To my parents: He, Yan and Hang, Qinghua
爸妈，我爱你们
Abstract

Many robot object interactions require that an object is firmly held, and that the grasp remains stable during the whole manipulation process. One of the most mature approaches to formalizing a grasp is by describing it through its contact forces and positions. Based on this, this thesis address the problems of measuring the grasp sensitivity against friction changes, planning contacts and hand configurations on mesh and point cloud representations of arbitrary objects, planning adaptable grasps and finger gaiting for keeping a grasp stable under various external disturbances, as well as learning of grasping manifolds for more accurate reachability and inverse kinematics computation for multi-fingered grasping.

Firstly, we propose a new concept called friction sensitivity, which measures how susceptible a specific grasp is to changes in the underlying friction coefficients. We develop algorithms for the synthesis of stable grasps with low friction sensitivity and for the synthesis of stable grasps in the case of small friction coefficients.

Secondly, for fast planning of contacts and hand configurations for dexterous grasping, as well as keeping the stability of a grasp during execution, we present a unified framework for grasp planning and in-hand grasp adaptation using visual, tactile and proprioceptive feedback. The main objective of the proposed framework is to enable fingertip grasping by addressing problems of changed weight of the object, slippage and external disturbances. For this purpose, we introduce the Hierarchical Fingertip Space (HFTS) as a representation enabling optimization for both efficient grasp synthesis and online finger gaiting. Grasp synthesis is followed by a grasp adaptation step that consists of both grasp force adaptation through impedance control and regrasping/finger gaiting when the former is not sufficient.

Lastly, to improve the efficiency and accuracy of dexterous grasping and in-hand manipulation, we present a system for fingertip grasp planning that incrementally learns a heuristic for hand reachability and multi-fingered inverse kinematics. The system consists of an online execution module and an offline optimization module. During execution the system plans and executes fingertip grasps using Canny’s grasp quality metric and a learned random forest based hand reachability heuristic. In the offline module, this heuristic is improved based on a grasping manifold that is incrementally learned from the experiences collected during execution.
Sammanfattning


Först introducerar vi ett nytt koncept som vi kallar friction sensitivity - friktionskänslighet, vilket mäter hur känsligt ett specifikt grepp är mot förändringar i friktionskoefficienterna. Vi presenterar algoritmer som beräknar stabila grepp med låg friktionskänslighet och som fungerar även då friktionen är låg.

För att snabbt planera kontaktpunkter samt handkonfigurationer för så kallade dexterous grasps - fingerfärdiga grepp, såväl som att bibehålla ett grepps stabilitet under rörelse, presenterar vi ett enhetligt system för planering av grepp samt anpassning av greppet medan objektet vilar i handen. Detta sker genom användning av visuell, taktill samt proprioceptiv återkoppling. Vårt främsta mål med systemet är att möjliggöra grepp med fingertopp genom att adressera problem med objekt som ändrar vikt, friktion samt externa störningar. Till detta ändamål introducerar vi Hierarchical Fingertip Space - Hierarkisk Fingertoppsrymd, en representation som möjliggör effektiv optimering av grepp samt så kallad finger gaiting, allt i realtid. Beräkning av grepp följs av ett steg som anpassar greppet både till de krafter som verkar på fingertopparna samt impedanskontroll och återgreppning och finger gaiting när det föregående inte är nog.

Slutligen, för att förbättra effektivitet och och precision vid dexterous grasping och manipulering av objektet när det vilar i handen presenterar vi ett system för fingertoppsgrepp som inkrementellt lär sig en heuristik för handens räckvidd och invers kinematik för grepp med flera fingrar. Systemet består av en modul som optimerar grepp i förhand samt en modul som exekverar greppet. När roboten genomför en uppgift planerar och utför systemet fingertoppsgrepp med Canny's greppkvalitetsmetrik (grasp quality metric) och lär sig en random forest-baserad räckviddsheuristik. I optimizeringsmodulen förbättras denna heuristik baserat på en grasping manifold - en modell som inkrementellt byggs upp från de erfarenheter som roboten samlat på sig.
Acknowledgments

This four years journey was an amazing experience, especially when it was hard. I am grateful for so many people who have been helping and encouraging me making it through. First and foremost, I would like to thank Dani for:

- Offering me the position to be your student.
- Sharing brilliant and inspiring ideas to guide me on research.
- Unconditionally supporting me on my research interests and creating opportunities for me to explore them.
- Always being patient teaching me on scientific writing.

Thanks to Florian for co-supervising me since the beginning of my Ph.D studies, your spirits of being always rigorous in work and curious about everything gave me the first taste of research. I really appreciated the chats you had with me when I was very unconfident in myself. Furthermore, the discussions with you inspired me so much to kickoff my research. Thanks to Marianna for involving me in your great project, that was the first work I did at CVAP and absolutely unforgettable. Thanks to Carl Henrik for all the fruitful suggestions for my work, I was quite impressed by your wide range of knowledge and you always had good answers to my random questions. Thanks to Renaud for sharing great views of research, I really enjoyed all the constructive discussions with you. Thanks to Yasemin for your great advices on how I should choose my research topics, and for your valuable suggestions on research methodologies.

Johannes, you have been a multi-role player. Firstly, I would like to thank you for the infinite amount of invaluable discussions about our research, as well as for your selfless and remarkable contributions in our collaborations. Secondly, traveling and sharing hotels with you have been nice experiences, the beers and chats we had on three different continents (so far) were unforgettable. Moreover, your perfectionism has made me to sometimes complain about myself :).

721! Of course this has to be highlighted, this is the greatest office ever! Joshua, I think we both deeply understand how it feels to sit with mathematicians. However, we started to enjoy it. Thank you for all the constructive discussions and your great contributions in our science work. I loved how you selected screws for installing the robot and our chats on some awesome topics, I wish you will enjoy reading more source codes! Anastasiia, thanks for never telling me who played tricks on my desk although you were sitting just in front. Also, thanks for your support (many times being the only one) in many interesting debates. João, you have shown me how it is possible to be always emotional. Thanks for upholding the group dynamics in our office, and most importantly, thanks for explaining me the manifold.

Team CVAP was definitely an amazing group! Thanks to captain Alessandro for your great management, we would not have had such a great team without your efforts! Also, I really enjoyed commenting on your codes :). Francisco, thank you for being a great architect in the team, I learned a lot from you while sitting next to
you coding together. Thank you also for organizing many awesome activities. Karl, no doubt that SimTrack was the core of the system. Thanks for many trainings we had and I really enjoyed your spirit of being "always thirsty". Thanks to Michele for your behavior tree, I really loved it when it behaved. Thanks to Johannes for discussing with me about localization and navigation, that was really helpful.

I would like to thank Patric for taking me as a TA in the AI course, it was very interesting developing the course together with the team, especially when I got pizzas. Thanks to Mårten for taking me as your teaching assistant, it was a great experience and I enjoyed it. Thanks to Hedvig and Anders, your advices during the supervisor group meetings greatly helped me in planning and developing my Ph.D.

Thanks to Aude for giving me the opportunity to visit your lab twice, giving me advices on research, and allowing me to heavily use the robots. Great thanks also to Miao for our great collaborations in quite a few works, as well as for your great contributions helping me to achieve the most important work in my Ph.D research.

Thanks to Nils for helping me with the Swedish abstract of this thesis. Thanks to Dan for the remarkable discussions about grasping. Thanks to Xavi for being a great officemate for a year, our fun time during the IEEE Xtreme 7.0 was unforgettable. Thanks to Yuquan for valuable discussions about control theory. Thanks to Yanxia for the nice chats about HRI. Thanks to Martin for sharing the office with me in the first two weeks of my Ph.D time. Great thanks to Cheng for sharing so much nice hometown food with me. Thanks to Christian on giving advices on my research. Thanks to Zhan for giving me a nice ride to Göteborg. Thanks to Akshaya for being a nice roommate in Kiruna. Thanks to Xi for many interesting chats in the Kitchen. Thanks to Atsuto for many interesting discussions about computer vision.

I also owe gratitudes to our administrators Friné, Jeanna, Camilla and Kerstin, your great efforts made it possible for me to focus on research. To CVAPers: Diogo, Rares, Hossein, Judith, Fredrik, Sergio, Silvia, Johan, Ali G., Püren, Rasmus, Emil, Virgil, Oskar, Ivan, Alejandro, Nauman, Ali R., Erik, Stefan, Jan-Olof, John, Josephine, Petter, Yiannis, Mikael, Hakan, Ramviyas, Niklas, Jeannette, Alper, Magnus, Kai, Vahid and Heydar (my sincere apologies if I missed anyone), thank you for making our CVAP such a great and cozy environment, it was my great pleasure and honor to work with you during these four years.

Finally, I would like to give my greatest thanks to my parents He Yan and Hang Qinghua, your warmest love and unconditional support have been always guiding me through the hard times. Because of you, I am now able to reach this point and write all this down.

Hang Kaiyu
Stockholm, April 2016
## Contents

**I Introduction**  
1 Introduction 3

1 A Brief History 5
2 Problems in Robotic Manipulation 6
3 Contributions 10
4 List of Papers 11

2 Robotic Grasping 13

1 Object Representation 14
2 Grasp Quality Metrics 17
3 Multi-Fingered Inverse Kinematics 18
4 Uncertainties in Grasping 19
5 Grasp Control and Adaptation 21

3 Summary of Papers 23

4 Conclusions 31

1 Summary 31
2 Future Work 32

References 33

**II Included Publications** 41

A Friction Coefficients and Grasp Synthesis A1

1 Introduction A3
2 Background and Related Work A4
3 Methodology A6
### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Evaluation</td>
<td>A8</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>A17</td>
</tr>
<tr>
<td>References</td>
<td>A17</td>
</tr>
<tr>
<td><strong>B Combinatorial Optimization for Hierarchical Contact-level Grasping</strong></td>
<td>B1</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>B3</td>
</tr>
<tr>
<td>2 Related Work</td>
<td>B5</td>
</tr>
<tr>
<td>3 Preliminaries</td>
<td>B6</td>
</tr>
<tr>
<td>4 Methodology</td>
<td>B8</td>
</tr>
<tr>
<td>5 Experiments</td>
<td>B12</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>B19</td>
</tr>
<tr>
<td>7 Acknowledgement</td>
<td>B20</td>
</tr>
<tr>
<td>References</td>
<td>B20</td>
</tr>
<tr>
<td><strong>C Hierarchical Fingertip Space for Multi-fingered Precision Grasping</strong></td>
<td>C1</td>
</tr>
<tr>
<td>1 Introduction and Contributions</td>
<td>C3</td>
</tr>
<tr>
<td>2 Problem Formulation and Notation</td>
<td>C5</td>
</tr>
<tr>
<td>3 Fingertip Space Representation</td>
<td>C7</td>
</tr>
<tr>
<td>4 Grasp Synthesis</td>
<td>C9</td>
</tr>
<tr>
<td>5 Experimental evaluation</td>
<td>C12</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>C19</td>
</tr>
<tr>
<td>References</td>
<td>C19</td>
</tr>
<tr>
<td><strong>D On the Evolution of Grasping Manifolds</strong></td>
<td>D1</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>D3</td>
</tr>
<tr>
<td>2 Terminology</td>
<td>D5</td>
</tr>
<tr>
<td>3 Methodology</td>
<td>D7</td>
</tr>
<tr>
<td>4 Experiments</td>
<td>D13</td>
</tr>
<tr>
<td>5 Discussion &amp; Conclusion</td>
<td>D18</td>
</tr>
<tr>
<td>6 Acknowledgments</td>
<td>D20</td>
</tr>
<tr>
<td>References</td>
<td>D20</td>
</tr>
<tr>
<td><strong>E Hierarchical Fingertip Space: A Unified Framework for Grasp Planning and In-Hand Grasp Adaptation</strong></td>
<td>E1</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>E3</td>
</tr>
<tr>
<td>2 Related Work</td>
<td>E5</td>
</tr>
<tr>
<td>3 Hierarchical Fingertip Space and Grasp Optimization</td>
<td>E6</td>
</tr>
<tr>
<td>4 Grasp adaptation</td>
<td>E15</td>
</tr>
<tr>
<td>5 Regrasping by finger gaiting</td>
<td>E17</td>
</tr>
<tr>
<td>6 Experimental evaluation</td>
<td>E19</td>
</tr>
<tr>
<td>7 Conclusion</td>
<td>E25</td>
</tr>
<tr>
<td>References</td>
<td>E26</td>
</tr>
</tbody>
</table>
Part I

Introduction
Chapter 1

Introduction

In 1921, the Czech playwright Karel Capek introduced the word *Robot* in his play Rossum’s Universal Robots. He defined the concept of a robot as a machine that resembles human skills and works long hours without being tired or bored. This was followed by a large volume of science fiction books and movies portraying robots with human-like embodiments, able of thinking, reasoning and complex decision making, even having personalities.

The actual trend and development in the industrial sector was rather different. The focus has always been on function, building machines that can perform repetitive and somewhat boring tasks, many of these requiring high precision. The focus has thus not been on developing anthropomorphic robots but achieving specific functionality. For example, robotic arms and end-effectors have been developed to repeatedly manipulate workpieces in various applications, maybe the best example being the car industry. The development has gradually gone from static arm robots to various mobile solutions, equipped with wheels or legs and also equipping robots with sensing to allow for an increased level of autonomy.

For the purpose of making robots autonomous and able of interacting with the environment, the ability to sense the environment is integral. As humans, robots can use various types of sensing, even some not present in humans, such as, for example, distance measuring sensors. One of the challenges has been to develop control and decision making algorithms that can in an optimal manner use information from several sensory modalities and also do this in real time. This is especially of importance when robots are expected to interact with various objects in realistic environments, taking into account problems such as deformability of objects, changing attributes in terms of weight, unstructured environment, and alike.

In this thesis, we address the problem of dexterous grasping which is an integral problem toward achieving interactive, autonomous robots. In our everyday life, hands are used constantly to enable us to interact with the world. We use our hands to pick up and manipulate things, to operate various tools, and do this
also differently if the structure of the environment is changing in some way. Our interaction with the environment is based both on our sensing but also experience that we gather and improve since our birth. In order to equip robots with similar functionality, there are many scientific problems in the wide area of robotic manipulation tat need to be addressed: mechanical design of robotic hands and grippers, force control and optimization for achieving and remaining the contacts, modeling of tactile feedback for adaptive grasping, the representation of objects and hand kinematics and planning of finger gaiting, to name some. Apart from addressing problems related directly to the hands or end-effectors, to allow a robots even more flexibility, the problem of mobile manipulation needs also to be addressed. An example of a fixed base manipulator and a mobile manipulator is shown in Figure 1.

Figure 1: Left: A Kuka-KR5 manipulator with a fixed base, performing a pick-and-place tasks on a table. Right: A PR2 robot training for the Amazon Picking Challenge\footnote{http://amazonpickingchallenge.org} task, the shelf and the collection bin are located in different places in the workcell and this requires mobile manipulation.

We continue the chapter by proving a brief history of robotic manipulation, followed by discussions of each individual component of an object manipulation systems. During the past few decades, extensive surveys have been written on this topic \cite{39, 52, 58, 6, 3}, and we do not aim to repeat these but focus on the work directly related to the problems addressed in this thesis.
1. A Brief History

The research on robotic manipulation can be traced back to 1940s when the first mechanical manipulator was developed at Argonne and Oak Ridge National Laboratories to handle radioactive material \cite{39}. In the early years, there was not so much autonomy and complex decision making integrated in the system. The robot worked in a teleoperation mode where the manipulator repeated the moves generated by a human user.

Later on in 1954, the first Computer Numerically Controlled manipulator was developed by George Devol. Rather than being teleoperated, the manipulator could be programmed to repeatedly perform tasks and could be reprogrammed at a relatively low cost to adapt it to different tasks. In 1969, the first robotic manipulator (aka. Stanford Arm, Figure 2) installed with cameras was designed at Stanford University by Victor Scheinman. For over 20 years, this robot was used in various projects and has been recognized as one of the first robots that opened up the modern research of robotic manipulation \cite{39}.

![Figure 2: The Stanford Arm developed by Victor Scheinman in 1969.](image)

The open kinematic chain structure of the robotic manipulator has enabled the robot to access a larger workspace in comparison to the space occupied by itself, and has provided more flexibility for performing more complicated tasks. However, it also brought up a whole set of new problems, such as the calibration throughout the whole kinematic chain, the position control of the cantilevered structure while regulating its stiffness, and the trajectory planning for the movement of the end
CHAPTER 1. INTRODUCTION

point, to name some. To address these problems, researchers all over the world have been working on the development of both hardware and software ever since. With the advancements of both, more complex object manipulation tasks have been demonstrated and the expectation are growing as we speak, even expectation on outperforming human dexterity.

2 Problems in Robotic Manipulation

In this section, we shortly overview various problems that have been addressed in the area of robotic manipulation such as hand design, sensing and perception, object grasping and manipulation. The goal is to provide the basics to a non expert reader, showing the complexity of the problem and what are the underlying assumptions being made in order to make the problem tractable.

2.1 Hand and Arm design

Embodiment design is one of the core parts of robotic manipulation, as it intrinsically determines the robot’s capabilities. This work can be divided into two parts: arm design and end-effector/hand design.

Arm: Robotic arms in general have five or more degree of freedom (DoF). DoFs are actuated through joints that can be either revolute joints or prismatic joints. In order to achieve arbitrary poses in the 3D Cartesian space, we need at least 6 DoF for translation (3 DoF) and rotation (3 DoF). However, due to the limitations of mechanical structures, a robotic arm can suffer from kinematic singularities at certain joint configurations, in the vicinity of which some of the joints would need to generate infinite velocity or torque to satisfy the movement control of the end-effector. Therefore, redundant manipulators with more than 6 DoF have been designed to achieve poses with more than one joint configuration, to avoid getting close to singularities. For example, as shown in Figure 1 the PR2 arm is a 7 DoF redundant arm.

End-Effector: Robotic end-effectors are usually designed to be the interface for a robot to interact with the external world. An end-effector is typically installed at the end of the robotic arm, its pose is determined by the joints configuration of the arm, and it is, in most of cases, the only part that can have physical contacts with the world. Currently, the majority of end-effectors are designed in an ad hoc manner for specific tasks and are usually simple, especially for industrial uses, e.g., grippers, pincers or tongs. In applications such as pick-and-place, suction devices have demonstrated their efficiency and robustness, see Figure 3.

In order to achieve better dexterity, multifingered hands have been developed to allow more actions to be performed, e.g., in-hand manipulation. In addition to changing arm configurations for repositioning the grasped object, a dexterous hand is able to reconfigure itself while holding an object, through either repositioning fingertips or rolling the contacts on the surface. Example multifingered hands can be seen in Figure 3. However, as the dexterous hands in general have many DoF,
the planning and control of the hand motions in this case are much more difficult than simple grippers.

In order to reduce the complexity of grasp planning for dexterous hand, as well as to provide better adaptability and robustness, adaptive under-actuated hands have been also designed. This type of hands usually have many joints to provide dexterity, however, very few active DoF to make the planning and control more tractable. An example of adaptive UNIPI-IIT hand is shown in Figure 3.

![Figure 3: Example designs of end-effectors: PR2 gripper, suction cup on the Baxter robot, Schunk SDH dexterous hand and UNIPI-IIT adaptive hand.](image)

### 2.2 Perception

Perception is another key problem in robotic manipulation. In order to perform certain tasks, one needs to first figure out which object to use, where is the object, what is its shape and how to grasp it, etc. Furthermore, once the object is successfully grasped, one also needs to regulate the forces applied by the end-effector to stabilize the grasp, and this needs some type of force sensing to close the control loop. In the research nowadays, there are mainly three ways of perception in the context of this problem: vision, tactile sensing and force/torque sensing.

**Visual Perception:** Visual perception is usually the first step of robotic manipulation. In cases when the object to be manipulated is unknown, a robot needs to first obtain the geometric model of it using visual sensors. This process is commonly composed of the following steps: raw data acquisition, segmentation, feature extraction and recognition. Depending on the tasks, the information of the object’s category is not always required. In many applications, it is assumed that the complete 2D or 3D model of the object is known. In this case, one needs to localize or track the object for further manipulation planning. For this, computer vision research has provided a rich set of different methods, two of the example trackers used in this thesis work are shown in Figure 4.

**Tactile Sensing:** Tactile feedback can be required at different stages of manipulation process. Before grasping, one can use tactile feedback to explore the shape of the object, as well as guiding the end-effector to plan a grasp by predicting stability from sensed data. The latter is known as vision-free grasping. Once the object is grasped, tactile feedback can provide further information to monitor
CHAPTER 1. INTRODUCTION

Figure 4: Example visual object trackers used in this thesis: SimTrack [41] and OptiTrack.

the system status. For example, one can predict the stability of the grasp, and even plan regrasping or finger gaiting accordingly. An example of tactile sensors used in this work is shown in Figure 5.

Figure 5: Left: SynTouch BioTac tactile sensor used in this work. Right: Force/Torque sensor installed on the wrist of the Dumbo robot.

**Force/Torque Sensing:** Force/Torque sensors are usually installed at the joints of manipulators, differently from the tactile sensors, it measures the force in 6D, however, not directly at contacts. This type of sensors can be used for compliant motion control, estimate of contact properties and force control of the manipulator etc. An example of force/torque sensor is shown in Figure 5.

2.3 Object Grasping and Manipulation

Object grasping is the core part of robotic manipulation. In most of cases, in order to manipulate the object, one has to grasp it first. As this is the main focus of this thesis, we provide a detailed discussion on this topic in Chapter 2.
In the area of robotic manipulation in general, researchers have tried to tackle different aspects or scales of the problem, such as contact modeling, task related object manipulation, dynamics and motion of manipulators, structure of the workspace etc. In the following, we give a brief overview of these different aspects. The approaches mentioned below intersect each other and it is not meant to give a formal taxonomy of robotic manipulation. We aim at summarizing major trend and show how this problem can be addressed from different perspectives.

**Pick-and-Place**: Pick-and-place is a typical and widely studied manipulation tasks. Here, a robot is expected to pick up an object from, e.g., a table, a shelf or a container, and then place them to another place. In some applications, objects are also required to be placed in a specific manner. For this, simple end-effectors are usually chosen, such as a parallel gripper, and the pose of the manipulated object is only determined by the configuration of the arm with a fixed base.

**In-Hand Manipulation**: With a dexterous hand, a robot is able to reconfigure the grasp while keeping the object always in the hand. By relocating fingertips on the object surface with a sequence of actions, the robot can change the contacts and object pose without having to move the arm. Due to the high dimensionality of the hand configuration space and complicated dynamics, it is still very challenging to achieve this type of manipulation. Currently, simple in-hand manipulation has been achieved by rolling contacts and finger gaoling.

**Mobile Manipulation**: In applications where the workspace of the robot is larger than the maximum extent of the arm, one can mount the robotic manipulator on a mobile base to expand its reachability. As this gives more DoF to the manipulator, such a system is usually redundant, i.e. there are infinite number of configurations to realize a required pose, and this makes it much more difficult than the manipulation with fixed base.

**Manipulation by Extrinsic Dexterity**: Differently from in-hand manipulation, in addition to make use of the dexterity of the hand to reconfigure a grasp, researchers have shown the possibility of manipulation by external forces. The sources of external forces can be contacts from anywhere in the environment, and even gravity.

**Non-Prehensile Manipulation**: Non-prehensile manipulation in general also uses external forces, however, it does not require a manipulator to grasp an object for manipulation. Current research shows that by pushing objects on a flat surface, one can achieve many interesting tasks. During the manipulation process, as there can be moments of time at which the object appears to be out of direct control, the planning of non-prehensile manipulation involves significant uncertainties that need to be modeled.

**Dual-Arm Manipulation**: This type of manipulation is motivated by several reasons: it provides more flexibility and manipulability by using two arms in a closed kinematic chain, and the robot can gain more possible actions by using two arms cooperatively. Additionally, it is easier to implement dual-arm manipulators in industrial setup, especially when we want to replace human workers with robots, since it requires less redesign of workspace.
3 Contributions

In this thesis, we focus on the problem of dexterous grasping for robotic manipulation. In particular, in order to improve the efficiency and robustness of fingertip grasping, we studied various types of uncertainties in the process of both grasp planning and execution, devising hand and object representations to facilitate contacts optimization, planning of online finger gaiting to resist external disturbances, and learning of grasping manifolds to enhance the reachability and inverse kinematics computation for multi-fingered grasping. Next, we summarize the contributions of this thesis in relation to the publications included in section 4. In Chapter 2, we further discuss our contributions in relation to the related work.

Friction Sensitivity: Friction coefficient is an important factor for the evaluation of contact-based grasp quality. As it is not trivial to estimate friction coefficients between the robot hand and arbitrary objects to be grasped, this uncertainty could make a grasp planner fail to generate stable grasps. In this direction, we proposed a metric, Friction Sensitivity for scoring contact-based grasps in terms of how susceptible a grasp is to changes in friction coefficients. Furthermore, we developed an algorithm for minimizing the friction sensitivity for grasp synthesis, as well as an algorithm that can still efficiently plan stable grasps when the friction coefficient is small. The details of this contribution is explained in the attached paper [A].

Object and Hand Representations: The synthesis of contacts for force-closed grasps is difficult for complex object shapes. In order to improve both the efficiency and the final quality of contacts optimization, we proposed an approach that directly operates on the mesh representations of arbitrary objects for finding high quality grasps. As detailed in the attached paper [B], we employed a multi-level approach that iteratively simplifies the objective function and results in fast convergence. The local search at each level effectively reduces the total number of evaluated candidate solutions to a practical amount, and it has significantly improved the final grasp qualities.

Moreover, by proposing the concept of Hierarchical Fingertip Space (HFTS), as explained in the attached paper [C], we extended this framework to operate on the point cloud representation of objects. In this work, we optimize contacts by further considering a hand reachability measure, which encodes fingertip contacts and hand kinematics in an affine invariant manner. This ensures the planned contacts on the object to not only form a stable grasp, but are also reachable by the adopted robotic hand.

Learning of Grasping Manifolds: In our work reported in the attached paper [C], we proposed a fast fingertip grasp planner that searches stable and reachable grasps in a hierarchical search space of fingertip contacts. To ensure kinematic feasibility for a dexterous redundant hand, an uniform sampling based affine-invariant hand reachability heuristic is used. However, as a common property of sampling based heuristics, the sampling resolution affects the efficiency and precision of this method.

In the attached paper [D], we presented a learning framework that incrementally
learns a subspace of the hand configuration space, a grasping manifold, that is relevant for fingertip grasping. As this learning process continues and uses the new grasp experiences to evolve the manifold, the grasp reachability and hand inverse kinematics heuristic learned based on this manifold is also being improved. We have shown that this effectively increases the accuracy and efficiency of grasp planning.

**A Unified Framework:** In order to improve the robustness of grasping systems, and to address the problem of grasp instability due to uncertainties such as changed weight of the object, slippage or external disturbances caused by collisions, in the attached paper [E], we presented a unified framework for fingertip grasping considering an integrated approach to grasp planning and in-hand grasp adaptation. This framework has three advantages as compared to the state-of-the-art: a) It provides an optimization framework for both grasp synthesis and finger gaiting; b) It closes the loop between grasp planning and control through stability estimation and finger gaiting; and c) It optimizes grasp adaptability and demonstrates informed finger gaiting optimization by considering viable hand configurations and object shape knowledge.

### 4 List of Papers

**Included Publications** This thesis is based on the following publications, a summary can be found in Chapter 3.


**Other publications**

Apart from the included publications, the following publication have been achieved during my Ph.D studies.


Chapter 2

Robotic Grasping

Most of the robot-object interactions require that an object is firmly held by the hands and that the grasp remains stable during the whole manipulation process. In 1956, Napier has classified human grasps into two main types [40]: power grasps and precision grasps.

The first type is power grasp that utilizes both the palm and fingers to achieve large contacts between hand and object surface, and it aims at maximizing stability. The other type is precision grasp that involves only fingertips at the contacts. This type of grasps in general provides more dexterity to change the object configuration in-hand. At the same time, it is less stable due to the limited contact area. Examples of these two grasp types are shown in Figure 1.

![Figure 1: From left to right: power grasp, precision grasp, and loop grasp.](image)

This grasp taxonomy has been later introduced into robotic grasping research [13]. Recently, based on the concept of caging grasp for objects in 2D [50], loop grasp has been developed using topological features of objects and hands [45], it provides a framework for planning caging grasps for objects with holes in 3D. This type of grasps is robust under sensory noise and does not require explicit planning of contacts, see Figure 1.

The planning and execution of grasps are composed of many subproblems, such
as the object representation, hand kinematics representation and computation, grasp stability metric design, handling of uncertainties and control [3, 6, 52, 68].

