Aalto University
School of Electrical Engineering
Degree Programme in Space Science and Technology

Usama Tariq

Robotic Grasping of Large Objects for Collaborative Manipulation

Master's Thesis
Espoo, September 25, 2017

Supervisors: Prof. Ville Kyrki, *Aalto University*
Prof. George Nikolakopoulos, *Luleå University of Technology*

Advisor: Rajkumar Muthusamy, *Aalto University*

Aalto University
School of Electrical Engineering
Degree Programme in Space Science and Technology

ABSTRACT OF
MASTER'S THESIS

Author:	Usama Tariq		
Title:	Robotic Grasping of Large Objects for Collaborative Manipulation		
Date:	September 25, 2017	**Pages:**	viii + 61
Major:	Space Robotics and Automation		
Supervisors:	Prof. Ville Kyrki, *Aalto University* Prof. George Nikolakopoulos, *Luleå University of Technology*		
Advisor:	Rajkumar Muthusamy, *Aalto University*		

In near future, robots are envisioned to work alongside humans in professional and domestic environments without significant restructuring of workspace. Robotic systems in such setups must be adept at observation, analysis and rational decision making. To coexist in an environment, humans and robots will need to interact and cooperate for multiple tasks. A fundamental such task is the manipulation of large objects in work environments which requires cooperation between multiple manipulating agents for load sharing. Collaborative manipulation has been studied in the literature with the focus on multi-agent planning and control strategies. However, for a collaborative manipulation task, grasp planning also plays a pivotal role in cooperation and task completion.

In this work, a novel approach is proposed for collaborative grasping and manipulation of large unknown objects. The manipulation task was defined as a sequence of poses and expected external wrench acting on the target object. In a two-agent manipulation task, the proposed approach selects a grasp for the second agent after observing the grasp location of the first agent. The solution is computed in a way that it minimizes the grasp wrenches by load sharing between both agents. To verify the proposed methodology, an online system for human-robot manipulation of unknown objects was developed. The system utilized depth information from a fixed Kinect sensor for perception and decision making for a human-robot collaborative lift-up. Experiments with multiple objects substantiated that the proposed method results in an optimal load sharing despite limited information and partial observability.

Keywords:	Grasp planning, Multi-agent grasping, Collaborative manipulation, Load sharing
Language:	English

Preface

بسم اللّٰه الرّحمن الرّحيم

I would like to express gratitude to my supervisor Prof. Ville Kyrki for his help and guidance throughout this thesis. I also want to thank my advisor Rajkumar Muthusamy for his help at every stage of this work. I also give my regards to everyone at Intelligent Robotics group who have always extended a helping hand at the time of need.

In addition, I would thank Tapolina Jha for helping me out in experiments that I could never have completed alone. A special thanks to my family and friends for supporting me and spacemasters for making last two year an unforgeable journey.

Finally, I would like to thank EU's EACEA and Erasmus+ programme for giving me the opportunity and funding to study in Europe for the Spacemaster program. My last two years here have been an unforgettable experience and a valuable addition to my profile.

Espoo, September 25, 2017

Usama Tariq

Abbreviations and Acronyms

AABB	Axis-Aligned Bounding Box
CG	Center of Gravity
EGA	Elementary Grasp Actions
HRI	Human Robot Interaction
MRS	Multi-Robot Systems
PCL	Point Cloud Library
RANSAC	RANdom SAmple Consensus
ROS	Robotic Operating System

List of Symbols

E	Equality matrix in quadratic programming
e_0	Equality vector in quadratic programming
$\boldsymbol{f}, \boldsymbol{f_1}, \boldsymbol{f_2}, \boldsymbol{f_t}$	Force vectors
$\boldsymbol{f_g}$	Gravitational force vector
$\boldsymbol{g}, \boldsymbol{g_1}, \boldsymbol{g_2}$	Grasp wrenches
h	Cost vector for collaborative manipulation
I	3×3 Identity matrix
O	3×3 Zero matrix
$\boldsymbol{o}$	Origin in object reference frame
$\boldsymbol{p_1}, \boldsymbol{p_2}$	Position vectors
Q	Cost matrix in quadratic programming
$\boldsymbol{w_t}$	Task wrench
x	State vector in quadratic programming
$\boldsymbol{\tau}, \boldsymbol{\tau_1}, \boldsymbol{\tau_2}, \boldsymbol{\tau_t}$	Torque vectors
ϵ	Scaling factor for a grasp wrench in cost evaluation
ω_1, ω_2	Scaling factors for torque vs force in cost evaluation

Contents

Chapter 1

Introduction

Expanding role of robots for manipulation tasks in domestic and industrial environments needs no introduction. Robots have been widely used for manipulation in industrial environments and are being increasingly investigated for use in domestic environments. Although industrial robots provide high throughput in production by efficiently repeating a single task in a pre-programmed way, the use of robots is limited to restricted environments with the expectation that the type of artifacts being manipulated remain within a narrow range of variance [1]. On the other hand, in a domestic environment a robotic system is expected to handle a dynamic range of objects, some of which might even be unknown to the system. However, in both cases, robots have to interact with humans for instructions, assistance or coordination to carry out their tasks.

The Human Robot Interaction - HRI problem refers to understanding the interaction between one or multiple robots and one or multiple humans to collaboratively accomplish a goal with the objective that the interaction is beneficial in some sense [2]. The interaction in HRI is not necessarily of physical nature; applications of the field span from remote interaction such as teleoperation of robots, to close physical interaction or social and cognitive interactions. However, technological advancements in artificial intelligence, computing power and robotic hardware have made physical or close proximity interaction between human and robots more practical [3]. Use of professional service robots designed to assist humans in work environments outside industrial settings [4] has increased substantially and significant research has been carried out in recent years to maximize utility of robots beyond traditional industrial setups.

In an environment where robots and humans are envisioned to share same workspace, physical human robot interaction is inevitable. These interactions will involve humans and robots working together on a single task. While the aspect of shared responsibility in such interactions has also been studied [3], better cooperation requires enhanced perception, analysis and decision making capabilities in these service robots. Advanced techniques have been developed for typical cooperation problems including human-robot object handover [5, 6], motion planning for cooperative manipulation [7] and other human-robot cooperative manipulation tasks [8].

Collaborative manipulation requires coordination between multiple agents handling a single object. While a multi-robot system increases the complexity in terms of planning and control, collaborative manipulation in human-robot interaction may use agility of human to ease strict coordinated control. However, a common goal in both these cases is the collaboration to manipulate an object that may not be manipulated by an individual agent. Thus both agents need to contribute in manipulation in order to accomplish the desired task.

Grasping plays a major role in manipulation of objects. In case of multiple agents, the grasp location of each agent with respect to the center of gravity of an object will be a defining factor towards the agent's contribution in load sharing. Therefore, along with control strategy, grasp planning is also crucial for better cooperation in both robot-robot and human-robot interactions.

1.1 Problem Statement

While a lot of work has been done on grasping in general, grasping of unknown objects and manipulation using multiple hands, field of grasping for multiple independent agents, lacks significant research. Particularly the problem of manipulating unknown target objects when multiple agents are influencing the target is unexplored. A common scenario in both domestic and work environments is collaborative lift and transport of large objects. Even if the object consist of simple geometric shapes, multiple independent agents handling a single object make the grasp planning problem non-trivial. Grasp planning with partial observability of the target and incomplete knowledge about other agents is addressed in this work to devise a grasping solution for the lifting task.

1.2 Motivation

The motivation for the proposed solution comes from human behaviour in collaborative manipulation. It is observed that during collaborative manipulation e.g. assisting in lifting and transporting objects, humans tend to take rational grasping decisions based on visual features of object. Furthermore, intuitive selection of grasp location is also appropriate for unknown objects that have not been previously handled by the person. Although a human decision is supported by years of learning and experience in object manipulation, the principle is based on basic laws of motion and inertia. If dynamics of the task being performed and adequate knowledge of the environment are known, a rather simple approach based on the laws of motions can be adopted to identify grasp locations that will minimize required forces and torques to complete the task.

1.3 Objective and Contributions

The aim of this thesis is to develop a real-time grasping solution for collaborative manipulation of unknown large objects. A collaborative lift is considered as the manipulation task. Depth cloud viewed from a fixed Kinect is taken as an input to the system. Partial depth information about the target object is then used to extract candidate grasps on the visible surface of the object. For an assistive role in manipulation, this system expects the other agent (possibly human or another robot agent) to first execute a grasp on the target object. For selection of best suitable grasp among candidate grasps, a new algorithm is proposed in this work that tries to minimize collective efforts needed from both agents to accomplish the manipulation task. Several configurable parameters allow tuning the method to incorporate capability and desired contribution of each agent. The proposed method is tested in an experimental setup with objects of different shapes and sizes.

1.4 Assumptions and Simplifications

A few assumptions have been made about the environment and target objects to simplify testing and verification of the system. To simplify target extraction process, the system expects a large target object on a planner surface in Kinect view. Objects consisting of basic geometric shapes are considered as target so that a simple grasp synthesis methods can be used for generation of candidate grasps. For the method proposed for collaborative grasp selection, it is considered that the extracted candidates result in stable grasps. Furthermore, due to immobility of the manipulator used for experiments, it was assumed that an optimal solution lies within the reachable workspace of manipulator.

1.5 Structure

The rest of the thesis is organized as follows. Chapter 2 provides a brief overview of grasp theory with further insight into grasp planning approaches for unknown objects. Recent studies in the field of cooperative and multi-agent grasping are discussed in Chapter 3. A new grasp planning method for collaborative manipulation is proposed in Chapter 4 and it's mathematical formulation is presented. Chapter 5 elaborates the design and implementation of the system, hardware and software components, and processing pipeline of the real-time collaborative grasping solution. Experiments to test and verify the components of the system are discussed along with the acquired results in Chapter 6. Finally, Chapter 7 concludes the work and provides future directions.

Chapter 2

Grasp Theory

Grasping and manipulation is a key feature to increase applicability of robots in domestic and professional environments. In recent years, significant technological advance has been achieved in the development of relevant hardware and its utility for better grasp execution. Availability of dexterous hands and tactile sensing has made it possible to imagine human like grasping and manipulation capabilities for robots in near future. As a result recent studies in this field have focused on more generic solutions, maximal utilization of available sensory information and real-time execution.

This chapter gives an overview of grasp theory and different factors that influence grasping solutions. In latter part of the chapter, approaches for grasping unknown objects are further discussed.

2.1 Grasp Synthesis

A grasp or grasp configuration can be defined with an appropriate hand configuration and/or contact points of fingers on the target object. Grasp synthesis is the process of finding a grasp configuration for a given object, which satisfies requirements of the grasping task [9]. Sahbani et al. in their overview of grasp synthesis algorithms [10] advocate that a grasp synthesis strategy should answer the question "where to grasp an object in order to accomplish a task?" along with ensuring stability, task compatibility and adaptability to novel objects. While the task to be performed is the primary objective of grasp synthesis, several other factors also influence the method that can be adopted for grasp planning. Bohg et al. [9] have identified these factors as shown in Figure 2.1. Major factors include type of gripper, prior knowledge about the target, available sensors and grasp synthesis approach.

2.1.1 Task Oriented Grasp Planning

Task oriented grasp planners use a task oriented quality measure to rank candidate grasps. The importance of grasp location in relation to a manipulation task is intuitive for humans, e.g. flipping a cylinder is easier if grasped from the center than if grasped from the top. For robotic grasping, an appropriate definition of task is also critical for task oriented grasp planning. Haschke et al. [11] specify a task with a single wrench, a wrench cone or a wrench polytope for task oriented grasp quality measures. In [12] a task was specified with sequence of desired poses of object and multiple external wrenches to show that grasp evaluation results in different quality for different tasks.

2.1.2 Type of Robotic Hand

The type of hand used for grasping also affects possible grasp configurations. While a simple gripper may simplify the planning problem, multi-fingered hands are usable with wide range of target objects. Critical factors to consider for a particular gripper are gripper pre-shape, number of contact points and type of contacts. Dexterous hands with multiple fingers result in multiple possible pre-shapes of hand and increased number of contacts, resulting in a larger number of possible grasp configurations.

2.1.3 Local vs Global Quality Measures

An infinite number of candidate grasps may exist for a single object. A quality measure is used to rank multiple possible grasps in order to select an optimal one. Relation of quality measure to the object is thus an important aspect for ranking. The measures can be local to the grasp location i.e. contact point, contact area or curvature of object around grasp location; or may depend on global characteristic of object e.g. principal component analysis or bounding box [9].

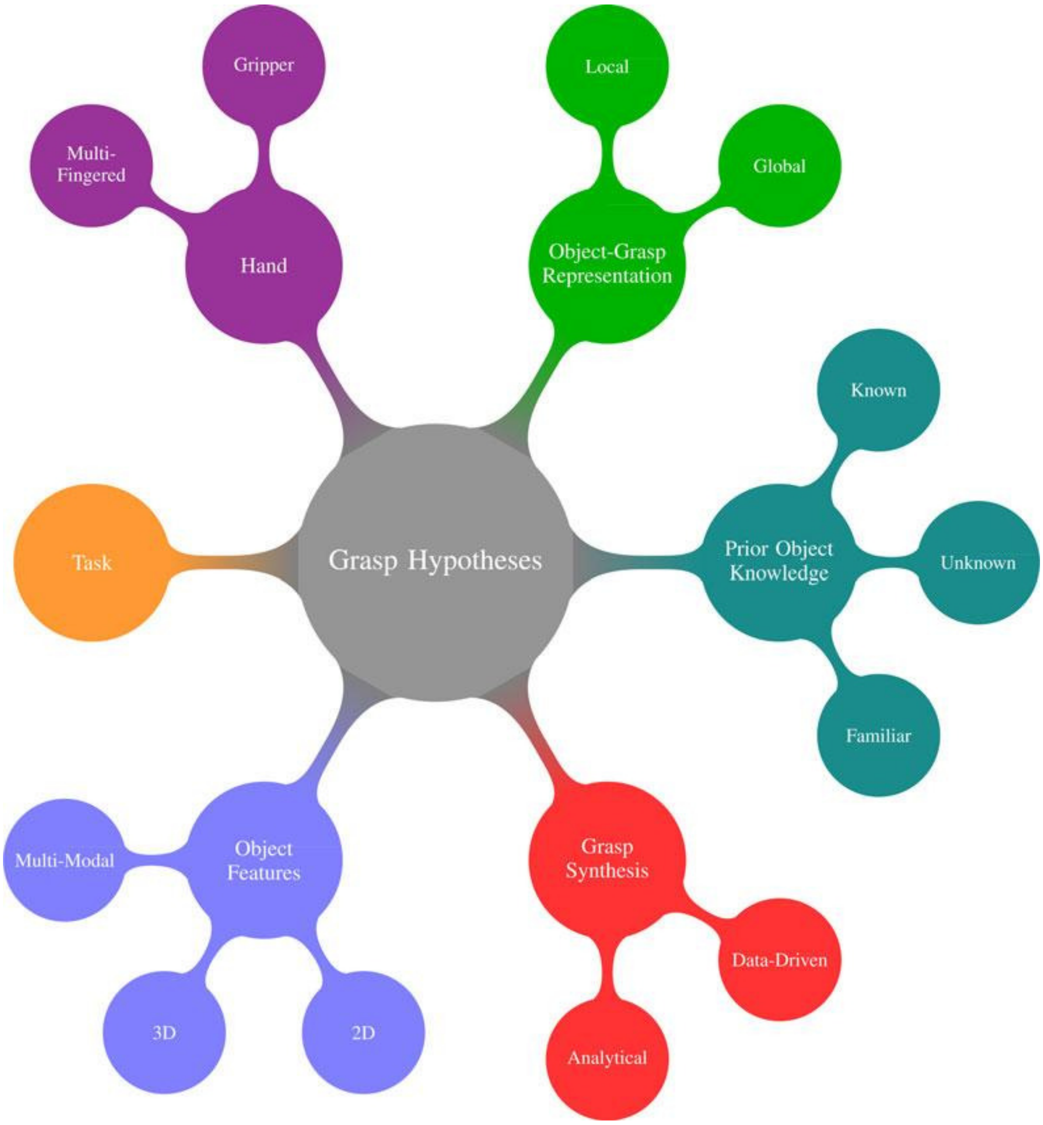


Figure 2.1: Aspects influencing generation of grasp hypotheses [9].

