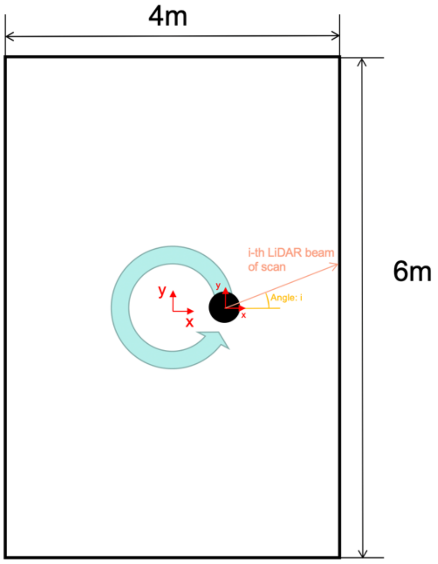
CS284: Simultaneous Localization and Mapping

Homework 1, Fall Semester 2021

- Contact: lkneip@shanghaitech.edu.cn
- TA: cuili@shanghaitech.edu.cn
- Deadline: **23:59, 6th of October 2021** (please submit a zip via email to TA)
- **Please reread our policies on academic integrity. It is a very serious matter, and any violations will be prosecuted to the fullest extend.**

Task Description

Homework 1 consists of implementing the ICP algorithm and applying it to a simulation dataset. The setup is simple: We have a small robot moving in a rectangular room of **4x6m**. The robot carries a 2D single beam Lidar that is mounted perfectly horizontally on the robot. The Lidar therefore simply measures points on the four walls. The robot moves two full loops on a circular trajectory in the room which is centered around the room's center and has a radius of 1m. The robot's forward direction always stays tangential to the circular trajectory. The rotational velocity of the robot about the vertical axis is **10°/s** and you may assume that the Lidar scans **5fps** frames per second. You may furthermore ignore the fact that the Lidar measures data over time and may assume that all depths in all directions are captured instantaneously in the very same moment. The angular resolution of the Lidar is **1°**. As a result, the robot captures a total of 360 scans over the course of the two loops, and 360 depth values for each scan. The data is given to you in file "**data.txt**" (a second file is "**data_noisy.txt**", where we added Gaussian noise, details provided below), which simply contains a matrix with the size of 360 * 360. The values in each row represent the depth reading of one frame, and column *i* contains the depth measurement when the beam is at the *i*-th angle. Figure 1 may help you understand the scenario described.



System parameters:

- Room Width: 4m
- Room Length: 6m
- Trajectory radius: 1m
- Vehicle stays tangential
- Rotation Speed: 10°/s
- Frame Rate : 5Hz
- Angular Resolution: 1°
- Total Loops: 2

Your task is to find the transformation between between consecutive frames through the ICP algorithm. The obtained relative pose estimations may then be concatenated in order to recompute the entire robot trajectory (see the simplified SLAM algorithm introduced during the initial lectures).

The detailed steps are as follows:

- Implement the correspondence search. What could be a straightforward way to speed up the correspondence search given that the data is coming in a very structured form. Do you have to perform outlier filtering in our case? Explain why.

- Calculate the rotation and translation that relates two consecutive point clouds using the method taught in class.

- Iterate over the above steps each time taking the already identified relative transformation as an initial transformation to be applied before the correspondence search.

- Propose suitable termination criteria. Verify the correctness of your result.

- **Bonus questions**: Compare the influence of noisy data on the ICP algorithm, what other factors will affect the results of ICP?

Submission

Please submit a PDF report (not more than 2 pages!) along with your code to the TA email address before the deadline (put professor in CC). Afterwards, arrange for a time with the TA to demonstrate your code. Your code and results must be the same than what you submitted in the email. Your code must be written in either Python or C++. Your report must include:

- A description of your implementation. This should cover a basic description of the algorithm. It should also include the physical structure of the program (i.e. file description, dependencies), and instructions on how to compile and use it (i.e. how to interpret the result).

- Your observations on the performance of your ICP algorithm, both in terms of efficiency and in terms of accuracy.

Please use ”SLAM HW1 – Your name” as the email-header when sending the email. Please pack all the files into a zip file, the structure should be:

[Your name] (folder)
– Your name.pdf
– [code] (folder)