In general, grasping can be addressed using a structure shown in Figure 2, which consists of three phases. In the pre-grasping phase, the system does grasp planning with the considerations of some subproblems, such as problem representation, inverse kinematics computation, task planning and stability evaluation etc. In the grasping phase, the system executes the planned grasp, which is usually represented by contacts, hand joint configurations and hand pose. This requires the system to consider various uncertainties to ensure successful executions. After the grasp is executed, the system enters the post-grasping phase, for which the main task is to maintain the grasp stability against either uncertainties such as external disturbances or task related dynamic changes, e.g., a grasped container can be filled and requires the grasp to adapt to the weight and inertia changes.

Figure 2: Overview of a common structure used by robotic grasping, this structure consists of three phases: pre-grasping, grasping and post-grasping.

In this chapter, we review related work in relation to our work and discuss the challenges of this problem.

1 Object Representation

In the research area of grasp planning, it is often assumed that a geometric model of the object is available, either as a point cloud or a mesh model. Depending on how the object model is obtained, it can be either complete or partial. For grasp planning, many works have proposed to construct a representation based on the raw sensor data, and then search algorithms are applied on the representation of the sensor data to generate grasp hypotheses. Next, we review the object representations for grasp planning and discuss what types of grasps can be generated using these.
1. OBJECT REPRESENTATION

1.1 Geometric Representation

Given the raw data of arbitrary objects, many works employ sampling algorithms for generating a set of candidate grasps, and then rank these according to a chosen quality metric. To limit the amount of candidate grasps and provide grasping relevant information for the sampler, extraction of shape primitives, such as spheres, cones, cylinders and boxes, has been studied [37, 29]. Rather than using predefined primitives, parametric models has been also used to approximate shapes. In [42], superellipsoids are used to parametrize object shapes for an SVM based grasp planner. In [22], superquadrics are used to hierarchically decompose shapes into parts, based on which a simple sampling heuristic is used to generate grasps on parts. Bounding box approximation is also used for approximating the global shapes of objects, and the surface normals on the box are used as a heuristic to sample grasps [15]. As the objects are approximated and the detailed surface information is not considered, the above mentioned methods can only be used to generate power grasps. Examples are shown in Figure 3.

![Figure 3: Object shape primitive approximation for grasp generation: (a) Box decomposition [29]. (b) Random sampled grasp contacts with a heuristic [8]. (c) Grasp candidate sampled based on surface normals and bounding box [15]. (d) SQ decomposition of parts of the object for grasp sampling [22].](image)

For generating precision grasps on simple shapes, [8] has shown that it is feasible to randomly sample a number of contacts dependent on the object surface, and then filter the contacts by a simple heuristic. For more complex arbitrary objects, offline planning algorithms have been used to generate a database of stable grasps for a
set of known shapes. By specifying an initial hand pose, \cite{17} generates a dense set of grasp contacts using an Axis Aligned Bounding Box (AABB) tree decomposition of objects. Using a non-linear optimization formulation with single constraint, \cite{17} finds optimal grasps on superquadrics. To enable grasp transfer between shapes that can be parametrized in the same coordinate system, our work in the attached paper \cite{C} proposed an infinite dimensional space, the Grasp Moduli Space, which models shapes and grasps in a continuous manner allowing to smoothly transfer grasps between different objects, see Figure 4. On arbitrary shapes that are not easily parameterizable, our work in the attached paper \cite{B} applies a hierarchical abstraction on the original object mesh, and then synthesizes contacts by multi-level refinement. With the Hierarchical Fingertip Space representation of an object’s point cloud, our work in the attached paper \cite{C} generates fingertip grasps by searching in a hierarchy of abstracted search space. It is worthwhile to note that the hand reachability and inverse kinematics computation are crucial for the planning of fingertip grasps, since all the planned contacts have to be precisely realized by fingertips only, see detailed discussions about this in section 3.

Figure 4: Samples from a finite-dimensional subset of the Grasp Moduli Space spanned by the three surfaces displayed at the vertices of the triangle \cite{44}.

1.2 Topological Representation

For representing object shapes by skeletal features, which do not describe object surface details, topological methods have been used together with heuristics to generate grasps. \cite{1} proposed to use Reeb Graph as a representation for part based grasp planning. \cite{67, 47} plans grasps using a medial axis representation and executes them using predefined policies. In the work of \cite{73}, topological synergies has been defined and used to transfer grasps between different hands. In order to enable
2. GRASP QUALITY METRICS

cage grasping in 3D, \cite{45, 63} uses a topological algorithm to find holes or loops in the object’s point cloud, and then plans a specific type of caging grasps termed as latching or hooking. Figure 5 shows some example grasps generated by different topological representations. Since this type of grasp planning algorithms aims at generating grasps wrapping around the skeleton of object, it is in general very stable and not sensitive to noise.

Figure 5: Topological representations for grasp planning: 5(a): Reeb Graph for grasping by parts \cite{1}. 5(b): Medial axis for generating a set of grasp candidates \cite{47}. 5(c): Objects with holes and example latching or hooking grasps on them \cite{45}.

2 Grasp Quality Metrics

Finding good contacts and hand configurations to realize these is crucial for grasp planning, especially when dexterous manipulation is considered. In order to allow an algorithm to automatically generate useful grasps, one has to first determine the criteria of evaluating the goodness and validity of grasps \cite{5, 49}. 


A classical way of doing this is to evaluate a grasp by its ability of resisting disturbances. Based on the contact positions and force limits, [46] defined three wrench spaces for stable grasps: 1) The limit wrench space which is bounded by the convex hull of all wrenches satisfying unit force and torque constraints; 2) The object wrench space which is bounded by the convex hull of all wrenches satisfying unit force and torque constraints, and is additionally constrained by object geometry. This space describes the best grasp can ever be achieved on this target object; and 3) The grasp wrench space that describes the wrench space of a contact-based grasp on the target object. Based on these definitions, two grasp qualities have been proposed by exploring the relationship between 2) and 3). Furthermore, [21] proposed to measure grasp quality by the radius of the inscribing ball of the convex hull spanned by contact wrenches, which aims at generating force-closure grasps. Form-closure gives a similar measure of grasp quality, however, without relying on frictional contacts [59]. The force-closure based grasp qualities have been implemented in many grasp simulators and grasp planners to generate stable grasps [15, 38, 4]. For addressing the problem of pose uncertainties, which makes the static object assumption does not hold in reality, [30] demonstrated a physical simulation based quality measure to improve the robustness of generated grasps.

In terms of hand configurations, a grasp can be also evaluated by its dexterity. This metric measures grasps in terms of the ability of changing the object configuration in any directions [55]. Based on the same concept, our work reported in paper [E] proposed to measure the manipulability in the tangential plane of contacts, which evaluates how much a grasp can adapt to its neighboring grasps by finger gaiting. With the constraints of specified tasks for a grasp, task related quality measures have been proposed to ensure the planned grasp is able to achieve the task [9, 26].

For the online evaluation of grasp stability, tactile feedback has been widely used to monitor the dynamics of a grasp and estimate its state. [2] proposed a probabilistic framework for assessing task-oriented grasp stability. Based on a Gaussian Mixture Model trained on a virtual frame of grasp impedance controller, [33] estimates grasp stability online and adapts grasp forces and contacts to keep the stability.

3 Multi-Fingered Inverse Kinematics

For fingertip grasping, localizing contacts on the object that provide a stable grasp [21, 8, 48] and finding a hand configuration for realizing these contacts [36, 7, 51] have been addressed as separate problems. As a result, in these approaches it is not guaranteed that the planned contacts are kinematically feasible for a specific hand [8, 52].

To overcome this limitation, various grasp optimization frameworks have recently been proposed that integrate both stability and hand reachability analysis.
4. UNCERTAINTIES IN GRASPING

In [17], the finger kinematics are modeled as an optimization constraint for efficiently finding contacts on objects that can be approximated by super-quadrics. Given a set of initial hand poses, [56] optimizes both hand configurations and contacts to generate a dense set of grasps offline. The framework presented in [23] plans contact triplets on incomplete 3D point clouds online, utilizing hand shape primitives as a heuristic to constrain the contacts optimization. In [18], a system first learns task related grasping parts of objects from human demonstrations. Next, the contact points on these parts are found by stochastic optimization, which is constrained by closed kinematic loops between the fingertips and the object part. In general, these systems require trade-offs between the desired computational efficiency and the precision of planning by either approximating the object shapes or the hand kinematics.

In the attached paper [C], a fast fingertip grasp planner is proposed that searches stable and reachable grasps in a hierarchical search space of fingertip contacts. To ensure kinematic feasibility for a dexterous redundant hand, an uniform sampling based affine-invariant hand reachability heuristic is used. However, as a common property of sampling based heuristics [66, 67, 26, 72], the sampling resolution affects the efficiency and precision of this method. To achieve the same resolution, the number of required samples increases exponentially with the dimension of the configuration space. Further, in practice many of the samples cover invalid or task irrelevant configurations.

Instead of sampling randomly, learning from humans effectively excludes irrelevant hand configurations [20, 53]. However, this requires significant effort from human teachers and is limited by the teacher’s experience. To address this, our work in the attached paper [D] limits the training set for a sampling based hand reachability heuristic to the manifold of for grasping relevant hand configurations. In doing so, it increases the heuristic’s accuracy in relevant regions of the configuration space, while reducing it in irrelevant regions, see Figure 6. It is worthwhile to note that in this work a robot learns this grasping manifold based on its own task relevant experiences.

4 Uncertainties in Grasping

Considering uncertainty in robotic grasping, both for planning and control, has become increasingly important. To deal with various uncertainties integral to the grasping problem, one approach is to use sensory feedback to perform grasp adjustment locally to find stable grasps near the original planned grasp [14, 28, 19]. For instance, [28] proposed a set of simple and efficient heuristics to reactively correct the alignment error of the PR2 gripper. In [19], a sensor-based grasp primitive of Barrett hand is developed to adapt to the variation of the task conditions. These methods are usually reactive using actual sensing data from force or tactile sensors. The reactive correction strategy is designed to alleviate the need for precise hand-object pose information and hence can be more robust to pose and location...
uncertainty. The main disadvantage of these methods is that, to design the corrective strategy, the grasp is usually limited to a predefined set of grasp primitives [19] or only simple hand kinematics is considered [28]. For a more complex dexterous hand with a possibility to execute a large variety of grasps, it becomes more difficult to design such a corrective strategy. In [33], an object-level grasp adaptation is proposed to deal with physical uncertainties in object mass and friction, focusing mainly on the grasping stage when the object is already in the hand.

Another approach to deal with the uncertainty is to consider uncertainty during the planning process. One way is to incorporate the robustness consideration into the planning, preferring grasps that are insensitive to the uncertainty or search for stable graspable regions on the object [7, 10, 43, 12]. For instance, the concept of independent contact regions (ICRs) is introduced to provide robustness to finger placement error [43] where any grasp with fingertip’s positions inside the ICR will achieve a force-closure grasp. Our work [A] proposed to score grasps by the friction sensitivity, which maximizes the robustness of grasps under friction uncertainty.

The uncertainty information can also be updated online using vision [51, 33] or tactile exploration [14, 51, 3]. However, these approaches usually use a set of predefined grasps and require several rounds of grasp trials. Few works have considered object shape uncertainty by integrating planning and control. In real robotic grasping tasks, object shape uncertainty is inevitable due to, for example, occlusion problems [5, 23, 4] or non-reachability from tactile exploration [51].

In our work in the attached paper [E], we addressed this problem by integrating grasp planning and grasp adaptation. By utilizing the online feedback from tactile sensors, our framework first plans grasps that are stable and adaptable. During
the execution period, our systems maintains the grasp stability by either for adap-
tation or finger gaiting. This approach tackles the uncertainties in grasping by
actively reacting to them. However, it requires fast online computation to trigger
the reactions before the grasp fails.

5 Grasp Control and Adaptation

Approaches to force based grasp control range from geometry based analytic meth-
ods [26, 9, 43] to learning-based frameworks for force optimization [70, 69]. In-hand
manipulation has been addressed as finger gaiting with a rolling contact model and
quasi-static assumption [24, 54]. Hybrid position and force control has also been
studied [34, 71, 11, 65] as well as impedance control [57, 65, 64, 82].

In the attached paper [E], we employed an impedance controller to regulate
the forces at contacts online. Based on an offline learned stability estimator, our
controller is able to adapt the forces against various uncertainties. Moreover, by
fast online planning of finger gaiting, we change the contacts of a grasp in-hand by
superimposing a virtual spring in the impedance controller. The ability of finger
gaiting has significantly improved the robustness of the grasping system, as external
disturbances can be dynamically balanced with forces more dedicated for it.
Chapter 3

Summary of Papers
Paper A

Friction Coefficients and Grasp Synthesis

Studying grasping under uncertainty is an important area in robotics. While most current state of the art approaches concentrate on aspects of imperfect object or robot models, we studied another fundamental problem in grasp synthesis in this work: the dependence of grasp stability on friction coefficients. We believe that this is an important problem when robots are to operate in open-ended environments with changing conditions.

We have in particular studied the statistics of stable grasps under changes in friction coefficients and have introduced the notion of friction sensitivity measuring the susceptibility of a grasp’s quality to changes in friction. Furthermore, we have proposed and evaluated two gradient ascent algorithms for synthesizing force-closed contact configurations on parametric surfaces with potentially low friction and for the synthesis of stable grasps with low friction sensitivity.

In our future work, we would like to evaluate our approach with a real robot and study changes in friction coefficients in a real application such as a household robot cleaning dishes which might be dirty or wet, impacting heavily on the resulting friction properties. Further directions might also include the study of alternative friction models and grasp quality scoring functions.

Contributions by the author

a. Proposed the metric Friction Sensitivity.

b. Proposed the algorithm for grasp synthesis under friction uncertainty.

c. Implemented all the experiments.

d. Evaluated the proposed algorithms.
CHAPTER 3. SUMMARY OF PAPERS

Paper B

Combinatorial Optimization for Hierarchical Contact-level Grasping

We proposed a method for synthesizing grasps based on point contact search using search space discretization. Grasp synthesis is formulated as a combinatorial optimization problem. Based on a multilevel refinement, we have considered a hierarchy of recursively coarsened mesh models and locally improved grasps at each level. For this, we have optimized grasp quality subject to a reachability measure at each level and matched grasps to the next lower level, to finally achieve a grasp on the original object model.

Our approach is applicable to complex input meshes, having thousands of faces and highly concave shape. Thus, it provides the first tractable method for search of grasp contacts on such input data.

Our empirical evaluation shows that the method produces feasible, high quality grasps from random and heuristic initializations. It outperformed random sampling relying on a large number of generated grasps and automatic hand closing technique. Analyzing the influence of object model properties and method parameters such as the number of coarsening levels and the size of the local search space, we have shown that our approach is viable. In the future, we plan to additionally consider hand kinematics in the optimization function and we intend to study the influence of input data uncertainty on our approach.

Contributions by the author

a. Proposed to use abstracted models for contacts optimization.

b. Implemented most of the algorithm.

c. Conducted most of the evaluations.
Paper C

Hierarchical Fingertip Space for Multi-fingered Precision Grasping

In this work, we have proposed a concept of Fingertip Space, which is an integrated representation of both object local geometry and fingertip geometry, and shown its use in precision grasp synthesis. By building a hierarchical representation of the fingertip space, we have enabled multilevel refinement for precision grasp synthesis. Our experimental evaluation with a Barrett hand has shown that the fingertip space and its hierarchy is a viable and efficient representation for precision grasp synthesis, and that the multilevel refinement facilitates the search procedure. We have also evaluated the positioning errors tolerance of our system, as well as demonstrated examples of our system working with noisy and incomplete data. In the future, we are planning to implement our system on a real robot and additionally make the modular system more compact and flexible for different robot embodiments and search algorithms to be plugged in.

Contributions by the author

a. Proposed the concept of Hierarchical Fingertip Space.
b. Proposed sampling based hand reachability metric.
c. Implemented all algorithms.
d. Conducted all evaluations.
CHAPTER 3. SUMMARY OF PAPERS

Paper D

On the Evolution of Grasping Manifolds

In this work, we presented a system that integrates a random forest based heuristic for hand reachability and multi-finger inverse kinematics with the fingertip grasp planner presented in [25]. The training set of the heuristic, the grasping manifold, is incrementally adapted by the system based on the experiences it gathered online. This allows to focus the heuristic’s limited accuracy on the regions of the robot hand’s configuration space that are task relevant. We showed that our system creates grasping manifolds that facilitate the performance of the grasp planner, while being able to generalize and continuously adapt to new types of experiences.

We further extended the proposed system to learn a heuristic from a grasping manifold that integrates both grasp quality and hand reachability, rendering the use of an analytic grasp quality metric unnecessary. As a result, we show that the planner’s runtime is reduced, while it is still capable of generating stable grasps.

The system was evaluated both in simulation as well as on a real robot, whereas training was only performed in simulation. Ideally our system would learn its grasping manifold from experienced grasps that were successful in the real world. However, our current implementation of the system does not take the environment nor the kinematics of the robot’s arm into account. As a result, the planner often produces grasps that are not kinematically reachable by the arm or in collision with the environment. Therefore, since the training requires a large number of executions, our evaluation on the real robot was limited to testing a system that was trained in simulation. In future work, we wish to explore the system’s performance when its experiences are gathered in the real world. As the grasping manifold is learned from successfully executed grasps, it would implicitly emphasize grasps that have been stable and suppress grasps that have falsely been predicted to be stable. This may occur due to errors in the used model, such as the simplified modeling of contacts and unknown friction coefficients. A grasping manifold trained from real world examples could overcome these limitations of analytic grasp quality metrics. Moreover, we would like to look into the scalability of the proposed method with respect to the DoFs of the hand and the range of different object shapes it is able to cover. Here, it is interesting to investigate the possibility of adaptively evolving the number of samples used to represent the grasping manifold.

Contributions by the author

a. Proposed the framework of grasping manifold evolution.
b. Proposed the learning algorithm for manifold update.
c. Implemented most of the algorithm.
d. Conducted approximately half of the evaluations.
Hierarchical Fingertip Space: A Unified Framework for Grasp Planning and In-Hand Grasp Adaptation

We have presented a unified framework for grasp planning and in-hand grasp adaptation using visual, tactile and proprioceptive feedback. The proposed Hierarchical Fingertip Space defines a hierarchy of surrogate solution spaces of fingertip grasping enabling both planning and adaptation. By augmenting the fingertip space in terms of local geometry and spatial relations, as well as optimizing hand configurations with respect to grasp adaptability, we demonstrated efficient planning and adaptation. Moreover, the probabilistic model for grasp stability estimation and adaptation has shown its feasibility in closing the loop between grasp replanning and control. We have evaluated the performance of the proposed system quantitatively and shown that the proposed system significantly improves the robustness of grasp execution. It also outperforms our previous work reported in [33]. To the best of our knowledge, this is so far the first system that accomplishes grasp synthesis, stability estimation, online replanning and in-hand adaptation in a unified framework, as well as evaluating this on a real physical system.

However, as a basic drawback of most learning based approaches, our probabilistic model is experience based, and hence relying on the training data, i.e. limited number of objects and examples to generalize from. As a potential future work, we plan to design an active learning strategy to update this model iteratively using new experiences over time, so as to evolve the model in a long term to generalize it to a broader set of objects, without retraining the models from scratch.

Contributions by the author

a. Proposed the unified framework for both grasp planning and grasp adaptation.

b. Proposed the search algorithm for online finger gaiting.

c. Implemented the grasp planning and finger gaiting algorithms.

d. Implemented and conducted approximately half of the evaluations.
Chapter 4

Conclusions

Grasp planning and in-hand grasp adaptation are two complex problems that have commonly been studied separately. Many important contributions to these problems have been made during the past two decades considering stability modeling and estimation, task based grasping, object representation, hand synergies and grasp adaptation, etc. In this thesis, we focused on studying the uncertainties of friction coefficients for dexterous grasping, efficient representations of object and hand kinematics for grasp optimization, incremental manifold learning of for grasping relevant hand configurations, and a unified framework for grasp planning and grasp adaptation.

1 Summary

In order to understand the effect of friction uncertainties in dexterous grasping, we systematically studied the impact of changes in friction coefficients on the stability of grasps in the context of a popular $L^1$ grasp quality measure. For this, we proposed the concept of friction sensitivity of a grasp with respect to grasp stability, and evaluated algorithms for synthesizing stable grasps with low friction sensitivity and for small friction coefficients.

For planning multiple contacts on arbitrary shapes represented by meshes, we introduced multilevel approach that iteratively simplifies the objective function resulted in fast convergence. Furthermore, the local search at each level effectively reduced the total number of evaluated candidate solutions to a practical amount.

In the case of shapes presented by point clouds, we proposed the Hierarchical Fingertip Space, which provides an optimization framework for both grasp synthesis and finger gaiting, as well as closing the loop between grasp planning and control through stability estimation and finger gaiting.

We also addressed the problem of improving the efficiency and accuracy of reachability and inverse kinematics computation for multi-fingered grasping. We
presented a grasp planning system that incrementally learns from experiences, and improves a random forest based reachability heuristic for a redundant hand. Additionally, we showed that we can extend this learning to estimate grasp stability for a variety of objects, rendering the use of an analytic grasp quality metric unnecessary.

2 Future Work

Grasping is not a standalone task and not the final goal we want to achieve. It has to be employed in a larger task setup to be actually useful, e.g., after picking up some object, we need to either place it to another place or use it for some other task. Additionally, a robot often interacts with and is constrained by the environment, modeling the grasping system in the combination of the robot itself and its environment will give more insights on how a grasp can be useful.

As discussed in Chapter 2, many problems in grasping still remain challenging. In the future work, we would like to look into the following:

- **In-Hand Manipulation**
  In-hand manipulation is very demanding for many tasks. Humans use this type of manipulation every day to achieve different goals, e.g., manipulating the mobile phone or picking a key and plugging it into a lockhole. As the main focus of this thesis, dexterous grasping is the cornerstone of in-hand manipulation. In the attached paper [E], we already touched upon it by finger gaiting. However, we only addressed the problem of maintaining stability.

  As a future work, we will explore the task-oriented finger gaiting, e.g., rotating or translating the object in the hand. This could require planning a sequence of consecutive finger gaits to achieve a single motion primitive, dependent on the shape of object and the initial configuration of the grasp. Furthermore, we would like to incorporate object recognition into grasp planning, with the information of the object’s category, the planning of dexterous grasping can be further optimized by considering future tasks associated to a grasp.

- **Mobile Manipulation**
  A manipulator with a fixed base is very limited in its reachability. Furthermore, for a non-redundant manipulator, there is only one configuration for realizing a desired pose. When the resulted configuration is close to singularities, the manipulability of the manipulator will be very small. To overcome these limitations, one of the straightforward solutions is to exploit mobile manipulation.

  Due to the extra DoF added to the manipulator, the planning of mobile manipulation becomes more challenging [27]. The robot needs to first navigate to a suitable position, and then manipulate the objects within its reachable range. More difficultly, in terms of the task or environment constraints, a
robot can also be required to manipulate objects while moving its base. In our future work, we would like to work on, in particular, the optimization of base pose and planning of navigation trajectory for better manipulation.

- Bimanual Manipulation
  This type of manipulation is motivated by several reasons. Firstly, it provides more flexibility and manipulability by using two arms in a closed kinematic chain. Secondly, the robot can gain more possible actions by using two arms cooperatively. Furthermore, it is easier to implement dual-arm manipulators in industrial setup, especially when we want to replace human workers with robots, since it requires less redesign of the workspace [60].

Extending from the work reported in this thesis, we would like to model the planning of bimanual grasping in our framework of fingertip grasp planning. This is feasible because we can regard the contacts made by two end-effectors as the contacts by fingertips, and the only difference is the kinematics. However, this would enable us to work on more complicated “finger gaiting” with more flexibility, which is given by the fact that while one end-effector is holding the object, the other one can freely move to other positions without the stability.

- Grasp Control and Adaptation
  Handling uncertainties is another important topic in dexterous grasping. Due to the noise in perception and inaccuracy in controllers, a planned grasp can never be exactly executed. Because of the potential environment changes or external disturbances, a grasp is desired to be adaptive and able to re-plan when needed. In the attached paper [E], we have touched upon the problem of force regulation under various uncertainties.

  However, this considers only the stability maintenance aspect, and assumes quasi-static models. For the applications of in-hand manipulation, this assumption can be invalid, since the dynamic motions of object can be desired. Therefore, in the future work will upgrade our current framework to provide grasp control or adaptation for more complicated manipulations.

“The most beautiful thing we can experience is the mysterious. It is the source of all true art and science.”

- Albert Einstein

Therefore, there is a lot to do.
Bibliography


35


REFERENCES


REFERENCES


Part II

Included Publications
Paper A

Friction Coefficients and Grasp Synthesis

Published in
Friction Coefficients and Grasp Synthesis

Kaiyu Hang  Florian T. Pokorny  Danica Kragic

Abstract

We propose a new concept called friction sensitivity which measures how susceptible a specific grasp is to changes in the underlying friction coefficients. We develop algorithms for the synthesis of stable grasps with low friction sensitivity and for the synthesis of stable grasps in the case of small friction coefficients. We describe how grasps with low friction sensitivity can be used when a robot has an uncertain belief about friction coefficients and study the statistics of grasp quality under changes in those coefficients. We also provide a parametric estimate for the distribution of grasp qualities and friction sensitivities for a uniformly sampled set of grasps.

1 Introduction

Friction coefficients are important for determining the quality of a specific grasp and for understanding whether a grasp is force-closed or not. Most state of the art grasp synthesis approaches typically assume fixed friction coefficients and evaluate an associated grasp quality measure such as the $L^1$ grasp quality $Q_\mu$. In reality, friction coefficients may vary depending on temperature, humidity and the presence of dirt on an object. Also, a robot will rarely have knowledge of precise friction coefficients to start with. Instead, we may only be able to estimate a confidence interval of friction coefficients. In this work, we address the following related issues:

a) We systematically study the impact of changes in friction coefficients on the stability of grasps in the context of a popular $L^1$ grasp quality measure $Q_\mu$.

b) We propose the concept of friction sensitivity $S_{n_0}(g)$ of a grasp $g$ with respect to $Q_\mu$ and fit a Dirichlet distribution to the distribution of $(Q_\mu(g), S_{n_0}(g))$ for uniformly sampled grasp configurations with three contact points.

c) We propose and evaluate algorithms for synthesizing stable grasps with low friction sensitivity and for small friction coefficients.

The paper is structured as follows: In Section 2 we discuss related work and introduce preliminaries. In Section 3 we define friction sensitivity and describe our
algorithms for grasp synthesis. We discuss our experiments in Section 4. Finally, we conclude our work and discuss future directions in Section 5.

2 Background and Related Work

In the following, we review the grasp quality function $Q_\mu$ and the basics of friction coefficients and grasp synthesis.

2.1 Grasp synthesis and $L^1$ grasp quality

Similarly to the work reported in [3], we focus on determining contact point configurations on a surface $S$ which result in a force-closed grasp $g$. We consider grasps

$$g = (c_1, \ldots, c_m, n_1, \ldots, n_m, z) \in \mathbb{R}^{3m} \times (\mathbb{S}^2)^m \times \mathbb{R}^3$$

going from contact points $c_i \in S$ on some surface $S \subset \mathbb{R}^3$ and with corresponding inward-pointing unit normal vectors $n_i \in \mathbb{S}^2 = \{x \in \mathbb{R}^3 : ||x|| = 1\}$ such that $S$ has centre of mass $z \in \mathbb{R}^3$. To determine if such a grasp $g$ can withstand external forces, we need to estimate if $g$ is a force-closed grasp [2]. Ferrari and Canny [5] introduced an $L^1$ grasp quality measure $Q_\mu$ which can be used for this purpose. For a fixed friction coefficient $\mu > 0$, the Coulomb friction model states that under the assumption that no slippage occurs – forces applied at a contact $c_i \in \mathbb{R}^3$ on some surface $S$ and with corresponding inward pointing unit normal vector $n_i \in \mathbb{R}^3$ satisfy $||f_i|| \leq \mu f_i^\perp$, where $f_i^\perp \in \mathbb{R}^3$ denotes the component of $f_i$ tangent to $S$ at $c_i$, $f_i^\perp \in \mathbb{R}$, and $f_i^\perp n_i$ denotes the component along the normal direction $n_i$ - i.e. these forces have to lie within the friction cone $C_i = \{f_i \in \mathbb{R}^3 : ||f_i|| \leq \mu f_i^\perp\}$. For

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{For the red and blue contact point configurations depicted on the object, we consider how the grasp’s stability, measured in terms of a popular grasp quality function, varies with changing friction coefficients. The vertical axis depicts grasp quality, while the assumed friction coefficient $\mu$ is varied from 0.2 to 1.0. The blue grasp remains more stable under changes in friction, while the red grasp yields more stable grasps for higher friction values.}
\end{figure}
2. BACKGROUND AND RELATED WORK

A particular grasp \( g \) as above, these friction cones \( C_i \) can then be approximated by \( C_i \approx \{ \sum_{i=1}^{l} \alpha_i f_{ij} : \alpha_i \geq 0 \} \) for \( l \) uniformly spaced vectors \( f_{i1}, \ldots, f_{il} \in C_i \) satisfying \( \langle f_{ij}, n_i \rangle = 1 \).