2.1.4 Sensing and Features

Another critical aspect influencing grasping methodology is observable features of target objects. Based on sensing devices, an object can be observed as 2-D images, 3-D data including depth information or other modalities. Furthermore, observation is also influenced by the mobility of the sensor. A fixed sensor will only be able to partially observe a static target object.

2.1.5 Analytical vs Empirical Approaches

Traditionally two grasp planning approaches are used [10]: analytical and empirical. Analytical approaches consider contact points between gripper and object to calculate force wrenches on the object as a result of a grasp. Force wrenches on target object are then analyzed for force closure [13] to ensure a stable grasp. Analytical grasp synthesis strategies thus try to make sure that a force closure is achieved on the grasped object. For the purpose, contact point normals are required to find the direction of resultant wrenches. Some quality criteria are used to select better grasps out of multiple force closure grasps. Quality criteria are usually based on ability of grasp to resist external wrenches in one or more directions. Figure 2.2 shows a common flow of analytical grasp synthesis. Both object and hand models are considered in grasp synthesis and a quality criterion is used to rank and choose the optimal grasp.

Empirical or data-driven strategies on the other hand try to plan a grasp by either observing the target object or by learning object grasping from human demonstration or repetitive grasp execution. Thus empirical strategies focus more on processing of perceptual data than grasp analysis. Major problems addressed in these strategies are object recognition and pose estimation, extracting features on target object and modeling [9]. Unlike analytical approaches, data-driven methodologies cannot provide a guarantee about grasp stability and can only be verified empirically. A common approach is shown in Figure 2.3. Sensing and signal processing is the key area of focus in these strategies. In empirical approaches a robotic system either observes human demonstration of task execution and tries to reproduce the same grasps, or learns the association between objects characteristics and hand shapes to compute a grasp solution [10].

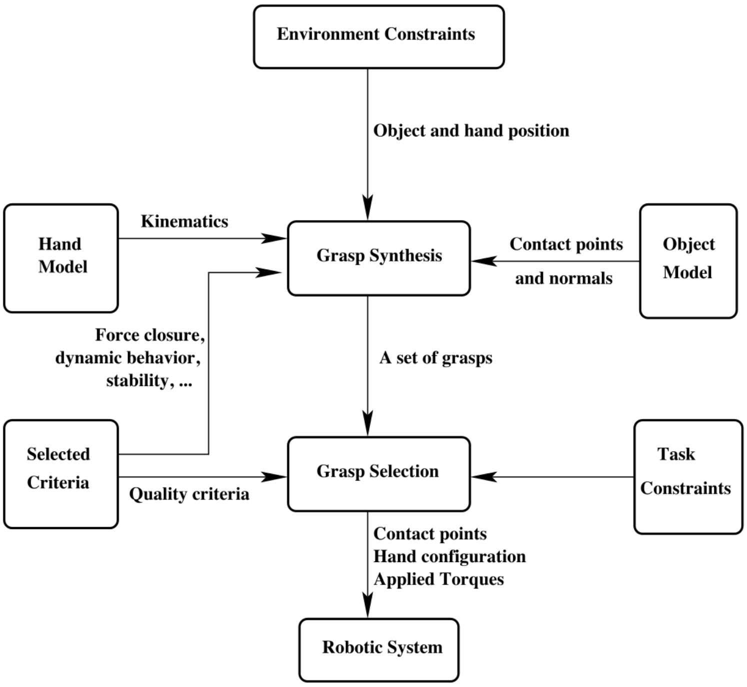


Figure 2.2: Typical grasp synthesis strategy in analytical approaches [10].

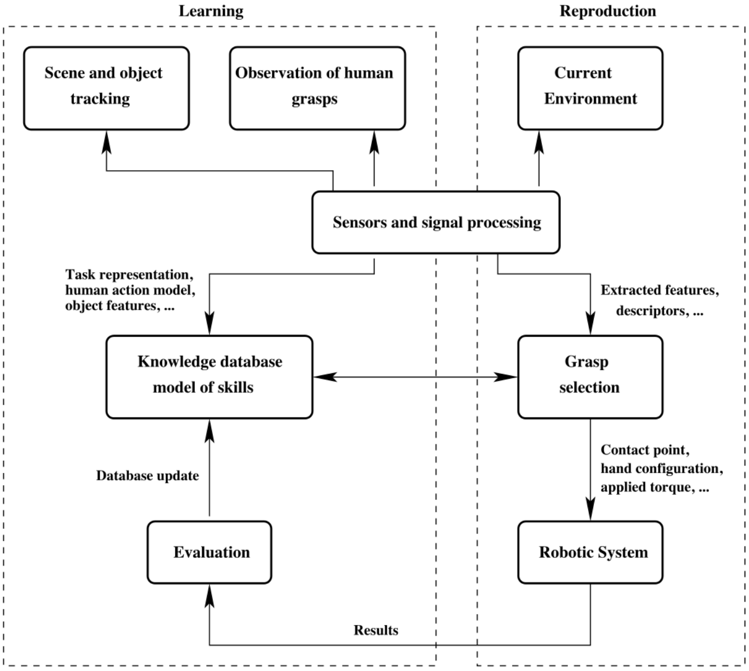

Learning
Reproduction
Scene and object tracking
Observation of human grasps
Current Environment
Sensors and signal processing
Task representation, human action model, object features, ...
Extracted features, descriptors, ...
Knowledge database model of skills
Grasp selection
Database update
Contact point, hand configuration, applied torque, ...
Evaluation
Robotic System
Results

Figure 2.3: Typical grasp synthesis strategy in empirical approaches [10].

2.1.6 Prior Knowledge of the Target Object

Prior knowledge of the target object plays a major role in dictating how a grasp synthesis problem is addressed. Grasp synthesis approaches in literature consider target object to be known, familiar or completely unknown [9]. In case of known objects, it is assumed that the target object has been encountered before by the system, or complete knowledge about object's shape is available. The system typically contains a set of candidate grasps computed off line for the object. Grasp synthesis problem is thus reduced to estimating current pose of object and selecting suitable grasp configuration.

Objects are considered familiar if the system has grasp experience for objects similar but not exactly same as current target. Approaches for such a scenario try to find similarities between known object models and the query object to apply previous grasp experience on current target. One example case is approximating target with a set of primitive shapes and applying primitive shape grasping experience to generate grasps [14].

Contrary to known and familiar objects, grasp planning methods targeting unknown objects usually consider no prior knowledge about expected model of the object [15, 16]. Instead these approaches rely mainly on information perceived by sensors. Methodologies for such scenarios try to link observed structure of the object to candidate grasps.

2.2 Grasping of Unknown Objects

An unknown object is an object never seen before by the robot system for grasping. The case is significantly different from known objects or familiar objects as those approaches consider available object model or other grasping experience [9]. Instead the sensed data has to be analyzed to generate candidate grasp solutions. Thus methods to grasp totally unknown objects focus more on data acquisition, processing and modeling of target objects for grasp synthesis.

Most common sensors used for robotic vision are stereo camera systems and RGB-D cameras to capture 3D structural details of the target object. Sensed data always contains noise and may also be incomplete. Furthermore the solutions need to consider how data is being acquired, e.g. in the form of images or depth point cloud. Similarly sensor's location may also be fixed or movable, resulting in perception from a single or multiple viewpoints respectively. Therefore methodologies for grasping unknown objects relying on sensed information need to take into account scene perception to extract required information that can lead to optimal grasp planning.

Considering preprocessing of acquired data before grasp planning, Bohg et al. [9] divide data driven grasp methodologies of novel objects in three categories:

- Approximating complete shape of unknown objects
- Grasps based on low level features
- Relying on partially observed shape of target

Latter two strategies do not rely on a complete model of object but differ in the way available partial information is utilized for grasp planning. Lei et al. [17] have thus categorized recent work for grasping unknown objects into two major groups: global and local approaches. Global approaches try to construct a complete 3D model of the object before extracting candidate grasps, whereas local approaches exploit shape features in available data that might help in finding suitable grasp locations.

In **global grasping approaches**, a complete model of the target object or a close approximation is constructed using either multiple views of the object, considering symmetries in object shape, factorizing objects into simple shapes or just by closing missing surface area in observation data to complete the shape.

Wang et al. [18] used a laser scanner attached to a multi fingered robotic hand to scan and reconstruct 3D object models before grasp planning. Different grasp configurations are then evaluated in simulation environment on a reconstructed model using wrench space metrics to choose the best grasp configuration before actual execution. Bone el al. [19] used a similar scanning concept with wrist mounted video camera along with a line laser to reconstruct a 3D model of the target object. Object silhouettes are extracted from 2D images to reconstruct an initial 3D model of the object, which is later refined by merging laser scan data. A force closure grasp is then generated by analyzing the object model. Dune et al. [20] used camera in hand to roughly approximate a 3D model of the object with quadratic functions, from multiple 2D views of the object. They assume that necessary features for a proper grasp are object's major axis, centroid point and size. A rough estimation of object model includes all these features.

If only one view of the object is available, the complete model of the object cannot be observed due to self occlusion, even with a depth camera. Such a scenario requires assumptions for occluded part of the object to complete the model. To make up for missing information Bohg et al. [21] made use of the observation that many target objects for service robotics are symmetric in shape. Assumption of symmetry was then used to complete partially observed model of object before grasp planning. Ilonen [22, 23] used same assumption for his work with Bohg and Kyrki for initial hypothesis of 3D model, which was later improved during grasping by optimally fusing tactile information with visual data.

To simplify grasp planning problem on complex objects, decomposition into simpler shapes has also been used for grasp hypothesis generation. Miller et al. [24] presented a framework for grasp planning on shape primitives. If a complex object is decomposed in smaller primitive shapes, same techniques can be utilized for grasp planning on subparts of the object. Huebner et al. [14] presented a maximum volume box decomposition algorithm to divide a given 3D point cloud data into primitive box shapes to be used for grasp planning. Schnabel et al. [25] used RANSAC algorithm to efficiently detect multiple basic shapes in unorganized point clouds.

The 3D model reconstruction problem has also been addressed by researchers in the field of computer vision and graphics [26–29] using multiple images or depth information without considering grasp planning task. Once a complete 3D model is available, different grasp planning approaches for known/familiar objects can be utilized to generate suitable grasps.

Contrary to global grasping approaches, **local grasping approaches** try to exploit local features such as edges or boundaries to generate candidate grasps. Local grasping approaches are more practical when only a single view of object is available as the global approximation in shape is no more required.

Richtsfeld et al. [30] used single view range data for grasping objects with cylindrical shapes or with flat top surface. High curvature points in visible point cloud were used to identify cylinders out of arbitrary shapes. Cylindrical shapes were tested for open or close shapes and side or top grasp was generated based on diameter of cylinder. For arbitrary parts on the other hand, only top surface grasps were evaluated. Lei et al. [31] also used partial point cloud data of unknown object from one or two view points to find grasp locations using force balance optimization. Partial point cloud was projected on 2 different planes and a suitable grasp location was calculated by maximizing force balance coefficients on projected contours of point clouds.

Ele et al. [32] used boundary information in a single 3D image to plan two finger gripper grasp on unknown objects. In other similar local grasping approaches Calli et al. [33] have used curvature information obtained from silhouette of an object; and Suzuki and Oka [34] used principal component analysis on partial point cloud for grasp planning on common household items.

2.3 Discussion

Grasp planning has been an active area of research over past few decades. Researchers have addressed the problem with different aspects for both known and unknown objects and have achieved significant progress towards both generic and specific solutions. Some of the recent works were discussed in this chapter. For the scope of this thesis, grasping solutions for unknown objects are of particular interest. Such solutions have shown to achieve acceptable success rate if the unknown target satisfies certain assumptions. To focus on the collaboration aspect of human-robot interaction, robotic grasping problem for an individual agent has not been addressed in this thesis. A candidate grasp generation method based on elementary grasp actions was used on targets consisting of simple geometric shapes. Aarno et al. [35] has shown that elementary grasp actions on low complexity objects result in up to 80% of successful grasps. Therefore during the collaborative manipulation, it was assumed that individual grasps are stable i.e. able to exert forces and torques in all directions on the target object.

Chapter 3

Cooperative Grasping and Manipulation

As discussed in Chapter 2, robotic grasping problem has been actively addressed in literature for different types of objects and environments. The solutions are typically limited to pick and place tasks by a single manipulator and gripper. Such solutions are applicable widely in service robotics where objects of different kinds are to be manipulated by a single agent. However, they usually does not consider influence by any external agent on the target object.

A wide range of manipulation tasks in human centric environments need an interaction between multiple agents or manipulators. Following sections discuss a few of such problems addressed in recent years.

3.1 Dual-arm Manipulation

Anthropomorphic robots have also gained popularity in recent times due to their applicability as a replacement to human workers without significant changes in workplace. Deployment of robots in human centric environments requires object handling abilities in robots similar to humans. This has led to an increased research in the field of dual arm manipulation which involves object handling by more than one robotic hands. Such a task requires a grasp solution for each hand. However, as most multi-arm systems are single agent, a centralized planning approach can be used to devise solution for both grasps simultaneously. Berenson [36] extended a multi-fingered single hand grasping approach to a two handed grasp by considering two hands as two fingers of single virtual hand. Rojas-de-Silva and Suárez [37] found a simultaneous grasping solution for manipulation of bulky objects with two anthropomorphic hands by slicing the target object's point cloud and evaluating quality of grasp combinations that satisfy force closure condition. However, the method required an existing model of the target object.

Such approaches are not applicable in distributed multi-agent systems as a distributed system is restricted by limited knowledge about the other agent's grasp. Besides, in human-robot interaction, communication beyond visual observation is impractical, restraining the possibility of a centralized grasp solution.

3.2 Human-Robot Handover

One of the problems addressed in recent works where an object is considered under influence of more than one agent is the hand-over task in human robot interaction. Human-robot object handover is a fundamental scenario of human robot cooperation in domestic environments. Since the object has to be transfered from a robot to a human or vice versa, solutions have to consider the handover phase when the object will be grasped by both the agents. The handover task completely transfers control of the object from robot to human or vice versa. Therefore, the grasp configuration for robot is chosen such that the pose of the object ensures stable grasps for both agents: giver and taker of the object. Even though the object will undergo a simultaneous influence by both agents in such manipulation task, the object will also be manipulated by both agents individually. Coordination between agents (human and robot) is of primary importance for a successful transfer. Strabala et al. [6] studied human-human interaction in terms of communication and coordination during a handover task to propose a coordination framework for human-robot handover scenario. In another interesting work, Edsinger [38] showed that the intuitive nature of human simplifies the grasping problem between human and robot, as humans tend to align the object according to position and orientation of robotic hand during a handover. Chan et al. [5] presented another framework to learn handover grasp configurations by observing humans handing over object to the robot. By observation, the robot was able to handover object back to the human.

Even though the handover problem represents a fundamental human robot interaction, the problem differs significantly from cooperative manipulation. A cooperation requires sharing required effort during manipulation between multiple agents. Thus both these cases will have a different optimal grasp solution. Besides, handover tasks typically involve small objects that can be handled by a single agent and do not require cooperative manipulation.

