Feroze Naina M Dheen Mohamed Ismail

Team D - HARP

Teammates – Alex Brinkman, Rick Shanor, Abhishek Bhatia, Lekha Mohan

ILR08

Feb. 25, 2016

Individual Progress

Our main focus since the last progress review was to transition from the simulated environment to the real robot platform. The Universal Robots UR5 arm was delivered to us on 23rd February. I worked on testing the SimTrack perception package and developing grasping models.

Our current perception pipeline utilizes only depth data from the Kinect2 for identification and pose estimation of items without occlusion. However, this year the Amazon Picking Challenge rules were changed to include occluded items. So we had to develop a method to include RGB data along with depth data for handling occlusions.

SimTrack is a ROS package developed at KTH Sweden. It is a simulation based framework for detecting and tracking the pose of multiple textured rigid objects in real time. It combines dense motion and depth cues with sparse keypoint correspondences to estimate 6DOF poses. SimTrack uses SIFT features of surface texture for item identification and ground truth 3D model with annotated distances for pose estimation. It allows us to track custom objects with known 3D models. I wanted to verify if SimTrack can be used for our application.

Initially, I installed SimTrack and tested it with the included sample images – 3 pictures of ROS logos. The items were tracked simultaneously and pose estimates was published.

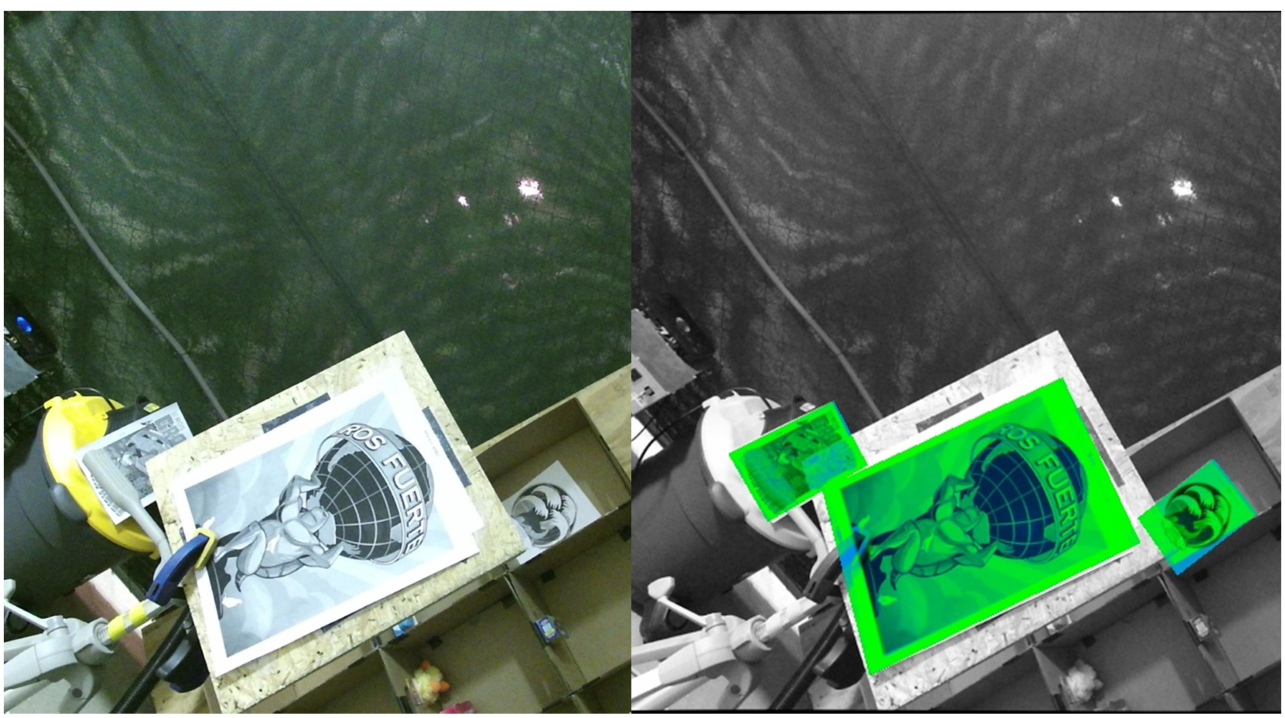

Figure 1 (a): Included sample images; (b): SimTrack tracking visualization

Figure 1 (a) shows the three sample images kept occluded and at different distances. Figure 1 (b) shows the items segmented and visualized by SimTrack. It was found to be quite robust to variations in lighting and occlusion.

Next, I tried building 3D models of some of our items – whiteboard eraser and box of straws. Autodesk 123D Catch was used to stitch together multiple images taken from a camera to form a 3D object. Figure 2 shows the generated 3D model of the Expo whiteboard eraser. The first image is the original object. The second two images are the 3D models visualized in MeshLab.

