# Robot Localization

Nate Sampo, MJ McMillen

October 9, 2018

Bob ROS

## 1 Overview

The goal of this project was to learn how to implement a particle filter. The particle filter helps a robot find its location using the probabilistic selection of potential robot locations. The particle filter begins by laying a set of particles randomly across a map of the room, and will gradually cluster the particles around those with higher probabilities through random weighted selection, hoping to eventually land on the robot's actual position. We wanted to push ourselves and be able to solve the kidnapped robot problem, meaning we do not know where the robot begins within the room and will have to locate the robot using our particle filter.

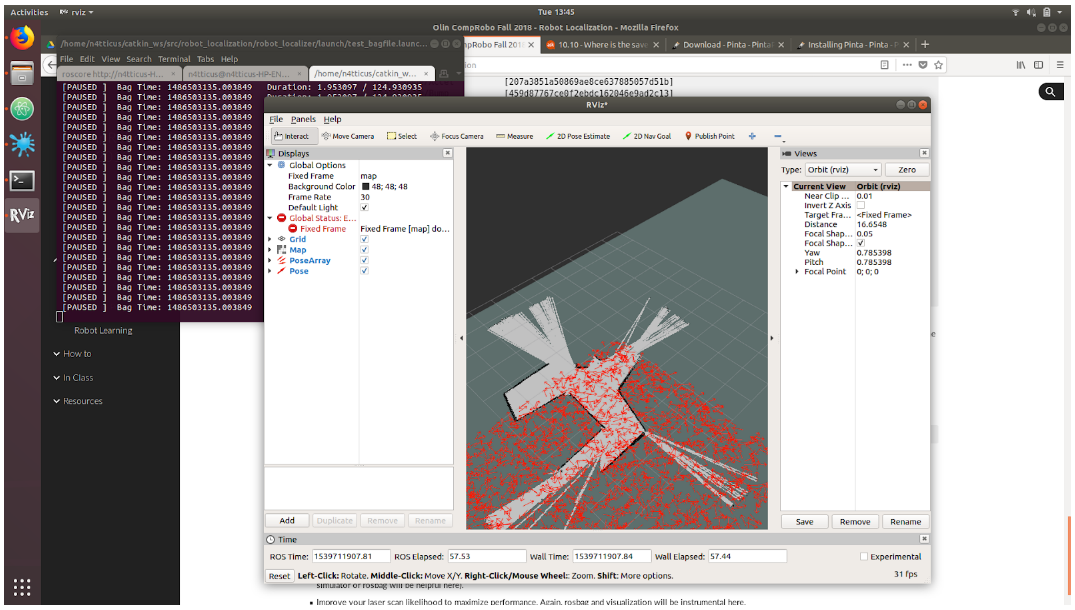
Our particle filter executed these steps:

1. Robot translates to new position
2. Get new laser scan data from new position
3. Translate all current particles to match the robot's translation
4. Update the probabilities of all of the particles based on the laser scan data
5. Trim particles through random weighted selection
6. Generate new particles around the kept particles with added noise
7. Repeat steps 1-6

## 2 Implementation

We implemented the particle filter by creating a series of interconnected classes, each with a series of well defined and simple functions. The goal of these classes was to separate out the functionality of the particle filter into many discrete chunks for more easily readable and debuggable code. We ended up writing four classes: a sensor manager, a particle manager, a particle class, and a general control class, and were initially provided with two classes: a helper function container and the occupancy field class. Each class had its own unique functionality and purpose, and this style of implementation helped our project tremendously.

Particles are initially placed randomly within the room and trimmed to remove particles which are unreasonably far away from obstacles as seen in the occupancy field. We then use the laser scan to scan the area around the robot and calculate the distance to the nearest obstacle for each angular increment. Next, we project new points around the randomized particles at angles matching the angular increments the robot's laser scan data gathered. We then calculate the distance to the nearest obstacle for those points. The smaller the difference between the distances found for the particles and the distances found by the robot, the greater the chance that the particle is in the right area and with the right rotation. After assessing the probability for each point, we trim the particle list. Our trimming algorithm randomly selects a subset of particles from a weighted list of current particles. The particles with a higher probability have a greater chance at surviving the culling. After weeding out the improbable points, we scatter new points around the remaining points. The robot then moves, the points are translated, and the whole process is repeated until the probability becomes extremely high for an extended number of moves.


Activities
rviz
Tue 13:45
Olin CompRobo Fall 2018 - Robot Localization - Mozilla Firefox
/home/n4tticus/catkin_ws/src/robot_localization/robot_localizer/launch/test_bagfile.launc...
File Edit View Search Terminal Tabs Help
roscore http://n4tticus-H...
n4tticus@n4tticus-HP-EN...
/home/n4tticus/catkin_w...
10.10 - Where is the save
Download - Pinta - Pinta
Installing Pinta - Pinta
RViz*
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Views
Add
Duplicate
Remove
Rename
Save
Remove
Rename
Time
Reset
Robot Learning
How to
In Class
Resources

```
[PAUSED ]  Bag Time: 1486503135.003849   Duration: 1.953097 / 124.930935
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
[PAUSED ]  Bag Time: 1486503135.003849
```

```
[207a3851a50869ae8ce637885057d51b]
[459d87767ce0f2ebdc162046e9ad2c13]
```

| | |
|---|---|
| Global Options | |
| Fixed Frame | map |
| Background Color | ⬛ 48; 48; 48 |
| Frame Rate | 30 |
| Default Light | ✓ |
| Global Status: E... | |
| Fixed Frame | Fixed Frame [map] do... |
| Grid | ✓ |
| Map | ✓ |
| PoseArray | ✓ |
| Pose | ✓ |