3.3 Collaborative Manipulation

Most existing literature in the field of collaborative manipulation mainly focuses on motion planning and control strategies [7, 39], assuming that a grasping solution already exists. Arai et al. [40] presented an assistance system to help transport long objects that are difficult to manipulate from a single point of support. The method focused on the control strategy for manipulation of the object rather than the grasping problem. It was assumed that the robot will grasp the object from one end and human from the other. Fink [41] and Mellinger [42] in their work addressed cooperative grasping and manipulation for aerial robots, but again the work was focused on planning and control of multiple aerial robots to manipulate or transport the payload in three dimensional space.

3.4 Decentralized Multi-robot Systems

Muthusamy [43] addressed the problem of cooperative grasp planning for decentralized agents in a multi robot system (MRS). He proposed a multi robot grasp planning method for coordinated grasps in a setting where agents do not have information about embodiment of the other agent. However, it was considered in his work that both agents have similar capabilities and the grasps were executed in a sequential manner. Moreover, both agents had knowledge that the object will be manipulated by multiple agents. Thus each grasp was chosen to maximize expected quality after both grasps. The approach can also be used in a human-robot interaction for manipulation as the intuitive nature of humans will also result in an appropriate human grasp if collaboration is expected.

Extending his work to task specific cooperative grasp planning in MRS [44], Muthusamy proposed task specific grasp planning strategies for decentralized multiple robots. He demonstrated that if task characteristics are known, task independent grasp planning is inferior to task specific grasps planning. However, the work considered grasp planning on known objects. If the object is unknown and partially observable, the quality measures used in his work may not be applicable.

3.5 Discussion

With increasing influence of robots in industrial and professional environments, human robot interaction and robot-robot interaction will be fundamental features in future robots. Such systems will require execution of tasks in collaboration with other agents. Manipulation being one of the elementary task in robotic applications will be a common scenario in collaborative execution. A significant amount of research has been carried out in the field of grasping by single agent and towards control strategies for multi-agent manipulation but the field of multi-agent grasping for manipulation task has received little attention.

The grasping and manipulation task becomes non-trivial when incomplete information is available about the object. Furthermore, if the object is to be manipulated by multiple distributed agents, an additional factor of uncertainty is introduced in the task execution. Incorporating robots in human centric environments to assist and work alongside humans will need robots capable of analyzing unfamiliar situations and human actions; and to make rational decisions despite limited information.

Chapter 4

Proposed Method

Collaborative manipulation requires agents to contribute in manipulation task in a way that the collaboration is beneficial in task completion. Therefore, it is considered that an optimal solution for a manipulation task must ensure task completion and minimize efforts required by the manipulating agents. In collaborative execution of task, if one agent has better capabilities such as a robot in a human-robot interaction, more contribution may be expected from the agent in an optimal solution.

The problem of grasping for collaborative manipulation of unknown objects is addressed in this work for a lift-up task on large objects. Limited prior knowledge of the environment is assumed and sensory information is utilized for logical decision making. A 3D range data of the target object from a single view is considered as input to the system. In case of large objects, partial view of the object is adequate for principal component analysis to approximate size and centroid of the target object. This work assumes that the target objects have uniform mass density. The center of gravity is approximated from axis aligned bounding box (AABB) of the target point cloud. Furthermore, to play an assistive role, a sequential coordination protocol is considered between manipulating agents where first agent makes a grasp on the target object followed by a decision for second agent grasp by the proposed method. It is assumed that approximate location of grasp by first agent can be observed from real time range data and the grasp location will remain constant.

The visible partial surface of the object is processed to extract candidate grasp locations. Chapter 2 discussed a number of approaches that can be used

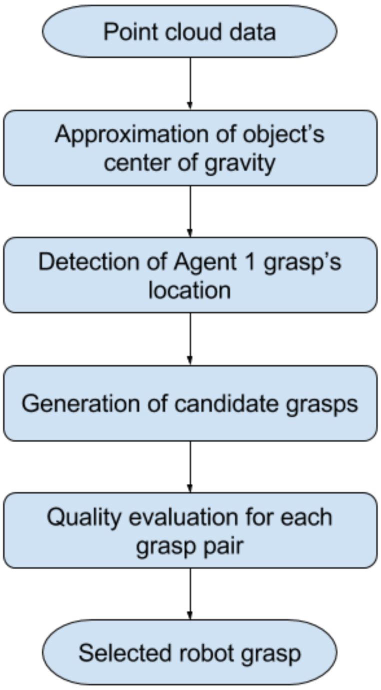

Point cloud data
Approximation of object's center of gravity
Detection of Agent 1 grasp's location
Generation of candidate grasps
Quality evaluation for each grasp pair
Selected robot grasp

Figure 4.1: Sequence of operations

for this purpose. For the collaborative grasp analysis method proposed in this chapter, it is assumed that one such approach will provide stable candidate grasps. A simple method based on elementary grasp actions (EGA) [35] is used in this work to extract candidate grasps on reachable part of the target object. The sequence of operations for robotic grasp planning is shown in Figure 4.1.

Given the approximate location of center of gravity, approximate location of grasp by the first agent and candidate robotic grasps, an algorithm for collaborative grasp quality evaluation is proposed in this chapter. The method tries to minimize total efforts required by both agents to complete manipulation task under given conditions. The problem is thus modeled as 2nd order minimization problem. Following sections explain the mathematical formulation of the proposed approach.

4.1 Grasp Representation

During robotic manipulation of an object, a single grasp results in multiple contact points on the target object. In typical grasp stability analysis, each contact point is represented by a single force (frictionless contact) or a friction cone (contacts with friction) in the direction of surface normal at the contact point. The grasp will be stable (force closure) if any external wrench can be resisted by these contact forces.

During the execution of a manipulation task, a stable grasp will be applying forces and torques in one or multiple direction on the target object. Thus at any time instant during the manipulation, a stable grasp can be represented by a wrench acting on the object.

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} \tag{4.1}$$

where $\boldsymbol{g}$ is the grasp wrench of an arbitrary stable grasp, $\boldsymbol{f}$ is the force vector, and $\boldsymbol{\tau}$ is the torque vector acting on the target object in the object reference frame.

4.2 Task Definition

Over the course of a manipulation task, the object undergoes a trajectory which can be described as a sequence of poses. At each pose during the manipulation, external wrenches such as gravitational force will be acting on the object. The manipulating agents need to cooperatively apply a wrench on the target object to compensate the external wrenches and change the object's state to next pose in the trajectory. In object reference frame, this wrench can be represented by a task wrench

$$\boldsymbol{w_t} = \begin{bmatrix} \boldsymbol{f}_t \\ \boldsymbol{\tau}_t \end{bmatrix} \tag{4.2}$$

where $\boldsymbol{f_t}$ is the task force vector and $\boldsymbol{\tau_t}$ is the task torque vector required at a particular time instant. A similar approach for task representation has also been used by Seredyński [12] for task specific grasp planning.

Consider a target object being manipulated by two agents A_1 and A_2, with its center of gravity at $\boldsymbol{o} = [0, 0, 0]^T$ in object reference frame. Assume that the locations of grasps by A_1 and A_2 are represented by vectors $\boldsymbol{p_1} = [p_{1x}, p_{1y}, p_{1z}]^T$ and $\boldsymbol{p_2} = [p_{2x}, p_{2y}, p_{2z}]^T$ respectively, where

$$\boldsymbol{p_1} = [p_{1x}, p_{1y}, p_{1z}]^T$$

$$\boldsymbol{p_2} = [p_{2x}, p_{2y}, p_{2z}]^T$$

Using the grasp representation discussed in Section 4.1, grasps by both agents can be expressed as:

$$\boldsymbol{g_1} = \begin{bmatrix} \boldsymbol{f_1} \\ \boldsymbol{\tau_1} \end{bmatrix}$$

$$\boldsymbol{g_2} = \begin{bmatrix} \boldsymbol{f_2} \\ \boldsymbol{\tau_2} \end{bmatrix}$$

where $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ are grasps by agent A_1 and A_2 respectively. For a successful manipulation, these grasps must equate the task wrench $\boldsymbol{w_t}$. Thus the system of equations for task execution can be expressed as

$$\boldsymbol{f_1} + \boldsymbol{f_2} = \boldsymbol{f_t} \tag{4.3}$$

$$\boldsymbol{p_1} \times \boldsymbol{f_1} + \boldsymbol{\tau_1} + \boldsymbol{p_2} \times \boldsymbol{f_2} + \boldsymbol{\tau_2} = \boldsymbol{\tau_t} \tag{4.4}$$

where $\boldsymbol{f_t}$ and $\boldsymbol{\tau_t}$ are force and torque vectors from task wrench.

For a lift-up task, if the object is kept in equilibrium after lift, the task wrench can be expressed as $\boldsymbol{w_t} = \begin{bmatrix} \boldsymbol{f_g} \\ \boldsymbol{0} \end{bmatrix}$ where f_g is the gravitational force on the object.

4.3 Grasp Quality Measure

The utility of a $\boldsymbol{g_1}$-$\boldsymbol{g_2}$ grasp pair is proportional to the wrenches required from both agents for the manipulation task where smaller wrenches represent a better grasp solution. Therefore, for a particular pose during the manipulation, a cost function for a grasp pair can be defined as

$$c(\boldsymbol{g_1}, \boldsymbol{g_2}) = \parallel \boldsymbol{f_1} \parallel^2 + {\omega_1}^2 \parallel \boldsymbol{\tau_1} \parallel^2 + \epsilon^2 \parallel \boldsymbol{f_2} \parallel^2 + \epsilon^2 {\omega_2}^2 \parallel \boldsymbol{\tau_2} \parallel^2 \tag{4.5}$$

The factors ϵ, ω_1, and ω_2 are non-negative coefficients, introduced to incorporate capabilities of manipulating agents in cost factor. ϵ indicates the ratio between contribution desired from each agent, whereas ω_i is the factor of torque compared to force for A_i in cost calculation. ω_i is typically related to the size of a gripper, which can be used as a scaling factor between forces and torques.

Setting $\epsilon = 1$ represents equal desired efforts by both agents. In a lift-up task, such a solution will try to distribute the object's load equally between both agents. A value of less than 1 for ϵ will result in higher efforts by A_2 compared to A_1.

For a $\boldsymbol{g_1}$-$\boldsymbol{g_2}$ grasp pair, the total cost of task execution can be calculated as sum of the costs over poses.

$$c_T(\boldsymbol{g_1}, \boldsymbol{g_2}) = \sum_i c_i(\boldsymbol{g_1}, \boldsymbol{g_2}) \tag{4.6}$$

where c_i is the cost of ith pose during the desired trajectory.

4.4 Cost Minimization and Grasp Solution

An optimum solution will offer a minimum total cost to ensure maximum quality. Since it was considered that the candidate grasps are available, the problem is reduced to grasp selection such that the cost of manipulation is minimized. Therefore, the load sharing problem can be considered as a quadratic minimization problem; minimizing cost expressed in (4.5) for each grasp pair while satisfying task equations presented in (4.3) and (4.4).

For the manipulation task, grasp wrenches are unknown for both grasps before task execution. However, minimum required wrenches for a known task wrench can be computed for each $\boldsymbol{g_1}$-$\boldsymbol{g_2}$ grasp pair where grasp by A_1 will remain same in all pairs. A candidate grasp by A_2 offering minimum total cost will result in an optimal collaboration under given cost factor. Quadratic programming [45] can be used to estimate minimum required wrenches.

The quadratic programming solution will estimate the unknown state vector $\boldsymbol{x}$ while minimizing the function

$$c(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q \boldsymbol{x} \tag{4.7}$$

$$s.t. \quad E\boldsymbol{x} + \boldsymbol{e_0} = \boldsymbol{0}$$

where

$$\boldsymbol{x} = [\boldsymbol{f_1} \ \boldsymbol{\tau_1} \ \boldsymbol{f_2} \ \boldsymbol{\tau_2}]^T$$

$$Q = \begin{bmatrix} I & O & O & O \\ O & {\omega_1}^2 I & O & O \\ O & O & \epsilon^2 I & O \\ O & O & O & \epsilon^2 {\omega_2}^2 I \end{bmatrix}$$

$$E = \begin{bmatrix} I & O & I & O \\ R_1 & I & R_2 & I \end{bmatrix}$$

$$\boldsymbol{e_0} = -\boldsymbol{w_t}.$$

The matrix Q is derived from the cost factor expressed in (4.5). I and O are 3×3 identity and zero matrices respectively. The matrix R_i is a cross product operator given by

$$R_i = \begin{bmatrix} 0 & -p_{iz} & p_{iy} \\ p_{iz} & 0 & -p_{ix} \\ -p_{iy} & p_{ix} & 0 \end{bmatrix}$$

such that the equality constraint $E\boldsymbol{x}+\boldsymbol{e_0} = 0$ is the equivalent representation of wrench equilibrium expressed in (4.3) and (4.4).

The state vector estimated by quadratic programming includes the minimum grasp wrenches required to satisfy task equations by a pair of grasps. Estimated grasp wrenches can then be used to calculate the cost of manipulation for a grasp pair using 4.5. For an established A_1 grasp at $\boldsymbol{p_1}$, grasp location of A_2 is selected from a given set of grasp candidates $P_2 = \{p_2^1, ..., p_2^n\}$ to accomplish a trajectory specified as a list of task wrenches $W_t = w_t^1, ..., w_t^n$ using

$$p_2 = \underset{p_2^k \in P_2}{\arg\min} \left(\sum_i \min_x c_i(x; w_t^i, p_1, p_2^k) \right) \tag{4.8}$$

The selected robot grasp at location p_2 is the one offering minimum expected wrenches required to complete the manipulation task.

4.5 Discussion

An novel approach was presented in this chapter to analyze candidate grasps for a collaborative manipulation task. The proposed solution estimates the grasp wrenches while trying to keep the cost of manipulation minimum. The cost of manipulation is formulated with grasp wrenches required to complete the manipulation task. The cost factor can also incorporate capabilities of manipulating agents and ability of an individual agent to generate force vs torque on the target object. The solution can also be extended to consider the force and torque limits that can be produced by manipulating agents.

Chapter 5

System Architecture and Design

This chapter describes the implementation details of the system developed for a human-robot collaborative lift-up. The complete system consists of a hardware setup including sensors and manipulator operated by a workstation. The workstation also hosts the analyses and decision making logic including an implementation of methodology proposed in Chapter 4.

5.1 Hardware

The hardware includes a depth sensor (Kinect), a manipulator (Kuka LWR) and a gripper (BarrettHand). A brief description of each component is given in following subsections.

5.1.1 Kinect

Kinect is an input device used by Microsoft for perception and motion sensing with gaming consoles. Different sensing components of Kinect are shown in Figure 5.1. The device is widely used in research work as a sensor for its RGB-D output. In this work, depth information from Kinect is used as sole input to the system. As mentioned in its specification [46], the device includes:

- An RGB camera with 1280×960 resolution

- An infrared emitter and sensor. Infrared light beams are emitted from IR emitter, the depth sensor senses reflected beams which are then

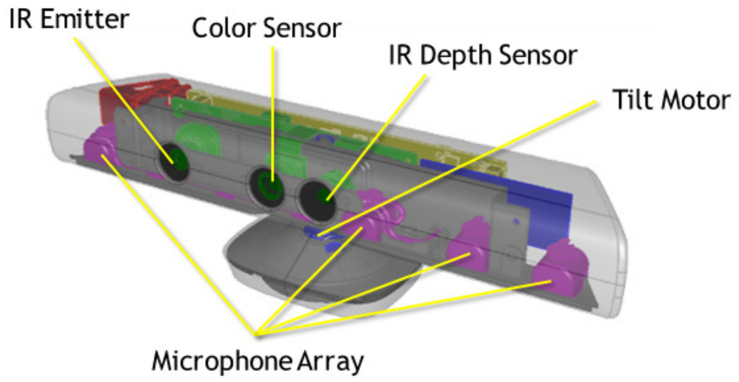


Figure 5.1: Kinect Sensor Components [46].

converted in depth information to measure distance between object and sensor.

