Registration Challenge

Welcome

Welcome to the GrayMatter Robotics Registration Challenge. This is your first step towards joining GrayMatter Robotics. This README details how you need to work on the challenge and submit it to us.

System Requirements

- Python 3.6.9
- libpointmatcher

Installation

To complete this challenge, you will need to install libpointmatcher along with its Python bindings (We've tested this on Ubuntu 18.04 so we'd recommend that). Instructions for compiling libpointmatcher can be found here and instructions for compiling the python bindings can be found here

Premise

The main objective of this challenge is to perform cross source pointcloud registration thereby, finding the transformation to match one pointcloud, taken from one sensor technology to the another. You will need to use the libpointmatcher library GitHub. You will need to use this library for registration but for any other auxiliary tasks, you can use any library you want. You are given 10 test cases to validate your algorithm before submission. Each of these folder are named according to the object that was scanned. Under each individual folder, you will find two scan that you need to match. You are also given the ground truth transform to debug you code as needed.

Steps

1. Reply to this email confirming that you've received it.
2. Create a PRIVATE repository on your GitHub with the starter code and add marshall@graymatter-robotics.com and rishav@graymatter-robotics.com as collaborators.
3. Install libpointmatcher following the steps detailed in their README.
4. Fill in the TODOs in the main.py script. Note that you can create any class, function, supporting python files etc. for your purposes. We would also evaluate you on your code organization skills.
5. When you've completed the challenge, email us the SHA of your submission. If you do not do this step, we will use the latest commit before the end of the 72-hour deadline as your submission.

Things to keep in mind

- Your code should not take more than 2 mins or 120 seconds to execute.
- Your alogrithm will be tested on several other such scenes and pairs and the error and runtime should not go over a certain threshold.
- The pointclouds in the dataset have different densities, so your alogrithm may need to account for that.