ASSIGNMENT 1

MTRN2500 Computing for Mechatronic Engineers - T3 2019

Online Submission Due Date: 11:55pm, Sunday 20th October 2019
In-class Assessments: Your tutorial in Weeks 4 and 5
Weighting of final course mark: 20%

1 Learning Outcomes

This assignment specifically targets the following learning outcome:

- To develop a fundamental knowledge of the Matlab framework and develop a skill to choose and use Matlab tools to solve problems in Mechatronic engineering.

2 Assignment Specification

This assignment is split into three components: a style component and two in-class assessments. This document will explain what you are required to do, the dates for each part, and the contributions each part will make to your overall assignment mark. Total marks available for this assignment are 20. You will complete this assignment individually.

In this assignment, you will process imagery provided by an RGB-D camera; which was mounted on a hexapod platform. The platform operated in a cluttered context (populated by certain obstacles). Your mission is to detect certain objects ("points of interest") when those are located (relative to the robot) at positions which belong to a certain Region Of Interest (ROI). For that purpose, you will implement a program which will offer defined processing and GUI (Graphic User Interface) capabilities. Your program will use data in an off-line fashion. The sets of images were acquired using a hexapod robot and on-board sensors, including an RGB-D camera. However, your program will perform the processing in a periodic fashion, and at a specified processing rate; for emulating real-time conditions.

- Style Component (5 marks)
  - This is due Sunday 20th October 2019 at 11:55pm by submission to Moodle (see Section 2.1 for more information).
- In-class Assessment Part 1 (5 marks): Display 3D Point Cloud
  - See Section 3 below. This will be demonstrated to your tutor during your week 4 tutorial.
- In-class Assessment Part 2 (10 marks): GUI and OOI
  - See Section 4 below. This will be demonstrated to your tutor during your week 5 tutorial.

2.1 Style component (5 marks)

For the style component you are required to submit all your .m files. All these files must be located in a folder named with your full zID (i.e. z1234567) then zipped and submitted to Moodle by 11:55pm on Sunday 20th October 2019.

A submission box in Moodle will be open for you to submit the .zip file. Feedback on this will be provided within 2 weeks of submission.

Things we will be looking for include:

- (1 mark) Consistent and neat structure
- (1 mark) Choice of names for variables
- (1 mark) Commenting and readability of the code
- (2 marks) Modularity: is the program broken up into well defined files and functions?

3 In-class Assessment Part 1 (5 marks): Display 3D Point Cloud

1. Receive (as an input argument) the name of the folder in which the data files are located. These files have well known file names (always the same names, independently of the folder in which they are located in your computer). The images are contained in files (MAT format). The format of the data files will be inferred, by the students, from the provided examples (which include useful comments, in addition to the example source code).
2. Read the files and sequentially play through the sequence of images. Show those images in a figure (in two subfigures, one for Depth and another for RGB), using Matlab graphic handles. See the provided example. Add some basic GUI capabilities for controlling the playback sessions (i.e. buttons for ending and pausing/continuing the session). Note: The loop should perform an iteration approximately every 100ms. You may use other values, provided that the images and results can be seen by the tutors during your demonstration.
3. Process the depth images, to produce the associated 3D points.
   (a) Obtain the associated 3D points, for all the pixels whose depth is not zero (pixels whose depth is zero are assumed to be faulty, i.e. the camera was unable to estimate the associated depths).
   (b) Show the 3D point cloud in a separate figure (use function plot3( ) and the obtained graphic handle, for subsequent updates). (Note: For calculating the 3D points, use the formula provided in Appendix 1.)
   (c) Rotate the 3D points for compensating the inclination (roll and pitch) of the camera. The camera had, during the experiment, a pitch of about 10 degrees (towards the floor, with respect to the horizontal plane) and almost zero roll. For doing the required transformation you may apply a simple 3D rotation of the 3D points. After applying this transformation, using proper angles, the walls and many objects will appear to be vertical; the floor, when detected, will appear horizontal.
   (d) Compensate the point cloud for the platform's height (about 20 cm, over the ground). Doing this, the floor would approximately appear at z=0 (if proper aligning rotation was applied as well).

