Exercise 3: Stereo vision, and triangulation

02504 Computer vision

Morten R. Hannemose, mohan@dtu.dk, DTU Compute

February 18, 2022

Epipolar geometry

Set up two cameras, both with the internal parameters

$$\boldsymbol{K} = \begin{bmatrix} 1000 & 0 & 300 \\ 0 & 1000 & 200 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

Now, for the first camera — let us call that `Cam1` — set the rotation to identity $\boldsymbol{R}_1 = \boldsymbol{I}$ and set the translation to zero $\boldsymbol{t}_1 = \boldsymbol{0}$. For the second camera `Cam2` use the rotation given by the $\boldsymbol{\mathcal{R}}$ function

$$\boldsymbol{R}_2 = \boldsymbol{\mathcal{R}}(0.7, -0.5, 0.8) \tag{2}$$

$$\boldsymbol{\mathcal{R}}(\theta_x, \theta_y, \theta_z) = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix}, \tag{3}$$

and the translation

$$\boldsymbol{t}_2 = \begin{bmatrix} 0.2 \\ 2 \\ 1 \end{bmatrix}. \tag{4}$$

The rotation can be constructed in Python using `Rotation` module from `scipy` as follows:

```
from scipy.spatial.transform import Rotation
R2 = Rotation.from_euler('xyz', [0.7, -0.5, 0.8]).as_matrix()
```

Exercise 3.1

Consider the 3D point

$$\boldsymbol{Q} = \begin{bmatrix} 1 \\ 0.5 \\ 4 \\ 1 \end{bmatrix} \tag{5}$$

and find the projections in `Cam1` and `Cam2`, respectively, points $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$.

Exercise 3.2

Implement a function `CrossOp` that takes a vector in 3D and returns the 3×3 matrix corresponding to taking the cross product with that vector. In the case that $\boldsymbol{p} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$ you should have

$$\texttt{CrossOp}(\boldsymbol{p}) = [\boldsymbol{p}]_{\times} = \begin{bmatrix} 0, & -z, & y \\ z, & 0, & -x \\ -y, & x, & 0 \end{bmatrix}. \tag{8}$$

As a good habit, verify that your function works by testing it on random vectors to ensure that

$$[\boldsymbol{p}_1]_{\times}\boldsymbol{p}_2 = \boldsymbol{p}_1 \times \boldsymbol{p}_2. \tag{9}$$

Exercise 3.3

Compute the fundamental matrix $\boldsymbol{F}$ of the two cameras.

Exercise 3.4

What is the epipolar line $\boldsymbol{l}$ of $\boldsymbol{q}_1$ in camera two?

Exercise 3.5

Is $\boldsymbol{q}_2$ located on the epipolar line from Ex. 3.4? Do the computations, but also explain why this *must* be so.

Exercise 3.6

Now assume that both camera one and two have local coordinate systems that are different from the coordinate system of the world.

Let $\boldsymbol{Q}$ and $\tilde{\boldsymbol{Q}}$ denote *the same* 3D point in world space and in the frame of camera one. In other words we have relation

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{Q}. \tag{13}$$

Make sure you understand why this is true.

Show analytically that

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{R}_1^{\mathrm{T}} & -\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{Q}}. \tag{14}$$

Exercise 3.7

Show that the projection can work only in the coordinate system of camera one, by showing that we can project points with

$$\boldsymbol{q}_1 = \boldsymbol{K}\begin{bmatrix}\boldsymbol{I} & 0\end{bmatrix}\tilde{\boldsymbol{Q}}, \text{ and } \boldsymbol{q}_2 = \boldsymbol{K}\begin{bmatrix}\tilde{\boldsymbol{R}}_2 & \tilde{\boldsymbol{t}}_2\end{bmatrix}\tilde{\boldsymbol{Q}}, \tag{18}$$

where

$$\tilde{\boldsymbol{R}}_2 = \boldsymbol{R}_2\boldsymbol{R}_1^{\mathrm{T}}, \text{ and } \tilde{\boldsymbol{t}}_2 = \boldsymbol{t}_2 - \boldsymbol{R}_2\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{t}_1. \tag{19}$$

Applied epipolar geometry

Exercise 3.8

Load the file TwoImageData.npy, and compute the fundamental matrix between camera one and two.
Tip: You can load the file with:
`np.load('TwoImageData.npy', allow_pickle=True).item()`

Exercise 3.9

Write code that can show both images at the same time. Now write code such that you can click on a point in image one, and display the corresponding epipolar line in image two. Experiment with your code, verifying that the point you click on is on the epipolar line in the other image.

Tip: To click on a point and get the pixel coordinates you can use `plt.ginput(1)`

Tip: To draw a line given in homogeneous coordinates, you can use the `DrawLine` function below. It takes as input a line in homogeneous coordinates `l` and the size of the image it will be drawn on, as returned by `im.shape`.

```
def DrawLine(l, shape):
    #Checks where the line intersects the four sides of the image
    # and finds the two intersections that are within the frame
    def in_frame(l_im):
        q = np.cross(l.flatten(), l_im)
        q = q[:2]/q[2]
        if all(q>=0) and all(q+1<=shape[1::-1]):
            return q
    lines = [[1, 0, 0], [0, 1, 0], [1, 0, 1-shape[1]], [0, 1, 1-shape[0]]]
    P = [in_frame(l_im) for l_im in lines if in_frame(l_im) is not None]
    plt.plot(*np.array(P).T)
```

Exercise 3.10

Do the same thing as the last exercise, but where you can click in image two and get the epipolar line displayed in image one.

