

A Robot Manipulation Framework of Novel Objects for Elementary Household Tasks

Ana C. Huamán Quispe

July 2015

School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332

Thesis Committee:

George Burdell I,
George Burdell II
George Burdell III
George Burdell IV

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

Abstract

Through the years, robot manipulation research has widened its application field from structured industrial environments to household scenarios. For a robot manipulator to operate in the latter, it must be able to handle diverse type of objects in order to perform equally diverse manipulation tasks. Although theoretically, a robot could take full advantage of on-line databases to identify objects and access their corresponding grasping information, we claim that a robot should also have the capability to perceive a novel object, create a suitable representation of it, and generate a manipulation plan in an efficient and online manner. In this proposal, we present a framework to manipulate novel objects to accomplish elementary household tasks. We present our work on object representation based on superquadrics and its direct relation to fast grasp generation. Second, we present an algorithm to generate manipulation plans for pick-and-place tasks by selecting a grasp that maximizes a metric based on a human heuristic (end-comfort effect). Third, in order to define grounded *elementary* household tasks, we present a study based on human dexterity tests to define benchmark tasks to evaluate our system. Finally, we explore handover tasks, as a final application to our approach, necessary to operate in household environments. We validate our results with experiments on 2 robotic platforms.

Preface

Most of the work presented in this proposal has been published (or is under submission) in diverse conference venues. Below follows the complete list of published work:

- A. Huamán Quispe, H. Ben Amor, and H.I. Christensen. “A Taxonomy of Benchmark Tasks for Bimanual Manipulators”. In: *International Symposium of Robotics Research (to appear)* (2015)
- A. Huamán Quispe et al. “Exploiting Symmetries and Extrusions for Grasping Household Objects”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 3702–3708
- A. Huamán Quispe, H. Ben Amor, and M. Stilman. “Handover Planning for Every Occasion”. In: *IEEE-RAS International Conference on Humanoid Robots (ICHR)*. 2014, pp. 431–436
- A. Huamán Quispe et al. “It takes two hands to grasp: Towards Handovers in Bi-manual Manipulation Planning”. In: *Workshop on Human vs Robot Grasping and Manipulation at Robotics, Science and Systems (RSS)*. 2014
- A. Huamán Quispe, T. Kunz, and M. Stilman. “Generation of Diverse Paths in 3D Environments”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013, pp. 5994–5999
- A. Huamán Quispe and M. Stilman. “Deterministic Motion Planning for Redundant Robots along End-Effector Paths”. In: *IEEE International Conference on Humanoids Robots (ICHR)*. 2012, pp. 785–790

Contents

1	Introduction	1
1.1	Thesis Statement	1
1.2	Overview	2
2	Preliminaries	2
2.1	Physical Platforms	2
2.2	Related Work	3
3	Object Representation	6
3.1	Superquadrics	6
3.2	Fitting Procedure	6
3.3	Implementation Considerations	8
3.4	Analysis of Noise and Partial Visibility	9
3.5	Objective Function Evaluation	10
3.6	Comparative Results	11
4	Grasp Generation	13
4.1	Assumptions and Considerations	13
4.2	Generating grasps from parametric equations	13
4.3	Generating reach paths	14
5	Task-Based Grasp Prioritization	15
5.1	A metric to prioritize grasps	15
6	Handover Interaction	17
6.1	Self-Handover	17
6.2	Multi-Agent Handover	18
7	Task Evaluation: Benchmarks and Metrics	18
8	Summary and Proposed Work	21
8.1	Summary	21
8.2	Proposed Work	21
8.3	Schedule of Work	22

1. Introduction

During the last decades, research in robot manipulation has gradually increased its scope from constrained industrial environments to further include unstructured human scenarios. Many factors contributed to this shift, such as the growing availability of hardware platforms, more affordable actuators and sensors (such as end-effectors and RGB-D cameras), increased processing power and online resources that facilitate information sharing.

Robotic manipulation in human scenarios presents unique challenges and constraints:

- Wide variety of objects to interact with, many of which might be novel for the robot.
- Perceptual information incomplete or inaccurate.
- Potentially many different manipulation tasks per each object.
- Online performance due to human requirements.

Most of the pioneering work on manipulation consisted on analytical approaches, considering grasping as an optimization problem on finger placement [59, 53, 28, 22]. By assuming that the object 3D models are always available (a natural assumption in controlled scenarios), the grasping problem could be addressed by building a database of objects for which corresponding grasps could be retrieved when needed. Although in human scenarios the number of potentially encountered objects is orders of magnitude larger, the advent of network resources allows for robots to exploit the advantages of external databases and vast computational infrastructures, empowering the robot with processing abilities it would not have otherwise [48].

A database, however, cannot store all the different instances of existing objects. Furthermore, since many robotic hands are available, a database might not have grasp information for a particular hand model. Given these reasons, a robotic system must be capable of operating *novel* objects. For household environments, *operating* implies performing elementary household tasks, relevant to the human end-user.

The work presented on this proposal concerns the manipulation of novel objects for elementary household tasks as well as handover tasks, equally necessary for a house-operating robot. We select Superquadrics [5] as our object representation given its ability to describe (with a small set of parameters) a wide variety of shapes that are characteristic of household objects [29]. Our work encompasses the complete, end-to-end pipeline implemented in our 2 robotic platforms to validate its applicability.

1.1 Thesis Statement

This proposal presents a framework for robot manipulation of novel objects in household environments. This framework is based on the use of Superquadrics (SQs) as its object representation unit. We postulate that SQs present a reasonable tradeoff of a small number of parameters capable of representing a significant amount of convex, single-part household objects. We base this assumption both in existing work in literature as well as on experimental results obtained from using a recently proposed benchmark object database.

Our proposed framework contains all the building blocks to perform elementary household manipulation tasks: From grasp generation to grasp selection based on a proposed metric based on a human heuristic. We present a taxonomy of benchmark tasks in order to define what is an elementary task and its corresponding metrics. Finally, we address handover tasks, which are necessary in the context of a house-operating robot that interact with human end-users.

We propose to demonstrate the applicability of our approach by performing experiments in our two physical robotic platforms and to compare its performance with respect to corresponding human execution.

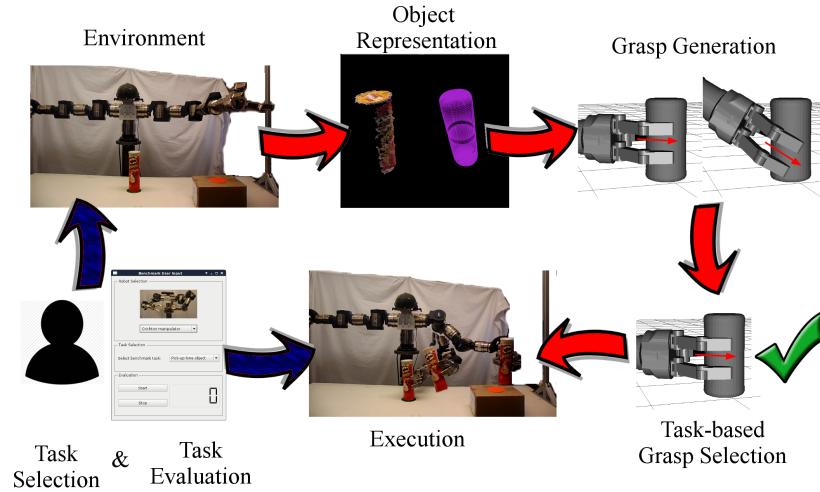


Figure 1.1: Diagram depicting the different parts of our end-to-end pipeline, to be explained in the successive chapters 3 to 6.

1.2 Overview

The rest of this proposal is organized as follows, with some of the chapters diagrammatically illustrated on Figure 1.1:

- *Preliminaries*: In Chapter 2 we present the hardware platform, the assumptions and constraints considered in the experiments. We also present a condensed summary of relevant related work.
- *Object Representation*: Chapter 3 presents our work on object representation using SQs and our implementation insights combining mirroring and hierarchical fitting. We also present our implementation results comparing diverse objective functions and the effects of noise and downsampling in the optimization results.
- *Grasp Generation*: In Chapter 4 we present our approach to grasp generation based on the analytical form of the superquadrics. We also present our experimental results using a subset of the YCB Dataset[14].
- *Grasp Selection and Manipulation Planning*: In Chapter 5 we present our work on selecting grasps according to our proposed manipulability-based metric based on the end-comfort effect.
- *Handover planning*: In Chapter 6 we present our previous work on manipulation involving two agents (either human-robot handover or handover between two limbs by the same robot), which is a needed skill for a robot interacting with humans.
- *Task Selection and Evaluation*: There are myriad of tasks in a household. A robotic system should be able to perform an elementary subset of these. In chapter 7 we present a study based on human dexterity tests to determine basic, useful tasks.
- *Future work*: Chapter 8 presents work proposed and the corresponding timeline of events.

2. Preliminaries

In this chapter we present background information regarding the current state of the art in robot manipulation and we also introduce the robotic platforms that are used through this document. Section 2.1 describe the physical robots and their mode of operation whereas Section 2.2 presents a summary of relevant work related to this proposal.

2.1 Physical Platforms

The robots used for the experiments cited on this document (Figure 2.1) are summarily described as follows:



Figure 2.1: Robots used on experiments

Crichton

Crichton is a bimanual manipulator consisting on a pair of limbs, each of them constituted by (1) A 7-DOF arm, (2) A F/T sensor at the wrist, (3) A 3-finger Schunk Dexterous Hand, featuring 2 fingers and a thumb. Each of these have 2 DOF. An extra coupling DOF allows both fingers to rotate and face either the thumb or each other.

Each arm has an extended length of 1.04 m. The hand length (from base to finger tip extended) is of 0.25 m. Each arm base is firmly connected to a plate (shoulder) which can be rotated with a vertical motor acting as a torso. The robot is fixed, with a shoulder height of 1.23 m above the floor.

Crichton obtains perceptual information through the use of a Carmine 1.09 attached on top of its shoulder. This sensor obtains RGB-D data in a short range (0.35m - 1.4m), adequate for manipulation of objects located on a tabletop in front of the robot.

The system architecture implemented follows the classical *Sense-Think-Act* paradigm, carried out with 2 PCs: (1) The Control PC is in charge of controlling the limbs' movements, fetching motor states and sending velocity motion commands at 100Hz. (2) The Planning/Perception PC is in charge of the Sense/Think components, processing the RGB-D input and running the online planning algorithms in our simulation environment, DART [24], updated with the perceptual information obtained online.

