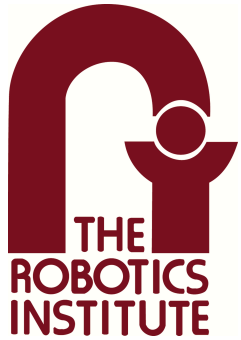
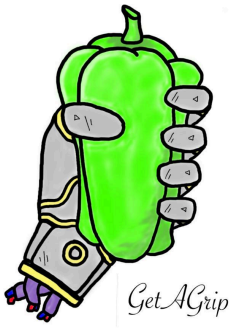


Carnegie Mellon University
The Robotics Institute, School of Computer Science

MRSD Individual Lab Report

ILR09: Progress Review #10



Team D: GetAGrip.AI

Teammates: Solomon Fenton, Jiyoon Park, Shri Ishwaryaa SV, Sridevi Kaza, Alec Trela

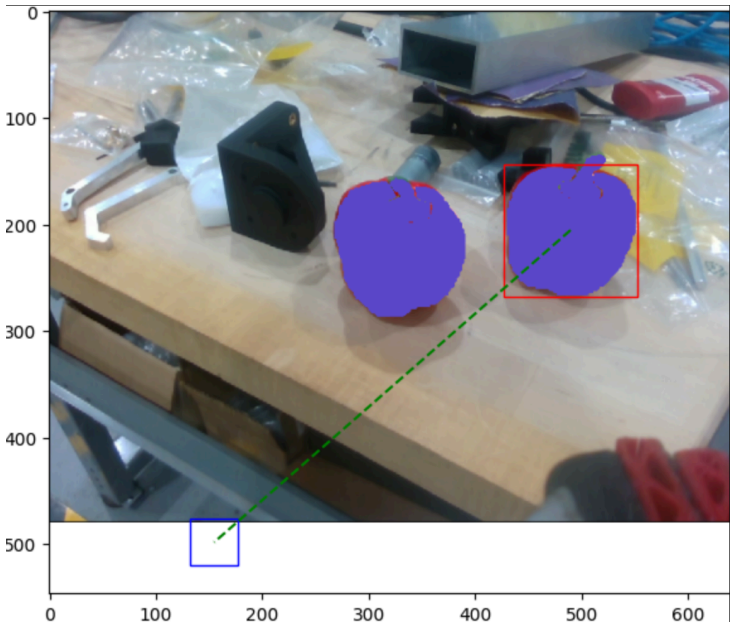
Author: Jiyoon Park

Submission Date: 11/02/2023

Table of Contents

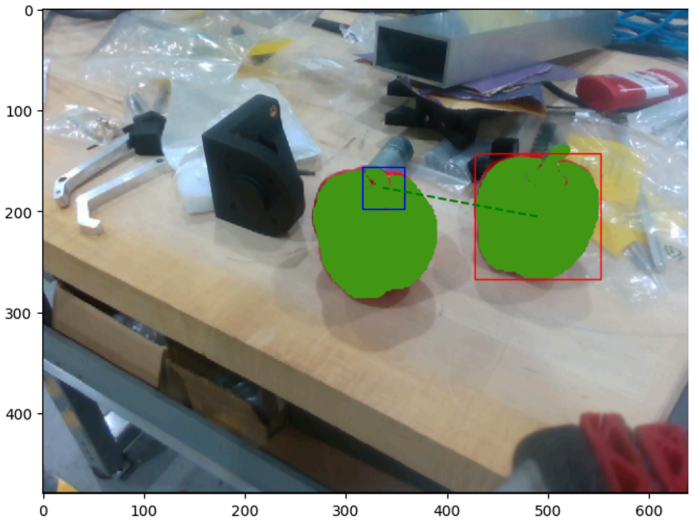
1 Individual Progress

During our latest project sprint, we mainly worked on making our perception code better and adding a user-in-the-loop system. This meant making the code work smoothly from start to finish and making it easier for users to interact with it. We faced some challenges and fixed them to improve how the system works.



[Figure 1] Axis translation problem. X, Y axis is not correct

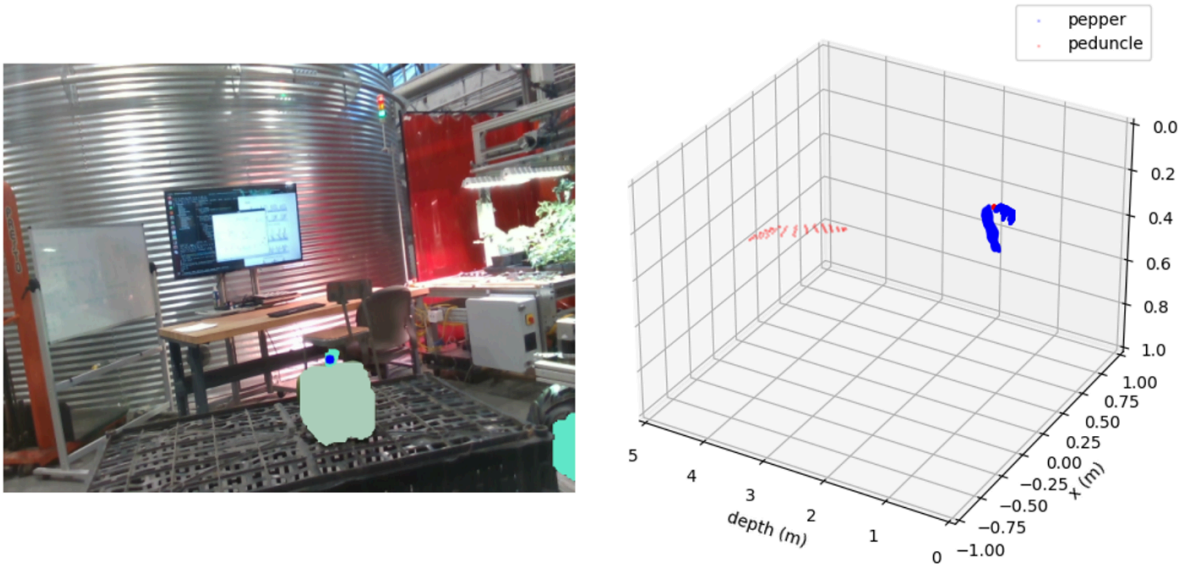
One big problem from the previous sprint was with pepper matching – getting our system to identify and match peppers and peduncles correctly. There were two main reasons for this problem. First, we discovered that the way numpy and OpenCV, two tools we use, handle their axes was different **[Figure 1]**. This issue started when we changed our code in the fall semester. It caused errors in how peppers were being matched. Another issue was with the way we were merging changes in our code. Since we did not review the changes thoroughly before adding them to our main code (we weren't using pull requests), it led to some things that needed to be corrected. For example, we had two ways of counting peduncles in the code, when they should have been unified and counted only once.



