Carnegie Mellon University

Individual Lab Report #7

Author:
Yuqing Qin

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

Sep. 29, 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 migrated the vision system to Jetson Xavier and optimized the vision system workflow. For this progress review, I worked on generating the vision system upgrade plans, including hardware upgrade, and software upgrade.

1.1 Vision System Hardware Upgrade Plan

Last semester, the vision system ran on a single depth camera (Realsense D435i), which showed the limited FOV for our use case. In our desired requirements, the system's task space shall cover 60 degrees in horizontal, and 100 degrees in vertical. By measuring the FOV of our current hardware setting, it can only cover 66 degrees in horizontal, and 42 degrees in vertical. Knowing that, our team decided to upgrade the hardware setting to enlarge the FOV.

During the last two weeks, I worked with Jonathan to determine the task space (i.e. target position), and hand space (i.e. target position and possible hand position) by running the simulation. By plotting camera space, we can clearly see the coverage of the FOV. Figure 1 below shows the camera space (in yellow) and task space (in blue) before the upgrade.

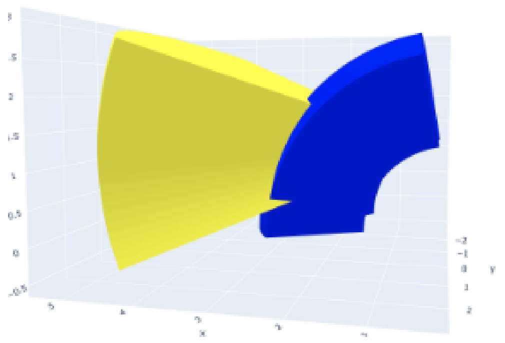

Figure1. Camera Space (Yellow) vs Task Space (Blue) before Upgrade

I also developed the trade study for the hardware upgrade. I proposed four alternatives to consider. The table below (Table1) summarizes all of the alternatives I proposed. In general, from easy implementation to hard implementation, we have considered 2 cameras approach and 1 camera with a motor approach (the fourth option can also be treated as another 2 DOF motor). The motor option could give more FOV, but it is more complex to implement and there will be a lot of edge cases and assumptions that we have to make in order to make it work properly. For example, if the human wears the camera, then we would need to assume that they will focus on the target part all the time and will not look around so that the camera could capture both hands. These assumptions would weaken our system's capabilities. Therefore, after comparing these alternatives, we decided to go with the first plan (2 cameras stacked). By confirming the plan with simulation (see Figure 2), we also shrank our requirements a little bit so that the cameras could capture all the possible hand positions.

Approach	2 cameras stacked	2 cameras on each side	1 camera with motor	1 camera put on the head
Horizontal FOV (deg)	66	120	66	180
Vertical FOV (deg)	~75 (~10 overlappings)	42	No limitation	180

Table 1. Trade Study Options (only show FOV impacts)

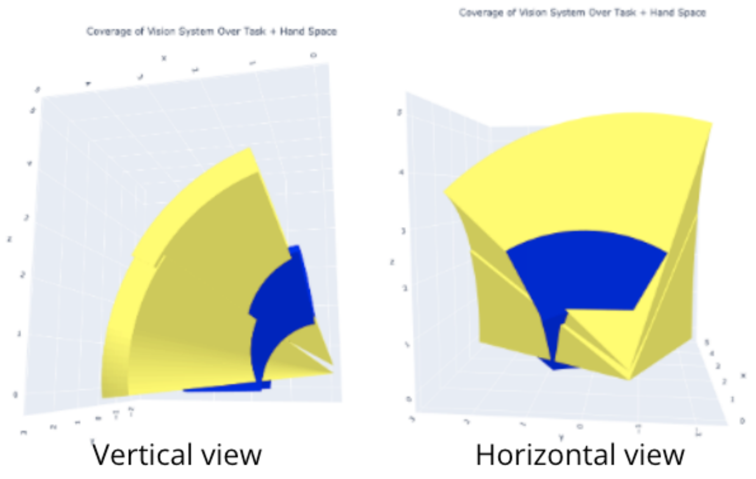
Coverage of Vision System Over Task + Hand Space
Coverage of Vision System Over Task + Hand Space
Vertical view
Horizontal view

Figure 2. Refined 2 Cameras FOV (Yellow) vs Hand Space (Blue)

1.2 Vision System Software Upgrade Plan

In the previous progress review, I have already simplified the vision system workflow. However, since we decided to go with the 2 cameras option, I have to consider the software upgrades again. Our current vision system uses YOLO v3, which achieves much more accurate detection compared to the tiny version. As a trade-off, the inference speed (~10FPS on Jetson) is much slower than the tiny version(~40FPS). Since we will work with 2 cameras, and it is possible that hands will be shown in both camera frames, YOLO will be running on two cameras at the same time. It would take a lot of power and computing resources. Therefore, I also prepared YOLO v3 tiny and v4 tiny models, which will be further tested after we have 2 cameras set up. By looking at the initial results on running the tiny models, it shows an unstable hand detection, which could be a big challenge for our whole system integration. Therefore, I am also working on optimizing the current YOLO v3 by using TensorRT to achieve a better run speed.

2 Challenges

The main challenges I faced are about the hardware upgrade plans. Even though we have decided the way to go, we still need to do a lot of iterations on the parameters and setup. These iterations will be mainly done by the next progress review. Also, even though we upgrade our hardware setup, we still cannot cover the full desired requirements. I discussed this with Jonathan about the shrink of our use cases. We came to an agreement at the end that we will limit our use cases, specifically in the vertical direction.

3 Teamwork

Team Member	**Teamwork Progress**
Feng Xiang	- Worked in manipulation system upgrade plan - Prototyped 4 DOF arm task space
Jonathan Lord-Fonda	- Worked on developing the task space simulation code for both manipulation system and vision system - Rewrite the system requirements based on the proposed vision upgrades
Gerry D’Ascoli	- Worked in obstacle avoidance demo - Integrated the obstacle avoidance to our current system
Husam Wadi	- Project management work - Worked on the CAD for the camera holder - Worked on the backpack upgrade

Table 2. Teamwork for Coborg

4 Plans

In the next two weeks, I will mainly focus on the iterations for the vision system hardware and software upgrade. For hardware, I will test with different positions around the one we got from the simulation. Considering the limitations on our hardware and materials, it might not follow the exact position parameters from the simulation. I will iterate on the positions in the next two weeks. Once the hardware is settled down, I will start to iterate the software upgrade plans, including testing with original YOLO v3, YOLO v3 tiny, v4 tiny, and v3 with TensorRT. By the end of the next progress review, the vision system should be ready to integrate into the whole system.