Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Integrating Perception and Optimization for Dexterous Grasping and Manipulation

Dissertation

Dissertation submitted to the University of Hamburg
with the aim of achieving a doctoral degree at the
Faculty of Mathematics, Informatics and Natural Sciences,
Department of Informatics.

Zhen Deng

Hamburg, 2019

Day of oral defense: 02.December.2019

The following evaluators recommend the admission of the dissertation:
Prof. Dr. Jianwei Zhang (advisor)
Department of Informatics,
University of Hamburg, Germany

Prof. Dr. Stefan Wermter (reviewer)
Department of Informatics,
University of Hamburg, Germany

Prof. Dr. Simone Frintrop (chair)
Department of Informatics,
University of Hamburg, Germany

Abstract

Dexterous grasping of unknown objects is a fundamental functionality for robots, which plays a critical role in industrial and home-assistant applications. However, stable dexterous grasping in a daily-life environment remains a challenging problem. Despite the inherent complexity of object grasping, humans are capable of accomplishing a variety of grasping tasks. Human hands inspire a variety of functions such as perception and demonstrate dexterous grasping and manipulation in daily activities. Our aim is to endow robots with human-like grasping and manipulation capabilities.

For this purpose, this thesis focuses on two important research topics in the area of robotics: robotic perception and grasping. A good perception of the environment helps robots to not only understand task requirements and object properties, but also reduce uncertainties during grasp planning. Among all aspects involved in the implementation of dexterous grasping, grasp planning is the crucial component. Four challenges relevant to the two research topics are addressed in this thesis. They are *stable grasp during implementing manipulation tasks*, *generalization of grasp planning*, *stability of grasps under uncertainties* and *complexity of in-hand manipulation learning*.

First, we address the stable grasping problem during the implementation of reach-to-grasp movements. Finding a feasible grasp configuration and generating a constraint-satisfied trajectory to reach it are challenging. We propose an optimization framework that combines grasp planning with trajectory generation. A Bayesian-based search algorithm is proposed to find the grasp configuration with the highest grasp quality computed by a trained network. For reaching movements, a model-based trajectory generation method inspired by human internal model theory is designed to generate a constraint-satisfied trajectory. The presented framework is validated both through a comparative analysis and in real-world experiments.

Thereafter, we explore the general-purpose problem of grasp planning. It is a natural behavior for humans to choose a feasible grasp type from multiple grasp types to grasp objects. We present an attention-based visual analysis framework that computes grasp-relevant information for fast grasp planning by a multi-fingered robotic hand. A computational visual attention model is employed to locate regions of interest in a scene. A deep convolutional neural network is trained to detect the grasp type and the grasp attention point for a sub-region of the object presented in a region of interest. We demonstrate the proposed framework with object-grasping tasks. Furthermore, a general-purpose grasp planning framework is proposed to allow various robotic hands with an arbitrary number of fingers to grasp objects considering the use of grasp types. A novel definition of pregrasping opposition is introduced, which is taken as a waypoint for grasp formation. A dual-stage grasp planning module beginning with pregrasping opposition prior to planning a feasible grasp configuration for object grasping is proposed. Complex grasp configurations are generated from the simper pregrasping opposition. Moreover, the proposed framework takes grasp types into consideration in grasp planning. The proposed framework is evaluated both in simulated experiments and in real-world experiments.

Tactile sensing is an essential element of dexterous grasping, which allows robotic hands to adapt to changes in their environments. In the next step, we present an object

stabilization framework for a multi-fingered robotic hand to stabilize unknown objects without prior knowledge of their shape or physical properties. The proposed framework consists of three components: an online detection model, a force estimation method, and an object stabilization controller. The online detection model based on a deep neural network is built to detect online contact events and object materials simultaneously from tactile data. A force estimation method based on a Gaussian mixture model is proposed to compute the contact norm force and location based on tactile data. By exploiting the results of tactile sensing, an object stabilization controller is employed to drive a robotic hand to adjust online the contact configuration for object stabilization. The effectiveness of the proposed method was evaluated with a Shadow Dexterous Hand in a real-world experiment.

