3D recovery of urban scenes: Fundamental matrix estimation and applications

Josep Brugués i Pujolràs[1], Sergi García Sarroca[1], Òscar Lorente Corominas[1], and Ian Pau Riera Smolinska[1]

Universitat Autònoma de Barcelona, Bellaterra, 08193, Catalonia, Spain
{josep.brugues, sergi.garciasa, oscar.lorentec, ianpau.riera}@e-campus.uab.cat

1 Introduction

In this project, the fundamental matrix (F) relating two different views of the same scene is estimated from a minimum of eight keypoint correspondences between them. F is estimated both with the normalized 8-point algorithm (algebraic method) and the robust normalized 8-point algorithm using RANSAC. After obtaining F, we can use some sample points and compute their corresponding epipolar lines to verify the correct estimation of F. Finally, we will test these algorithms in a real application, which is a simplification of the Photo-sequencing method by T. Dekel (Basha) et al [1].

In this document, we present the methodology that was followed for each section, as well as the results obtained and the problems that have arisen.

2 Fundamental matrix estimation

The first task is to create the function that estimates F given a set of point correspondences between a pair of images.

Algorithm 1 Normalized 8-point algorithm

Input: *points*1, *points*2,
Output: F

1. Normalize points.
2. Create matrix W from p_i and p'_i correspondences.
3. Compute the SVD of matrix W=UDV^T.
4. Create vector f from last column of V.
5. Compose fundamental matrix F_rank3.
6. Compute the SVD of fundamental matrix F_rank3 = UDV^T.
7. Remove last singular value of D to create Ď.
8. Re-compute matrix F = UĎV^T (rank 2).
9. Denormalize F.

2.1 Normalized 8-point algorithm

To estimate the fundamental matrix, we will follow the normalized 8-point method, described in Algorithm 1.

We use a toy example to check if the implemented function works properly. To do so, we compute the ground truth F_{gt} from two camera matrices: $P = [I|0]$ and $P' = [R|t]$. Knowing $F_{gt} = K^{\prime -T}[T'_x]RK^{-1}$ and that we can obtain K from the camera matrix $P = K[R|t]$, from P' we can easily infer that in our case $K = I$, and therefore $F_{gt} = [T'_x]R$. Then, eight random points are used to estimate F_{est}, which is compared to F_{gt}. Both matrices are presented in Eq. 1. As observed, our algorithm effectively estimates the fundamental matrix, as the difference between them is 0 (e.g. 2.74x10^{-14} for a sample iteration):

$$F_{gt} = \begin{bmatrix} 0.098 & 0.365 & -0.189 \\ -0.365 & 0.098 & 0.567 \\ 0.036 & -0.597 & -0. \end{bmatrix} \qquad F_{est} = \begin{bmatrix} 0.098 & 0.365 & -0.189 \\ -0.365 & 0.098 & 0.567 \\ 0.036 & -0.597 & 0. \end{bmatrix} \tag{1}$$

Another way to check if the fundamental matrix has been correctly estimated, is to select a random point in the first image (p_1) and use its correspondent point in the second image (p_2) to verify the epipolar constraint: $p_2^T F p_1 = 0$. This is presented in the notebook.

2.2 Robust normalized 8-point algorithm

The goal of this section is to estimate F in a real situation where the image correspondences may contain outliers. For that, we implement a robust version of the previous algorithm by using RANSAC to estimate the fundamental matrix.

To do so, we use two real images instead of a toy example. First, we detect and compute the keypoints and matches between them using ORB.

Then, we use RANSAC to estimate F using eight random point correspondences for each iteration. To obtain the inliers, we are using the Sampson error, which is a first order approximation of the geometric distance and it is described by Eq. 2.

$$\sum_i \frac{\left(x_i'^T F x_i\right)^2}{(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T x_i')_1^2 + (F^T x_i')_2^2} \tag{2}$$

The matches and inliers of RANSAC method are presented in Fig. 1. We can observe how most of the mismatches in Fig. 1a were discarded when applying the robust normalized 8-point algorithm (Fig. 1b), although not all of them. As we will explain in Section 2.3, this causes the estimated F not to be accurate enough in some applications.