[Figure 2] All pepper fruit was being matched to the first detected peduncle

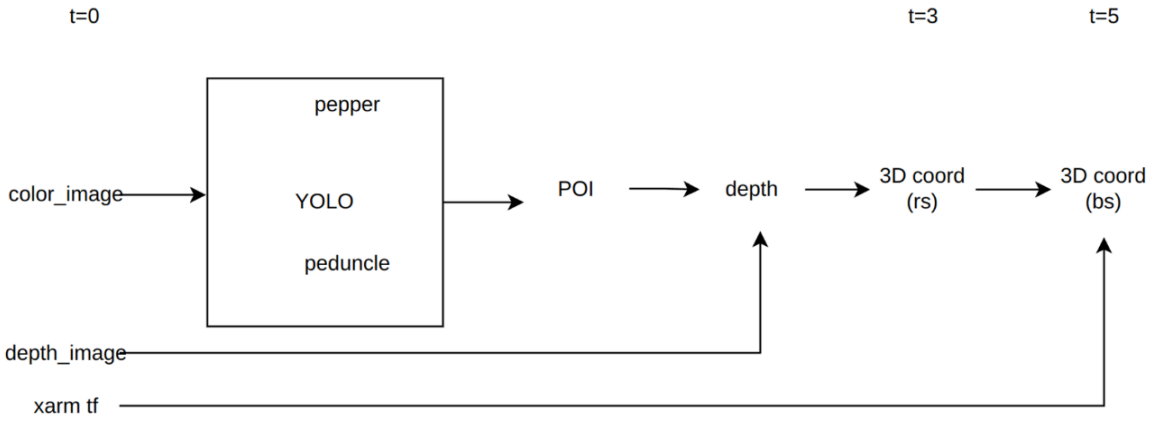
This mistake meant all the peppers were getting matched to the same peduncle as the peduncel count that was passed into making Pepper objects were never getting updated **[Figure 2]**. I solved this by ensuring the correct count was used throughout the code.

We also did a lot of work during our fall break. We compiled all the code updates and tested them to achieve our spring semester goal. We found and fixed two significant issues to achieve this. First, we had a problem with how our robot's arm (the manipulator end effector) linked with our camera (the Realsense frame). After fixing this, the robot's arm could go to the correct POI every time. The second issue was about getting the wrong depth information for peduncles. At first, I thought it was because of errors in marking (segmenting) the peduncles in our images. But after checking, I found that the problem was because the camera was moving **[Figure3]**.



[Figure 3] Inaccurate depth reading when arm is moving

The real problem was that the timing of when we took pictures, when we got depth information, and when we got other essential data (the tf) wasn't lined up. We were taking data from three different timestamps **[Figure 4]**. This mismatch was why our depth readings were off. After identifying the problem, we changed our code to align the timestamps between datas.



[Figure 4] Taking data from three different timesteps

In summary, this sprint helped us fix a lot of issues. We made sure our axes in the code matched up, fixed problems with how we were merging code changes, and aligned our timing better. All these improvements have made our system work a lot better.

2 Challenges

We did not face any significant issues throughout this sprint. There were minor issues that we had to solve on the fly.

3 Teamwork

Everyone has been very accommodating and supportive throughout this sprint. Although we didn't have an apparent division of who needed to do what task (which was natural as we were integrating and testing the system), the team tried to help where they could. **Solomon** has been working on the end effector subsystem by changing out the materials and getting the FVD testbed ready. **Alec** has been perfecting the perception subsystem by filtering out the noisy measurements. **Sri** had been filling out the missing parts of the manipulation subsystem by recording trajectories, adjusting speed, etc. **Ishu** has been working on getting the orientation of the peduncle so that we can harvest peppers hanging from diverse orientations. Aside from the things mentioned above, we all were integrating and testing our system end to end to ensure system robustness.

4 Plans

The system is almost complete and ready for FVD. We have tested the entire system today, and it robustly harvests the peppers. Tomorrow, we plan to test the system as if we were running our FVD. Although we expect the test to go smoothly, if any problem arises, we will create tasks to debug the issues accordingly.

Aside from that, I am working on completing the function for users to select POIs instead of having the perception system give the POIs. I am also designing and implementing the system GUI for the FVD. Finally we also need to retrain our YOLO model for our new testing location. But I expect these tasks to be done smoothly.