In this paper, we use \( l = 8 \) such uniformly spaced vectors. To define this \( L^1 \) quality measure, one then approximates the set of wrenches satisfying \( \sum_{i=1}^{m} |f_{ij}| \leq 1 \) by the convex hull \( \text{Conv}(\{0\} \cup S(g)) \), where

\[ S(g) = \text{Conv}(\{w_{ij} : i \in \{1, \ldots, m\}, j \in \{1, \ldots, l\}\}) , \]

and where \( w_{ij} = (f_{ij}, (c_i - z) \times f_{ij}) \). \( Q_\mu(g) \) is then defined to be the radius of the largest ball inside \( \text{Conv}(\{0\} \cup S(g)) \) and centred at the origin. To compute \( Q_\mu(g) \), \( S(g) \) is represented as an intersection of affine half-spaces \( S(g) = \cap_{j=1}^{l} \{ x \in \mathbb{R}^6 : \langle x, v_j \rangle \leq \lambda_j \} \) for some \( \lambda_j \in \mathbb{R}, v_j \in \mathbb{R}^6, \|v_j\| = 1 \), which can be obtained using the Quickhull algorithm [1]. Then \( Q_\mu(g) = \max(0, \min_j \lambda_j) \). If a grasp \( g \) satisfies \( Q_\mu(g) > 0 \), it is force-closed and can withstand external wrenches in arbitrary direction. Furthermore, grasps are considered more stable the larger \( Q_\mu(g) \) is.

2.2 Friction coefficients and grasp synthesis

For the purpose of robotic grasping, friction is commonly modelled using Coulomb’s friction laws [2] for some friction coefficient \( \mu \) as above. Friction coefficients depend on various parameters: [8] discusses in particular the influence of humidity on friction, while [5] study the dependence of friction on temperature. Further factors influencing friction include surface properties such as dusty or oily vs. dry and clean surfaces [18].

Since robots are to work in extreme conditions such as in search and rescue operations and in manufacturing applications, the impact of environmental factors on friction coefficients should be considered an important component in the analysis of grasp hypotheses. Even in less-extreme scenarios, such as that of a service robot in a home environment, friction can be influenced by dusty or dirty surfaces and can vary even during a manipulation task, e.g. when a robot is washing dishes. Clearly, none of these environmental factors can be determined exactly, and the robot hence needs to operate with an expected friction value. In current grasp synthesis work, such friction coefficients are often set to a fixed value according to friction tables for various material combinations [2, 3].

To the best of our knowledge, the problem of assessing the goodness of a force-closed grasp with respect to robustness under changes in friction has so far not been studied in depth. One work which mentions the problem of uncertainty in friction coefficients is [13] where the impact of uncertainties in friction and contact positions on grasp synthesis is discussed. In order to deal with uncertainty in friction coefficients, the authors suggest to work instead with ‘effective friction coefficients’ which are obtained by multiplying the coefficients of friction by some fixed reduction rate \( \frac{1}{\kappa} \leq 1 \) which is assumed to be known. In the work of [14], independent contact regions are computed on discretized objects taking into account uncertainties in
friction coefficients. There, these uncertainties are also modeled using a reduction rate. Based on the same concept, [17] developed an algorithm to compute minimal required friction coefficients and contact forces.

3 Methodology

Fig. 1 displays two examples showing how the grasp quality measure $Q_\mu$ changes with respect to changes in the assumed friction coefficient $\mu$ for the depicted contact configurations and for $\mu \in [0.2, 1.0]$. This figure highlights several important features of the function $\mu \mapsto Q_\mu(g)$. Observe, for example, that the graphs are monotonically increasing with increasing friction and that they have a distinct almost piecewise-linear looking shape which seems to be a generic property we encountered also for other object shapes. Furthermore, we observe that, for $\mu = 1$, a ranking of these two grasps based on grasp quality alone would return the red grasp as a preferable grasp hypothesis, while this grasp is unstable for $\mu = 0.2$ where the blue grasp remains stable. A natural question arises: which grasp should we choose if we only have knowledge of a confidence interval $\mu \in [0.2, 1]$?

A friction coefficient of 0.2 corresponds to the friction of a polythene (plastic) surface in contact with a steel surface, while a friction of 1.0 corresponds to e.g. copper against copper. In this section, we introduce a simple sensitivity measure $S_{a,b}^n(g)$ which we will use to assess a grasp’s stability under variations in friction. Furthermore, we devise a parametric approach for studying the sensitivity of generic grasps as well as grasps on specific objects. Finally, we describe a gradient based approach for synthesizing grasps robust under changes in friction coefficients and develop a new algorithm that can be used to determine force-closed grasps even for small friction coefficients.

3.1 Quantifying a grasp’s sensitivity to friction

To provide a computationally tractable first-order approximation of the average slope of the graph $\mu \mapsto Q_\mu(g)$, for $\mu \in [a,b]$ and for a fixed grasp $g$, we make the following definition:

**Definition 3.1.** Consider a grasp configuration $g$ and a friction interval $[a,b] \subset \mathbb{R}_{\geq 0}$. Fix $n \in \mathbb{N}$ and consider $\delta = \frac{1}{n}(b-a)$, $\{x_0,\ldots,x_n\} \in [a,b]$, $x_i = a + i\delta$ for $i \in \{0,\ldots,n\}$. We define the sensitivity $S_{a,b}^n(g)$ of $g$ with respect to the parameters $a, b, n$ to be:

$$S_{a,b}^n(g) = \frac{1}{n-m} \sum_{i=m}^{n-1} k_i,$$

where $k_i = \frac{1}{n}(Q_{x_{i+1}}(g) - Q_{x_i}(g))$ and $m$ is the smallest integer $i \in \{0,\ldots,n-1\}$ such that $Q(x_i)$ is not zero. If no such $m$ exists, we define $S_{a,b}^n(g) = 0$.

We then consider grasps with large $S_{a,b}^n(g)$ to be sensitive to changes in friction, while grasps with small $S_{a,b}^n(g)$ are considered to be insensitive to such variations.
3. METHODOLOGY

Suppose a robot has computed a set of grasp hypotheses \( H_\mu = \{g_1, \ldots, g_m\} \) of grasps \( g_i \) with underlying friction coefficient \( \mu > 0 \) and such that \( Q_\mu(g_i) > 0 \). While traditional ranking based approaches would select a grasp with largest grasp quality \( Q_\mu \), our definition of grasp sensitivity allows us to react to uncertainty in the friction coefficients. Returning to Fig. 1 we can compute that \( S_{0.2,1}^{\mu}(g_{\text{blue}}) \approx 0.1640 \) for the blue grasp, while \( S_{0.2,1}^{\mu}(g_{\text{red}}) \approx 0.2287 \) for the red grasp. If we are working under the assumption that \( \mu \approx 0.6 \), a ranking by \( Q_{0.6} \) now favours \( g_{\text{red}} \), while a ranking by \( S_{0.2,1}^{\mu} \) for grasps with \( Q_{0.6}(g) > 0 \) returns \( g_{\text{blue}} \), which indeed stays stable over the whole friction interval \([0, 2, 1]\).

To provide a simple measure balancing the benefits of large grasp quality with low friction sensitivity, we define

\[
\Phi_{a,b,\mu}^n(g) = \frac{Q_\mu(g)}{S_{a,b}^n(g)},
\]

which provides a simple scoring function for grasps. We propose that grasps with large \( \Phi_{a,b,\mu}^n(g) \) are desirable since they arise from a comparatively large grasp quality and low friction sensitivity.

3.2 Statistical properties of grasps and friction

Let us now describe how we shall study some of the basic statistical properties of grasp quality and friction sensitivity.

Generic random sampling

To study grasps with \( m \) contact points generically, that is without a notion of an object, we consider the set \( \mathcal{D}(r) = \mathbb{B}(r)^m \times (S^2)^m \times \{0\} \), where \( \mathbb{B}(r) = \{x \in \mathbb{R}^3 : \|x\| \leq r\} \) and \( S^2 = \{x \in \mathbb{R}^3 : \|x\| = 1\} \) and 0 denotes the origin in \( \mathbb{R}^3 \).

An element \( g = (c_1, \ldots, c_m, n_1, \ldots, n_m, z) \in \mathcal{D}(r) \) then represents a grasp with contacts \( c_i \), inward pointing unit contact-normals \( n_i \) and with centre of mass \( z \) at the origin, and where the contacts are constrained to lie in the ball \( \mathbb{B}(r) \) around the origin. Using the uniform probability distribution on \( \mathcal{D}(r) \), we can now produce an arbitrary number of random grasps in this set. We employed a similar approach of grasp sampling in our work [12]. The grasp quality \( Q_\mu \) and our sensitivity measure \( S_{a,b}^n \) can in this context be considered as random variables on this space whose properties we can study statistically. In our experiments, we will in particular show that a Dirichlet distribution provides a good fit to \( (Q_\mu(g), S_{a,b}^n(g)) \).

Random sampling and surfaces

To study grasps on an arbitrary surface \( S \) , we shall employ uniform random sampling on \( S \) (as in [12]) to obtain a set \( C = \{c_1, \ldots, c_l\} \) of contact points. We can then study the set of \( \binom{l}{m} \) tuples of distinct configurations of \( m \) such contact points as grasp candidates. Using this procedure, we shall then obtain information about the distribution of \( Q_\mu \) and \( S_{a,b}^n \) for a specific surface.
3.3 Synthesizing stable grasps with small friction coefficient

As we shall show, stable grasps are difficult to synthesize with sampling based approaches such as the ones used by GraspIT [10] if the friction coefficients are small (e.g. \( \mu \leq 0.5 \)). We hence propose a new procedure using ‘virtual’ friction coefficients. Suppose we have a parametric form for our graspable surface \( S \), so that points and unit normals on \( S \) are given by coordinates \((x, y) \in \mathbb{R}^2\) as \( c(x, y) \) and \( n(x, y) \) respectively. Since we would like to execute a gradient based method using \( Q_\mu \), we shall use a modified definition, where \( \hat{Q}_\mu(g) = \min_j \lambda_j \), rather than \( Q_\mu(g) = \max(0, \min_j \lambda_j) \), where \( \lambda_j \) are the offsets of the hyperplanes defining the wrench space \( S(g) \) which we mentioned in Section 2. The advantage of \( \hat{Q} \) here is that we can obtain numerical gradients even when \( \hat{Q} < 0 \), while \( Q \) just takes on the value zero in those regions. Note that \( \hat{Q}_\mu(g) = Q_\mu(g) \) when \( Q_\mu(g) > 0 \).

For grasps with \( m \) contacts, we then obtain a function \( F_\mu : \mathbb{R}^{2m} \rightarrow \mathbb{R} \) mapping \( m \) contact point coordinates to the grasp quality \( \hat{Q}_\mu(g) \) of the corresponding grasp at those contact points. We then proceed by iterating \( M \) steps of a standard gradient ascent of \( F_\mu \) with a small decrease in \( \mu \) until a desired target friction value \( \mu_{end} \) is reached. Alg. 1 provides details, where \( \text{GradientAscent}(F_\mu, g, \delta, M) \) executes \( M \) gradient ascent steps with step size \( \delta \) with respect to \( F_\mu \) and starting configuration \( g \). Here, gradients are approximated using finite differences. \( \text{SampleGrasp}(S) \) returns a uniformly sampled grasp configuration on the surface \( S \).

4 Evaluation

In the following, we describe an evaluation of our proposed methodology.

4.1 The impact of friction on grasp stability

Let us now study the impact of changes in friction coefficients on \( Q_\mu \). For this purpose, we sampled 10 sets of 10000 uniform samples, \( U_1, \ldots, U_{10} \), from the uniform distribution on \( D(2) \). Fig. 2 displays the mean percentage of stable grasps (i.e.
4. EVALUATION

$Q_{\mu}(g) > 0$ among the grasps in these sets $U_i$ against friction coefficients ranging from 0 to 5 in increments of 0.1.

![Graph showing the percentage of stable grasps for uniformly sampled grasps from $D(2)$.](image)

Figure 2: Percentage of stable grasps for uniformly sampled grasps from $D(2)$.

In this and all the following experiments, we used $l = 8$ edges to approximate the fiction cones used in the calculation of $Q_{\mu}$. Fig. 2 additionally indicates the standard deviation for our 10 sets of grasp samples $U_1, \ldots, U_{10}$. Observe that the percentage of stable grasps increases substantially as the friction coefficient is increased and that this percentage decays rapidly as $\mu$ tends to zero as can be seen in the second plot. A friction coefficient of $\mu = 0.2$ corresponds to the friction of polythene plastic against steel, while a friction of 1.0 corresponds to copper against copper. For $\mu = 0.2$, only about 0.084% of the grasps were stable, while for $\mu = 0.5$, 2.4% were stable and, for $\mu = 1.0$, about 16.2% of the grasps were stable.

Friction coefficients hence significantly influence the success of sampling based grasp synthesis algorithms such as [3]. While previous work has certainly been aware of this phenomenon, the above simple ‘generic’ sampling based approach provides us with a first quantitative analysis of this phenomenon which, to the best of our knowledge, has not previously been formalized in this way. Fig. 2 provides clear evidence that ‘straight-forward’ sampling approaches for grasp synthesis used e.g. by the popular simulation environment GraspIT [10], are inappropriate for low friction coefficients.
4.2 Friction sensitivity for generic grasps

Recall that, when $Q_\mu(g)$ of a grasp $g$ is relatively small for the expected friction coefficient $\mu$, a big $S^\mu_{\mu-\epsilon, \mu+\epsilon}(g)$, for $\epsilon > 0$, indicates that it may be inappropriate to use the grasp $g$ when we are uncertain about the exact value of $\mu$. Let us now investigate the relationship between friction sensitivity and grasp quality $Q_\mu$ for a generic set of grasps.

For this purpose, we sampled a set $W$ of 1 million grasps with three contact points uniformly from the set $D(2)$. We assume that the true underlying friction coefficient $\mu$ lies in the interval $[0, 2]$ with an expected value of 0.6, and we hence compute $Q_{0.6}(g)$ for all grasps $g \in W$. Let us consider the set of grasps $W' = \{g \in W : Q_{0.6}(g) > 0.001\}$. $W'$ contained 29236 stable grasps. For each $g \in W'$, we now compute an associated friction sensitivity $S^{20}_{0.2, 1.0}(g)$, using a partition of the interval $[0.2, 1.0]$ into 20 equally spaced sub-intervals.

Figure 3: The distribution of grasp quality $Q_{0.6}$ (horizontal axis) against sensitivity $S^{20}_{0.2, 1.0}$ (vertical axis) is displayed on the left and the mapping of this data onto the standard simplex in $\mathbb{R}^3$ is shown on the right.

Fig. 3 displays the distribution of $(Q_{0.6}, S^{20}_{0.2, 1.0})$ for our set of stable grasps $W'$. We will now study this distribution in more detail.

A parametric density estimate

Observe that the data in the left part of Fig. 3 is located in a cone with apex at the origin. We can see that grasps with low sensitivity and high grasp quality are sparse in this data-set. In order to be able to quantify statements about the likelihood of encountering grasps with prescribed grasp quality and friction sensitivity, we propose a parametric density fit as follows: as a first step, we determined edges $e_1, e_2$ of the smallest triangle in $\mathbb{R}^2$ enveloping all the samples and with apex at the origin. Both $e_1, e_2$ have one end-point at the origin and satisfy $\langle e_1 - e_2, d \rangle = 0$, where $d$ is the vector which equally divides $\angle(e_1, e_2)$. Moreover, the length of $e_1, e_2$ is chosen as small as possible, and such that the triangle still contains all the samples. In our case, $e_1 = (0.5756, 0.9094)$ and $e_2 = (0.0017, 1.0762)$. The triangle containing the edges $e_1, e_2$ is mapped to the standard 2-simplex $\Delta = \{(x_1, x_2, x_3) : x_1 \geq 0, x_1 + x_2 + x_3 = 1\}$ by an affine map mapping the origin to the vertex $(0, 0, 1) \in \Delta$. The right part of Fig. 3 displays the image of our data-points on $\Delta$. Given our transformed data points in $\Delta$, we determined a Dirichlet distribution fit to the data. Recall that a Dirichlet distribution $Dir(\alpha_1, \alpha_2, \alpha_3)$ on $\Delta$ is determined...
4. EVALUATION

by three concentration parameters $\alpha_i > 0$. We performed a maximum likelihood fit of the parameters to the data using the fastfit Matlab toolbox [11]. The estimated parameters were $(\alpha_1, \alpha_2, \alpha_3) = (1.0001, 2.2273, 9.8739)$.

Figure 4: Comparison between the fitted Dirichlet distribution and the observed data.

The surface plot in Fig. 4 shows the resulting density function of $Dir(1.0001, 2.2273, 9.8739)$ over the projection of $\Delta$ onto the $xy$ plane together with a standard histogram density estimator. As we can see in that figure, the chosen Dirichlet density provides a visually satisfying fit to the data following the histogram density estimate closely.

To further quantify the quality of the fit, we ran a Pearson $\chi^2$ test [4] to test the difference between the observed and expected frequencies. For this purpose, we used Mathematica’s Monte-Carlo-based $\chi^2$ testing function to evaluate the goodness of our fit and used a significance level of $\alpha = 0.05$. After repeating the test 10 times, the resulting average $p$-value was 0.833, indicating that our fit is of high quality.

The usefulness of our parametric fit lies in the fact that it provides a summary of the data enabling us to compute probabilities for the occurrence of samples in different regions of the quality/sensitivity parameter space.

Table II provides examples of computed probabilities for encountering grasps in selected parameter regions based on our Dirichlet distribution fit. The bracketed expressions in the table indicate the number of samples lying in those regions divided by the total number of samples. Since these are very close to the probabilities predicted by our Dirichlet distribution fit, this provides further assurance that the parametric representation can be used to calculate these probabilities without the knowledge of the full sample set.

4.3 Friction sensitivity for example objects

Having studied properties of grasp quality and friction sensitivity in a generic setting, we now concentrate on grasps on the four surfaces displayed in Fig. 5. For the purpose of this experiment, we assume a parametric representation of these surfaces allowing us to compute normals at each surface point $p$. We used four of
Table 1: Probabilities for encountering a grasp in the selected parameter regions in $(S_{0.2,1.0}, Q_{0.6})$ space for uniform samples in $\{g \in D(2) : Q_{0.6}(g) > 0.001\}$ determined using our Dirichlet distribution fit. The corresponding relative observed frequencies from our data-set are displayed in brackets below each such value.

<table>
<thead>
<tr>
<th>$Q_{0.6}$</th>
<th>$S_{0.2,1.0}^0(g) \leq 0.2$</th>
<th>$S_{0.2,1.0}^0(g) \leq 0.4$</th>
<th>$S_{0.2,1.0}^0(g) \leq 0.6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\geq 0.02$</td>
<td>0.1003</td>
<td>0.4317</td>
<td>0.6002</td>
</tr>
<tr>
<td></td>
<td>(0.1127)</td>
<td>(0.4576)</td>
<td>(0.6109)</td>
</tr>
<tr>
<td>$\geq 0.05$</td>
<td>0.0196</td>
<td>0.2097</td>
<td>0.2913</td>
</tr>
<tr>
<td></td>
<td>(0.0189)</td>
<td>(0.1994)</td>
<td>(0.3062)</td>
</tr>
<tr>
<td>$\geq 0.10$</td>
<td>0.0001</td>
<td>0.0352</td>
<td>0.0923</td>
</tr>
<tr>
<td></td>
<td>(0.00007)</td>
<td>(0.0379)</td>
<td>(0.0902)</td>
</tr>
<tr>
<td>$\geq 0.15$</td>
<td>0</td>
<td>0.0042</td>
<td>0.0203</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>(0.0046)</td>
<td>(0.0236)</td>
</tr>
</tbody>
</table>

the surfaces studied in [12] and followed the same uniform contact point sampling procedure as outlined in that paper.

In particular, to study the surface-specific distributions of grasp quality and friction sensitivity, we sampled 100 contact points $C$ uniformly from each of these surfaces and computed the resulting $\binom{100}{3} = 161700$ distinct grasps with three.
4. EVALUATION

contact points chosen from $C$. Fig. 5 displays the resulting distributions for each of the surfaces depicted in Fig. 5 analogously to the generic case in Fig. 3. Observe that, while the general concentration of the grasp qualities and sensitivities towards the origin remains a dominant feature, we can observe object-specific properties in these distributions such as the sparse fringes for the box object and a stronger concentration towards the origin for the top left and bottom right object. Fig. 5 additionally displays sample grasps which corresponds to the respective red points in Fig. 6. These initial investigations provide evidence that such distributions in terms of grasp quality and friction sensitivity could be used also for the classification of the graspability of various objects under varying friction assumptions.

Figure 6: Distributions of $Q_{0.6}$ (horizontal axis) and $S_{0.2,1}$ (vertical axis) for random grasps on the 4 surfaces displayed in Fig. 5 are shown in the same order as in that figure. The red dots correspond to the example grasps in Fig. 5 respectively.

4.4 Gradient ascent on $\Phi_{a,b,\mu}^n(g)$

We now experimentally verify that, for any grasp $g$ with $Q_{\mu}(g) > 0$, a simple fixed step-size gradient ascent can dramatically improve the value of $\Phi_{a,b,\mu}^n(g)$ and hence result in a more desirable grasp. In the following, unless specified elsewhere, we set $a = 0.2$, $b = 1.0$, $n = 20$ and $\mu = 0.6$. Analogously to the proposed algorithm Alg. 1, we consider a parametric surface representation $\varphi : \mathbb{R}^2 \rightarrow S$ of our object $S$ and perform gradient ascent of the function $H_{a,b,\mu}^n$ sending a grasp $g$ specified by the centre of mass $z$ of the surface and a triple of surface contact point coordinates to the resulting value of $\Phi_{a,b,\mu}^n(g)$.

We consider the bottle and the box surface depicted in the left column of Fig. 5. The grey points in Fig. 7 display the distribution of grasp quality and sensitivity values for the bottle and the box surface which we computed previously and which are also displayed in Fig. 6. We divide the parameter region $[0.01, 0.285] \times [0.01, 0.385]$
into uniformly spaced boxes of size $0.025 \times 0.025$ and picked a grasp from the grey sample points for each non-empty box. This results in a set of grasps $G$ for each of the two surfaces. $G$ is depicted by blue dots in Fig. 7. We then apply 200 steps of a standard fixed step-size gradient ascent with respect to $H^n_{a,b,\mu}$ for every grasp in $G$ and compute gradients numerically using small finite differences.

Fig. 7 shows the result of this gradient ascent on both the bottle and the box surface by green points. We can see that almost all the blue dots in Fig. 7 have moved towards the right edge of the distribution cone, indicating an improvement in $\Phi^n_{a,b,\mu}(g)$. Fig. 8 illustrates the performance of the gradient ascent using bar-plots, with black and blue bars showing $\Phi^n_{a,b,\mu}(g)$ values before and after gradient ascent respectively. It is worth mentioning that, looking at Fig. 8, the final value of $\Phi^n_{a,b,\mu}$ seems to be bounded by similar upper bounds, both for the bottle and the box surface.

Fig. 9 displays two examples of gradient ascent on both surfaces and corresponding to the red dots in Fig. 7. The trajectory on the object surface represents the location of contacts in each iteration of the gradient ascent. Note that, if we imagine the bottle to be wet or slippery, the red grasp is intuitively less stable than the blue grasp, which is confirmed by the graph of the grasp quality depicted next to the bottle.
4. **EVALUATION**

Figure 8: Results of the gradient ascent on $H_{n,a,b,\mu}$ on the bottle (top) and the box (bottom) surface. Each bar represents a grasp sample shown in Fig. 7. Bars are sorted in ascending order of the final $\Phi_{n,a,b,\mu}(g)$ values which is depicted along the vertical axis. Black bars depict $\Phi_{n,a,b,\mu}$ values of the original grasp samples and blue bars are values after gradient ascent.

Figure 9: An example of gradient ascent for $\Phi_{n,a,b,\mu}$ corresponding to the red points in Fig. 7. The red grasps converge to the blue ones under our gradient ascent. The trajectories are depicted as faint lines.
4.5 An evaluation of Algorithm 1

We now come to an evaluation of Alg. 1 which we proposed in order to synthesize force-closed contact configurations on surfaces with low friction coefficients. Again, we consider the bottle and box surfaces displayed in Fig. 5.

![Figure 10: Percentages of stable grasps for each of the 10 runs of our experiment with 120 grasps per experiment for the bottle (top) and box (bottom) surfaces and for $\mu_{\text{start}} = 1$ and $\mu_{\text{end}} = 0.2$. In red, we display the percentages of stable grasps for the original random grasps, in green, the percentages of stable grasps after a simple gradient ascent of $F_{\mu_{\text{end}}}$ and, in blue, the percentage of stable grasps synthesized using Alg. 1.

We sampled 10 contact points uniformly on these surfaces and computed all $\binom{10}{3} = 120$ distinct 3-contact grasp configurations for these contacts, resulting in a grasp set $G$ for each surface. Next, we studied the effectiveness of Alg. 1 for these grasps, setting $\mu_{\text{start}} = 1.0$, $\mu_{\text{end}} = 0.2$ and descending from $\mu_{\text{start}}$ to $\mu_{\text{end}}$ in $N = 50$ steps and using a gradient ascent with step size $\delta = 0.05$ and $M = 40$ steps per iteration according to Alg. 1. We repeated this experiment 10 times for each surface, resulting in the percentages of stable grasps ($Q_{\mu_{\text{end}}}(g) > 0$) depicted by blue dots in Fig. 10.

To compare our result to a more straightforward approach, we tested the alternative approach of simply performing a gradient ascent of $F_{\mu_{\text{end}}}$ for each grasp, and with step-size $\delta = 0.05$ and for $M = 200$ iterations which resulted in the much lower percentages of stable grasps depicted in green. If we simply use a sampling based approach and evaluate the grasp quality for each grasp in our set with friction $\mu_{\text{end}}$, almost none of the sampled grasps had positive grasp quality as indicated by red dots. Our results hence indicate that Alg. 1 can be used to successfully synthesize stable grasp configurations on objects with low friction coefficients by repeating the
algorithm a few times with different random starting grasps until a stable grasp is found.

5 Conclusion

Studying grasping under uncertainty is an important area in robotics [7, 16, 13, 15, 9]. While most current state of the art approaches concentrate on aspects of imperfect object or robot models, we studied another fundamental problem in grasp synthesis in this work: the dependence of grasp stability on friction coefficients. We believe that this is an important problem when robots are to operate in open-ended environments with changing conditions.

We have in particular studied the statistics of stable grasps under changes in friction coefficients and have introduced the notion of friction sensitivity measuring the susceptibility of a grasp’s quality to changes in friction. Furthermore, we have proposed and evaluated two gradient ascent algorithms for synthesizing force-closed contact configurations on parametric surfaces with potentially low friction and for the synthesis of stable grasps with low friction sensitivity.

In our future work, we would like to evaluate our approach with a real robot and study changes in friction coefficients in a real application such as a household robot cleaning dishes which might be dirty or wet, impacting heavily on the resulting friction properties. Further directions might also include the study of alternative friction models and grasp quality scoring functions.

References


Paper B

Combinatorial Optimization for Hierarchical Contact-level Grasping

Published in
*IEEE International Conference on Robotics and Automation, 2014*
Combinatorial Optimization for Hierarchical Contact-level Grasping

Kaiyu Hang, Johannes A. Stork, Florian T. Pokorny and Danica Kragic

Abstract

We address the problem of generating force-closed point contact grasps on complex surfaces and model it as a combinatorial optimization problem. Using a multilevel refinement metaheuristic, we maximize the quality of a grasp subject to a reachability constraint by recursively forming a hierarchy of increasingly coarser optimization problems. A grasp is initialized at the top of the hierarchy and then locally refined until convergence at each level. Our approach efficiently addresses the high dimensional problem of synthesizing stable point contact grasps while resulting in stable grasps from arbitrary initial configurations. Compared to a sampling-based approach, our method yields grasps with higher grasp quality. Empirical results are presented for a set of different objects. We investigate the number of levels in the hierarchy, the computational complexity, and the performance relative to a random sampling baseline approach.

1 Introduction

The synthesis of feasible and stable grasps on an object remains an important problem in robotics which involves sensory perception and representation [21, 28, 18], the encoding of task constraints [2], and high dimensional configuration spaces [14]. The planning and execution of stable grasps with point contacts on complex object shapes is a particular aspect of this larger problem for which no efficient complete solutions have been found yet. In this work, we propose an efficient method that generates and locally optimizes stable force-closed grasps. Our method can be seen as complementary to state-of-the art heuristic approaches such as [5, 23, 7, 11, 35, 1, 22, 18].