Type: Orbit (rviz) Zero

| Current View | Orbit (rviz) |
|---|---|
| Near Clip ... | 0.01 |
| Invert Z Axis | ☐ |
| Target Fra... | <Fixed Frame> |
| Distance | 16.6548 |
| Focal Shap... | 0.05 |
| Focal Shap... | ✓ |
| Yaw | 0.785398 |
| Pitch | 0.785398 |
| Focal Point | 0; 0; 0 |

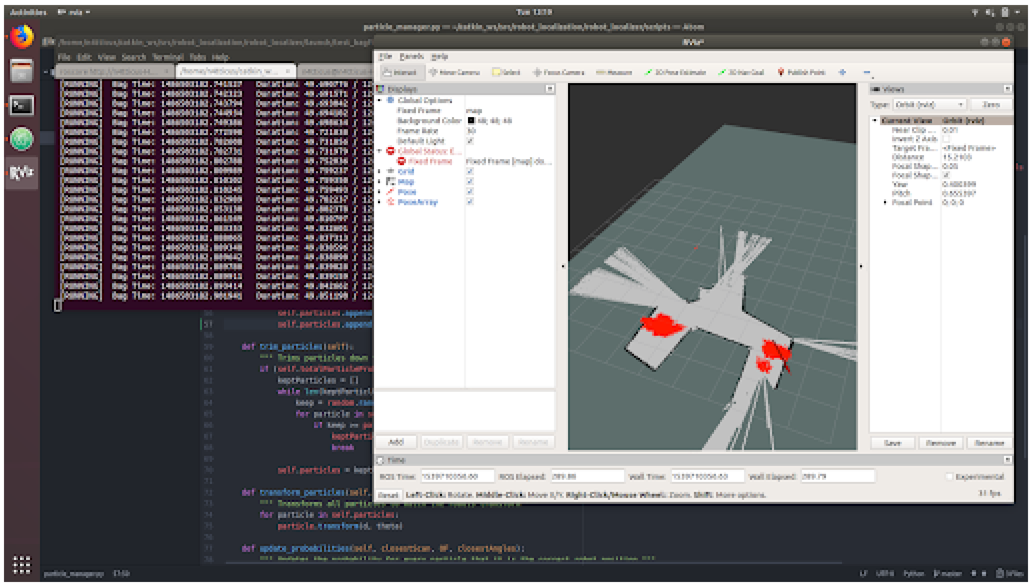
ROS Time: 1539711907.81 ROS Elapsed: 57.53 Wall Time: 1539711907.84 Wall Elapsed: 57.44 ☐ Experimental

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options. 31 fps

simulator or rosbag will be helpful here).

- Improve your laser scan likelihood to maximize performance. Again, rosbag and visualization will be instrumental here.

Initial particle scattering as seen in Rviz


Particles converging on two likely robot positions

## 3 Design Decisions

Throughout our project, we tried to keep an emphasis on clean, easily readable code. We believed that this would make the project much more manageable, and would help us write better code in the future. To this end, we believe our implementation was relatively slick, and the pieces all worked together well. Having many distinct classes that worked together well allowed us to work on different parts of the project with ease, quickly find and fix bugs within the program, and make the code much more understandable to any future observers.

Another aspect of the project we were particularly fond of was our decision to attempt the kidnapped robot problem. Rather than beginning all of our particles on the current robot pose and tracking the robot throughout the room, we wanted to find the robot and track it as it moved without knowing its starting location. This proved to be a much more difficult challenge than we anticipated, though we felt we did a solid job of handling the added hurdles.

## 4 Reflections

Our particle filter was not perfect, in fact it had many flaws. Despite our best efforts, we could not get the filter to consistently converge on the true robot position. It often was able to correctly locate the robot and follow its motion, however many times it would converge on two or three likely points, and sometimes lose the robot altogether. Although frustrating at times, we feel that this project has been an excellent learning opportunity, and have both come away feeling much more confident in our abilities.

One of our main takeaways from this project is that object oriented programming makes our code elegant and organized. Our class structure made it easy to keep track of each moving part and made it easy to manipulate whole groups of objects. In the future, we will endeavor to use more object oriented programming principles. We also increased our understanding of rosbags and rviz, and many of the embedded debugging options.

If we were to move forward and continue working on this project, one of the main places where we could have benefited most is with more testing on parameters within our program. Constants like how many particles to keep after each round of trimming, how many total particles to generate, how often to update the laser scan data, how many angles of the laser scan data should we consider, and the like were mostly numbers that we guessed would be within a ballpark of an optimal value.

Overall, we both felt like we learned much from this project, and though we struggled throughout, seeing moderate success was exhilarating.