points is stored and considered an inlier.

$$sampson\_distance = \frac{(x_i^T F x_i)^2}{(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T x'_i)_1^2 + (F^t x_i)_2^2} \tag{6}$$

After finishing the iterations, we recompute F with the set of points whose F had more inliers. An example can be seen in Figures 1 and 2. As we can see in Figure 2, some correspondences that were not of interest around the van have been discarded.

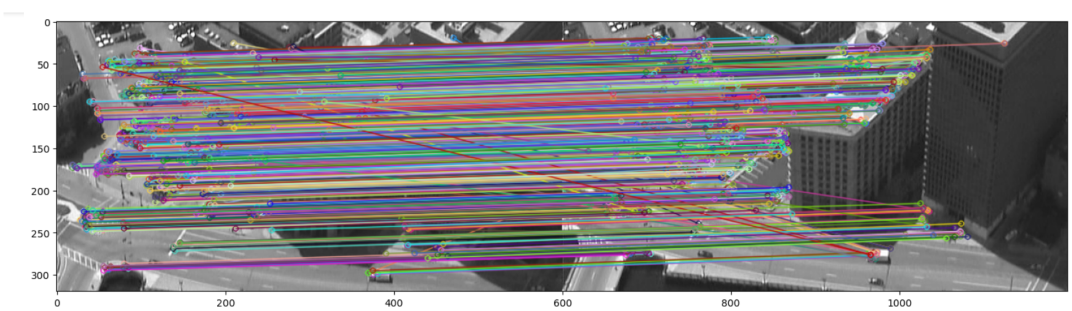

Fig. 1: Detected correspondences in images frame_00000.tif and frame_00001.tif

This reduces the effect of outliers, as they are filtered out if they have more error than the threshold. We have to keep in mind that this does not necessarily mean that it's the optimal solution of the problem, as we are not checking all of the possible 8 point combinations to obtain F.

2.3 Epipolar lines

In this step we compute the epipolar lines of the matching points between two images. To do this, we first obtain through the computed keypoints of both images the two set of homogenous points x on image 1 and their corresponding homogenous points x' on image 2. Then, with the previously

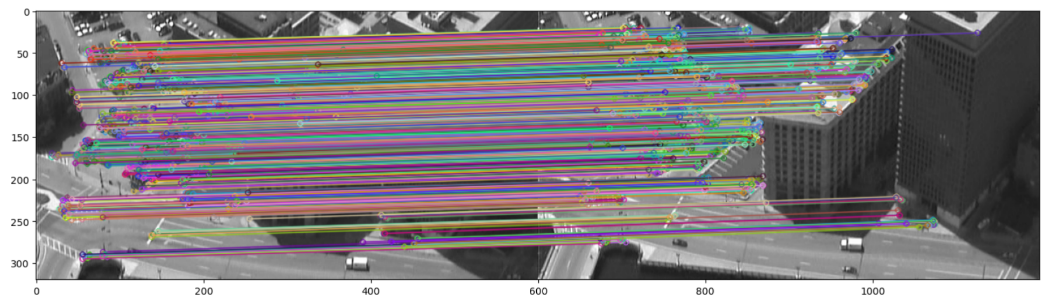

Fig. 2: Best set of correspondences in images frame_00000.tif and frame_00001.tif after running RANSAC

obtained fundamental matrix and each corresponding set of points we compute the epipolar lines l1 and l2 with Equations 7 and 8. Observe that to obtain l' we must use the transpose of the fundamental matrix of Equation 8 instead of the fundamental matrix per se. The resulting epipolar lines for three random points in x with their corresponding points in x' can be seen in Fig 3 and 4. Note that we can visually inspect that in each corresponding epipolar line of a point x_i on x, the corresponding point x_i' can be found. This comes from the definition of an epipolar line.

$$l = Fx \tag{7}$$

$$l' = F^T x \tag{8}$$

2.4 Application: Photo-sequencing

One of the applications of the estimation of the fundamental matrix is the Photo-sequencing, which is a useful technique for evaluating (or visualizing) a dynamic picture captured by still photographs.

In this part, we will compute a simplified version of the algorithm explained in the Photo-sequencing paper [2]. The point of the method is that given a set of images of a dynamic scene taken at different viewpoints and different time instants, it establishes an ordering of the images according to the time they were taken. To do so, two underlying hypotheses need to be considered:

- Object trajectories can be approximated by straight lines.
- Two of the images are taken from approximately the same position.

The main idea of the general algorithm is to use the static feature points to compute the fundamental matrix between every image and the reference image and map all dynamic features to their epipolar lines in the reference image.

In our implementation, to compute the fundamental matrix between the images, we use the Robust 8-point normalize algorithm explained in the previous section. We do this process twice in order to compute the fundamental matrix between image 1 and 2, and between image 1 and 3.

