Information Processing in Robotics

Exercise Sheet 7

Topic: Iterative Closest Point

Exercise 1: Using ICP program

In this exercise we will use the pmicp program to do ICP. pmicp is a tool provided by libpointmatcher, an ICP library. In order to get this library, you can:

$ sudo add-apt-repository ppa:stephane.magnenat/lsb_release -c -s'
$ sudo apt-get update
$ sudo apt-get install libpointmatcher-bin

This pmicp tool takes point clouds (as csv files for instance) as input and computes the transformation matrix between them.

We will use the office_icp package to generate the point clouds.

- Install office_icp and run the office_icp.launch and the keyboard_teleop.launch launch files. Observe what happens to the laser scans in rviz.
- Write a function that, given a LaserScan message\(^1\) as input, dumps the content of the scan in a csv file in which each line is x,y (the coordinates of the point in the sensor coordinate frame).
- Write a function that calls pmicp and parses its result to obtain the displacement.
- Publish this information as a Twist message\(^2\)
- Write a subscriber to the /scan_throttle topic to receive the scans, call pmicp then publish the motion estimate.
- Run your program alongside office_icp in order to observe when ICP works well and when it fails (you can look at the documentation of pmicp and play with the numerous options to improve the result).

\(^1\)It's useful to look at its definition: [http://www.ros.org/doc/api/sensor_msgs/html/msg/LaserScan.html](http://www.ros.org/doc/api/sensor_msgs/html/msg/LaserScan.html)