All the processes communicate with each other using the ACH IPC library [23].

Alita

Similar to Crichton, Alita is also a bimanual manipulator with two 7-DOF arms. It, however, has a 1-DOF gripper as end-effector. Its maximum aperture size is of 10 cm. Each finger has a length of 11 cm. In contrast to Crichton, Alita does not have a torso motor. Its perceptual input similarly comes from an Asus Xtion RGB-D sensor pointing towards the table on front of the robot.

2.2 Related Work

In this section we give a brief overview of work directly relevant to this proposal.

Object Representation

There is not an unified object representation for robotic manipulation. Diverse approaches have been proposed in the literature, differing on the assumptions considered and the available hardware used to acquire external information.

Most of the original work on manipulation assumed 3D model of objects are always available, hence objects are implicitly represented by meshes [33]. In household environments, however, this assumption does not always hold. With depth data (i.e. using 2D lasers, Time-Of-Flight cameras), objects could further be represented by using pointclouds either in their raw form or through local descriptors encapsulating geometrical characteristics of relevant parts of an object. Along these lines, Rusu conducted pioneering work on semantic 3D object models [67].

Another type of approaches seek to represent objects by approximating them to primitive shapes, which has the advantages of requiring a small number of parameters describing the object geometry.

In [44], Huebner et al. proposed to represent shapes based on box primitives. Miller et al.[58] considered a wider range of shapes (cylinders, boxes, spheres) for grasping purposes and Goldfeder used superquadrics [32]. In an approach more tailored to grasping strategies, Przybylski et al. proposed the *grid of medial spheres* representation [64]. In most of these approaches, the 3D model is assumed available and/or the results presented are mainly on simulation, making it hard to decide their applicability on real-world data.

It is interesting to observe that object representation techniques varied over the years according to the capability of affordable sensors. On 2008, Saxena et al. [70] represented objects (for grasping purposes) as a set of good-to-grasp points obtained from 2D images and an offline learning phase. Due to the advent of accessible depth-sensors, approaches dealing with pointclouds became more extensively used. On [45], Huebner extended his work on box primitives to approximate partial pointclouds online. On 2010, Hsiao [18] successfully used raw pointclouds, and their bounding boxes, to generate grasps for the PR2 based on a set of heuristics. Bohg et al.[10] proposed an interesting approach to mirror partial pointclouds and reconstruct their corresponding meshes, assuming the objects are symmetrical.

Grasp Planning

In this section, we review work concerning grasp synthesis and grasp selection. For a more detailed review of previous research in the area, we refer the reader to the representative reviews from Bohg et al. [11] and Sahbani et al. [69].

Grasp planning consists of the problem of finding a grasp configuration that satisfies a set of criteria relevant to a grasping task [11]. Pioneering work on grasp selection was conducted by Cutkosky [21], who observed that humans select grasps in order to satisfy 3 main types of constraints: (1) Hand geometry, (2) object characteristics and (3) task constraints. Sahbani et al. [69] classifies grasp planning approaches in 2 general classes: Analytic approaches and Empirical approaches.

Analytical methods consider grasping as an optimization problem, in which finger location is the result of minimizing a metric related to stability (i.e. force-closure [9]). A particularly known metric is the ϵ metric proposed by Canny [30]. For approaches to be tractable, assumptions regarding friction, contact types and 3D models are considered. A few example of proposed strategies are [59, 53, 28, 22].

Classical metrics are also used in popular software libraries, such as GraspIt! [17] and OpenRAVE [25], to filter out unstable grasps and to rank the remaining candidates according to their metric value. However, it has been noted that in real-world execution, these grasps tend to be fragile [11]. In contrast, empirical approaches, which generate grasps based on human data such as heuristics or kinesthetically-taught trajectories, have shown comparable and even better performance on physical implementations [3, 4]. Multiple approaches presenting alternative metrics combining classical metrics, human heuristics and additional environmental factors were also proposed [7, 65, 49]

Most of the approaches mentioned above have been tested with simple objects, as most common household objects are simple convex shapes, however methods to explicitly deal with composited objects have also been proposed, such as [32], in which Goldfeder et al. used superquadrics trees to represent complex objects in simulation. In a similar manner, Aleotti et al. [2] represent an object as a Reeb Graph by obtaining a full pointcloud from multiple views, then a mesh is generated and used as input to generate grasps.

To conclude, it is important to observe that the majority of the existing work only considers the hand and object constraints, not taking into account the task to be executed once the object is grasped. A few of the works that explicitly address the use of tasks in grasp selections are [52], where Li and Sastry proposed the concept of the *task ellipsoid*, which maximizes the forces to be applied in the direction of the task. Pollard [51] use shape matching and a task-based metric to select grasps for specific tasks in simulation (a similar metric was used in more recent work in [27]); Pandey et al. [61] proposed a framework for grasp selection in a human-robot interaction scenario, where the task was more loosely constrained.

Manipulation Planning

The problem of grasp planning is usually considered isolated from arm planning. Sample-based approaches, such as RRT [50] and PRM [12] were among the first to be widely adopted due to their ability to be used for high-dimensional spaces, such as the ones found in redundant robot arms ($DOF > 6$). Diverse frameworks based on sampling approaches incorporate task constraints in the search process, such as Task Space Regions [6].

Another type of approaches attempt to solve both the motion planning problem and the grasp selection at the same time: In [75] Vahrenkamp et al. proposed Grasp-RRT in order to perform both grasp and arm planning combined. In a similar vein, Roa et al. also proposed an approach that solve both problems simultaneously [31]. Both approaches focus on *reaching tasks*. While reaching is a necessary part of every manipulation task, it is usually just an intermediate step.

Elementary Household Tasks and Benchmarking

Up until now we have referred to *manipulation tasks* in a rather general way. In this proposal our domain scenario are human environments such as households, hence the tasks that we aim to solve should be relevant to the corresponding end-users, that is, regular humans.

There are many possible tasks a robot can be required to execute. Selecting a subset of *elementary* tasks is important because: (1) It allows different researchers to compare their results upon the same experimental suite. and (2) It avoids the danger of cherry-picking tasks that might be interesting from a research point of view, but not necessarily useful for end-users. Multiple studies on task prioritization have been developed with different objectives in mind. Some of the most popular are in the form of robotic competitions, in which a set of manipulation tasks is arbitrarily chosen and diverse teams present their own particular solutions. Examples of these include the DARPA ARM project [37], NAO@Home[76], the Darpa Robotics Challenge [63] (which included manipulation among other required skills) and more recently the Amazon Picking Challenge[73].

Besides competitions, there is increasing interest in developing a principled, structured framework for manipulation benchmarking [20, 46, 55]. One of the earliest approaches in this category is the Cranfield test [19], used in the assembly industry to compare robot manipulation performance without considering perception. In general, most approaches can be roughly divided in two categories: (1) Task-based approaches, which seek to define a set of tasks (actions) upon which metrics can be calculated [35]. (2) Object-based approaches, which define a set of standard common objects (and their physical and geometrical properties) [14, 57, 16]. In this proposal, we present a task-based benchmark study based on human dexterity studies.

Handover Planning

For a robotic system operating in a household, handover capabilities are not only useful but necessary. In this proposal we consider two specific handover scenarios: (A) *Self handover*: A robot passes an object from one hand to the other. This is useful to take advantage of the full workspace of bimanual robots. (B) *Human-robot handover*: Robot picking up objects on demand of a human receiver. Most of the current literature refers to the second type of handover, so we will mostly cite works of this type:

In [71], Sisbot and Alami presents a planner to find handover poses to give an object to a human receiver. The planner considers the human comfort when planning the robot's arm movements but assumes that the object is already grasped by the robot. Aleotti et al. [1] presents a more general approach which plans a reaching action such that the object can be delivered in a comfortable pose for the user. The main disadvantage of this approach is that it considers assumptions that cannot be generalized (i.e. the robot always can reach the area directly in front of the human receiver).

Handover planning is specially challenging because it involves humans, whose comfort we wish to maximize, but at the same time it is not a hard-constraint, as humans can adapt their posture to facilitate the handover action. In [13] , Cakmak et al. studied human preferences for handovers and in [56] Mainprice and colleagues investigated how the spatial preferences of a human partner can be used (in the form of a mobility metric) to facilitate the motion planning of a robot giver.

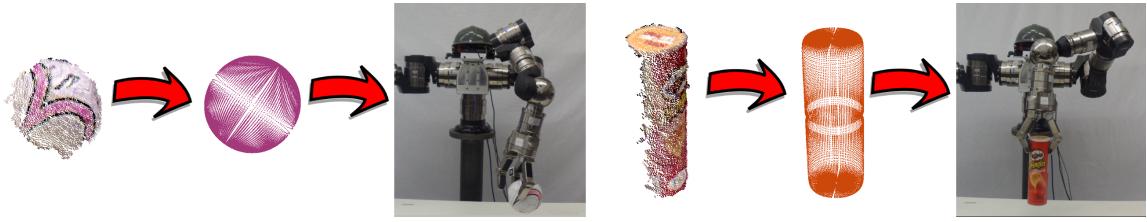


Figure 3.1: Example results of the superquadric fitting results in two input clouds

3. Object Representation

In this section we discuss *superquadrics* (SQ), the object representation used through this document. We chose them over other available representation types, such as the ones discussed on Section 2.2, for a number of reasons: (1) ability to represent a wide variety of geometrical shapes (one-part, convex objects) (2) small number of parameters, which is crucial to perform the pointcloud fitting in a fast manner (3) partial occlusion tolerance (4) existing previous work that justifies its usage and (5) possibility of extension to represent more complex objects by using a part-level approach).

In the following lines, we succinctly introduce the SQ mathematical formulation. An explanation of the fitting procedure follows, comparing the performance of 5 different objective functions using both synthetic and real data obtained from our physical robot platform. We conclude by presenting important practical considerations that are critical to the successful performance of this approach in real environments, such as the selection of an adequate downsampling value, a mirroring pre-processing to constrain the minimization process and an optional multi-scale step to reduce processing times.

