Ankeet Mutha and Rahil Dedhia
Robot Localization Project

What was the goal of your project?

We were trying to figure out where a robot was if we gave an initial guess of the robot's location and drove around the room for a while.

How did you solve the problem? (Note: this doesn't have to be super-detailed, you should try to explain what you did at a high-level so that others in the class could reasonably understand what you did).

We tried to use a particle filter to figure out where the robot was. We input a guess of our location and put several random particles around the guess. Then as we drove the robot around, the particles would move according to how much the robot moved. We would use the laser scanner of the robot to look at the nearest obstacle the robot was facing and then try and see if which particle if advanced by that distance, would be close to that object. Those particles would be given a larger probability to be reselected next time the robot moved. Eventually the particles should converge upon the robot's location.

Describe a design decision you had to make when working on your project and what you ultimately did (and why)? These design decisions could be particular choices for how you implemented some part of an algorithm or perhaps a decision regarding which of two external packages to use in your project.

Originally, we tried to figure out where our robot was by only looking the nearest object using the laser scan and then looking at our particles nearest object using the occupancy grid. This kind of worked but would get easily confused. We decided pivot to the solution where the particles advanced forward the amount the closest object was and detect where the closest object was from that point. If the number was low, the particle would be more likely. This solution would have been much more accurate if we got it working properly.

What if any challenges did you face along the way?

We had difficulty debugging our code, because we weren't sure what was causing our particles to diverge. It was easy to see using rviz or gazebo where there were issues in our localization, but it was hard to actually know what was causing them.

We also made a critical mistake in only looking at the minimum range across all of the laser scans, and adding that to the front of each of the particles. This caused our particles to diverge more than we wanted them to. We should have looked at the nearest object in the direction our neato saw the closest object.

What would you do to improve your project if you had more time?

There are a few things we could have done to improve our debugging process, including publishing markers to rviz to visualize the weights of each of our particles.

We also had difficulty tuning the parameters on our Gaussian for determining particle weights and on adding noise to our system. We could have also probably added noise in other locations other than after taking laser scans.

We also have not tested our latest code with an actual Neato, because we only recently found out about the issue with the minimum range. We are tentatively planning on trying to work with a neato on Thursday.

Did you learn any interesting lessons for future robotic programming projects? These could relate to working on robotics projects in teams, working on more open-ended (and longer term) problems, or any other relevant topic.

We learned to carefully map out exactly what is happening in each step in order to determine exactly what is happening and if it makes sense. Doing this might have enabled us to find our mistake earlier so that we could fix it.