Most robot object interactions, static or dynamic in nature, require that an object is firmly held. The force and contact position based grasp description [13, 6] is currently the most mature approach available to formalize this concept. In this work, we consider point contact dependent aspects of the grasp synthesis problem and mostly neglect other issues such as physical constraints and kinematic limitations. We furthermore utilize a simplified concept of reachability which our system
Figure 1: System outline: Our approach applies the multilevel refinement paradigm to point contact grasps generation. We recursively simplify a mesh representation (1) and find locally optimal grasps at each level (3), starting at the topmost level (2). Solutions are transferred to the level below by finding similar faces (4). The process is sketched for one contact point only.

integrates with a notion of grasp quality. We frame the search for force-closed grasps on 3D mesh objects as a combinatorial optimization problem and employ an iterative abstraction and refinement approach. Following the generic multilevel refinement paradigm, our approach creates a hierarchy of approximations to the grasp search problem on the original object. For this purpose, we recursively simplify the original mesh representation, resulting in a set of object models of varying resolution. We then consider the midpoint of each triangle face as a possible contact position. At each level of the hierarchy, an initial grasp is iteratively refined by local search and finally extended to the level below by finding a similar face in the higher resolution mesh. This process is exemplified in Fig. 1. The main contributions of this work can be summarized as follows:

- We introduce an approach operating directly on an input mesh, without the need for further high-level features [28, 1].

- Our approach can process complicated object geometry as well as complex and high resolution meshes consisting of thousands of faces to produce high quality grasps.

- We do not require that the input data has the same resolution at all parts of the object—a potential we leave for future exploration.
2. RELATED WORK

- The multilevel approach iteratively simplifies the objective function resulting in fast convergence. Furthermore, the local search at each level effectively reduces the total number of evaluated candidate solutions to a practical amount.

2. Related Work

The synthesis of stable grasps on shapes of high complexity and with exact contact positions is a challenging problem. Research has addressed grasp synthesis in various ways which can be summarized into two main groups: Firstly, by developing compact and expressive object representations to deal with the underlying complexity of the problem space \cite{31, 29, 1, 18}, and secondly, by devising methods for analyzing and evaluating grasp quality \cite{13, 8, 26} - in each case possibly taking into account constraints related to the task or embodiment. In this section, we discuss the two areas in relation to our contributions.

2.1 Object Representations for Grasp Synthesis

Object representations for grasp synthesis are often formed by extracting either global or local features from an object description. Using local features such as surface geometry properties, it is possible to evaluate grasp stability \cite{8} and sensitivity \cite{17}, or to compute independent contact regions \cite{30}. By exploring global features of the target object in the form of skeletal or topological descriptors, one can furthermore synthesize grasps by sampling grasp parameters such as pre-shapes, positions and approach directions of a robot hand. The reported results using these methods show relatively good success rates for obtaining feasible grasps \cite{1, 12}. However, many of these approaches need to execute a grasp policy \cite{5, 23} in simulation to subsequently conduct a force-analysis of the resulting grasp configuration. Therefore they can be regarded as a form of heuristic to generate point contact grasps.

Many proposed object representations extract skeletal features from 3D data \cite{29, 1}. The Reeb Graph \cite{1} and Medial Axis \cite{12}, were successfully used for generating initial positions, approach directions and pre-shapes for whole hand grasps \cite{35, 1}. Similarly, the work of \cite{27} detects holes on objects as a basis for executing caging grasps. Approximating objects’ shape using basic primitives has been recognized as another object representation \cite{22, 18}, where approach vectors and pre-shapes are generated by sampling on shape primitives. By parameterizing objects using superquadrics \cite{7, 11, 5} and superellipsoids \cite{24}, grasp parameters are sampled on the approximated objects.

Our approach relates to the above methods relying on local features since it is based on grasp quality evaluation and requires information about the surface and its normals. It is also related to methods using global features since recursively coarsening the mesh model results in an increasingly rougher object description, maintaining only global characteristics. In contrast to global feature methods, our approach optimizes grasp contacts directly without the detour of describing the grasp by grasp parameters. As a consequence, we do not require the execution of
a grasp policy \cite{5,22} in simulation to get access to a candidate solution. Additionally, as we provide exact grasping contact positions which are required in some applications such as finger gaiting \cite{32,16}, our method can act as a plug-in in such applications to complete the grasp synthesis step in a pipeline.

2.2 Grasp Contact Synthesis

Sampling-based grasp synthesis methods, as discussed above, are nonconstructive in the sense that they assess grasp stability based on contacts resulting from the execution of a grasp policy, but they do not explicitly relate the synthesis to the quality measure and solve the grasp synthesis problem indirectly. In \cite{8}, it was shown that randomized grasp generation is fast and suitable for some simple test objects. However, this is not directly applicable for objects of complex shapes as we shall show later in this paper. As also have shown in \cite{17}, sampling based methods furthermore suffer in the case of low friction coefficients, where stable grasps become increasingly sparse.

In the work of \cite{25}, a local gradient based grasp quality optimization was performed and grasps were also transferred between objects by a continuous shape/grasp deformation and optimization process. However, this approach required a particular smooth surface parametrization. In our work, the grasp configuration space is instead discretized according to a hierarchically coarsened mesh representation of the object surface. This enables an optimization of contact based grasps by a discrete local search and following a multilevel combinatorial optimization approach.

3 Preliminaries

We start by presenting a combinatorial optimization problem formalism and continue with a discussion of grasp stability and reachability metrics.

3.1 Combinatorial Optimization and Multilevel Refinement

Presented with a combinatorial optimization problem, where an optimum over a discrete set of candidate solutions needs to be determined, it is often infeasible to carry out an exhaustive search to determine the global optimum. In this case, the problem can be relaxed to finding a good solution in reasonable time by applying a search heuristic. The multilevel refinement paradigm is a metaheuristic that can be applied to difficult combinatorial optimization problems and describes a recursive abstraction and refinement pattern. It has been used to address a variety of problems in graph theory and numerical algebra as described in the overview work \cite{9,33,37}.

As a metaheuristic, the multilevel refinement paradigm describes a general search strategy: a problem instance is recursively approximated to form a hierarchy of increasingly approximated problem instances. Starting at the top level, a solution is found for the current instance and then extended to the level below.
Provided with an initial candidate solution on the top level, the procedure finally results in a solution to the original problem defined in the lowest level.

Formally, we denote the initial problem instance by $P_0$, its set of candidate solutions by $X_0$, and the objective function for all levels by $\theta$. We write $P_0, P_1, \ldots$ for the problem hierarchy, where each instance $P_l$ is created recursively by a coarsening of its parent $P_{l-1}$. At level $l$, the initial solution $C^0_l$ is refined to $C_l$, where on the top level $C^0_0$ is provided by an initialization algorithm and for all other levels by extending $C_{l+1} \in X_{l+1}$ from the child problem. A detailed description of the approach is reported in [37] and our implementation is described in Alg. 2.

\begin{algorithm}
\begin{algorithmic}
  \State \textbf{Input:} Problem instance $P_0$
  \State \textbf{Output:} Solution $C_0$
  \State $l \leftarrow 0$
  \While {coarsening}
    \State $P_{l+1} = \text{coarsen}(P_l)$
    \State $l \leftarrow l + 1$
  \EndWhile
  \State $C^0_l = \text{initialize}(P_l)$
  \State $C_l = \text{refine}(C^0_l, P_l)$
  \While {$l > 0$}
    \State $l \leftarrow l - 1$
    \State $C^0_l = \text{extend}(C_{l+1}, P_l)$
    \State $C_l = \text{refine}(C^0_l, P_l)$
  \EndWhile
\end{algorithmic}
\caption{Description of the multilevel refinement paradigm metaheuristic for combinatorial optimization.}
\end{algorithm}

While the use of sophisticated techniques for refinement have been demonstrated [36, 19, 34, 20], the refinement algorithm is often selected from a family of local search heuristics. There, only a small region around the current solution is explored for improvement. This notion is formalized by the neighborhood set $N(x) \subseteq X_l$ of a candidate solution $x \in X_l$ which defines the locally searched space.

For application, the paradigm requires problem specific definitions of algorithms for \textit{refinement}, problem \textit{coarsening}, \textit{initialization}, and solution \textit{extension}. We describe the objective function, candidate solution space and the algorithms for our grasp synthesis problem in Sec. 4.

\subsection*{3.2 Grasp Stability Metric}

Many approaches to robotic grasping of rigid objects are based on force analysis and in particular on the concept of force-closure [13, 6]. To identify stable grasp configurations, the forces exerted by the robot end effector and friction between the robot hand and the object surface are considered. We choose to evaluate the
force-closure property of a grasp with the $L^1$ grasp quality measure $Q_\mu$ reported in \cite{8} and which is based on the Coulomb friction model.

To this end, a grasp $g$ is formalized by $m$ point contacts $p_1, p_2, \ldots, p_m \in \mathbb{R}^3$ and their inward-pointing unit surface normals $n_1, n_2, \ldots, n_m \in \mathbb{R}^3$. The grasp quality $Q_\mu$ is then a function of all contact positions and normals, the center of mass of the object $z \in \mathbb{R}^3$, and the friction coefficient $\mu \in \mathbb{R}^+$. A grasp is force-closed if $Q_\mu$ is larger than zero. In the following, the relation between a grasp $g$ and $z, n_i, p_i$ will be implicit in cases where this does not cause confusion.

### 3.3 Reachability Measure

It is possible that a given set of contacts on an object resemble a force-closed grasp, but that the robot hand is incapable of realizing this grasp. The reasons for this are several: kinematic infeasibility of the hand pose, object or self-collisions, or the required number of contact points cannot be achieved due to lack of independent joints. These issues are in part addressed in recent work \cite{38}.

In the simplest case, the individual point contacts are just too distant from each other even when kinematics and collisions are neglected. We adopt this simplified distance-based concept of reachability to approximate the robot hand’s workspace. Formally, we define reachability $R(g)$ of a point contact grasp $g$, given by its contact positions as described in Sec. 3.2, as

$$R(g) = \frac{1}{m} \sum_{i=1}^{m} \|p_i - \psi\|$$  \hspace{1cm} (1)

where $\psi$ is the centroid of all contact points $p_1, p_2, \ldots, p_m$ of the grasp $g$ and $\|\cdot\|$ is the Euclidean distance. We consider a grasp as reachable if $R(g) \leq r$ for a predefined reachability upper bound $r \in \mathbb{R}^+$. Intuitively, the pairwise distances are limited by $2r$.

### 4 Methodology

In this section, we describe our approach to search for grasp contacts. In Sec. 4.1, we formalize the grasp search problem on a surface mesh using the terminology of combinatorial optimization introduced in Sec. 3. Subsequently we explain our realization of the multilevel paradigm algorithm.

#### 4.1 Problem Formalization

To search for a stable force-closed point contact grasp on a rigid object, we represent the object’s surface by a mesh of oriented triangle faces $F = \{f_1, f_2, \ldots, f_k\}$. A triangle mesh is the simplest piece-wise linear approximation of the original surface, assuming that the vertices originated from the original surface. Instead of considering the space of all possible contact positions on all the faces, we only consider...
face midpoints $\Delta = \{\delta_1, \delta_2, \ldots, \delta_k\}$ as possible contacts. The normal for contact point $\delta_i$ is the face normal $n_i$. To simplify the notation, we will refer to faces $f_i$ and midpoints $\delta_i$ indiscriminately and let each of the symbols $f_i$, $\delta_i$, or $n_i$ refer to any subset of $\{f_i, \delta_i, n_i\}$ where the context is clear.

The previously introduced discretization induces a combinatorial search space $F = \prod_{k=1}^{m} F$ where each of the $m$ contacts has to be assigned one face or midpoint. To complete the combinatorial optimization problem, we define an objective function $\theta: F \rightarrow \mathbb{R}$ to evaluate candidate solutions by employing the grasp quality measure $Q_\mu$ described in Sec. 3.2 and the reachability measure $R$ defined in Eq. (1):

$$
\theta(g) = \begin{cases} 
Q_\mu(g) - (e^{\alpha (R(g) - r)} - 1), & R(g) > r \\
Q_\mu(g), & \text{else}
\end{cases}
$$

where $\alpha \in \mathbb{R}^+$ is a penalty factor. If the contacts are reachable, $\theta$ only describes grasp stability. Unreachable grasps with $(R(g) - r) > 0$ are increasingly dominated by the negative right-hand term. The further apart the contacts are positioned in space, the lower is the value of $\theta(g)$.

Figure 2: Sketch of the objective function values with large reachability bound if two contacts are kept fixed. Red color marks high quality. For each contact there are multiple local maxima rendering the global joint optimization problem difficult. However, the objective function changes gradually, allowing for improvement by local search.

Some properties of the search space $F$ and the objective function $\theta$ can be inspected in Fig. 2 for three contact grasps. Faces are colored by the objective function while iterating one contact over all faces and keeping the other contacts fixed. Even if only a single contact would be optimized at a time, the search space exhibits multiple local maxima at different places (red) rendering global joint optimization a difficult problem. However, the objective function changes gradually from face to face which allows for exploitation with local search and multilevel refinement.

Applying the terms introduced in Sec. 3.1, the search space of the initial problem instance $P_0$ is identified with the Cartesian product of the original mesh faces $X_0 = F$. Writing $F_l$ and $F_i$, we will use subscript to refer to different levels. A problem instance $P_l$ is coarsened to $P_{l+1}$ by reducing the number of faces in $F_l$ using a mesh simplifying procedure described in Sec. 4.2. We stop coarsening the
mesh before it degenerates and then apply a random initialization of contacts. For the refinement step, we apply a greedy hill climbing procedure explained in Sec. 4.3 using the Cartesian product of the $\eta$-order neighbor faces of a contact to form the search neighborhood. A level $l$ candidate solution $C_l = (f_{i_k})_{k=1}^n \in F_l$ is extended to level $l-1$ by matching $f_{i_l} \in C_l$ to a similar face in $F_{l-1}$, as explained in Sec. 4.4.

### 4.2 Mesh Simplification

For the coarsening procedure, we automatically produce simpler and approximated versions of mesh models $F_l$ using the surface simplification algorithm of [15]. The algorithm was suggested for multi-resolution modeling and iteratively produces high quality approximations of polygonal models at a decreasing level of detail. Each simplification step joins one pair of vertices which is selected according to a per-vertex surface error metric. In the contraction, all incident edges are connected to the remaining vertex, degenerated edges and faces are removed, and the vertex is placed to minimize a quadric surface error metric. This procedure ensures high fidelity to the original model in every simplification step.

Since the algorithm removes only one vertex per iteration, a sequence of models with only gradually decreasing level of detail is produced in place. We apply a percentage reduction of the number of faces as termination condition for each recursive coarsening step. Thereby, we decide the number of levels $u \in \mathbb{N}$ and final number of faces $o \in \mathbb{N}$ beforehand:

$$|F_0| \gamma^u = o \quad \text{and} \quad |F_l| \gamma = |F_{l+1}|$$

In this way, we obtain a sequence of models having the same ratio of $1-\gamma$ less vertices with respect to the previous one. An example of such a sequence of mesh models is depicted in Fig. 1.

### 4.3 Hill Climbing

For the refinement procedure, we turn to local search and employ hill climbing until convergence. Given an initial candidate solution $C_0^0 = (f_{i_k})_{k=1}^m \in F_0$, we define the neighborhood of $C_0^0$ as the Cartesian product of the $\eta$-order neighboring faces for each $f_i \in C_0^0$. Formally, we write

$$N(C_0^0) = \prod_{k=1}^m N^\eta(f_{i_k}),$$

where $N^\eta(f_i)$ denotes the set of triangles in the mesh $F_l$ which can be reached from $f_i$ by $\eta$ or less steps, and including $f_i$ itself. An example of neighbor faces at different step distances are shown in the left of figure Fig. 3.

In each hill climbing iteration, we select the best grasp from $N(C_l)$ until no improvement is achieved. This procedure is formalized in Alg. 3. We maintain a
look-up table for already calculated grasps to reduce time complexity. In experi-
ments, we have found that it is particularly beneficial to increase $\eta$ from 1 to 2
when the current grasp is unstable. This technique is later referred to as an *adaptive*
approach.

| Input: Initial grasp $C_l^0 = (f_{ik})_{k=1}^m \in \mathcal{F}_l$ |
| Output: Solution $C_l$ |
| $i \leftarrow 0$ |
| repeat |
| $N = N(C_l^i)$ |
| $i \leftarrow i + 1$ |
| $C_l^i = \text{argmax}_{c \in N} (\theta (c))$ |
| until $C_l^i = C_l^{i-1}$ |
| $C_l \leftarrow C_l^i$ |

**Algorithm 3** Description of the refinement procedure using hill climbing until convergence.

4.4 **Face Matching**

The *extension* procedure in our approach translates a grasp $C_l = (f_{ik})_{k=1}^m$ from
the mesh $F_l$ to the mesh $F_{l-1}$. Since the objective function heavily depends on the
contact positions and contact normals, we match every face $f_i \in C_l$ with a face
in $F_{l-1}$ that has similar midpoint and normal. For this, we introduce a distance
function between pairs of faces. First, we translate and scale each mesh $F_l$ such
that the centroid is in the origin and each vertex is at most at distance 1 from the
origin. This leads to normalized midpoint positions $\bar{\delta}_i$. A face $f_i \in F_l$ is matched with a face $f \in F_{l-1}$ according to the following

$$f = \arg\min_{f_k \in F_{l-1}} \left( \| (\bar{\delta}_i, n_i) - (\bar{\delta}_k, n_k) \| \right),$$

(5)

where $(\bar{\delta}_i, n_i) \in \mathbb{R}^6$. Equation Eq. (5) can be efficiently implemented using a $k$-d tree data structure [4]. Fig. 3 shows how a face on a coarser mesh (red) is matched to a face (blue) on a finer mesh on the right side.

5 Experiments

In this section, we present empirical evidence for the viability of our approach. For all experiments, the Coulomb friction coefficient is set to $\mu = 1$ and the penalty factor in the objective function is $\alpha = 4$. The models bunny, plane, dinopet, and homer are found in [3, 10] and have 52,000, 17,354, 8,996, and 10,202 faces respectively. Each model was centered and scaled so that the maximum distance between any vertex and the origin is 1. Unless stated otherwise, the top level always has 100 faces, we use 4 coarsening levels resulting in 5 levels in total, and the number of contacts is $m = 3$. Figures 5, 7, 9, and 11 show standard box-plots indicating median, 25th and 75th percentiles and individual outliers.

5.1 Varying the Number of Levels

First, we investigate how the number of recursive coarsening steps influences the quality and reachability of the resulting grasp. For each model, we consider 0 to 4, and 6, 8, and 10 coarsening steps with 100 faces at the top level. We generate 100 random initial grasps on the coarsest mesh and execute our approach for each of these 8 settings. The reachability bound is always set to $r = 0.5$ and only the 1st-order neighborhood is considered.

The ratio of reachable grasps is described in Fig. 4. A maximum is generally reached for level 6 or 8, but in each setup at least 73% of the grasps were reachable. For the models bunny, plane, and homer, the percentage of reachable grasps was over 90% for more than 2 coarsening levels. Since the initialization was done randomly and our approach performs greedy optimization, it was not always possible to move the contacts close enough to satisfy the reachability constraint, particularly for the highly concave object dinopet.

The distribution of grasp quality $Q_\mu$ for reachable grasps can be inspected in Fig. 5. We observe a general trend showing that grasp quality is increased and variance reduced as the number of levels is increased. However, one recursive coarsening step already improves results considerably over mere hill climbing on the original mesh at level 0.
5. EXPERIMENTS

5.2 Empirical Complexity

Since the experiment in the previous section shows that more levels in general lead to better expected grasp quality, we have to investigate how the average and worst case complexity relate to the number of levels. For this, we consider how many hill climbing steps the approach requires at each level and how many grasps need to be compared in each of these steps. The presented data is taken from the above experiment and for the bunny object.

The average and maximal number of steps per level in relation to the number of coarsening steps is depicted in Fig. 6. When we initialized the algorithm with 4 or more levels, the maximal number of steps per level were similar and initialization with 0–3 levels required substantially more steps. For 2 or more initialization levels, the average number of steps per level stabilized between 2.5 and 3.1, and 3.4 steps were required in the final level. However, the number of grasp quality evaluations per hill climbing step depends on the cardinality of the set of neighbor faces. A statistic for the 1st-order neighbors on the bunny model for different coarsening levels is shown in Fig. 7. In our experiment, the average number of 1st-order neighbors did not change for different levels and so the complexity of each hill climbing step is not influenced by the number of levels.

5.3 Complexity of Objects

Intuitively, grasp synthesis is easier on simpler objects. We investigate this claim in Fig. 8 where grasp quality distributions for objects of varying complexity are displayed. The setup of this experiment is the same as in Sec. 5.1 except that the reachability bound was set to $r = 1.0$ and only 4 coarsening levels were used. In our
Figure 5: Grasp quality distribution for the reachable grasps from 100 runs with random initialization. Generally, grasp quality improves with the number of levels used.

<table>
<thead>
<tr>
<th>Initial Level</th>
<th>Max. #Steps</th>
<th>Avg. #Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 2 3 4 6</td>
<td>0 1 2 3 4 6</td>
</tr>
<tr>
<td>10</td>
<td>5 5 4 5 7 5 6</td>
<td>2.6 2.6 2.5 2.5 2.7 2.1 2.4 3.4</td>
</tr>
<tr>
<td>8</td>
<td>6 6 5 6 6 6 4 6</td>
<td>3.0 2.8 2.2 3.1 2.9 2.6 3.4</td>
</tr>
<tr>
<td>6</td>
<td>4 4 5 6 5 6</td>
<td>2.6 2.7 2.6 2.5 2.5 3.4</td>
</tr>
<tr>
<td>4</td>
<td>6 5 7 5 6</td>
<td>3.0 3.0 3.0 2.6 3.4</td>
</tr>
<tr>
<td>3</td>
<td>7 10 8 6</td>
<td>3.0 3.4 2.8 3.4</td>
</tr>
<tr>
<td>2</td>
<td>6 7 6</td>
<td>3.3 3.1 3.4</td>
</tr>
<tr>
<td>1</td>
<td>16 6</td>
<td>4.7 3.4</td>
</tr>
<tr>
<td>0</td>
<td>38</td>
<td>10.6</td>
</tr>
</tbody>
</table>

Figure 6: Maximum and average number of hill climbing steps sorted by initial level and for the experiment in Sec. 5.1 for the bunny model.

The grasp quality variance increases with the complexity of the object, since the number of local optima increases with the complexity of the object.
5. EXPERIMENTS

Figure 7: Statistics of the number of 1st order neighbor faces for the bunny model for different coarsening levels. The average number of faces is stable and so the complexity of one hill climbing step does not depend on the number of levels.

Figure 8: Grasp quality distribution for objects of varying complexity.

5.4 Neighborhood Orders

In the previous experiment, we saw that the level number does not dramatically influence the average amount of 1st-order neighboring faces which governs the complexity of a hill climbing step as described in Eq. (4). According to Sec. 4.3 it is possible to consider a larger neighborhood for local search or even to adapt the order of the neighborhood dynamically. We now investigate the influence of the size of the neighborhood on the final grasp quality and on the number of grasp quality evaluations. We generate 100 random initial grasps for the plane model and compare the results for different sizes of neighborhoods. The reachability bound is set to a small value $\tau = 0.133$ and we compare 1st-order 2nd-order, 3rd-order, and 4th-order neighborhoods, as well the adaptive approach mentioned in Sec. 4.3 that
increases $\eta$ from 1 to 2 if the current grasp is not force-closed.

Figure 9: Comparison of the final grasp quality and number of grasp quality evaluations when the hill climbing procedure considered only 1st-order, adaptively 1st-order, 2nd-order, 3rd-order, or 4th-order neighborhoods.

The results presented in Fig. 9 show that the low reachability bound leads to many unstable grasps for the 1st-order approach. Variance in grasp quality reduces if more than the 1st-order neighborhood is considered. This means that only considering 1st-order neighborhoods makes the method more sensitive to local maxima. Using larger neighborhoods comes at a cost in form of an increased number of grasp quality evaluations. However, the number of grasp quality evaluations is not dramatically increased, since – when a larger neighborhood is considered – the algorithm tends to converge much quicker and requires less optimization steps at each level.

5.5 Comparison to Random Sampling

From the previous experiment, we know that our approach requires less than 40,000 grasp quality evaluations on average on the objects considered. We now compare the results of our adaptive method using random initialization to a batch sampling approach where the best of 50,000 reachable randomly sampled grasps from $F_0$ is selected in each batch. We set a small reachability bound $r = 0.133$ and executed 100 runs each for both methods on the model plane.

Grasp quality results are presented in Fig. 10 where it can be clearly seen that the best grasps of each batch were on average worse than the resulting grasps from our approach. The low overall value of grasp quality is to be attributed to the small reachability bound which reduces the quality measure. While our approach stayed below one minute total execution time, the sampling-based approach consumed 85s in our implementation.
5. EXPERIMENTS

Figure 10: Histogram of maximal grasp quality from randomly sampled reachable face midpoints on the original mesh (green) and grasp quality when our method with random initialization, adaptive neighborhood size and 4 coarsening steps (blue) is used on the plane model.

5.6 Heuristic Initialization

In all previous experiments, we used a random initialization for our approach. However, our approach is designed to improve grasp quality and therefore can be complementary to any heuristic that generates initial grasp contacts. In this experiment, we compare the performance of our approach to a simplified automatic fingertip closing grasp policy and initialize both using the same heuristic.

Figure 11: Comparison between automatic closing of fingertips and our approach when both are initialized with a heuristic. The used mesh models are bunny (A), plane (B), dinopet (D), and homer (D). Fingertip closing is indicated by (1) and our approach is marked with (2).

The initialization heuristic used, considers the same set of vectors $(\delta_i, n_i) \in \mathbb{R}^6$ as described in Sec. 4.4. We cluster the data $\{(\delta_i, n_i) \mid f_i \in F_4\}$ into 6 sets.
Figure 12: A couple of examples from the evaluation runs: Provided random initialization, our approach is capable to synthesize point contact grasps that comply to the different reachability constraints. The dotted lines indicate the path each contact took in 3D space during the iterated refinement and hill climbing steps. In many cases, the contact positions had to be heavily adjusted to fulfill reachability. The reachability bounds used in the example were (from left to right): $r = 0.033, 0.066, 0.2$ and $0.33$.

S1, S3, . . . , S6 using k-means and consider each combinations of clusters \{Si, Sj, Sl\}. From each combination, we retain the best quality grasp, resulting in 20 different grasps. These grasps serve directly as initialization for our approach. Our automatic closing considers the contact position and normals as grasp parameters. For each contact, a virtual fingertip is placed on an approach line outside of the object. Now using the original mesh model $F_0$, all fingertips are moved along the approach line until collision, where contact positions and normals are recorded. The reachability bound $r = 0.33$ was used for this experiment.

The results depicted in Fig. 11 suggest that automatic fingertip closing generally produces grasps with lower quality. On the right-hand side of Fig. 11, we show the difference in grasp quality for the same initial grasp. In the majority of the cases, the difference was positive, indicating that our approach performed better for almost every initialization.

5.7 Reachability Bound and Number of Contacts

Finally, we present qualitative results that support our choice of reachability measure explained in Sec. 3.3. For a reachability bound of 0.033, 0.066, 0.2, and 0.33, we execute our adaptive approach for several mesh models. The results are depicted
6. CONCLUSIONS

in Fig. 12 where it can be seen that despite random initialization, our approach is capable to synthesize point contact grasps that comply to different reachability constraints. The dotted lines indicate the path each contact took in 3D space during the iterated refinement and hill climbing steps. In many cases the contact positions had to be heavily adjusted to fulfill reachability. By setting reachability bound to $r = 0.33$, we also show that our approach is able to synthesize point contact grasps with varying numbers of contacts. As is to be expected, Fig. 13 shows that grasp quality is improved the more contacts are available.

![Grasp quality distribution](image)

Figure 13: Grasp quality distribution for synthesized grasps on the plane model with 3 and 4 contacts.

6 Conclusions

We proposed a method for synthesizing grasps based on point contact search using search space discretization. Grasp synthesis is formulated as a combinatorial optimization problem. Based on a multilevel refinement, we have considered a hierarchy of recursively coarsened mesh models and locally improved grasps at each level. For this, we have optimized grasp quality subject to a reachability measure at each level and matched grasps to the next lower level, to finally achieve a grasp on the original object model.

Our approach is applicable to complex input meshes, having thousands of faces and highly concave shape. Thus, it provides the first tractable method for search of grasp contacts on such input data. Our empirical evaluation shows that the method produces feasible, high quality grasps from random and heuristic initializations. It outperformed random sampling relying on a large number of generated grasps and automatic hand closing technique. Analyzing the influence of object model properties and method parameters such as the number of coarsening levels and the size of the local search space, we have shown that our approach is viable. In the future, we plan to additionally consider hand kinematics in the optimization function and we intend to study the influence of input data uncertainty on our approach.
7 Acknowledgement

We thank Manuela Ortlieb, Research Group Foundations of AI, University of Freiburg, Germany, for her useful suggestions.