3.1 Superquadrics

Superquadrics (SQ) are a family of geometrical shapes that can represent diverse types of simple objects. Superquadrics, in their canonical form, are defined by points $\mathbf{p} = (x, y, z)$ satisfying the following implicit equation:

$$f(\mathbf{p}, \Lambda) = \left(\left(\frac{x}{a} \right)^{\frac{2}{e_2}} + \left(\frac{y}{b} \right)^{\frac{2}{e_2}} \right)^{\frac{e_2}{e_1}} + \left(\frac{z}{c} \right)^{\frac{2}{e_1}} = 1 \quad (3.1)$$

where a, b, c are the dimensions of the SQ along the X, Y and Z directions. e_1 and e_2 are the shape parameters, which define the section of the superquadric in the XY plane and in the plane perpendicular to it, respectively. In general, e_1 and e_2 could adopt any real value. Since for our purposes, most of the objects to be represented are simple convex objects we restrict these parameters to $e_1, e_2 \in [0.1, 1.9]$.

In its canonical form, Λ has then 5 parameters. To account for rigid transformations, 6 additional parameters are added. This makes a total of 11 parameters: $\Lambda = (a, b, c, e_1, e_2, t_x, t_y, t_z, \gamma, \beta, \alpha)$.

3.2 Fitting Procedure

In order to find the parameters Λ that best fit a pointcloud P , we must minimize the following expression:

$$\min_{\Lambda} F(\Lambda) = \min_{\Lambda} \sum_{i=1}^N f^2(p_i, \Lambda) \quad \forall p_i \in P \quad (3.2)$$

which is a non-linear least-squares minimization problem. We use the Levenberg Marquardt (LM) algorithm[60], an iterative method to solve Equation 3.2. The LM update rule is the following:

$$\Lambda_{i+1} = \Lambda_i - (H + \lambda \text{diag}(H))^{-1} \nabla F(\Lambda_i) \quad (3.3)$$

where: $H = J(\Lambda_i)^T J(\Lambda_i)$, $J(\Lambda_i) = \frac{\partial F(\Lambda_i)}{\partial \Lambda_i}$ and $\nabla F(\Lambda_i) = J(\Lambda_i)^T r(\Lambda_i)$. $f(p_i, \Lambda)$ is a non-linear objective function that expresses the discrepancy between a point p_i and a superquadric defined by Λ . Different types of $f(\cdot)$ functions have been used through the literature. In the next section we

present 5 objective functions and subsequently compare their performance using classical metrics. To the best of our knowledge, with the exception of [78], no other study has been done to purposely compare the cited functions.

3.2.1 Objective Functions

The objective functions $f(\cdot)$ to be evaluated and compared are the following:

Radial Function (f_r)

Originally proposed by Gross and Boult [34]. f_r represents the accumulated radial distance from each point to the surface of the superquadric. The radial distance is an approximation to the true Euclidean distance. Given a line joining a point and the origin of the superquadric, the radial distance is measured as the length of the segment between the point and the intersection of the line with the superquadric surface.

$$f_r(\mathbf{p}, \Lambda) = \sqrt{x_c^2 + y_c^2 + z_c^2} \left| 1 - f(\mathbf{p}, \Lambda)^{\frac{e_1}{2}} \right| \quad (3.4)$$

where:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} R & -Rt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{with:} \quad \begin{aligned} R &= R_z(\alpha)R_y(\beta)R_x(\gamma) \\ t &= (t_x, t_y, t_z)^T \end{aligned}$$

f is the same from Equation 3.1.

Solina Function (f_s)

This function, the most popularly used, represents the accumulated error of the function f with respect to its optimal value of 1, evaluated at each point in P . In [72], the authors suggested to use a scale factor representing the volume of the superquadric in order to generate solutions with the minimum possible dimensions:

$$f_s(\mathbf{p}, \Lambda) = \sqrt{abc} (f(\mathbf{p}, \Lambda)^{e_1} - 1) \quad (3.5)$$

Chevalier Function (f_c)

This function was proposed in [15]. It borrows factors from both F_r and F_s .

$$f_c(\mathbf{p}, \Lambda) = \sqrt{x_c^2 + y_c^2 + z_c^2} (f(\mathbf{p}, \Lambda)^{\frac{e_1}{2}} - 1) \quad (3.6)$$

Ichim Function (f_i)

This function corresponds to an approximation of the radial function scaled with the volume of the superquadric.

$$f_i(\mathbf{p}, \Lambda) = \sqrt{abc} \sqrt{x_{c_i}^2 + y_{c_i}^2 + z_{c_i}^2} (f(\mathbf{p}, \Lambda)^{\frac{e_1}{2}} - 1) \quad (3.7)$$

Alternative Function 5 (f_5)

We additionally tested a function we devised based on the objective functions found in the literature. f_5 is simply a version of Solina's function scaled by each point's norm (distance with respect to the canonical superquadric) rather than with the superquadric principal dimensions.

$$f_5(\mathbf{p}, \Lambda) = \sqrt{x_c^2 + y_c^2 + z_c^2} (f(\mathbf{p}, \Lambda)^{e_1} - 1) \quad (3.8)$$

3.2.2 Metrics used for comparison

In this subsection we describe the quantitative metrics used to compare the performance of the objective functions evaluated.

Goodness of Fitness (E_g)

This metric measures the error of each point with respect to the implicit function (f , which should be 1 from Equation 3.1). Notice that our metric is slightly different as the one proposed in [36]. Instead of using the absolute value of the error we just use its squared value. This metric has no units.

$$E_g(\Lambda, P) = \frac{1}{N} \sum_{i=1}^N (f(\mathbf{p}_i, \Lambda)^{e_1} - 1)^2 \quad (3.9)$$

Radial Distance (E_r)

This metric quantifies the added radial distance from the points in the cloud with respect to the origin of the superquadric. f_r is the same as in Equation 3.4:

$$E_r(\Lambda, P) = \frac{1}{N} \sum_{i=1}^N f_r(\mathbf{p}_i, \Lambda) \quad (3.10)$$

This metric has length units (m).

Volume Error (V_r)

This metric measures the difference between the real volume and the approximated volume of the fitted superquadric. The volume of a superquadric can be expressed with the following expression:

$$V(\Lambda) = 2(abc)(e_1 e_2) B\left(\frac{e_1}{2} + 1, e_1\right) B\left(\frac{e_2}{2}, \frac{e_2}{2}\right) \quad (3.11)$$

where B is the beta function. In this document we express V_r as the ratio of the volume error with respect to the real volume(%):

$$V_r = \frac{|V(\Lambda_{\text{approx}}) - V(\Lambda_{\text{real}})|}{V(\Lambda_{\text{real}})} \times 100\% \quad (3.12)$$

Fitting time (t)

Measures the total fitting time for a pointcloud. This metric has time units.

3.3 Implementation Considerations

Solving a non-linear least-square optimization problem with 11 parameters is not trivial. Although the minimization results heavily depend on the objective function used, there are additional factors involved, specially when dealing with real data. In this section we examine 2 main aspects:

- Each $\mathbf{p}_i \in P$ adds a residual factor to Equation 3.2. Given that a dense pointcloud can potentially have thousands of points, it would be desirable to apply downsampling to the input cloud, provided that the filtering does not affect the quality of the fitting result.
- Real pointclouds are rarely complete. Only the points \mathbf{p}_i facing the RGB-D sensor are stored and they present some degree of noise. As we will show later on, having partial clouds negatively affect the quality of the fitting results. We propose to use a mirroring approach to alleviate this error source.

In the following 2 subsections we will analyze these factors (downsampling and partiality/noise effects). We will use synthetic data to generate several different test cases.

Implementation Details

All the experiments in this section were carried out on a quad-core Intel Xeon E5-1620 (3.60GHz) with 8GB of RAM, running Debian 8.0. All the functions were written in C++, using the Levenberg-Marquardt implementation of the levmar library [54]. Additionally, all the experiments use $N = 500$ as a referential number of tests per experiment. This number was experimentally determined as to be the minimum number of tests needed to obtain consistent results (with $N > 500$ results present a similar pattern).

3.3.1 Downsampling

As it was pointed out in [26], using a downsampled version of an originally dense pointcloud dramatically reduces the minimization processing time. Ideally, for generic applications, the downsampling voxel size should be chosen such that the SQ parameters obtained from the original pointcloud and from the downsampled version are comparable.

In order to evaluate the downsampling effect, we performed a set of 500 randomized experiments in which all the 11 parameters of Λ were randomized (within the limits shown in Table 3.1). The input to each experiment was a perfect SQ pointcloud described by Λ . This input was downsampled with 6 different voxel sizes, ranging from 0.5cm to 3cm with a step of 0.5cm. We applied the minimization procedure to each downsampled version and recorded 3 error metrics (E_g , E_r and ΔV). In total we processed 3000 clouds (6 per experiment instance).

Param.	Range
a, b, c (m)	[0.025, 0.4]
α, β, γ (rad)	$[-\pi, \pi]$
t_x, t_y (m)	[-0.8, 0.8]
t_z (m)	[0.35, 1.4]
e_1, e_2	[0.1, 1.9]

Table 3.1: Synthetic data limits

For the minimization procedure we used f_r as the objective function. For different objective functions the values of the error metrics would be different, but the tendency (increasing/decreasing) is the same, so for this section we present only results with f_r . Our results are shown in Table 3.2. We observe that the downsampling alone have no significant effect on the final results. As it can be seen, the volume error is in the worst case less than 2%, whereas the average radial distance is less than 4 mm per point.

For most of our experiments we consider a conservative voxel size of 0.015m, as a good compromise between speed gain and error control.

Table 3.2: Error with respect to downsampling voxel size. Average results for N=500 random trials

Metric	0.005m	0.01m	0.015m	0.020m	0.025m	0.030m
E_g	0.040	0.054	0.061	0.064	0.069	0.069
E_r (mm)	2.3	3.3	3.8	4.2	4.3	4.6
ΔV (%)	0.573%	0.697%	0.821%	0.994%	1.201%	1.472%
$t(s)$	6.246	3.787	1.763	1.425	0.919	0.669

3.4 Analysis of Noise and Partial Visibility

Real pointclouds are incomplete and noisy. In this experiment, our goal is to evaluate the effect of both noise and partial visibility in the minimization results.

Similarly to the procedure in Section 3.3.1, we generate a set of synthetic data ($N = 500$) with the 11 parameters of Λ randomized. Each instance consisted of 4 pointclouds: (1) *Full*: A perfectly fit superquadric. (2) *Full noisy*: Similar to *full* with added Gaussian noise ($\delta = 0.0025$). (3) *Partial*: Obtained from *full* sliced by a plane containing the centroid and perpendicular to the line joining the origin and the centroid. (4) *Partial Noisy*: Same as *partial* with added Gaussian noise ($\delta = 0.0025$). An example of a random sample is shown in Figure 3.2



Figure 3.2: Sample instance with $e_1 = 0.5$ & $e_2 = 0.5$. L to R: Full, noisy, partial full and partial noisy cloud

The objective function used for minimization was F_r . The results are shown in Table 3.3. It can be observed that the effect of partiality (last 2 columns) is preponderant, much more than the effect of noise (first 2 columns). With partiality, we observe a severe increase on the radial error (from 3mm to 1.4cm). Since volume is directly related to the distance metric, ΔV also soars from less than 1% to 20%.

