INDIVIDUAL LAB REPORT 2

February 22, 2019

John Macdonald

Wholesome Robotics, Team E

Teammates:
Aman Agarwal
Aaditya Saraiya
Hillel Hochsztein
Dung-Han Lee

Contents

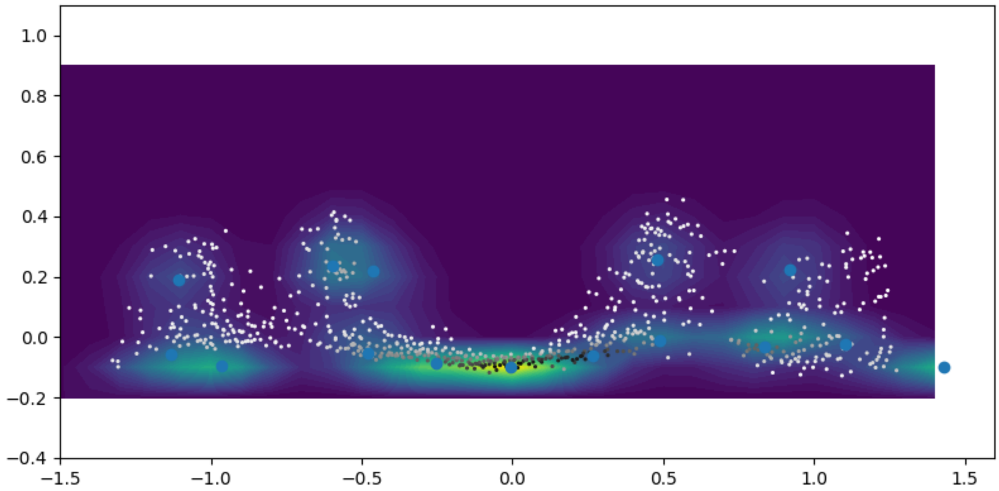

Figure 1: Visualization of the measurement model evaluating a point cloud. The white dots are the points from the lidar sensor and the gradient in the background is the probability distribution $P(z|x=(y,\theta))$.

0.1 Individual Progress

0.1.1 Project Management

After discussion first during the software sprint review and then with the team, we have agreed to all use a common Kanban board. I am very excited about this as it accomplished my previous goal of bringing the team members closer in terms of program management. We decided to move away from GitHub Projects due to (1) not having due date functionality and (2) being attuned towards software development. We are currently using Trello as our Kanban board tool, and are quite happy. We are currently displaying the Kanban board during every standup and making sure the tickets are updated then in order to keep everything up-to-date. I feel like we are much better keeping track of work items and not tasks fall under the radar which was an issue previously.

Additionally, I am currently discussing with the Software team how to handle merging into master and potentially a process for code reviews.

0.1.2 Engineering

I was able to fix the issue with the measurement model not having a likelihood peak in the expected location by realizing that the probability was being dominated by the

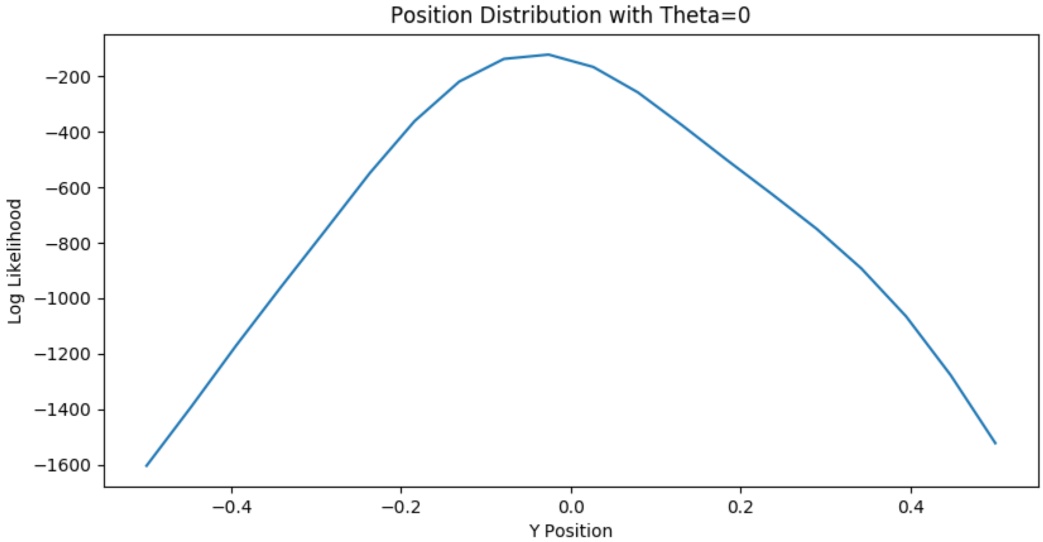

Figure 2: Graph of the likelihood $\sum_{i=0}^{N} \log(P(z_i|\theta = 0))$ with the changed to the measurement model. There is a smooth peak around $y = 0$, which is what we would expect from the data! ($y = 0$ corresponds to the robot being in the center of the row.)

