ILR #8: Progress Review 9

Harry Golash – Team A

March 2, 2017

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

1. <u>Individual Progress:</u>

1.1 Tasks:

For this progress review our team had the following tasks accomplish:

1. Egomotion estimation using GPS
2. Improved stereo depth estimation and visualization
3. Choose a powerful computer for real-time performance
4. Display radar detection-level data

The task I was partially responsible for was the fourth one in the list above. In addition to acquiring detection-level data from the radar via Ethernet, I also conducted load testing on the PDS PCB that we built last semester.

1.2 Implementation:

1.2.1 Displaying radar detection-level data:

Now that we have decided to write our own code to acquire and display data from the radar, we will no longer be using PolySync for this purpose (as we have done until now). Detection-level data contains the obstacles and boundaries detected from the radar without any post-processing or clustering. Using this data, we will be able to visualize tracking and some shape information about our obstacles, since the number of points of interest correspond to the size and distance of the obstacles and boundaries. The reason we chose to switch from acquiring tracking-level data to detection-level data is due to the large amounts of noise we were obtaining when we visualized the clustered data (tracking-level data) from the radar (Fig 1.). Going forward, we can use the detection-level data to create our own tracking information, since the inbuilt processing method used by Delphi does not seem to work for our purposes.

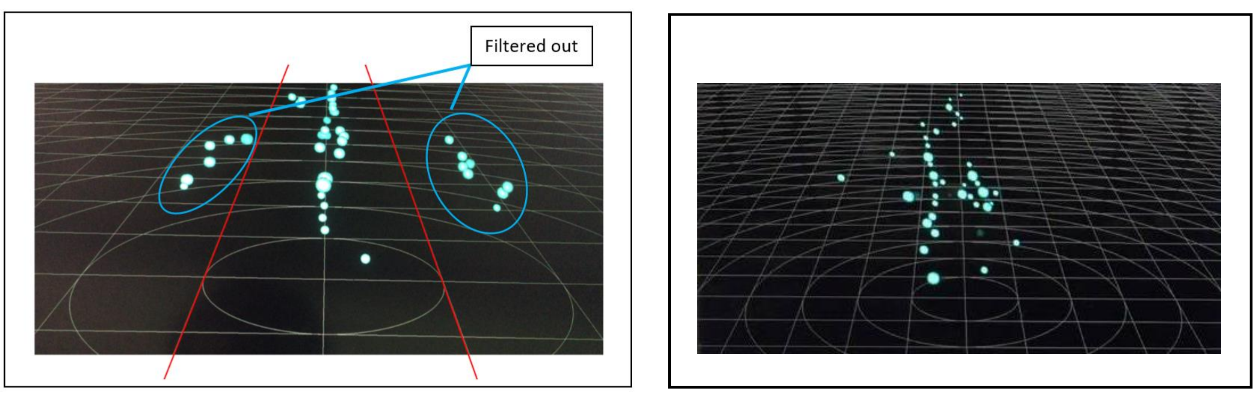

Fig. 1: Tracking-level data, before and after filtering. The data is noisy and unusable