2.3 Epipolar lines

The goal of this section is to calculate epipolar lines of a point using the fundamental matrix estimated by the robust normalized 8-point algorithm. Epipolar

(a) (b)

Fig. 1: ORB Keypoint (a) matches and (b) inliers between images 1 and 2

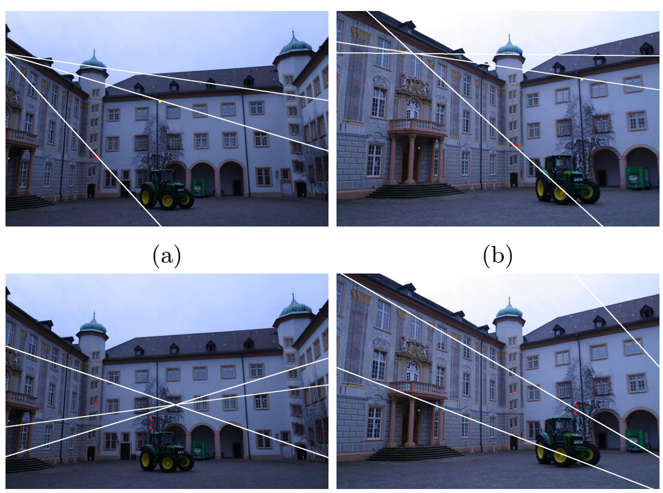
lines are the intersection between the epipolar and image planes. Each point in the 3D world projects into a line in each of the image planes in which it is represented. Having two homogeneous correspondent points p and p' from different image planes and F, their epipolar lines can be computed as:

$$l = F^T \widetilde{p'} \tag{3}$$

$$l' = F \widetilde{p} \tag{4}$$

where l is the epipolar line in which p is located, and vice versa.

The correctness of the calculated epipolar lines can be checked visually in Fig. 2a and Fig. 2b, which show that each epipolar line passes through its corresponding point. In the event that the estimated fundamental matrix F is wrong, epipolar lines would not have gone through the points. An example of this is shown in Fig. 2c and Fig. 2d.



(a) (b) (c) (d)

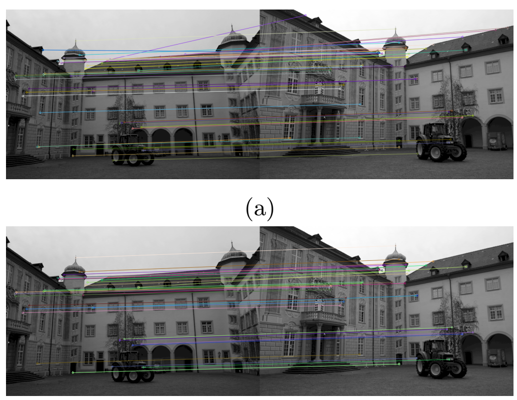
Fig. 2: ORB: good (a,b) and wrong (c,d) epipolar lines on images 1 and 2, from left to right.

SIFT

Since ORB returns many outliers, it is hard to obtain consistently good fundamental matrix estimations. To solve it, we also tried using SIFT to extract the keypoints, so the matching algorithm gets better results and we don't depend so much on the randomness of RANSAC.

Fig. 3b shows the inliers used to estimate F. As it can be seen, there are less correspondences than in ORB, but they are perfectly matched so the estimated F is closer to the ground truth due to RANSAC working better. As we can see now in Fig. 4, the epipolar lines pass through the points.

Another way to check if the epipolar lines are correct is to compute the distance between each point and their correspondent lines, as it is presented in Table 1. It is clear that SIFT works better than ORB for computing fundamental matrices. Furthermore, we don't need to compute F as many times to obtain the desired result.



(a)

(b)

Fig. 3: SIFT Keypoint (a) matches and (b) inliers between images 1 and 2

(a) (b)

Fig. 4: SIFT: good epipolar lines on image (a) 1 and (b) 2

Table 1: Distance between points and epipolar lines

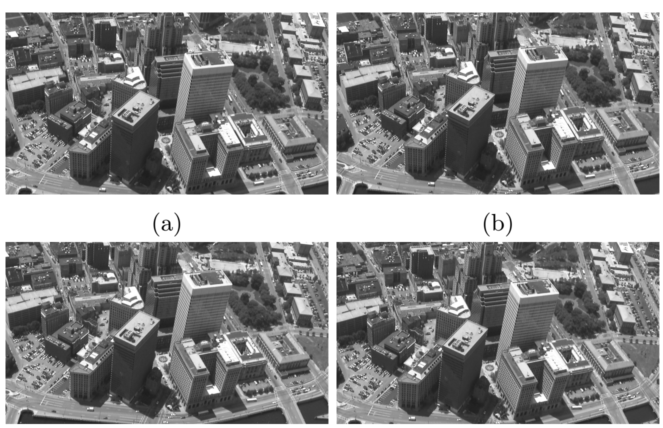
Descriptor	Figure	Point color		
		Blue	Yellow	Red
ORB	2a	3.49	0.40	5.92
	2b	3.61	0.42	6.50
SIFT	4a	0.39	0.38	0.04
	4b	0.38	0.40	0.04

3 Application: Photo-sequencing

In this section, we use a simplified version of [1] to project three points of a van's 3D trajectory at three different time instants that would allow us to order a sequence of frames taken from approximately the same position (Fig. 5). To do so, we pick a dynamic point corresponding to a point on a van and we compute these projections (to the reference image) using the fundamental matrices estimated for each pair of images and the van's trajectory.

