CSE 483 / ECE 483: Mobile Robotics

Assignment 4: Multi-view triangulation and Non-linear optimization

General Instructions:

1. Read this document very carefully!
2. The purpose of this assignment is to equip the students with implementation skills. It is not enough to know the theory, one needs to implement it to gain a better understanding of the topic.
3. **Plagiarism** is strictly prohibited. Any instance of plagiarism will resort to a score of zero over this assignment (irrespective of the scale and nature of plagiarism).

---

1. **Multi-view triangulation:** In this part of the assignment, you will be reconstructing a synthetic cube (made up of 56 points) using multi-view triangulation. Multi-view triangulation is a straight forward extension of 2-view triangulation which you have already coded in the previous assignment. Similar to the 2-view triangulation, you will be using the projection matrices of all the 8 views and setup a least square system of the form Ax = b and then solve it using SVD. For example 3D point X3 must satisfy the following constraints P1*X3 = x13, P2*X3 = x23, ..., P8*X3 = x83, where x13 denotes the 2D projection of X3 in image 1, x23 denotes the 2D projection of X3 in image 2,..., x83 denotes the 2D projection of X3 in image 8.

   The images of the synthetic cube is provided to you in the form of a $8 \times 2 \times 56$ ($numOfViews \times 2 \times 56$) tensor named `cube_imgs.mat`. And the corresponding projection matrices are provided as 8×1 ($numOfViews \times 1$) MATLAB cell array named `projMatrices.mat`; Both stored in **MatFilesQues1** folder. **Note: Be wary of the fact that the name of the matrices might change after loading them into the MATLAB workspace**

   Using the matrices:

   - **Extracting 2D points of some view (one of the eight views) in MATLAB:**

     ```
     % Once you have loaded the cube_imgs.mat, Lets say you want to extract the
     % 2D points of the cube in View-3.
     pts2D_view3 = squeeze(image_pts(3,:,:));
     ```

   - **Extracting projection matrix of some view (one of the eight views) in MATLAB:**

     ```
     % Once you have loaded the projMatrices.mat, Lets say you want to extract the
     % Projection matrix corresponding to the View-3.
     projMat_view3 = projMatrices{3};  % mind you, these are braces and not parenthesis.
     ```

   The output should be as shown in the figure below (you should also provide appropriate title and axis labels):

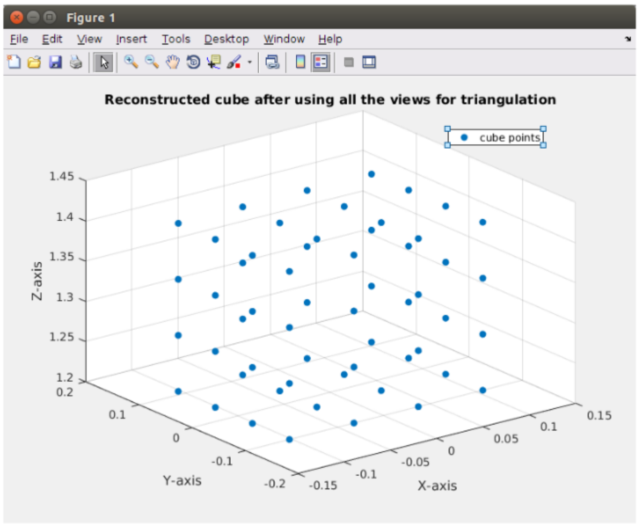
Figure 1
File Edit View Insert Tools Desktop Window Help
Reconstructed cube after using all the views for triangulation
cube points
Z-axis
1.45
1.4
1.35
1.3
1.25
1.2
0.2
0.1
0
-0.1
-0.2
Y-axis
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
X-axis

2. **Levenberg-Marquardt (LM) Algorithm for non-linear least square:**In this section, you will implement ONLY the trust-region strategy of Levenberg-Marquardt (LM) method for non-linear least square problems. You are given a set of scripts. Your task is to go through the files and understand them, especially `testGaussNewton.m`. Then, you need to look at the file `testLevenbergMarquardt.m` and complete the trust region part for that. The `testLevenbergMarquardt.m` script has a few **"CODE HERE"** blocks that you need to fill our for the code to work as expected.

   The codes for this part are placed in the **CodeLM** folder inside the assignment folder. Edit the codes as required and let them be in the same folder while submitting.