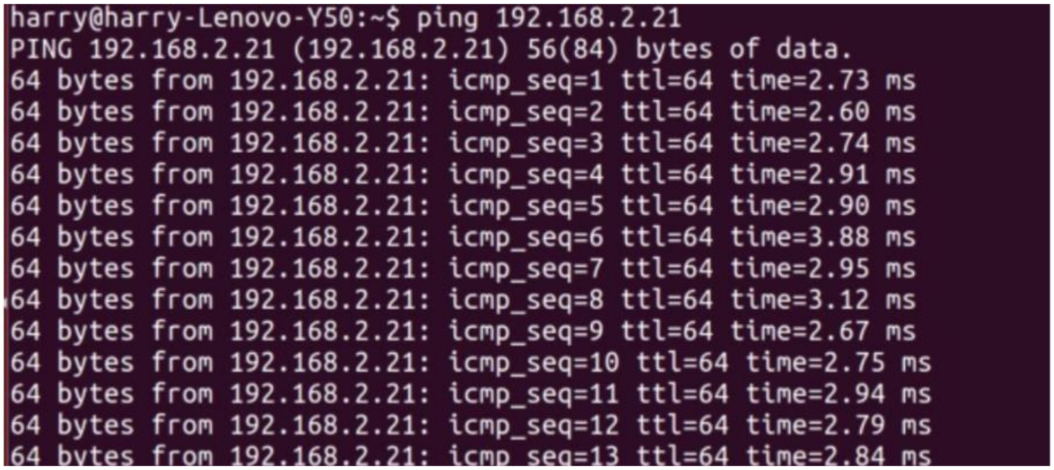

Fig. 2: Pinging the radar to confirm that it has been connected successfully.

Fig. 3: Real-time radar detection-level data. The format of the data was displayed in special characters and was not readable.

To connect the radar and read the detection-level data, we had to use the “ifconfig” and “netcat” commands in the linux terminal, since WireShark was unable to parse the data. Using “ifconfig”, we set the Ethernet port to have a similar local ip address to the radar (local IP was 192.168.2.10, radar is 192.168.2.21). Then we pinged the radar (Fig 2.) to confirm connectivity and used the “netcat” command to obtain the raw data (Fig 3.). Unfortunately, the data we obtained was displayed inappropriately (special characters and irregular formatting) in the terminal (Fig 3.). Since, we know the meaning of the data in hex, we had to convert the special character information into useable hex information. Going forward, we will use the Delphi dictionary to write a GUI to meaningfully visualize the detection-level data from the radar unit. We hope to accomplish this soon, so that we can move forward with system integration. At this point, we are lagging behind our schedule for the radar and we need to visualize the data and process it in order to catch up.

1.2.2 PDS PCB load testing:

The PDS PCB that we made last semester has been unreliable and has caught on fire twice before. To ensure that it would not damage our expensive sensors, I decided to do some load testing on it to see if it could supply the power we need continuously. To do so, I connected a motor and a potentiometer in series to each of the three output channels one by one, and then tried to draw approx. 1 amp of current from each channel (by varying the pot resistance). Unfortunately, the PCB fell apart due to poor assembly before testing could be concluded. The testing will be resumed once we have replaced the IC step-up converter that is broken. In the meantime, we have ordered off-the-shelf power distribution boards to test and use.

2. Challenges:

The Delphi team was unable to assist us with a lot of the initial connectivity issues we experienced with the radar. Moreover, WireShark recently failed to detect and parse the radar data, and we had to use netcat instead to view the data. Typically, we have used WireShark since it is easy to use and can save the data in a hex format directly. There is little documentation available as to what kind of data we expect to receive when the radar is functioning normally, so it is often hard to guess whether the data we are getting is typical or not. The Delphi team has been responsive, but unfortunately neither of our points of contacts could shed much light on our issues.

3. Teamwork:

There appear to still be difficulties regarding overall morale and positivity towards the project. We have set up meetings three times a week so as to track our progress and ensure that our team is functioning well towards our goals.

We all realize that there is a considerable amount of work that needs to be done in the next few weeks in order to get back on schedule. We are working on distributing the workload better and assigning tasks more clearly to improve our productivity.

Amit and I will work on the radar filtering and data-acquisition using our own GUI. Once we have properly visualized and understood the detection-level data, we will work on our own clustering methods to assist with vision-based tracking. Zihao and Yihao and Menghan will continue to improve our stereo vision system with regard to multi-object tracking and localization. By the next PR, we need to demonstrate an integrated system. This will allow us to focus on fine-tuning our system in the final weeks of the project.

4. Future Plans:

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

1. Filtered data from all sensors
2. Multiple-object simultaneous tracking
3. Vehicle egomotion using GPS and/or OBD port velocity data
4. Preliminary sensor fusion and system integration

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