Logan Wan
MRSD Team B: Arcus
Clare Cui, Maitreya Naik, Angad Sidhu
ILR07
2/16/17

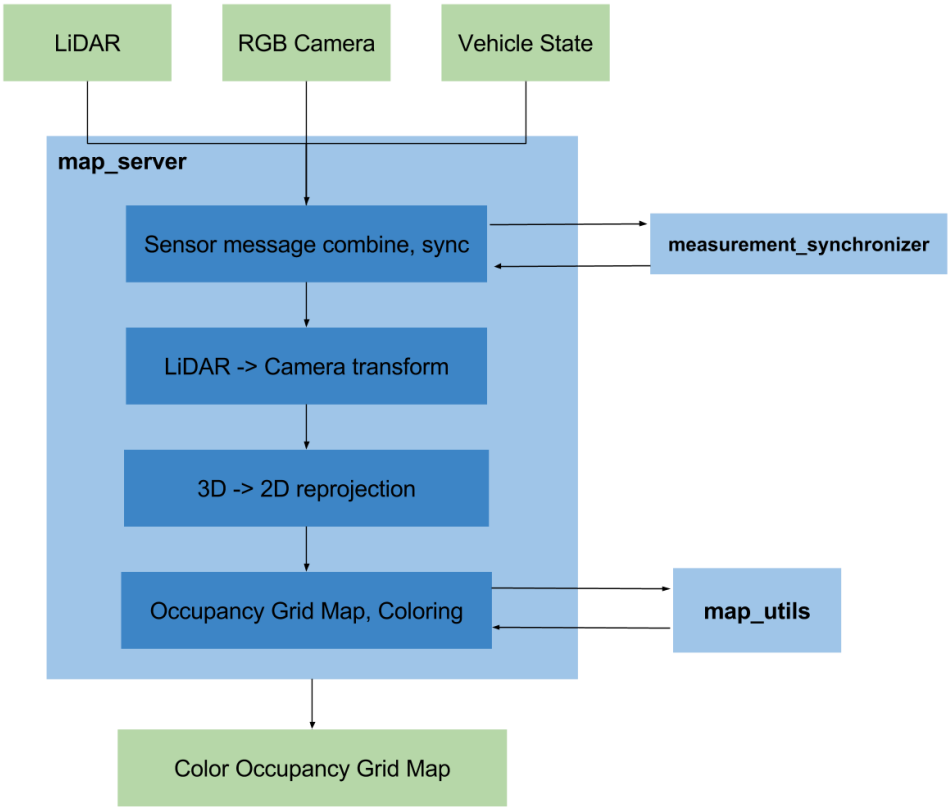
Individual Progress

On the project management side, we received a new schedule from our advisor that detailed a project timeline dramatically different from the one previously proposed to both the project course and necessary stakeholders in the project. It described a set of milestones that accelerated our final experiment by 3 weeks while adding additional autonomous functionality that we had not planned on. In order to do so, our work would be limited to completing the tasks currently on hand, while introducing additional lab members to accelerate the development of autonomous capabilities. Our MRSD team would have effectively stopped working at the end of February.

These issues were eventually solved through extra communication with Wennie, our project advisor. We later found out that a partial motivation for the accelerated timeline was to publish results in a paper due April 1st, as well as a dramatically more complex autonomous capability demonstration. By limiting the scope of the experiment (to the original SVE) and effectively working towards a parallel set of milestones, we were able to both preserve the need to develop advanced research functionality for publication while maintaining our desire to stick to our original SVE and perform our own implementations for minimum viable autonomous behaviors.

On project development: particularly motivated in part by a coding challenge, I dove deeper into the workings of C++ and the overall architecture of the mapping functionality that Clare and I are working on. Before last week, we and our project advisor had an understanding of what work would be required. It turned out that we all had a very different idea of how the functionality would be implemented, as well as how it would interact with other packages in the software stack. Most surprisingly was the difficulty in actually obtaining access to RASL packages that our project advisor insisted we use, and improvising while we were eventually (today) granted access to the ~21 packages that were all necessary to begin work.

After working with Wennie, I sketched out the diagram below:



The green boxes are ROS messages, with light blue boxes indicating separate ROS packages. In dark blue are discrete functions that we would need to add to map_server. For myself, I would be working on reprojecting LiDAR point cloud data into the RGB image to grab corresponding color values. This also requires converting the current octree map data structure into a occupancy grid map, which would discretize the map into voxels with a occupancy likelihood as well as R,G,B values.

Map_server, in conjunction with map_utils, provides a set of tools that would be used to combine sensor data and publish map data that could be visualized with RViz. However, it still requires major customization to add the features described, much less run in real (enough) time. I set up a branch of our main software repo, adding the required packages and integrating them into our build system. I've also modified map_utils to handle colorized point cloud data, and am currently working on reprojecting the corresponding LiDAR point onto the RGB image.

Challenges

A major challenge for myself was managing the seemingly conflicting requirements of both our advisor and our project; mediating a set of milestones that both parties could be happy with.

Like last progress review, a challenge was quickly getting up to speed on a seemingly (to me) complex build system. In an attempt to modularize the code as much as possible, it looks like almost every tiny feature has been broken out into its own individual package.

Teamwork

Angad: Worked on integrating the IMU, improving BLAM! Capability.
Clare: Finished hardware improvements, found new sensor transformations per hardware updates, began map work.
Matireya: MAVROS and pixracer integration.

Plan

Continued work on localization software and mapping software. Maitreya will be moving on towards beginning working on autonomy features (obstacle detection and path planning).