Programming exercise: Triangulation

Exercise 3.11

Create a function `triangulate` that takes a list of pixel coordinates (`q1`, `q2`, . . . , `qn`), and a list of projection matrices (`P1`, `P2`, . . . , `Pn`), and the function returns the triangulation of the point using the linear algorithm.

Test your function by defining some 3D points — use the `box3D` function, for example — project them to the image planes of the two cameras, and then re-triangulate them back to the original coordinates.

You can similarly use a number of corresponding pixel coordinates, triangulate them, and then re-project the 3D points into the cameras. Do you find the same 2D pixels? If not, why?

Solutions

Answer of exercise 3.1

The projections are

$$\boldsymbol{p}_1 = \begin{bmatrix} 550 \\ 325 \end{bmatrix}, \text{ and} \tag{6}$$

$$\boldsymbol{p}_2 = \begin{bmatrix} 582.473 \\ 185.990 \end{bmatrix}. \tag{7}$$

Answer of exercise 3.3

The fundamental matrix is

$$\boldsymbol{F} = \begin{bmatrix} 3.293 \cdot 10^{-7} & 8.194 \cdot 10^{-7} & 1.792 \cdot 10^{-3} \\ 5.155 \cdot 10^{-7} & -8.769 \cdot 10^{-7} & 9.314 \cdot 10^{-5} \\ -1.299 \cdot 10^{-3} & 1.520 \cdot 10^{-3} & -1.101 \end{bmatrix}. \tag{10}$$

Answer of exercise 3.4

The epipolar line is found by $\boldsymbol{F}\boldsymbol{p}_1$

$$\boldsymbol{l} = \begin{bmatrix} 8.956 \cdot 10^{-3} \\ 3.668 \cdot 10^{-4} \\ -5.285 \end{bmatrix}. \tag{11}$$

Remember that you can get different numbers, if you have a different scale of $\boldsymbol{p}_1$, but the line should be the same up to scale.

Answer of exercise 3.5

To see if a point $\boldsymbol{q}$ is on a line we can use that $\boldsymbol{q}^{\mathrm{T}}\boldsymbol{l} = 0$:

$$\boldsymbol{q}_2^{\mathrm{T}}\boldsymbol{l} = -1.943 \times 10^{-15} \approx 0. \tag{12}$$

Taking numerical precision into account, the point is on the line.

This must be true, since both the point $\boldsymbol{q}_2$ and the line $\boldsymbol{l}$ are derived from the same 3D point $\boldsymbol{Q}$. This 3D point yields a single epipolar plane, and the plane yields a single line in each camera. The projections of the 3D point must lie on the epipolar lines.

Answer of exercise 3.6

We insert Eq. (14) into Eq. (13):

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_1^{\mathrm{T}} & -\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{Q}}, \tag{15}$$

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{R}_1\boldsymbol{R}_1^{\mathrm{T}} & -\boldsymbol{R}_1\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{t}_1 + \boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{Q}} \tag{16}$$

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{Q}}. \tag{17}$$

And we find that it is valid! This is true as the matrices are inverses of each other.

Answer of exercise 3.7

For the first projection in camera one we reduce the projection equation:

$$\boldsymbol{q}_1 = \boldsymbol{K}[\boldsymbol{R}_1|\boldsymbol{t}_1]\boldsymbol{Q}, \tag{20}$$

$$= \boldsymbol{K}\begin{bmatrix} \boldsymbol{I} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{Q}, \tag{21}$$

$$= \boldsymbol{K}\begin{bmatrix} \boldsymbol{I} & 0 \end{bmatrix} \tilde{\boldsymbol{Q}}, \tag{22}$$

$$\tag{23}$$

For the second projection into camera two we insert Eq. (14)

$$\boldsymbol{q}_2 = \boldsymbol{K}[\boldsymbol{R}_2|\boldsymbol{t}_2]\boldsymbol{Q}, \tag{24}$$

$$\boldsymbol{q}_2 = \boldsymbol{K}[\boldsymbol{R}_2|\boldsymbol{t}_2] \begin{bmatrix} \boldsymbol{R}_1^{\mathrm{T}} & -\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{Q}}, \tag{25}$$

$$\boldsymbol{q}_2 = \boldsymbol{K}\begin{bmatrix} \boldsymbol{R}_2\boldsymbol{R}_1^{\mathrm{T}} & \boldsymbol{t}_2 - \tilde{\boldsymbol{R}}_2\boldsymbol{t}_1 \end{bmatrix} \tilde{\boldsymbol{Q}}, \tag{26}$$

$$\boldsymbol{q}_2 = \boldsymbol{K}\begin{bmatrix} \tilde{\boldsymbol{R}}_2 & \tilde{\boldsymbol{t}}_2 \end{bmatrix} \tilde{\boldsymbol{Q}}. \tag{27}$$