Computer Vision/Visual Perception
ESIREM Le Creusot & Master in Computer Vision
Renato Martins – Universite de Bourgogne, February 2022

Practical III - Epipolar Geometry & 8-Point Algorithm

In this exercise, we will estimate the fundamental matrix ($\mathbf{F}$) in the case of an uncalibrated camera. For estimating the fundamental matrix from correspondences, we will implement the well known (widely adopted) 8-Point Algorithm [1][1].

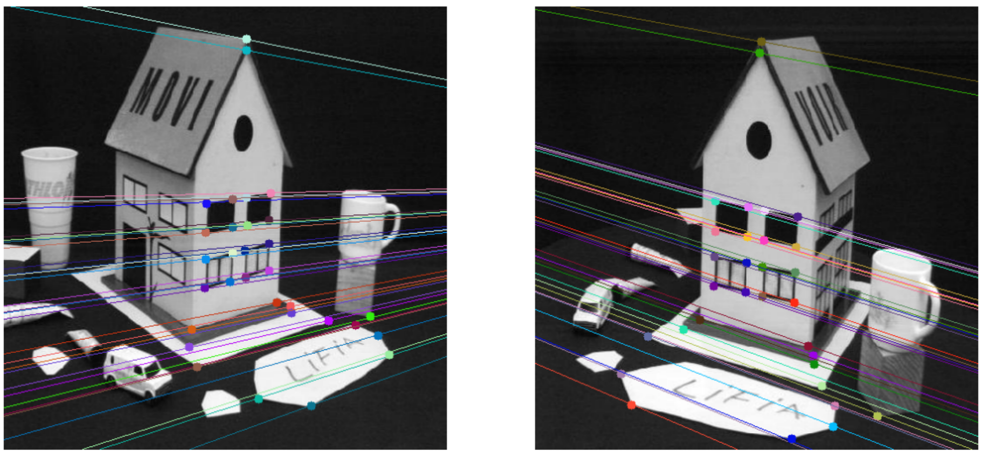

Figure 1: Epipolar lines in left and right views using the estimated Fundamental matrix from correspondences.

IMPORTANT: You can use any high-level OpenCV functions in this practical, with the exception of *cv.findFundamentalMat*. However, you can use *cv.findFundamentalMat* to debug and compare your algorithm.

Part I - Environment Setup & Inputs

Please create a notebook on Colab or follow these steps to configure your conda environment if you are working in your local machine:

```
>> conda create -n epipolar
>> conda activate epipolar
>> conda install jupyter
# Please go to your workspace directory of the course and run jupyter
>> jupyter-notebook
```

We provide the left and right images taken by the same camera as shown in Figure 1, in the folder **images/** in the file *cv_practical_epipolar.zip*. We also provide two text files with the coordinates of corresponding points between the images.

[1]Since $\mathbf{F}$ has only 7 degrees of freedom, only 7 constraints would be required (and indeed there is a 7-Point Algorithm). However, the 8-Point Algorithm is as a linear formulation (instead of the nonlinear 7-Point). Please notice the 8-Point algorithm is also valid and adopted for computing the essential matrix in the calibrated setting (instead of the 5-Point Algorithm)

Part II - Normalized 8-Point Algorithm

We start by estimating the Fundamental matrix F relating the two images. The normalized 8-Point algorithm can be summarized as follows:

> Objective
>
> Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix $\mathtt{F}$ such that $\mathbf{x}'^{\mathsf{T}}_i \mathtt{F} \mathbf{x}_i = 0$.
>
> Algorithm
>
> (i) **Normalization:** Transform the image coordinates according to $\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i$, where $\mathtt{T}$ and $\mathtt{T}'$ are normalizing transformations consisting of a translation and scaling.
>
> (ii) Find the fundamental matrix $\hat{\mathtt{F}}'$ corresponding to the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ by
>
> (a) **Linear solution:** Determine $\hat{\mathtt{F}}$ from the singular vector corresponding to the smallest singular value of $\hat{\mathtt{A}}$, where $\hat{\mathtt{A}}$ is composed from the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ as defined in (11.3).
>
> (b) **Constraint enforcement:** Replace $\hat{\mathtt{F}}$ by $\hat{\mathtt{F}}'$ such that $\det \hat{\mathtt{F}}' = 0$ using the SVD (see section 11.1.1).
>
> (iii) **Denormalization:** Set $\mathtt{F} = \mathtt{T}'^{\mathsf{T}} \hat{\mathtt{F}}' \mathtt{T}$. Matrix $\mathtt{F}$ is the fundamental matrix corresponding to the original data $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

Remember the steps to do so are:

1. Data conditioning: In [1] Hartley proposed a pre-processing step to normalize the image points for better numerical conditioning. This normalization involves applying an affine transformation to the pixel coordinates of each image (as done in the DLT camera calibration assignment).

2. Linear estimation: from the epipolar constraint equation (as shown in the slides of the class and in Chapter 11 of [2]), build the linear system with the entries of F as the unknowns and solve it using SVD. How many correspondences are required?

3. Enforce rank-2 condition: the estimated matrix may not necessarily satisfy the rank-2 condition of a Fundamental matrix. Enforce this constraint explicitly using SVD by setting the last singular value to 0.

4. Denormalize your estimated fundamental matrix.

Part III - Fundamental Matrix Estimation Tests

We will now test the 8-Point Algorithm in some different setups.

1. Using your implemented algorithm, compute the Fundamental matrix and visualize the epipolar lines using the provided ground-truth correspondences. You should obtain a similar result as shown in Figure 1.

2. Compute the estimation error as the average epipolar distance i.e. the average distance from each point to its epipolar line.

3. Test the performance with noise in the image point correspondences:

   (a) Add noise to the pixel coordinates (zero-mean Gaussian noise with an increasing value of the standard deviation, s=0.5, 1, 1.5 ... pixels), and repeat the above steps.

   (b) Observe (and report) the effect of noise on the estimation error.

(EXTRA POINTS) Part IV - Estimation of the Fundamental Matrix with Detected and Described Features

We will now use the SIFT detector and descriptor for extracting real features, instead of the ground truth correspondences.

1. Detect and extract features of the images using SIFT. For having more keypoints you can change the value of *contrastThreshold* of SIFT (such as 0.02). You would have something similar to this:


2. Perform the matching between the descriptors with the SIFT ratio and cross validation tests.

3. Compute the fundamental matrix and estimation error done in **Part III**. What you can observe compared to the case of using the ground-truth correspondences? What can you do for improving this result?

Submission

Please return a concise PDF report explaining your reasoning and equation developments. You should also include your python notebook (commented) with the corresponding implementations/visualizatons, inside a [name]_practical_epipolar.zip file (replacing [name] with your name ;-)). The submission should be done via the teams channel of the course using teams assignment.
Deadline: 08/03/2022 at 23:59pm.

Notes:

- Plagiarism copying the work from another source (student, internet, etc.) will be awarded a 0 mark. In case there are multiple submissions with the same work, each one will receive a 0.
- **Credits:** This assignment was partially inspired by the exercises kindly provided by Devesh Adlakha and by Silvio Savarese/Jeannette Bohg.

References

[1] Hartley, Richard I. "In defense of the eight-point algorithm." IEEE Transactions on Pattern Analysis and Machine Intelligence 19, no. 6 (1997): 580-593.

[2] Richard Hartley and Andrew Zisserman. "Multiple View Geometry in Computer Vision", 2003.