Carnegie Mellon University

Individual Lab Report #4

Author:
Yuqing Qin

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

April 1, 2021

Table of Content

1 Individual Process

The Coborg platform is a wearable robotic arm that can help people hold the target objects overhead. My work in the Coborg project mainly focused on the Vision Subsystem design and implementation. At the design stage, our team came up with the hand detection idea to indicate the target object location when people hold the objects. From the last progress review, I have shown the 3D bounding boxes from YOLO. Based on that outputs, I further post-processed those bounding boxes to retrieve the single goal position.

1.1 Post-processing Goal Position

From the last progress review, I have already demonstrated the outputs of 3D bounding boxes around hands. The 3D bounding box example coordinates relative to 'camera_link' are shown in Figure 1. To further extract the final 3D goal position, I post-processed all of the 3D bounding boxes based on different scenarios.

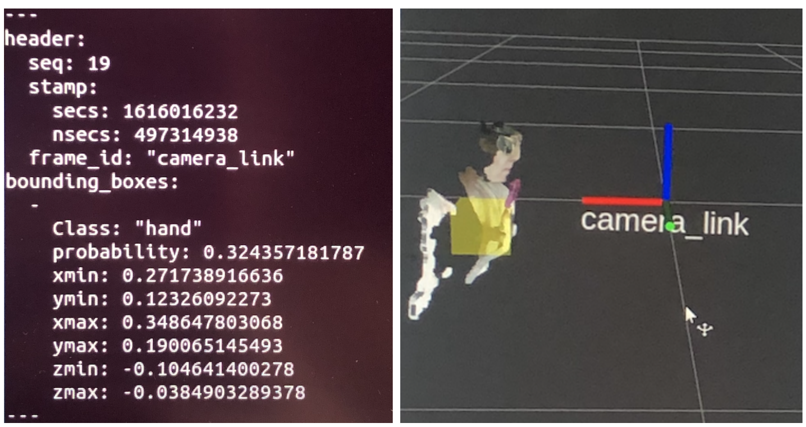

Figure 1. 3D hand detection ROS node implementation (left: 3D bounding box coordinates, right: 3D bounding box showing in yellow in rviz)

There are two scenarios that I considered dealing with. First of all, if there is only one hand holding the object, the system should extract the center position of the hand as the goal position. Once the motion planning system obtains the position, we can move our hand somewhere else so that the robot arm will not hit the hand. The second scenario is more than one hand holding the object. In this case, our system should take the averaged position as the goal position sending out to the motion planning system.

The post-processing result is shown in Figure 2. The post-processing step is implemented in one ROS node. It subscribes to the bounding boxes topic and publishes the single goal position to the '/goal' topic every timestamp. The output location is relative to the 'camera_link' frame, which can be further transformed to the robot arm frame in the motion planning system. The '/goal' ROS topic also shows useful information about how many hands showing in the frame.

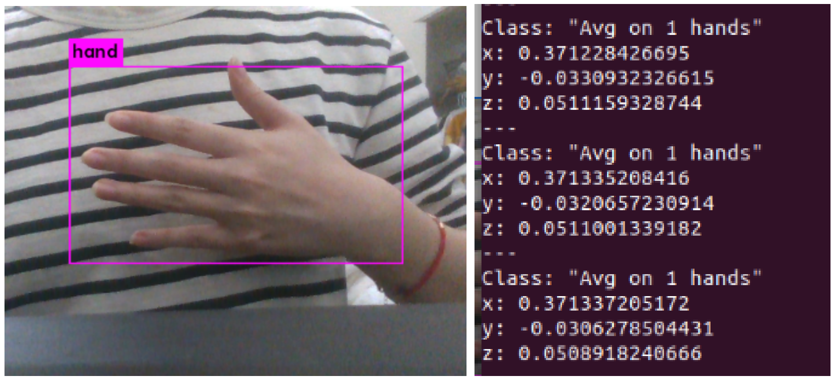

Figure 2. Goal Position Example

2 Challenges

The main challenges I faced in the last two weeks are related to the YOLO performance. I found that the YOLO detection is not that stable. It has some color dependency and texture dependency. Specifically, if the board color is similar to the hand color, it may not detect hands well. It may only detect one hand when two hands are holding the board. It is a common problem in computer vision tasks. However, fortunately, the performance may not affect our use case much. Our motion planning system will grab the goal position every two seconds, so even sometimes the YOLO is not stable on the detection, the overall performance of it is still acceptable. Even if it only detects one hand, it also falls into the first scenario I mentioned above. Post-processing can also deal with this situation properly.

Another challenge I found is the lack of GPU memory in the integration laptop we decided to use at the start of this semester. We just found out that the YOLO model takes about 4GB of memory to ensure its real-time performance. However, the integration laptop only contains 2GB of GPU memory. We have decided to buy a new integration laptop for our project.

3 Teamwork

Team Member	Teamwork Progress
Feng Xiang	-Created ROS Node to publish goal positions every second to MoveIt -Moved robot arm based on relative frame on URDF based on goal position rostopic

Jonathan Lord-Fonda	-Wrote SVD/FVD 1-pagers -Added the speaker to requirements and validation plans -Checked in with Gerry, Jason, and Yuqing to ensure validation plans still matched voice, actuated manipulation, and vision subsystems and updated validation plans -Began setting up validation testing -Ran through voice validation with Gerry -Ran through strength test -Started writing impedance controller for stabilization
Gerry D’Ascoli	- Designed power distribution PCB layout - Re-evaluated parts with Husam based on system requirements and Luis’s recommendations - Ran trial runs of Jonathan’s validation tests for the voice subsystem. - Fixed major issue with Voice Subsystem having too many false positives
Husam Wadi	-Created launch files for main node and voice node -Assisted Gerry with removing false positives in voice subsystem -Assisted Gerry with PCB design and refinement

Table 1. Teamwork for Coborg

4 Plans

For the next two weeks, I will first work on measuring the accuracy of the vision system. Also, I will work with Jonathan to do the Spring Validation Demonstration rehearsal on the vision system. By applying the vision system in different circumstances, I need to make sure the accuracy is good enough before integrating it with motion planning. Moreover, integration between the motion planning and vision system will be started once we finish measuring the performance of these two systems.