References

REFERENCES


Paper C

Hierarchical Fingertip Space for Multi-fingered Precision Grasping

Published in

*IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014*
Hierarchical Fingertip Space for Multi-fingered Precision Grasping

Kaiyu Hang, Johannes A. Stork and Danica Kragic

Abstract

Dexterous in-hand manipulation of objects benefits from the ability of a robot system to generate precision grasps. In this paper, we propose a concept of Fingertip Space and its use for precision grasp synthesis. Fingertip Space is a representation that takes into account both the local geometry of object surface as well as the fingertip geometry. As such, it is directly applicable to the object point cloud data and it establishes a basis for the grasp search space. We propose a model for a hierarchical encoding of the Fingertip Space that enables multilevel refinement for efficient grasp synthesis. The proposed method works at the grasp contact level while not neglecting object shape nor hand kinematics. Experimental evaluation is performed for the Barrett hand considering also noisy and incomplete point cloud data.

1 Introduction and Contributions

Research in robotic grasping ranges from the sensory perception problem [24, 29, 20] to task level grasp planning [2]. For applications such as dexterous in-hand manipulation, precision grasping is a necessary requirement [35, 16, 19, 18, 12]. The synthesis of precision grasps has been in particular addressed in [6, 7, 8, 33, 36] but in a rather limited manner. In this paper, we address the problem of generating precision grasps on objects of complex shapes and propose the following:

- A concept of Fingertip Space – an integrated representation of object/fingertip contacts space that takes into consideration both local object geometry and fingertip shape. It directly operates on the object point cloud and establishes a basis for the grasp search space.

- A hierarchy of the Fingertip Space for multilevel refinement of grasps allowing for an efficient search of stable grasps.

Our work is motivated by the fact that most of the contemporary object representation approaches concentrate on the global rather than local surface properties and are therefore not suitable for generating precision grasps. Examples
Figure 1: System pipeline: Given an object point cloud and a robotic hand as input, our system (A) extracts a fingertip space directly from the object point cloud and builds a hierarchical representation of it. (B) By incorporating the fingertip space hierarchy and a hand reachability measure, the multilevel refinement procedure searches for a feasible combination of contacts with an initial hand configuration. (C) In the end, the synthesized grasp is realized by local contact positions optimization with respect to the synthesized contacts.

include Reeb graph [11], Medial Axis [30], topological features [28], primitive shapes [25, 20] and approximated parametrized volumes [5, 11, 3]. Inspired by [27], which has proven that nearby grasps with certain bounded contact differences are also bounded in grasp quality, it enables us to consider Fingertip Unit, which preserves surface local information, as the basis of Fingertip Space. On the other hand, hierarchical representation of the Fingertip Space provides a multi-resolution global view of the target object to facilitate the grasp planning in an efficient way. In comparison with the widely used sampling based precision grasp planners [26, 13, 7], our representation makes the grasp planning more reliable on complex shapes. Moreover, reachability is an important component ensuring that the synthesized grasp is applicable [39, 40]. By sampling and encoding feasible hand configurations, we approximate the reachability manifold non-parametrically to produce reachable grasps. Finally, the execution of the synthesized grasp is computed similarly to [10].
The system pipeline is depicted in Fig. 1. We assume that the friction coefficients are known and that the center of mass of the object is the centroid of its point cloud. The rest of this paper is organized as follows: In Sec. 2, we formulate the problem of precision grasp synthesis in the context of our system. In Sec. 3, we introduce the extraction of Fingertip Space and its hierarchy, which is shown as block (A) in Fig. 1. Multilevel refinement of grasps, shown as block (B), is described in Sec. 4 along with Stochastic Hill Climbing. In Sec. 5, we describe details of our system implementation and grasp execution (Block (C)) and present the experimental evaluation. We conclude the work together and introduce ideas for the future work in Sec. 6.

![Figure 2: Formation of the fingertip hierarchy exemplified for four levels. Left: An AHC clustering tree is used to retrieve a partitioning of the fingertip space into $|\Phi|$, 10, 3, and 1 cells. For each cell a circle symbolizes the representative fingertip unit. Right: The representative units are used as parents in a DAG. Edges to siblings (in red) and to cousins (in blue) are only shown for the fingertip unit $\varphi_1 = \varphi_{0,1}$.](image)

### 2 Problem Formulation and Notation

We begin by presenting the notation in the table below and then continue with the formalization of the problem.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_g$</td>
<td>Number of fingers of a robot hand</td>
</tr>
<tr>
<td>$L$</td>
<td>Set of all contact locations</td>
</tr>
<tr>
<td>$g_g$</td>
<td>Fingertip grasp with contacts $C_g$, joint values $\text{Joint}_g$, and hand pose $\text{Pose}_g$</td>
</tr>
<tr>
<td>$\Phi = {\varphi_i}$</td>
<td>Fingertip Space of fingertip units $\varphi_i$</td>
</tr>
<tr>
<td>$d_{\Phi} : \Phi \times \Phi \rightarrow R$</td>
<td>Distance measure on fingertip units</td>
</tr>
<tr>
<td>$\Phi = \Phi \cup {\varphi_{i,j}}$</td>
<td>Fingertip Space with additional representative parents $\varphi_{i,j}$</td>
</tr>
<tr>
<td>$G_\Phi$</td>
<td>Fingertip Hierarchy graph</td>
</tr>
<tr>
<td>$(G_\Phi)_i$</td>
<td>$i$th hierarchy level induced by $G_\Phi$</td>
</tr>
<tr>
<td>$S$</td>
<td>Fingertip grasp solution space</td>
</tr>
<tr>
<td>$S_i$</td>
<td>$i$th hierarchy level solution space</td>
</tr>
<tr>
<td>$Q : C_g \rightarrow R$</td>
<td>Grasp quality function</td>
</tr>
<tr>
<td>$R : C_g \rightarrow R^+$</td>
<td>Reachability function</td>
</tr>
<tr>
<td>$P = {(p_i, n_i)}$</td>
<td>Point cloud with unit length normals</td>
</tr>
<tr>
<td>$N_r$</td>
<td>Point neighborhood within radius $r$</td>
</tr>
<tr>
<td>$H_{zi}^k \subseteq \Phi$</td>
<td>Cell $k$ of a $z_i$-element partitioning of $\Phi$</td>
</tr>
<tr>
<td>$\hat{M} = {m_\hat{g}}$</td>
<td>Sampled reachability manifold of sampled grasps $\hat{g}$</td>
</tr>
</tbody>
</table>
For synthesizing precision grasps, which we refer to as fingertip grasps in this work, we next introduce how to construct a representation that hierarchically integrates global and local features of the object and fingertips based on the **Fingertip Space**. Starting from the top level of the hierarchy, our system starts from an initial grasping pose and then optimizes the contacts through the hierarchy in a coarse-to-fine manner to finally produce a stable and reachable fingertip grasp.

2.1 Fingertip Grasps

We consider fingertip grasps for a hand with \( n_f \) fingers, formalized as the tuple \( g = (C_g, \text{Joint}_g, \text{Pose}_g) \). We refer to contacts between the robot fingertips and the object as \( C_g = \{c_i\}_{i=1}^{n_f} \), the values for the end effector joints as \( \text{Joint}_g \), and the position and orientation of the hand as \( \text{Pose}_g \). A fingertip grasp provides one individual contact, \( c_i \), for each fingertip which means that for each finger there exists one location \( l_i \in \mathcal{L} \) on the object that is in contact with the fingertip. If no value for \( \text{Joint}_g \) exists ensuring that the fingertips can exert force onto the object via the individual contacts, \( C_g \) is considered not reachable.

2.2 Fingertip Space

The set of potential contact locations, \( \mathcal{L} \), is large but many locations are not viable due to the local surface geometry. To keep grasp synthesis tractable, we propose a finite discrete set of locations on the object, \( \Phi = \{\varphi_i\} \), that consists of only viable locations and denote it as **Fingertip Space**. The elements, \( \varphi_i \) of this space are named **Fingertip Units**. Thus, viable grasp locations take into account local object surface and fingertip geometry. Sec.3 provides more details on this.

Provided a similarity measure for fingertip units, it is possible to assign structure to the space \( \Phi \subset \Phi \) in form of a directed acyclic graph \( G_\Phi = (\Phi, E_\Phi) \). We define \( G_\Phi \) such that similar fingertip units are pairwise connected by edges and introduce new parent units as representatives of all their descendants in \( \Phi \). The set \( \Phi = \Phi \cup \{\varphi_{i,j}\}_{i,j} \) consists of fingertip units and introduced ancestor units. The symbol \( \varphi_{i,j} \) with \( i > 0 \) denotes the \( j \)th parent unit in the \( i \)th level of the hierarchy, representing all fingertip units that are commonly represented by its children in the \( (i-1) \)th level. As shown in Fig.2 on the right, the resulting hierarchy of fingertip space has a single root and similar fingertip units are connected. In the graph, we refer to members of \( \Phi \) as elements \( \varphi_{0,j} \) and denote with \( (G_\Phi)_i = \bigcup \varphi_{i,j} \) the \( i \)th hierarchy level induced by \( G_\Phi \), e.g. especially we have \( (G_\Phi)_0 = \Phi \). Connected nodes from the same level are neighbors, and connecting pairs of parent units can be exploited (see Sec.3.2).

The graph \( G_\Phi \) and the induced hierarchy levels \( (G_\Phi)_i \) form our object representation. For fingertip grasp synthesis, the above definitions efficiently provide relevant information by adding an explicit similarity-based structure to the fingertip units space: i) Similar units are directly connected. ii) Dissimilar units are found by considering the units represented by dissimilar distant ancestors. iii) An
3. **Fingertip Space Representation**

Increasingly coarser representation is found by considering the members or levels further up in the hierarchy. iv) Similar fingertip units are collected under a common parent.

### 2.3 Fingertip Grasp Selection by Optimization

To synthesize a feasible fingertip grasp on an object, it is necessary to select locations from $L$ that afford stable contacts and ensure reachability. We first search for stable and reachable contacts and after that check if there are solutions for $\text{Joint}_g$ and $\text{Pose}_g$ that realize the grasp. By approximating the set of all possible contacts with $\Phi$ as described in Sec. 2.2, we can formalize the contacts as $C_g = (\varphi_1, \varphi_2, \ldots, \varphi_{n_g}) \in S$. Thereby, we denote $S = \prod_{k=1}^{n_g} \Phi_k$ as the solution space consisting of fingertip spaces of different robot fingers $\Phi_k$. All surface locations that do not support the placement of a specific fingertip are disregarded.

Given a measure of grasp quality in terms of fingertip units $Q(C_g) \in \mathbb{R}$ and a measure of reachability, $R(C_g) \in \mathbb{R}^+$ we can formulate an optimization objective in terms of solution space elements as $\theta(C_g) = Q(C_g) + \alpha R(C_g)$. Here, we assumed perfect reachability for $R(C_g) = 0$ and set $0 > \alpha \in \mathbb{R}$. The optimization problem is then given as

$$C_g^* = \arg\max_{C_g \in S} \theta(C_g) \quad (1)$$

Concretely, we do not solve Eq. 1 directly but formulate a hierarchy of increasingly approximated problem instances as explained Sec. 4.3. Grasp synthesis is finalized in continuous coordinates by inverse kinematics for the selected contacts $C_g^* \in S$ as described in Sec. 4.4. If the resulting grasp is obstructed or not reachable, we start a new search with a different initialization.

### 3 Fingertip Space Representation

In this section we explain fingertip unit extraction from arbitrary point clouds with normals using a simple finger model. We first provide a definition of fingertip units in terms of input data and elaborate on the fingertip hierarchy which is used in Sec. 4.3.

#### 3.1 Extraction of Fingertip Units

In Sec. 2.2 we only state a qualitative definition of fingertip units as locations on the object that allows the placement of a fingertip. Observing an arbitrary point cloud $P = \{(p_i, n_i)\}_i$ with normal vector estimates, we need to extract a finite set of such locations by investigating $P$ while taking a finger model into account. For the purpose of this work, we focus on Barrett hand and consider contacts where the inside of the distal links rests on the object surface. The fingertip model describes a flat circular region located at the center of the distal link’s inner surface and has radius $r$, as shown in Fig. 3.
The layout is a combination of text and a figure. The text discusses a filter for rejecting points and defines a position and normal pair for a fingertips unit in terms of variance criteria. The hierarchy of fingertip space is also described, involving agglomerative hierarchical clustering of a similarity-based graph induced by the filter. The image shows a point cloud with rejected points highlighted in red and a magnified view indicating points rejected due to variance criterion.
We exploit this property to construct the hierarchy in \( G_\Phi \) by computing a sequence of \( l \) partitions with \(|\Phi| = z_0 > z_1 > \cdots > z_l = 1\) number of cells. For each cell \( H^{z_i}_j \subseteq \Phi \) with \( i > 0 \), we create a representative fingertip unit \( \varphi_{i,j} \in \Phi \) from the median position and the mean normal of all contained fingertips units. Parent-child edges are introduced for each two fingertips units \( \varphi_{i,j} \) and \( \varphi_{i-1,k} \) with \( i > 0 \) if the child’s cell is contained in the parent’s cell.

\[
\forall H^{z_{i-1}}_k \subseteq H^{z_i}_j : (\varphi_{i,j}, \varphi_{i-1,k}) \in E_\Phi
\]

Additionally we connect all siblings nodes and introduce edges to all nodes who’s parents are siblings. This process is exemplified in Fig. 2.

Concretely, we are interested in grasp similarity for search and require similar fingertip units to be grouped together. The admittedly crude fingertip distance measure of Eq. 4 provides plausible results in terms of positions and normals.

\[
d_\Phi(\varphi_i, \varphi_j) = \| p_i - p_j + \eta (n_i - n_j) \|
\]

The parameter \( \eta \in \mathbb{R}^+ \) balances position and normal. Larger \( \eta \) induces more parallel or flat geometry and small \( \eta \) results in compactly shaped cells but allows more normal vector variance. Furthermore, we need to specify the number of levels \( l \) and the number of nodes per level \( z_i \) in order to get cutoff values from AHC. For simplicity, we base the number of nodes per level on an incrementation ration and \( m_l - 1 \). Note that \( m_l = 1 \) and \( m_0 = |\Phi| \). Fig. 3 shows 20 cells of different size with \( \eta = 3 \).

4 Grasp Synthesis

In Sec. 2.3 we have stated the discrete version of our grasp synthesis as a combinatorial optimization problem. This section serves to describe our choice of reachability measure \( R \) and grasp quality function \( Q \) for Eq. 1. We also describe the optimization procedure using the multilevel refinement metaheuristic.

4.1 Grasp Stability Metric

All functions in Eq. 1 are defined on sets of fingertip units \( \varphi_i = (p_i, n_i) \). It is therefore convenient to focus on quality measures for point contacts as many approaches to robotic grasping are based on force analysis and the concept of force-closure \[14, 4\]. There, the forces exerted by the robot and friction of the surfaces are considered. For \( Q \), we choose to evaluate the force-closure property of a grasp with the \( L^1 \) grasp quality measure \( Q_\mu \) reported in \[7\] that employs the Coulomb friction model. The grasp quality \( Q = Q_\mu \) is a function of all contact positions and normals, the center of mass of the object and the friction coefficient \( \mu \in \mathbb{R}^+ \). We can thus directly refer to the fingertip units for point contacts \( p_i \in \mathbb{R}^3 \) and inward-pointing unit surface normals \( n_i \in \mathbb{R}^3 \). A grasp is force-closed if \( Q_\mu \) is larger than zero.
4.2 Reachability Measure

For the optimization objective in Eq. 1 we require a non-binary reachability measure \( R \) of a set of fingertip units that relates to values for \( \text{Joint}_g \) and \( \text{Pose}_g \). However, computing an approximate inverse kinematic solution and measuring the residual error in each optimization step is computationally infeasible. Instead, we consider an approximation of the fingertip reachability manifold \( \hat{\mathcal{M}} \) and assume a free-floating hand model. Feasible hand configurations are generated by rejection sampling and their fingertip positions and normals are recorded in an affine invariant encoding. To this end, we retain a vector, \( m_g \), of pairwise fingertip distances and normal differences from a sampled grasp \( \hat{g} \) and keep the associated \( \text{Joint}_{\hat{g}} \) values.

For a grasp hypothesis \( C_g \) we can calculate the encoding \( m_g \) and access the nearest neighbor in encoding space \( m_{\hat{g}} \in \hat{\mathcal{M}} \). If the manifold is sampled sufficiently, the differences between \( m_g \) and \( m_{\hat{g}} \) can be considered as the reachability residual

\[ R(C_g) = \| m_g - m_{\hat{g}} \| \] (5)

This reachability measure relies on the distances in an encoding space and we are aware that better techniques exist, e.g., density estimation as in [23] that takes into account also object-level impedance control. However, our \( R \) is used in a heuristic way to reduce the searched space for Eq. 1 and to initialize hand configurations \( \text{Joint}_g := \text{Joint}_{\hat{g}} \) and has served sufficiently for this purposes.

4.3 Multilevel Refinement Optimization

As in our previous work [17], we apply the multilevel refinement metaheuristic [38] for a hierarchy of combinatorial optimization problems. The fingertip space object representation defined in Sec. 2.2 offers a multiresolution view of the object and can be exploited for refinement search. We refer to the fingertip hierarchy levels \( (G_\Phi)_i \) to form increasingly approximated instances of the solution space. This is achieved by defining \( S_i = \prod_{k=1}^{n_g} (G_{\Phi_k})_i \) as the solution space on the \( i \)th refinement level. On each level \( i \), a solution \( C_i^g \) is initialized by extending the solution of the previous level \( i + 1 \) and optimizing it, resulting with a solution \( C^g = C_0^g \) in the search space \( S_0 = S \).

In this context the individual optimization problems are usually optimized using local optimization methods for convex problems [34]. As argued in our previous work, [17] we cannot expect convex objective manifolds for complex objects. Furthermore, the result of hill climbing techniques is heavily dependent on initialization in non-convex solution spaces. Different algorithms [21, 31] have been proposed to escape local minima. In this work, we adopt stochastic hill climbing [15].

In this algorithm, the objective function is not directly used to improve the current solution. Instead, the change between two solutions \( C_g \) and \( C_g' \) is conditioned on the probability stated in Eq. (6).

\[ \Pr(C_g, C_g') = \left(1 + \exp \frac{\theta(C_g) - \theta(C_g')}{\zeta}\right)^{-1} \] (6)
4. GRASP SYNTHESIS

The search randomness is determined by $\zeta$. Large values make the steps completely random, whereas the algorithm degenerates to hill climbing when $\zeta$ is very small.

Our basic optimization procedure is shown in Alg. 4. The function rand(0,1) produces uniformly distributed real numbers between 0 and 1. Children and neighbors of grasps are created from the respective children and neighbors of the constituting fingertip units in the graph $G_\Phi$ as defined in Sec. 2.2.

\begin{algorithm}
\textbf{Input:} $\zeta$, maxIter, $G_\Phi$, $\theta$
\textbf{Output:} grasp $g$
\begin{algorithmic}[1]
\State $\text{for } i = l - 1 \text{ to } 0 \text{ do}$  \Comment{Initialization}
\State $C^i_g \leftarrow$ random from $S_i$
\State $\text{else}$ \Comment{Extension}
\State $C^i_g = \underset{C_i \text{ child of } C_{i+1}^g}{\text{argmax}} \theta(C^i_g)$
\State $\text{end if}$
\State $\text{for } 1 \text{ to maxIter do}$ \Comment{Refinement}
\State $C_g \leftarrow$ some neighbor of $C^i_g \in S_i$
\State $\text{if } Pr(C_g^i, C_g) \geq \text{rand}(0,1) \text{ then}$
\State $\text{end if}$
\State $\text{end for}$
\State $\text{end for}$
\end{algorithmic}
\caption{Multilevel refinement with stochastic hill climbing for grasp synthesis}
\end{algorithm}

4.4 Grasp Realization

The optimization procedure described in Sec. 4.3 results in a grasp comprised of discrete fingertip units $C^*_g$. For this grasp, the sampling-based reachability measure from Sec. 4.2 provides the joint configuration $\text{Joint}_g$ of the closest recorded grasp $\hat{g}$ in encoding space. The optimization procedure described below employs a continuous optimization for $\text{Joint}_g$ and $\text{Pose}_g$ in terms of $C^*_g$ to close the gap between discretization, sampling and applicable continuous solutions. This is achieved by first approximately aligning the hand to the grasping pose with an affine transform between $C^*_g$ and fingertips of $\hat{g}$, and then locally optimizing simulated contact positions.

The initial affine transform can be found by minimizing the Euclidean error for the known correspondences between $C^*_g$ and the fingertips of $\hat{g}$. An example of initial hand alignment is shown in Fig. 4. As can be seen in Fig. 4 not all fingers have an initial single surface contact. For this reason, we first open the colliding fingers using proportional joint value increments and then close all fingers until contact to get simulated contact positions $C^*_g$. We then turn to gradient decent to
minimize the error between the positions of $C_g^\dagger$ and $C_g^*$ for which we compute the gradient numerically.

Figure 4: An example of initial alignment and grasp realization. Left to right: Marker positions represent $C_g^*$. The initial joint values for the grasp from $\mathcal{M}$ Hand alignment by affine transform. Final grasp after contact optimization.

5 Experimental evaluation

In this section, we first provide implementation details and then present the results of evaluation. The evaluations have been conducted in OpenRave \cite{13} on six objects: Stanford Bunny \cite{37}, Plane \cite{9} and Waschmittel \cite{22}, as well as Cup, Spoon and Milk Box scanned by ourselves.

5.1 Implementation Details

As described in Sec. 2, Fingertip Units are locations on the object surface where fingertip contacts are viable and the Fingertip Space is a finite set of Fingertip Units. If two contacts have similar locations and orientations, they would have similar contributions to the grasp stability \cite{27}. Therefore, prior to fingertip space extraction, we uniformly subsampled the object point cloud to produce fingertip unit candidates. Concretely, the subsampling was done in the scale of half of the fingertip unit size on the point cloud.

For the reachability measure $R(C_g)$ in the objective function $\theta(C_g)$, a set of feasible grasps were sampled and encoded. For the sake of efficiency, we saved all the codes in a kd-tree and consider the Euclidean distance between codes as the reachability residual.

After Alg. 4 has synthesized a grasp hypothesis, we discard the hypothesis and restart the algorithm if: a) grasp hypothesis is unstable, or b) the reachability residual is too large or c) it is not collision-free. The collision is checked by firstly aligning the configured robot hand to the grasping pose by the affine transform described in Sec. 4 and then in the simulation check whether the hand has collisions at positions other than the fingertips.

In all the experiments shown below, we use $l = 4$ layers, $m_{l-1} = 20$ and $\eta = 1$ for constructing the hierarchy of the fingertip space. We set $\alpha = 0.4$ to weight between $Q(C_g)$ and $R(C_g)$ in $\theta(C_g)$, and set $\text{maxIter} = 100$ for grasp refinement.
5. EXPERIMENTAL EVALUATION

5.2 Fingertip Space Extraction and System Evaluation

Different definitions of fingertip units result in different fingertip spaces. As described in Sec. 3, a fingertip unit in this work is defined as a circular area at the center of the distal link and has radius $r$. In this section, we show two different definitions of fingertip units of the Barrett hand and their corresponding fingertip spaces, and then we evaluate the performance of the system using these two fingertip spaces respectively.

In Fig. 5, the fingertip unit is located at the center of the distal link in both rows expressing the representative position of a fingertip. In the upper row, the radius of the circular area is the distance between the center and the long edge of the distal link, denoted as $r_1$, whereas in the lower row, the radius is the distance between the center and the short edge of the distal link, denoted as $r_2$. As we can see from their corresponding fingertip space, $r_2$ is indeed a more restricted condition that requires a larger area on the object to fit the fingertip, and in the meanwhile it results in a much sparser fingertip space.

Fig. 6 records the statistics of our system evaluation. Recall that the initialization of the system is random in this work and there is also randomness in the stochastic hill climbing procedure, the results generated by system can be different between each single run of the system. Therefore, we ran it 100 times on each of the six test objects using fingertip unit sizes of both $r_1$ and $r_2$ and investigate the averaged performance. We can see that the result for the plane model is much worse than others. This is due to the fact that the plane has many parts that are highly concave and that the Barrett hand is coupled with only 4 DoFs, the grasp realization is therefore much more difficult. For the same reason, it is much easier to expect collisions between the robot hand and the plane surface and more search iterations are therefore required. However, it is intuitive that if a more dexterous hand is employed, it is easier for us to deal with more complex object shapes. The averaged time per iteration is related to the size of extracted fingertip space $\Phi$, which is shown in the parenthesis in the first column: the larger fingertip space an object has, the more time it takes to search for a precision grasp.

Figure 5: Fingertip Space with different fingertip unit sizes.
It is worth noting that the performance of the system is generally better when fingertip radius was set to $r_2$. Because given the same fingertip embodiment, a larger fingertip unit makes it safer to stabilize a contact. Fig. 6 displays some example stable grasps synthesized by the system.

<table>
<thead>
<tr>
<th>Object(radius: #Units)</th>
<th>Stable(%)</th>
<th>Rounds</th>
<th>Time/Round</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bunny($r_1$: 3276)</td>
<td>98</td>
<td>1.73</td>
<td>13.08s</td>
</tr>
<tr>
<td>Bunny($r_2$: 293)</td>
<td>100</td>
<td>1.64</td>
<td>6.03s</td>
</tr>
<tr>
<td>Plane($r_1$: 579)</td>
<td>75</td>
<td>3.16</td>
<td>8.35s</td>
</tr>
<tr>
<td>Plane($r_2$: 96)</td>
<td>89</td>
<td>3.98</td>
<td>5.97s</td>
</tr>
<tr>
<td>Waschmittel($r_1$: 4236)</td>
<td>95</td>
<td>1.51</td>
<td>17.25s</td>
</tr>
<tr>
<td>Waschmittel($r_2$: 644)</td>
<td>95</td>
<td>1.22</td>
<td>9.13s</td>
</tr>
<tr>
<td>Cup($r_1$: 3068)</td>
<td>98</td>
<td>1.42</td>
<td>13.00s</td>
</tr>
<tr>
<td>Cup($r_2$: 730)</td>
<td>97</td>
<td>1.51</td>
<td>8.19s</td>
</tr>
<tr>
<td>Spoon($r_1$: 91)</td>
<td>88</td>
<td>1.83</td>
<td>8.22s</td>
</tr>
<tr>
<td>Spoon($r_2$: 49)</td>
<td>91</td>
<td>1.91</td>
<td>3.31s</td>
</tr>
<tr>
<td>Milk Box($r_1$: 3936)</td>
<td>100</td>
<td>2.69</td>
<td>13.68s</td>
</tr>
<tr>
<td>Milk Box($r_2$: 842)</td>
<td>100</td>
<td>3.73</td>
<td>9.98s</td>
</tr>
</tbody>
</table>

Figure 6: Statistics of algorithm evaluation. **Stable(%):** The percentage of stable grasps after the grasps were executed. **Rounds:** The averaged rounds of Alg. 4 to successfully output a good grasp, note that Alg. 4 is restarted if the final check is not satisfied. **Time/Round:** The averaged time in seconds that one round of the algorithm takes.

### 5.3 Positioning Error Tolerance of Fingertip Space

In Fig. 8 grasps are shown with their realized contacts (green) and synthesized contacts (red). The realized grasps are usually a bit different from what was synthesized, both in contact positions and normals. This is due to the fact that the reachability measure employed in Sec. 4 is an approximation of the real reachability manifold and that the Barrett hand is not dexterous enough to always sufficiently deal with non-zero reachability residual. In this section, we examine whether the synthesized grasps will remain stable if the final executions of them have positioning errors with respect to synthesized contacts.

The experiments have been conducted on Stanford Bunny, Plane and Waschmittel models by assuming that the positioning errors are within one and two fingertip unit sizes, given the fact that the positioning errors recorded in our experiments were smaller than two fingertip unit size. Similarly to the concept of Independent Contact Regions[32], we consider a grasp as tolerant to positioning errors if all contacts can be freely positioned within a certain range without losing stability. In this experiment, 100 grasps have been synthesized on all three objects, and contacts within the error limit were sampled and the percentages of sampled nearby stable grasps were recorded for each grasp. Test results are shown in Fig. 9 as the percentages of the nearby stable grasps with standard deviation on the bar plot.
5. EXPERIMENTAL EVALUATION

As can be seen from the results, neighbors of synthesized stable grasps remain stable with high probabilities. This is an evidence for the fact that synthesized grasps are tolerant to small positioning errors and that our reachability measure retains the relevant information. This can be explained by the fingertip space extraction: since fingertip units are positions where the object surface is smooth, small positioning errors will not heavily influence contact positions and normals, and the grasp stability is therefore also not heavily influenced, which can be referred back to our motivation in Sec. 1.
5.4 Precision Grasp Synthesis with Noisy Data

In this section, we examine the performance of our algorithm considering noisy sensory data. As shown in Fig. 10, we scanned the Stanford Bunny, Plane and Waschmittel models using a virtual 3D sensor while adding Gaussian noise in the viewing direction. For the extraction of fingertip units, the fingertip size was set to \( r_2 \). As we can see, the fingertip space becomes different comparing to noise-free objects. However, it is worth to note that, although the objects are noisy, the extracted fingertip units are still retaining the property of flatness and smoothness.