– A multi array microphone

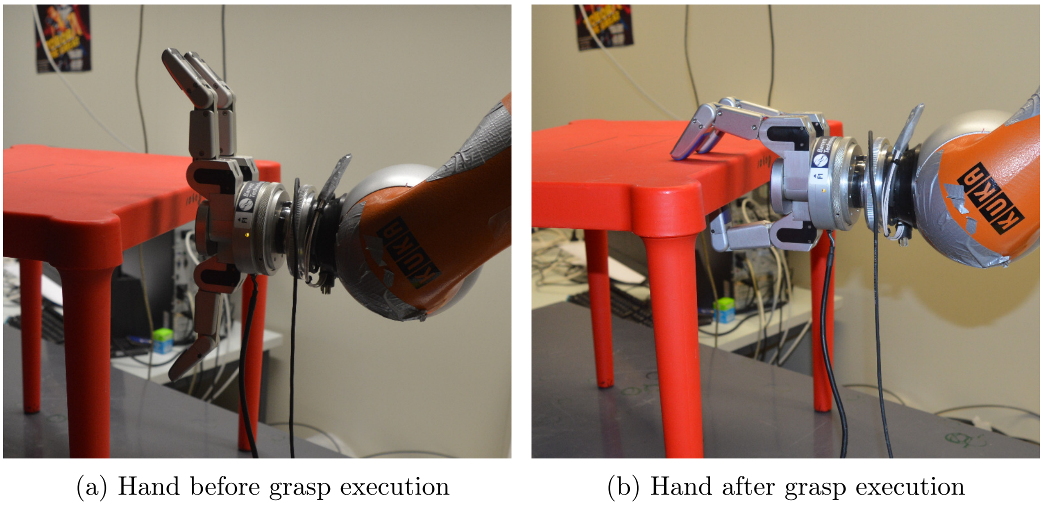
– A 3-axis accelerometer to determine current orientation of Kinect

5.1.2 KUKA LWR

KUKA Lightweight Robot (LWR) is a product of research collaboration between KUKA Roboter and the German Aerospace Center (DLR) [47]. The robotic arm has been specifically designed for robotic research and future manufacturing; and is commonly used by researchers for manipulation tasks. The problem considered in this work addresses grasp planning without the consideration of trajectory planning and control required for manipulation. Path planning and execution for the manipulator is carried out by existing planner and control solutions; and are not the focus of this work.

5.1.3 BarrettHand BH8-282

BH8-282 is a 3-fingered grasper that can be used to grasp object of different sizes, shapes and orientation [48]. Light weight hand of under 1kg can be used to lift payload of up to 6kg. The gripper is also equipped with tactile sensors



(a) Hand before grasp execution

(b) Hand after grasp execution

Figure 5.2: Grasp execution on a target object.

in finger tips and palm; and torque sensors installed in finger joints. In manipulation tasks, tactile sensing can be used to enhance perceived knowledge about the target object.

In this thesis, the hand is used as a simple gripper for grasping. Before making a grasp, fingers are kept open and spread (Figure 5.2a). To make a grasp, fingers are closed until the target object is grasped (Figure 5.2b). ROS package barrett_hand [49] is used to control the BarrettHand.

5.1.4 Calibration between Kinect and KUKA Arm

KUKA manipulator and Kinect being independent systems and installed at different spatial locations, have separate coordinate systems for representation of a point in space. To synchronize perception of environment and actuation within the environment, a coordinate transformation was required between Kinect and KUKA coordinate frames. Therefore, a calibration process was carried out to estimate the required transformation.

In the calibration process, a blinking IR emitter was attached to the palm of robotic hand. The robotic arm was moved to multiple random poses and IR image were captured using Kinect at each pose. Pose information and corresponding IR images were used to estimate coordinate transforms between the two coordinate systems. The calibration process used here was

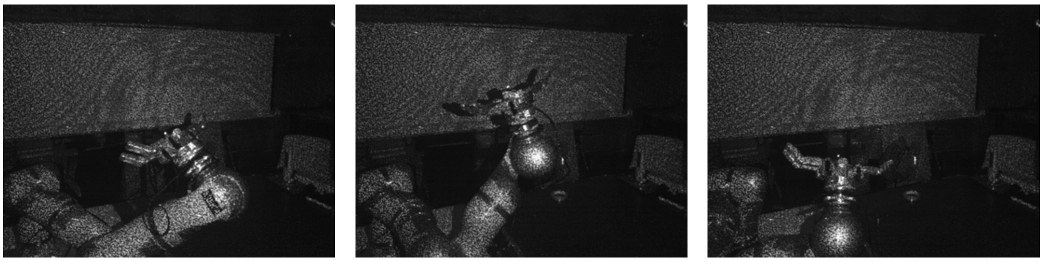


Figure 5.3: IR images captured during calibration process.

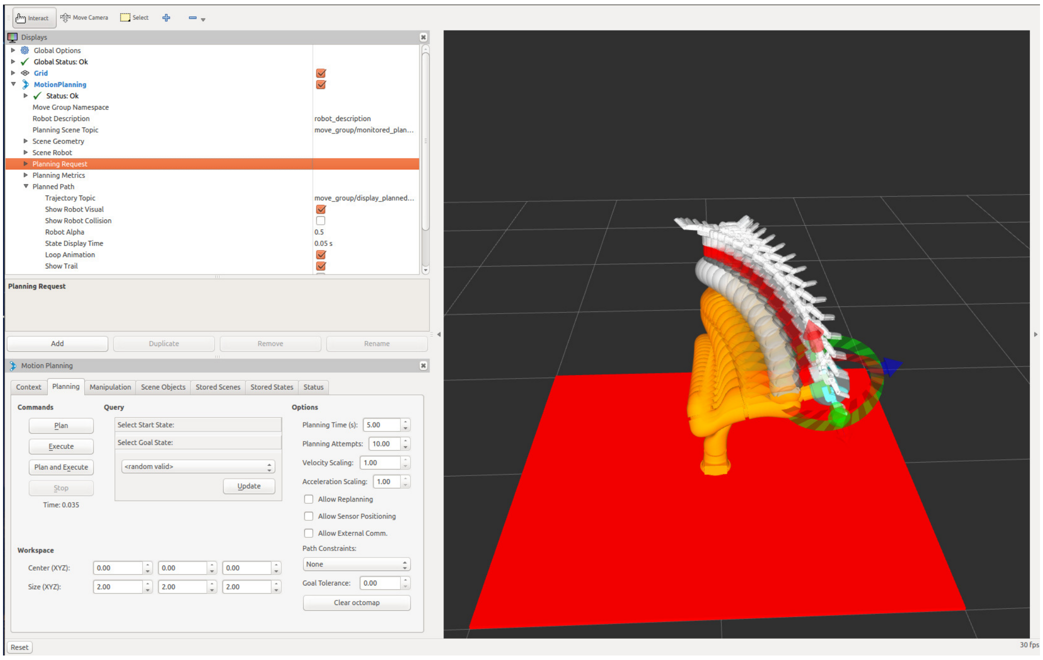
proposed by Ilonen [50]. Figure 5.3 shows sample IR images captured during the calibration process. A bright spot (IR emitter) can be observed in all images near the palm of BarrettHand.

5.2 Software Components

Several existing software components and APIs were used in this work. Use of already developed frameworks not only speeds up the development process but has also increases reliability of the system as they have been intensively tested by developers. All 3rd party components used in development for this work are open source and free to use; and are widely used in academia and industry in their respective fields.

5.2.1 Robot Operating System - ROS

The Robot Operating System is a framework developed to help in making robotic software applications. The open source framework is built to encourage collaborative robotic software development by mutual contributions from numerous research institutes and individual in robotic community. Flexible development framework contains a number of tools, libraries and packages to make a robotic software application simple and modular; easier to develop and test; and to reuse existing components. A visualization tool RViz is also included in ROS to visualize commonly used data types including images, objects, robot models and point clouds. Structural components in ROS including nodes, services and topics simplify communication between modules and integration of components. [51]



Displays

- Global Options
- ✓ Global Status: Ok
- Grid ☑
- MotionPlanning ☑
  - ✓ Status: Ok
  - Move Group Namespace
  - Robot Description: robot_description
  - Planning Scene Topic: move_group/monitored_plan...
  - Scene Geometry
  - Scene Robot
  - Planning Request
  - Planning Metrics
  - Planned Path
    - Trajectory Topic: move_group/display_planned...
    - Show Robot Visual ☑
    - Show Robot Collision ☐
    - Robot Alpha: 0.5
    - State Display Time: 0.05 s
    - Loop Animation ☑
    - Show Trail ☑

Planning Request

Motion Planning

Commands

Time: 0.035

Query

Select Start State:

Select Goal State: <random valid>

Options

Planning Time (s): 5.00

Planning Attempts: 10.00

Velocity Scaling: 1.00

Acceleration Scaling: 1.00

☐ Allow Replanning

☐ Allow Sensor Positioning

☐ Allow External Comm.

Path Constraints: None

Goal Tolerance: 0.00

Workspace

Center (XYZ): 0.00 0.00 0.00

Size (XYZ): 2.00 2.00 2.00

Figure 5.4: Path planned using MoveIt! planner, displayed in ROS-Rviz.

5.2.2 MoveIt! - Motion Planning Framework

MoveIt! is an open source motion planning, manipulation, control and navigation software [52] widely used in research and development. The framework includes implementation of state of the art algorithms for trajectory planning and control. In this thesis, MoveIt! integration with ROS is used for motion planning and control of KUKA arm. Figure 5.4 shows a MoveIt planned path as a sequence of intermediate poses for KUKA LWR.

5.2.3 Point Cloud Library

Point Cloud Library (PCL) is a large scale open source project for point cloud data processing [53]. The framework contains implementation of several algorithms for filtering, feature extraction, recognition, segmentation, model estimation and fitting; and visualization of data. Most point cloud data processing carried out in this work is performed using existing APIs and implementations by PCL. Information about different APIs and tutorials can be found at [54].

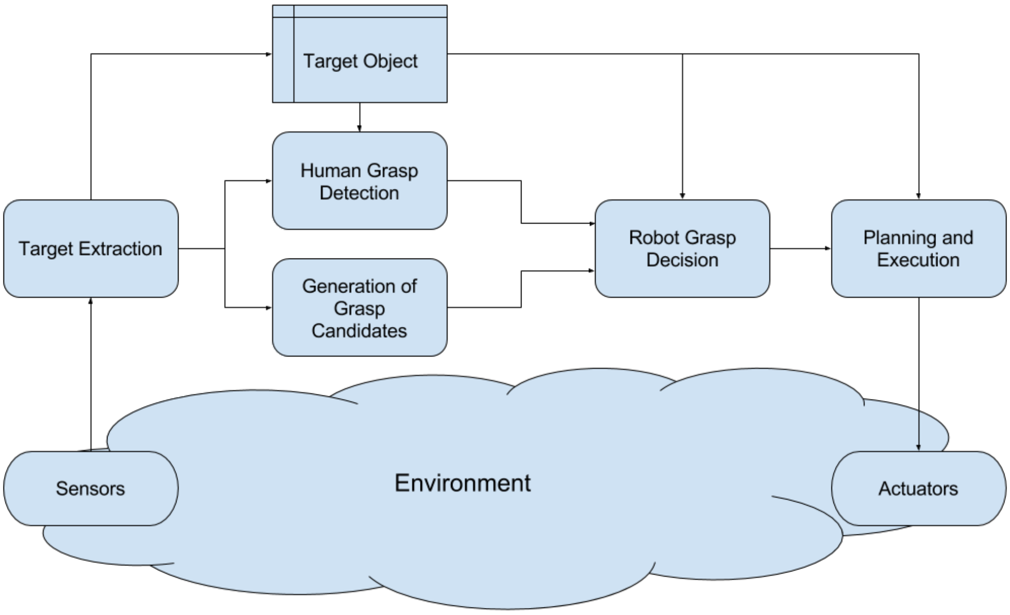


Figure 5.5: Block diagram of system.

5.3 Software System Architecture

A block diagram of software system is shown in Figure 5.5. A brief explanation about implementation of each component is provided in following subsections.

5.3.1 Target Extraction

Range data captured with a Kinect in the form of a point cloud is used as real time input to the system. However depending on the location of sensor, the data contains view of environment including clutter along with the object of interest. For a real time system, the first task is to extract the object of interest from environment for further processing. Besides, received data also includes random noise and variations which must be reduced before further processing.

A preprocessing step extracts the target of interest from cluttered environment. It is assumed that object of interest is the largest object in sensor's field of view. A number of filters are applied in preprocessing chain to extract the target object. The preprocessing sequence of operations is shown in

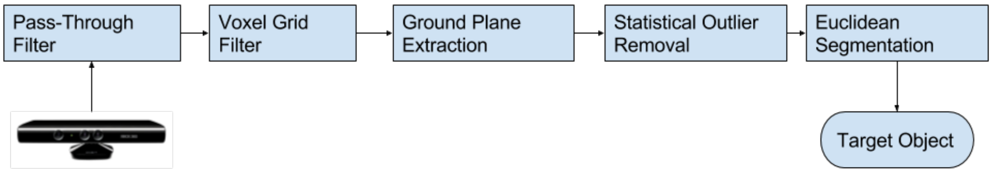


Figure 5.6: Operations on point cloud input for target extraction.

Figure 5.6. Filter parameters are made configurable to adjust system's performance according to requirements and quality of input. A brief descriptions of these filters is given below in the order of operation.

The **Pass-though filter** limits the range data to limited space where the object of interest is expected. This helps exclude clutter in background (far from area of interest) and significantly reduces data to be processed in later steps.

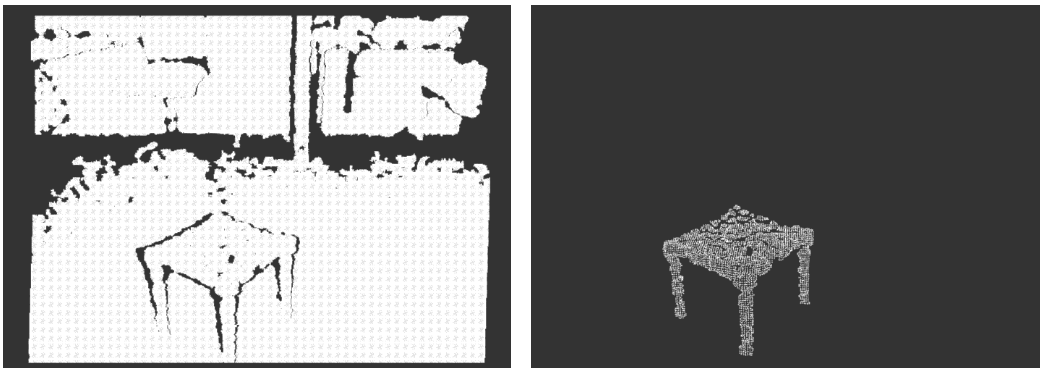
A **Voxel grid filter** is subsequently applied to down sample data if required. Performance is also of critical importance in a real time system,. Thus the system is designed to complete processing chain in reasonable time. Down sampling using Voxel grid filter reduces number of sample while keeping the overall structure of the target in point cloud data.

