Probabilistic Mobile Robotics - EE698G

Assignment- 2

- Due Date: **February 5, 2017, 11:59pm**
- Late submission penalty is 20% for every late day.
- Please submit MATLAB codes to moodle. You can submit other solutions by
  - (i) **(Recommended)** Typing it in latex and uploading the PDF to moodle along with codes.
  - (ii) Dropping handwritten solutions in the mailbox.
  - (iii) Creating a PDF by scanning handwritten documents and uploading it to moodle – should be clear.
- **MATLAB**
  - (i) Submit a single .zip file for this assignment with every question in different directory.
  - (ii) The name of the top-level directory should be your roll number.
  - (iii) Include a README textfile in the .zip file. It should mention which scripts to run to generate the desired results for each question.

Q.1 [MATLAB problem] Projection of 3D lidar point cloud 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 asssume 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 $[N \times 3]$ 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 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 $[6 \times 1]$, 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 $[P_{C_i}]$. This rigid body transformation is given by:

   $$X_{C_iL} = \ominus X_{HC_i} \oplus X_{HL}$$

   So you will have to write the code for this tail-2-tail transformation (see the lecture slides). After doing projection of points from lidar to camera coordinate system, consider only points that are in front of camera and points that are above ground plane (Approximately around 2-2.5m). Transform only those points from camera to pixel coordinate system. After doing projection the expected result would be like 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. [20 Marks]

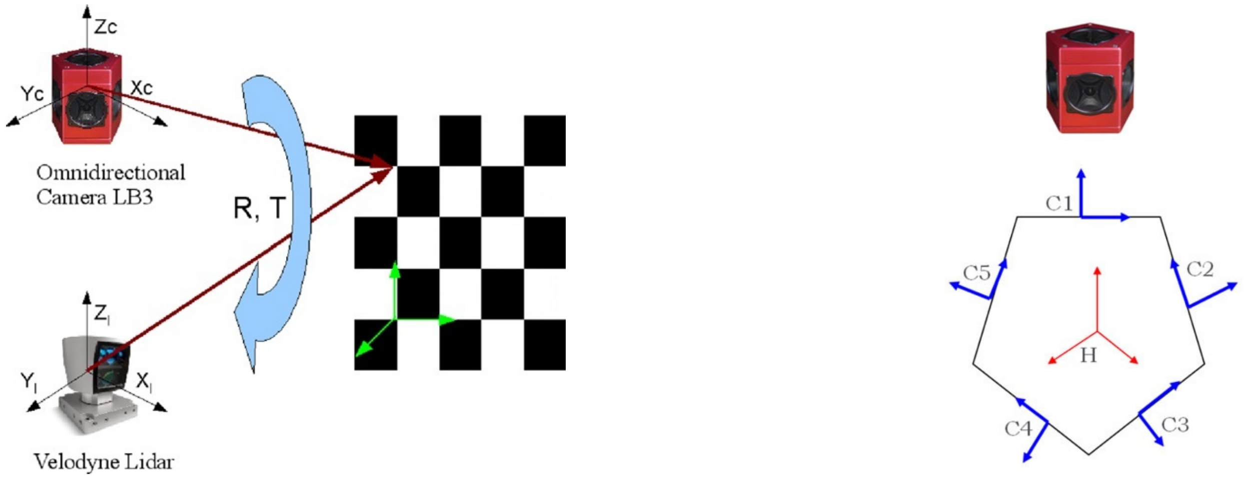

Figure 1: A depiction of lidar and camera coordintate system. Lidar has its own reference frame [L], the omnidirectional-camera system has the reference frame [H] and each camera of the omni-directional camera system has its own reference system $[C_i]$

Q.2 [MATLAB problem] Construct a symmetric matrix $R \in [2 \times 2]$ that has eigenvectors $\overrightarrow{x_1}$ & $\overrightarrow{x_2}$

$$\overrightarrow{x_1} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}; \qquad \overrightarrow{x_2} = \begin{bmatrix} -5 \\ 1 \end{bmatrix}$$

with corresponding eigenvalue $\lambda_1 = 10$,$\lambda_2 = 2$

(a). Generate and plot 200 points of zero mean Gaussian data that have the covariance R.

(b). Estimate the covariance of generated data, and compute the principal components of the data, and plot them. Do they match with the original eigen vectors ?

Note : Use **randn** function to generate Gaussian samples. Also recall eigen value decomposition of covariance matrices discussed in the class. In MATLAB you can use $eig()$ function to compute the eigen values and eigen vectors. [10 Marks]

Q.3 Assume that polar co-ordinates, r and θ of an object's location are independent and normally distributed, i.e $N(\mu, \Sigma)$, where $\mu = [\mu_\Theta, \mu_R]^T$ and $\Sigma = \begin{pmatrix} \sigma_\Theta^2 & 0 \\ 0 & \sigma_R^2 \end{pmatrix}$. Polar co-ordinates can be converted to euclidean co-ordinates after applying a suitable transformation.

1.a.1 Will the euclidean co-ordinates of the points also be normally distributed ?

1.a.2 Find the joint probability distribution of euclidean co-ordinates (using Jacobian).

1.a.3 Calculate the approximated covariance matrix of euclidean co-ordinates by the method of linearization.

[MATLAB] data.mat is attached with assignment and contains n points in polar co-ordinates following the afore-mentioned distribution. The first column corresponds to theta axis and the second column corresponds to ρ with $\mu = [1, 3]^T$, $\sigma_R^2 = 1$ and $\sigma_\Theta^2 = 0.5$. Let the mean and covariance of point in euclidean co-ordinates be μ_{euc} and Σ_{euc}.

1.b.1 Plot the points taking r and θ as independent axes. Visualize the 3-σ error ellipse using Σ in the same plot (see Hint).