Figure 10: Noisy objects used in experiments and their corresponding fingertip space.

Fig. 11 records the statistics of 100 runs of our approach. Grasps were synthesized using noisy data and the final grasp qualities are computed after the synthesized grasps have been executed on the perfect objects. The result shows that
the percentage of stable grasps have been decreased in comparison to the noise-free experiments, however, the system can still synthesize stable precision grasps.

<table>
<thead>
<tr>
<th>Object(#Units)</th>
<th>Stable(%)</th>
<th>Rounds</th>
<th>Time/Round</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bunny(122)</td>
<td>92</td>
<td>2.12</td>
<td>7.63s</td>
</tr>
<tr>
<td>Plane(111)</td>
<td>83</td>
<td>4.16</td>
<td>7.24s</td>
</tr>
<tr>
<td>Waschmittel(582)</td>
<td>90</td>
<td>2.05</td>
<td>9.75s</td>
</tr>
</tbody>
</table>

Figure 11: Statistics of algorithm evaluation with noise. *Stable(%)*: The percentage of stable grasps after the grasps were executed. *Rounds*: The averaged rounds of Alg. 4 to successfully output a good grasp, note that Alg. 4 is restarted if the final check is not satisfied. *Time/Round*: The averaged time in seconds that one round of the algorithm takes.

5.5 Grasp Synthesis with Partially Observed Data

It is difficult to observe complete point clouds of target objects in real applications. In this section, we simulate partial views of objects by setting locations of a virtual camera, and then we show example stable grasps synthesized by the system, see Fig. 12.

Figure 12: Upper: Fingertip space of the partially observed objects. Lower: Grasps synthesized on partially observed objects. Unobserved parts on the object are shown in transparency.

As shown in the examples, grasps can still be successfully synthesized and the contacts are only synthesized for visible positions. This is because the fingertip space extraction and the hierarchy construction operate directly on the observed point cloud and does not require the object to be completely observed. This implies another advantage of the proposed object representation that the system is able to synthesize precision grasps as long as the observed parts of the object are graspable.
In the real applications, if no successful grasps can be synthesized by the system from a single view of the object, the robot can move to a different position to find graspable parts.

5.6 An Example of Grasp Synthesis and Realization

In this section, we present an example of grasp optimization and execution.

Multilevel Grasp Optimization

As the refinement procedure in Alg. 4 aims at improving the objective function $\theta(C_g)$, it searches for larger $Q(C_g)$ and smaller $R(C_g)$ values.

Figure 13: Left: Records of multilevel grasp optimization. Right: Records of contact positions optimization, $\rho = \|C_g^+ - C_g^*\|$. The horizontal axes are number of iterations in both figures.

Fig. 13 displays one example of $\theta(C_g)$, $Q(C_g)$ and $R(C_g)$ curves of Alg. 4 applied on the Bunny model. We can see that the $\theta(C_g)$ value is generally increasing with a few decreases due to the randomness in the Alg. 4 and that the $Q(C_g)$ value is also generally increasing. However, the $R(C_g)$ value is decreasing but sometimes increasing, this is because the search procedure was attempting many different joint configurations to fit a grasp while balancing between other objectives. Next, we apply the contacts optimization to realized the grasp with synthesized contacts.

Contact Positions Optimization

As shown in Fig. 13, $\rho$ value is generally decreasing during the gradient descent but is occasionally overshooting. The overshots are due to the joint space of robot hand and the object surface is very complicated and has many local optima. After the contact positions optimization is done, the final stable precision grasp was achieved as shown on the right. It is worth to mention that as the fingertips’ positions after affine transform was already very close to the desired position, the gradient descent did not need many steps to converge.
6 Conclusion

In this paper, we have proposed a concept of *Fingertip Space*, which is an integrated representation of both object local geometry and fingertip geometry, and shown its use in precision grasp synthesis. By building a hierarchical representation of the fingertip space, we have enabled multilevel refinement for precision grasp synthesis. Our experimental evaluation with a Barrett hand has shown that the fingertip space and its hierarchy is a viable and efficient representation for precision grasp synthesis, and that the multilevel refinement facilitates the search procedure. We have also evaluated the positioning errors tolerance of our system, as well as demonstrated examples of our system working with noisy and incomplete data. In the future, we are planning to implement our system on a real robot and additionally make the modular system more compact and flexible for different robot embodiments and search algorithms to be plugged in.

References

REFERENCES


Paper D

On the Evolution of Grasping Manifolds

Published in
IEEE International Conference on Robotics and Automation, 2016
Efficient and accurate planning of fingertip grasps is essential for dexterous in-hand manipulation. In this work, we present a system for fingertip grasp planning that incrementally learns a heuristic for hand reachability and multi-fingered inverse kinematics. The system consists of an online execution module and an offline optimization module. During execution the system plans and executes fingertip grasps using Canny’s grasp quality metric and a learned random forest based hand reachability heuristic. In the offline module, this heuristic is improved based on a grasping manifold that is incrementally learned from the experiences collected during execution. The system is evaluated both in simulation and on a Schunk-SDH dexterous hand mounted on a KUKA-KR5 arm. We show that, as the grasping manifold is adapted to the system’s experiences, the heuristic becomes more accurate, which results in an improved performance of the execution module. The improvement is not only observed for experienced objects, but also for previously unknown objects of similar sizes.

1 Introduction

Significant progress has been made in the area of robotic grasping. For fingertip grasping, localizing contacts on the object that provide a stable grasp and finding a hand configuration for realizing these contacts have been addressed as separate problems. As a result, in these approaches it is not guaranteed that the planned contacts are kinematically feasible for a specific hand.

To overcome this limitation, various grasp optimization frameworks have recently been proposed that integrate both stability and hand reachability analysis. In [8], the finger kinematics are modeled as an optimization constraint for efficiently finding contacts on objects that can be approximated by super-quadrics. Given a set of initial hand poses, [24] optimizes both hand configurations and contacts to generate a dense set of grasps offline. The framework presented in [12] plans contact triplets on incomplete 3D point clouds online, utilizing hand shape primitives as a heuristic to constrain the contacts optimization. In [9], a system first learns task related grasping parts of objects from human demonstrations. Next, the contact
Figure 1: A 2D-projection of the seven dimensional configuration space of the Schunk-SDH hand with example grasps. The colored areas show a grasping manifold that is learned from experiences. Video: [http://www.csc.kth.se/~kaiyuh/videos/graspManifold.mp4](http://www.csc.kth.se/~kaiyuh/videos/graspManifold.mp4).

points on these parts are found by stochastic optimization, which is constrained by closed kinematic loops between the fingertips and the object part. In general, these systems require trade-offs between the desired computational efficiency and the precision of planning by either approximating the object shapes or the hand kinematics.

In our recent work [14], we proposed a fast fingertip grasp planner that searches stable and reachable grasps in a hierarchical search space of fingertip contacts. To ensure kinematic feasibility for a dexterous redundant hand, an uniform sampling based affine-invariant hand reachability heuristic is used. However, as a common property of sampling based heuristics [25, 26, 15, 27], the sampling resolution affects the efficiency and precision of this method. To achieve the same resolution, the number of required samples increases exponentially with the dimension of the configuration space. Further, in practice many of the samples cover invalid or task irrelevant configurations.

Instead of sampling randomly, learning from humans effectively excludes irrelevant hand configurations [10, 23]. However, this requires significant effort from human teachers and is limited by the teacher’s experience. Our key insight in this work is that we can limit the training set for a sampling based hand reachability heuristic to the manifold of for grasping relevant hand configurations. In doing so, we increase the heuristic’s accuracy in relevant regions of the configuration space,
while reducing it in irrelevant regions, see Fig. 1. We propose that a robot should learn this grasping manifold based on its own task relevant experiences. The contributions of this work are:

- We present a learning framework that incrementally learns a subspace of the hand configuration space, a grasping manifold, that is relevant for fingertip grasping.
- We show that this allows us to incrementally train a heuristic for hand reachability and multi-fingered inverse kinematics, which improves as experiences are gained.
- We further show that we can extend this approach to learning a heuristic that integrates both grasp quality and hand reachability.

We define the terminology in Sec. 2 and introduce the proposed approach in Sec. 3. In Sec. 4, we evaluate our approach both in simulation and on a real robot and end in Sec. 5 with a conclusion and a discussion of potential future work.

2 Terminology

2.1 Fingertip Grasp and Hand Kinematics

In this work, we consider a dexterous multi-fingered robotic hand with \( d \) degrees of freedom and \( f \) fingertips. Let \( C \) denote the \( d \)-dimensional configuration space of the hand and accordingly \( C_{free} \subseteq C \) the set of self-collision-free hand configurations. For an object let \( \mathcal{O} \subseteq \mathbb{R}^3 \) denote the object’s surface. A fingertip grasp \( \gamma = (c_1, \ldots, c_f) \) is a \( f \)-tuple of point contacts on the surface \( \mathcal{O} \), where \( c_i \) is the contact of fingertip \( i \) with the object. Each contact \( c_i = (p_i, n_i) \) consists of its position \( p_i \in \mathcal{O} \) and normal \( n_i \in S^2 = \{ x \in \mathbb{R}^3 : \| x \| = 1 \} \) defined in the object’s frame.

If the hand is in a configuration \( \Theta \in C_{free} \), the fingertip poses in the hand frame are \( \mathcal{Z} = (\zeta_1, \ldots, \zeta_f) = F(\Theta) \), where \( F \) denotes the forward kinematics of the hand. Similar to a contact on \( \mathcal{O} \), a fingertip pose is a tuple \( \zeta_i = (p_i, n_i) \), where \( p_i \in \mathbb{R}^3 \) is the position and \( n_i \in S^2 \) the normal of fingertip \( i \) in the hand frame. Let \( \mathcal{H} \) denote the pose of the hand frame relative to the object frame. A fingertip grasp \( \gamma \) is achieved by the hand configuration \( \Theta \), if \( T_O(F(\Theta), \mathcal{H}) = \gamma \) holds. \( T_O(Z, \mathcal{H}) \) transforms a tuple of fingertip poses, \( Z \), from the hand’s frame to the object’s frame.

2.2 Affine Invariant Grasp Encoding

We adopt an affine invariant grasp encoding as exemplified in Fig. 2 with \( f = 3 \) fingertips to easily determine whether a hand configuration can achieve a given grasp on an object. Given a fingertip grasp \( \gamma = (c_1, c_2, c_3) \), we compute the grasp code as

\[
C(\gamma) = (\| p_1 - p_2 \|, \| p_2 - p_3 \|, \| p_3 - p_1 \|, \| n_1 - n_2 \|, \| n_2 - n_3 \|, \| n_3 - n_1 \|).
\]  

(1)
Given a hand configuration with fingertip poses \( Z = (\zeta_1, \zeta_2, \zeta_3) \), we compute \( C(Z) \) in the same way as Eq. (1). A hand configuration \( \Theta \) with \( F(\Theta) = Z \) is feasible to achieve a grasp \( \gamma \), if \( C(\gamma) = C(Z) \). Note that this encoding allows us to check feasibility without explicitly computing \( H \).

Figure 2: The fingertip grasp encoding visualized for the Schunk-SDH hand and a 2D example object. The red dot shows the center of the contacts.

### 2.3 Grasping Manifold

We wish to train a heuristic for hand reachability and multi-fingered inverse kinematics on the set of hand configurations that are relevant for fingertip grasping. Let \( G \) denote the set of all stable fingertip grasps achievable by the robot hand on all objects. The manifold with corners \([10]\) of hand configurations that achieve the fingertip grasps in \( G \) is defined as

\[
X = \{ \Theta \in C_{free} | \exists \gamma \in G : C(\gamma) = C(F(\Theta)) \}. \tag{2}
\]

Because of the wide definition of \( G \), it is reasonable to focus on a robot specific subset

\[
\bar{X} = \{ \Theta \in C_{free} | \exists \gamma \in G' : C(\gamma) = C(F(\Theta)) \} \subset X, \tag{3}
\]

where \( G' \subset G \) is the set of fingertip grasps on the objects the robot at hand will actually encounter during its lifespan.

Computing \( \bar{X} \) as well as \( G' \) explicitly, however, is infeasible. Further, the set of objects a robot may encounter during its lifespan may not be known beforehand. Hence, the system we present provides an incrementally improved estimate of \( \bar{X} \) based on its own experiences. Throughout this work we denote such an estimate of \( \bar{X} \) as grasping manifold.
3. METHODS

3.1 Execution Module

We adopt the grasp planner presented in our previous work [14], which computes a grasp $\gamma$ by maximizing the objective function

$$\Gamma(\gamma) = Q(\gamma) + \alpha R(\gamma)$$

in the Hierarchical Fingertip Space (HFTS). Here, $Q(\gamma) \in \mathbb{R}$ denotes the point contact based grasp quality measure $Q_1$ from [11], $R(\gamma) \in \mathbb{R}$ the reachability residual that measures the kinematic feasibility of realizing the fingertip grasp $\gamma$ with the robot hand and $\alpha \in \mathbb{R}$ a scaling factor.

In [14], $R(\gamma)$ is computed by querying a kd-tree based data structure for a stored hand configuration $\Theta_\gamma$ that minimizes the distance $\|C(F(\Theta_\gamma)) - C(\gamma)\|$. In general, due to its limited sample density, for an arbitrary input grasp $\gamma$ there is no hand configuration $\Theta_\gamma$ stored in the data structure that achieves an exact reachability residual of zero, i.e. $\|C(F(\Theta_\gamma)) - C(\gamma)\| \neq 0$. Therefore, after determining a grasp $\gamma^*$ that maximizes $\Gamma$, the planner is required to perform a post-optimization step to adjust the hand configuration $\Theta_\gamma$ to minimize the residual. Finally, the hand pose $\mathcal{H}$ is computed such that the fingertip positioning error $\epsilon = \|\gamma^* - T_0(F(\Theta_\gamma), \mathcal{H})\|$ is minimized. Note that the computation of $\mathcal{H}$ and $\Theta_\gamma$ may fail in case $\gamma^*$ is not reachable. In this case the system needs to re-plan.

In this work in contrast to [14], we replace the kd-tree based heuristic by our heuristic $\mathcal{R}$. Similar to our previous heuristic, $\mathcal{R}$ is defined as

$$\mathcal{R} : \gamma \mapsto (\Theta_\gamma, R(\gamma))$$

and provides a reachability estimate $R(\gamma) \in \mathbb{R}$ of how well a fingertip grasp $\gamma$ is kinematically reachable, as well as an inverse kinematics solution $\Theta_\gamma$ such that $C(F(\Theta_\gamma)) \approx C(\gamma)$. 

3.2 Offline Optimization Module

The execution module plans and executes grasps and, if successful, feeds its experiences to the memory. The dreaming module takes these experiences from the memory and evolves a grasping manifold, which is used as training set for the heuristic $\mathcal{R}$. As the system acquires a more specialized grasping manifold, i.e. a better estimate of $\mathcal{X}$, the heuristic $\mathcal{R}$ becomes more accurate and in turn facilitates the execution module in the future.
Every time the system successfully computed and executed a grasp on some object, the post-optimized hand configuration for this grasp is stored in the memory $M_E$. In the offline dreaming phase, the system recomputes its grasping manifold based on the collected experiences. As the heuristic is retrained on the updated grasping manifold, it provides a better estimate of hand reachability and hand inverse kinematics.

3.2 Experience Oriented Manifold Evolution

The dreaming phase is summarized in Algorithm 5. We represent the grasping manifold by a discrete set of $N_t$ hand configuration samples $M_M \subset C_{free}$, which is stored in the system’s memory $M$. Before any experiences have been gathered, there is no knowledge about $M$. Hence, it is assumed that any self-collision-free hand configuration may lie within it and we initialize our initial estimation from an uniform distribution, see Algorithm 5 line 2. At this point, $R$ can be learned as...
3. METHODOLOGY

**Input:** Memory $M = (M_M, M_E)$, Covariance $\sigma$, Number of manifold samples $N_i$, Sample parameters $\beta$ and $\rho$

**Output:** Updated memory $M = (M_M, M_E)$, Heuristic $R$

```
if $M_M = \emptyset$ then
  Bootstrap
  $M_M \leftarrow \text{SampleUniformly}(C_{free}, N_i)$
else
  $\Omega \leftarrow \text{FitGMM}(M_M)$
  $S_n \leftarrow \emptyset$
  for $\Theta_E \in M_E$ do
    $N_r \leftarrow \frac{\|M_M\|}{\|M_E\|} \left(1 - \rho_m \int_C g(\Theta_E, \sigma I) p(\Theta|\Omega) d\Theta\right)$
    $N_r \leftarrow \max(N_r, 0)$
    $S_n \leftarrow S_n \cup \text{SampleGaussian}(\Theta_E, \sigma I, N_r)$
  end for
  $M_M \leftarrow \text{SampleUniformly}(S_n \cup M_M, N_i)$
  $M_E \leftarrow \emptyset$
  $R = \text{ComputeR}(M_M)$
end if
```

Algorithm 5 The *dreaming* phase: Evolution of grasping manifold

---

detailed in Sec. 3.3 and the system is similar to our previous work [14] and ready for execution.

Once the robot gathered some experiences and enters the *dreaming* phase anew, we update $M_M$ to more accurately approximate $\bar{X}$. This update is performed by replacing samples in $M_M$ with self-collision-free hand configurations scattered around the experiences stored in $M_E$. For each experience $\Theta_E$, new samples are sampled from a Gaussian distribution $\mathcal{N}(\Theta_E, \sigma I)$ centered at $\Theta_E$ with a user specified diagonal covariance matrix $\sigma I \in \mathbb{R}^{d \times d}$, see Algorithm 5 line 9. The covariance controls the adaptation of the manifold. A large value leads to a diffuse, whereas a smaller value to a more concentrated adaptation. The motivation for sampling a neighborhood of $\Theta_E$ lies in the fact that the grasp quality measure $Q$ is Lipschitz continuous [19]. Hence, since an experience $\Theta_E$ is a hand configuration that achieves a stable grasp on some object with fingertip poses $Z$, it is reasonable to assume that fingertip poses similar to $Z$ can also achieve good quality grasps on similarly shaped objects.

We need, however, to balance between evolving the grasping manifold towards new experiences and preserving the manifold learned so far. For this, we compute a probability distribution $p(\Theta|\Omega)$, denoting the probability of a configuration $\Theta \in \mathcal{C}$ lying in $\bar{X}$ given our current grasping manifold. We model $p(\Theta|\Omega)$ as Gaussian Mixture Model (GMM) and use the expectation maximization (EM) algorithm to compute the GMM parameters $\Omega$ from our sample set $M_M$, see Algorithm 5 line 4. As shown in Algorithm 5 we select $\Omega$ such that the Bayesian information criteria (BIC) is minimized.

In order to adapt the grasping manifold, the system determines for each $\Theta_E \in$
Input: $S$ set of samples
Output: Best fitting $\Omega$
1: $BIC = \infty$
2: for $m = 1 \ldots \hat{m}$ do
3:    $\Omega' = \text{EMAlgorithm}(S, m)$
4:    if $BIC(\Omega') < BIC$ then
5:        $\Omega = \Omega'$
6:        $BIC = BIC(\Omega)$
7:    end if
8: end for

Algorithm 6 FitGMM: Determine parameters $\Omega$ for a set of sampled configurations

$\mathcal{M}_E$ its novelty utilizing $p(\Theta_E|\Omega)$. Experiences that are likely given our current grasping manifold should result in little adaptation, whereas experiences that are unlikely are considered novel and should result in significant adaptation. The degree of adaption of the grasping manifold to an experience $\Theta_E$ is governed by the number of samples $N_r \in \mathbb{N}^+$ we sample in its neighborhood. Given the GMM parameters $\Omega$ with $m$ Gaussians, $N_r$ is computed as follows:

$$N_r = \max\left(\frac{\beta |\mathcal{M}_M|}{|\mathcal{M}_E|} \left( \frac{1}{\rho m} - \int_{\mathcal{C}} g(\Theta_E, \sigma I)p(\Theta|\Omega) \, d\Theta \right), 0 \right)$$

where $g(\Theta_E, \sigma I)$ is the probability density of the Gaussian $\mathcal{N}(\Theta_E, \sigma I)$, and $p(\Theta|\Omega)$ is the probability density of the GMM parametrized by $\Omega$. $\beta \in \mathbb{R}$ controls the evolution speed and $\rho \geq \beta$ is a relaxation parameter. $N_r$ is large, if the probability of samples in the neighborhood of $\Theta_E$ belonging to the current grasping manifold is small and vice versa. The term $\frac{1}{\rho m}$ in Eq. (6) denotes the probability mass we desire this neighborhood to have.

Although the dreaming phase is performed offline, we choose to limit its computational effort by performing batch updates rather than updating $\mathcal{M}_M$ for each experience $\Theta_E \in \mathcal{M}_E$ individually. In the rare case that all experiences lie in the same small area of $\mathcal{C}_{free}$, this could lead to a rapid adaptation of our manifold to a single mode. To prevent this, $N_r$ decreases as $|\mathcal{M}_E|$ increases and vice versa.

Once all samples $S_n$ around all new experiences are sampled, we recompute the grasping manifold by uniformly re-sampling $N_i$ samples from $S_n \cup \mathcal{M}_M$, see Algorithm 5 line 11. Note that while we desire $\mathcal{M}_M$ to resemble $\bar{X}$, the proposed method does not guarantee that the estimate $\mathcal{M}_M$ lies strictly within $\bar{X}$. However, as the evolution progresses, the probability of including outliers in $\mathcal{M}_M$ decreases.

### 3.3 Reachability Heuristic by Regression Forest

Our reachability heuristic is a function as defined in Eq. (5), which maps a fingertip grasp $\gamma$ to a hand configuration $\Theta_\gamma$ as well as the probability $R(\gamma)$ that
3. METHODOLOGY

\[ C(\gamma) = C(F(\Theta_\gamma)). \]

For modeling this high dimensional non-linear mapping, we adopt the regression forest from [6] consisting of \( T \) regression trees. In order to maximize \( R \)'s accuracy for relevant hand configurations, the grasping manifold \( \mathcal{M}_M \) serves as basis for the training set. Since fingertip grasps are defined in an object frame and therefore object specific, we learn a mapping from the encodings \( C(\gamma) \) to hand configurations, making \( R \) object frame independent. The labeled training set for the regression forest is

\[ \mathcal{T} = \{ (\Theta, C(F(\Theta))) | \Theta \in \mathcal{M}_M \}. \] (7)

Each leaf of each tree in the forest provides a prediction model that is learned by probabilistic linear regression with a Gaussian distribution [2] from some subset of \( \mathcal{T} \). In order to achieve a fully probabilistic output, the splits of the training set within each regression tree are performed such that the information gain reported in [6] is maximized.

Given an input grasp \( \gamma \), the \( t \)-th tree in the forest, \( t = 0, \ldots, T - 1 \), provides the posterior \( p_t(\Theta | C(\gamma)) \). The posterior of the whole forest is the average of the posteriors of the individual trees

\[ p(\Theta | C(\gamma)) = \frac{1}{T} \sum_{t=1}^{T} p_t(\Theta | C(\gamma)). \] (8)

With the posterior at hand we can compute the heuristic value \( R(\gamma) = (\Theta_\gamma, R(\gamma)) \) as

\[ \Theta_\gamma = \mathbb{E}[\Theta | C(\gamma)] = \int_C p(\Theta | C(\gamma)) \Theta \, d\Theta, \] (9)

\[ R(\gamma) = p(\Theta_\gamma | C(\gamma)). \] (10)

Note that the evaluation of \( R(\gamma) \) is computationally inexpensive and thus suitable for grasp optimization.

3.4 Pure Manifold Based Grasp Planning

The planner as presented in Sec. 3.1 computes a grasp by maximizing both the reachability of a fingertip grasp as well as its quality, Eq. (4). As reported in [11, 5], the scaling factor \( \lambda \in \mathbb{R}^+ \) between the force term and the torque term in the computation of the grasp quality \( Q(\gamma) \) is somehow arbitrarily chosen. With a large value, the grasp will be more optimized to counteract torsional disturbances, whereas a small value makes the grasp to be more optimized towards translational disturbances. Additionally, as we do not have an accurate prior of friction coefficients, the planned grasps can be vulnerable to friction changes [13].

Recall that the evolution of the grasping manifold is based on successful stable grasps. Therefore, during the evolution, the system learns not only the relevant grasp configurations, it also implicitly suppresses grasps that are not robust, i.e. grasps that are not always stable during execution. As an extension to our system,
we therefore propose a modification of our heuristic $R$ that not only provides a reachability estimate, but also a quality estimate. Formally, we define the modified heuristic as

$$R^* : (r, C(\gamma)) \mapsto (\Theta_\gamma, R^*(\gamma)).$$

(11)

Additional to the contacts encoding $C(\gamma)$, $R^*$ takes the distance $r \in \mathbb{R}^+$ between the center of mass of the object and the center of the contact positions as argument.

In the execution module the system initially uses $R$ as described before. When a grasp $\gamma$ is successfully executed, we save both $r$ and $\Theta_\gamma$ in an additional memory $\mathcal{M}^*_E$. When the grasping manifold evolution becomes stable, i.e. $S_n$ is small, the grasping manifold provides a good estimate of $\bar{X}$. In this case, we train $R^*$ using the same method as in Sec. 3.3 with $r$ as additional variable in the input space.
4. EXPERIMENTS

The training set in this case is

\[ T^* = \{ (\Theta, (C(F(\Theta)), r)) | (\Theta, r) \in \mathcal{M}_E^* \}. \] (12)

Henceforth, we use \( R^* \) as a generative model: instead of explicitly computing the grasp quality \( Q(\gamma) \), the objective function \( \Gamma \) for the HFTS grasp planner becomes the posterior \( R^*(\gamma) \) of the \( R^* \) prediction. Since \( R^* \) is not generative for new types of objects, the system falls back to \( R \) and re-enables its evolution to adapt to the new types of objects, when these are encountered.

4 Experiments

We implemented the proposed system in Python and evaluated it for the Schunk-SDH hand with \( d = 7 \) DoFs and \( f = 3 \) fingertips in the OpenRAVE simulation environment [7] on a machine with an Intel Core i7-3770 CPU @ 3.40GHz×8 and 32GB RAM. In the execution module, when planning grasps using the HFTS grasp planner [14], the maximum iteration per hierarchy level is set to 40. In the dreaming module, the grasping manifold is represented by \( N_i = |\mathcal{M}_i| = 10^5 \) samples and the GMM model \( \Omega \) is estimated using full covariance matrices. For the construction of the random forest, the forest size is set to \( T = 10 \), the maximum tree depth is set to 10 and the minimum samples for splitting a decision node is 20.

We first evaluate the evolution of the grasping manifold using experiences collected in simulation. Thereafter, we present quantitative results to evaluate the extension described in Sec. 3.4. Finally, we show real world experiments with a Schunk-SDH hand mounted on a KUKA KR5 sixx 850 arm.

4.1 Evolution of Grasping Manifold

For the evaluation of the grasping manifold evolution, we investigate whether we can observe the following properties:

- **P.1** As the grasping manifold evolves, the number \( |S_n| \) of new samples drawn from the neighborhoods of the gathered experiences decreases. A larger evolution speed \( \beta \) results in a faster decrease in \( |S_n| \), but less generality across different grasp types and vice versa.

- **P.2** As the grasping manifold evolves, \( R \) provides better predictions of hand configurations for grasps on the training objects as well as on unexperienced objects of similar sizes.

- **P.3** A grasping manifold that is learned from experiences on some objects continues evolving towards new experiences once it is exposed to objects requiring significantly different grasps. By doing so, it generalizes over different objects and \( R \)'s performance for previously experienced objects does not deteriorate.
Figure 5: Examples grasps on the objects used in evaluation, from left to right: rivella, box, pen, jug, ball, key.

Figure 6: Number of new samples $|S_n|$ of the evolution examples shown in Fig. 4.