The last column shows the variation in average times. It is worth noticing that the last two columns report the times for partial pointclouds, which have half of the number of points than the full clouds in the first 2 columns. However, due to the effect of noise and partial visibility, the processing times increase.

Table 3.3: Error with respect to noise and partial view. Average results for N=500 random trials

Metric	Full	Full noisy	Partial	Partial Noisy
E_g	0.061602	0.062367	1.235281	25.410158
$E_r(m)$	0.003733	0.003822	0.013045	0.014554
$\Delta V(\%)$	0.827583%	0.769512%	19.9356%	20.641092%
$t(s)$	1.602047	1.865845	5.078875	4.604418

The increased error due to partial visibility can be explained by the fact that, when a pointcloud is incomplete, its surface is not closed. The limits in one or more dimensions are not implicitly enforced, letting the fitting procedure free to increase the size of the superquadric as much as it reduces the error.

3.5 Objective Function Evaluation

In the previous section, we used synthetic data to gather information regarding the influence of down-sampling, noise and partiality in the fitting procedure. In this section, our goal is to select an objective function to produce the best possible fitting parameters.

3.5.1 Experimental Setup

The scenario used for the experiments described here is shown in Figure 3.3. A PrimeSense sensor is attached on top of the robot's shoulders (highlighted with a blue rectangle) and pointed down towards the table. The 41 objects evaluated are a subset of the YCB Dataset [14]. We further divided this set in 5 groups to make the results easier to visualize (Figure 3.4). As it can be observed, the first 3 groups correspond to defined geometric shapes, whereas the last 2 are non-homogeneous.

The procedure to obtain the input data follows the standard tabletop segmentation pipeline implemented by the PCL library[68]: First, the table is segmented as the biggest plane in the field of view. Second, after filtering out the table points, the remaining points are clustered using an Euclidean approach, with a minimum cluster size of 300 points and a distance between clusters of at least 3 cm. We perform our evaluation over these clusters. In the second row of Figure 3.4 the segmented areas are highlighted in color. Notice that, while most of the segmentation results capture the majority of the objects' visible pointclouds, there are a few cases (i.e. the SPAM can in the boxes set) in which the segmentation is not complete.



Figure 3.3: Robot, tabletop scenario and some of the 41 objects evaluated

3.5.2 Mirror Approach to alleviate partial-view effect

In Section 3.4 we noted the effect of partiality in the fitting results for F_r . When we first tried to apply the minimization procedure to the segmented pointclouds, we found that in a number of cases the fitting far exceeded the real size of the object (see Figure 3.5).

In [43], we proposed to use a mirroring approach (such as the one in [10]) to bound the dimensions of the superquadric principal axis. Mirroring the pointcloud equals to add the missing visual constraint on the back of the object, allowing a more accurate fit. In the rest of this section, we will use the mirroring approach as a preprocessing step to our raw input clouds.



Figure 3.4: The 5 object sets: Balls, cylinders, boxes, fruits and miscellaneous objects. Segmentation results. Notice that for some objects (SPAM can and plastic containers) the segmented points are incomplete.



Figure 3.5: Erroneous results obtained when not using the mirroring approach

3.6 Comparative Results

We performed the Levenberg-Marquadt minimization procedure with each of the 5 objective functions described over our subset of 41 objects. The raw input pointclouds can be observed in the first column on Figure 3.7. The metrics we used to compare the approaches were the same as presented in the previous section with the exception of the volume error. Initially we had considered using the reconstructed ground truth meshes of the objects (available in the YCB website). However, we observed these meshes were not usable for our purposes (reconstruction inaccuracies).

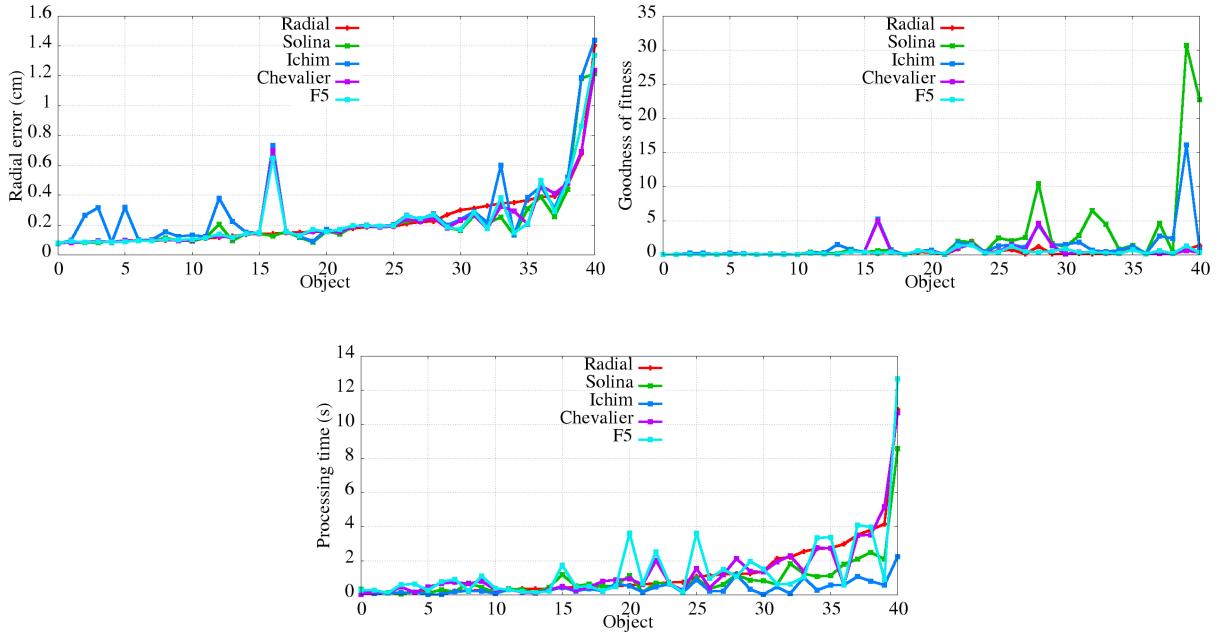


Figure 3.6: Radial error, Goodness-of-fitness error and processing time results for the 41 objects

The metric results are depicted on Figure 3.6, the x-axis represent each of the 41 object instances and the y-axis shows the measured metric. In order to make the plots clearer to the reader, we ordered the objects in ascending order of their metric (i.e. the objects with $x = 0$ and $x = 40$ in the first plot are the objects that have the lowest and highest average goodness-of-fitness metric from the set). From the



Figure 3.7: Left column: Input pointclouds, right column: Fitting results. Point density was duplicated for better visualization.

plots, we draw the following observations:

- E_g : F_r and F_5 have consistently lower errors. The other 3 functions have a similar behavior with exceptions on a few of the objects instances (the spikes on the plot). Since this error is non-dimensional, it is unclear how to evaluate this error.
- E_r : F_r presents a consistent behavior, keeping low error values. F_s presents a close similar performance with a few small peaks. It is worth noticing that with a few exceptions, most of the error values per object have a relative difference not more than a few millimeters.
- t : F_i present the lowest times for all the object tests.

If we were guided strictly by the metric results E_g and E_r , we could conclude that effectively, the radial function F_r is the best objective function given the numerical results. This conclusion would confirm the results presented by Zhao [78], who performed comparison experiments between F_s and F_r using synthetic and real data, concluding that F_r had the best overall performance. However, when confronting the resulting fitted models with the real ones, we observe that the fitting results are in many cases equivalent. In particular, 2 functions F_s and F_i , consistently produce more adjusted results, specially in cases in which the shape is hard to determine for segmentation limitations, such as in the cylinder group.

From the experiments realized, we concluded that for our purposes, the Ichim function F_i was the most suitable for our application. It is notably faster and the results are comparable with the ones produced by the other functions evaluated. In [43] we presented preliminary results of our mirroring approach combined with a hierarchical fitting procedure [26] which further helped reduce processing times.

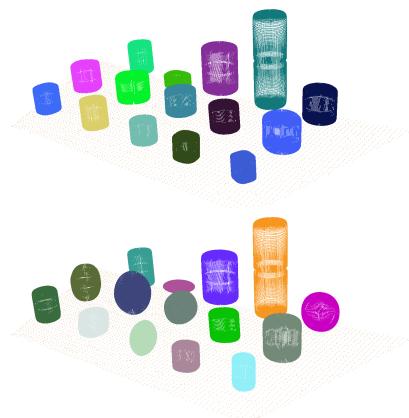


Figure 3.8: Comparing fitting results of Ichim (upper) vs Solina (lower).

4. Grasp Generation

In Chapter 3, we presented our approach for object representation. In this section, we summarize our approach to grasp generation, which builds upon the superquadric parametric representation of the object to systematically generate candidate grasps. We also detail the path planning algorithm used to generate the jointspace trajectories for the robot arm in order to execute reach movements.

4.1 Assumptions and Considerations

The robotic hands used for the experiments presented in this thesis proposal are 2, shown in Figure 4.1:

- A Schunk gripper with an aperture width of 10 cm, a finger length of 11 cm, and 1 degree of freedom (DOF).
- A Schunk Dexterous Hand, a 3-fingered end-effector with 2 DOF at each finger and thumb and an additional DOF coupling both fingers. This last DOF is not considered in our grasp planning approach, meaning that we consider the fingers and the thumb always opposing each other (using only 6DOF).

For both cases, we consider a coordinate frame H with origin in the Tool Center Point (O_h), a 3D reference point conveniently placed above the center of the palm face. The axis x_h runs parallel to the line joining the fingers (or in the SDH case, the line joining the thumb and the opossing fingers). The z_h axis represents the approach direction, and is a vector pointing outside from O_h , perpendicular to the palm. The axis y_h is the corresponding to a standard right-hand frame.

