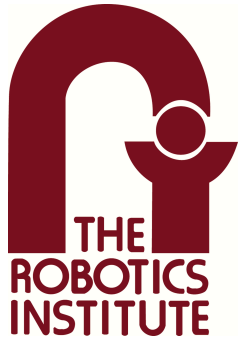
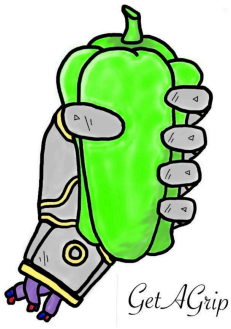


Carnegie Mellon University
The Robotics Institute, School of Computer Science

MRSD Individual Lab Report

ILR09



Team D: GetAGrip.AI

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

Author: Shri Ishwaryaa S V

Submission Date: 2 November 2023

Table of Contents

1 Individual Progress

In this sprint, my goals were to complete the integration of the new ROS architecture with the old SVD code and perform a successful SVD test, add features required for FVD, and test the full system on the simplest FVD testbed (visible peppers that hang downwards). The first task that I performed was figuring out the transformation from the last link of the end-effector to the camera. Alec and I utilized the CAD model of our end-effector and the CAD model of the RealSense camera provided in its ROS package to accurately determine the transform. In the past, we moved away from our SVD code to visual servoing because we were unable to perform camera extrinsic calibration properly. However, we realized that we made some mistakes earlier and the new transform worked very well, validating our SVD code.

The next task that I did in collaboration with Alec was to ensure time synchronization between the color image, depth image, and the transformation from the camera to the base link. Previously, we would access the color image, depth image, and the transform at slightly different times in our pipeline which led to wrong locations for the pepper and the peduncle. Hence, we fixed this by accessing all three objects at a single point in time and passing them to a function that does all the processing using these time-synchronized images and transforms.

Next, I also added code to the pipeline for handling edge cases such as the pepper fruit and peduncle having zero or maximum depth values. If a fruit has zero or maximum depth, there is no other way of obtaining its true depth so this fruit is not considered as a valid detection. However, if the fruit has a reasonable depth value and the peduncle's depth is incorrect, we limit its depth to be within some range of the fruit's depth. In the process of making these changes, I restructured the perception node to have the code compartmentalized into smaller, specific functions. This aids in understanding the code.

One main feature that I worked on this week is determining the peduncle's orientation. Most of the code for this was already written in the spring semester. It was modified and improved to suit the current pipeline. A curve is fit to the peduncle's mask, from which the point of interaction is chosen to be the point at a distance of half the total curve length from the top of the pepper. A point that is approximately ten pixels away is picked as the 'next point'. Both the POI and the 'next point' are obtained in the base frame coordinate system and their difference is determined by the peduncle's orientation vector. These orientation vectors are visualized in Rviz as shown in Fig. 1 using the PoseArray message type.

