INDIVIDUAL LAB REPORT 11

November 22, 2019

John Macdonald

Wholesome Robotics, Team E

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

Contents

Figure 1: The robot successfully navigating in the field!

0.1 Individual Progress

I was finally able to make the robot reliably and accurately follow a row of crops! I did this by switching from a strategy which tried to estimate the position and orientation of the rows of crops to one that tries to estimate the the position and orientation of the center of the ground which the robot can navigate in. This actually yielded better results than the approach of fitting to the rows. The reality is, the plants are not uniform and the points that lie on the plants do not even necessarily capture the full structure of the plants, as they are relatively complex. The ground, on the other hand, is very uniform in our case, due to the anti-weed tarp which is currently laid on the field at Rivendale.

The approach is simply to threshold the pointcloud in the ground-stabilized frame on XYZ coordinates. After thresholding, I can apply the same line-fitting technique used previously to fit to rows. Specifically, I find the minor axis of the points by taking the SVD of the covariance matrix corresponding to this group of points. The, I take the eigenvector corresponding to the second largest eigenvalue, which is also the normal to the line of best fit (in a linear least squares sense).

0.2 Challenges

At the field test, I was able to make the robot navigate using a simple localizer which simply always outputs X=0, and sets y and theta based on the current row detection. For this reason, it will now work outside the row and is only useful for following a row. I tested integration with a modified version of the old particle filter, in order to enable navigation outside the row. We discovered that while the particle filter tends to converge to the correct value, it does so very slowly. This actually causes worse row-following

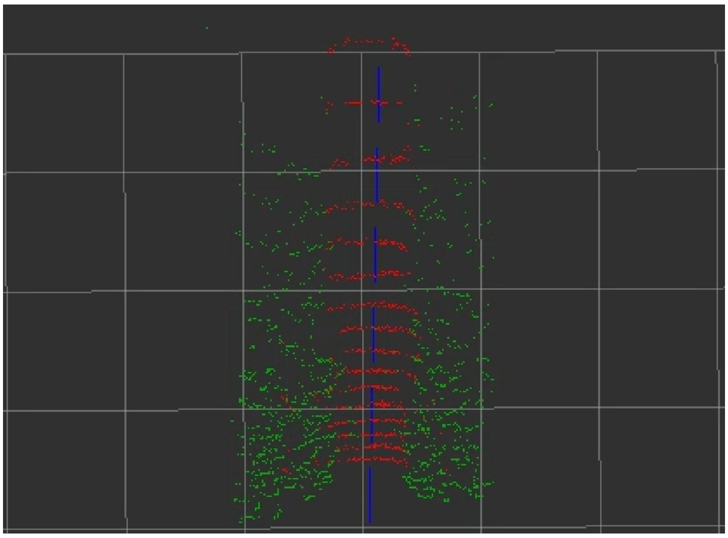

Figure 2: Visualization of the new row detector.

performance than the lidar-only row following approach. One limitation of the particle filter is that since it is so slow due to being written in Python, we are only using 100 particles to estimate the position of the robot.

0.3 Teamwork

1. Hillel: UI for visualizing plant health, plant guards
2. Aaditya: UI for visualizing plant health, particle filter
3. Aman: Work on using `robot_localization` for fusion of VO + IMU + RTK
4. John: Improved row detection, particle filter
5. Dung-Han Lee: Plant health monitoring, monitoring pipeline integration

I helped Aman to debug his robot_localization package integration issues, finding a bug in which he was importing two different versions of tf and trying to use both of them together. I also collaborated with Aaditya on adapting the particle filter to work with the new row detector.

0.4 Plans

We are currently planning to try to speed up the particle filter, which might enable similar row following performance to that we saw last semester.

One thing I would like to try is a very simple localizer that simply projects the visual odometry readings onto the line defined by the row detection when there is a row detection avaliable. This should behave the same as the simple row following method inside the rows and behave like visual odometry outside of the rows. This method seems like it could provide a simpler and easier to debug method of localization. Alternatively, I could write code to publish pose estimates with covariances from the row detections, by setting covariance along the axis parallel to the row to be very large.