CMPEN/EE 454 Project 2

Due 11/06/2023 by 11:59pm (upload to Canvas)

Overview
The goal will to be to use the tools we have developed through the last few lectures by considering stereo image pairs coupled with motion capture data. You'll be given a pair of images taken from different cameras simultaneously. The person in the image is tagged with motion capture markers which were used to get accurate 3D point measurements. In this scenario you have all of the intrinsic and extrinsic camera information, so you'll be able to explore the 3D scene using the methods we've developed and use the motion capture data as a ground truth.

Data

1. Two images *im1corrected.jpg* and *im2corrected.jpg* which represent the stereo pair.
2. Intrinsic and Extrinsic parameters for each camera *Parameters_V1_1.mat* and *Parameters_V1_2.mat*. Note that you'll need to determine what parameters are what for this data using the information you know about the pinhole camera model. You'll need to think about the pieces of the camera model (i.e. matrices used for the conversions) in order to determine what each piece means. This exploration is part of the project.
3. Motion capture measurements *mocapPoints3D.mat* which contains point locations of the 39 markers on the person's body. These are given in the world coordinate system with $(0, 0, 0)$ being the middle of the floor, positive Z-axis is up, and the units are in millimeters.

Tasks

1. **Project 3D mocap points into 2D pixel locations.** Your projections should results in pixel locations which correspond to the markers on the person's body (note they may be not always be visible). <u>Provide a plot indicating the locations of your 2d point locations overlayed on the image.</u>
2. Use the 39 2D pixel locations computed in task 1 and **perform triangulation to recover the 3D points.** <u>Compare the accuracy of your computed points and the mocap data using the mean square error (this should be small).</u>
3. **Use triangulation to measure objects in the scene.** You'll obtain the pixel coordinates by clicking on the 3D object in both images; you'll then use these to construct the 3D point in the scene. Performing the following tasks:
   (a) Measure the 3D locations of 3 points on the floor and fit a 3D plane to them. Verify that your computed floor plane is approximation $Z = 0$. <u>What is the equation of the floor plane?</u>
   (b) Measure the 3D locations of 3 points on the wall that has white vertical stripes painted on it and fit a plane. <u>What is the equation of the wall plane?</u>
   (c) Answer the following additional questions:
      - How tall is the doorway?
      - How tall is the person?
      - What is the 3D location of the center of the camera that can be seen in both images?

4. **Compute the Fundamental matrix using the 8-point algorithm.** Show the fundamental matrix and the epipolar lines.

5. **Compute the accuracy of the epipolar lines using the Symmetric Epipolar Distance**. This is computed as follows. In Task 1 we generated 39 image points. For a given 3D point let (x_1, y_1) denote its location in image 1 and (x_2, y_2) its location in image 2. Use your Fundamental matrix from the previous task to compute an epipolar line in image 2 using (x_1, y_1) and compute the squared geometric distance of (x_2, y_2) from the line. Then do the reverse, compute the epipolar line in image 1 using (x_2, y_2) and compute the squared geometric distance of (x_1, y_1) from the line. Note that for a line defined by $ax + by + c = 0$, the square geometric distance for a point $(\hat{x}, \hat{y})$ is

$$d^2 = \frac{(a\hat{x} + b\hat{y} + c)^2}{a^2 + b^2}.$$

   Then repeat this process for each of the 39 points. To get a single number describing the accuracy of the epipolar lines, take all of the distance you have computed and report the mean of these errors and report it. This will be a single number.

What to turn in

A single zip file with the following.

1. Your code in the following format:
   - Make separate matlab files for each task labeled: *task_1.m*, *task_2.m*, *task_3a..m*, *task_3b.m*, *task_3c.m*, *task_4.m*, *task_5.m*. If you have any additional helper functions please give them very descriptive names.
   - Include very descriptive comments so it is clear you know what you are doing. Think of this as a written explanation to be graded.
   - Each file should run like a demo, where the output clearly displays your results and answers to the questions. Please leave no ambiguity, we should not have to search for answers.
2. A 5 minute video presentation which includes the following:
   - Title slide with group members and their contributions
   - Run each of your demos and explain: what is begin computed, what the output is, and how that answers the questions. If you didn't get something working, this is a good time to explain what you have and what the problem was.

Project Parameters

1. You'll need to implement your solutions in Matlab
2. You're restricted to the general routines we've been using in class thus far. **You'll want to implement what we have detailed in this project description. Do not use code off of Github or Stack Exchange.** If your implementation is some super elegant generalization of the procedures we detail, we will be suspicious of your code and may request a meeting for you to provide additional explanations of what you've implemented in person.