3.1 Assessment for Part 1

Demonstrate the tasks below before the end of your tutorial in week 4 using the lab computers.

1. (1 mark) Load and iterate through sample RGB-D data as images for both RGB and depth.
2. (2 marks) Display 3D points in a plot, not including any points with zero depth.
3. (1 mark) Display rotated 3D points so that the floor and walls appear horizontal and vertical.
4. (1 mark) Display 3D points in a plot with 20cm vertical offset, so floor appears at z=0.

4 In-class Assessment Part 2 (10 marks): GUI and filtering

1. Removal of points of no interest: All the points whose z coordinate is lower than -5cm or higher than 1m should be ignored. The remaining points are termed "useful points".

2. Filtering useful points. They have a position which are inside in a vertical cylindrical ring centred at the origin (of user defined inner and outer radii), and are higher than 15cm in Z. Those points are then to be shown in red, while the rest of the points are shown in blue.

3. Additional GUI capabilities: Include, in your GUI, controls for the following capabilities.

   (a) One checkbox for enabling/disabling angle correction for alignment of the 3D point cloud.

   (b) One checkbox for enabling/disabling the detection of useful points.

   (c) A control, such as a button or a text box or a slider, for allowing the user to specify the desired roll and pitch correction angles. Valid angles are in the range [-45, +45] degrees. Angles are specified by the user in degrees, independently of the angular units/convention you use internally in your program.

In order to obtain full marks, you will need to use graphics handles for updated plots and the program should refresh the display at approximately 10Hz on the lab PCs. Note: A proper program would behave like the one shown in this video: `https://youtu.be/FHrdN7HzSsQ`.

Read the example code on Moodle prior to trying to solve this project.

4.1 Assessment for Part 2

During the week 5 tutorial you will need to demonstrate your code running on the lab computers.

Marking criteria

- (1 mark) Making the program display the points from part 1 at a rate of at least 10 Hz.
- (1 mark) Correctly removing the points of no interest.
- (2 marks) Correctly filtering useful points.
- (1 mark) Implementing a checkbox for enabling/disabling angle correction.
- (1 mark) Implementing a checkbox for enabling/disabling detection of useful points.
- (4 marks) Implementing GUI control of roll and pitch correction angles.

5 Additional Information

- A plagiarism check will be performed on all assignments and any instances of plagiarism will be dealt with under the UNSW plagiarism policy (linked from the course outline).
- Late submissions are not accepted, as per the course outline, unless Special Consideration has been granted. Refer to the course outline for how to do this if you are unsure.
- If you miss your in-class assessment, you are not permitted to join another class and must apply for Special Considerations to sit another.
- Please post questions about this assignment on MS Teams.
- Finally, enjoy learning how to work with Matlab.

6 Appendix 1: Conversion from Depth to 3D

The conversion from Depth to 3D points, is approximated by a function of the pixel's depth and its subscripts (row and column number). Given a pixel whose position is (r,c), and whose associated range is Depth(r,c), then its associated 3D point is approximated as:

$$\boldsymbol{x(r,c) = +Depth(r,c)}$$

$$\boldsymbol{y(r,c) = +Depth(r,c) \cdot (c - 80) \cdot \frac{4}{594}}$$

$$\boldsymbol{z(r,c) = -Depth(r,c) \cdot (r - 60) \cdot \frac{4}{592}}$$

These calibration parameters are not optimal, but are close enough for our purposes.

7 Appendix 2: Axes Convention

$\boldsymbol{X > 0}$: Forward. $\boldsymbol{Z > 0}$: Up; $\boldsymbol{Y > 0}$: Left.

8 Appendix 3: 3D Angle Convention

Please, see document "MTRN2500 Appendix 3.pdf".