BLG634E 3D Vision – Project Part 2

Guideline for submitting your homework:

In submitting your homework:

- Comment your codes clearly.
- All your code should be written in Python language.
- The code of this homework will be the basis fort he final project. Thus, be careful to write your code as functions.
- Do not send your HWs via e-mail. No exception!

Task 1 - Camera Calibration

Figure 1: Some example photos to be used in this part.

Camera calibration estimates the intrinsic and extrinsic parameters of a camera, including radial lens distortion coefficients. For this task you will use Zhang's camera calibration approach to calibrate a camera using a set of images. **You may code direct implementation of Zhang's camera calibration for bonus points.**

First, you should detect the checkerboard corners, the axis of the checkerboard in each image. Set the origin to the top left corner you detected. You can manually pick the corner points in the images in a certain order.

Once you have the measured image points, follow the steps in Zhang's technical report, which we studied in class, and implement the calibration including lens distortion.

In the end, display a scatter plot of the reprojection errors. In practice, reprojection error of less than 1 mm is typically considered "good", particularly for a webcam.

Show the distorted images and lens undistorted images. Note the change in appearance, particularly further away from the centre of the image.

Once you are finished with the final nonlinear optimization (bundle adjustment/refinement), you have the camera calibration parameters, say cameraParams.

First, check the camera intrinsics, in terms of the focal lengths, principal point (centre of image), skew, lens distortion, and intrinsic matrix. There should be also N sets of extrinsic parameters, one for each image. Now, form you full camera projection matrix M.

Task 2 – Augmented Reality

With the camera intrinsics and extrinsics known, we can project from 3D space into the images. To do this, we must:

1. Undistort the image
2. Compute the projection matrix
3. Project 3D points to 2D points

Using this procedure, project a the mesh model onto calibration map from the images. An example output is given in Figure 2.

Figure 2: An example AR output.

Task 3 – 8-Point Algorithm & Depth Estimation

Figure 3: Inputs for Task 3.

You will use 1.JPG, 2.JPG and Calib.mat files for this task. “Calib.mat” is a mat file which contains a MATLAB dictionary of

- x1 and x2: homogeneous point pairs from image 1 and image 2
- calib: The intrinsic camera calibration matrix

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.

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

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

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. Discuss the performance of your result, and possible sources of error.

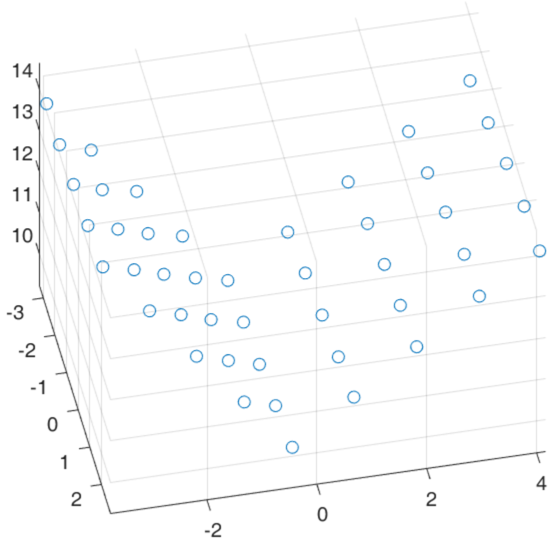

Figure 4: Output point cloud for the given images.