# Progress Review 12

Individual Lab Report 11

Aaditya Saraiya

Nov 19, 2019

**Team E:**

Wholesome Robotics

**Teammates:**

Aaditya Saraiya

John Macdonald

Dung-Han Lee

Aman Agarwal

Hillel Hochsztein

# Individual Progress

## Capstone Project

Team E is creating an organic monitoring robot which has to autonomously navigate through crop rows. For this phase of the project, I had two main tasks which can listed as follows:

1) Integrating the complete monitoring pipeline and testing the functionality of the complete pipeline.

2) Working with John to test the particle filter and identify potential reasons for divergence.

### Monitoring pipeline integration

**Previous status**

Before this PR, the monitoring pipeline was technically feature complete. However, the integration step which would tightly couple the plant health detection module was not complete. The exposure checking script had been independently tested, but not integrated into the pipeline. The pipeline was integrated with the old Mask R-CNN based detection model and did not have support for the new UNet model. The majority of this progress review was focused on tightly integrating and testing different features from the pipeline and creating an MVP for the monitoring pipeline. Some of the key steps involved have been included in the next section.

**Progress updates**

The image exposure script which was developed and elucidated in previous ILRs was integrated into the pipeline. The final functionality involves checking for exposure and providing statistics on how good the exposure of the images is and if 75% of images from the total ROS Bag are usable. This module also has a feature where all the under-exposed and over-exposed images are deleted automatically and are not passed into the detection process.

The UNet-based plant health detection model developed by Dung-Han Lee, is able to achieve sufficient results which surpass our requirements. This pipeline is based on Pytorch. As the current Mask R-CNN pipeline is based off a TensorFlow module, certain changes had to be made for the plant health detection module to be compatible with the UNet model. I collaborated with Dung-Han Lee to successfully integrate this model into the pipeline.

I also developed a script to run the visualizer in two modes, creation and visualize. The creation mode is utilized only the first time, while the visualizer model is run for subsequent validation and testing. This added functionality allows us to store multiple pickle file and observe visualization results simultaneously in a loop. Although trivial, this functionality was not present in the previous module. Hence, addition of this functionality allows us to quickly test and visualize results.

I also added a feature to skip extraction of images with a stride in the ROS Bag. This allowed us to test our key features on smaller subsets of the ROS Bag. Additional features like this helped us to exponentially cut our testing time.

