ECE 472 Robotics and Vision
Prof. K. Dana
Homework 4: Stereo Reconstruction: Camera Calibration

Submit the code as well as a pdf file with the output of your code execution (can be screenshots).

1. In prior assignments you completed the following task: *Create a 3D wireframe model of a simply polygonal house. Write a python program to image a wireframe house described by a list of 3D vertices and a list of edges (vertex pairs) in world coordinates (pts). Choose an intrinsic parameter matrix (K), and* **two** *camera views (rotation and translation). Use the intrinsic parameter matrix and the camera view to project the 3D world points pts to 2D points. The projections should be handled by using the camera matrix (image formation pipeline) discussed in class. Create a function drawmyobject in python to draw 2d lines between 2d vertices. Do not use plot3 or any 3d plotting commands.* For this assignment you will implement **Stereo Reconstruction** Write a python program to triangulate to find reconstruct the world coordinates of 3 points of the house. Utilize your known calibration parameters and your known point correspondences. Show the *reprojection error* by imaging the *estimated* 3D point with the known camera matrices and compute the different between this *reprojection* and the original pixel coordinates. Report the sum of the square of the reprojection error over all vertices.

2. Use the CalTech Camera Calibration method in opencv-python to calibration your own camera using images of a checkerboard pattern. You should obtain the value of K for your camera as well as one of the wTc extrinsic parameters (corresponding to one pose of the camera). Show the visualization of the camera views reconstructed. Output your calibration matrix M.

3. Write code to find M find K, and ${}^{c}T_{w}$, using the methods that we discussed in class.

4. Does your ${}^{c}T_{w}$ make sense from your camera configuration? Why? Use sentences and sketches to justify your answer.

5. **Estimate Fundamental Matrix** Image a polygonal house in a right camera and left camera positioned for a non-obstructed view of your model. Do not use parallel cameras. Use point correspondences from the images of your model to estimate the fundamental matrix. Write python code to estimate and output the fundamental matrix using the 8-point algorithm.

6. **Estimate the Essential Matrix** Using your knowledge of the camera's K matrix of intrinsic parameters, write python code to find the essential matrix.

7. **3D reconstruction with E** Write python code to reconstruct the points in 3D using the essential matrix and triangulatePoints() algorithm. Show the following images as output:

   (a) left image of the polygonal model (draw 2d lines between 2d vertices)

(b) right image (draw 2d lines between 2d vertices)

(c) reconstructed polygonal model (draw 3D lines between 3D vertices)

8. **Euclidian 3D reconstruction** Write python code to reconstruct the points in 3D using the camera matrix and triangulatePoints() algorithm. Show the following images:

   (a) left image of the polygonal model (draw 2d lines between 2d vertices)

   (b) right image (draw 2d lines between 2d vertices)

   (c) reconstructed polygonal model (draw 3D lines between 3D vertices)