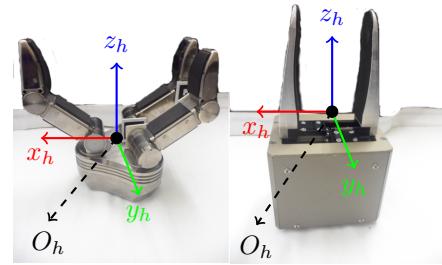


Figure 4.1: 3-fingered hand and gripper

4.2 Generating grasps from parametric equations

A common approach to generate grasps for known objects is to use the object normals as candidate approach directions for the hand. If the object mesh is available, then the normals of the triangle mesh are used. Ideally, the normals should be evenly distributed on the object surface. A disadvantage of using meshes is that the normal distribution depends on the triangulation quality of the mesh. An even distribution is obtained with a fine-grained mesh, which implies longer processing times. By using our object representation, we take advantage of the fact that the normals can be obtained in a fast manner, with their density being determined only by a sampling input parameter.

As we pointed out in Chapter 3, a superquadric can be defined with its implicit expression (Equation 3.1). To generate grasps, however, it is more convenient to define the points with the equivalent explicit form, parameterized with the 2 variables ω and η :

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a \cos^{\epsilon_1} \eta \cos^{\epsilon_2} \omega \\ b \cos^{\epsilon_1} \eta \sin^{\epsilon_2} \omega \\ c \sin^{\epsilon_1} \eta \end{bmatrix} \quad \text{with} \quad \begin{aligned} \frac{\pi}{2} < \eta < \frac{\pi}{2} \\ \pi < \omega < \pi \end{aligned} \quad (4.1)$$

The outward-pointing normal corresponding to each point is defined as:

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \frac{1}{q} \cos^{2-\epsilon_1} \eta \cos^{2-\epsilon_2} \omega \\ \frac{1}{b} \cos^{2-\epsilon_1} \eta \sin^{2-\epsilon_2} \omega \\ \frac{1}{c} \sin^{2-\epsilon_1} \eta \end{bmatrix} \quad (4.2)$$

In order to have a proper distribution of normals on the surface, the sampling of points p must be uniform. However, since the superquadric equation is highly non-linear, uniform sampling cannot be always obtained by sampling ω and η at equal intervals, in particular, for objects whose geometry is expressed with low values of e_1 and e_2 , such as boxes and cylinders (see Figure 4.2). Instead, we use the method proposed by Pilu and Fischer [62] to obtain a homogeneous point distribution. The only sampling input parameter is N , which controls the point density.

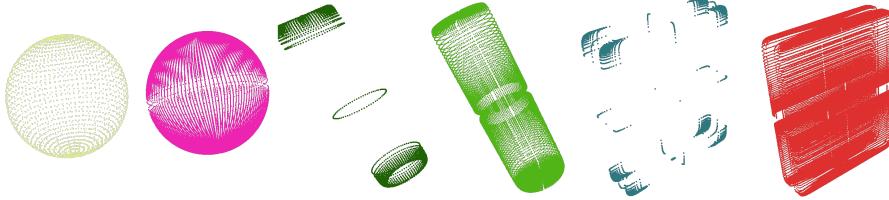


Figure 4.2: Naively sampled superquadric vs uniform sampling from [62]

A set of candidate grasps is generated from each (p, n) pair. The hand TCP O_h is set above this point, with its z_h axis pointing opposite to n . The axis joining the opposite fingers x_h is aligned with the object's smallest principal axis. The fingers are initially set with a default open configuration and are progressively closed until collision with the object is detected. Implicitly with this approach, power grasps (palm close to the object surface) are favored. Grasps with less than 3 contact points are filtered out, as well as grasps that are not kinematically feasible to reach with the arm.

4.3 Generating reach paths

The procedure above generates a number of candidate grasps \mathcal{G} from which one has to be chosen in order to generate a reaching arm movement to achieve it. How to choose a grasp depending on the task will be dealt with in Chapter 5. Assuming that a grasp g has already been chosen, we generate the reach path by using a standard IKBiRRT [8].

Table 4.1 shows results of picking up experiments for some household objects and Figure 4.3 depicts some sample grasps executed. As it can be observed, the planning times are in the order of seconds (these times include the grasp generation + reach path planning). The majority of the total time is taken by the execution per se, which takes nearly 20 seconds. For initial testing, we limited both the velocity and acceleration of the arm to 50% and 25% of their nominal values.

Table 4.1

Object	Segmentation	Fitting	Plan	Execution	Total	Success Rate
1. Fruit Box	1.7s	2.4s	1.7s	18.5s	38.3s	88.8 %
2. Milk Tetrapak	1.7s	2.0s	2.3s	19s	36.5s	88.8 %
3. Raisins container	1.6s	1.6s	1.7s	18.7s	35.3s	85.1 %
4. Mustard bottle	1.7s	2.3s	4.2s	19.6s	39.2s	92.5 %
5. Coffee jar	1.6s	2.3s	3.5s	18.9s	37.6s	85.1 %



Figure 4.3: Sample executions of reach + grasp execution using our superquadric representation

5. Task-Based Grasp Prioritization

In Chapter 4, a simple method to generate a set of candidate grasps \mathcal{G} was presented. Although all grasps $g_i \in \mathcal{G}$ are kinematically feasible, this does not guarantee that a reaching path (moving the arm from its start configuration to actually grasping the object) exists. In order to find a solution using a brute-force approach, a grasp g_i would be chosen randomly and a path planning procedure would be performed. If a path exists, then it is executed. Otherwise, this grasp is discarded and another g_i is selected, repeating the process until a path is found.

In general, humans (and robots) grasp objects for reasons deeper than just for the sake of picking them up. A typical example is pick-and-place, a two-step task in which the goal is to transport an object \mathcal{O} from its initial location in wT_s to a final pose wT_g while avoiding obstacles during execution. As in the picking-only case, a set of candidate grasps \mathcal{G} is generated such that each g_i is kinematically feasible to execute in both wT_s and wT_g . A brute-force approach in this instance would imply that, until a solution is found, two paths should be generated per each candidate g_i : A *reach path* - from the arm start configuration to reach \mathcal{O} in wT_s - and a *transport path* to move \mathcal{O} from wT_s to wT_g .

In this section, our goal is to prioritize the grasps g_i such that grasps that are more likely to render feasible paths are tried first. Ideally, only one grasp should be tried and the corresponding paths should also be simple to plan (not requiring awkward arm movements to reach or set the object). Our problem can then be defined as finding a metric to compare the grasps $g_i \in \mathcal{G}$ such that the desired conditions above are met.

End-Comfort Effect

Humans perform pick-and-place tasks effortlessly on a daily basis and in a near automatic way. Numerous psychological studies show that humans choose what grasp to use depending on the *goal to be accomplished*. Rosenbaum et al. [66] carried out several experiments with humans which mainly consisted on pick-and-place tasks with varying start and goal poses for a tested object. Rosenbaum observed that the subjects strongly tended to choose the grasp that, when executed, allowed the arm to finish in a comfortable configuration, with the arm joint values far from their limits. Rosenbaum termed this human preference the *end-comfort effect*. A common example of this effect occurs when a pick-and-place task consists on picking up an upside-down glass from a table to set it on an upright pose. The grasp universally chosen demands the hand to grasp the object with a rotated wrist and the elbow awkwardly up. However, this is justified - according to the end-comfort effect - since once the glass is set, the arm finishes in a relaxed position (elbow down). Inspired by these observations, we devised a simple metric to rate grasps based on the end-comfort effect.

5.1 A metric to prioritize grasps

Manipulability [77] is a metric that measures how dexterous the end-effector of a robotic arm is at a given joint configuration \mathbf{q} . The farther the joint values are from their limits, the higher the $m(\mathbf{q})$ value is for that specific configuration.

$$m(\mathbf{q}) = \sqrt{|J(\mathbf{q})J^T(\mathbf{q})|}. \quad (5.1)$$

Manipulability is typically defined for a single joint configuration. Remember that in our case, we want to define a metric for a grasp g_i , not for an arm configuration. Furthermore, each g_i does not have an unique corresponding \mathbf{q} . Rather, many possible \mathbf{q} can be associated to a single g_i due to arm redundancy (our robot platform has 7-DOF arms).

Given the scenario described above, we introduce the concept of *situated grasp manipulability*(m_g).

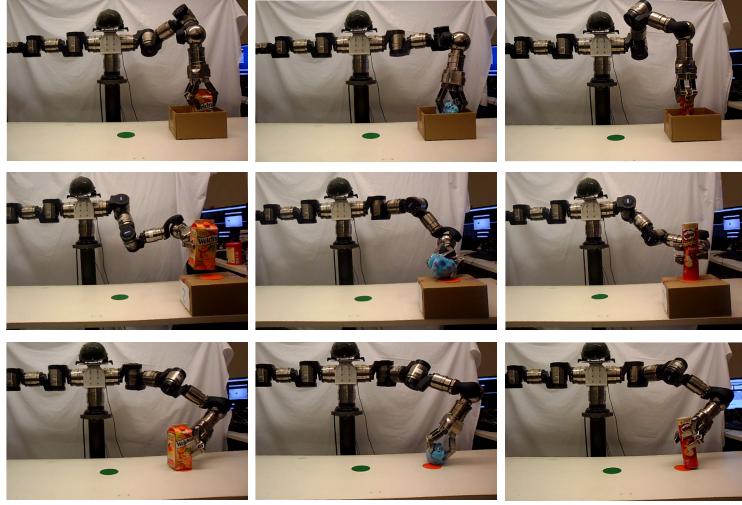


Figure 5.2: Final grasp configuration during manipulation with goal constraints (top and middle) and without goal constraints (bottom).

5.1.1 Situated grasp manipulability(m_g)

Given a target object \mathcal{O} located at pose wT_o , and its corresponding grasp \mathbf{g}_i , we define m_g as the average manipulability of a uniform set of collision-free arm configurations \mathbf{q}_i that allow executing \mathbf{g}_i :

$$m_g = \frac{1}{N} \sum_{i=1}^N m(\mathbf{q}_i) \quad (5.2)$$

Please note that m_g depends on both \mathbf{g}_i , wT_o and the environment (for collisions) since only collision-free grasps that reach the object are considered. Figure 5.1 shows an example of a pick-and-place task where the green and red markers indicate wT_s and wT_g respectively. In this case, m_g at wT_g is bigger than wT_s (where $N_s = 76$ and $N_g = 108$ are the number of IK solutions for both situations). When the object is at wT_g , in average the arm configuration is more relaxed.