We can estimate the different locations of the van on the reference image using the intersection of the trajectory (which is assumed to be straight) and the corresponding epipolar lines.

First of all, we need to ensure correspondences on the dynamic object between the different pairs of images. We use SIFT to obtain the keypoints and the corresponding matches. Visualizing these matches, we manually select the match of interest. An example is presented on Fig. 6. Then, we can proceed to estimate the fundamental matrix F between each pair of images. The matches and resulting inliers are presented in Fig. 7.



(a) (b) (c) (d)

Fig. 5: Sequence of frames

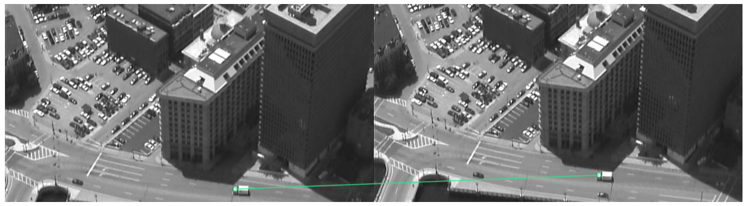
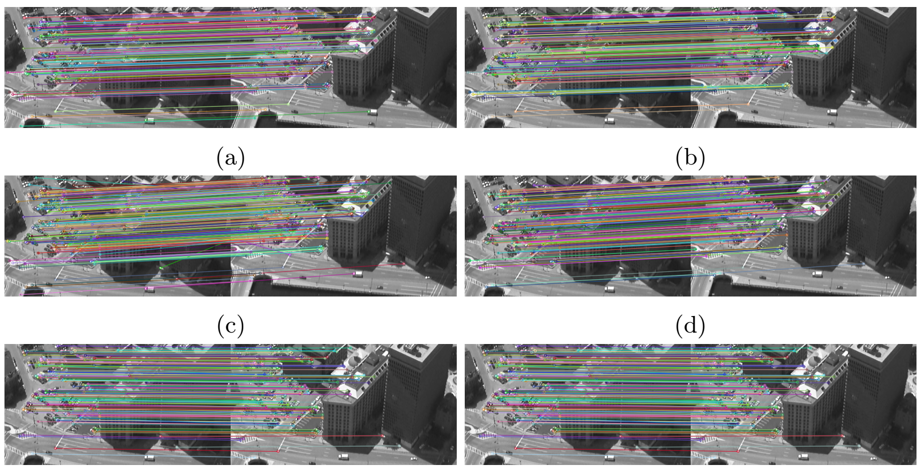


Fig. 6: Match of the keypoints corresponding to the van in images 1 and 2



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 7: SIFT: matches (left) and inliers (right) between image (a,c,e) 1 and images (b) 2, (d) 3 and (f) 4

The line that approximates the straight trajectory of the van is computed with the cross product between the keypoint of interest on the reference image and a manually selected point that is assumed to correspond to that trajectory.

Then, we compute the epipolar lines of each point of interest using Eq. 4. Finally, we obtain the projection of the three points at three different time instants as the intersection between the van's trajectory and the corresponding epipolar lines.

The results are presented in Fig. 8. As observed, we can accurately approximate the location of the van in the reference image for each of the other three time instants. Specifically, the yellow, cyan, blue and red points correspond to the frames of Fig. 5a, 5b, 5c and 5d, respectively. Knowing *a priori* the correct order of the sequence, we can confirm that the implemented algorithm works properly.

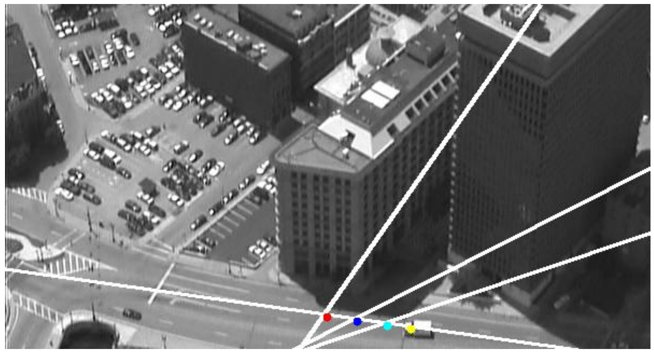


Fig. 8: Projection of three points of the van's 3D trajectory in the reference image

3.1 Optional: Our own images

Now we will apply the implemented method to pictures taken by the members of the team 7, to satisfy our curiosity as engineers.

