Laboratory Session 2: Homography, Fundamental Matrix and Two View SfM

In this laboratory session we are going to introduce the structure from motion problem from the estimation of the fundamental matrix and homographies.

Goals of the assignment:

1. Triangulate 3D points from matches in two localized and calibrated cameras.
2. Two view structure from motion based on fundamental matrix estimation
3. Understanding the scale loss in structure from motion problem
4. Understanding and computing homographies between two views.

Evaluation of the assignment:

The resulting work will be shown by presenting the obtained results in the following Laboratory session, and the code will be submitted through the ADD (Moodle).

1. Point triangulation

image 1, undistorted image   Image 2, undistorted image

Figure 1: Undistorted images from EuRoC dataset.

Consider two monocular images from EuRoC dataset, with common field of view and significant parallax.

From a set of matches from the two views, assuming that the camera poses and calibration is known, compute the projection matrices and triangulate the 3D coordinates of the corresponding 3D points in the world (`W`) reference.

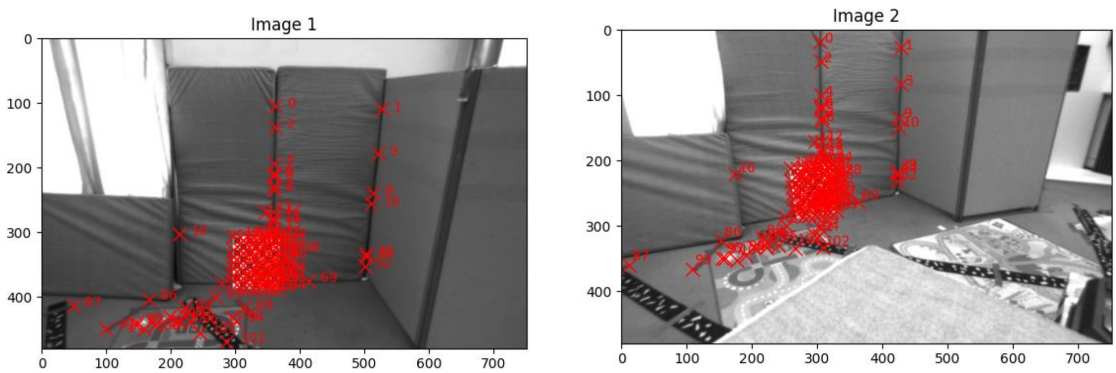

Figure 2: Matched points between both images

We provide the intrinsic calibration of the camera `K_c` (both images have been acquired with the same camera), the extrinsic parameters (poses `T_w_c1` and `T_w_c2`) of the two cameras and the point matches of each images.

Intrinsic parameters of the camera (both images had been captured with the same camera). α_x= 458.654 px, α_y= 457.296 px, x_0= 367.215 px, y_0= 248.375 px

The extrinsic parameters for both cameras can be found in `T_w_c1.txt` and `T_w_c2.txt`.

```
T_w_c1 =[[-0.0744, -0.3079,  0.9485,  0.8801],
         [-0.997 ,  0.0428, -0.0643,  0.8695],
         [-0.0208, -0.9505, -0.3102,  1.73  ],
         [ 0.    ,  0.    ,  0.    ,  1.    ]]
T_w_c2 = [[-0.512 , -0.2779,  0.8128,  0.5236],
          [-0.8586,  0.194 , -0.4745,  1.9537],
          [-0.0258, -0.9408, -0.338 ,  1.2876],
          [ 0.    ,  0.    ,  0.    ,  1.    ]]
```

The point matches of each image are given in files `x1Data.txt` and `x2Data.txt`. The solutions of the 3D coordinates of each point are in the file `X_w.txt`. We also provide the example code `plotData.py` where you can find an example of plotting the ground truth data (poses, 3D points) that you can use for plotting your own results.

