1 Introduction

Clamps and "helping hands" are critical tools for skilled, dexterous work. They hold working items like circuit boards steady in a useful position. This seems like an obvious task for a robot - more flexible and user friendly than a purely mechanical solution, but stronger and more tireless than a human assistant. However, the task - holding items to aid dexterous work - has two components which often contradict one another: the grasp needs to hold the object rigidly, but also compliant when necessary. At the same time, it is constrained to avoid certain areas that may be delicate, or in the way of the human. The "Jarvis" project is an attempt to address the problem of providing workplace mechanical assistance in a distributed way - each subproblem is handled by a separate module.

This paper focuses on the Jarvis' perception module, which has two major goals:

- Identify the human to prevent painful collisions.
- Identify candidate grasp points on the target.

The perception module sends this information to other modules - path planning, human intent, and feasibility.

2 Prior Work

Many studies have investigated the problems of robotic grasp selection and human hand-offs. A recent review paper by Bohg et. al.[1] gives an overview of state-of-the art grasping approaches and breaks them down into three major categories: grasping known objects, familiar objects, or unknown objects. Grasping completely unknown objects requires 3D reconstruction of an object, tactile feedback,[2] or machine learning on observed features.[3][5] Our approach lies at the intersection of familiar and unknown grasping.

Human interactions and handoffs are often separated from grasp selection. Adding a human to the mix means that the object is no longer sitting still on a flat surface, so detection is harder. Arms and hands constrain the possible grasp points. Micelli et. al. used bounding boxes and partitions based on the human position to move the gripper near the target, but did not select grasp points.[4] Many algorithms exist to produce to detect humans and

The plethora of approaches to handoffs and grasping stems from their constraints, with each approach making slightly different assumptions about sensors, known information, and the situation.

3 Collision Avoidance

The main challenge that the collision avoidance algorithm has to overcome is turning the sensor data into an obstacle that will prevent the path planner from running into the human, while at the same time allowing access to the target item. This rules out pure bounding-box based methods, which are almost always too conservative and would prevent access to the target. Instead, we generate an "arm volume" based on skeleton tracking data.

The collision avoidance algorithm uses the elbow and hand points from the OpenNI skeleton tracking data to generate a vector that points from the elbow to the hand ($\mathbf{v}_{eh}$). It then generates a cylinder about this vector to conservatively approximate the volume of the forearm to avoid. This method doesn't specify the entire hand as an obstacle to avoid over-constraining the planner.

4 Grasp Selection

Jarvis uses a three step process to identify the target object and generate possible grasps:

1. Use the location of the hand and elbow to find a region of interest.

2. Perform a heuristic-based segmentation to generate a point cloud for the target object.

3. Select a grasp based on the target's orientation with respect to the human and its surface normals.

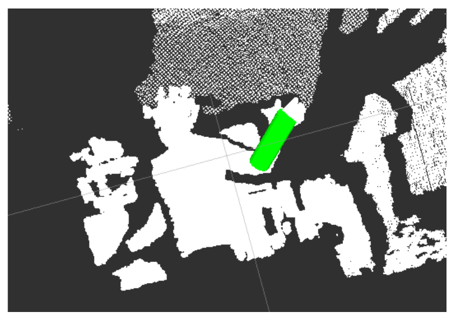

Figure 1: Jarvis should avoid the forearm avoidance region shown in green.

4.1 Area of Interest Detection

The first challenge in selecting a target without a prior model is even knowing where to look - large search spaces increase the probability of misidentifying a heuristically-modeled target. Jarvis takes advantage of the human interaction by assuming the target will be near the human's hand. The area of interest (AoI) is a 0.2 m cube that centers on a target point extended from the hand along the elbow-hand line.

Jarvis respects basic human cuing by checking whether the target is being held towards the robot or not - a common human indication of "grab this" vs. "mine!" Jarvis checks whether the elbow or hand is closer to the robot, and only proceeds in the latter case.

