Carnegie Mellon University

Individual Lab Report #10

Author:
Yuqing Qin

Team C: COBORG
Gerald D'Ascoli | Jonathan Lord-Fonda | Yuqing Qin | Husam Wadi
Feng Xiang

Nov. 11, 2021

Table of Content

1 Individual Process

The Coborg platform is a wearable robotic arm that can help people hold objects overhead. In this semester, my work in the Coborg project mainly focuses on the Perception (Vision) Subsystem upgrade. From the last progress review, I have upgraded the software part for the vision system. By running two YOLO on two cameras with a moving average to smooth the output, the vision system could get the target position successfully. For this progress review, I worked with other teammates to integrate the subsystem together and test the integrated system with our full use case.

1.1 Vision System Update During Integration

During the past two weeks, I worked on some power issues related to multiple cameras. When I developed the subsystem, I simply launched three cameras(T265, D435i) in separate terminals. This week, we are doing the integration, so we have to put all the nodes in a single launch file and only need to launch it once. During the integration testing, we have seen several times that the camera is failed to launch the first time, and we have to run the camera launch file in separate terminals to fix the issue. By doing some research online, I found out that this is a known hardware issue with multiple Intel Realsense cameras launching at the same time, and there is no official support on this issue yet. After checking several potential solutions provided by other people, I worked with Jason and solved the issue by adding a 1-second delay between each camera launch node. The issue is most likely related to the power and the USB connections. By adding the delay, it can give the system a little bit of time to recognize the USB port with a different camera. By running 20 repeated testings (i.e. relaunching our main script several times), it turns out that the current solution can solve the problem with all of the 20 testings launching multiple cameras successfully. Figure 1 below shows the tf frames with all of the cameras launched successfully.

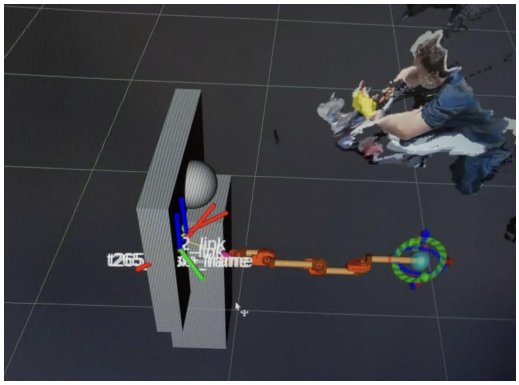

Figure 1. Camera frames when launching all together

Another main update is related to the speed and accuracy trade-off. After adding the moving average, it can smooth the output. However, considering the running speed, I only used 3 as the moving average window size. After doing the integration with other subsystems, I found out that the whole system ran much slower, even with a smaller window size. Therefore, I switched to

YOLO tiny version and at the same time increase the window size to be 10. Even though the tiny version would decrease the accuracy of hand detection, with larger window size, it can improve the accuracy a little bit. However, by using the tiny model, the running speed increases a lot, from 4FPS (full version) to 20 FPS (tiny) when running the full pipeline. Currently, I am still continuing on the window size tuning.

1.2 Integration

During the last two weeks, we all focused on integration. The vision system is the main component of our system. Jason and I worked together to integrate the vision with the motion planning system first. Also, since the cameras are mounted on the robot, we have to recalibrate the cameras again to make sure of the performance. The current camera configuration is almost the same compared to the last PR except for the T265 camera currently is mounted on the back of the D435i cameras. Figure 2 below shows the hand positions and end-effector position when running the full use case. From the figure, we can see that the end-effector reaches the middle point between two hands.

Figure 2. Full Use Case Demo

2 Challenge

The main challenges I faced are about the accuracy and speed trade-off. By looking at our functional requirements, for distance error, it should be within 6 inches. The latency of the full use case should be within 5 seconds. After comparing these two requirements, I found out that for our use case, the distance might have a higher tolerance than the latency requirement. Therefore, I experimented with different model sizes with different smoothing window sizes to balance the accuracy and run speed.

3 Teamwork

Team Member	Teamwork Progress
Feng Xiang	- Worked with me on integration on vision and motion planning - Calibrated the camera system with me - Worked with the whole team to do the integration testing
Jonathan Lord-Fonda	- Worked on debugging the resolved rate - Worked with Jason on smart manipulation testing - Worked with the whole team to do the full use case testing
Gerry D'Ascoli	- Worked with Jason on resolved rate refining - Worked on the integration - Updated the voice system and add a voice feedback - Worked on new hardware assembly
Husam Wadi	- Worked on new hardware design and integration - Worked on 3D printed parts - Worked on the assembly of new hardware design - Worked with the whole team to test the full use case

Table 2. Teamwork for Coborg

4 Plans

In the next week before FVD, I would focus on tuning the parameters related to the vision system to ensure performance. During PR11, I saw a failed case on the vision system. I will also look into it and debug that issue. Also, dealing with the edge cases (i.e error out feedback) in my vision node is another main focus for next week. The vision system should transfer the error message to the main state machine and let the user know about the current stage. I would focus on this feature implementation as well.