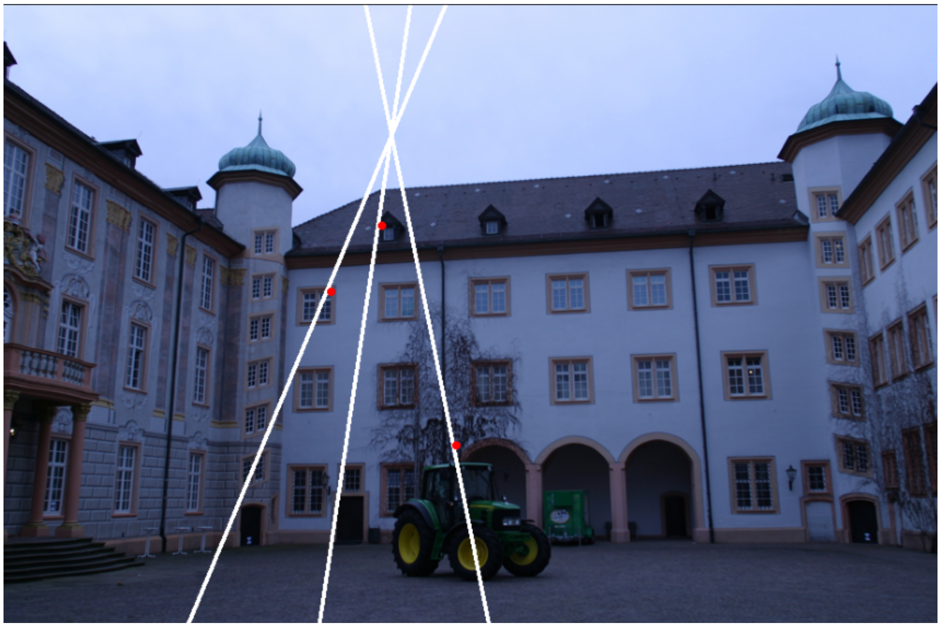

Fig. 3: A set of three epipolar lines of l1 corresponding to the transpose fundamental matrix multiplication with three random points in x' of image 2

Fig. 4: A set of three epipolar lines of l2 corresponding to the fundamental matrix multiplication with three random points in x of image 1

In our case, since we do not have two images taken from roughly the same viewpoint at two different time instants (second hypothesis), we manually select a dynamic point corresponding to a point in a van. To find this point, we iterate through the matching points between image 1 and image 2 (that we have stored previously computing the fundamental matrix) using the coordinates of the van in the first image.

Once we have done this, we can obtain the indices of the keypoints found previously. Then, using these indices, we can extract the x and y coordinates of two points of the trajectory projected in image 1. Next, we obtain the trajectory by doing the cross product between these two points. After that, we compute the epipolar lines in the second and third images by taking the transpose of the fundamental matrix and multiplying it with the coordinates of the matched keypoints.

Finally, we project the points of the trajectory onto the epipolar lines using the cross product between the trajectory and each epipolar line, as shown in Figure 5.

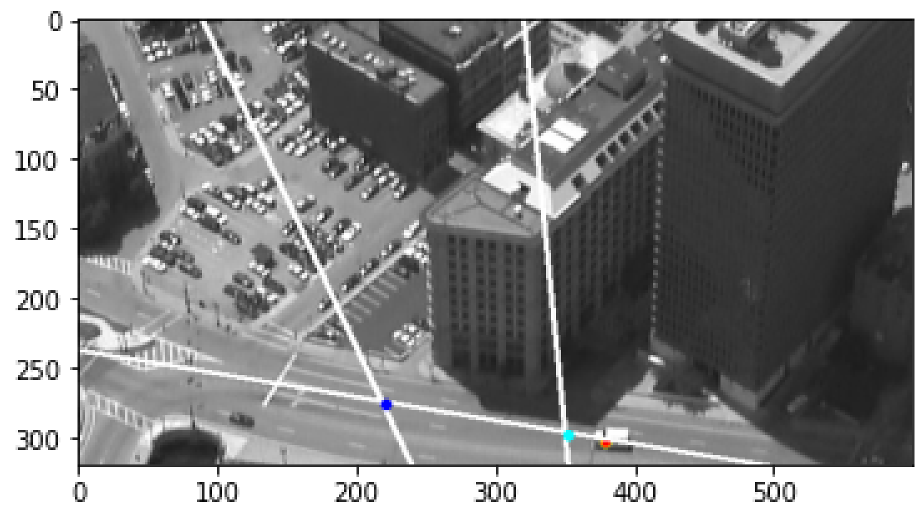

Fig. 5: Projection of the 3D van trajectory. Red, cyan and blue points represent the point of the van selected manually, the point of the trajectory onto the first epipolar line and the point of the trajectory onto the second epipolar line, respectively.

3 Conclusions

In this work, we have implemented and evaluated several techniques for the fundamental matrix estimation. The 8-point algorithm and its variants have been tested and some insight into the implementation and results of the algorithms and applications are shown. It is worth mentioning that the current implementation for photo-sequencing is not completely automatic, as the correspondence of the object of interest has to be selected manually.

Some problems that appeared during the implementation of the exposed techniques are:

- One of the main issues we have faced was selecting the pair of correspondences that referred to the van in image frame_00000.tif, as the default value selected a static correspondence.
- Due to the random nature of RANSAC, some matching points may be removed. This could affect the performance of other algorithms, such as the photo-sequencing method.

Overall this has been an interesting project, where we have seen both the inner implementations for fundamental matrix estimation and their results. As a relevant bibliography, we have used the book Multiple View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman [1] and the paper Photo Sequencing by Tali Dekel, Yael Moses, and Shai Avidan [2].

References

1. Hartley, R., Zisserman, A. (2004). Multiple View Geometry in Computer Vision (2nd ed.). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511811685
2. T. Basha, Y. Moses, and S. Avidan. Photo Sequencing, International Journal of Computer Vision, 110(3), 2014.