Angad Sidhu
Team B Arcus
Logan Wan, Maitreya Naik, Clare Cui
ILR 7
Feb 15 2017

Individual Progress

Since the last PR I have been working on getting IMU measurements pre-integrated and added into our ISAM2 factor graph. I ran into some issues, which I'll discuss further below, related to performance and related to loop closures. The lion share of work was involved in porting code from example GTSAM implementation of IMU preintegration and from an implementation from someone else in the RASL group. Some modifications I made include pre-integrating IMU measurements as they come in rather than pre-integrating them in a batch when we are adding factors into the graph. I've also taken preintegration estimates and fed them into laser odometry ICP alignment to help it converge faster. The algorithm is now functional and runs in real time.

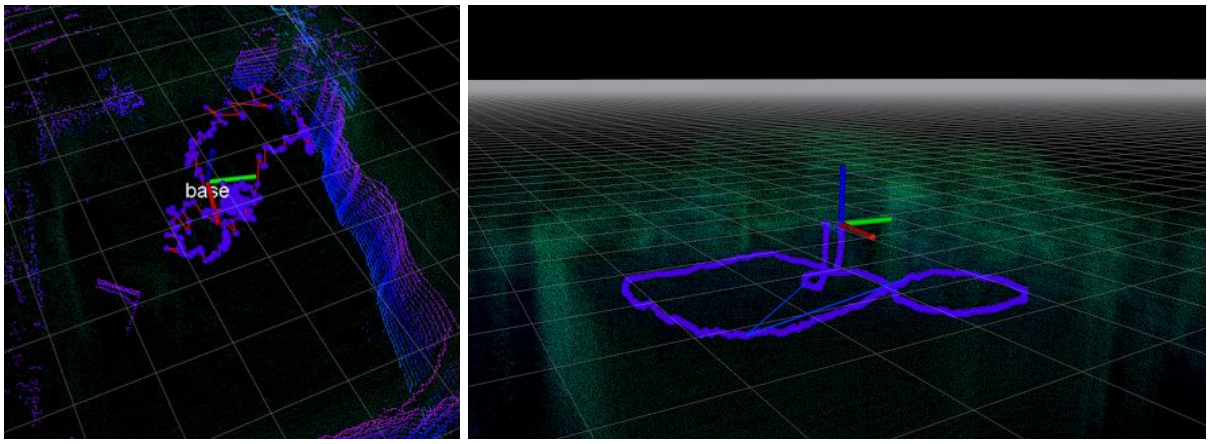

Figure 1. State estimation with pure laser odometry Vs IMU preintegration and loop closure.

However, it seems to be fragile and numerically blows up when running one of the bags we have collected at LaFarge Quarry. If you look at Figure 2, you will see that there is a ton of oscillation and vibration in our driving bagged data compared to a data set we captured of us walking around the motion capture arena. We will need to implement some sort of filter to help reduce noise to robustify our state estimation.

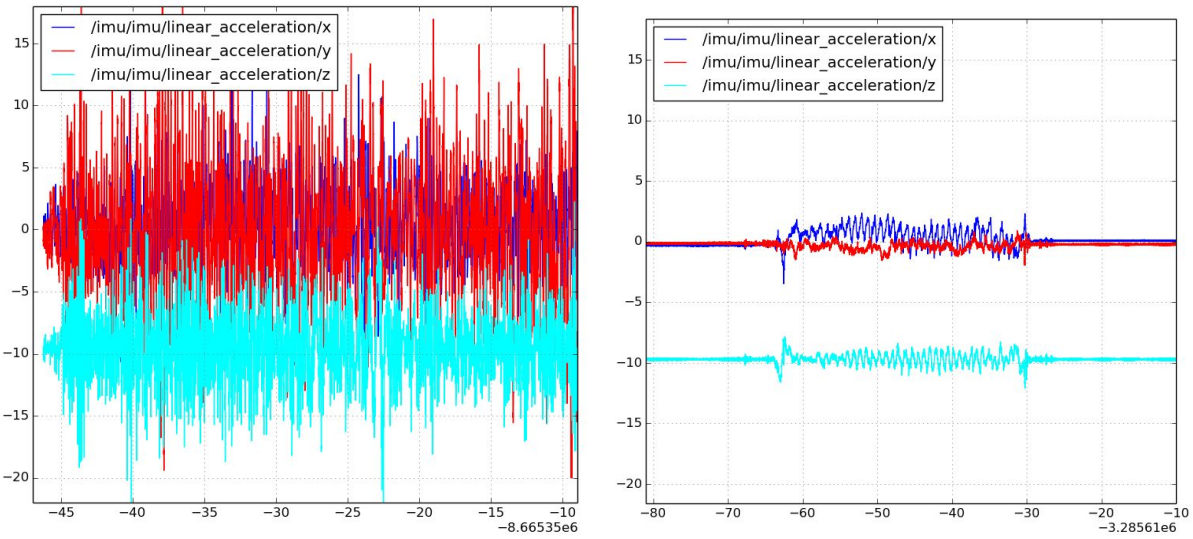

Figure 2. Two plots of IMU acceleration over time. Left is from driving around in Lafarge, Right is us walking the hex rotor around in a Motion capture arena.