Gaussian component representing the ground. Because there are a lot of ground points in the data, without incorporating the information about individual beams, the model expects any point to be on the ground with high probability. I knew that this would be an issue (ignoring conditioning variables is generally bad!), but I didn't anticipate the specific way in which it manifested itself, causing me to search for other explanations. I am quite confident that creating a new likelihood function which incorporates the prior knowledge of the angle and azimuth of the point observation, thus fixing a line along the observation must lie, will remove this issue. The current work-around is to remove the weights associated with each Gaussian component, causing all components to have equal weight. This helps in the case where the angle of the robot θ is fixed, shown in Figure 1. A visualization of the likelihood being evaluated is shown in Figure 2. Note that the issue where there is a peak in probability even for extreme θ still applies. I hope that this can finally be fixed by incorporating the knowledge of beam angle and azimuth.

Additionally, I labeled images for Dung-Han's plant health work, and discovered several ways in which the current labeling workflow could be sped up. As part of this, I wrote a script for Dung-Han which automates the second half of the original labeling process, in which unique labels are assigned to a number of plant health indicators (currently, discolorations and holes).

I additionally implemented a particle filter in order to evaluate my sensor model's

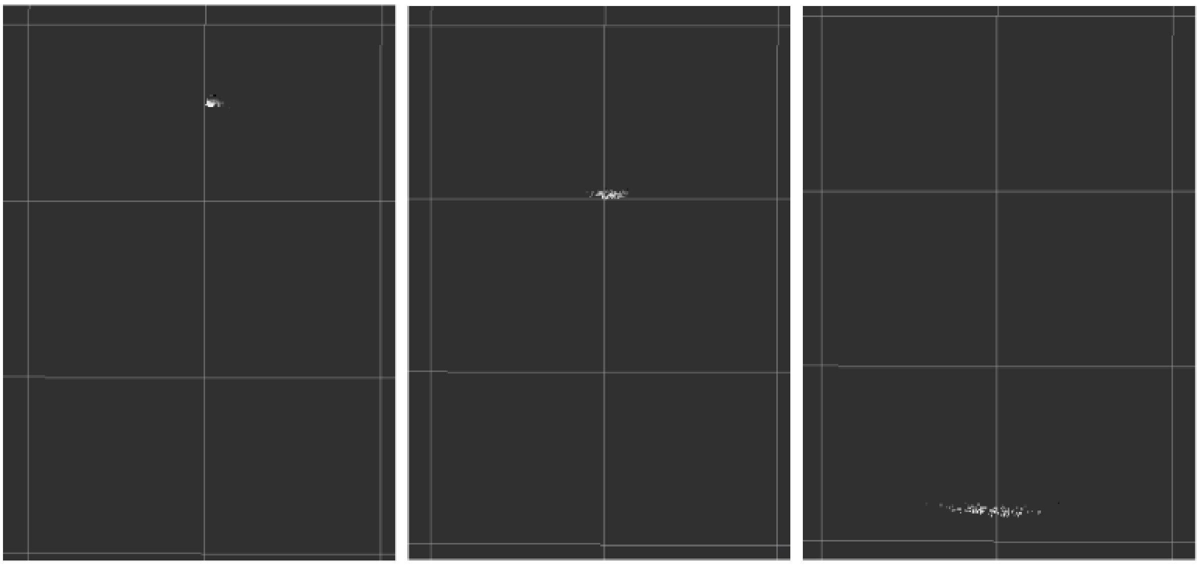

Figure 3: Visualization of the particle filter in `rviz` over several time steps.

usefulness in estimating position over time. The particle filter was implementing in Python, using ROS publishers and subscribers to retrieve data from a rosbag being played using `rosbag play`. (Previously, a `test_sensor_model.py` script, using my `BagProcessor` class to directly read rosbag data, was used.) This will enable future use of the localization algorithm on the robot. Numpy was used for computation, and my particle filter directly imports my sensor model, `GMMSensorModel`, stores it in a instance variable of my `Localizer` class (which implements a particle filter) and calls its `get_observation_likelihood` method appropriately. The output of the particle filter is visualized in `rviz` in Figure 3.

0.2 Challenges

0.2.1 Program Management

The introduction of Progress Reviews in the project course has complicated our project management workflow. We had planned our sprints around having deliverables every two weeks, but the project course requires progress reviews every two weeks. Reorganizing the sprints in order to address the added constraints of the project course is a work item which still need to be completed.

0.2.2 Engineering

The particle filter has issues with divergence, directly due to the issue where it believes certain states with high θ to be likely for a particular y. I believe that incorporating the knowledge of beam angle and azimuth may fix this.

0.3 Teamwork

We continue to work on the following domains:

- John: Software, robot localization
- Aaditya: Software, SLAM
- DHL: Software, plant health indicators
- Aman: Hardware, robot platform
- Hillel: Hardware, weeding

I most closely collaborated with Dung-Han, with whom I developed a new labeling process to boost the amount of data for our deep learning based plant health algorithm (currently, a fine-tuned Mask-RCNN model). I additionally collaborated with Aman regarding sensor mounting and wheel guards.

0.4 Plans

I would like to evaluate the particle filter on ground truth data from our farm visit to get a clearer idea of performance. This would also open the door to being able to evaluate other models in a quantitative way.