CSCE 452 Robotics and Spatial Intelligence, Spring 2025

Project 3

Overview and purpose

This assignment asks you to create software that processes data from a lidar sensor to detect and track the movement of people around a robot. The purpose is to give some experience designing and implementing algorithms that cope with the complexity of real sensor data, along with exposure to some additional ROS features.

Logistics

This assignment is may be completed in teams of size 1, 2, or 3. Expectations and grading standards will be the same regardless of team size. The instructor will not mediate conflicts between team members.

You should assign yourself to a team within the Canvas system —Use the People link within the course page. If you are choosing to work alone, this step is still required: Simply add yourself to a group with no other members. After this date, teams for this project will be final.

Exactly one member of each team should submit.

Reference material

In addition to the concepts you used in Projects 1 and 2 (nodes, topics, publishers, subscribers, services, parameters, etc.), you'll need to use **launch files** and **visualization markers**. You will likely want to consult the tutorials and reference documentation on those subjects.

Your task

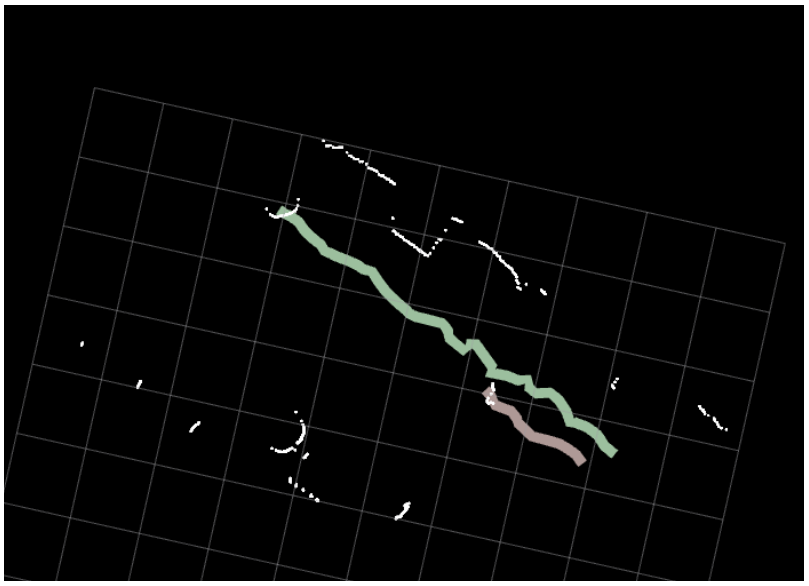
Here is a collection bag files recorded by a robot equipped with a laser range finder (i.e. lidar):

Project 3 Bag Files

When each of these bags was recorded, the robot and the nearby obstacles remained motionless. However, in each case, zero or more **people** walked past the robot.

Your job is create a system to **detect** and **track** the unique people that the robot sees.

Several times per second, your system should publish a message of type `visualization_msgs/MarkerArray` on the topic `/person_markers`. This message should contain one marker for each person visible at the time. The marker type should be `LINE_STRIP`, and the other fields of the marker object should be filled appropriately in to enable rviz to show the history of that person's movements, starting from the place where they first appeared until the present time. An example with two markers appears below.



To keep track of which person is which, the `id` field of each marker should uniquely and persistently identify each person that appears in the scene. If someone is temporarily hidden, when they reappear their marker should have the same `id` as before. The specific id numbers you use are not important, as long as the `id` remains consistent for each person and distinct people are assigned distinct `id`s.

Your system must consist of at least two distinct nodes. One node should subscribe to `/scan` and a different node or nodes should publish the markers. You may use more than two nodes if you wish. You should design your system carefully to make a meaningful subdivision of the work between the nodes in your system. Your nodes should communicate by publishing messages on a topic or topics of your own design. (The idea of this constraint is to practice the ROS philosophy of designing nodes that each do a single, simple job well, and then connecting those parts together to form a complete system.)

Finally, because your system will consist of several distinct components that must be started in concert, you should create a **launch file** that starts the system. This launch file should:

- Have an argument called `bag_in`. The argument will be set on the `roslaunch` command line to the name of a bag file (one of the provided bags) to play.
- Have an argument called `bag_out`. The argument will be set on the `roslaunch` command line to the name of the bag file to record all published messages to.
- Start `ros2 bag play` to play the provided sensor data, `ros2 bag record` to capture all topics, including the output for your system, along with all of the nodes needed for your system.
- Terminate all of the other nodes when `rosbag play` completes.

Your solution will be evaluated primarily on the accuracy of its results. The quality of the system design and the code will also play roles in the grading.

What to submit

Three required parts:

1. A PDF report including:
    a. A concise description of how the system works. This should be a couple of paragraphs, describing your overall approach at a conceptual level. Some questions to consider: What choices did you make in designing the program, and why? What nodes does your system contain, and what role does each one play? What specific parameters are part of your method, and how did you select values for those parameters?
    b. A short answer to these questions: Did the results meet your expectations? Why or why not?
2. The source code for your programs. This will span at least two source code files. Be sure to consult the source code style guide on the course Canvas page.
3. Rosbag recordings of **all topics** throughout the **entire execution** of your program for **each** of the provided input bags. The directory of the bag file must be named with your two digit group number for this project, followed by a dash, followed by the number of the input bag file. For example, if you are in group Group 3, your submission

should include 9 bag directories, called `03-01`, `03-02`, ..., `03-10`. Incorrectly named bags will not be graded.

It is essential for these bags to contain both the original sensor data on `/scan` as well as the outputs on `/person_markers` because this information will play a primary role in evaluating the correctness of your solution. We cannot award credit for correctly detecting and tracking people unless you submit these outputs correctly.

One optional part:

1. A short video demonstrating your system in some way that may be helpful in grading.

[Submit via Canvas.]