4.1.1 Investigation of P.1

Fig. 4 shows the evolution of four different grasping manifolds with different choices for the parameters $\beta$ and $|\mathcal{M}_E|$. In all cases the system gathers experiences by generating grasps on the two objects rivella and box shown in Fig. 5. We initialize the grasping manifolds with $N_i = 10^5$ uniformly sampled collision-free hand configurations. Except for the evolution shown in Fig. 4b, our system enters the dreaming phase every time it has experienced $|\mathcal{M}_E| = 40$ stable grasps (evaluated in simulation using [11]), i.e. 20 per object. In case of the evolution shown in Fig. 4b, it enters the dreaming phase every time a random number $|\mathcal{M}_E| = n_r + n_b$ of experiences has been made, where $n_r, n_b \in [10, 30]$ denote the number of experiences on rivella and box respectively. The parameter $\beta$ is chosen such that in Fig. 4a and Fig. 4b we expect a balanced evolution, whereas in Fig. 4c we expect an aggressive evolution towards new experiences and in Fig. 4d a conservative evolution that slowly adapts to new experiences.

As shown in Fig. 6 we observe that the number of new samples in the evolution
process is decreasing for all four evolutions. The first three evolutions shown in Fig. 4 reach a smaller number of new samples faster than the conservative evolution in Fig. 4d. We can observe visually in Fig. 4 that after generations 73, 67, 41 and 85 the grasping manifolds do not change significantly anymore.

It is worth noticing that the more aggressive the evolution is, the faster it achieves a stable manifold. However, as we can see from the evolution in Fig. 4c, an aggressive evolution results in a grasping manifold that is concentrated on few areas, which is where early experiences originate from. The conservative evolution in Fig. 4d takes the longest to achieve a stable grasping manifold, however, it is more spread. The balanced evolutions achieve a greater variety of modes than the aggressive evolution, while reaching a stable grasping manifold in less generations than the conservative one. The balanced manifold from Fig. 4a is also shown in Fig. 1, which shows the diversity of grasp configurations it contains. In contrast, the aggressive evolution suppresses the variety of grasps the system can generate. In summary, the above results support our assumption about property P.1.

4.1.2 Investigation of P.2

Fig. 7 shows the development of the positioning error $\epsilon = \|\gamma - T_O(F(\Theta), H)\|$ between fingertips and the desired contacts on an object surface for different objects as the evolution of the balanced manifold from Fig. 4a progresses. We can see that before the evolution starts, our heuristic performs worse than the reachability heuristic proposed in [14], since the linear regression performed by the random forest performs poorly for sparse data points. As the evolution proceeds, however, the positioning error decreases for all six objects. Note that the evolution is performed with the objects rivella and box, for the other objects we only evaluate $\mathcal{R}$ without feeding the experiences to the dreaming module. Nevertheless, the error also reduces for these objects, indicating that the system is capable of generalizing. This result supports our assumption about property P.2: as the evolution of the grasping manifold progresses, $\mathcal{R}$’s performance increases both for previously seen and unseen objects of similar sizes.

4.1.3 Investigation of P.3

For evaluating property P.3, we run the evolution in Fig. 4a again with a different setup: as shown in Fig. 8, once the evolution for rivella and box is stable at generation 73, we ask the system to additionally plan grasps for a new object rivella big, which is the rivella bottle scaled up by a factor of 4.0. An object of this size requires grasps for which the hand needs to be opened far wider than for any of the others. We can see that since the system has not experienced an object of this scale before, the positioning error is initially large for rivella big. However, as first experiences are made, the system starts to evolve anew, adapts its grasping manifold and eventually becomes stable again at generation 99. Thereafter, by asking it to plan grasps for a large disk, which requires the coupled joint of the Schunk-SDH hand to open
even further, the system evolves further until generation 126. It is worthwhile to note that when the new objects are introduced to the system, the positioning errors for the previously experienced objects do not increase. In summary, these results support our assumption about property P.3.

4.2 Evaluation of $R^*$

As described in Sec. 3.4, once the grasping manifold evolution becomes stable, we can learn a new heuristic $R^*$ from all gathered experiences. Next, we investigate whether we can observe the following property:

- **P.4** Once the grasping manifold becomes stable, our grasp planner with learned heuristic $R^*$ is able to plan stable grasps with a performance equal to the planner with heuristic $R$, while achieving faster runtimes.

When the grasping manifold evolution shown in Fig. 4(a) is stable at generation 73, the system has experienced $73 \times 2 \times 20 = 2920$ grasps. Hereafter, the system uses the 2920 experiences to learn the new heuristic $R^*$. From the table shown in Fig. 9, we can see that the mean positioning errors and the grasp qualities for both heuristics are similar. This is expected since the stability manifold learned from the 2920 experiences tends to lead the planner to plan grasps similar to its experiences. Note that as the computation of the grasp quality metric is omitted, the runtime for the grasp planner using $R^*$ is in general decreased by $2/3$ to $3/4$. This result supports our assumption about property P.4.
4. EXPERIMENTS

Figure 8: Positioning errors $\epsilon$, the evolution is controlled by the parameters shown in Fig. 4(a). Once the evolution becomes stable for rivella and box, the new objects rivella\textsubscript{big} and disk are introduced into the system and trigger the continuation of the evolution. The reported results are collected from 100 simulated grasps on each object per generation.

4.3 Evaluation on a Real Robot

Finally, we wish to ensure that our system produces grasps that can be executed successfully on a real robot. In particular, we assume the following property:

- **P.5** Grasps generated with the learned heuristic $R^*$ are stable in the real world.

For this, we execute grasps on the system illustrated in Fig. 10. The grasp planner uses $R^*$ which was trained on the shown five objects in simulation. For each experiment, we place one of the objects at an arbitrarily chosen reachable pose on a table in front of the robot. The robot is tasked with picking up the object and placing it in a bin located next to the table. We execute this sequence for each object six times and record the number of successful placements. As our grasp planner does not consider collisions with the environment nor the kinematics of the robot arm, we repeatedly plan grasps anew until a reachable grasp has been found. To execute grasps, we utilize a position controller for the finger joints. To exert some force on the grasped object, we additionally close the fingers approximately
<table>
<thead>
<tr>
<th>Object</th>
<th>Err(cm)</th>
<th>Q</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rivella</td>
<td>0.42 ± 0.04</td>
<td>1.13 ± 0.09</td>
<td>0.37 ± 0.06</td>
</tr>
<tr>
<td>box</td>
<td>0.39 ± 0.04</td>
<td>1.24 ± 0.37</td>
<td>0.79 ± 0.02</td>
</tr>
<tr>
<td>pen</td>
<td>0.55 ± 0.02</td>
<td>0.20 ± 0.04</td>
<td>0.99 ± 0.07</td>
</tr>
<tr>
<td>jug</td>
<td>0.85 ± 0.13</td>
<td>1.27 ± 0.22</td>
<td>1.21 ± 0.27</td>
</tr>
<tr>
<td>ball</td>
<td>0.09 ± 0.01</td>
<td>0.42 ± 0.02</td>
<td>0.94 ± 0.04</td>
</tr>
<tr>
<td>key</td>
<td>0.65 ± 0.06</td>
<td>0.22 ± 0.02</td>
<td>1.14 ± 0.23</td>
</tr>
</tbody>
</table>

Figure 9: Statistics for the comparison between $\mathcal{R}$ and $\mathcal{R}^*$ (shaded). The results reported are collected by generating 100 grasps in simulation for each object for both heuristics respectively. Err: positioning errors, Q: grasp quality measured by [11]. Time: runtime for planning a grasp. The evaluations were implemented in Python and run on a machine with Ubuntu 12.04 running on an Intel Core i7-3770 @ 3.40GHz×8 with 32GB RAM.

0.1cm along the contact normals.

The results are summarized in Fig. 11. In total, we only observed three failures. In two cases the executed grasps were unstable and the objects slipped, in the third case the object tipped over before all contacts were reached. Overall, the results support our assumption that the system has property P.5. However, in most cases the planner had to be executed several times to find a grasp that was reachable by the robot. This highlights a weakness of the current state of the proposed system. For fast fingertip grasp planning, a system needs to take an object’s environment as well as the robot arm’s kinematics into account.

5 Discussion & Conclusion

In this work, we presented a system that integrates a random forest based heuristic for hand reachability and multi-finger inverse kinematics with the fingertip grasp planner presented in [14]. The training set of the heuristic, the grasping manifold, is incrementally adapted by the system based on the experiences it gathered online. This allows to focus the heuristic’s limited accuracy on the regions of the robot hand’s configuration space that are task relevant. We showed that our system creates grasping manifolds that facilitate the performance of the grasp planner, while being able to generalize and continuously adapt to new types of experiences.

We further extended the proposed system to learn a heuristic from a grasping manifold that integrates both grasp quality and hand reachability, rendering the use of an analytic grasp quality metric unnecessary. As a result, we show that the planner’s runtime is reduced, while it is still capable of generating stable grasps.
5. DISCUSSION & CONCLUSION

Figure 10: Top: As experimental setup we utilize a Schunk-SDH hand mounted on a KUKA-KR5 industrial arm. For localizing objects in the robot’s workspace, we use SimTrack\cite{18} in combination with a Microsoft Kinect\textsuperscript{TM}. Bottom: Stable fingertip grasps executed on the five different objects used in our experiments.

<table>
<thead>
<tr>
<th>Object</th>
<th>#Trials</th>
<th>#Success</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>rivella</td>
<td>7 ± 4.1</td>
<td>3/6</td>
<td>unstable</td>
</tr>
<tr>
<td>glue</td>
<td>9.5 ± 6.0</td>
<td>6/6</td>
<td></td>
</tr>
<tr>
<td>crayola</td>
<td>3.0 ± 3.95</td>
<td>6/6</td>
<td></td>
</tr>
<tr>
<td>toblerone</td>
<td>12.2 ± 10.09</td>
<td>4/6</td>
<td>tip over &amp; unstable</td>
</tr>
<tr>
<td>cheezit</td>
<td>11.0 ± 9.7</td>
<td>6/6</td>
<td></td>
</tr>
</tbody>
</table>

Figure 11: Statistics of running $R^*$ on a real robot system, where six grasps were tested on each object. #Trials: number of grasps planned until a reachable grasp was found. #Success: number of successful pick-and-place executions. Notes: reason of failures.

The system was evaluated both in simulation as well as on a real robot, whereas training was only performed in simulation. Ideally our system would learn its grasping manifold from experienced grasps that were successful in the real world. However, our current implementation of the system does not take the environment nor the kinematics of the robot’s arm into account. As a result, the planner often produces grasps that are not kinematically reachable by the arm or in collision with the environment. Therefore, since the training requires a large number of executions, our evaluation on the real robot was limited to testing a system that was trained in simulation. In future work, we wish to explore the system’s performance when its experiences are gathered in the real world. As the grasping manifold is

learned from successfully executed grasps, it would implicitly emphasize grasps that
have been stable and suppress grasps that have falsely been predicted to be stable.
This may occur due to errors in the used model, such as the simplified modeling of
contacts and unknown friction coefficients. A grasping manifold trained from real
world examples could overcome these limitations of analytic grasp quality metrics.
Moreover, we would like to look into the scalability of the proposed method with
respect to the DoFs of the hand and the range of different object shapes it is able
to cover. Here, it is interesting to investigate the possibility of adaptively evolving
the number of samples used to represent the grasping manifold.

6 Acknowledgments

This work was supported by the European Union framework program FP8-645403
RobDREAM and FP7-288533 ROBOHOW.COG.

References

framework for classification, regression, density estimation, manifold learning and
2012.
[8] S. El Khoury, Miao Li, and A. Billard. Bridging the gap: One shot grasp synthesis
REFERENCES


Hierarchical Fingertip Space: A Unified Framework for Grasp Planning and In-Hand Grasp Adaptation

Published in
IEEE Transactions on Robotics, 2016
Hierarchical Fingertip Space: A Unified Framework for Grasp Planning and In-Hand Grasp Adaptation

Kaiyu Hang, Miao Li, Johannes A. Stork, Yasemin Bekiroglu
Florian T. Pokorny, Aude Billard and Danica Kragic

Abstract

We present a unified framework for grasp planning and in-hand grasp adaptation using visual, tactile and proprioceptive feedback. The main objective of the proposed framework is to enable fingertip grasping by addressing problems of changed weight of the object, slippage and external disturbances. For this purpose, we introduce the Hierarchical Fingertip Space (HFTS) as a representation enabling optimization for both efficient grasp synthesis and online finger gaiting. Grasp synthesis is followed by a grasp adaptation step that consists of both grasp force adaptation through impedance control and regrasping/finger gaiting when the former is not sufficient. Experimental evaluation is conducted on an Allegro hand mounted on a Kuka LWR arm.

1 Introduction

Grasp planning and in-hand grasp adaptation are two complex problems that have commonly been studied separately. Lots of contributions to these problems have been made during the past two decades considering stability modeling and estimation, task based grasping, object representation, grasping synergies and grasp adaptation [52, 9, 55, 64, 44, 47, 3, 24, 32, 8].

In this paper, we present a framework for fingertip grasping considering an integrated approach to grasp planning and in-hand grasp adaptation. The main objective of the framework is to address the problem of grasp instability due to problems such as changed weight of the object, e.g., a container to be filled during grasping, slippage or external disturbances caused by collisions. The framework integrates our previous work of Hierarchical Fingertip Space (HFTS) [23] and grasp adaptation [32], and provides efficient grasp synthesis, grasp force adaptation through impedance control and regrasping/finger gaiting when the former is not sufficient. The approach consists of i) a pre-grasping phase executing grasp synthesis on an efficient representation including both object and hand properties, ii) grasp execution, and iii) a post-grasping phase where tactile feedback and experiences are used for in-hand grasp adaptation, see Fig. 1 and Fig. 2.
Figure 1: A visualization of the proposed Hierarchical Fingertip Space concept: Initial fingertip locations are determined by optimizing grasp stability and adaptability using a hierarchical discretization of the object surface and an impedance controller is used to balance grasping forces. If a large disturbance occurs, the grasp is adapted by fingertip gaiting to maintain grasp stability. The new fingertip location is computed using an optimization in the HFTS.

In the pre-grasping phase, grasp synthesis is formulated as a combinatorial optimization problem considering grasp stability, contact locations and finger gaiting in an integrated manner. In the post-grasping phase, tactile feedback provides information of the stability of the executed grasp. An offline learned probabilistic model is used to assess the grasp stability and initiate an adaptation of grasp forces, followed by finger gaiting if needed. To the best of our knowledge, this is so far the first system that accomplishes grasp synthesis, stability estimation, online replanning and in-hand adaptation in a unified framework.

Compared to the state of the art and our previous work in [23] and [32], our
2. RELATED WORK

The area of robotic grasping includes problems such as grasp stability analysis, grasp synthesis and hand kinematics, object and task representation, grasp adaptation [6, 47, 8] etc. Although each of these have been studied extensively during the past couple of decades there are rather few systems that have addressed grasp synthesis and in-hand grasp adaptation in an integrated manner.

In terms of object representation for grasping, there are many examples of works that rely on encoding shape properties of objects: Reeb Graph [1], Medial Axis [42, 58], hierarchical box decomposition [29], super-quadrics [37, 7, 13, 5]. More recent work demonstrates topological analysis of shape for grasping and caging [54, 40]. Our Hierarchical Fingertip Space (HFTS) proposes a method for shape

integrated system:

- provides an optimization framework for both grasp synthesis and finger gaiting;
- closes the loop between grasp planning and control through stability estimation and finger gaiting;
- optimizes grasp adaptability and demonstrates informed finger gaiting optimization by considering viable hand configurations and object shape knowledge.

We review the related work in Sec. 2 and present the methodology in Sec. 3 - Sec. 5. We evaluate in Sec. 6 and then conclude in Sec. 7.

2 Related Work

The area of robotic grasping includes problems such as grasp stability analysis, grasp synthesis and hand kinematics, object and task representation, grasp adaptation [6, 47, 8] etc. Although each of these have been studied extensively during the past couple of decades there are rather few systems that have addressed grasp synthesis and in-hand grasp adaptation in an integrated manner.

In terms of object representation for grasping, there are many examples of works that rely on encoding shape properties of objects: Reeb Graph [1], Medial Axis [42, 58], hierarchical box decomposition [29], super-quadrics [37, 7, 13, 5]. More recent work demonstrates topological analysis of shape for grasping and caging [54, 40]. Our Hierarchical Fingertip Space (HFTS) proposes a method for shape
representation that encodes both the global and local geometric properties of the object.

Classical works formulate contact-level grasp synthesis as an optimization problem \[6, 23, 10, 35, 50, 45, 18\] for which the objective — grasp stability — is commonly measured using force analysis in the contact wrench space \[17\]. The problem of calculating feasible hand configurations has also been addressed in this context \[9, 10\]. To account for uncertainties in physical properties of objects, grasp friction sensitivity \[22\] and independent contact regions \[43\] have been investigated.

Our approach formulates fingertip grasping as an optimization problem considering grasp stability, adaptability and hand reachability to prepare a grasp for future adaptive execution against physical uncertainties.

Approaches to force based grasp control range from geometry based analytic methods \[25, 11, 38\] to learning-based frameworks for force optimization \[62, 61\]. In-hand manipulation has been addressed as finger gaiting with a rolling contact model and quasi-static assumption \[20, 49\]. Hybrid position and force control has also been studied \[33, 65, 12, 57\] as well as impedance control \[51, 34, 56, 31\].

Our approach allows for grasp stabilization through both contact force adaptation and finger gaiting planned in real-time using tactile feedback and the proposed Hierarchical Fingertip Space.

In realistic tasks, the ability to maintain a stable grasp on an object is an integral property of robust systems. A grasp that is originally stable may be perturbed while performing a manipulation with the held object. This is also valid for cases where some properties of the object change - weight can change if a glass held by the robot gets filled, environmental changes can affect friction coefficients, collision may cause slippage, etc. In addition, many of these properties may not be precisely known to start with. Thus, in-hand grasp adaptation may be needed after a grasp has been applied on an object. For this purpose, relying on visual feedback is not sufficient and many of the recent approaches facilitate haptic and proprioceptive information \[26, 27, 36, 66, 14, 15, 4, 3\]. Finger gaiting may be further required when applying higher grasping force does not suffice \[48, 32\]. Our work here builds upon \[48, 32\] and additionally allows for replanning during grasp execution.

3 Hierarchical Fingertip Space and Grasp Optimization

We start by providing a list of notations used in the paper:

In the pre-grasping phase, we formulate fingertip grasp synthesis as an optimization problem considering each object represented as a point cloud \( P = \{ p_i \in \mathbb{R}^3 \mid i \in \{1, ..., n_p\} \} \). We seek \( m \) contact locations, \( C_g = \{ c_1, ..., c_m \mid c_i \in P \} \), on the object surface and a hand configuration, \( J_g \in \mathbb{R}^d \) where \( d \) are controlled joint angles.

We define two concepts: Fingertip Space and Hierarchical Fingertip Space (HFTS). Fingertip Space represents a finite set of contacts on an object surface that are locally flat and large enough for a fingertip \[23\]. We denote the Fingertip Space as
3. HIERARCHICAL FINGERTIP SPACE AND GRASP OPTIMIZATION

Φ(P) = {φ1, ..., φn_f} ⊂ P and an element of this set φi is called a **Fingertip Unit**. Fingertip Space Φ(P) is parametrized by locations and normals of Fingertip Units. We extract the Φ(P) from P based on the surface curvature estimated from a set of points N_r(p_i) ⊂ P within one fingertip size r, around a potential contact p_i. The fingertip space of P is given by

\[ \Phi(P) = \{ \varphi_i | K(N_r(p_i)) \leq \kappa, \varphi_i \in P \} \tag{1} \]

where K(N_r(p_i)) is the local surface curvature estimated from N_r(p_i) and \( \kappa \in \mathbb{R} \) is the empirically determined curvature threshold. In the rest of this paper, we write \( \Phi \) instead of \( \Phi(P) \). Fig. 3(left) shows an example of **Fingertip Space**. To enable finger gaiting, we want our Fingertip Space to encode the space around each Fingertip Unit in an efficient manner. To achieve this, we put a penalty term on admissible regions using a logistic function. Let c(φ_i) ∈ P \ Φ be the closest point to Fingertip Unit φ_i that has been rejected by Eq. (1), the penalty \( w_i \) for φ_i is computed as:

\[ w_i = \frac{1}{1 + e^{-\gamma \|\tilde{c}(\phi_i) - \tilde{c}(\phi_i)\|}} \tag{2} \]

where \( \gamma \in \mathbb{R}^+ \) is an elasticity factor, see Fig. 3(left).

---

**Table 1: List of notations used in the paper**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P \subset \mathbb{R}^3 )</td>
<td>An object's point cloud</td>
</tr>
<tr>
<td>( C_g = { c_1, ..., c_m</td>
<td>c_i \in P } )</td>
</tr>
<tr>
<td>( J_g \subset \mathbb{R}^d )</td>
<td>A hand configuration with d DoFs</td>
</tr>
<tr>
<td>( \Phi(P) = { \varphi_1, ..., \varphi_n_f } \subset P )</td>
<td>Fingertip Space built from P</td>
</tr>
<tr>
<td>( \varphi_i \in \Phi(P) )</td>
<td>A Fingertip Unit</td>
</tr>
<tr>
<td>( w_i \in \mathbb{R} )</td>
<td>A hierarchy of surrogates of ( \Phi )</td>
</tr>
<tr>
<td>( \ell_g = (E_g, V_g) )</td>
<td>The i-th surrogate approximation of ( \Phi )</td>
</tr>
<tr>
<td>( h_{\text{hop}}(\varphi_i, \varphi_j) )</td>
<td>A node representing m contacts in ( \Lambda_\varphi )</td>
</tr>
<tr>
<td>( \lambda_g \in \mathbb{R} )</td>
<td>A grasp defined by ( \lambda_g ) and ( J_g )</td>
</tr>
<tr>
<td>( q(\lambda_g) \in \mathbb{R} )</td>
<td>Grasp quality defined in [17]</td>
</tr>
<tr>
<td>( R(\lambda_g) \in \mathbb{R} )</td>
<td>Grasp reachability residual</td>
</tr>
<tr>
<td>( \Lambda_\varphi \in \mathbb{R} )</td>
<td>Grasp adaptability</td>
</tr>
<tr>
<td>( \theta(\varphi) \in \mathbb{R} )</td>
<td>Grasp synthesis objective function</td>
</tr>
<tr>
<td>( V_\varphi, E_\varphi )</td>
<td>Gaussian Mixture Model learned over ( \Lambda_\varphi ) for grasp estimation and adaptation</td>
</tr>
<tr>
<td>( \tilde{g} )</td>
<td>A grasp with stiffness ( K ), rest length ( L ), and tactile readings ( S )</td>
</tr>
<tr>
<td>( \Theta^* \lambda_g \in \mathbb{R} )</td>
<td>Virtual frame for grasp impedance control</td>
</tr>
</tbody>
</table>

---

\(^1\)For the SynTouch sensor used in this work, the fingertip size \( r \) is 14mm, http://www.syntouchllc.com/
3.1 Multilevel refinement of Fingertip Space

Given the large number of Fingertip Units per object, formalizing grasp optimization on all combinations of these is computationally impractical. A feasible strategy is to apply Surrogate Models or multilevel refinement, that recursively approximate the original optimization problem in a hierarchy of simpler, more tractable problems i.e. surrogate models. We first explain a representation for single fingertip contact optimization and then continue with the definition of Hierarchical Fingertip Space for multiple fingertip contacts.

Surrogate approximation of $\Phi$ is constructed by recursively grouping Fingertip Units by cluster analysis using geometric properties. For the optimization of a single contact in $\Phi$ we construct a hierarchy of surrogate approximations of $\Phi$ (see Fig. 4) as a similarity-based graph $G_\Phi = (E_\Phi, V_\Phi)$, with the hierarchy levels $i \in \{0, \ldots, l - 1\}$ representing different scales of surrogate approximations. $\Phi$ is recursively partitioned into smaller sets of fingertip units, denoted as $\hat{\varphi}_{i,j} \subset \Phi$, and is represented as a node $\varphi_{i,j} \in V_\Phi$ in graph $G_\Phi$, where $i$ is the level of $\varphi_{i,j}$ in the hierarchy and $j$ is the index of the partition in level $i$. We partition the set $\Phi$ in a top-down manner, with parent $\varphi_{i,j}$ nodes split into children nodes if $|\hat{\varphi}_{i,j}| > 1$. Ultimately, the bottom level of $G_\Phi$ consists of nodes representing single fingertip units, $|\hat{\varphi}_{0,j}| = 1$. Experimentally and as shown in Fig. 4, the number of partitioning centers for the level $l - 2$ is set to 20 and in the remaining levels to 4 similar to [24].

In the process described above, we require a method for cluster analysis, that fulfills the following properties: 

a) The method must be able to group Fingertip Units according to relevant geometric properties. In more detail, the employed
3. **Hierarchical Fingertip Space and Grasp Optimization**

The hierarchy of fingertip units is represented as a graph in Figure 4. Fingertip unit partitions are represented as nodes in different levels of the hierarchy and the connectivities in this graph are represented by edges. Extra connectivities defined by Eq. (3) are exemplified in level $l-3$ for $\phi_{l-3,1}$, with red edges for 2 hops and blue edges for 4 hops.

Similarity measure has to be based on the grasp relevant properties. For point contact with friction, this is captured by position and normal information [10].

- **b)** The recursive grouping in each hierarchy level must result in partitions having similar variance in relevant geometric properties.
- **c)** The individual clusters should correspond to connected and compact surface areas such that their average elements represent possible contact locations. These requirements may initially be violated on higher levels but they become increasingly important for lower hierarchy levels. Any clustering method that fulfills these requirement can be employed to construct $G_\Phi$, e.g., in our earlier work, we refer to Agglomerative Hierarchical Clustering with complete-linkage [23], which is sensitive to noise.

Given real sensor data, there is noise associated with the computation of surface normals. To address this, we employ a Gaussian process (GP) based filter with Thin Plate Spline kernel. This approach fulfills the requirement stated above by integrating both position and normal information into similarity measure [53].

Higher sampling frequency for GP centers is used in areas of higher curvature, see Fig. 3 (middle). The distribution of centers captures the geometric similarities (locations and normals) and therefore relate to the similarities in the grasp wrench space [39]. GP partitioning is regulated using a threshold $T_p$, so that if $|\hat{\phi}_{i,j}| \leq T_p$, a node is not further divided by GP partitioning but it is split up into all its fingertip
units. Nodes consisting of single fingertip units are copied to the next level as long as some other nodes can be partitioned. This guarantees a balanced partitioning tree, and hence a valid surrogate approximation for every level in the hierarchy. As discrete optimization relies on relevant neighbors in the solution space, we introduce connectivity by introducing extra edges between nodes in the same level into $E_{\Phi}$. More precisely, the extended edge set consists of parent-child edges and extra-edges $E_{\Phi} = E^P_{\Phi} \cup E^E_{\Phi}$ which are given as:

$$E^P_{\Phi} = \{ \{ \varphi_{i,j}, \varphi_{i-1,k} \} \in V_{\Phi} \times V_{\Phi} | \hat{\varphi}_{i-1,k} \subseteq \hat{\varphi}_{i,j} \}$$

$$E^E_{\Phi} = \{ \{ \varphi_{i,j}, \varphi_{i,k} \} \in V_{\Phi} \times V_{\Phi} | \text{hop}(\varphi_{i,j}, \varphi_{i,k}) \leq h \}$$

(3)

The function $\text{hop}(\varphi_{i,j}, \varphi_{i,k})$ denotes the hop distance between $\varphi_{i,j}$ and $\varphi_{i,k}$ along edges in $E^P_{\Phi}$. The hop limit $h \in \mathbb{N}$ defines the size of the neighborhood and is set to 4 in our experiments, resulting in neighborhoods of size e.g., ca. 4cm in the second top level. Using the definitions above, we can now define the $i$-th surrogate approximation of the Fingertip Space $\Phi$ as:

$$(G_{\Phi})_i = ((V_{\Phi})_i, (E_{\Phi})_i)$$

$$(V_{\Phi})_i = \bigcup_j \{ \varphi_{i,j} \}$$

$$(E_{\Phi})_i = \{ \{ \varphi_{i,j}, \varphi_{i,k} \} | \{ \varphi_{i,j}, \varphi_{i,k} \} \in E_{\Phi} \}$$

(4)

which is a subgraph of $G_{\Phi}$ and an approximation at the $i$-th resolution level.

We define the mean location and orientation of the set of fingertip units contained in the partition $\hat{\varphi}_{i,j}$ as $p(\hat{\varphi}_{i,j}) \in \mathbb{R}^3$ and $n(\hat{\varphi}_{i,j}) \in \mathbb{R}^3$, this will be used later for stability analysis. In terms of Eq. (2), the penalty assigned to a node $\varphi_{i,j}$ is defined as:

$$w_{i,j} = \frac{1}{|\hat{\varphi}_{i,j}|} \sum_{\varphi_k \in \hat{\varphi}_{i,j}} w_k$$

(5)

Given the hierarchy $G_{\Phi}$ of surrogate approximation models, we can optimize a fingertip location in a top-down manner. By optimizing the contact in a coarse to fine fashion, a final contact will be found in the bottom level of the hierarchy. Next, we investigate the grasp synthesis with multiple contacts.

