İTÜ

BLG634E 3D Vision

Guideline for submitting your homework:

In submitting your homework:

i. Comment your codes clearly.
ii. All your code should be written in Python language.
iii. Do not send your HWs via e-mail. No exception!

In this assignment, you will work on Epipolar Geometry. You will implement the well-known 8-point Algorithm in 3D vision to reconstruct camera poses, and 3D structure (depth) using point correspondences. Load the images (1.JPG and 2.JPG) provided in assignment zip file. In order to work within a calibrated epipolar geometry scenario, we will use the intrinsic camera parameters, which are given in the cube data.mat file. Load the mat file and see the 3x3 intrinsic camera calibration matrix "Calib". This mat file also contains a set of 45 point correspondences selected where x1 and x2 contain homogeneous point pairs from image 1 and image 2 respectively. You will estimate the rotation and translation (R,T), i.e. the relative camera pose. In addition, you will obtain a point cloud of the 3D structure from a collection of corresponding points from two views of the given object.

Task 1: 8-pt Algorithm

Estimate Essential Matrix E by implementing the 8-point algorithm as we learned in class. Do not forget to normalize the image points as we learned before (centroid 0 and variance of distances to centroid/origin normalized to 1). Also, remember to project your essential matrix onto the Space of Essential Matrices.

Task 2: Decompose the Essential Matrix

Decompose the essential matrix you estimated in Task 1 to obtain the possible rotation, translation (R,T) pairs.

Task 3: Impose Positive Depth Constraint

Remember to impose the positive depth constraint to choose the correct (R,T) pair.

Task 4: Recover 3D structure as a Point Cloud

Recover the 3D structure, i.e. a 3D point cloud, as we learned in the class by triangulation. Visualize/print your point cloud from different render viewpoints (**Nonobligatory Hint from Yusuf:** You may obtain better visual results by using VisPy library). Discuss the performance of your result, and possible sources of error.

Task 5: Dense Correspondence with RANSAC and 3D Reconstruction (EXTRA)

Are you satisfied with your 3D reconstruction? If not, in order to improve the reconstruction, you can go for a dense correspondence matching between the two views (by your favorite dense feature extraction and matching scheme such as SURF), eliminate the outliers by RANSAC, perform your triangulation on the resulting dense set of points and visualize the result by fitting a mesh surface to your 3D point cloud. Visualize/print your result from different render viewpoints. Then you can even try to color your mesh by texture mapping.