I have also been working to help get teammates acquainted with some software packages by walking them through some of the more complicated C++ language constructs.

Challenges

Since last PR I spent a lot of time working on the software packages to get IMU measurements into GTSAM. I faced a few issues with performance and various bugs left over from sample code. I had to diagnose a lot of bugs regarding changes to BLAM's implementation of GTSAM. BLAM used unsigned integer keys to reference factors in GTSAM factor graph, however, GTSAM uses size_t which is unsigned long int. This results in numerical overflow when you stuff the keys into unsigned ints. This is a bug that existed in BLAM but was very prominently exposed when adding factors of velocity and imu bias measurements. This required transforming a lot of BLAM code to replace unsigned ints with GTSAM typedefs.

I tackled the performance problem by trying to diagnose the causes. I first attempted to profile the node, which is depicted in Figure 3. It told me PCL was the bottleneck which was strange because that aspect of the code hadn't changed much from the FVE. I determined that it was likely some sort of linker issue that was linking in a debug version of a library that it should not be. After examining build logs I found that it was a debug version of GTSAM that was being linked to instead of the release version. After linking to the correct version of GTSAM performance jumped to 105% of the real time on a bag.

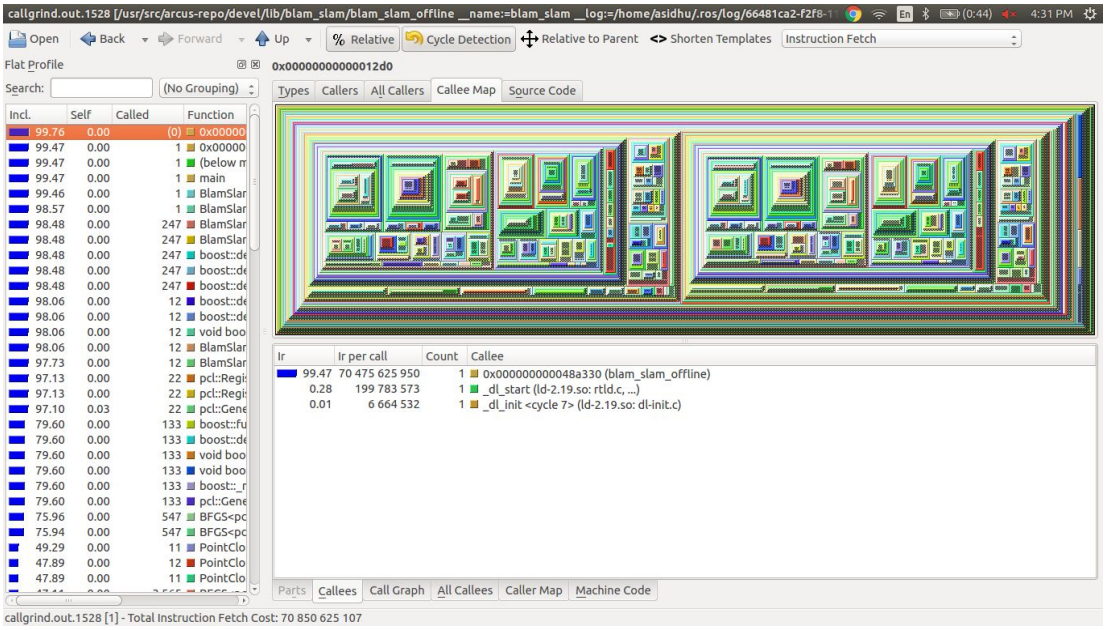

Figure 3. Valgrind profiler highlights where time is being spent in our algorithm.

Teamwork

Maitreya has been working on getting MAVROS communicating with the Pixracer. After working through a few issues, he was able to get MAVROS (and associated libraries) compiled and functioning. For a demo, he was able to send commands and targets to the Pixracer as well as report status.

Clare and Logan have been focusing on familiarizing themselves with the code in mapping packages provided by RASL. It is their job to analyze and extend these packages to add in color information to our map.

Plans

Since our last PR, we had a bit of turmoil with our advisor in terms of schedule, scope, and task management. We all met with her today to discuss a schedule she had sent to all of us. Since that conversation, we have come to an agreed path forward with our schedule.

I'll continue to focus on state estimation and making it more robust so that Logan and Clare have good state estimates to project RGB data onto the map. I'll be working with a Ph.D. student in the lab who has developed a very robust Kalman filter for estimating state. We are hoping to test the results of using sensor measurements of the lidar odometry and IMU measurements to generate a stable pose estimate that will run much faster and more robust to noise than integrating IMU measurements through

GTSAM. Along with this line of thought, I will be working on optimizing BLAM through its many tweakable parameters to increase speed and reduce error in a systematic way.