Last, we also exploit grasp types as guidance for learning in-hand manipulations. Owning to the high dimensionality of robotic hands and intermittent contact dynamics, effectively programming a robotic hand for in-hand manipulations remains a challenging problem. This thesis proposes a Multi-agent Reinforcement Learning (MARL) algorithm to learn in-hand manipulation with a multi-fingered robotic hand. Two additional rewards are designed based on both the analysis of hand synergies and its learning history. The two additional rewards cooperate with an extrinsic reward to assist in manipulation learning. Three agents represented by value functions are trained jointly concerning their reward functions. They then cooperate to optimize a control policy. The experimental results demonstrate that the proposed MARL algorithm allows a five-fingered robotic hand to learn in-hand manipulation effectively.

Zusammenfassung

Geschicktes Greifen unbekannter Objekte ist eine grundlegende Funktionalität für Roboter, die in Industrie- und Heimassistenzanwendungen eine entscheidende Rolle spielt. Stabiles, geschicktes Greifen in einer alltäglichen Umgebung bleibt jedoch eine Herausforderung. Trotz der inhärenten Komplexität, ist der Mensch in der Lage, eine Vielzahl dieser Probleme zu lösen. Menschliche Hände inspirieren Funktionen wie Wahrnehmung und demonstrieren geschicktes Greifen und Manipulation in täglichen Anwendungen. Unser Ziel ist es, Robotern menschenähnliche Greif- und Manipulationsfähigkeiten zu ermöglichen.

Zu diesem Zweck konzentriert sich diese Arbeit auf zwei wichtige Forschungsthemen auf dem Gebiet der Robotik: Greifen und Wahrnehmen mit Robotern. Eine gute Wahrnehmung der Umgebung hilft Robotern nicht nur die Aufgabenanforderungen und Objekteigenschaften zu verstehen, sondern auch Ungewissheiten bei der Ausführung von Aufgaben zu verringern. Bei allen Aspekten des autonomen Greifens ist Planung die entscheidende Komponente. In dieser Arbeit werden vier Herausforderungen behandelt, die für die beiden Forschungsthemen relevant sind: *Griffstabilität während Manipulationsaufgaben*, *Generalisierung der Griffplanung*, *Griffstabilität unter Ungewissheiten* und *die Komplexität des Lernens von In-Hand-Manipulation*.

Zuerst befassen wir uns mit dem Problem der Stabilität während dem Greifen von unbekannten Objekten. Eine realisierbare Griffkonfiguration und eine kollisionsfreie Bewegung zu dem Objekt zu finden ist eine groe Herausforderung. Wir stellen ein Optimierungsframework vor, das Griffplanung mit Trajektoriengenerierung kombiniert. Ein Bayes-basierter Suchalgorithmus wird eingeführt, um die Greifkonfiguration mit der höchsten Qualitöt zu finden, die von einem trainierten Netzwerk berechnet wird. Um Bewegungen zu erreichen, wird eine modellbasierte Methode zur Trajektoriengenerierung entwickelt, die von der Theorie des internen menschlichen Modells inspiriert ist. Das vorgestellte Framework wird sowohl durch eine vergleichende Analyse als auch in realen Experimenten validiert.

Anschlieend Es ist für Menschen ein natürliches Verhalten, aus mehreren möglichen einen geeigneten Grifftyp auszuwählen, um Objekte zu greifen. Wir präsentieren ein auf dem Prinzip der Aufmerksamkeit basierendes visuelles Analyseframework, zur Berechnung relevanter Informationen für eine schnelle Griffplanung mit einer Roboterhand mit mehreren Fingern. Ein berechnetes visuelles Aufmerksamkeitsmodell wird verwendet, um interessante Regionen in einer Szene zu lokalisieren. Ein tiefes neuronales Faltungsnetzwerk (CNN) wird trainiert, um den Grifftyp und den Griffschwerpunkt für den Teil des Objekts zu finden, der in dem interessanten Bereich liegt. Das vorgestellte Framework wird anhand von Greifbewegungen demonstriert. Auerdem wird das generelle Griffplanungsproblem untersucht. Ein Framework wird vorgestellt, das verschiedenen Roboterhänden, mit einer beliebigen Anzahl von Fingern, ermöglicht, Objekte zu greifen. Eine neue Definition des Oppositionsprgriff wird eingeführt, der als Wegpunkt bei der Bildung der gesamten Bewegung dient. Ein zweistufiges Greifplanungsmodul, beginnend mit dem Oppositionsprägriff und anschlieender Planung einer möglichen Griffkonfiguration wird vorgestellt. Komplexe Griffkonfigurationen werden dabei aus dem simpleren Oppositionprägriff generiert. Darüber hinaus berücksichtigt das vorges-