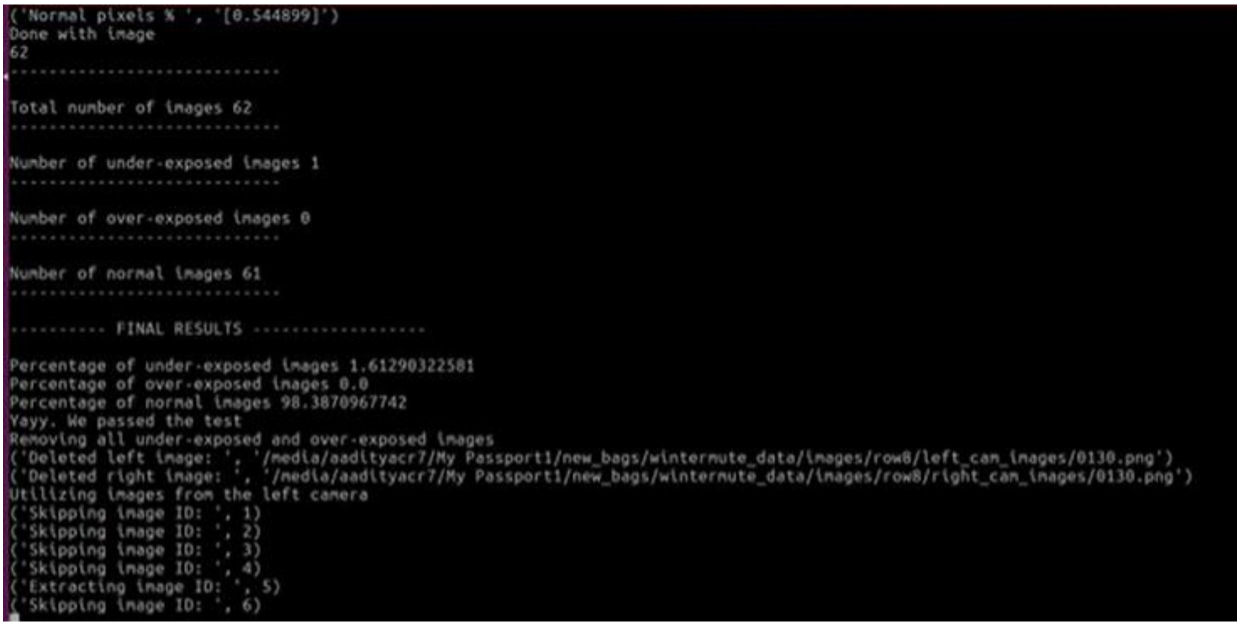


Figure 1: Demonstration of image skipping and exposure checking statistics

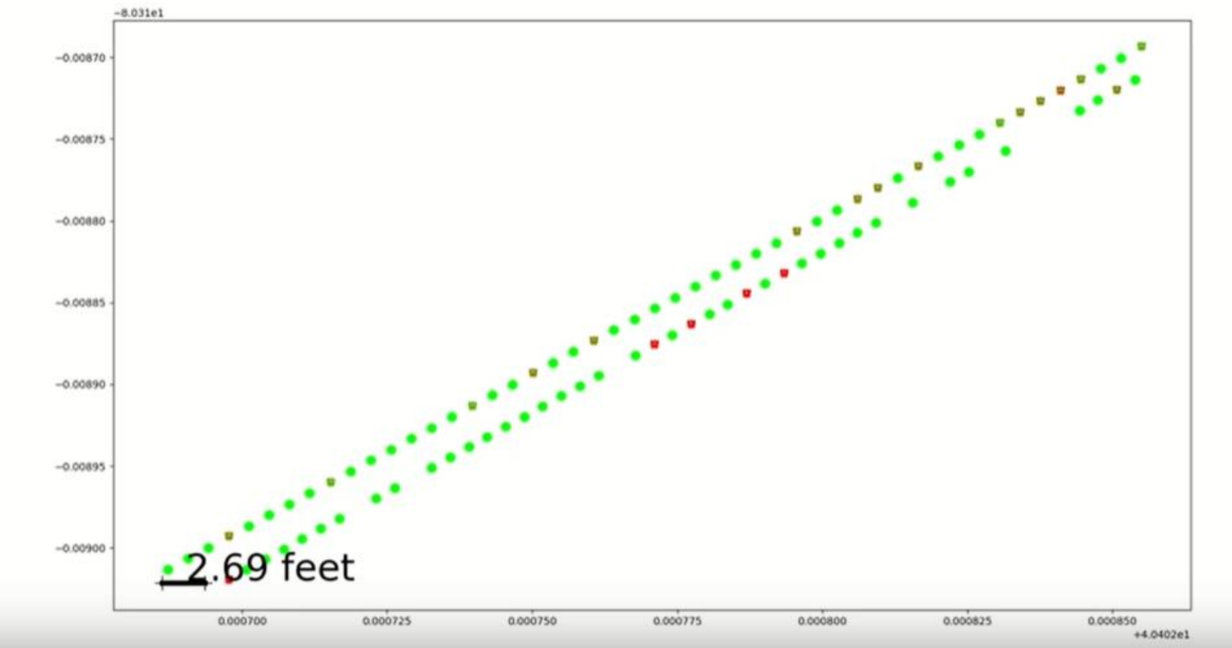


Figure 2: Demo view from the full integrated monitoring pipeline

## Sensor Fusion for Localization

**State of pipeline before PR**

Before this given PR, my efforts towards the sensor fusion pipeline was to collaborate with Aman to get the UKF + EKF based localization working. There were some issues specific to the robot_localization ROS package which prevented us from testing the pipeline successfully on the field. With the recent success of John's row detector module with segmented ground points, my focus during this progress review was to check performance of the particle filter pipeline on newly collected ROS Bags and perform parameter tuning to check improvement in performance. Because of lack of corresponding ground truth, the performance evaluation has been performed qualitatively.