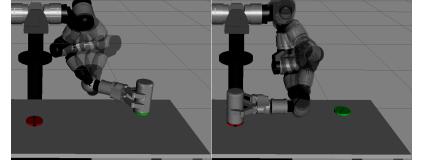


Figure 5.1: Examples of m_g measured at wT_s and wT_g

From the shown example, it becomes evident that for a pick-and-place manipulation problem, there are at least two possible metrics to use per grasp \mathbf{g}_i : m_g measured either at wT_s or wT_g . Choosing the first option means that we prioritize grasps in which the pick phase is executed comfortably (wT_s), whereas by choosing the latter, we favor grasps in which the arm configuration used at placing the object (wT_g) is more relaxed.

In order to compare these candidate metrics, we performed a set of experiments in simulation where we compared the performance of 3 metrics: 2 of them were the ones we discussed above (m_g evaluated at wT_g and wT_s) and the third one being the average of these 2. We generated pick-and-place test cases by randomly generating start poses wT_s and corresponding wT_g obtained by applying a translation and/or a rotation w.r.t. the table normal. Since the main goal of prioritizing grasps is to obtain a solution in the first trial, we compare the performance of these 3 metrics by counting the number of times the grasp with the highest metric value produced a solution. The results are shown in Table 5.1. It can be seen that using m_g evaluated at wT_g produced the largest number of one-shot solutions.

Metric	Planning Time	Success
m_g at wT_s	2.17	21.8/35
m_g at wT_g	2.218	33/35
Avg. m_g	2.29	31.4/35
Metric	Planning Time	Success
m_g at wT_s	4.70	255/278
m_g at wT_g	4.49	275/278
Avg. m_g	3.83	260/278

Table 5.1: Evaluation without/with rotation change

We applied the metric m_g evaluated at wT_g to select automatically the grasp to be used for pick-and-place problems such as the shown in Figure 5.2. In these experiments, 3 different objects had to be transported from an initial pose (green marker) to a final pose (orange marker) for 3 scenarios: First, the goal location was inside a box. Second, the object had to be put on top of a box. Finally, the object had to just be translated on the table. It can be seen that different grasps were selected to perform the different tasks.

6. Handover Interaction

In Chapter 5, we went over a strategy to prioritize grasps based on the goal pose to achieve. Up until now, we have considered only single-arm problems, in which interaction only happens between the object and the robot hand. While many tasks can be accomplished this way, there exists scenarios where the use of more than one limb might be needed:

- *Self-Handover*: Handing over an object from one hand to another. Having the ability to use both arms increase the effective workspace of the robot, allowing it to perform pick-and-place tasks which could not be achieved otherwise (i.e. picking up an object only reachable with the left hand and placing it on a shelf only reachable with the right hand).
- *Multi-agent Handover*: A household robot does not work on isolation so tasks involving handing over objects to another entity (most likely a human) are among the most likely to arise on a real environment.

We have completed work studying planning strategies for both types of handover. A brief description of each of them follows:

6.1 Self-Handover

Our approach, presented in [42], studies the problem of finding a handover pose for a pick-and-place bimanual task, such as the one shown in Figure 6.1. The problem is defined by an object located in a starting pose wT_s , which has to be moved to wT_g , with each pose only available in either the left or right arm workspace, making a handover step necessary.

Our method exploits the knowledge of wT_s , wT_g and the reachability space[47] of the bimanual robot to select a handover pose for the object wT_H such that it facilitates the final placing of the object in wT_g . The reachability space is a voxelized representation of the workspace of a robot, where each voxel contains information of the average manipulability of each arm when the end-effector coincides with the voxel centroid. The voxels can be either 3D or 6D and their information is generated offline and stored for online use.

The handover pose wT_H is calculated by considering the area on the reachability space in which both arms have an average manipulability index above a certain minimum value. From this area, we select the voxel which is closest to the line joining the 3D positions of wT_s and wT_g and use its centroid as the translation part of wT_H . The rotation is calculated by interpolating the rotation matrices of wT_s and wT_g . Once wT_H is defined, the self-handover problem reduces to 2 pick-and-place planning problems, for which we also apply the grasp prioritization scheme presented on Chapter 5.

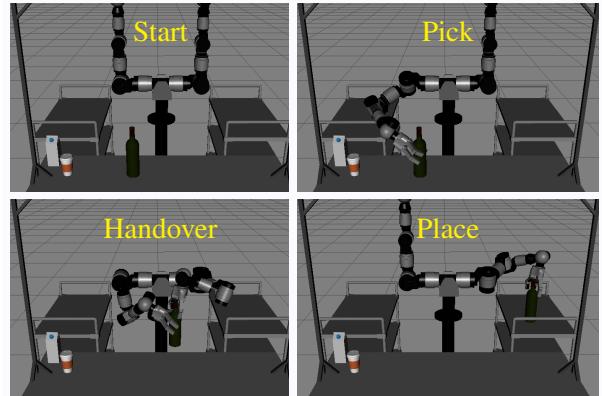


Figure 6.1: Bimanual manipulation task: Transporting a bottle from the table to a utility cart at the right of the robot

6.2 Multi-Agent Handover

The approach delineated above was further extended to be applied to handover transfers between 2 different agents [39], which we call the giver and receiver. The core idea of using the reachability space to find handover poses that maximize the manipulability of the limbs involved was the same. The key difference with respect to the self-handover approach is that in this work we favor handover poses that are likely to maximize the manipulability of the receiver, which will most likely be a human, hence its comfort is a higher priority than the manipulability of the robot giver. This is achieved by favoring rotations of wT_h such that the receiver's hand orientation is aligned with the line joining the shoulders of the receiver and the giver agent.

Experiments in simulation were performed to evaluate this approach. Two agents were considered: A simulated Hubo robot was set as the receiver and the LWA4 robot was the giver. We used a Hubo model instead of a human since its Inverse Kinematics closed form solution were available, making it easier to evaluate. 3 test scenarios were created, similar to the ones depicted in Figure 6.2: (1) Butler scenario: The giver has to pass an object to the receiver, which in turn must place the object on its holding tray. (2) Kneeling scenario: The giver must hand off an object to the receiver, which then must translate the object (a cylindrical container) and rotate it such that its final pose is above the plate on the floor and pointing downwards. (3) Stool scenario: The receiver must hand off a hammer to the receiver, which is standing on a 3-step stool. Notice that in most of these scenarios, the giver and the receiver are not constrained to face each other. In fact, most of the time, they are not. In unconstrained scenarios like these, our heuristic of using the line joining the shoulders between giver and receiver to guide the wT_H plays an important role to generate feasible solutions.

We performed 100 randomized experiments per each scenario, where the randomized variables were the receiver initial pose as well as the object start and final poses wT_s and wT_g . In Table 6.1 we see some statistics of these 100 tests. A run is considered successful if at least 1 valid handover plan is found.

Table 6.1: Results for 100 randomized tests for butler, stool and kneeling scenarios

Metric	Butler	Stool	Kneeling
1. Avg. planning time	2.31s	4.62s	3.18s
2. Successful plans	100/100	100/100	100/100
3. Avg. number of grasps for giver	32	56	30
4. Avg. number of grasps for receiver	39	143	72
5. Avg. number of handover poses	16	89	39

7. Task Evaluation: Benchmarks and Metrics

In the previous chapters we have discussed manipulation planning for a small set of tasks: Pick-up, pick-and-place and handover actions. Ideally, our results could be more complete if we could compare our performance with other existing approaches. In reality, this is a hard goal to achieve, since results obtained by different authors consider different assumptions, operate on different manipulated objects and use different metrics, many of which evaluate only a submodule of a robotic system (i.e. perception, planning) rather than the integral performance of the robot with respect to the task at hand.

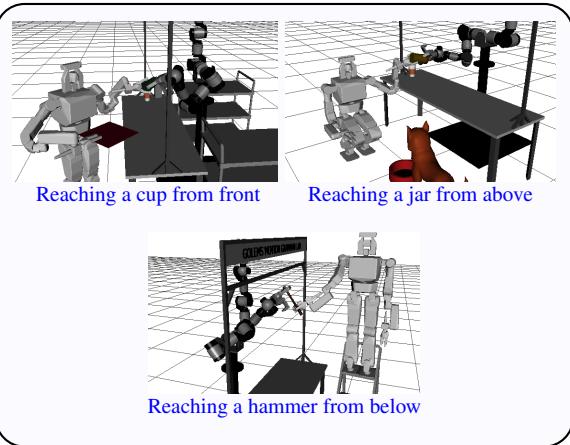


Figure 6.2: Shared manipulation tasks with non face-to-face interaction between agents

As it was described in 2.2, multiple attempts at benchmarking and robot performance comparison have been made, mostly in the form of competitions. However, most of these instances have still ignored one of the main goal of benchmarking which is to evaluate performance with respect to meaningful tasks. What tasks are considered *meaningful* depends on the end-user of the robotic system. As we explained in our thesis statement, our goal is to develop a framework for household robots, which means that our solutions are expected to be used by common people. The definition of benchmark tasks then should be supported by the needs of real people.

Fields such as ergonomics, occupational therapy and medicine have devoted a great deal of research efforts to design tests-commonly known as *dexterity tests*-to measure the manipulation capabilities of human subjects, usually to keep track of the recovery of dexterity abilities of people who have suffered some health impairment, such as a stroke. In [38], we presented a taxonomy of benchmark tasks, which is based on a review of dexterity tests applied on human subjects over the last 50 years. The following aspects were considered for the design of this benchmark:

- **Hardware-agnostic:** Tests should not be designed with a specific platform in mind. In order for benchmarks to be widely accepted, they should be realizable under minimum hardware capabilities assumptions. This is not different from human dexterity tests, in which people with different physical and mental conditions (or even impairments) are equally evaluated.
- **Flexibility:** A lesson learned from the review of human tests is that there is not an unique “right way” to solve a task. The robot should be allowed to apply a strategy that better suits its particular situation (i.e. a robot equipped with a gripper might have to adopt a different approach to grasping an object than a robot with a 3-fingered hand).
- **Time efficiency:** Benchmarking is not a goal by itself. Rather, it should be seen as a diagnostic tool to be applied regularly, to make sure our systems are comparable (or within reasonable distance) to the state-of-the-art. Accordingly, since benchmarks are a sidestep tool, they should be selected such that they can be evaluated in a rapid manner, without the need of a complex setup or highly constrained rules.
- **Relevant metrics:** A metric is only useful if it can be compared against a standard value. Tasks should be selected such that the evaluation can be objective, numerically expressed and important for both the researcher and, eventually, for the end-user. From the numerous studies surveyed, we believe that (1) Task completion time and (2) Success rate are the two objective, informative metrics that can more easily be used.
- **Statistically relevant:** Evaluation should consider a minimum number of attempts to be considered valid. This has already been seen in the ARM project evaluations, in which 5 trials were performed and the average results were evaluated.
- **Realistic assumptions:** While a Laboratory environment is not the same as a real-home space, we should stress the importance of avoiding to rely on assumptions that in no way will exist in the real world. For instance, assumption of markers placed in objects or off-board visual sensors are very unlikely to occur, so their use might not translate into a realistic evaluation of capabilities.
- **User-focused tests:** Service robots will be deployed at human homes, so it is reasonable to take into account feedback from the end-users to evaluate our systems. Studies in assistive technologies have shown that the main objective performance measures used by humans with assistive robotic arms are: Their capacity to perform activities of daily living and time to task completion[74]

Figure 7.1 shows the diagram of the benchmark proposed. Sample tasks per each sublevel of the benchmark taxonomy are shown in Table 7.1, Table 7.2 and Table 7.3.