It is assumed that the target object is placed on a flat surface. Therefore, a ground plane should be visible in the input data. **Ground plane extraction** step extracts largest plane in point cloud using segmentation APIs provided in PCL. RANSAC algorithm is used in PCL for extracting largest plane in a given point cloud.

Input data captured by depth sensor contains noisy measurements in the form of outliers. A **Statistical Outliers Removal Filter** is used to remove those noisy measurements which utilizes statistical analysis techniques. An introduction to how the filter works and how it can be used in PCL can be found at [55].

After the processing steps discussed above, the data contains the object of interest along with other small objects present around the target in the work environment. To extract point cloud of the object of interest from its surroundings, a **Euclidean segmentation** technique is used. Euclidean segmentation clusters given point cloud data based on Euclidean distance between points in the point cloud. PCL implementation of Euclidean segmentation is based on Rusu's work [56].

Output of target extraction step is a point cloud containing range data of the target object. Figure 5.7 shows a point cloud before and after target extraction steps. The object point cloud is stored as reference for subsequent steps and also registered as obstacle in MoveIt! planner after triangulation. Center of gravity of the object is estimated by computing an axis aligned bounding box (AABB) of object point cloud. The center of AABB is considered as the CG.



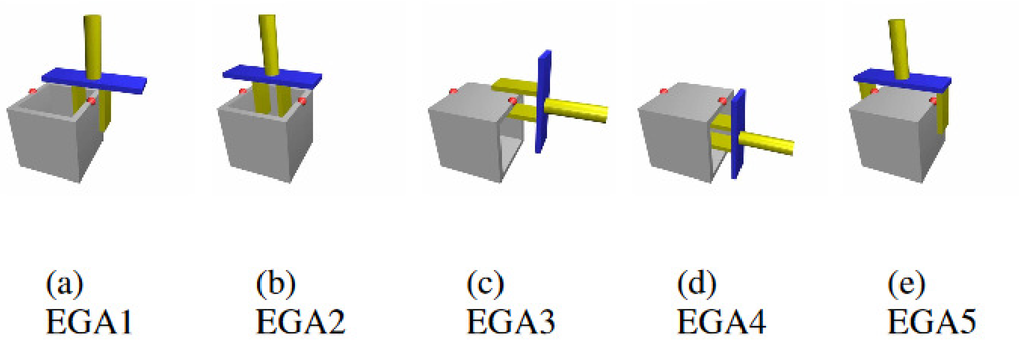
(a) Kinect captured range data (b) Extracted target after processing

Figure 5.7: Point Cloud before and after preprocessing cycle.

5.3.2 Generation of Grasp Candidates

Once point cloud of the target object has been extracted, a candidate grasp generation step extracts candidate graspable locations by analyzing visible shape of the target. Candidate grasp generation is simplified to focus more on collaborative behavior of the manipulation task. An elementary grasp analyses based method is used for generation of candidates. In this method, the system approximates target object with a set of planes and candidate grasp poses are generated on extracted planes, considering the size of robotic gripper.

Elementary grasp actions make use of planar surfaces and boundaries to generate candidate grasps on unknown objects. Figure 5.8 shows different elementary grasp actions. In this thesis, only EGA5 grasping approach [35]



(a) EGA1 (b) EGA2 (c) EGA3 (d) EGA4 (e) EGA5

Figure 5.8: Elementary Grasp Actions [35].

is used to extract candidate grasps. On a planar surface, EGA5 grasp will have the direction of gripper parallel to the surface normal.

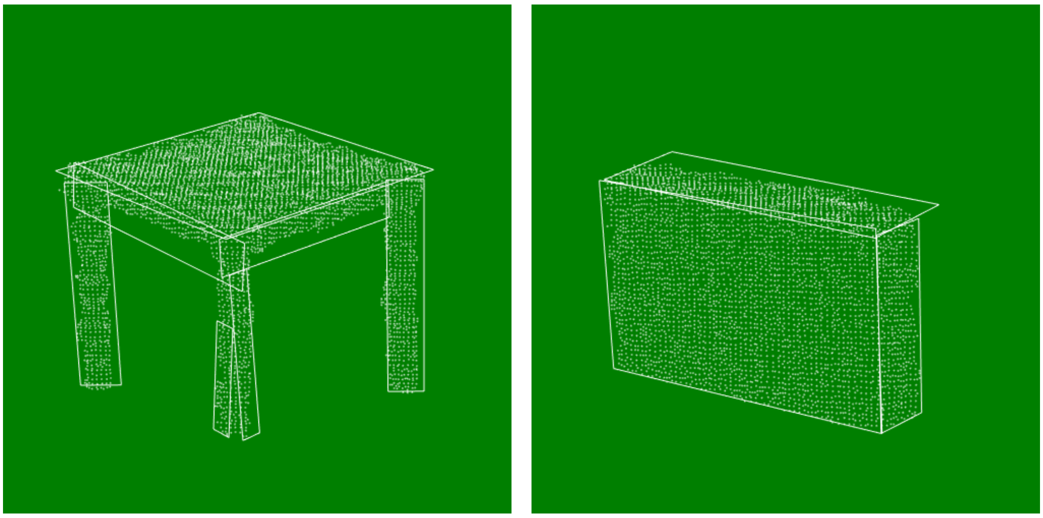
A candidate grasp consists of a pose of the gripper at grasp location. To execute the grasp, the gripper needs to be moved to the grasp pose with an open pre-shape and closed to make a grasp. An example was discussed in Section 5.1.3.

5.3.2.1 Rectangular Approximation

Plane segmentation API provided in PCL is used to extract planner surfaces in partial point cloud of target. PCL uses RANSAC algorithm to approximate plane parameters that contain maximum inliers. Once a plane is extracted, all points in close proximity of approximated plane are removed from the point cloud. Process is repeated until number of points remaining in point cloud are below a particular threshold.

Plane parameters approximated in aforementioned step do not represent a closed surface. A closed surface is generated by calculating convex hull of all inlier points for each plane. Convex hull is then replaced by minimum bounding rectangle to simplify closed shapes for further processing. However, the rectangles extracted this way do not ensure enclosed area to be filled with points i.e. side of a table which is mostly empty will be approximated as a single rectangle. A recursive algorithm is used to subdivide these surfaces into smaller regions until density of resultant rectangles is above configured threshold.

Figure 5.9 shows rectangular approximation of table and box objects along with the point cloud data. It can be seen that shape as a set of rectangles is reasonably close approximation of original point cloud data.



(a) Rectangular approximation of a table. (b) Rectangular approximation of a box.

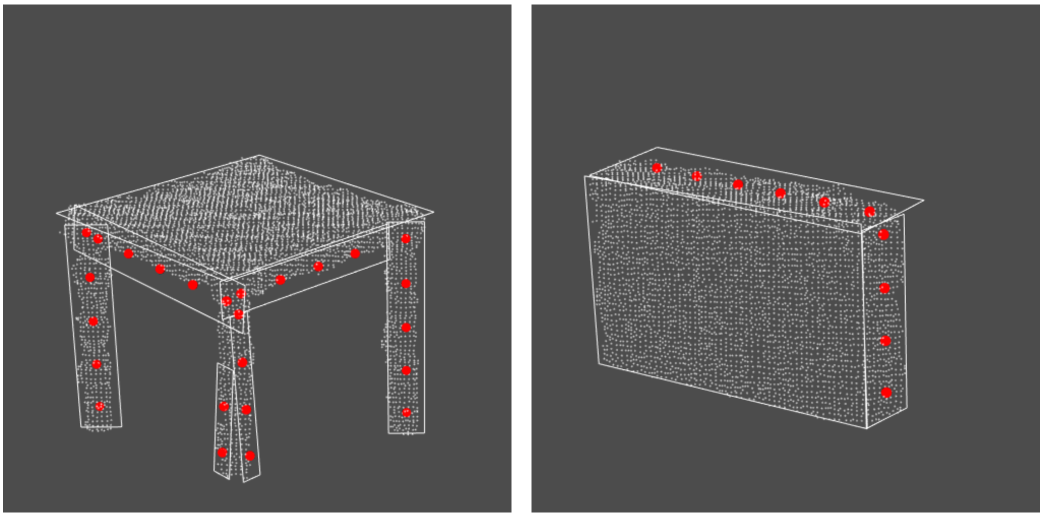
Figure 5.9: Rectangular approximation of different point clouds.

5.3.2.2 Generation of Grasps

Once a rectangular approximation of target object has been calculated, candidate EGA5 grasps (Figure 5.8) can be calculated on approximated rectangles. Only rectangles with width less than maximum gripper width are used to generate candidate grasps. To generate multiple candidate grasps on a rectangle, a configurable step size is used. Direction of the gripper at grasp location is parallel to the surface normal of rectangle in a way that the gripper approaches grasp location on the outer surface of the object. Candidate grasp locations extracted for a table and a box target are shown in Figure 5.10.

5.3.2.3 Collision Avoidance

Extracted candidate grasps are generated on each rectangle individually. The generation method does not ensure feasibility of grasp execution and collision avoidance. To filter out candidates which result in a definite collision, a collision check is performed on candidate grasps.



(a) Candidate grasp locations on a table.

(b) Candidate grasp locations on a box.

Figure 5.10: Candidate grasp locations generated for different objects.

A grasp will not be feasible if the gripper collides with the target object before making a grasp. A stable grasp thus can be validated by placing the gripper at candidate grasp point in simulation environment and checking for potential collision. Gripper is modeled as set of cuboids for the purpose, which is placed at candidate grasp location with point cloud data (Figure 5.11). If one or more points from point cloud are found inside gripper model, grasp will result in a collision. Such grasps are therefore removed from candidate grasp points.

5.3.3 Human Grasp Detection

Prior to making grasp decision, the designed system waits for a human to attempt a grasp on the target object. Human grasp is detected by comparing object's reference point cloud (extracted in Section 5.3.1) and real time point cloud data from Kinect. Difference in both point clouds identifies a human grasp. Figure 5.12 shows an attempted human grasp identified in a point cloud. Once a difference in point clouds is detected, location of human grasp is approximated by taking mean position of 100 points closest to the target object.

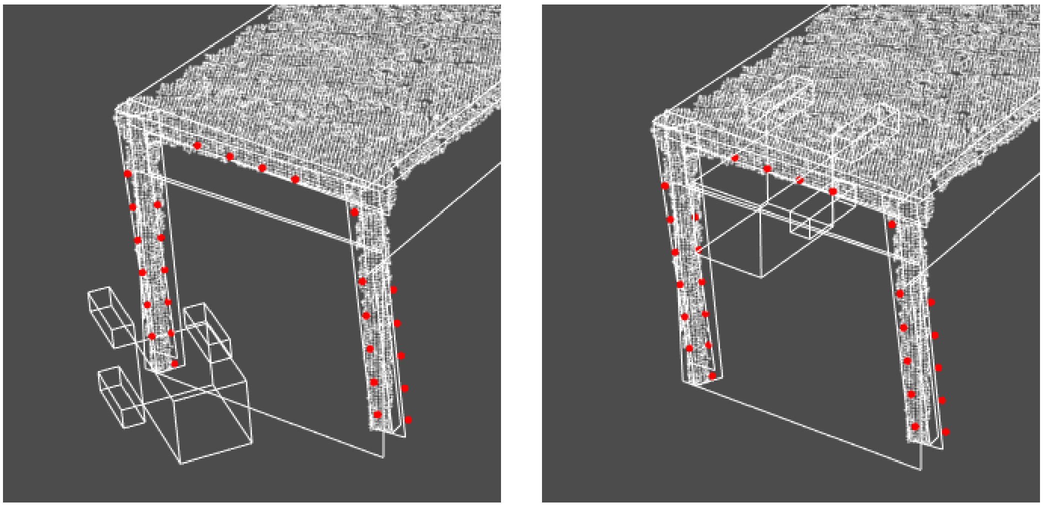


Figure 5.11: Collision check of different candidate grasps for a table.

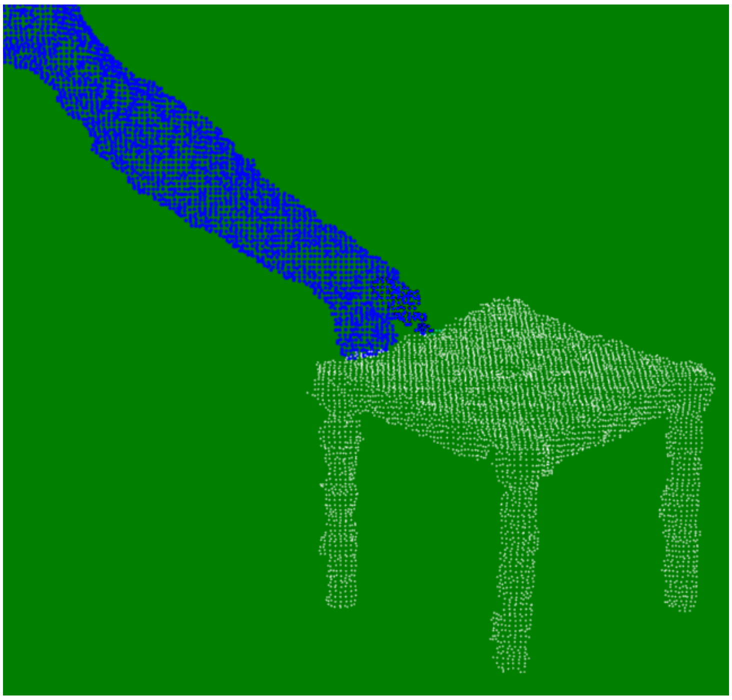


Figure 5.12: Point cloud of human grasp on a table.

5.3.4 Robot Grasp Decision and Execution

A new approach was proposed in Chapter 4 to rank candidate grasp locations in the order of maximum quality for collaboration. For experiments in this work, a value of one is considered for ϵ and ω. Thus the system will try to achieve an equal contribution by both agents during the manipulation task. Such a setup simplifies evaluations of results. System's module responsible for grasp decision making takes as input the center of gravity location of the object, candidate robot grasps and human grasp location. Since a lift-up task is considered for a human-robot collaborative manipulation, the task definition includes only the final pose of the object with task wrench

$$\boldsymbol{w_t} = \begin{bmatrix} \boldsymbol{g_g} \\ \boldsymbol{0} \end{bmatrix} \tag{5.1}$$

where f_g is the force due to gravity. The task equations for the object after lift-up can be expressed as

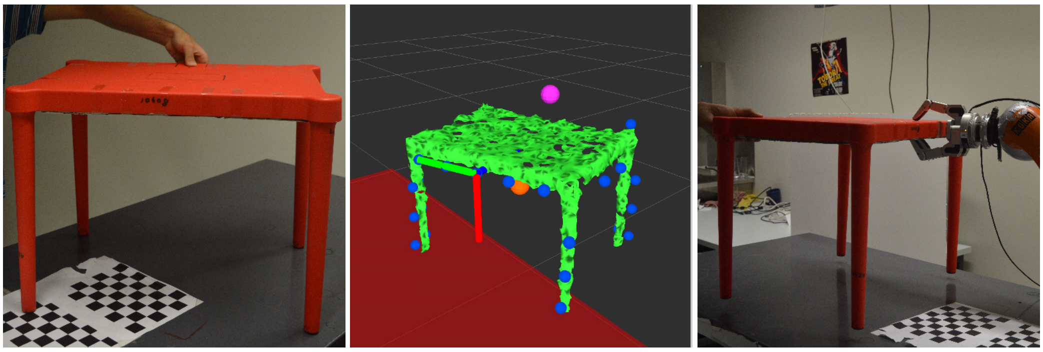
$$\boldsymbol{f_1} + \boldsymbol{f_2} = \boldsymbol{f_g} \tag{5.2}$$

$$\boldsymbol{p_1} \times \boldsymbol{f_1} + \boldsymbol{\tau_1} + \boldsymbol{p_2} \times \boldsymbol{f_2} + \boldsymbol{\tau_2} = 0 \tag{5.3}$$

For a fixed human grasp, the cost of task execution was computed for each candidate grasp - human grasp pair using the method proposed in Chapter 4 and the candidates were ranked in ascending order of cost. A candidate grasp producing the minimum cost was selected for robotic grasp. Subsequently, communicated to MoveIt planner and controller for execution to complete the manipulation task.

