ILR #4: Progress Review 3

Harry Golash – Team A

November 11, 2016

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

1. Individual Progress:

1.1 Tasks:

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

1. Acquire data from the radar
2. Test cameras outdoors
3. ~~Test stereo vision using Apriltags~~ Calibrate the cameras using checkerboards
4. ~~Test the camera housings for harsh weather durability~~ Fix 3D printer and redesign camera housings
5. Write program to trigger and acquire camera data from both cameras
6. Complete the PDS PCB
7. Test object detection algorithms

The tasks I was originally responsible for were the third, fourth, and fifth ones in the list above. In the end, we accomplished the tasks shown in blue for this progress review, and postponed the tasks in red for future reviews. I ended up redesigning the camera housings to use an aluminum square channel for better heat dissipation, and I worked on the GUI/Software implementation for controlling and triggering the cameras.

1.2 Implementation:

1.2.1 Triggering the cameras simultaneously:

Before we could test our stereo vision capability, we needed to calibrate the cameras to determine their intrinsic and extrinsic properties, and we also needed to ensure that we could trigger them simultaneously. Yihao and Menghan worked on the calibrating the cameras while I developed the software implementation for triggering the cameras using a pulse signal (1.2.3).

For the source of the pulse, I used an Arduino Uno R3 from the lab. The cameras were programmed to take pictures when a digital "high" input was received via the GPIO inputs of the cameras. A 12 V DC power supply adapter and the Arduino output were wired to the cameras using a solderless breadboard. Both GPIO pin inputs were wired using a common ground to the same Arduino PWN pin (non-PWM pins only triggered the cameras infrequently). This way the cameras triggered simultaneously every time.

I wrote a simple C# GUI similar to the one I wrote for the sensor-motor lab, so that I could test the triggering of the cameras. To test the triggering, I focused both cameras on a single computer monitor displaying a stopwatch. The cameras captured the same time every time, down to the millisecond, in over 20 tests. Assuming the monitor has a refresh rate of 60 Hz, we have a time delay of less than 17 ms.

1.2.2 Fix the 3D printer and redesign the camera housings:

Initially, we hoped to test and use 3D printed camera housings. However, both the 3D printers in the lab have recently been malfunctioning. Given the large amount of time it takes to 3D print even small parts, I couldn't afford the risk of a losing many hours to bad print jobs. So, I did two things – (1) I redesigned the camera housings to use an aluminum square channel that I bought from McMaster Carr, thereby decreasing 3D print time for other (smaller) parts of the housing, and (2) I got the old third 3D printer from the Fence(d) Lab and troubleshot it and set it up so that we could use it.

The third 3D printer initially did not recognize button presses, making it useless. After taking apart this printer, I initially thought that the PCB connected to the button input was malfunctioning (what appeared to be a force/pressure sensor board). I removed this sensor board and then used a 200-ohm resistor to hotwire the main circuit board to get the 3D printer running again. This worked. However, upon switching out the pressure sensor board with one from another 3D printer, the problem persisted. Eventually, I found the cause of the problem was to be the rubber buttons themselves, strange as that is. When we switched the rubber button pad of the third 3D printer with that from another printer, it worked as usual. Moreover, the rubber button pad from the third printer did not work with the other two (previously functioning) 3D printers.

The shape and functionality of the old camera housing was maintained in the new design. However, to address the heat dissipation problem of the cameras, I chose to use a square aluminum channel as the primary body for the camera housing. The idea now is to use a 3D-printed face (with a clear polycarbonate sheet) and a 3D-printed rear hatch (with an embedded rubber sheet acting as a valve) to ensure that our housing passes IP 54 enclosure standards. This means that our housing will be splash proof and dust proof. Currently, I am experimenting with treating the 3D-printed parts using acetone and acetone vapors to make those parts water-proof.

2. Challenges:

When the trigger method was initially set up and tested, I used a non-PWM pin from the Arduino to trigger the cameras individually. When I wired both cameras into one circuit and to one trigger pin on the Arduino, I was only occasionally able to trigger the cameras simultaneously or at all. Changing the trigger pin to a PWM pin solved this issue. I suspect there was a current draw limit from the initially chosen pin on the Arduino.

The initial C# GUI/code to interface with Arduino was very simple – click a button, trigger the cameras. Upon using the PointGrey SDK for the cameras, the program grew a lot more complex. I am still working on setting up the camera parameters via the code and then setting up a trigger and capture routine which will eventually interface with our detection and tracking programs, such as the detection program Yihao is working on.

The 3D printers could be more reliable. The print quality and probability of job completion have been uncertain lately, even with the newly functional third 3D printer. Acetone dissolves our ABS parts slowly and at a non-linear rate, which makes it hard to determine the best amount of time for which to treat a given part.

My car was in the shop during the last two weeks, which meant we couldn't mount the rack onto it. Additionally, I was unwell for most of the last week from a chronic health issue (sleeping disorder) which prevented me from being productive.

3. Teamwork:

Our team worked harmoniously and effectively over the last week. Amit and Zihao worked on getting the radar up and running, with small contributions from me towards Zihao's Visual Studio code. Zihao also worked on the PCB board layout and design. Yihao and Menghan worked on testing the cameras outdoors and testing various object detection algorithms using OpenCV. They also worked on calibrating the cameras, which we will need to do once again once we have the cameras mounted within the housings. By working in parallel in this way, we were able to combine our efforts efficiently and save time.

Additionally, given the peer evaluation feedback from both the Systems Engineering and MRSD Project courses, we were able to clear up interpersonal miscommunications that had hindered us in the past. Right now, it looks like everyone is on the same page and is working hard towards the progress of this project. As I mentioned previously, I was unwell for most of last week. As a result, Amit played a major role in managing the team in my absence.

4. Future Plans:

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

1. Meaningfully visualize data from the radar
2. Calibrate and test stereo vision (may involve Apriltags)
3. Test camera enclosures for IP 54
4. Finalize radar mounting solution
5. Display stereo vision output in a GUI
6. Test our PDS PCB in a car if possible

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