chlagene Framework die Grifftypen während der Planung. Das Framework wird sowohl in simulierten als auch in realen Experimenten evaluiert.

Tastsensorik, als wesentliches Element des geschickten Greifens, ermöglicht einem Roboter sich an Veränderungen seiner Umgebung anzupassen. Im nächsten Schritt präsentieren wir ein Framework zur Objektstabilisierung für eine Roboterhand mit mehreren Fingern, um unbekannte Objekte ohne vorherige Informationen über Form und physikalischen Eigenschaften zu stabilisieren. Das vorgestellte Framework besteht aus drei Komponenten: einem Echtzeitmodell zur Erkennung, einem Verfahren zur Kraftabschätzung und einem Controller zur Objektstabilisierung. Das Echtzeitmodell zur Erkennung basiert auf einem tiefen neuronalen Netzwerk und erkennt Kontaktereignisse und Materialien gleichzeitig aus taktilen Daten. Das Verfahren zur Kraftabschätzung basiert auf einem Gauschen Mischverteilungsmodell und berechnet die Normalkraft und den Ort des Kontakts basierend auf taktilen Daten. Die Ergebnisse der Tastsensorik werden von dem Controller zur Objektstabilisierung verwendet, um die Kontaktkonfiguration einer Roboterhand für die Objektstabilisierung in Echtzeit anzupassen. Die Effektivität der vorgestellten Methode wurde mit einer Shadow Dexterous Hand in einem realen Experiment evaluiert.

Zuletzt benutzen wir Grifftypen als Leitlinie für das Erlernen von In-Hand Manipulationen. Aufgrund der hohen Dimensionalität von Roboterhänden und der intermittierenden Kontaktdynamik bleibt die effektive Programmierung einer Roboterhand für In-Hand Manipulationen eine Herausforderung. In dieser Arbeit wird ein Multi-Agenten Reinforcement Learning Algorithmus (MARL) zum Erlernen der In-Hand-Manipulation mit einer Roboterhand mit mehreren Fingern vorgestellt. Die beiden zusätzlichen Belohnungen basieren sowohl auf der Analyse von Handsynergien als auch auf der Lernhistorie. Drei Agenten die durch Wertefunktionen repräsentiert werden, werden hinsichtlich ihrer Belohnungsfunktionen gemeinsam trainiert. Diese arbeiten zusammen, um eine Kontrollstrategie zu optimieren. Die experimentellen Ergebnisse zeigen, dass der vorgeschlagene MARL-Algorithmus einer Roboterhand, mit fünf Fingern, ermöglicht die In-Hand-Manipulation effektiv zu erlernen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Autonomous grasping is one of the fundamental abilities of robots when they implement manipulation tasks, and it plays a critical role in industrial and home-assistant applications. Advances in the grasping ability of robots to hand objects could have an enormous impact on many critical application areas. For example, autonomous grasping by robotic hands could help a manufacturer to increase productivity and reduce costs significantly. Service robots for personal and private use also require stable grasping to accomplish manipulation tasks. With the ability of stable grasping, a robot could effectively help older people live independently by assisting them. There is also an increasing impact in the medical world. The development of hand rehabilitation robotics helps people suffering from stroke to recover their hand functions. However, despite the development of control and sensing technology, the robotic hands almost exclusively implement highly repetitive tasks in conditioned operating environments. It would be highly beneficial if robots had the ability to perceive and grasp a variety of unknown objects in an unstructured environment