It is to be noted that the gravitational force in the task wrench depends on mass of the object which cannot be estimated for an unknown object. However, a unity value can be assumed for the unknown mass since the optimal location for a robotic grasp will be independent of object's mass. A constant value of unknown mass used for cost evaluation of each candidate will scale the approximated cost with the same factor.

Figure 5.13 shows an example execution. Estimated location of center of gravity of the objected is displayed as an orange sphere in the Figure 5.13b. Candidate grasp locations and estimated location of human grasp are identified with pink and blue spheres respectively. The decision for robotic grasp is shown as a coordinate axis indicating the pose of gripper at the time of grasp.



(a) Real environment. (b) Visualized environment and grasp decision. (c) Collaborative manipulation.

Figure 5.13: Human-robot collaborative lift-up of a table.

5.4 Discussion

A real time collaborative manipulation system was presented in this chapter and its implementation was discussed. Free and open source frameworks and libraries were used where ever possible during the development in this work. The system also includes visualizations and hardware integrations. All software components were implemented keeping in mind modularity and reusability of the developed modules. Furthermore, to ensure acceptable real-time performance and reliability, computational cost was also considered and minimized during the development. For wide applicability of the system, minimum assumptions were made about the environment and generic approaches were used.

Chapter 6

Experiments and Results

This chapter presents experiments conducted to analyze performance of the developed system and grasping solution for collaborative manipulation. Target objects used in these experiments were a table, a rectangular box, a cylindrical pipe and a chair.

6.1 Experiment 1: Verification of Candidate Grasps

One of the fundamental assumptions towards the methodology proposed in Chapter 4 was the stability of individual grasps, which is also crucial for expected collaborative behavior in manipulation. As explained in Section 5.3.2, a simple EGA5 method is used in this work for generation of candidate grasps. It was assumed that the extracted candidates result in a successful grasp.

Before proceeding with collaborative manipulation experiments, grasp candidates were verified on different target objects. Furthermore, it was validated in this experiment that the method used for generation of candidate grasps is applicable on objects of different shapes.

6.1.1 Experimental Setup

An unknown object was placed in work area of the robotic manipulator. Partial object model acquired by sensory information was registered as obstacle with MoveIt! planner. Candidate grasps were generated on visible surface of the object. Each candidate grasp was tested individually by approaching the grasp location with an open gripper and making a grasp by closing the gripper. If the object was held inside robotic hand, the grasp was considered successful.

This experiment was performed on a table and a box taking two random poses for both the objects. For a cylindrical pipe and chair object, only one random pose was considered. At least ten candidate grasps were tested for each target object.

6.1.2 Results

Figure 6.1 shows objects in the real environment vs visualized by the system. Generated candidate grasps are displayed as blue markers in the visualized environment. A robotic grasp was attempted for each reachable candidate grasp location. The success rate was found to be around 90% on considered objects. Some of the attempted grasps resulting in a success are shown in Figure 6.2.

The grasp success rate on archived in this experiment is on a par with [35], which showed that for low complexity objects, grasp success rate can go above 80% using EGAs.

The results also validate that the EGA as a candidate grasp generation method can be considered suitable for subsequent experiments on collaborative manipulation.

6.2 Experiment 2: Real-time Grasp Decision

System design discussed in Chapter 5 is a real time system able to detect the human grasp on a target object and decide a collaborative robotic grasp for manipulation accordingly. To verify the real time behavior of the system, this experiment was aimed to test system's ability to detect changes in human grasp location and adjust its decision accordingly.

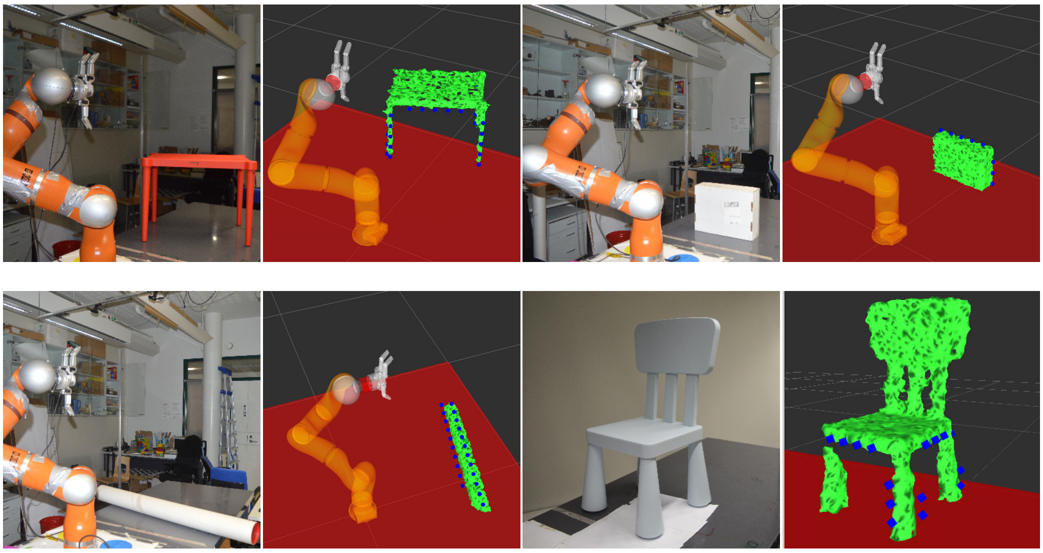


Figure 6.1: Real vs visualization environment. Blue markers in visualization environment represent extracted candidate grasp locations.

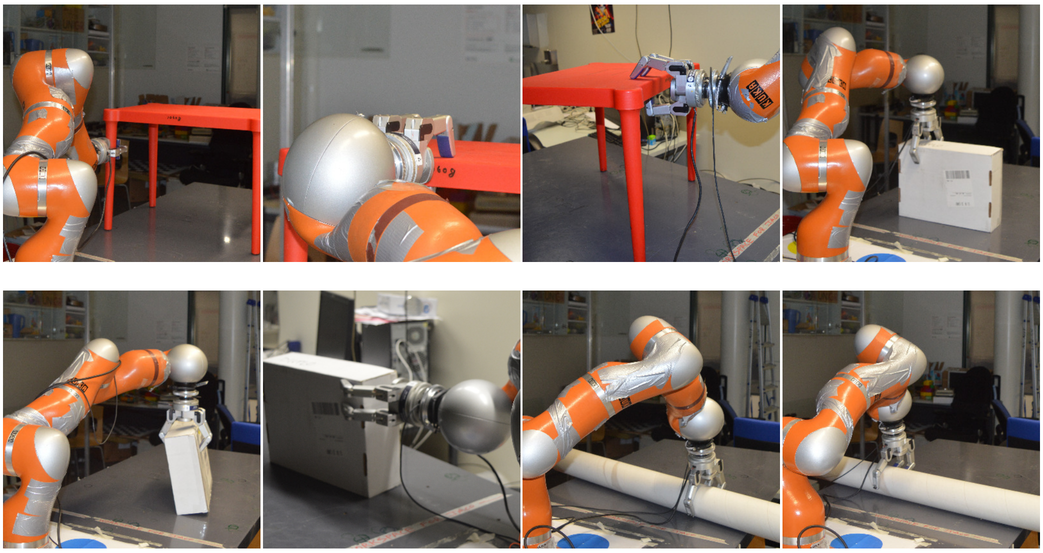


Figure 6.2: Grasps attempted on different objects.

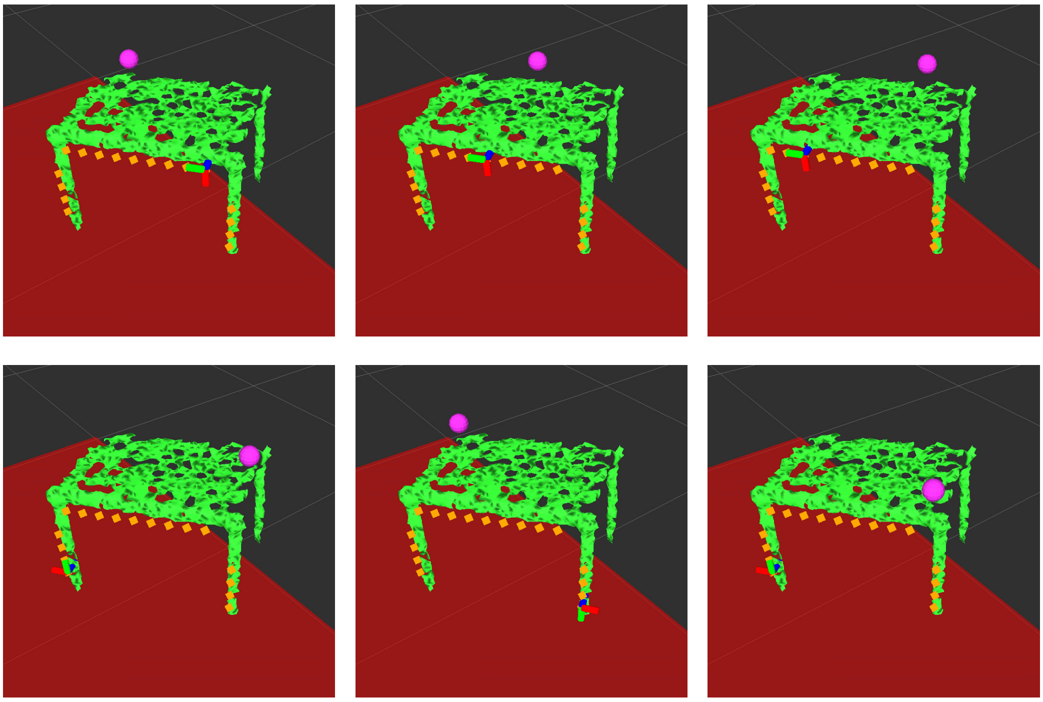


Figure 6.3: Target object (green), candidate grasp locations (orange squares), human grasp location (pink sphere) and decision for robotic grasp (RGB coordinate axis marker) for multiple human grasp attempts.

6.2.1 Experimental Setup

A table object was placed in workspace of KUKA robotic manipulator. Human grasps were attempted at different locations on the target object and system's decisions for robotic grasp were visualized without executing robotic grasps.

6.2.2 Results

The system was able to detect human grasp on the target object and make its decision accordingly in a period of less than 2 seconds. Figure 6.3 shows multiple human grasps attempts on a target object and corresponding decision made by the developed system for robotic grasp.

Since the proposed grasping approach selects one of the available candidate grasps, the ability of the system to adjust its decision is limited to discrete set of available candidate locations. Moreover, the designed system approximates human grasp location from difference between reference point cloud data of the target and real time point cloud data. If a robotic grasp has already been made, manipulator will also appear in difference of point clouds making it impossible for the system to detect or update human grasp location. However, the limitation can be eliminated by incorporating a self see filter to exclude points belonging to the manipulator from sensory point cloud data.

Nevertheless, the results demonstrate ability of the developed system to be used in an real-time setup for online decision making.

6.3 Experiment 3: Collaborative Manipulation

In a manipulation task, multiple candidate grasps can result in successful manipulation. However, the quality of task completion in terms of stability and load sharing differs for each solution. The manipulation task considered for this experiment was a lift-up of 5 cm above the ground surface. The quality of the result can be evaluated by either measuring the tilt of the target object after lift or by measuring the total effort applied by both agents (human and robot) during the manipulation task. Applied effort can be approximated by observing forces and torques exerted by manipulating agents. In case of tilt based evaluation, a successful lift with less or no tilt was considered as a desirable result.

6.3.1 Experimental Setup

To measure the quality of different grasps, a target object was placed in work space of the robotic arm. A human grasp was executed on the target object. Human grasp in this experiment was emulated by a constant support made of legos. This eliminated any possible variations in actual human grasp execution during multiple experiment iterations. For a fixed human grasp, manipulation task was performed with multiple candidate grasps to measure the quality of collaborative manipulation for each pair. The quality measures were subsequently correlated with grasp decisions by the proposed method to observe optimality of the method. Two primary aspects: collaborative grasp stability and load sharing, were investigated for collaborative manipulation. Experiments and results for both measures are discussed in following subsections.

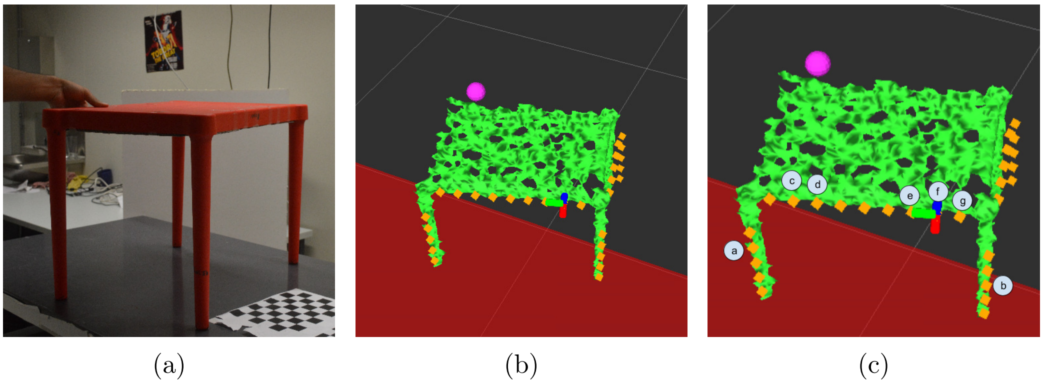
6.3.2 Collaborative Grasp Stability

To analyze collaborative grasp stability, quality of manipulation was evaluated by measuring tilt produced in the table after lift. Tilt was measured using accelerometer measurements taken before and after the lift at table top surface. Roll and yaw angles of tilt were computed using tri-axis tilt sensing [57]. Sum of square of both roll and yaw angles gave total tilt produced in the table. The experiment was performed on two random poses of the table with different human grasp locations.

Figure 6.4 shows first experimental case along with a human grasp. Approximated human grasp location, candidate robot grasp locations and grasp decision by the developed system are displayed in the visualization environment. To verify the quality of decision, the manipulation task was performed with seven candidate robot grasp. The evaluated grasps are labeled in Figure 6.4c. Manipulation task was performed with each of these candidates as shown in Figure 6.5. It can be observed that manipulation task (i.e. lifting of object) is successful in most cases but the object has a different tilt angle after each manipulation. A sum of absolute roll and yaw tilt angles is plotted in Figure 6.6. It can be seen that the tilt after manipulation is minimum for Grasp f, which was also selected for robot grasp by the proposed solution.

