1 Overview

Project Summary

Clamps and "helping hands" are critical tools for skilled, dextrous work. They hold working items like circuit boards steady in a useful position. Wouldn't it be great if instead of painstakingly placing clamps and hoping they hold just they way you want them to, a robotic arm could take the item from you and hold it? This system would be more flexible and user friendly than a purely mechanical solution, but stronger and more tireless than a human assistant.

The goal of this project is to create a robotic system that:

- Perceives an item to be held
- Accepts human commands to grasp, let go, etc.
- Grasps that item while attempting to maintain the item's pose
- Exhibits compliant behavior that allows the human to reposition the item
- Exhibits rigid behavior to keep the item steady during work

Perception Component

This component of the system will focus on perceiving an object and determining initial grasp points for a simple robot hand.

Inputs RGBD data from one or more kinect sensors mounted above the workspace. Human inputs filtered through the human interaction module.

Outputs A conservative convex hull of the human arm. An estimate on the object's pose. A suggested position and normal vector for each of the two sides of the gripper.

Challenges The problems that perception will tackle:

- Identifying an object held up by a human
- Distinguishing where the human hand is (so the robot avoids it)
- Distinguishing the item's pose in 6-dof (so the robot can attempt to match it)
- Identifying and evaluating a potential grip candidate

This component will focus primarily on the last challenge.

2 Approach

Most approaches to grasping assume that the target item is just as a human would find it and use a plethora of methods to create a potential grasp, whether it is scanning the object and calculating force closure, learning from previous

grasps/human graps, or simply assuming nothing and acting off of image features. None of these methods place requirements on the item itself. However, most useful robotics take advantage of regularization of an environment. Humans are excellent at evaluating grasps, so why not meet partway and have the human suggest a grasp by marking two small dots on the item where it would be useful to pinch? This strategy might be considered 'cheating', but it may allow for faster and better grasping with little extra human effort. In the future, there's no reason why items couldn't be marked up to suggest robot grips inconspicuously with tags that are invisible to the human eye.

In this approach, the human would place two markers on the item to 'suggest' where the robot should grab. This is still a perception challenge, because the robot needs to identify the markers and place them in the context of the item's surface. These markers would be used as a heavily weighted feature in an offline learning algorithm where the robot grasps many marked items, finds the surface around them, and evaluates the grip based on whether the item shifts when gripped and moved. This way, it will know when suggested markers are a bad idea or the grip should be slightly different than the suggestion. In addition to the markers in the rgbd image, the learning algorithm would use cubes of voxels around the marks as features.

Possible Addition Integrating human feedback could be an additional novel part of the project. It's often clear to us which grip is good and strong and which one is terrible. Commands like "too loose" could reset the estimation of the grasp and, going even further, commands like "shift up a little" could modify the seed for the estimator.

2.1 Actual tasks

1. Write out the structure of the neural network for the learning algorithm
2. Create an image library of rgbd views of items with marks held by a human hand
3. Find the bounding box for the human hand/arm in the images
4. Estimate the pose of the item in the image (take 3 surface features, track them and create body axes with them to define a pose)
5. Divide images into features wrt the markers
6. Do offline testing to assign weights to features