### Particle filter tuning observations

The process of tuning the particle filter involved interfacing between Johns's modified row detector module and the old version of the particle filter. The old version of the particle filter had been tested indoors. The major results from the tuning process can be summarized in the results table as follows:

**Table 1: Particle filter tuning results**

| Parameter tuned | Reason for tuning | Outcome and observation |
|---|---|---|
| Number of particles | Increased number of particles would be robust to noise and have a better chance to convergence to the true state of the robot | Increase in number of particles decreased the divergence of the filter in multiple trials on the same ROS Bag.<br>However, the filter started lagging behind from the expected rate. This lead to a conclusion that the sensor model based of Python was slow and potentially a faster sensor model written in C++ was required. |
| Standard deviation for distance and theta parameters in sensor model | Changing the standard deviation for particle filter estimates shows how much we believe in the measurement models ability to capture the true observation | Decreasing the d and theta deviation lead to a faster convergence as well as decreased in divergence from the expected path. However, it also induced oscillatory motion into the filter. |
| Changing the linspace of Gaussian for sampled observations | This parameter was supposed to be the reason for decrease in speed of the filter. | Decreasing the linspace affects the performance drastically beyond a certain value. The precision lost is not directly compensated with an increased speed of the sensor model. |

The general conclusion from the parameter tuning process is that the noise and covariance parameters which work successfully on the ROS Bag, do not translate directly to the field. Considerable time was spent in tuning these values on the field. Tuning of parameters will be even more important for row switching when the LIDAR observations will be lost. Hence, this process needs to be done more formally before the last field visit.

## Challenges

This particular progress review involved a lot of challenges, for both the sensor fusion pipeline as well as the integration task. The challenges section has been divided into two parts to elucidate the challenges faced with both the pipelines.

### Sensor Fusion Pipeline

For the sensor fusion pipeline, some of the major challenges were involved with the variance in performance of our localization modules on ROS Bags and on real data in the fields. Firstly, in order to test the particle filter on ROS Bags, considerable tuning was performed in order to get good performance. However, it was noticed that the same set of parameters did not work well on the farm. As the row detector module can only be tested on the field, reliability with the row detector has been a major concern.

For the EKF and UKF fusion process, there were issues faced with the UTM to map transform not being published by the package. Inherently, this was a risk factor which we should have considered while working with off-the-shelf ROS packages.

Monitoring pipeline integration

For the monitoring pipeline, the goal for this progress review was to perform a complete integration of the monitoring pipeline. For the monitoring pipeline, at the end of the last PR, there were certain compatibility issues between Python2 and Python3 source code which prevented certain features of the Visualizer to be properly displayed. However, we learnt our lessons from our last experiences. I spent considerable time working with Hillel and Dung-Han to integrate the new deep learning model as well as make test the MVP for potential failure cases. This allowed us to successfully deliver all the required features for the Progress review.

## Teamwork

### John Macdonald

John specifically worked on tuning the row detector model on recently collected ROS Bags. He also worked on tuning the particle filter to make it robust for field testing.

### Aman Agarwal

Aman worked primarily on the robot_localization package to perform sensor fusion with the IMU, visual odometry from the ZED camera and GPS data. The majority of his effort was trying to debug the robot_localization package and make it ready for field testing.

### Hillel Hochsztein

Hillel worked primarily on fixing version compatibility issues between Python2 and Python 3. He also worked in unison with me to push forward for a first MVP version of the monitoring pipeline.

### Dung-Han Lee

Dung-Han Lee performed further analysis and tests with the UNet based model. He also collaborated with me to integrate the UNet based model into the complete monitoring pipeline, tightly integrated with the visualizer.

## Future Plans

### Team

With the monitoring MVP implemented and presented at the last progress review, the team's goals for the upcoming week before the Fall Validation demonstration have been summarized below. Firstly, the main aim of the team is to get row switching working for the given set of two rows on the field. The potential approaches which could be used include a tuned version of the current particle filter, a particle filter with a motion model augmented using a UKF based fusion node, EKF + UKF based node which fuses visual odometry, IMU and GPS data. Secondly, on the monitoring side, the focus would be to train the UNet based model with freshly labelled data and quantify the performance for the network. Completion of these two tasks will finish majority of our deliverables for the Fall Validation demonstration.

### Individual

The individual tasks for the next one week have been listed below. Firstly, I will focus on speeding up the particle filter. This will be done by converting either just the sensor model or both sensor and motion model to C++ based ROS nodes instead of Python based ROS nodes. Currently, speed has been observed as a bottleneck as we are not able to increase our number of particles above 150. Secondly, I will collaborate with Hillel and Dung-Han Lee to integrate any minor modifications into the monitoring pipeline. As demonstrated in the last progress review presentation, the MVP of the monitoring pipeline is ready. Hence, the majority of tasks will involve generation of new visualization results from both freshly collected data as well as new data. This will be done in order to showcase the scalability of our tool.