1.b.2 Transform the points to euclidean co-ordinates (x $= r\cos(\theta)$ and y $= r\sin(\theta)$) and plot the points in the euclidean space. Calculate the sample covariance matrix of euclidean co-ordinates and call it Σ_{sam}. Visualize the 3-σ error ellipse using Σ_{sam} in the same plot (see Hint).

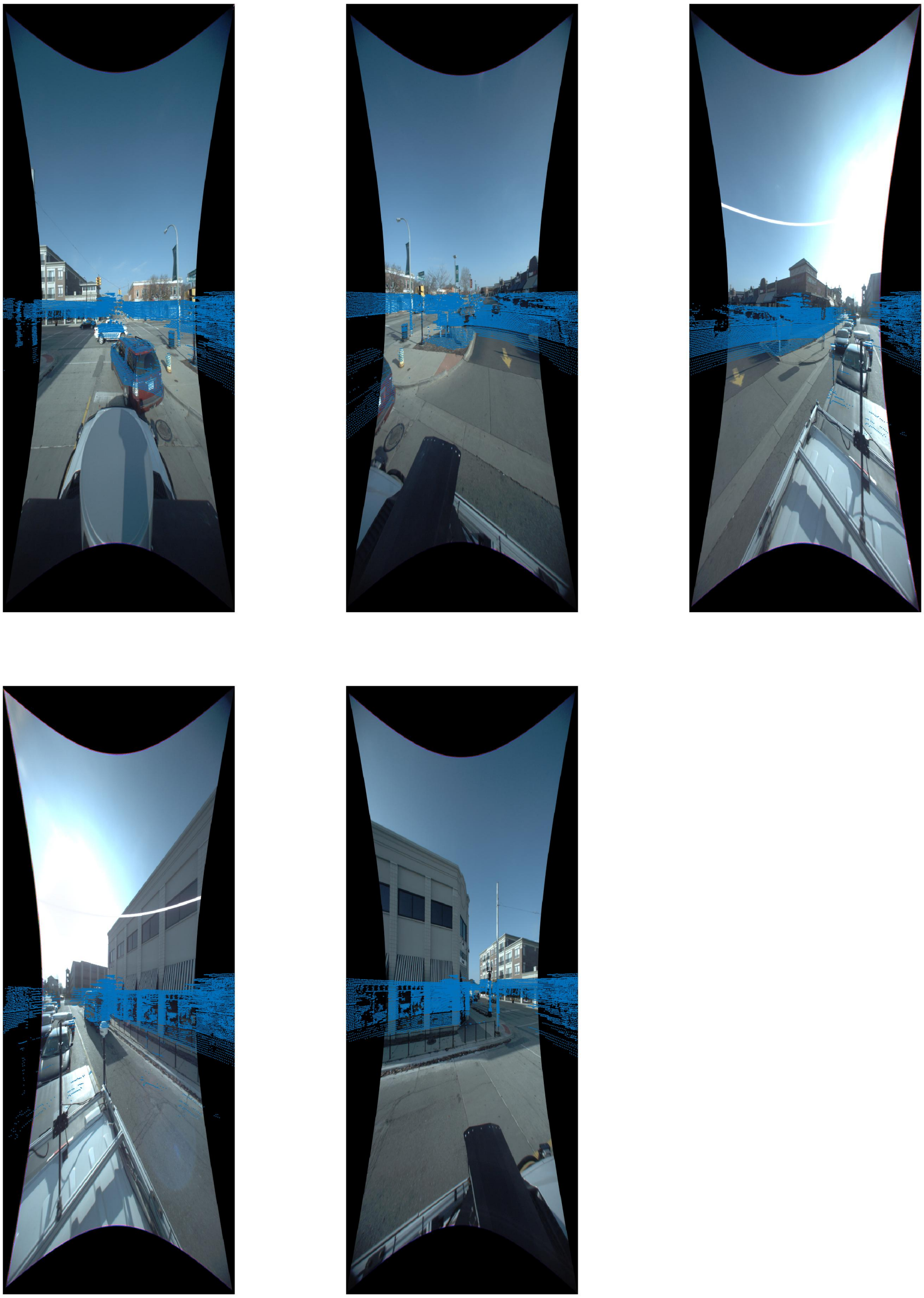

Figure 2: 3D points projected on the images

1.b.3 Calculate the approximated covariance of euclidean coordinates by linearization using Jacobian and given parameters (essentially, perform error propagation) and call it Σ_{lin}. Again, visualize the 3-σ error ellipse using Σ_{lin} in the same plot (see Hint)

1.b.4 Draw inferences from the plot and comment on relative errors Σ_{lin} and Σ_{sam} .

Hint : 3-sigma error ellipse is 99.7% confidence region that represents the iso-contour of the Gaussian distribution. For multivariate gaussian of dimension k, $(x-\mu)^T\Sigma^{-1}(x-\mu) \leq \chi_k^2(p)$ denotes the region which will contain the 'p' probability and $\chi_k^2(p)$ is the quantile function for χ^2 distribution with degree of freedom k. For k = 2, taking $\chi_k^2(p) = 3$ amounts to p $\approx$ 0.99.

[20 Marks]

Q.4 [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 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 approximately 680 measurements.
- 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 visualization. Your job is to fill up **ICP.m** function.
- Most importantly ICP eliminates outlier correspondences by setting a maximum distance treshold. Note that proper tuning of this parameter is required for better registration.

[10 Marks]

Q.5 Verify that the transformation matrix has the following relationship : $\mathbf{H}_{ij} = inv(\mathbf{H}_{ji})$, where $\mathbf{H}_{ij}$ represents the transformation from j^{th} frame to i^{th} frame. [5 Marks]