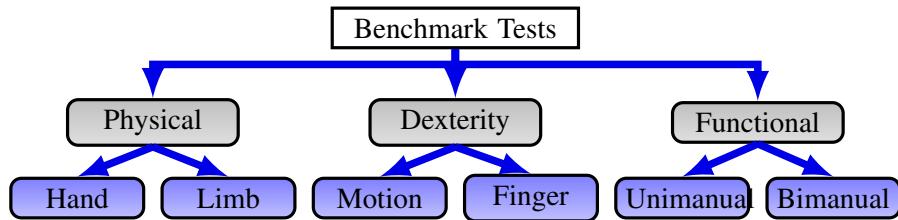


Figure 7.1: Proposed benchmark taxonomy

Table 7.1: Sample Physical Tasks

Sublevel	Examples	Sublevel	Examples
Hand	<ol style="list-style-type: none"> 1. Maximum finger aperture 2. Maximum payload when picking up a high-friction object 3. Perform static grasps from existing taxonomies. 4. Perform static grasps of benchmark objects (i.e. YCB dataset) 	Limb	<ol style="list-style-type: none"> 1. Position the palm on a table surface 2. Open a screw cap 3. Point at diverse objects on a table 4. Sequentially point at a moving target

Table 7.2: Sample Dexterity Tests

Sublevel	Examples	Sublevel	Examples
Manual	<ol style="list-style-type: none"> 1. Pick-up an object on a clear table 2. Pick an object from a table and place it on a cupboard 3. Pick an object from a box and set it at an adjacent box 	Finger	<ol style="list-style-type: none"> 1. Unscrew a bottle using fingers only 2. Rotate a chopstick 3. Grab a short cylinder from the table and rotate it such that its supporting face ends up facing upwards.

Table 7.3: Sample Functional Tests

Sublevel	Examples	Sublevel	Examples
Unimanual	<ol style="list-style-type: none"> Open a door (handle/knob) Plug in a power plug Press level on an electric kettle Pick up a glass from a full dish rack Push an emergency button Pour a liquid in a wide container Stir slowly in a pot Spray from a bottle Stack cans Spoon beans (simulated feeding) 	Bimanual	<ol style="list-style-type: none"> Cut a piece of Play-Doh on a table Rotate a steering wheel 45 degrees Empty a trash can (turn it upside down) Pick up a tray with a glass on it and transport it Open a jar with screw lid Grab an open tetrapak and pour liquid in a cup for 2s.

8. Summary and Proposed Work

8.1 Summary

In this document we have presented a framework to perform elementary manipulation tasks using superquadrics as the object representation of choice. We have shown work on the pipeline blocks of (1) perception and fitting, (2) grasp generation, (3) grasp selection according to a proposed human-heuristic task metric and (4) Motion planning for handover tasks. We also have introduced a taxonomy of benchmark tasks to fully define the domain of action of our framework.

In the next final section we propose our next steps and the expected timeline of results.

8.2 Proposed Work

Extension of SQ Deformations

Chapter 3 described our use of general superquadrics to represent objects. Additionally, there exists 2 type of deformations (tampering and bending) which increase the shapes that can be expressed at the cost of adding 4 additional parameters (2 per tampering, 2 per bending). We have devised a new, simple bending deformation with only 1 DOF, as well as implemented the tampering deformation considering only 1 parameter. A writeup describing the proposed bending deformation and the evaluation of these 2 deformations on real data(and their impact on the minimization processing time) is proposed.

Extension of Grasp Prioritization to multiple soft constraints

The work presented on Chapter 5 involve only pose constraints, which are a classical example of hard constraints. We are interested on modifying our approach to also be usable for soft constraints, which makes the grasp prioritization harder as the space of possible grasps increases. An example of this type of constraint is the pick-and-place task in which the goal pose of the object is not defined with other constraints than to keep the object upright (translation + a 1-DOF rotation).

Another aspect we would like to investigate (which was actually feedback from reviewers of this work) is the consideration of manipulation plans with more than 2 steps(i.e. pick and place). We plan to improve our approach to deal with 3 constraints, for problems such as *picking up a bottle, pouring the contents in a bowl and then putting the bottle on a cupboard*.

Since both kinds of problems have an added level of complexity, we intend to incorporate human heuristics to guide the grasp prioritization. We intend to retrieve human data from manipulation experiments in which human subjects perform manipulation tasks such as the ones described above while tracking data from the object and their hands is recorded. We have already performed a few tests using a Polhemus Liberty Motion tracker system and have encountered some interesting patterns that hopefully could lead to a solution to the proposed problems.

Implementation of sample Benchmark experiments

In Chapter 7 we outlined a theoretical benchmark taxonomy, but no extensive results were published yet. We propose to perform at least 1 sample test per each benchmark sublevel. Most of these tests have already been performed before, but not in a strict, extensive manner. We propose to use the newly introduced YCB dataset in order to make our results more easily used for comparison reasons.

Implementation and improvement of handover planning algorithms

The experiments presented on Chapter 6 were done in simulation mainly because the robotic platform was not fully available at the time of writing. We would like to validate our results with real experiments including (1) the robot passing an object from one hand to another, for a small subset of household objects (from the YCB dataset) and (2) a handover demonstration between the robot and a human subject.

Diverse, End-Effector-based Path Planning

This is optional. In very early work [41] and [40], we proposed two coupled approaches to deterministically generate arm paths based on end-effectors trajectories. If time allows, we would like to incorporate these 2 approaches into our pipeline (as of now, our path planning is obtained, as we explained in Section 4.3, using a randomized approach, which - while working reasonably well most of the times - is nonetheless non-deterministic. Deterministic approaches for motion planning are particularly relevant for household robots, because they operate around humans, hence, an expected, deterministic behavior of the robot arm movement is desirable.

8.3 Schedule of Work

- **August 2015:** Proposal
- **September 2015:** Extension of SQ deformations
- **October-November 2015:** Extension of Grasp prioritization
- **December 2015-January 2016:** Sample benchmark experiments
- **February 2016:** Handover implementation
- **March 2016:** Dissertation writeup
- **April 2016:** Defense

Bibliography

- [1] J. Aleotti, V. Micelli, and S. Caselli. “Comfortable robot to human object hand-over”. In: *IEEE International Symposium on Robot and Human interactive Communication (ROMAN)*. 2012, pp. 771–776.
- [2] J. Aleotti, D.L. Rizzini, and S. Caselli. “Perception and Grasping of Object Parts from Active Robot Exploration”. In: *Journal of Intelligent & Robotic Systems* 76.3-4 (2014), pp. 401–425.
- [3] R. Balasubramanian et al. “Human-guided grasp measures improve grasp robustness on physical robot”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2010, pp. 2294–2301.
- [4] R. Balasubramanian et al. “Physical human interactive guidance: Identifying grasping principles from human-planned grasps”. In: *The Human Hand as an Inspiration for Robot Hand Development*. Springer, 2014.
- [5] A. Barr. “Superquadrics and angle-preserving transformations”. In: *IEEE Computer graphics and Applications* 1.1 (1981), pp. 11–23.
- [6] D. Berenson, S. Srinivasa, and J. Kuffner. “Task Space Regions: A Framework for Pose-Constrained Manipulation Planning”. In: *The International Journal of Robotics Research* 30.12 (2011), pp. 1435–1460.
- [7] D. Berenson et al. “Grasp planning in complex scenes”. In: *IEEE-RAS International Conference on Humanoid Robots (ICHR)*. 2007, pp. 42–48.
- [8] D. Berenson et al. “Manipulation planning with workspace goal regions”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2009, pp. 618–624.
- [9] A. Bicchi and V. Kumar. “Robotic grasping and contact: A review”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2000, pp. 348–353.
- [10] J. Bohg et al. “Mind the gap: Robotic grasping under incomplete observation”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 686–693.
- [11] J. Bohg et al. “Data-Driven Grasp Synthesis: A Survey”. In: *IEEE Transactions on Robotics* 30 (2 2013), pp. 289–309.
- [12] R. Bohlin and L.E. Kavraki. “Path planning using lazy PRM”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2000, pp. 521–528.
- [13] M. Cakmak et al. “Human preferences for robot-human hand-over configurations”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2011, pp. 1986–1993.
- [14] B. Calli et al. “Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols”. In: *IEEE Robotics and Automation Magazine (in press)* (2015).
- [15] L. Chevalier, F. Jaillet, and A. Baskurt. “Segmentation and superquadric modeling of 3D objects”. In: *Journal of WSCG* (2003).
- [16] Y.S. Choi et al. “A list of household objects for robotic retrieval prioritized by people with ALS”. In: *IEEE International Conference on Rehabilitation Robotics (ICORR)*. 2009, pp. 510–517.
- [17] M. Ciocarlie, C. Goldfeder, and P.K. Allen. “Dimensionality reduction for hand-independent dexterous robotic grasping”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2007, pp. 3270–3275.
- [18] M. Ciocarlie et al. “Towards reliable grasping and manipulation in household environments”. In: *Experimental Robotics*. Springer. 2014, pp. 241–252.

