ILR #6: Progress Review 7

Harry Golash – Team A

February 2, 2017

Teammates: Amit Agarwal, Yihao Qian, Menghan Zhang, Zihao Zhang

1. Individual Progress:

1.1 Tasks:

For this progress review our team had three tasks to accomplish:

1. Meaningfully extract data from the radar based on literature review of PolySync.
2. Attempt to correlate Bounding Boxes from the stereo images for depth estimation of targets such as cars/pedestrians.
3. Familiarization of CUDA for future parallelization uses in perception computing.

The tasks I was partially responsible for were the first and third ones in the list above. In addition to working on installing and familiarizing myself with the PolySync interface and API, I also made some minor repairs to our test vehicle to make it presentable for our team meeting with Delphi this week.

1.2 Implementation:

1.2.1 Setting up PolySync and Kvaser adapter:

I did a clean install of Ubuntu 14.04 in order to use PolySync on my laptop. I contacted PolySync to get an activation license and then installed the PolySync middleware platform as per the instructions. Getting familiar with the API is what Amit and Zihao spent a fair bit of time last semester working on; I found the publisher-subscriber model and API to be fairly user friendly thus far. I am still working on integrating the raw data from the radar into our stereo vision output display that I created last semester. This should help us down the line when we need to demonstrate all our sensor fusion output on a single common interface. The Kvaser CAN-to-USB adapter that we bought last semester has a comprehensive Linux CAN library which can also be used in my code to potentially filter the noisy data we are still obtaining from the radar. I am also currently working on familiarizing myself with it.

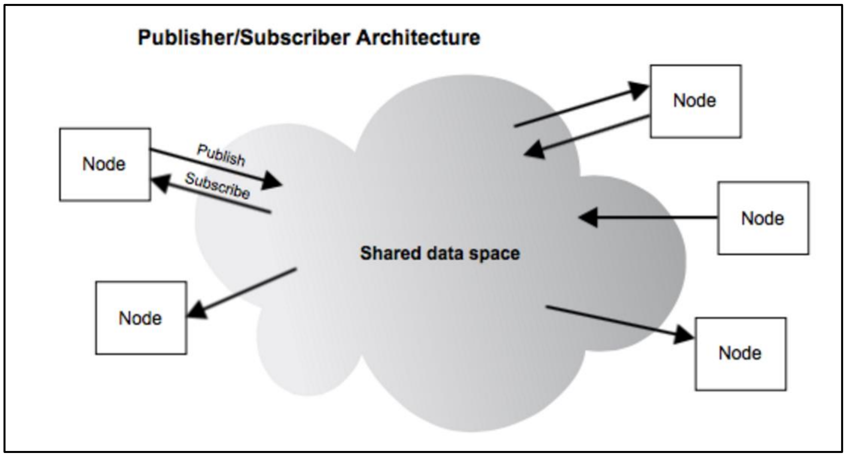

Fig. 1: The publisher-subscriber model used by PolySync

1.2.2 Radar filtering:

Despite our efforts on making the radar (sensibly) work last semester, at the start of this semester the radar is still giving data that is largely unusable. The tracking points received from the radar tend to flicker and get lost in the background noisy data. We have brought up the issue with Delphi, and they have decided to test our radar unit this week at their facility to ensure that the unit we have is not faulty.

To get a sense of the nature of the noisy data we were receiving, I used steel sheets to selectively block out the input to the radar from different directions (hardware filtering). I also used a steel sheet to try and “focus” the radar. These attempts did not yield much positive result – the best case was blocking out all radar data completely. I will continue to look into other possible ways to improve the radar performance via hardware modification.

On the software side, Zihao and I used basic C++ code and the PolySync API functions to output raw data into the Linux terminal. We were able to selectively display data points based on confidence ratings and selective filtering done by the radar itself. We will still need to work on filtering techniques significantly, since the radar data is still not clean enough for our object detection and tracking purposes.

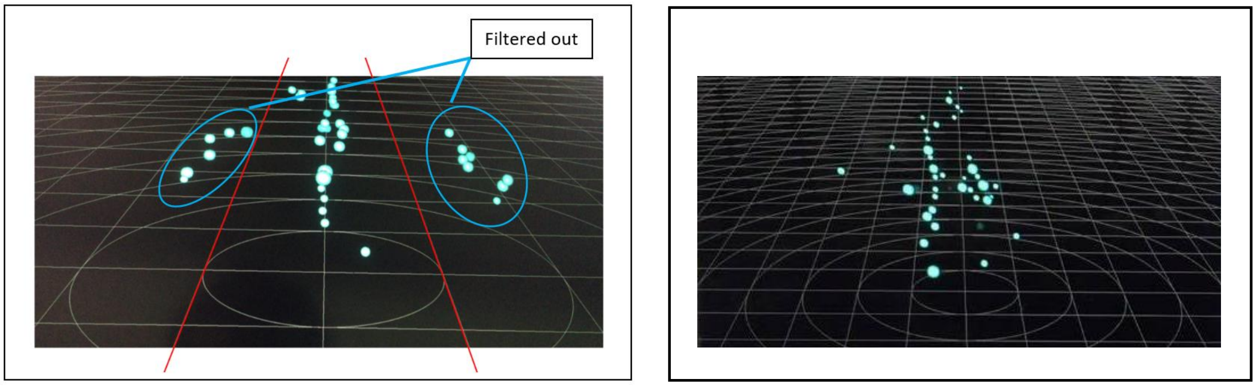

Fig. 2: Current status of preliminary radar-data filtering

1.2.2 Car repair:

I replaced the passenger-side broken mirror and the passenger-side broken taillight. I also buffed off some scratches from the rear passenger-side quarter panel and tried to remove a large dent (damage acquired last semester while testing). This was done to make the car more presentable and usable for the rest of the semester, especially given the fact that we plan on meeting our sponsor, Delphi, frequently in-person this semester.

2. Challenges:

The PolySync installation requires an activation ID during installation. To acquire this ID, I had to reach out to PolySync again, since the ID given to us last semester has expired. The PolySync API and the Kvaser CAN library have taken a bit more time than expected to get familiar with. Even with our current data-acquisition and filtering from the radar, we are not getting useable data for the purposes of this project. We need to keep working on better filtering techniques before the next Progress Review.

3. Teamwork:

There have been team difficulties regarding overall morale and positivity towards the project. I think this is largely in part due to the inavailability of us our test vehicle over the last couple of weeks. The Volvo we're using had some starter and battery issues owing to the inclement weather conditions and the age of the car. Additionally, we have been unsuccessful in scheduling dedicated time to work on the project as a team.

On the plus side, our team realizes and agrees that there is considerable work that needs to be done this semester and also before the next PR. We will work on distributing the workload better and assigning tasks more clearly to improve our productivity. Additionally, we will schedule more meeting times where we can collectively work on the project.

Zihao and I plan to work on the radar filtering and data-acquisition in to our own GUI (instead of relying on the PolySync GUI. Amit and Yihao and Menghan will continue to improve our stereo vision system to improve its accuracy and speed. We plan on demonstrating real-time stereo vision very soon.

4. Future Plans:

By the next progress review, we hope to accomplish the following:

1. Improve data filtering from the radar and synchronize it with the cameras using software methods
2. Demonstrate stereo-vision with depth estimation
3. Install PCB or off-the-shelf power solution (inverter) in the car
4. Interface with the car's systems to enable possible wireless control

I will primarily be working on the first, third, and fourth items in the list above.