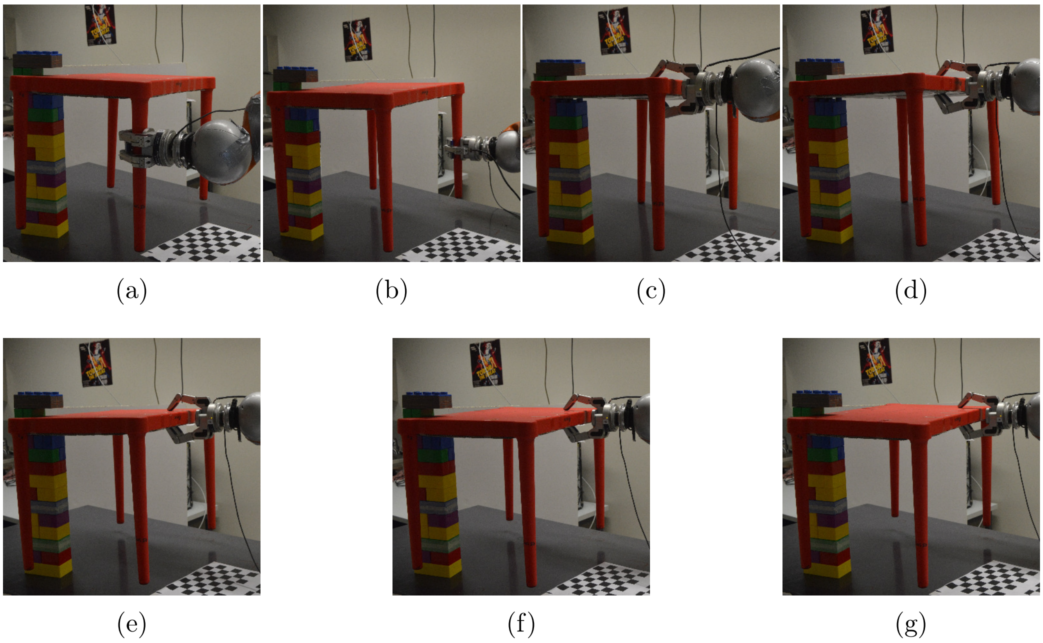
The same experiment was repeated on the table object with a different pose and human grasp location. Real and visualization environments are shown in Figure 6.7 along with candidate grasp locations and selected robot grasp solution. Similar to the prior experiment, a lift of 5 cm was performed with pairs of fixed human grasp and eight candidate robot grasps (Figure 6.7c and 6.8). Tilt in the table was measured after the manipulation. Total tilt angle for each evaluated grasp pair is plotted in Figure 6.9. Grasp f was observed to have minimum tilt after manipulation which is again same as the computed solution by the developed system.

The proposed method was able to choose the optimal solution in both executions. This substantiates that the method plans stable coordinated grasps for stability defined using tilt and thus can be utilized for collaborative



(a) (b) (c)

Figure 6.4: Table object (pose 1) (a) Real environment with a human grasp (b) Visualization environment (c) Visualization environment - Pink sphere is estimated location of human grasp, yellow squares are robot grasp candidates and rgb axes marker indicates grasp decision by proposed solution. Candidate grasps evaluated in this experiment are labeled (a-g).



(a) (b) (c) (d)

(e) (f) (g)

Figure 6.5: Table object (pose 1) - 7 candidate robot grasps (a-g) executed against an emulated human grasp.

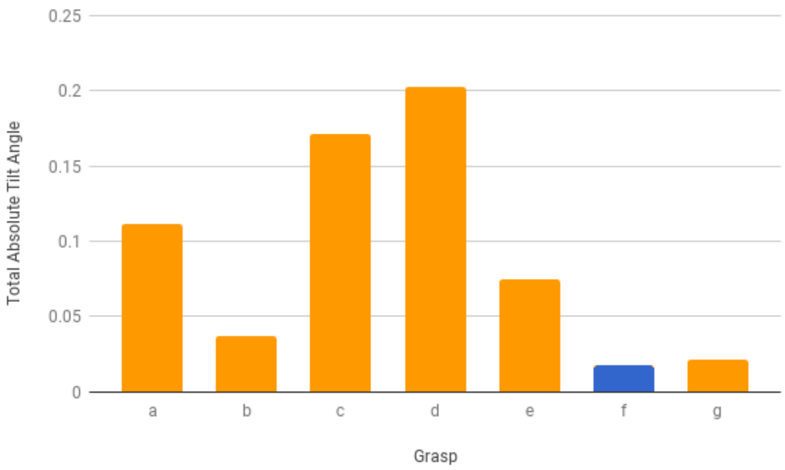
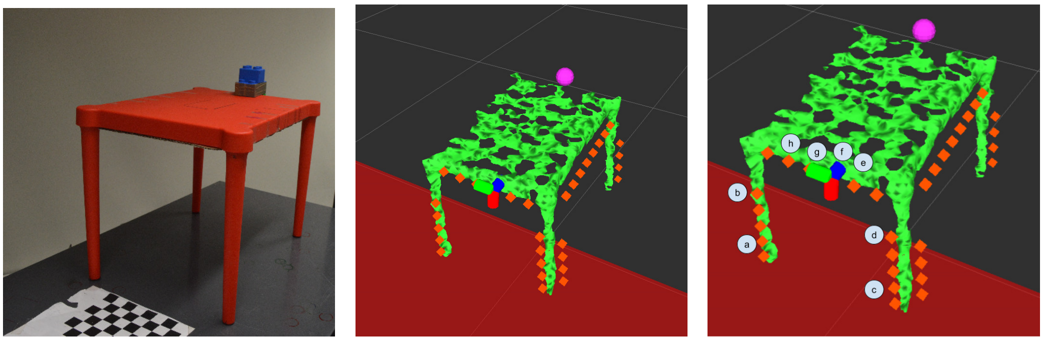


Figure 6.6: Table object (pose 1) - Total absolute tilt angle (radians) for robot grasps (a-g).



(a) (b) (c)

Figure 6.7: Table object (pose 2) (a) Real environment with an emulated human grasp (b) Visualization environment (c) Visualization environment - Pink sphere is estimated location of human grasp, orange squares are robot grasp candidates and rgb axes marker indicates grasp decision by proposed solution. Candidate grasps evaluated in this experiment are labeled (a-h).

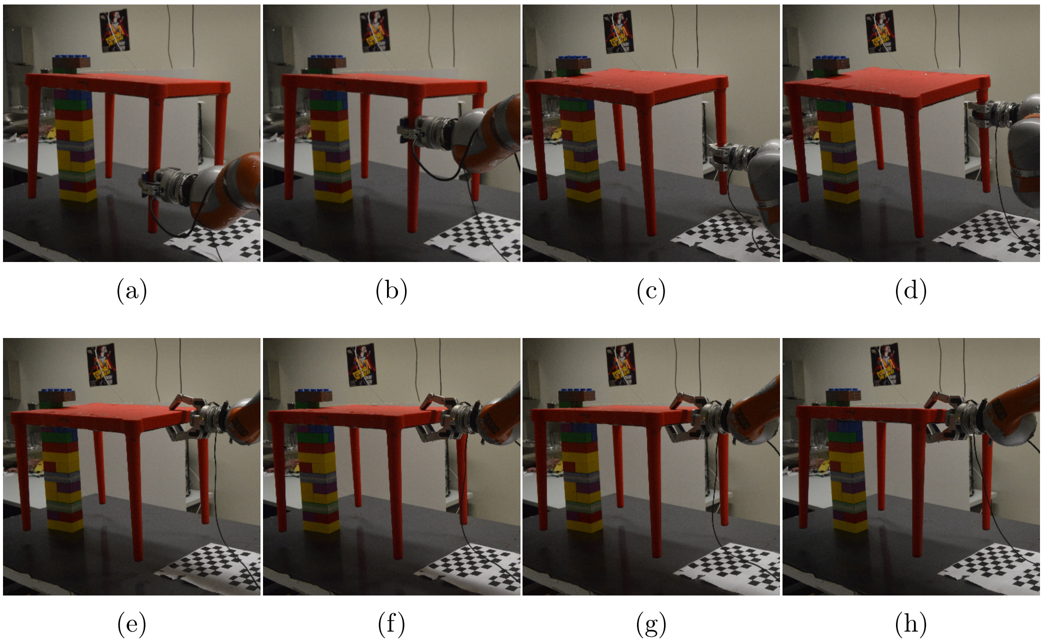


Figure 6.8: Cooperative manipulation of the table (pose 2) - 8 candidate robot grasps (a-h) executed against an emulated human grasp.

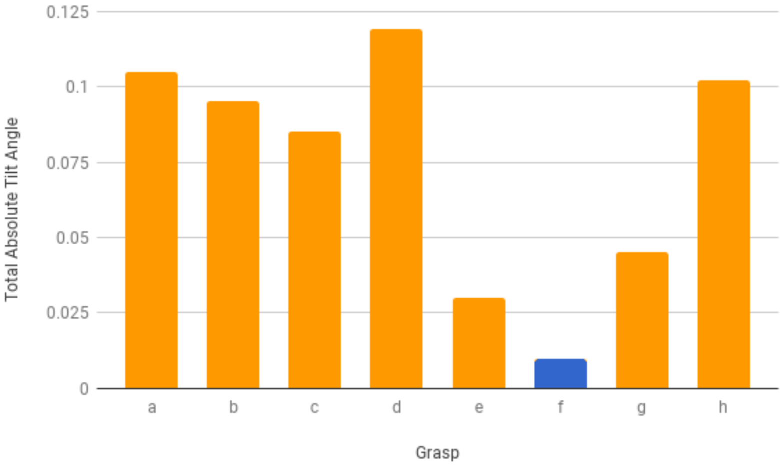


Figure 6.9: Table object (pose 2) - Total absolute tilt angle (radians) for robot grasps (a-h).

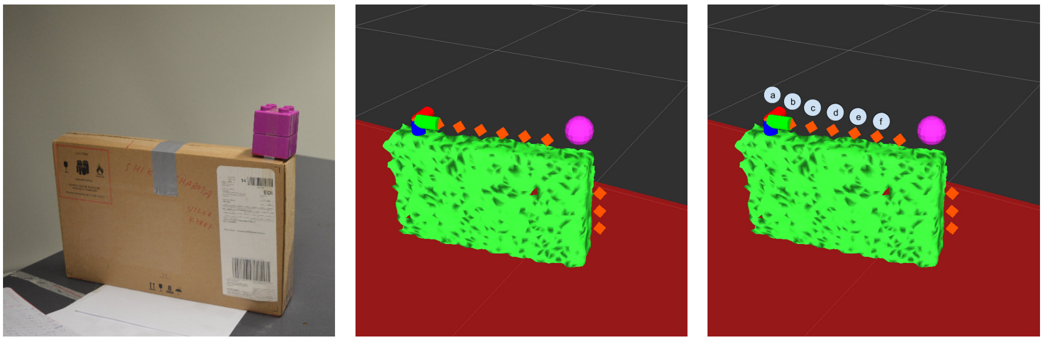
grasp planning.

6.3.3 Load Distribution

If human grasp location, center of gravity of the object and all candidate robot grasp locations on a target object are collinear, selection of different candidate grasps will not produce significant tilt in the object after lift-up. An example object is a rectangular box as shown in Figure 6.10. However, object's load is distributed differently for different candidate grasps. To validate load sharing between agents, force contributed by manipulator in a lifting task is measured using a force/torque sensor. Since an equal contribution was desired from both manipulating agents, an optimal solution will be when the robot carries half of the total load of object. In this experiment, total load was measured by lifting object with only the manipulator and observing the force exerted on the force/torque sensor and was observed to be approximately 4N.

A rectangular box is shown in real and visualization environment in Figure 6.10. For a fixed human grasp, manipulation task is performed with six candidate robotic grasps. Evaluated grasp are labeled (a-f) in Figure 6.10c and corresponding task execution is displayed in Figure 6.11. It can be observed that orientation of the target object is identical after manipulation in all cases. Therefore, instead of the tilt, force experienced by the gripper in the direction of gravity (negative Z axis in chosen world coordinate frame) was observed to approximate robot's contribution in the manipulation task. The force was measured for different robot grasps and has been plotted in Figure 6.12. As the robotic grasp location moves closer to the object's center of gravity, contribution by the robot will increase. Since the total load of the object was observed to be 4N, an optimal solution must contribute close to half of the total required force i.e. 2N, which makes Grasp b to be the optimal solution in this case.

As displayed in Figure 6.10b, the proposed collaborative grasping solution selected Grasp a for robotic grasp which is closest to the actual optimal solution. Since the proposed solution depends on location of human grasp and the center of gravity of object, both of which are approximated in this experiment, difference in approximated and actual locations of these parameters will effect the optimality of selected solution. Moreover, due to the physical



(a) (b) (c)

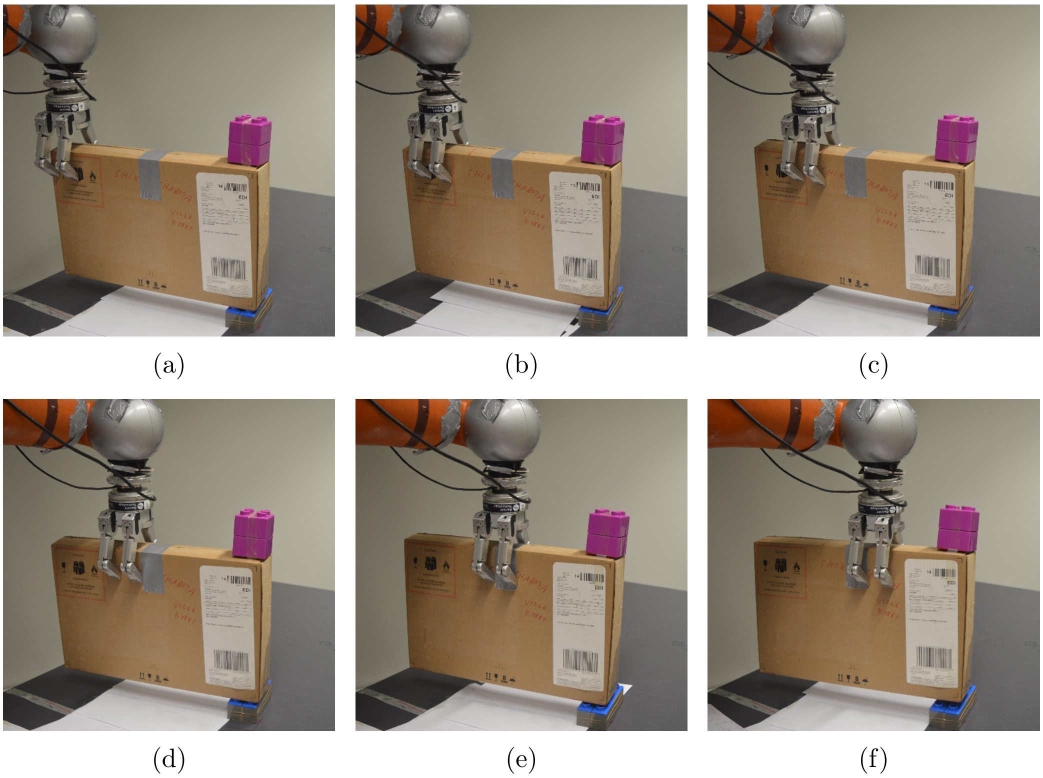
Figure 6.10: Box object (a) Real environment with an emulated human grasp (b) Visualization environment (c) Visualization environment - Pink sphere is estimated location of human grasp, orange squares are robot grasp candidates and rgb axes marker indicates grasp decision by proposed solution. Candidate grasps evaluated in this experiment are labeled (a-f).

properties of target object in this particular case, emulated human grasp with legos also introduces additional uncertainty in perceived and actual location of the grasp effecting the measured forces as a result.

The force measurements indicate that the proposed method can be used for load distribution among agents. However, the quality of load distribution depends on the accuracy of estimated parameters such as the object center of gravity.

6.3.4 Collaborative Manipulation of Complex Shaped Objects

Individual grasp success is fundamental to collaborative manipulation. A simple EGA method (as discussed in Section 5.3.2) was used in this work for generating candidate grasp locations using plane estimation of object's surface. Therefore, if the surface is not planar, the grasp may not be stable and manipulation can result in a failure. This was observed while conducting experiments with cylindrical shaped objects as shown in Figure 6.14 and Figure 6.13d. The grasps resulted in a success according to the criterion used in experiments discussed in the Section 6.1 but the lifting task failed.



