Alex Brinkman

Team D: Project HARP (Human Assistive Robotic Picker)

Teammates: Abhishek Bhatia, Lekha Mohan, Feroze Naina, Abhishek Bhatia

ILR #8: Progress Report

Submitted 2/25/2015

1. Individual progress

The UR5 arrived on Tuesday 2/23/2016 and drove a lot of my work since the last lab report.  To prepare for the arm, we migrated our lab equipment over to the cage and prepared a new branch for software development. I worked on making modifications to our framework to transition from software simulation to real hardware, configured the aruco_hand_eye package to solve for the transform between the manipulator and camera frame, and worked to get the UR5 communicating with our framework.

Several significant changes were required to connect to the UR5. In simulation, the Kinect was assumed to be fixed in the world frame since we had no means to actuate the location of the Kinect. On the UR5, the Kinect is attached to the end effector with a custom mounting bracket so the software configuration is updated to reflect that. There were some dependencies in other modules that relied on the fixed frame and needed adjustments to function properly.

We needed some way to empirically get the transform between the Kinect optical frame and the last link on the UR5.  First, we started by characterizing the camera's intrinsic parameters using the default Kinect calibration routines provided in the iai_kinect package. This improved the estimation between the optical frame and the ir depth frame and adjusted for distortion.  For the extrinsic calibration, we found the aruco_hand_eye package that uses an aruco marker tracker and VISP solver to calculate this transform.  I configured this to work with our system, tested the aruco tracker on the Kinect, and showed the solver could work – but without the arm this was all I could do to verify the accuracy of this package.

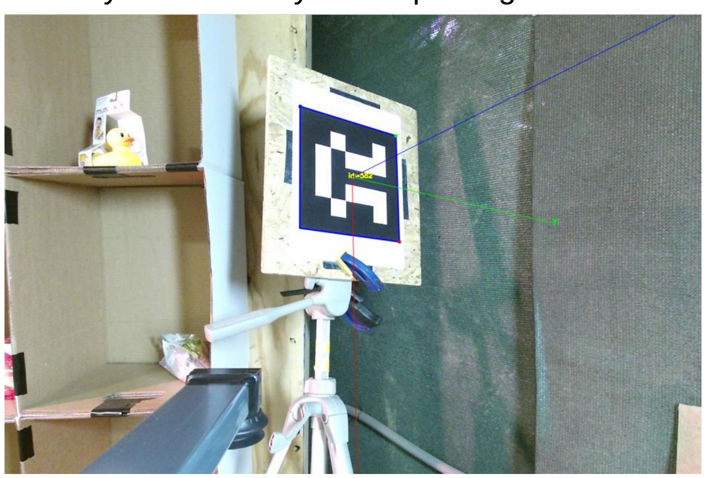

FIGURE 1: ARUCO TRACKER ESTIMATING MARKER POSE

Once the UR5 arrived, I setup the LAN with the UR5 controller and began the tedious process of getting the robot to communicate with ROS. After experiencing some difficulty, we found a change in the latest version of the UR5 controller v3.2

was not compatible with the latest ROS package provided by Universal Robot. We read forums and found a branch in the shadowhand forked repo that fixed this issue. Once this was fixed, we tested the default control of the arm and proved ROS control over the UR5.

Next, we needed to control the UR5 using our custom Moveit configuration from within our framework. We used the default test script provided to determine what changes were required to connect to the hardware. One short delay was our framework did not create a trajectory action lib interface needed by the UR5 to execute the planned trajectories and this was promptly fixed.

2. Challenges

The big challenge I am facing now is how to calibrate the camera extrinsic parameters. We were able to integrate the aruco_hand_eye package with a collection of 35 poses, but the results were not as accurate as our manual estimation. Below you can see the visual representation of the manual and computed camera extrinsics. The aruco_hand_eye results show a pretty sever rotation and the optical frames are outside of the physical camera.