Over the past two decades, robotic grasping has attracted much attention from a wide range of researchers. Although considerable effort has been made, stable robotic grasping in daily-life environments is still a challenging problem. Finding a feasible grasp configuration for a robotic hand involves multiple tasks (such as robotic perception and grasp planning). Despite the inherent complexity of object grasping, humans can effortlessly accomplish a variety of grasping and manipulation tasks. Human hands demonstrate a variety of functions, from perception to delicate grasping and manipulation, in daily activities. In terms of object perception, the skin of human hands has mechanoreceptors that provide feedback signals for grasp control. The human hand can maintain an adaptive contact force to grasp objects stably. In terms of the motor control of human hands, studies from neuroscience have explained that human Central Nervous System (CNS) coordinates the neural activation of groups of hand muscles. The human hand quickly generates a remarkable degree of dexterity in its capacity to grasp objects. Moreover, humans are capable of exploiting a prior domain knowledge (e.g., experience and culture) to infer and understand the objects and tasks. Humans can effectively grasp

a variety of objects to achieve complex manipulation tasks (e.g., writing and tool usage). The amazing abilities of the human hand motivate researchers in the robotics community to develop novel robotic techniques to achieve similar grasping and manipulation abilities.

Among all the aspects involved in the implementation of dexterous grasping, grasp planning is a crucial component. The aim of grasp planning is to find a feasible grasp configuration (including the 6D pose of the hand relative to the object and the finger configuration) for robotic hands. The planned grasp configuration should satisfy a user-defined grasp quality metric. Stability is the most crucial consideration during grasp planning. Traditionally, researchers defined a quality metric (such as form- or force-closure) geometrically to consider stable constraints for object grasping. However, these geometric methods rely on an accurate object model that is difficult to obtain. Owing to the uncertainties arising from sensor noise, the planned grasps based on a geometric metric might end up being unstable in a real-world environment. Thanks to the power of deep neural networks, more recent works employed deep learning techniques to approximate a grasp quality metric. These learning-based methods could predict grasp quality directly from visual data (such as RGB-D images or point clouds). Dexterity is another crucial characteristic of a robotic hand, especially for a multi-fingered robotic hand. Previous work has rarely discussed how to exploit the dexterity of robotic hands to perform complex grasping tasks. It is a natural behavior for humans to choose a feasible grasp type from multiple grasp types to grasp an object. If a robotic hand is able to grasp an object with multiple grasp types, it could carry out more dexterous manipulation tasks. The generalization of grasp planning is also important, but has not been addressed well in the past. Most of the previous grasp planning methods are designed for one specific robotic hand and only consider one specific grasp type (such as *power* grasp). Little work was done to address the general-purpose problem of grasp planning. These requirements (i.e., stability, dexterity, and generalization) for a grasp implemented in a real-world environment still highlights the need to develop advanced grasp planning techniques for robotic hands, especially for multi-fingered robotic hands.

Robotic perception is of critical importance for a robot to plan and operate in real-world environments. Robotic perception could allow a robot to learn from sensory data and make decisions to interact with its environment. Previous work on robotic perception mainly focuses on constructing object models from visual data. Apart from model reconstruction, another important application of robotic perception is to learn meaningful information (such as object class, visual relationship, and grasping affordance) from sensory data. The ability to rapidly extract relevant information from sensor inputs is an important mechanism and a natural behavior for humans. For example, humans can rapidly determine where and how to grasp an object through visual sensing. This ability is also essential for robots. Visual analysis methods should be proposed for the robot to compute relevant information that could be taken as guidance for grasp planning. Tactile sensors provide another important sensory modality for robots to perceive their environments. As stated above, the mechanoreceptors in the skin of human hands provide feedback signals for grasp control. With the development of sensing technology, robots have the potential to use various types of tactile sensors to sense objects and their surroundings. For example, multimodal tactile sensors (such as BioTac sensors) could

be used to detect the contact force, vibration, and temperature when a robot is grasping an object. Given multiple sensory modalities as feedback, a core problem is how to process and represent sensory data for robotic grasping and manipulation.