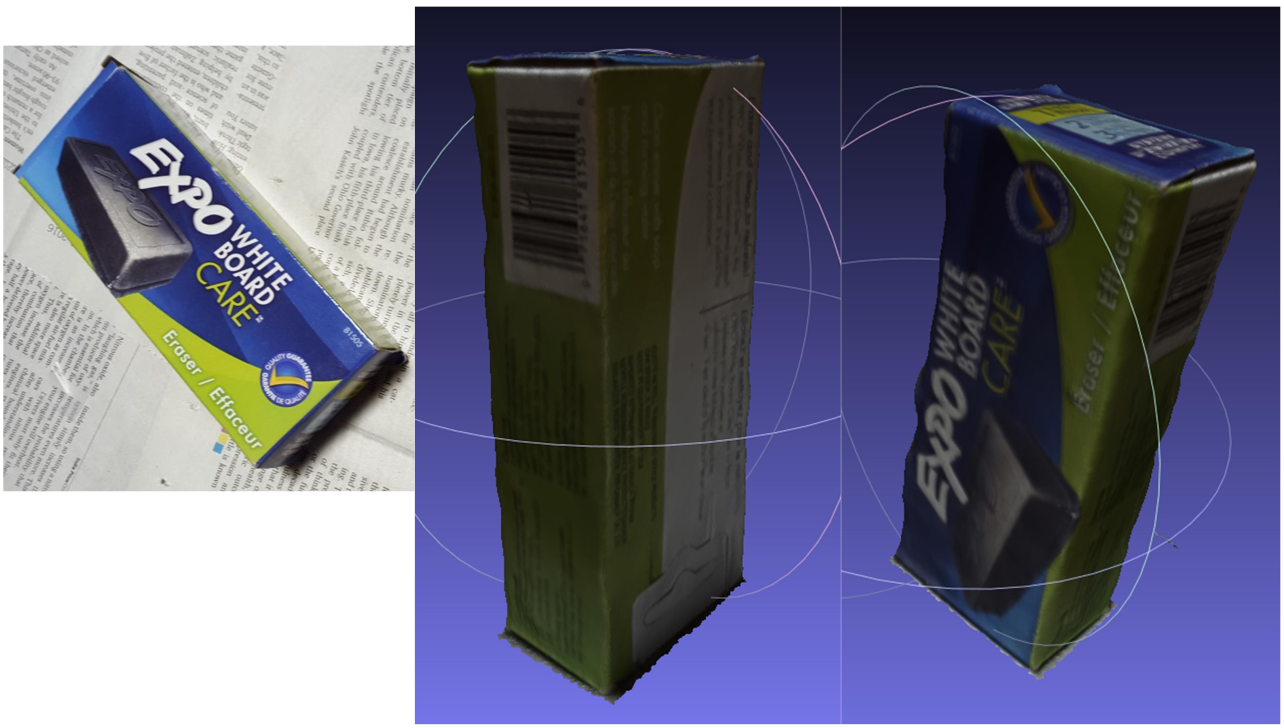

Figure 2: Generated 3D model white board eraser

For the grasping subsystem, we found that our current approach of estimating grasp surface was incorrect as we were measuring the normals of the surfaces with respect to the centroid of the pointcloud. We decided that it would be a better option to first develop a complete pipeline to manually generate grasp surfaces and test it with our integrated system. Later, if this was found to be insufficient, we would programmatically generate grasp surfaces.

Challenges

The main challenge I faced this week was to correctly generate 3D models. It required me to capture photos in a particular order and orientation. It was also hard to model objects with reflective surfaces. Some items had surfaces without features and 123D Catch wrongly stitched these images. Figure 3 shows a picture of an incorrectly modeled box of straws which had a white featureless surface. You can see that the edges are blurred and appear deformed.

Figure 3: Incorrectly modeled box of straws

I was unable to track the white board marker using SimTrack. I think the issue may have been because I used different cameras to model it and track it. I am currently working towards resolving this.

Team Work

The whole team worked towards integrating subsystems for testing on the UR5 robot arm. We also had an interview with the Amazon Picking Challenge team on 19th February to discuss our entry.

Rick and Abhishek worked together on improving the perception subsystem using RGB image data from Kinect2. The image was first converted to Lab color-space to prevent issues caused due to variations in lighting. The color features were extracted for multiple orientations of each item and histograms were generated. Items were then identified using nearest neighbor clustering. It was successfully demonstrated for 5 different items.

Alex worked on intrinsic calibration of the Kinect2 depth camera. He also worked to make the existing code-base run on the UR5 robot. He tested a teleoperation interface using the keyboard and Xbox controller.

Lekha worked on creating a GUI for choosing grasp locations for each item manually. The user can view the pointcloud and click on the correct positions for grasping. This samples all the points within a specified radius from the clicked point. These points are then taken as the valid grasp surfaces.

Plans

For the next week, I will be working on creating models of the different items and testing it with SimTrack. I want to confirm if SimTrack is useful for our system.

I will also be helping Rick to develop the grasping strategy for use with his RGB perception system. An altered version of grasp planner is required as the exact pose of the item is not known.