### 3.2 Hierarchical Fingertip Space

In the previous section, we introduced the similarity-based graph $G_{\Phi}$ for a single fingertip. For $m$ fingertips, we define the product graph $\Lambda_{\Phi} = (V_{\Lambda}, E_{\Lambda})$ named Hierarchical Fingertip Space (HFTS) as in Eq. (6). Thus, nodes in $V_{\Lambda}$ represent combinations of $m$ contacts, $\lambda_{i,j} = (\varphi_{i,j}^1, ..., \varphi_{i,j}^m)$, and the graph-distance between nodes in the same level reflects the similarity of the individual contacts. Formally, the HFTS is defined as:

$$\Lambda_{\Phi} = G_{\Phi}^1 \times \cdots \times G_{\Phi}^m$$

$$V_{\Lambda} = \{ \lambda_{i,j} = (\varphi_{i,j}^1, ..., \varphi_{i,j}^m) | \varphi_{i,j}^k \in (V_{\Phi})_k \}$$

(6)
3. HIERARCHICAL FINGERTIP SPACE AND GRASP OPTIMIZATION

where $G^k = (V^k, E^k)$, $k \in \{1, \ldots, m\}$ is the surrogate hierarchy for the $k$-th fingertip. The penalty value for a set of contacts is defined as minimum of all individual contact penalties:

$$w^*_{i,j} = \min\{w_{i,j_1}, \ldots, w_{i,j_m}\}$$ (7)

Optimization on $\Lambda\Phi$ requires definition of neighborhoods and we define two types of edges for $E^\Lambda$: 1) Edges between nodes and their parent, $E^E^\Lambda$, such that $\Lambda\Phi$ inherits the hierarchy levels from the individual $G^k\Phi$, and 2) edges between nodes in the same level, $E^P^\Lambda$, for which the individual contacts are identical or neighbors in their graph $G^k\Phi$, respectively. Formally, we obtain $E^\Lambda = E^E^\Lambda \cup E^P^\Lambda$:

$$E^E^\Lambda = \{\{\lambda_{i,j_1}, \lambda_{i-1,j_2}\} | \forall k: \{\lambda^{(k)}_{i,j_1}, \lambda^{(k)}_{i-1,j_2}\} \in (E^E^k)\}$$ (8)

where $\lambda_{i,j_1} \in V^k$ is the $k$-th item of tuple $\lambda_{i,j}$. Similarly to the surrogate models for a single fingertip contact, we define the $i$-th surrogate approximation of multiple fingertip grasping in HFTS as:

$$\Lambda_{i} = ((V\Lambda)_i, (E\Lambda)_i)$$

$$(V\Lambda)_i = \bigcup_j \{\lambda_{i,j}\}$$ (9)

$$(E\Lambda)_i = \{\{\lambda_{i,j_1}, \lambda_{i,j_2}\} | \{\lambda_{i,j_1}, \lambda_{i,j_2}\} \in E\Lambda\}$$

3.3 Grasp Optimization in HFTS

So far, we described the solution space for grasp synthesis using nodes $\lambda_g \in \Lambda\Phi$ from different levels, which are combinations of contacts on the object surface. However, to realize the contacts with a robot hand, we additionally need the joint angles $J_g \in \mathbb{R}^d$. A valid grasp solution, $g = (\lambda_g, J_g)$, is a combination of contact positions and joint angles.

3.3.1 Grasp Stability

During the pre-grasping phase, when we synthesize a grasp, only visual information of object is available and we need to evaluate or predict grasp stability without feedback. This can be done using contact based force closure analysis [17]: Given a grasp solution $g$, the grasp quality measure $Q(\lambda_g) \in \mathbb{R}$ is the minimum offset between the origin of the wrench space and facets of the convex hull spanned by friction cones of contacts parametrized by positions and normals [10]. The value is positive when the grasp is force closed and larger for more stable grasps.

3.3.2 Grasp Reachability

Not all combinations of contacts $\lambda_g$ can be realized by a given robotic hand and we can classify contacts into reachable or unreachable using a function $R^*: V\Lambda \rightarrow \{0, 1\}$
so that the optimization can be constrained to reachable grasps with $R^*(\lambda_g) = 0$. Since a robotic hand can have many degrees of freedom and complicated coupled kinematics, it can be too costly to analytically compute $R^*(\lambda_g)$ in each optimization step. For this, various forms of constraints have been formulated \cite{16, 28}. To achieve required speed and precision, we linearly relax it to a measure of dissimilarity between $\lambda_g$ and the closest known reachable contacts $\lambda^*_g$ of grasp solution $g^* = (\lambda^*_g, J^*_g)$. The reachability measure of $\lambda_g$ is then reformulated as a residual $R(\lambda_g) \in \mathbb{R}^+$:

$$R(\lambda_g) = \| C(\lambda_g) - C(\lambda^*_g) \|$$  \hspace{1cm} (10)

where $C(\cdot) \in \mathbb{R}^{6 \times (m-2)}$ is an affine invariant encoding of $m$ contacts in terms of its contact locations and normals \cite{21}. Note that a smaller residual indicates more reachable contacts.

To generate a set of viable grasps, we randomly sample hand configurations and save the encoded contacts and corresponding hand configuration $J_g$ into a $k$-d tree like data structure $T$ offline with the query time $O(n \log n)$. Using $T$, we can compute the residual by lookup and find the hand configuration for realizing the contacts if the residual was small.

$$T: \lambda_g \mapsto (J^*_g, R(\lambda_g))$$  \hspace{1cm} (11)

### 3.3.3 Grasp Adaptability

We use grasp adaptability to enable finger gaiting already in the grasp synthesis stage. By decomposing the hand Jacobian and calculating the manipulability \cite{63} of a hand configuration in the tangential plane of contacts, we measure the adaptability of a grasp, denoted as $A(J_g) \in \mathbb{R}^+$. Concretely, given the Jacobian $J_f(J_g) \in \mathbb{R}^{3 \times n}$ and the normal $n_f \in \mathbb{R}^3$ of fingertip $f$, the Jacobian can be rotated by $R_f \in \mathbb{R}^{3 \times 3}$ such that the last row of $\tilde{J}_f(J_g) = R_f J_f(J_g)$ corresponds to the movement of fingertip in the direction of $n_f$. The first two rows of $\tilde{J}_f$, denoted by $\tilde{J}_f(J_g) \in \mathbb{R}^{2 \times n}$, are then the projection of $J_f$ in the tangential plane of the fingertip normal.

$$A(J_g) = \sum_f \sqrt{\det(\tilde{J}_f(J_g)\tilde{J}_f^T(J_g))}$$  \hspace{1cm} (12)

Note that we can assume that the fingertip normal (on the robot hand) and the fingertip unit normal will be similar when the grasp is realized if $R(\lambda_g)$ is small. An example of grasp adaptability measure is shown in Fig. 5. Since this measure is hand configuration based, it is affine invariant, and hence grasp pose independent.

In order to capture grasp stability, reachability and adaptability in the grasp optimization, the optimization objective is defined as follows:

```
Priority 1: Maximize $\theta(g)$  \hspace{1cm} (13)
Priority 2: Maximize $A(J_g)$  \hspace{1cm} (14)
```
3. HIERARCHICAL FINGERTIP SPACE AND GRASP OPTIMIZATION

Figure 5: Grasp Adaptability for fingertip 1 (red). Adaptability is computed for fingertip positions sampled in joint space. The colored volume shows finger adaptability values at sampled fingertip positions.

with

$$\theta(g) = Q(\lambda_g) - \alpha R(\lambda_g), \quad \alpha \in \mathbb{R}^+$$

(15)

where $\alpha$ is a weighing factor to account for the hand size, which is determined by the range in which the grasp quality values $Q(\lambda_g)$ vary, as it is related to the grasp sizes [17]. To optimize the second objective, we use a sorted lookup table for $R(\lambda_g)$ which returns the most adaptable joint configuration in the area of the best grasp according to $A(\lambda_g)$ [19, 2] when querying reachability residuals (line 7 and 10 in Alg. 7). As we can see in Fig. 6, for the same contact locations, there can be multiple hand configurations for realizing it, however, our prioritized lookup table will always return the hand configuration with the best adaptability.

Having defined the objective function, we can now proceed to grasp synthesis. Using a surrogate-based optimization metaheuristic, we need to find solutions on each of the surrogate approximations and extend them to the next model. For optimization in each model, we adopt stochastic hill climbing which can escape from local optima by means of randomness. Switching from solution $g$ to $g'$ is determined by the probabilistic function in Eq. (16):

$$Pr(g, g') = \left(1 + \exp \frac{w_g \theta(g) - w_{g'} \theta(g')}{\zeta}\right)^{-1}$$

(16)
Figure 6: Grasps with same contacts and different adaptabilities: the left grasp has the highest adaptability.

where $w_g$ is the penalty assigned to a tuple of contacts defined by Eq. (7). The randomness in the optimization is determined by $\zeta$, it makes the optimization more random when a large value is chosen, while it behaves more like pure hill climbing if a small value is applied. The grasp optimization algorithm is shown in Alg. 7.

\begin{algorithm}
\caption{Surrogate-Based Optimization in HFTS}
\begin{algorithmic}[1]
\STATE {Input: stopCondition, $\Lambda_\Phi$, maxIter}
\STATE {Output: grasp $g = (\lambda_y, J_g)$}
\FOR {$i = l - 1 \to 0$}
\IF {$i = l - 1$} \Comment{Initialization}
\STATE $\lambda_y \leftarrow \text{random from } (\Lambda_\Phi)_i$
\ELSE
\STATE $\lambda_y \leftarrow \text{random child of } \lambda_g$
\ENDIF
\FOR {$1 \to \text{maxIter}$} \Comment{Optimize on Surrogate}
\STATE $\lambda_y' \leftarrow \text{random neighbor of } \lambda_y \in (\Lambda_\Phi)_i$
\STATE $(J_g', R(\lambda_y')) \leftarrow T(\lambda_y')$
\IF {$\Pr(g, g') \geq \text{rand}(0,1)$} \Comment{Good Solution}
\STATE $g \leftarrow g'$
\ENDIF
\ENDFOR \Comment{Extend to Lower Surrogate}
\IF {stopCondition($g$)} \Comment{Good Solution}
\STATE break
\ENDIF
\ENDFOR
\end{algorithmic}
\end{algorithm}

For realizing the grasp, we can transform the hand base to the pose where the fingertips meet the contact locations $[23]$. In cases when the final reachability residual $R(\lambda_y) \neq 0$, a local optimization of joint configuration by linear interpolation
is required to realize desired contacts. To avoid too small and time consuming incremental improvements at each level, we utilize a stopCondition. It can be set to false if we want to explore the space until convergence or we control the number of iterations by setting a threshold for the optimization function in Eq. (15).

4 Grasp adaptation

A synthesized grasp is executed using a simple position control. When contacts are made and tactile readings are available, an object-level impedance controller is used to regulate grasp forces. The object-level impedance control for dexterous robotic hands is still an open question and is currently feasible for 3 fingers or 4 fingers with virtual linkage. For the demonstration of the entire system, although we have shown that we are able to plan grasps for m fingertips, we will in the rest of the paper explain the control and adaptation of grasps by examples of only 3 fingers.

The grasp impedance controller is formulated in a virtual frame (VF) defined in terms of fingertip locations as

\[
R_o = [v_x, v_y, v_z] \in \text{SO}(3)
\]

\[
v_x = \frac{p_3 - p_1}{\|p_3 - p_1\|}
\]

\[
v_z = \frac{(p_2 - p_1) \times v_x}{\| (p_2 - p_1) \times v_x \|}
\]

\[
v_y = v_z \times v_x
\]

(17)

where \(p_1, p_2\) and \(p_3\) are locations of the fingertips, see Fig. 7.

A grasp in the VF is denoted \(\hat{g} = (K, L, S)\) where \(K = (K_x, K_y, K_z) \in \mathbb{R}^3\) is the grasp stiffness and \(L = (L_1, L_2, L_3) \in \mathbb{R}^3\) is the grasp rest length, i.e. the distance between each fingertip and the center of VF. \(S = (S_1, S_2, S_3) \in \mathbb{R}^{57}\) denotes the tactile readings, in our case from SynTouch sensors.

Grasp stability is monitored using a probabilistic representation relying on a Gaussian Mixture Model \(\Theta\) that is trained offline, see Fig. 8. As described in detail in our previous work, \(\Theta\) is trained over \(K, L, S\) parameters for a variety of objects. Given \(\Theta\), grasp stability is estimated by

\[
p(\hat{g}|\Theta) = \sum_{i=1}^{n_g} \pi_i \mathcal{N}(\hat{g} | \mu_i, \Sigma_i)
\]

(18)

where \(n_g\) is the number of Gaussian components, each of which has a prior \(\pi_i\). \(\mathcal{N}(\hat{g} | \mu_i, \Sigma_i)\) is the Gaussian distribution with mean \(\mu_i\) and covariance \(\Sigma_i\).

A grasp \(\hat{g}\) is considered unstable if the log likelihood of Eq. (18) is smaller than a predefined threshold determined by the ROC curve. If a grasp \(\hat{g}\) is unstable, we compute its Mahalanobis distance to each component in \(\Theta\) and denote
the minimum distance as $m_d$. If $m_d$ is within two standard deviations, we apply force adaptation by changing stiffness $K$ to the value obtained by computing the maximum expectation of $K$ conditioned on $L$ and $S$. The details of this process have been described in detail in our previous work [32]. Otherwise, a finger gaiting
strategy is initiated as explained in detail in next section.

5 Regrasping by finger gaiting

Stiffness adaptation is not enough in cases when there is an upper bound on the force that can be exerted by the hand. Thus, to stabilize a grasp, the system initiates finger gaiting. Finger gaiting is defined as an optimization problem based on the current rest length \( L \) represented in VF:

\[
\theta^*(\lambda_g) = \|L - L^*\| + \beta R(\lambda_g)
\]  

(19)

where \( R(\lambda_g) \) is the reachability defined in Eq. \( [10] \), \( \beta \in \mathbb{R}^+ \) is a weighing factor to account for the hand size, as \( L \) values range differently in terms of hand sizes. \( L^* \) is the desired rest length obtained from the closest Gaussian center \( \hat{g}^* = (K^*, L^*, S^*) \) in terms of \( m_d \). The reasoning above is to find the closest stable and reachable grasp to the current one, taking into account the current tactile readings.

For the robot hand we use in this work, we can only re-locate fingertip \( F_1 \) or \( F_2 \), as shown in Fig. 7, since relocating the thumb \( F_3 \) leaves the grasp without contacts on the opposite side of the object. Our strategy of choosing between \( F_1 \) and \( F_2 \) is straightforward: we compute the optimization for \( F_1 \) and \( F_2 \) in parallel for minimizing the objective value from Eq. (19), the one resulted with smaller values is chosen. Our optimization procedure employs breadth-first search in \( \Lambda_\Phi \) starting from the initial contact. The search is terminated in a branch if the reachability measure grows beyond a predefined threshold \( \epsilon_R \). Since we move only one finger, we need an additional rule:

\[
Prune(\lambda_g, \lambda_g', f_o) = \begin{cases} 
True, & \exists i: i \neq f_o \land \lambda_g^{(i)} \neq \lambda_g'^{(i)} \\
False, & \text{otherwise.}
\end{cases}
\]  

(20)
where $f_o$ is the fingertip to be relocated, $\lambda_g$ is the node that represents the current grasp contacts, and $\lambda_g'$ is the new solution. Since the search fringe can go upwards in the hierarchy graph $\Lambda_\Phi$, this rule asserts that only a single fingertip is moved while the remaining two are fixed. The main idea is sketched in Fig. 9 and the procedure summarized in Alg. 8. Note that it includes the penalty factor from Eq. (7).

**Algorithm 8** Fingertip Gaiting by Optimization in $\Lambda_\Phi$

### 5.1 Fingertip Gaiting in Practice

When grasp stability changes rapidly and finger gaiting is triggered frequently, to avoid switching between impedance and position control, we stay in impedance control mode during finger gaiting by sliding the finger to the desired position. To allow this, we formulate fingertip gaiting using impedance controller defined in Vf. A virtual spring with stiffness $k$ is defined to connect the current location of the moved fingertip and $\hat{p}$, which is equivalent to a fingertip impedance controller superimposed on the original grasp controller. An example of fingertip F1 gaiting is depicted as in Fig. 7.

The stiffness $k$ of the virtual spring is determined by the distance $d_{\hat{p}} \in \mathbb{R}$ between the fingertip’s current location and $\hat{p}$, and an empirical parameter $\Gamma \in \mathbb{R}$ as: $k = d_{\hat{p}} \Gamma$. In this way, the fingertip will be slid towards $\hat{p}$ while keeping the contact on the object. Since $\hat{p}$ is computed in the HFTS, we ensure that the desired position is on the object surface. If a new goal position is requested during finger gaiting, the system will either continue to the new position if the same fingertip is concerned, or stop the current gaiting and initiate gaiting with another fingertip. An example situation is depicted in Fig. 10 where fingertip F2 stopped moving before the desired position is reached, since the grasp was estimated as stable on the way.
6 Experimental evaluation

We perform experimental evaluations with an Allegro hand mounted on a Kuka LWR arm. The hand is equipped with SynTouch tactile sensors on three fingertips. The systems performance is evaluated using six objects shown in Fig. 11 which are tracked using the OptiTrack\textsuperscript{2} real-time motion tracking system. The evaluations presented below demonstrate the performance of the grasp synthesis system alone as well as the integrated system for grasp adaptation.

6.1 Grasp Synthesis

Grasp synthesis is performed on a point cloud representation of objects obtained offline. We also generated a reachability table with $10^6$ hand configurations using rejection sampling: configurations are first uniformly sampled in the hand joint space and we keep those collision-free configurations with adaptabilities larger than 0.02, which is determined empirically since we observed that the grasps are rarely adaptable with adaptabilities lower than 0.02.

Alg.\textsuperscript{7} generates both contact locations and hand configurations. Simple position-based control is used to execute a grasp \cite{32}. A few examples are shown in Fig. 11 and Fig. 12.

For evaluating the performance of the grasp planner, we repeat the grasp optimization according to Alg.\textsuperscript{7} for each test object. In order to keep an equal number of iterations for each repetition of the algorithm, we set $maxIter = 100$ and $stopCondition(g) = false$. For each object, we run the algorithm with random initialization until we achieve 100 stable and collision free grasps. Evaluation results are summarized into a table shown in Fig. 13.

First, Fig. 13 shows that the number of levels of the graph $G_\Phi$ are between 4 and 5 when $T_p = 10$, or between 3 and 4 when $T_p = 40$. This indicates that our

\textsuperscript{2}http://www.naturalpoint.com/optitrack/
Figure 11: Six example objects used in the evaluation: there is both variation in global geometry as well as local surface properties. From top-left to bottom-right: bottle1, bottle2, jug, rivella, milk and spray.

Figure 12: Example grasps generated by Alg. 7 with stopCondition($g$) that, as soon as the grasp is stable and the reachability residual is smaller than 0.006, we stop optimizing on the current level of $\Lambda_{Q}$ and continue on the next level.

system produces similar depth of the HFTS independent of the shape of the object. However, the shape of the object affects the number of nodes at each level, given that some branches are terminated earlier for objects of simpler geometry, such as the milk package. It is worthwhile to note that, the partitioning of rivella ended up with more levels than milk, and this is reverted when it was $T_p = 40$. This is due to the fact that if one sets a small threshold $T_p$, a larger sub-partition would continue being partitioned, where as smaller ones are terminated earlier. This causes the
rivella to have more levels when \( T_p \) is smaller, due to its uneven sub-partitioning in the lower levels of the hierarchy.

Regarding the success rate (SR), we can see that SR lies at approximately 90%. Fig. 14 shows average adaptabilities for the 100 stable grasps for each object. Average adaptability values, computed by Eq. (12), are large showing that our methodology considers the adaptability effectively.

### 6.2 Grasp Adaptation

Once a grasp is executed and contacts established, the system will enter the post-grasping phase and start monitoring the stability based on tactile feedback. Instead of position control, the impedance controller is used to control the grasp using GMM based model \( \Theta \). The log likelihood threshold for Eq. (18) is set to \(-100\) in terms of the ROC curve with a false positive rate \( FPR = 15\% \) [32]. For the force control of the hand, we set the initial grasp stiffness \( K = (12, 2, 2) \) and use it for the execution of all grasps, as described in Sec. 4.

For the evaluation, we run two sets of experiments: 1) We continuously increase the objects’ weight by filling them to evaluate the maximum weight each grasp can withstand, and 2) we shake the grasped objects by linearly increasing acceleration in different directions to evaluate the maximum acceleration each grasp can withstand. For comparison, we conduct the same experiments without any grasp adaptation and on the system proposed in [32] which does not consider object shape information when relocating fingertips.
Figure 14: *Upper:* Grasp adaptability distribution of $10^6$ hand configurations in the reachability lookup table. *Lower:* Average grasp adaptabilities for the 100 grasps generated in the evaluations for all objects.

### 6.2.1 Testing maximum weight

For each object, we execute the best out of 100 grasps generated in Sec. 6.1 and align the object with vertical axis as shown in Fig. 18. We then gradually fill object with black pepper beans and record the maximum weight the grasp can withstand. The maximum weight is reached when the stability estimator predicts unstable grasp for more than 2 seconds or if the object drops. We repeat this test for each grasp 5 times and summarize the results in Fig. 15.

Naturally, the system without any adaptation performs the worst and the integrated system outperforms the system from [32]. This is since our system: i) takes into account grasp reachability during the exploration, and ii) the new location is computed in the HFTS, thus ensuring it is valid, avoiding problems shown in Fig. 16, and iii) considers two fingers for gaiting, resulting in increased flexibility.

A quantitative evaluation of the proposed system and the system in [32] has
6. EXPERIMENTAL EVALUATION

<table>
<thead>
<tr>
<th>Object</th>
<th>Weight</th>
<th>Without</th>
<th>With [32]</th>
<th>Improved</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottle1</td>
<td>34</td>
<td>55.1 ± 7.11</td>
<td>153.1 ± 12.31</td>
<td>165.3 ± 13.27</td>
</tr>
<tr>
<td>bottle2</td>
<td>39</td>
<td>62.8 ± 6.63</td>
<td>102.3 ± 13.38</td>
<td>121.3 ± 9.91</td>
</tr>
<tr>
<td>jug</td>
<td>112</td>
<td>125.3 ± 14.90</td>
<td>147.4 ± 9.62</td>
<td>162.1 ± 13.12</td>
</tr>
<tr>
<td>rivella</td>
<td>24</td>
<td>36.0 ± 6.96</td>
<td>76.5 ± 9.4</td>
<td>92.7 ± 7.45</td>
</tr>
<tr>
<td>milk</td>
<td>34</td>
<td>63.5 ± 8.20</td>
<td>151.8 ± 7.24</td>
<td>157.4 ± 8.35</td>
</tr>
<tr>
<td>spray</td>
<td>61</td>
<td>75.7 ± 7.21</td>
<td>102.2 ± 6.02</td>
<td>121.6 ± 7.15</td>
</tr>
</tbody>
</table>

Figure 15: The comparison of the supported object weights (mean ± std, Unit: gram).

- **without**: without grasp adaptation;
- **with [32]**: with grasp adaptation in [32];
- **improved**: the adaptation approach proposed in this paper.

Figure 16: The risk of moving a fingertip to an non-existing position present in [32] is addressed by using our HFTS representation. The red point shows the fingertip position before gaiting.

been conducted with respect to optimization residual. We first execute the grasp in simulation and then trigger the fingertip gaiting by sending desired rest lengths randomly sampled around the current values within a ball of radius 20\text{mm}. The result is shown in Fig. 17. Due to the object shape constraint, the systems cannot provide zero residuals. Our system performs much better for non-planar objects given that HFTS representation considers shape in an effective way.

An example of the supported weight test for the rivella bottle is shown in Fig. 18. In the beginning when the object is not too heavy, the likelihood \( p(\hat{g} | \Theta) \) is larger than \(-100\) and the grasp stiffness \( K \) is constant. As the weight increases, the grasp becomes unstable and stiffness adaptation is initiated. Stiffness changes rapidly when the weight increases, and when the force adaptation is not able to handle the current weight, a finger gaiting is triggered and fingertip \( F2 \) is relocated. After finger gaiting, grasp stiffness is decreased since the new grasp requires less force to be stable. As the weight increases again, the whole process is repeated, resulting in \( F1 \) finger gaiting.

6.2.2 Shaking Test

External disturbances, such as collisions, may occur once a grasp has been executed. To evaluate the proposed system, we designed a shaking test. We first execute the
best out of the 100 generated grasps for each object according to Eq. (15), and then pose the arm to the configuration shown in Fig. 19. Thereafter, we start to shake the arm in either vertical or horizontal directions while linearly increasing the acceleration from $2m/s^2$ to $8m/s^2$. The shaking magnitude is limited to 10cm in either directions, which means that the hand is accelerating in the first 5cm and decelerating in the second 5cm. After every period of shaking, we increase the acceleration by $1m/s^2$ and therefore have 14 shakes for every test.

Similarly to the supported weight test, we evaluate each grasp by measuring the maximum acceleration it can withstand. The criterion is similar: the maximum acceleration is recorded when the grasp is predicted as unstable for more than 2 seconds or if the object drops. The shaking test is conducted in both directions separately and on each object by filling it with 10g, 20g, 30g, 40g and 50g black pepper beans. Each test is repeated 5 times.

Experimental results are summarized in Fig. 20. If the maximum acceleration rate is $8m/s^2$, it means that the grasp has been kept stable during the test. On the other hand, if the maximum acceleration rate is $0m/s^2$, it means that the grasp could not withstand any shaking. We can see that our system outperforms both the system without adaptation and the system proposed in [32]. The advantage of our approach is that we ensure that the finger gaiting has resulted in an actual contact with the object which is not the case in [32]. In addition, the flexibility of gaiting two fingers provides additional strength.

Additional quantitative results are shown in Fig. 21. We can see that the average computing time of Alg. 8 is between 20ms and 40ms. The average number of explored nodes shows that the pruning is efficient since less than 5% of all nodes in $G_\Phi$ are considered. Note that the computation time and number of explored nodes are heavily dependent on the connectivity of graph $G_\Phi$: less nodes in the graph does not mean less computing time. Therefore, the connectivity in $G_\Phi$ indirectly
7. Conclusion

We have presented a unified framework for grasp planning and in-hand grasp adaptation using visual, tactile and proprioceptive feedback. The proposed Hierarchical Fingertip Space defines a hierarchy of surrogate solution spaces of fingertip grasping enabling both planning and adaptation. By augmenting the fingertip space in terms of local geometry and spatial relations, as well as optimizing hand configurations with respect to grasp adaptability, we demonstrated efficient planning and measures how complex an object is in the context of this system.

Figure 18: A record of supported weight test of a grasp on rivella bottle. Upper: The norm of grasp stiffness and fingertip gaiting. Lower: Likelihood for grasp stability estimation using $\Theta$ defined in Eq. (18).
adaptation. Moreover, the probabilistic model for grasp stability estimation and adaptation has shown its feasibility in closing the loop between grasp replanning and control. We have evaluated the performance of the proposed system quantitatively and shown that the proposed system significantly improves the robustness of grasp execution. It also outperforms our previous work reported in [32]. To the best of our knowledge, this is so far the first system that accomplishes grasp synthesis, stability estimation, online replanning and in-hand adaptation in a unified framework, as well as evaluating this on a real physical system.

However, as a basic drawback of most learning based approaches, our probabilistic model is experience based, and hence relying on the training data, i.e. limited number of objects and examples to generalize from. As a potential future work, we plan to design an active learning strategy to update this model iteratively using new experiences over time, so as to evolve the model in a long term to generalize it to a broader set of objects, without retraining the models from scratch.

Acknowledgement

This work was supported by the European Union Seventh Framework Program FLEXBOT–FP7–ERC–279933 and ROBOHOW–FP7–288533.
Figure 20: Results of shaking tests. In the legend, $H$ and $V$ refer to horizontal shaking test and vertical shaking test respectively. $A$, $B$, and $C$ refer to 3 grasp strategies: grasp without adaptation, grasp adaptation in [32] and the grasp adaptation proposed in this paper. The larger the maximum acceleration rate shown in the graph is, the more external disturbances a grasp could withstand during the tests.

References


Figure 21: Results for the horizontal shaking tests when the objects are filled with 20g of pepper beans (from left to right): average duration for one time of fingertip gaiting; Average stability likelihood improvement after fingertip gaiting; Average computation time of Alg. 8 for each computation; Average errors between achieved rest lengths and the rest lengths computed by Alg. 8. Average number of nodes explored in Alg. 8. Number of fingertip gaiting required during a shaking test with 14 shakes. The evaluations were implemented in C++ and run on a machine with Ubuntu 12.04 running on Intel Core i7-2820QM 2.30 GHz processors.
REFERENCES


REFERENCES