The objective of this thesis is to endow a robot with human-like grasping ability. This thesis mainly focuses on the two important research topics of robotic perception and grasping in the area of robotics. Effectively perceiving an environment is of critical importance for autonomous robots that are expected to interact with a variety of objects. A good perception of the environment not only reduces the uncertainties of grasp execution but also provides an understanding of the task goal and object properties. We expect that robotic hands with human-like dexterity to perform stable and robust grasping tasks. To this end, this thesis addresses the following four challenges: *stable grasping during manipulation tasks*, *generalization of grasp planning*, *stability of grasps under uncertainties* and *complexity of in-hand manipulation learning*. This thesis proposes novel algorithms and approaches to address these four challenges inspired by the analysis of human grasping behavior, exploiting the benefits of machine learning techniques and combining perception, planning and learning methods. Addressing these open challenges in the area of robotic perception and grasping has the potential of allowing robots to accomplish stable and robust grasping and manipulations in daily-life environments.

1.2 Problem Statement and Research Objectives

Dexterous grasping and manipulation of objects are two fundamental abilities for robotic hands that have to interact with their environments. Owing to uncertainties arising from the environments and the complex dynamics of robotic hands, effective implementations of dexterous grasping and manipulation are still difficult for robotic hands. Robotic grasping and manipulation are two core research topics in the area of robotics, which have attracted much attention in the past. Despite there has been much progress, robotic hands are still far from being capable of implementing dexterous grasping and manipulation like humans. To allow robotic hands to implement dexterous grasping and manipulation effectively, in this thesis, we explore the following four open challenges.

- **Stable grasping during manipulation tasks:** We address the stable grasping problem of the implementation of reach-to-grasp (RTG) tasks in an unstructured environment. There are three main sub-tasks required to complete RTG tasks: detecting the object of interest from the table, determining a feasible grasp configuration, and generating a collision-free trajectory to reach it. Due to the lack of object models and uncertainties from perceptual noise or kinematic errors of robots, the stable implementation of RTG tasks remains an open problem. The first objective of this thesis is to investigate a unified framework that allows robots to reach to grasp the object of interest stably. In particular, we expect that the proposed framework to combine grasp planning and trajectory generation together.

- **Generalization of grasp planning:** Most of the state-of-the-art grasp planning approaches are designed for one specific robotic hand. In real-world robotic ap-

plications, multiple robots with different kinds of robotic hands may work together. This requires generalizing grasp planning approaches across different robotic hands. Moreover, previous grasp planning approaches only consider two grasp types (i.e., *power* and *precision* grasp) for robotic hands to implement object grasping tasks. It is a natural behavior for humans to choose a feasible grasp type from multiple possible grasp types to grasp objects. This ability is also essential for robots. However, little previous work addresses the general-purpose problem of grasp planning. The second objective of this thesis is to develop a general-purpose grasp planning approach. We expect that the proposed approach suits various robotic hands with an arbitrary number of fingers to grasp objects considering the use of grasp types.

- **Stability of grasps under uncertainties:** Realizing stable object grasping with a multi-fingered robotic hand under uncertainties is a challenge. The uncertainties may arise from sensor noise, slippage or external disturbances. In real-world applications, the uncertainties may cause a planned-to-be stable grasp to be unstable. Humans are capable of reacting to the unstable through tactile sensing. However, the ability of object stabilization through tactile sensing for robotic hands is still missing. To stabilize a grasped object effectively with a robotic hand, three related problems should be addressed: (1) How to detect the properties (such as material) of the grasped object. (2) How to perceive the contact event (i.e., slippage or not) effectively. (3) How to correct the contact configuration of a robotic hand online through tactile sensing to avoid slippage. The third objective of this thesis is to enable robotic hands to make use of tactile sensing for object stabilization. We expect to take advantage of the spatial-temporal characteristics of tactile data to perceive the grasped object and the contact between the robotic hand and the object.