- [19] K. Collins, A.J. Palmer, and K. Rathmill. “The development of a European benchmark for the comparison of assembly robot programming systems”. In: *Robot technology and applications*. Springer, 1985, pp. 187–199.
- [20] US Department of Commerce. *NISTIR 7940: Dexterous Manipulation for Manufacturing Applications Workshop*. 2014. URL: www.nist.gov/el/isd/upload/NIST-IR-7940.pdf.
- [21] M. R. Cutkosky. “On grasp choice, grasp models, and the design of hands for manufacturing tasks”. In: *IEEE Transactions on Robotics and Automation* 5.3 (1989), pp. 269–279.
- [22] B. Dang-Vu, M. Roa, and C. Borst. “Extended independent contact regions for grasping applications”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013, pp. 3527–3534.
- [23] N. Dantam and M. Stilman. “Robust and efficient communication for real-time multi-process robot software”. In: *IEEE-RAS International Conference on Humanoid Robots (ICHR)*. 2012, pp. 316–322.
- [24] *DART: Dynamic Animation and Robotic Toolkit*. 2013. URL: <https://github.com/dartsim/dart>.
- [25] R. Diankov and J. Kuffner. *OpenRAVE: A Planning Architecture for Autonomous Robotics*. Tech. rep. CMU-RI-TR-08-34. Pittsburgh, PA: Robotics Institute, 2008.
- [26] K. Duncan et al. “Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects”. In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*. 2013, pp. 4238–4243.
- [27] S. El-Khoury, R. De Souza, and A. Billard. “On computing task-oriented grasps”. In: *Robotics and Autonomous Systems* 66 (2015), pp. 145–158.
- [28] S. El-Khoury, M. Li, and A. Billard. “On the generation of a variety of grasps”. In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1335–1349.
- [29] T. Feix, I.M. Bullock, and A.M. Dollar. “Analysis of human grasping behavior: Object characteristics and grasp type”. In: *IEEE Transactions on Haptics* 7.3 (2014), pp. 311–323.
- [30] C. Ferrari and J. Canny. “Planning optimal grasps”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 1992, pp. 2290–2295.
- [31] J. Fontanals et al. “Integrated Grasp and Motion Planning using Independent Contact Regions”. In: *IEEE-RAS International Conference on Humanoid Robots (ICHR)*. 2014, pp. 887–893.
- [32] C. Goldfeder et al. “Grasp planning via decomposition trees”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2007, pp. 4679–4684.
- [33] C. Goldfeder et al. “The Columbia grasp database”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2009, pp. 1710–1716.
- [34] A.D. Gross and T.E. Boult. “Error of fit measures for recovering parametric solids”. In: *IEEE International Conference on Computer Vision*. 1988, pp. 690–694.
- [35] G. Grunwald, C. Borst, and J. et al. Zöllner. “Benchmarking dexterous dual-arm/hand robotic manipulation”. In: *Workshop on Performance Evaluation and Benchmarking at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008.
- [36] A. Gupta, L. Bogoni, and R. Bajcsy. “Quantitative and qualitative measures for the evaluation of the superquadric models”. In: *Workshop on Interpretation of 3D Scenes*. IEEE. 1989, pp. 162–169.
- [37] D. Hackett et al. “An overview of the DARPA Autonomous Robotic Manipulation (ARM) program”. In: *Journal of the Robotics Society of Japan* 31.4 (2013), pp. 326–329.
- [38] A. Huamán Quispe, H. Ben Amor, and H.I. Christensen. “A Taxonomy of Benchmark Tasks for Bimanual Manipulators”. In: *International Symposium of Robotics Research (to appear)* (2015).

- [39] A. Huamán Quispe, H. Ben Amor, and M. Stilman. “Handover Planning for Every Occasion”. In: *IEEE-RAS International Conference on Humanoid Robots (ICHR)*. 2014, pp. 431–436.
- [40] A. Huamán Quispe, T. Kunz, and M. Stilman. “Generation of Diverse Paths in 3D Environments”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013, pp. 5994–5999.
- [41] A. Huamán Quispe and M. Stilman. “Deterministic Motion Planning for Redundant Robots along End-Effector Paths”. In: *IEEE International Conference on Humanoids Robots (ICHR)*. 2012, pp. 785–790.
- [42] A. Huamán Quispe et al. “It takes two hands to grasp: Towards Handovers in Bimanual Manipulation Planning”. In: *Workshop on Human vs Robot Grasping and Manipulation at Robotics, Science and Systems (RSS)*. 2014.
- [43] A. Huamán Quispe et al. “Exploiting Symmetries and Extrusions for Grasping Household Objects”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 3702–3708.
- [44] K. Huebner and D. Kragic. “Selection of robot pre-grasps using box-based shape approximation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008, pp. 1765–1770.
- [45] K. Huebner et al. “Grasping known objects with humanoid robots: A box-based approach”. In: *International Conference on Advanced Robotics (ICAR)*. 2009, pp. 1–6.
- [46] I. Iossifidis et al. “Towards Benchmarking of Domestic Robotic Assistants”. In: *Advances in Human-Robot Interaction*. Vol. 14. Springer, 2005, pp. 403–414.
- [47] D. Kee and W. Karwowski. “Analytically derived three-dimensional reach volumes based on multijoint movements”. In: *Human factors: The Journal of the Human Factors and Ergonomics Society* 44.4 (2002), pp. 530–544.
- [48] B. Kehoe et al. “Cloud-based robot grasping with the Google Object Recognition Engine”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2013, pp. 4263–4270.
- [49] J. Kim et al. “Physically based grasp quality evaluation under pose uncertainty”. In: *IEEE Transactions on Robotics* 29.6 (2013), pp. 1424–1439.
- [50] J.J. Kuffner and S.M. LaValle. “RRT-Connect: An efficient approach to single-query path planning”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2000, pp. 995–1001.
- [51] Y. Li, J.L. Fu, and N.S. Pollard. “Data-driven grasp synthesis using shape matching and task-based pruning”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.4 (2007), pp. 732–747.
- [52] Z. Li and S.S. Sastry. “Task-oriented optimal grasping by multifingered robot hands”. In: *IEEE Journal of Robotics and Automation* (1988), pp. 32–44.
- [53] Y. Liu, M. Lam, and D. Ding. “A complete and efficient algorithm for searching 3D form-closure grasps in the discrete domain”. In: *IEEE Transactions on Robotics* 20.5 (2004), pp. 805–816.
- [54] M. Lourakis. *levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++*. 2004. URL: <http://www.ics.forth.gr/~lourakis/levmar>.
- [55] R. Madhavan, R. Lakaemper, and T. Kalmár-Nagy. “Benchmarking and standardization of intelligent robotic systems”. In: *International Conference on Advanced Robotics (ICAR)*. 2009, pp. 1–7.
- [56] J. Mainprice et al. “Sharing effort in planning human-robot handover tasks”. In: *IEEE International Symposium on Robot and Human Interaction (ROMAN)*. 2012, pp. 764–770.

- [57] K. Matheus and A. Dollar. “Benchmarking grasping and manipulation: Properties of the objects of daily living”. In: *IEEE/RSJ International Conference on Intelligent Systems (IROS)*. 2010, pp. 5020–5027.
- [58] A.T. Miller et al. “Automatic grasp planning using shape primitives”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2003, pp. 1824–1829.
- [59] B. Mirtich and J. Canny. “Easily computable optimum grasps in 2D and 3D”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 1994, pp. 739–747.
- [60] J. Moré. “The Levenberg-Marquardt algorithm: Implementation and theory”. In: *Numerical analysis*. Springer, 1978, pp. 105–116.
- [61] A.K. Pandey et al. “Towards planning human-robot interactive manipulation tasks: Task dependent and human oriented autonomous selection of grasp and placement”. In: *IEEE/RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 2012, pp. 1371–1376.
- [62] M. Pilu and R.B. Fisher. *Equal-distance sampling of superellipse models*. University of Edinburgh, Department of Artificial Intelligence, 1995.
- [63] G. Pratt and J. Manzo. “The DARPA Robotics Challenge”. In: *IEEE Robotics & Automation Magazine* 20.2 (2013), pp. 10–12.
- [64] M. Przybylski, T. Asfour, and R. Dillmann. “Planning grasps for robotic hands using a novel object representation based on the medial axis transform”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2011, pp. 1781–1788.
- [65] M. Przybylski et al. “Human-Inspired Selection of Grasp Hypotheses for Execution on a Humanoid Robot”. In: *IEEE/RAS International Conference on Humanoid Robots (ICHR)*. 2011, pp. 643–649.
- [66] D.A. Rosenbaum, C.M. van Heugten, and G.E. Caldwell. “From cognition to biomechanics and back: The end-state comfort effect and the middle-is-faster effect”. In: *Acta psychologica* 94.1 (1996), pp. 59–85.
- [67] R.B. Rusu. “Semantic 3D object maps for everyday manipulation in human living environments”. In: *KI-Künstliche Intelligenz* 24.4 (2010), pp. 345–348.
- [68] R.B. Rusu and S. Cousins. “3D is here: Point Cloud Library (PCL)”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 1–4.
- [69] A. Sahbani, S. El-Khoury, and P. Bidaud. “An overview of 3D object grasp synthesis algorithms”. In: *Robotics and Autonomous Systems* 60.3 (2012), pp. 326–336.
- [70] A. Saxena, J. Driemeyer, and A.Y. Ng. “Robotic grasping of novel objects using vision”. In: *The International Journal of Robotics Research* 27.2 (2008), pp. 157–173.
- [71] E.A. Sisbot and R. Alami. “A human-aware manipulation planner”. In: *IEEE Transactions on Robotics* 28.5 (2012), pp. 1045–1057.
- [72] F. Solina and R. Bajcsy. “Recovery of parametric models from range images: The case for superquadrics with global deformations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.2 (1990), pp. 131–147.
- [73] *The Amazon Picking Challenge*. 2014. URL: <http://amazonpickingchallenge.org/>.
- [74] K. Tsui et al. “Performance evaluation methods for assistive robotic technology”. In: *Performance Evaluation and Benchmarking of Intelligent Systems*. Springer, 2009, pp. 41–66.
- [75] N. Vahrenkamp, T. Asfour, and R. Dillmann. “Simultaneous grasp and motion planning: Humanoid robot ARMAR-III”. In: *IEEE Robotics & Automation Magazine* (2012), pp. 43–57.
- [76] T. Wisspeintner et al. “RoboCup@Home: Scientific competition and benchmarking for domestic service robots”. In: *Interaction Studies* 10.3 (2009), pp. 392–426.

- [77] T. Yoshikawa. “Manipulability of robotic mechanisms”. In: *The International Journal of Robotics Research* 4.2 (1985), pp. 3–9.
- [78] Y. Zhang. “Experimental comparison of superquadric fitting objective functions”. In: *Pattern Recognition Letters* 24.14 (2003), pp. 2185–2193.