Pedestrian and van trajectories

In a first experiment, we will examine the trajectory of two dynamic objects that we can daily see on the Barcelona streets from our window: a pedestrian and a van (Fig. 9).

To make it more interesting, their trajectories are in opposite directions. We cannot compute both at the same time, but we want to test if the same sequence of frames allows us to project both 3D trajectories correctly, and thus provide us the same sequence order independently of the dynamic object direction.

The corresponding matches and inliers for each pair of images is presented in the notebook. As an example, we show the matches of interest for the pedestrian in each frame in Fig. 10.

The resulting projections of the 3D trajectories are presented in Fig. 11. As observed, we obtain the same sequence order using any of the dynamic objects.

(a) (b) (c)

Fig. 9: Sequence of frames: Barcelona street

(a)

(b)

Fig. 10: Match of the keypoints corresponding to a pedestrian in images (a) 1 and 2 and (b) 1 and 3

(a)

(b)

Fig. 11: Projection of two points of (a) a pedestrian and (b) a van 3D trajectories in the reference image

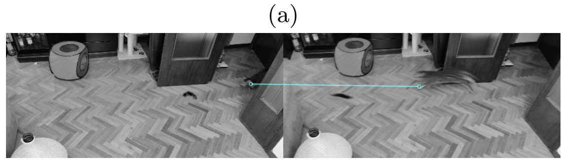
Nala's trajectory

In a second experiment, we will examine the trajectory of our cat (Nala from now on) running and from a much closer point of view (Fig. 12). To test the limits of the algorithm, we will try to estimate the 3D trajectory of a blurry dynamic object.

(a) (b) (c)

Fig. 12: Sequence of frames: Nala playing

(a)



(b)

Fig. 13: Match of the keypoints corresponding to Nala in images (a) 1 and 2 and (b) 1 and 3

In this case, the blurriness of Nala makes more difficult to detect useful keypoints, so we had to tweak the parameters of SIFT descriptor and the matching algorithm to finally obtain the desired keypoint matches. These are presented in Fig. 13.

The resulting projections of Nala's 3D trajectory are presented in Fig. 14. As observed, we can correctly estimate its trajectory with the implemented method.

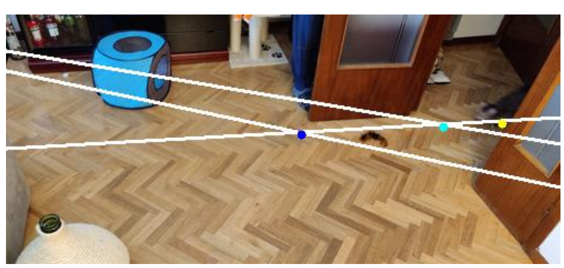


Fig. 14: Projection of two points of Nala's 3D trajectory in the reference image

4 Conclusions

In this week's lab we have worked on fundamental matrix estimation and some of its applications, such as the computation of epipolar lines and photo-sequencing. The obtained results have been certified both qualitatively, by looking at the placement of epipolar lines on an image; and quantitatively, by comparing the estimated and ground truth fundamental matrices, and calculating the distance of the points to their corresponding epipolar lines.

The normalized 8-point algorithm is an efficient method to estimate the fundamental matrix, as it has been demonstrated using a toy example. However, it is sensitive to outliers, so we need to use a more robust method to face this issue. A common alternative is the robust normalized 8-point algorithm, which has been proven to correctly estimate F in different scenarios. Nevertheless, when using ORB keypoint detector we obtain a large number of outliers, which makes it difficult to accurately approximate F. This estimation is too dependent on RANSAC randomness, so we decided to use SIFT, which is a more reliable detector that provides much less outliers. For this reason, RANSAC needs less iterations to find a suitable solution for F, and it is more consistent with the results. Furthermore, the epipolar lines computed for each of the methods confirm that the best results are obtained with SIFT, as the distance between each point and its correspondent epipolar line is minimum.

Regarding the photo-sequencing, we confirmed empirically the usefulness of the aforementioned algorithm, as we could deduct the order of a frame sequence from the trajectory of a dynamic object on the image and the epipolar lines. Specifically, we have been able to project different points of a (van, pedestrian and cat) 3D trajectory at different time instants using frames taken from approximately the same position.

In addition, we have been able to estimate the order of a sequence of frames independently of the dynamic object direction. Finally, we have checked that we can estimate a 3D trajectory of a blurry dynamic object from a much closer point of view. In this case, the main challenge was to find the most suitable parameters to correctly match the keypoints.

References

1. Tali Dekel, Yael Moses, and Shai Avidan, "Photo sequencing," *International Journal of Computer Vision*, vol. 110, no. 3, pp. 275–289, 2014.