High Level Representation of Kinesthetically Learned Motions for Human-Robot Collaborative Tasks

Heramb Nemlekar, Max Merlin, John Chiodini, Zhi Li

I. Introduction

Robots that assist humans in collaborative tasks need to frequently transfer and manipulate objects in an intuitive manner. However, most assistive robots still lack human-like task fluency, mainly due to their inability to infer human intent and fully understand the objects in play. Such intuitive behavior can be encoded in a robot's motion by kinesthetically training it for a human partner's movements [1]. This method can be further extended for interacting with objects by training the robot's response for an object's affordance. Affordance is the propety of an object that determines its manipulability. For example, a hammer used to hit an object and a book used to hit the object, both fall into the same affordance class of "used for hitting". Therefore this property can be utilized for planning the low-level grasping motions.

Early approaches to grasp planning train a set of basic skills on a robot, and attempt to reason a set of causes and effects; while Nguyen et. al. [2] proposes an approach based on Convolutional Neural Networks (CNN) to classify objects by affordance. But this approach requires large datasets to achieve high accuracy. Other methods approach grasping as a reinforcement learning problem. We follow a hybrid approach to grasp planning based on Active Vision techniques [3], where a 2D model is optimized using Elliptical Fourier Descriptors (EFD) [4] along with a force closure test. This method helps to determine the optimum points for grasping, for which trajectory can be planned. But the primary challenge for robots in collaborative environments is implementing human-like high-level abstract decision making by controlling low-level motions. This requires construction of a symbolic representation for evaluating plans composed of sequences of actions in a continuous environment [5]. Such a representation is created by reasoning about potential propositional symbols that describe the preconditions and effects of each action. The resulting representation can be expressed in PDDL (Planning Domain Description Language) that enables fast planning using a graph search planner for determining the optimal sequence of the learned actions.

This paper describes our methods and preliminary results towards developing a system that combines object recognition, grasp planning through demonstration and high level task abstraction into a comprehensive package for application in human-robot collaborative tasks.

[1] Heramb Nemlekar, Max Merlin, John Chiodini, and Zhi Li are with the Robotics Engineering Program, Worcester Polytechnic Institute, Worcester, MA 01609, USA

II. Methodology

A. Planning of low-level motions

The object recognition is done by correspondence grouping [6] and this initial guess of the object location is used to filter the point cloud. To simplify detection, pure colored objects are used. We use the pixel centroid of the object to find the corresponding 3D point on the point cloud and then calculate the 3D coordinates of the Kinect sensor frame. Grasp points were determined based on a representation of the object contour using EFD. EFD was chosen due to its invariance to noise, and ability to preserve desirable features of an object. A generalized model for the Fourier approximation of a contour can be shown as follows:

$$P_x(t) = A_0 + \sum_{n=1}^{k} (a_n cos\frac{2n\pi t}{T} + b_n sin\frac{2n\pi t}{T}) \tag{1}$$

$$P_y(t) = C_0 + \sum_{n=1}^{k} (c_n cos\frac{2n\pi t}{T} + d_n sin\frac{2n\pi t}{T}) \tag{2}$$

First derivatives compute the tangent vectors to the model. Normal vectors *N* can be found by normalizing the derivative of the tangent vector *Z*.

Curvature is the main feature used in selecting model grasp points, and can be determined as the sign of the dot product between normal and tangent vectors:

$$Curvature = sign(\|Z \cdot N\|) \tag{3}$$

An algorithm (Algorithm 1) was designed to find the grasp point pair residing in optimal curvature regions. We model the robot gripper as a pair of frictionless contact points. Grasp points must pass a force closure test determined by the geometry of the Fourier descriptor.

Algorithm 1 Compute Optimal Grasping Pair

```
Rank all possible pair sets by descending x + y curvature
for each set x,y with positive α do
    β = PerformForceClosure()
    if β above threshold return
Rank sets by ascending and repeat for negative α

procedure PERFORM FORCE CLOSURE(x, y)
    A = N_m1/||N_m1|| · (Pm_1 − Pm_2)/||Pm_1 − Pm_2||
    B = Nm_2/||Nm_2|| · (Pm_1 − Pm_2)/||Pm_1 − Pm_2||
    return A^2 + (π − B)^2
```

For reaching to the object that needs to be grasped we train Multi-dimensional Interaction Probabilistic Movement