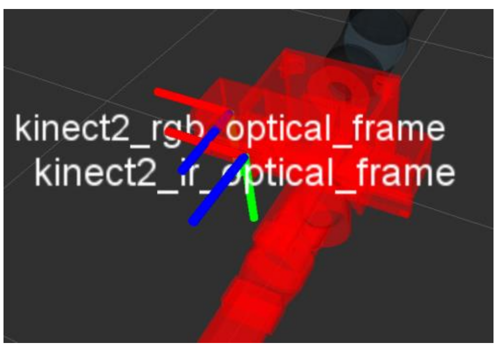


FIGURE 2: MANUAL CAMERA EXTRINSICS

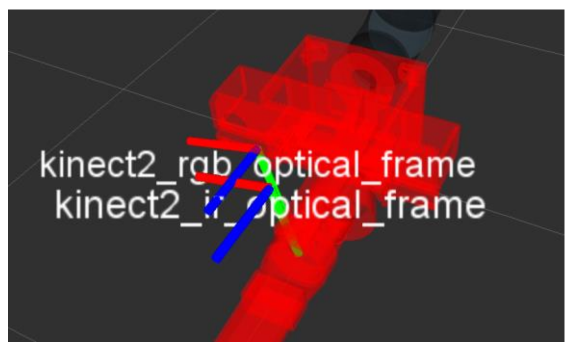


FIGURE 3: ARUCO_HAND_EYE CAMERA EXTRINSICS

We think the problem with calibration may be in the sampling of points taken, so we attempted to capture images from a more comprehensive set of 150 poses. After about 80 poses, the calibration results were diverging from the general ball park of what they should be. The next test will be with a smaller set of poses with Gaussian noise added to ensure we are getting more random poses and coving more of the areas of uncertainty. Another source of error may be the chosen aruco marker. There may be better markers to use to get more accurate homography from.

3. Teamwork

We all banded together this week to setup our new lab and get the arm working. I worked a lot with Rick to assemble the stand and increase its stiffness with the addition of stability plates. Rick and Abhishek worked on improvements to the existing perception pipeline and demonstrated a working way to extract color histograms to detect items. Feroze and Lekha worked on generating the offline grasping models and Feroze worked on generated ground truth models using the ROS packing Simtrack.

4. Plans

Going forward, we are working toward the single bin autonomous demonstration that will integrate all aspects of the system to run localization, perception, and grasping on at least one item on the real UR5. We will work towards completing the modules shown below in the perception and grasping pipeline. Green modules are completed, yellow modules mean work is underway, and red modules have yet to begin work.

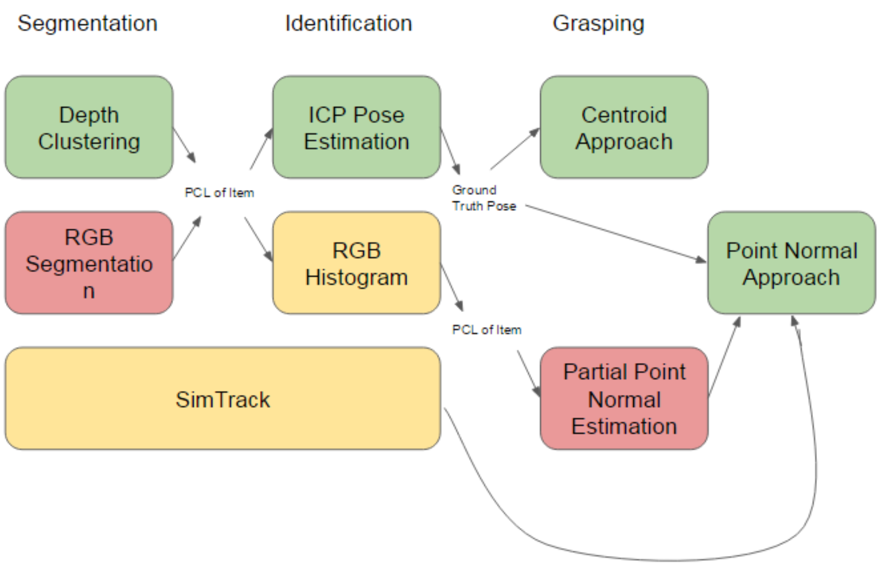


FIGURE 4: PERCEPTION AND GRASPING PIPELINE

I will be working on improving extrinsic calibration, integrating the grasp and perception modules into the framework, integrate the SBPL ARA* planner, and begin failure mode analysis and mitigation.