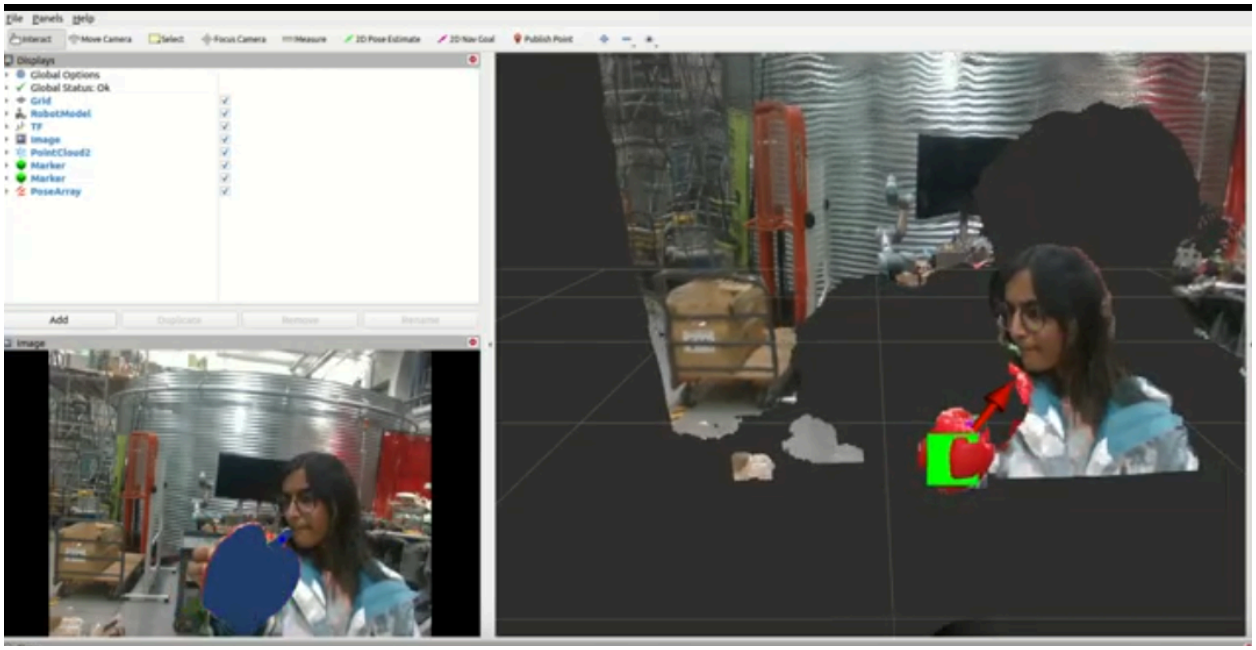


Figure 1: Peduncle orientation visualized in RViz (represented by the red arrow).

Finally, I actively participated in performing system-level testing in the NSH basement on the easy FVD setup. The perception and manipulation subsystems work together to consistently move the arm to the initial position, then to the chosen pepper, and drop the pepper in the basket before going to the initial position again. The end-effector was never actuated in all these tests. Therefore, we aim to perform proper FVD dry runs in the next two weeks. In addition, we will integrate the orientation feature with the full system and continue to do end-to-end testing.

2 Challenges

Some of the challenges that we tackled in this sprint were calculating an accurate transform from the base link to the camera, utilizing color images, depth images, and transforms with the same timestamp, and obtaining reasonable depth values for the fruit and the peduncle.

As mentioned in the progress section, we inspected the CAD models carefully to get a correct transformation from the last joint of the arm to the camera. We extract a color image, a depth image, and a transform from the camera to the base link at the same instant and use only these to do any processing. This fixes time synchronization issues. Finally, some simple checks are added on the fruit's and the peduncle's depth values to reject noisy depth.

3 Teamwork

Alec Trela

Alec and I collaborated a lot in this sprint. We worked together to get camera extrinsics, incorporate the time synchronization feature, and include a Kalman filter in our code. Currently, we are working together to output a stable orientation vector by passing the noisy ones through his Kalman filter node.

Sridevi Kaza

Sri and Solomon created the basket mount on the Amiga. She also recorded a trajectory for the arm to drop a picked pepper in the basket and go back to the initial position. Next, she also made sure that the arm moved up and down in its initial position to improve the chance of YOLOv8 detecting a pepper. She and I also spent a significant amount of time performing system-level testing in the NSH basement.

Solomon Fenton

Solomon worked on replacing the gears in the end-effector with commercial gears and ones that are cut from sheet metal. He assembled an end-effector with these metal gears and is working on creating spares. He also played a huge role in setting up our testbed.

Jiyoon Park

Jiyoon mainly worked on debugging the pepper-peduncle matching code, which is now completed. She also added a feature that allows a user to select a POI from a camera image. Some code in the perception node was restructured by her too.

Team

The entire team is working closely to be prepared for FVD. We have work blocks on Tuesdays and Thursdays. A short meeting is conducted before and after these work blocks to assess progress and assign tasks to everyone. We are also constantly testing our code as we make changes.

4 Plans

The plan for this sprint is to integrate the orientation feature with the system and perform system-level testing. I will be participating in FVD dry runs, contributing to the development of a GUI, and debugging any issues encountered during testing. We aim to have our system working robustly within the next week.