

## Exercises in 3D Computer Vision I

### Exercise 1      **Recap: Linear Algorithm for Homography Computation**

This exercise aims to quickly repeat the basics of the Direct Linear Transform algorithm (DLT) for homography computation. It is very similar to a linear algorithm for computation of the fundamental matrix which we will discuss in Exercise 2.

- a) Outline the main steps of the DLT algorithm for homography computation. Assume you are given point correspondences  $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}_{i=1\dots N}$ , where  $\mathbf{x} = (x, y, w)^\top$ . How many point correspondences are required to compute a homography  $\mathbf{H}$  such that  $\mathbf{x}' = \mathbf{H}\mathbf{x}$ ?
- b) In practice, the DLT as described above often does not give satisfactory results. Outline why normalization has to be applied to the point correspondences and how you would perform normalization and denormalization.

### Exercise 2      **Linear Computation of the Fundamental Matrix**

In this exercise, we will introduce a linear algorithm for computing the fundamental matrix  $\mathbf{F}$  from point correspondences in two images. Assume again you are given point correspondences  $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}_{i=1\dots N}$  in two images, originating from common 3D points  $\mathbf{X}_i$ . We write  $\mathbf{x} = (x, y, w)^\top$  and similarly for  $\mathbf{x}'$ .

- a) Starting from the defining equation of the fundamental matrix,  $\mathbf{x}'^\top \mathbf{F} \mathbf{x} = 0$ , think of how to obtain a homogeneous linear system of equations of the form  $\mathbf{A} \mathbf{f} = \mathbf{0}$ . Specifically, answer the following questions:
  - How many constraints do you obtain per point correspondence?
  - How many point correspondences do you need to fully determine the entries of  $\mathbf{F}$ ?
  - What are the dimensions of  $\mathbf{A}$  and  $\mathbf{f}$ ?
- b) How do you solve the linear system of equations in order to get a solution for  $\mathbf{F}$ ?
- c) An additional constraint on  $\mathbf{F}$  is that its rank is 2, i.e.  $\mathbf{F}$  is a singular matrix. Does the matrix computed in the step before necessarily satisfy this constraint? If not, how can the singularity constraint be enforced algebraically when computing  $\mathbf{F}$ ?
- d) As in the case of the DLT algorithm, normalization is required to avoid numerical instability. Explain how to apply normalization and denormalization when computing  $\mathbf{F}$ .

### Exercise 3      **Computing the Fundamental Matrix based on 6 Point Correspondences**

At least 8 point correspondences are required to determine  $\mathbf{F}$  when the 3D points generating the correspondences are assumed to be in a general configuration (e.g. no collinear or coplanar points). However, special configurations allow us to compute the fundamental matrix between

two views from less than 8 point correspondences. For example, six point correspondences are sufficient, if four of the 3D points lie on a plane and two of the 3D points are out-of-plane. The resulting method of computing the fundamental matrix is called the **6-point algorithm**.

Assume that there are four coplanar 3D points  $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4$  and two out-of-plane points  $\mathbf{X}_5, \mathbf{X}_6$ . The respective projections in the two views are denoted by  $\mathbf{x}_i$  and  $\mathbf{x}'_i, i \in \{1, \dots, 6\}$ .

- a) The projections of the four coplanar points in the two views can be used to obtain a homography  $\mathbf{H}$  between the two images. How can you compute this homography?
- b) Show that  $\mathbf{F} = [\mathbf{e}']_{\times} \mathbf{H}$ , where  $\mathbf{e}'$  is the epipole in the second view. Remember from last week's exercise that such a homography induced by a 3D plane can be decomposed as  $\mathbf{H} = \mathbf{K}_2(\mathbf{R} + \mathbf{t}\mathbf{n}_1^{\top}/d_1)\mathbf{K}_1^{-1}$ , where  $\mathbf{K}_1, \mathbf{K}_2$  are the intrinsic parameters matrices of the two cameras,  $\mathbf{R}$  and  $\mathbf{t}$  is the rotation and translation between the camera centers, and  $\mathbf{n}_1$  is the normal of the 3D plane.

*Hint:* Use the following properties:

- $\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$ .
  - $\mathbf{F} = \mathbf{K}_2^{-\top} \mathbf{E} \mathbf{K}_1^{-1}$ .
  - For any non-singular  $\mathbf{A} \in \mathbb{R}^{3 \times 3}$  and any  $\mathbf{b} \in \mathbb{R}^3$ :  $[\mathbf{A}\mathbf{b}]_{\times} = \mathbf{A}^{-\top} [\mathbf{b}]_{\times} \mathbf{A}^{-1}$ .
- c) Applying  $\mathbf{H}$  to the projections of the out-of-plane points  $\mathbf{x}_5, \mathbf{x}_6$  in the first image will not necessarily give you the corresponding true projections  $\mathbf{x}'_5, \mathbf{x}'_6$  in the second image. However, the mapped points  $\mathbf{H}\mathbf{x}_5, \mathbf{H}\mathbf{x}_6$  will lie on the correct epipolar lines in the second image. Think of why this is the case, e.g. with a drawing.
  - d) You have seen in part b) of this exercise that you can compute the fundamental matrix based on the homography  $\mathbf{H}$  induced by the coplanar points and the epipole in the second image  $\mathbf{e}'$ . Now, think of how you can obtain this epipole.

#### **Exercise 4 (H) 8-Point Algorithm in MATLAB**

In this exercise, you are asked to implement the 8-point algorithm as discussed above in MATLAB. Your algorithm should contain normalization/denormalization and it should enforce the singularity constraint on your estimated matrix  $\mathbf{F}$ .

- a) Write a MATLAB function `eightpoint` that takes an arbitrary number ( $\geq 8$ ) of point correspondences  $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$  and outputs the fundamental matrix  $\mathbf{F}$  such that  $\mathbf{x}'_i{}^{\top} \mathbf{F} \mathbf{x}_i = 0$ .
- b) Write a MATLAB script that performs the following steps: Load and display the images `mag_table_1.jpg` and `mag_table_2.jpg` from the previous homework (Exercise 3) and ask the user to manually specify at least 8 point correspondences between the two images (e.g. using `ginput`). Use your function `eightpoint` to compute the fundamental matrix consistent with the selected points.
- c) Repeat the tasks in Exercise 3 of the previous homework, this time not using the provided matrix  $\mathbf{F}$  but one you computed using your own code, and compare the results.