(a) (b) (c)

(d) (e) (f)

Figure 6.11: Cooperative manipulation of box object - 6 robot grasps (a-f) executed against an emulated human grasp.

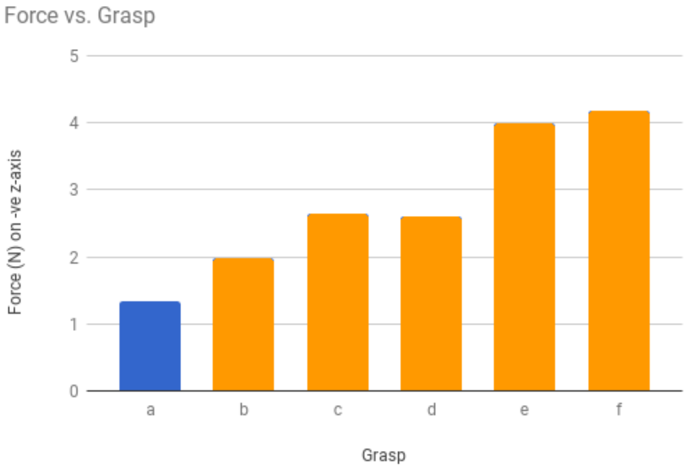


Figure 6.12: Box object - Forces experienced by gripper for Grasps (a-f). Grasp (a) is the grasp decision by proposed solution.

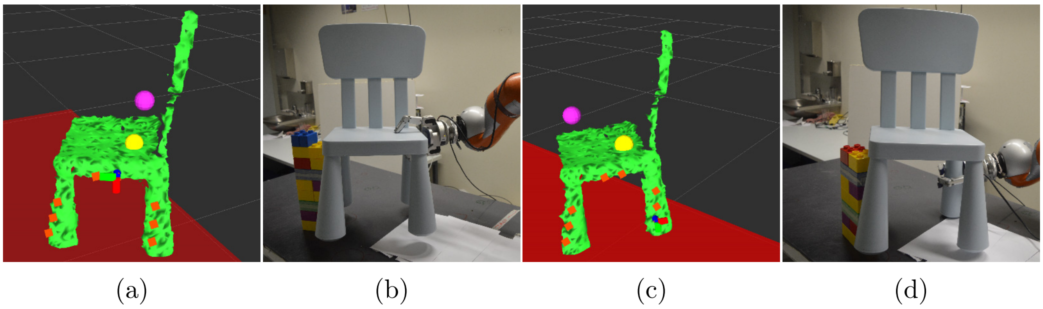


Figure 6.13: Cooperative manipulation with a chair object. (a) Pose 1 - Visualization environment (b) Pose 1- Cooperative lift-up by selected solution. (c) Pose 2 - Visualization environment (d) Pose 2- Cooperative lift-up by selected solution. - Pink sphere is estimated location of human grasp, yellow sphere is the estimated location of center of gravity, orange squares are candidate robot grasp locations and rgb axes marker indicated grasp decision.

The cylindrical shape and weight of the object caused it to slip from the hand. However, if the individual grasp is stable, the grasp solution by the proposed method results in a successful manipulation as shown in Figure 6.13b.

Thus a successful collaborative manipulation can be achieved using the proposed method if a close approximation of object parameters and set of stable candidates is available.

6.4 Discussion

This chapter presented experiments performed to validate the proposed grasping solution. It was shown that the proposed methodology produces satisfactory results for objects of different size, shape and pose despite the partial knowledge about the object. It was also observed that stability of an individual grasp is critical for success of the manipulation task. The approach used for generating candidate grasps produced good results on simple shaped objects but was not suitable for complex shapes. However, the proposed solution for collaborative manipulation does not bounded with the method used for generation of candidate grasps. Thus, a more advanced approach

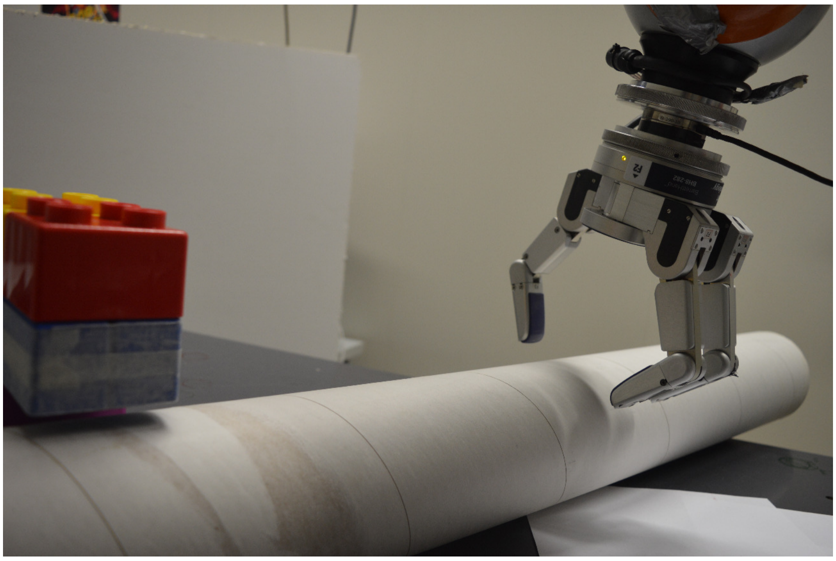


Figure 6.14: Failed lift on a cylindrical shaped object.

can be used to generate better individual candidates. It was also shown by experiments that the grasp decision exploiting sensory information was very close to the measured optimal solution. This substantiates potential of the proposed method as an on-line collaborative grasping solution.

Chapter 7

Conclusion and Future Work

The objective of this thesis was to develop a solution for collaborative manipulation of unknown large objects. A human-robot collaborative lift-up was considered as the manipulation task. A novel approach was proposed to analyze candidate robotic grasps with respect to the collaboration criterion. The proposed approach modularized collaborative grasping problem into generation of grasp candidates and grasp quality analysis for collaborative manipulation, allowing the use of existing grasp syntheses techniques for generation of candidate grasps. The complete solution for human-robot collaborative lift-up included analysis of target object using range data from a single Kinect view, estimation of object's center of gravity, candidate robotic grasps generation, detection of human grasp on the target object, robot grasp decision and execution.

For robotic grasping, a grasp coordination method was proposed for optimal load sharing in collaborative manipulation. The method aims to determine collaborative grasp that maximize the distribution of load among the agents.

A real time system was developed to experimentally demonstrate and study the performance of the proposed method using real hardware. Several existing components were used to develop the complete system. It included ROS modules for hardware control, PCL for point cloud processing and MoveIt! for trajectory planning and execution. The architecture of developed system and specifications of the hardware were discussed in Chapter 5. The system made it feasible to test the method proposed for collaborative manipulation on a range of unknown objects with limited prior knowledge of the environment.

To validate the effectiveness of the load sharing grasp solutions for physical systems, two experimental metrics were investigated to quantify quality of the executed collaborative grasps. Experiments were performed on multiple poses of table, box, cylinder and chair objects. The results were discussed in Chapter 6. Experimental results showed that the collaborative grasp solutions and their performance based on cost evaluated correlate with experimental metrics. The method showed potential for collaborative manipulation in human centric environments.

Experiments also revealed shortcomings in the method used for generation of candidate grasps. It was observed that unstable candidate grasps caused a failure of the manipulation task on complex shaped objects.

For the collaborative grasp planning method proposed in this work, all candidate grasps were considered stable and equally good for the manipulation task. The assumption in reality however will not be true as the stability and quality of a grasp will depend on the surface properties and expected contact points of grasp. In future extensions of the work, a better grasp synthesis method can be used and quality of individual grasps can be incorporated for even better grasp selection. Furthermore, a simple lifting task was considered for manipulation in experiments. Future works may experiment the method with complex manipulation tasks.

Bibliography

[1] C. Heyer, "Human-robot interaction and future industrial robotics applications," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 4749–4754.

[2] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Foundations and trends in human-computer interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[3] P. J. Hinds, T. L. Roberts, and H. Jones, "Whose job is it anyway? a study of human-robot interaction in a collaborative task," *Human-Computer Interaction*, vol. 19, no. 1, pp. 151–181, 2004.

[4] S. Thrun, "Toward a framework for human-robot interaction," *Human–Computer Interaction*, vol. 19, no. 1-2, pp. 9–24, 2004.

[5] W. P. Chan, K. Nagahama, H. Yaguchi, Y. Kakiuchi, K. Okada, and M. Inaba, "Implementation of a framework for learning handover grasp configurations through observation during human-robot object handovers," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 1115–1120.

[6] K. W. Strabala, M. K. Lee, A. D. Dragan, J. L. Forlizzi, S. Srinivasa, M. Cakmak, and V. Micelli, "Towards seamless human-robot handovers," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 112–132, 2013.

[7] A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 223–237, Apr 2003.

[8] A. K. Pandey, J.-P. Saut, D. Sidobre, and R. Alami, "Towards planning human-robot interactive manipulation tasks: Task dependent and human oriented autonomous selection of grasp and placement," in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob).* IEEE, 2012, pp. 1371–1376.

[9] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[10] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326 – 336, 2012, autonomous Grasping. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889011001485

[11] R. Haschke, J. J. Steil, I. Steuwer, and H. Ritter, "Task-oriented quality measures for dextrous grasping," in *2005 International Symposium on Computational Intelligence in Robotics and Automation*, June 2005, pp. 689–694.

[12] D. Seredyński, T. Winiarski, K. Banachowicz, and C. Zieliński, "Grasp planning taking into account the external wrenches acting on the grasped object," in *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, July 2015, pp. 40–45.

[13] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.

[14] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1628–1633.

[15] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[16] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, "Visual grasp planning for unknown objects using a multifingered robotic hand,"

IEEE/ASME Transactions on Mechatronics, vol. 18, no. 3, pp. 1050–1059, 2013.

[17] Q. Lei, J. Meijer, and M. Wisse, "A survey of unknown object grasping and our fast grasping algorithm-C shape grasping," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2017, pp. 150–157.

[18] B. Wang, L. Jiang, J. Li, and H. Cai, "Grasping unknown objects based on 3d model reconstruction," in *2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings.* IEEE, 2005, pp. 461–466.

[19] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 292–298.

[20] C. Dune, E. Marchand, C. Collowet, and C. Leroux, "Active rough shape estimation of unknown objects," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 3622–3627.

[21] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 686–693.

[22] J. Ilonen, J. Bohg, and V. Kyrki, "Fusing visual and tactile sensing for 3-D object reconstruction while grasping," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3547–3554.

[23] ——, "Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 321–341, 2014. [Online]. Available: http://dx.doi.org/10.1177/0278364913497816

[24] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE Interna-*

tional Conference on Robotics and Automation (Cat. No.03CH37422), vol. 2, Sept 2003, pp. 1824–1829 vol.2.

[25] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2007.01016.x

[26] K. Rzążewska and M. Luckner, "3d model reconstruction and evaluation using a collection of points extracted from the series of photographs," in *2014 Federated Conference on Computer Science and Information Systems*, Sept 2014, pp. 669–677.

[27] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 559–568. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047270

[28] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02. New York, NY, USA: ACM, 2002, pp. 438–446. [Online]. Available: http://doi.acm.org/10.1145/566570.566600

[29] W. Wang and G. G. Grinstein, "A survey of 3D solid reconstruction from 2D projection line drawings," *Computer Graphics Forum*, vol. 12, no. 2, pp. 137–158, 1993. [Online]. Available: http://dx.doi.org/10.1111/1467-8659.1220137

[30] M. Richtsfeld and M. Zillich, "Grasping unknown objects based on 2 $\frac{1}{2}$d range data," in *2008 IEEE International Conference on Automation Science and Engineering*, Aug 2008, pp. 691–696.

[31] Q. Lei and M. Wisse, "Fast grasping of unknown objects using force balance optimization," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2454–2460.

[32] R. Ala, D. H. Kim, S. Y. Shin, C. Kim, and S.-K. Park, “A 3D-grasp synthesis algorithm to grasp unknown objects based on graspable boundary and convex segments,” *Information Sciences*, vol. 295, pp. 91–106, 2015.

[33] B. Calli, M. Wisse, and P. Jonker, “Grasping of unknown objects via curvature maximization using active vision,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 995–1001.

[34] T. Suzuki and T. Oka, “Grasping of unknown objects on a planar surface using a single depth image,” in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2016, pp. 572–577.

[35] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger, “Early reactive grasping with second order 3D feature relations,” in *Recent Progress in Robotics: Viable Robotic Service to Human.* Springer, 2007, pp. 91–105.

[36] D. Berenson and S. S. Srinivasa, “Grasp synthesis in cluttered environments for dexterous hands,” in *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, Dec 2008, pp. 189–196.

[37] A. R. de Silva and R. Suárez, “Grasping bulky objects with two anthropomorphic hands,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 877–884.

[38] A. Edsinger and C. C. Kemp, “Human-robot interaction for cooperative manipulation: Handing objects to one another,” in *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2007, pp. 1167–1172.

[39] L. Overgaard, B. J. Nelson, and P. K. Khosla, “A multi-agent framework for grasping using visual servoing and collision avoidance,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, Apr 1996, pp. 2456–2461 vol.3.

[40] H. Arai, T. Takubo, Y. Hayashibara, and K. Tanie, “Human-robot cooperative manipulation using a virtual nonholonomic constraint,” in

ICRA'00. IEEE International Conference on Robotics and Automation, 2000. Proceedings., vol. 4. IEEE, 2000, pp. 4063–4069.

[41] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 324–334, 2011. [Online]. Available: http://dx.doi.org/10.1177/0278364910382803

[42] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems.* Springer, 2013, pp. 545–558.

[43] R. Muthusamy and V. Kyrki, "Decentralized approaches for cooperative grasp planning," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec 2014, pp. 693–698.

[44] R. Muthusamy, C. P. Bechlioulis, K. J. Kyriakopoulos, and V. Kyrki, "Task specific cooperative grasp planning for decentralized multi-robot systems," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6066–6073.

[45] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical programming*, vol. 27, no. 1, pp. 1–33, 1983.

[46] "Kinect for Windows sensor components and specifications," https://msdn.microsoft.com/en-us/library/jj131033.aspx, accessed: 2017-07-27.

[47] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The KUKA-DLR lightweight robot arm - a new reference platform for robotics research and manufacturing," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, June 2010, pp. 1–8.

[48] "BH8-282 datasheet," accessed: 2017-07-27. [Online]. Available: http://web.barrett.com/support/BarrettHand_Documentation/BH8-282_Datasheet.pdf

[49] "ROS package to control the Barrett Hand," http://wiki.ros.org/barrett_hand, accessed: 2017-07-27.

[50] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," in *2011 15th International Conference on Advanced Robotics (ICAR)*, June 2011, pp. 67–74.

[51] "The Robotic Operating System," http://www.ros.org/, accessed: 2017-07-27.

[52] I. A. Sucan and S. Chitta, "Moveit!" http://moveit.ros.org.

[53] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[54] "PCL - documentation," http://pointclouds.org/documentation/tutorials/, accessed: 2017-07-27.

[55] "Removing outliers using a Statistical Outlier Removal filter," http://pointclouds.org/documentation/tutorials/statistical_outlier.php, accessed: 2017-07-27.

[56] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[57] "Tilt measurement using a low-g 3-axis accelerometer," Jun. 2014, accessed: 2017-07-27. [Online]. Available: www.st.com/resource/en/application_note/dm00119046.pdf