Angad Sidhu
Team B Arcus
Clare Cui, Maitreya Naik, Logan Wan
ILR 2
10/21/2016

Progress

Since the beginning of this semester, we have been focusing on getting the quadcopter and sensor rigs running. I specifically have been focused on downloading and setting up drivers for the IMU, camera, LiDaR, and GPS. I have also been working on getting calibration software for the intrinsics for the camera and extrinsics for the imu and camera running. I also focused on getting our RTK setup for the GPS tested and running so we can establish a ground truth for our state estimation software. After getting all the software and drivers running, we took our rig out to LaFarge quarry and drove around the quarry roughly 50-100m for two runs while collecting data. The run specifically involved driving down a path in the quarry surrounded by rock walls, driving in loops in a slightly larger area, and then returning to our initial position.

Since the last ILR, I have spent time on the calibration of sensors, research, and testing. I am using the Kalibr library to calibrate the intrinsics of our camera and the extrinsics of our IMU and camera setup. By establishing a pipeline for calibration we can ease integration of the hardware after the system is built. I have been doing research into various algorithms to use for state estimation and localization. In particular, I have been examining BLAM and LOAM to determine their effectiveness and suitability to our problem.

LOAM operates in roughly 3 steps; motion distortion correction, feature extraction, correspondence and homography calculation. The motion distortion correction essentially remaps points in a scan based on the motion estimated (either by IMU or state estimate generated from LOAM itself), assuming motion is linear. This may not be as big a challenge for our implementation as the paper describes using a lidar mounted to a - relatively - slow rotating joint, whereas we are using a VLP-16 and have no slow rotating joint. The feature extraction works primarily by determining if a point is an edge or a plane by estimating curvature. The correspondence and homography calculation is then done by using an ICP method to calculate a homography.

BLAM is a toolkit created, mostly, using off the shelf algorithms from PCL and other libraries. I haven’t fully determined the structure but from what I know it operates using Generalized ICP from PCL and it also stores its data points in a Octree which is published for consumption. Fig 1 shows an offline map generated with BLAM from our bagged data. The challenges with BLAM is that it’s online performance was much better than LOAM’s because of its loop closure feature, however this also causes it to be much more computationally expensive. Another improvement we could make to increase performance of BLAM is by feeding in IMU data when performing ICP as a homography estimate. This, hopefully, will decrease convergence time and increase accuracy. You can see in figure 1 that there is some drift and some problem with loop closure as we approach our starting point. Another part of BLAM we will likely need to alter is the mapping portion. It currently stores points only as XYZ however we need to use or design a data structure which contains points which may or may not have multiple modalities of perceptual sensor data. Whether this falls under simply adding an additional octree with XYZRGB* data, or creating a special format which can accept conditional data is a design decision we will have to make.

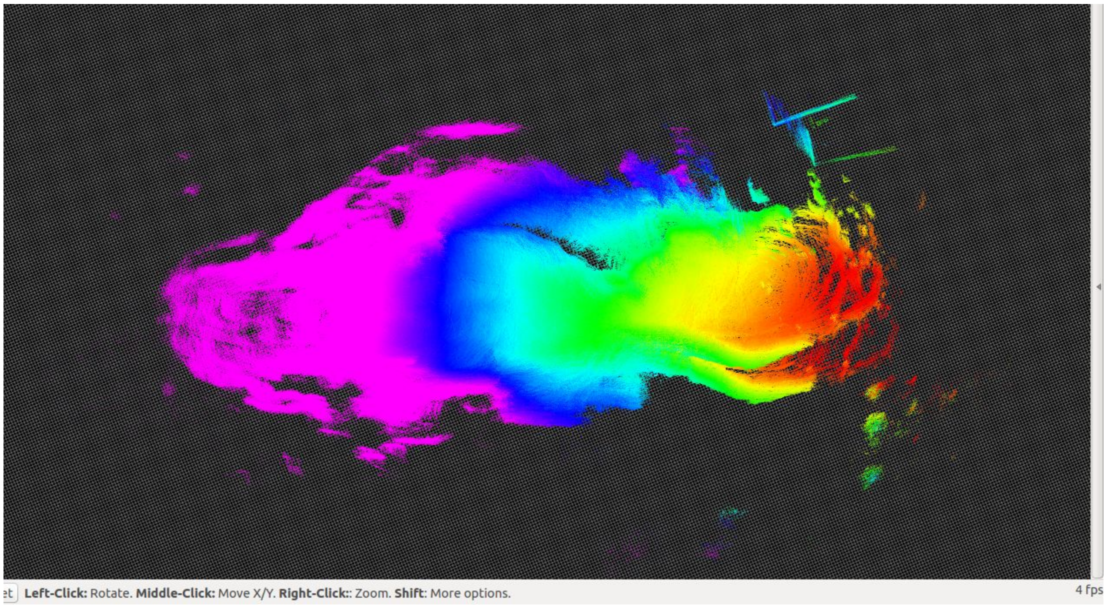
et
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
4 fps

Figure 1. Map generated from BLAM

Challenges

The challenges I encountered have mostly been in trying to get Kalibr running and calibrating the cameras using my machine. I have spent quite a bit of time debugging the Kalibr calibration problem but am not quite sure I have arrived at a solution just yet. The issue is that Kalibr is a cross language library which involves bindings between C++ and python. I am getting cryptic errors which lead me to believe that Kalibr is instantiating these bindings correctly likely due to version mismatch between libraries. I believe the issue lies in a version mismatch of BoostLib with which the bindings are generated. Assuming that is indeed the issue, the fix is to install a second version of boost and swap when running Kalibr. Kalibr also features a compiled binary distributable which I will try using instead of my version which I compiled from source. Perhaps attempting to use that will either work right out of the box, or throw another error message that will give me more information about solving my problem.

Another issue we had was with testing and setting up the RTK GPS. It is very sensitive to satellites and requires at least 4 with 6-7 being preferred in order to get a good fix. While testing near our lab we could barely get the required signals we needed for an RTK fix. We ended up moving to the Cut in order to get away from large buildings which obstructed our view of the sky. We managed to get the required number of satellites but the setup was so flakey that once we picked up the rig to move it and record data, the fix was lost. We solved this by borrowing some survey grade antennas in the lab and setting those up. After that we managed to acquire a good RTK fix and grabbed great data just moving around in the Cut. This turned out not to be an issue, however, when we tested in LaFarge Quarry. There was no obstructed view of the sky and we managed to get great RTK fixes with no loss in signal even with our non survey grade antennas.

Teamwork

As mentioned before, we all travelled to La Farge quarry to collect data. Logan and Clare worked throughout the day preceding the trip CADing and building a sensor mount for the lidar, gps, and antenna setups. Maitreya helped me with calibration of cameras by assisting me with installation and capturing of calibration data. Maitreya also was responsible for calibrating and setting up the Pixhawk and getting the rotor flight worthy.

Plans

My future plans are to work through BLAM to understand it better, including associated papers on techniques used for loop closure, etc. I would like to figure out more about occupancy grid maps and other mapping data structures we can use to store data. We also need to bag more data so we can run modify and run BLAM with IMU data to get more accurate matches. Another thing I need to work on is the general breakdown of software tasks so that we can better distribute work.