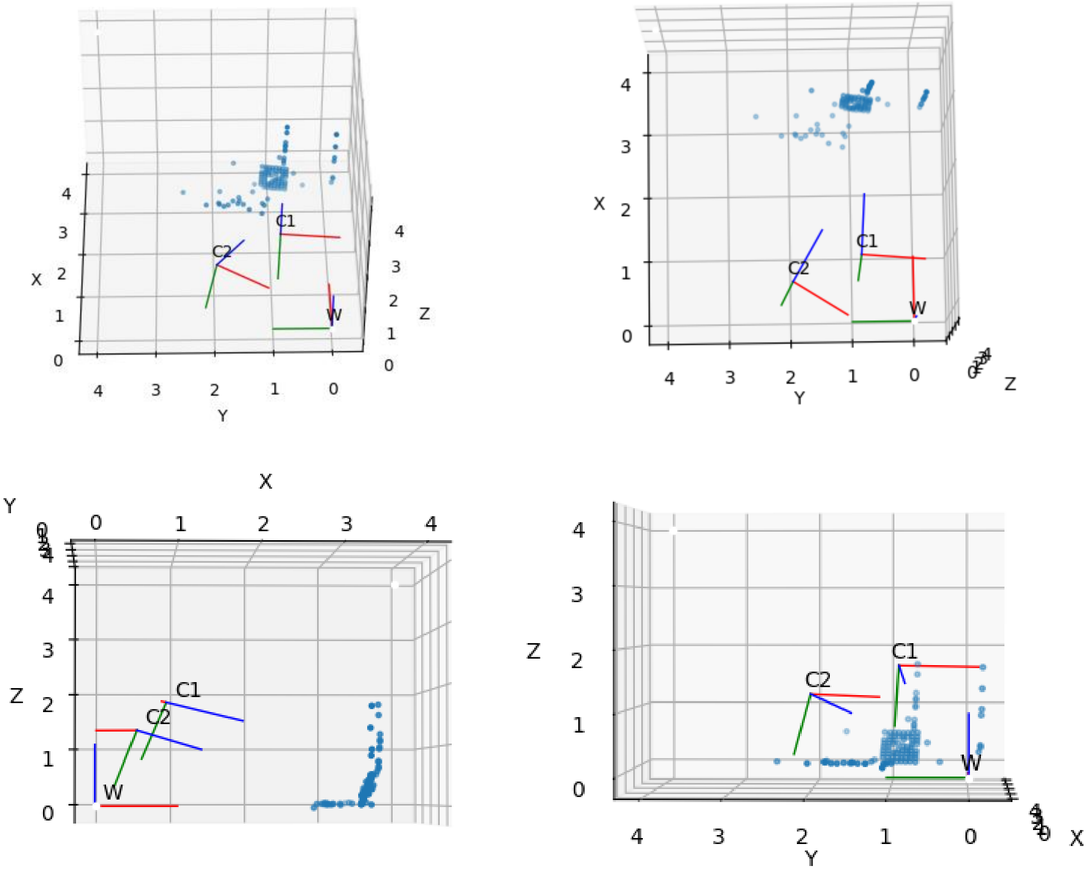

Figure 3: 3D Plotting of the two ground truth poses and the 3D points example.

2. Fundamental matrix and Structure from Motion

2.1 Epipolar lines visualization

Implement a function for representing an epipolar line in image 2 given a clicked point on image 1. For checking your code you can use the testing fundamental matrix `F_21_test.txt` provided in the support files of the practice.

2.2 Fundamental matrix definition

From the provided ground truth poses (`T_w_c1` and `T_w_c2`), compute the fundamental matrix that relates both images also visualizing the epipolar lines and the epipole. **Notice that the provided ground truth is an approximation**.

2.3 Fundamental matrix linear estimation with eight point solution.

Compute an estimation of the Fundamental Matrix from the set of matches `x1Data.txt` and `x2Data.txt` (see Fig. 2). Use the estimated Fundamental matrices to compute and visualize the epipolar lines and the epipole.

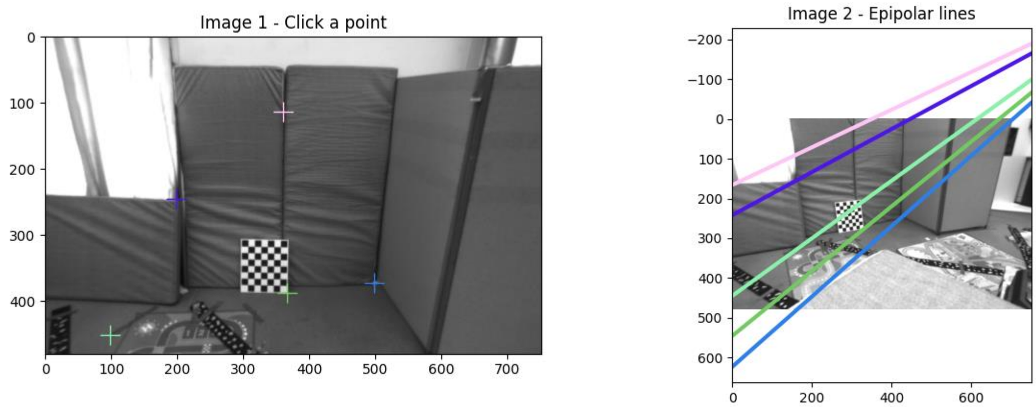

Figure 4: Example of visualization of epipolar lines

2.4 Pose estimation from two views

From `F_21`, obtain the essential matrix `E_21` and the resulting four different estimations of the camera transformation `T_21`. From the matches between image one and two triangulate the 3D points and uses them to discriminate the correct solution.

2.5 Results presentation

Visualize the cameras and the 3D points and compare them with the provided ground truth. Propose and use a metric for evaluating the accuracy of your results.

3. Homographies

3.1 Homography definition

From the provided poses and the plane equation coefficients on camera reference 1, compute the homography that relates both images through the floor plane. **Notice that the provided ground truth is an approximation**.

```
Pi_1 = (Pi_w.T @ T_w_1).T = [[0.0149,0.9483,0.3171,-1.7257]].T
```

3.2 Point transfer visualization

Implement, represent and visualize the point transfer between projections of 3D points lying on the ground plane via the estimated homography.

3.3 Homography linear estimation from matches

From a set of matches of points lying on the floor (`x1FloorData.txt` and `x2FloorData.txt`) estimate the homography matrix relating both images.

Propose and use a metric for evaluating the accuracy of your results.