CAP 6415: Computer Vision Assignment 4

Due Date: Sunday 03/15/2020 **11:59pm**
Submission Type: Upload onto Canvas

Problem Statement:

Construct a **three-dimensional** (not planar) calibration pattern with known 3D locations. It is not easy to get high accuracy, but you can get close using heavy plywood and printed patterns.

Find the corners, e.g, using a line finder and intersecting the lines. You may use OpenCV or other code to accomplish this. Remember to acknowledge in your report if you do so.

Find code that implements Roger Tsai's calibration method.
R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," in *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323-344, August 1987.

Take two views of the calibration object and estimate the intrinsic and extrinsic parameters using the Tsai calibration code.

Using the estimated camera parameters, write triangulation code to estimate 3D locations from 2D image points in the two camera views. Test your triangulation accuracy.

(Extra Credit - 100% of earned grades if you implement the Tsai calibration code too.)

Submission Requirements:

Please upload a ZIP file containing the following files:

1. All your code files, including any helper files/dependencies.
2. A README file detailing how to run your code along with any compilation instructions.
3. A 2-Page technical report containing the following sections:
    a. A short description of the algorithm
    b. A description of any code/algorithms that were used/re-used by you for your implementation.
    c. A few examples of results from your implementations, comparison with the original implementation (if needed).
    d. A general discussion of lessons learned based on your experiments with the algorithm. E.g. What did you struggle with, issues faced while implementing the code, scopes for and/or proposed improvements, etc.
4. Your code will be tested on different test inputs and graded based on the progress of your approach on these test inputs. A demo session will be scheduled for evaluating your implementation.

Grading:

Each assignment will be graded out of 100:
Code (out of 30) quality of coding, readability, understandability (comments, variable names, etc.)
Report (out of 40), point 3a, 3b, 3c, and 3d above
Demo and Results (out of 30) Compilations, execution, correctness of results on test cases,

Solutions to your **programming assignments** have to be self-sufficient and **not dependent on other computer vision code, such OpenCV or Matlab vision package**. You may use packages for display graphics or mathematics packages, such as for linear algebra (numpy, for example, matlab (but not computer vision module)) or graphs or optimization.

All reuse of code has to be clearly acknowledged in the source code, any README files, and also in the report. Failure to do so will be considered plagiarism.