AE640A: Autonomous Navigation (Spring'22)
Assignment #2: LiDAR Visualisation and ICP

Due Date: 17/02/2022 Maximum Marks: 100

Instructions

- **This is a coding assignment.** Boilerplate templates have been provided in MATLAB, with clear instructions. The code should be clearly written, properly commented and submitted with a README file to execute the code. It should also be accompanied by a report explaining the results.
- **MATLAB:**
  (i) Submit a single .zip file for this assignment with every question in a different directory.
  (ii) The name of the top-level directory should be your roll number.
  (iii) Include a README txt file in the .zip file. It should mention which scripts to run to generate the desired results for each question.
- The use of any unfair means such as plagiarism by any student would be severely punished. For details regarding the mode of submission and penalties imposed for late submissions, compilation errors, etc., refer to this document.
- Contact the course staff ae640a@gmail.com for any further queries.

Task 1 (50 points)

1. [MATLAB problem] Projection of 3D lidar point clouds onto the corresponding camera images.

   In this problem you are given data from a 360 degree field of view lidar and an omni-directional camera both mounted on top of an autonomous vehicle. The coordinate system associated with each of the system components is given in Fig 1. You have to write a code to project the 3D points obtained in the lidar reference frame onto the corresponding camera images. You will assume a standard pin-hole camera model for projection of points into the images. The attached *problem.mat* file contains the following:

   1. *Problem.scan:* A [N × 3] matrix of 3D points in lidar reference frame [P_L]
   2. *Problem.Image(i).I:* A structure of images from each camera. Total five images corresponding to 5 horizontally located cameras of the omni-directional camera system.
   3. *Problem.X_{hc}(i).X_{hc}:* This is the pose of the i^{th} camera w.r.t. the omnidirectional camera system's reference frame [H].
   4. *Problem.K(i).K:* This is the intrinsic matrix for the i^{th} camera. Once you know the points in the camera reference system you can now project them onto the image frame using this matrix.
   5. X_{hl} : This is the pose of the laser [L] in the omnidirectional camera's reference system [H].

Note: All pose vectors are [6 × 1], first three elements are translation in "m" and last three are rotation angles in "degrees". You will have to convert the angles into "radians". Use rotation matrix $R = R_z R_y R_x$

So you will first find the rigid body transformation that projects the lidar points [P_L] to camera reference system [P_{Ci}]. This rigid body transformation is given by:

$$X_{C_i L} = \ominus X_{HC_i} \oplus X_{HL}$$

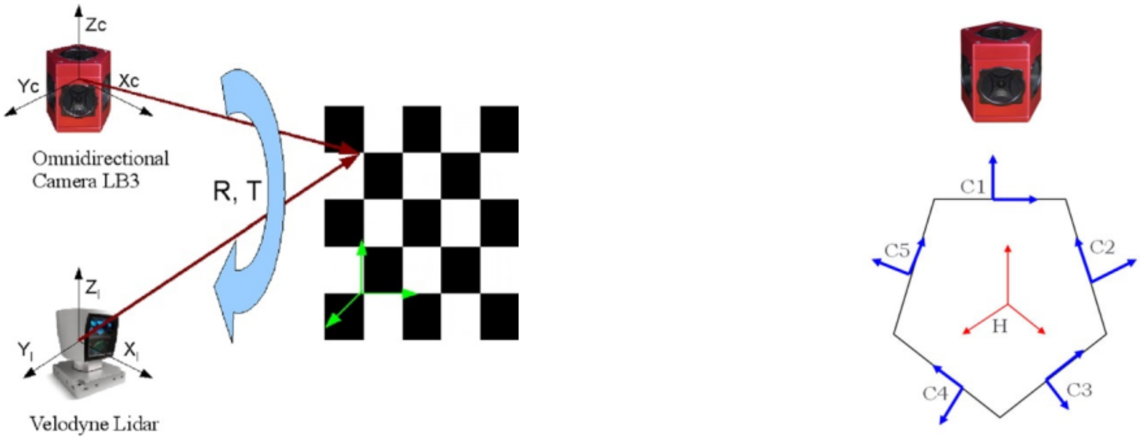

Figure 1: A depiction of lidar and camera coordintate system. Lidar has its own reference frame [L], the omnidirectional-camera system has the reference frame [H] and each camera of the omni-directional camera system has its own reference system [C_i]

