Individual Lab Report 8
By Clare Cui

Team B: Arcus
Clare Cui
Maitreya Naik
Angad Sidhu
Logan Wan

2 March 2017

<u>Individual Progress</u>

For this progress review, I was primarily responsible for working on the mapping framework. Since there has been a lot of code to sift through, much of my time has been in understanding the framework already provided to us by RASL. As an intermediary step to facilitate this, I am working on a summary of one of the packages to send to our PhD advisor for review. I have also set up some member functions for Logan to use as we were originally both working on the same package. It was decided earlier this past week that working on separate packages would be a cleaner split of our roles with fewer dependencies on one another.

The mapping framework consists of two main packages: map_server and map_utils. Map_server subscribes to the LiDAR and RGB camera nodes, sets up the map parameters, and processes messages from the sensor topics. Map_server also uses a package called observation_synchronizer. This package takes in data and sorts it according to a timestamp. Map_utils is responsible for visualizing the map and creating a dense voxel grid map that shows the log-likelihood of whether or not a voxel is occupied. It is updated continuously with point cloud information.

We are trying to improve the mapping framework functionality by taking in LiDAR point clouds, transforming them into the body frame, and then transforming them to a point in time that aligns with the RGB camera. Once this is accomplished, the LiDAR point cloud will be transformed into the RGB camera frame. Using the 3D coordinates and the intrinsic camera matrix, we can back-project the 3D points onto a 2D plane. This will then be sent into map_utils to colorize different voxels and mark them as visited. This is where my role comes in. In order to visualize the color data, map_utils will need to be able to handle a new struct that defines the parameters of a voxel by XYZRGB. This will require meticulously going through the map_utils package and accommodating RGB in all the functions. A small part of this will also include converting sensor_msgs::PointCloud to pcl::PCLPointCloud2 because the sensor information is more easily accessible in this new format. To test all of this code, I will be using a RASL simulation. Although we have collected bags from Lafarge, the simulation will not have any noise and will help us confirm that our algorithms are actually working correctly. The UAV with RGB camera and LiDAR sensor has already been imported, and the UAV currently autonomously explores its environment while mapping simultaneously (to be clear, the exploratory planner was written by RASL, not us). An image of the simulation can be seen in Figure 1.

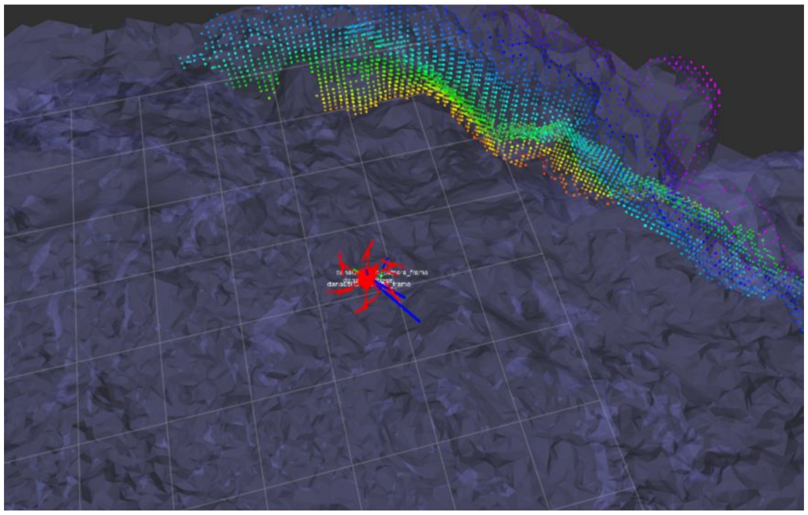

Figure 1: Simulation with Arcus UAV mapping.

Challenges

By far the most difficult challenge has been in understanding the map_server and map_utils code. It is quite complicated with class member functions all referencing other classes in different packages, and oftentimes it is not immediately clear what the function does, either because I am unfamiliar with the terminology that is used or with abbreviated variable names. I am slowly working through this and making sure to ask our PhD advisor plenty of questions in the process.

Teamwork

Logan Wan: Logan worked on the mapping framework, prepared for the field test, and helped rebuild the broken drone.

Angad Sidhu: Angad worked on repairing the drone, setting up udev rules for the UAV, and updated software on the UAV before we did the test flight. He also set up the UAV for the flight test at Lafarge.

Maitreya Naik: Maitreya prepared the new ESCs for mounting on the drone, loaded the RASL firmware onto the Pixracer, and tested it with old motors.

Future Plans

For the next progress review, I will continue to work on the visualization aspect of the mapping framework. I hope to have a complete or near-complete understanding of the map_utils package and have modified map_utils and any other linked packages be able to visualize the colorized dense voxel map.