4.2 Segmentation

After identifying the AoI, Jarvis uses a set of heuristics to identify the target. Our application is geared towards soldering assistance, so the heuristic looks for a flat pcb board. It uses Random Sample Consensus (RANSAC) to fit a plane to the points in the AoI

$$ax + by + cz + d = 0 \tag{1}$$

and then filters out any point not on that plane. This procedure assumes that the target the largest plane in the small, floating AoI. While it puts restrictions on the target, it is able to adapt to boards of different sizes and shapes. This step can also be easily swapped out for a different heuristic-based segmentation depending on the application.

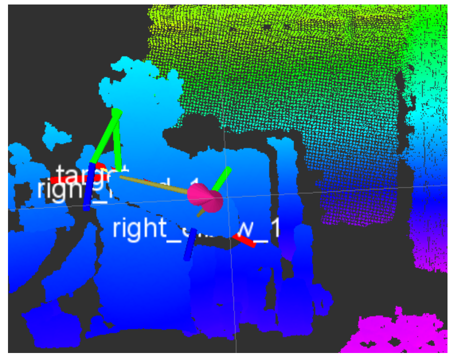

Figure 2: Display showing the center of the AoI (target) along with the detected hand and elbow centers.

4.3 grasp Generation

Jarvis represents grasps as a coordinate frame with its origin at the target point and oriented so that its z axis points directly towards the gripper and the x axis is along the gripper's closing direction (figure 2.) This representation assumes a simple pinch gripper, but could be used by more complicated grippers as well.

5 Discussion

The collision avoidance algorithm does well when the skeleton tracker gives consistent output, but has a tendency to locate a 4cm forearm in the human's head when the skeleton tracker produces noisy data. These tracking problems can be avoided by standing an appropriate distance from the Kinect (1-4m) and moving slowly. This suggests that future grip generators shouldn't rely on skeleton tracking data to ID human parts. One alternative method would be to use RGB data and OpenCV to either recognize hand colors or train a classifier to recognize the edges of the arm and hand and then project those

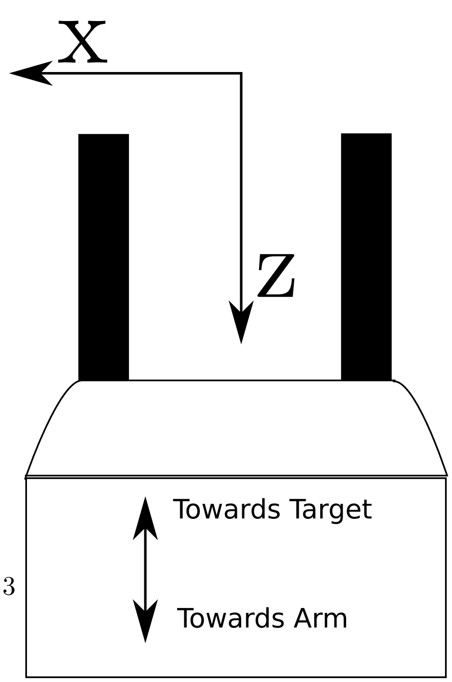

Figure 3: Grip coordinate system

pixels back onto their corresponding point cloud voxels in 3-space. Another alternative method is matching sets of segmented points to a preexisting model of an arm and hand.

Approaches not used

- Creating a convex hull model for the object after planar segmentation
- Implementing a kalman filter on the points in the object or
- Running a filter on the plane coefficients

6 Bibliography

References

[1] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-Driven Grasp Synthesis—A Survey. 30(2):289–309, 2013.

[2] Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E Gil Jones. Contact-Reactive Grasping of Objects with Partial Shape Information. pages 1228–1235, 2010.

[3] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep Learning for Detecting Robotic Grasps. January 2013.

[4] Vincenzo Micelli, K Strabala, and S Srinivasa. Perception and control challenges for effective human-robot handoffs. 2011.

[5] Ashutosh Saxena, Justin Driemeyer, Justin Kearns, and Andrew Y Ng. Robotic Grasping of Novel Objects.