So you will have to write the code for this tail-2-tail transformation. After doing projection of points from the lidar to the camera coordinate system, consider only points that are in front of the camera and points that are above the ground plane (Approximately around 2-2.5m).Transform only those points from camera to pixel coordinate system. After doing the projection the expected result would be as shown below in Fig 2. The result images shown below are rotated, if you see the original images given in Problem.mat file then you will find that they are rotated by 90 degrees.What is given in problem.mat is actually the way images are captured by the omni-directional camera and that is mainly to increase the field of view.

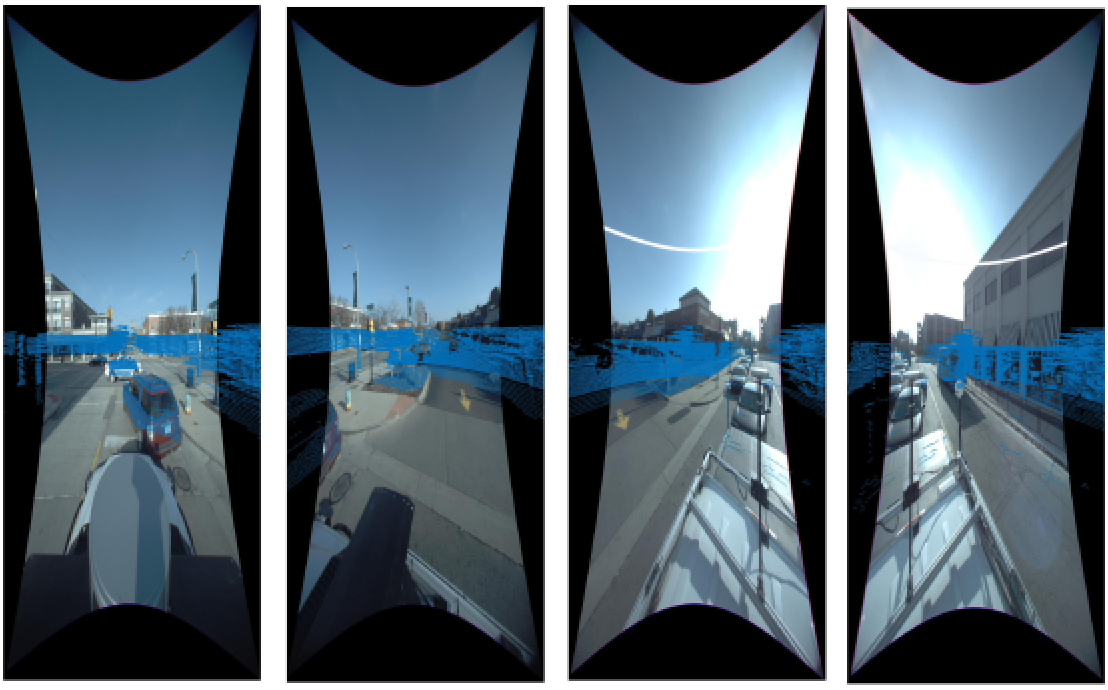

Figure 2: 3D points projected on the images

Task 2 (50 points)

1. [MATLAB problem] Laser Scan Matching using Iterative Closest point (ICP) algorithm.

Set up : Assume that you have a mobile robot mounted with a Lidar in such a way that the axis of robot is aligned with the axis of the lidar sensor.

Data given : You are given pre-processed raw scans that are recorded at two different positions of the mobile robot in an indoor environment that is completely simple and static.

All you have to do is to register those two scans and hence find the relative transform between the poses. Also report the limitations of the standard ICP algorithm.

FYI :

• Read the comments in the **main.m** file to understand the complete flow.
• **lidar_scans.m** file given to you contains two scans named scan1 and scan2. Each scan has approx 680 parameters.
• You need not to code for pre-processing of raw scans, for your ease it is done and hence you have the cartesian coordinates of each scan point.
• You need not to worry about visualisation. Your job is to fill up the **ICP.m** function.
• Most importantly, ICP eliminates outlier correspondences by setting a maximum distance threshold.

Note that proper tuning of this parameter is required for better registration.