- **Complexity of in-hand manipulation learning:** Owing to the high dimension of multi-fingered robotic hands and intermittent contact dynamics, effectively programming a robotic hand for in-hand manipulations is still a challenge in the area of robotics. Deep reinforcement learning (DRL) algorithms have been previously employed to learn manipulations directly from interactions with the environments. However, learning complex manipulations (such as in-hand manipulation) with DRL is usually slow and unstable. The sample complexity limits the applicability of DRL to learn complex manipulations. The last objective of this thesis is to explore a learning method that exploits domain knowledge to assist in in-hand manipulation learning. In particular, we look at how the information extracted from the analysis of hand synergies can be used to improve the learning performance of the DRL.

1.3 Novelty and Contribution of the Thesis

This thesis presents approaches that integrate visual/tactile perception with optimization methods for both grasp planning and manipulation learning. All these approaches are

designed to address the problem statement of this thesis. The main contributions of this thesis are summarized in the following points:

- **Optimization for reach-to-grasp movements:** Implementing RTG tasks stably is an essential ability for a robot. The implementation of RTG tasks typically contains three main aspects: object detection, stable grasp planning, and trajectory generation. To achieve RTG tasks, we propose an optimization framework that combines stable grasp planning with model-based trajectory generation. Different from previous methods, we address the grasping and trajectory generation problem jointly when implementing RTG tasks. The proposed optimization framework covers the complete path from perception to decision-making when implementing RTG tasks. In terms of stable grasping, most of the state-of-the-art grasp planning approaches detect a feasible grasp directly from visual data. They may not be robust to uncertainties. This thesis trains a grasp quality metric with a deep neural network and employs a Bayesian-based search algorithm for determining high quality grasps. In this way, grasp stability is improved. Moreover, the proposal grasp planning method does not rely on an accurate object model or pose estimation. A model-based trajectory generation approach inspired by human internal model theory is presented to produce collision-free reaching movements. The proposed model-based trajectory generation approach can generalize quickly to new tasks.

- **General-purpose grasp planning with visual analysis:** Developing general-purpose grasp planning approaches is a necessity for intelligent robots. This thesis proposes a general-purpose grasp planning approach that suits various robotic hands with an arbitrary number of fingers to grasp objects considering the use of grasp types. We train a grasp type detection model with a deep neural network, which can detect a feasible grasp type directly from visual data. Moreover, the concept of *pregrasping opposition* is proposed to encode the grasp type information. A dual-stage grasp planning method with the *pregrasping opposition* is proposed to find a feasible grasp configuration involving a set of contact points. The use of the *pregrasping opposition* not only reduces the computation complexity of grasp planning but also provides a way to generalize the grasp planning approach across different robotic hands.

  The proposed approach has three advantages compared to the state-of-the-art: (1) It trains a deep neural network to detect grasp type directly from visual data. Six different grasp types are considered. Moreover, the proposed detection method does not rely on handcrafted features. (2) The grasp-relevant information (i.e., grasp type and grasp attention point) extracted from visual data is used as guidance for grasp planning. (3) The proposed approach suits various robotic hands with an arbitrary number of fingers to grasp objects considering the use of grasp types.

  The proposed general-purpose grasp planning approach suits scenarios where multiple robots with different robotic hands work together. Instead of designing different grasp planning methods for different robotic hands, the proposed

approach is able to plan feasible grasp configurations for various robotic hands with an arbitrary number of fingers.

- **Dynamic object stabilization through tactile sensing:** We present an object stabilization framework for multi-fingered robotic hands to stabilize unknown objects without a prior domain knowledge of their shape or physical properties. The proposed framework consists of three components: an online detection model, a force estimation method, and an object stabilization controller. There are two main differences between the proposed tactile sensing method and the state-of-the-art: (1) The proposed method addresses the slip detection, material detection, and force estimation problems jointly. The spatial-temporal characteristics of tactile data are exploited for perception. (2) The proposed method uses deep learning techniques to learn the latent representation of tactile data for multi-task classifications. In this way, we do not need to compute features of tactile data manually for classification. By exploiting the results of tactile sensing, an object stabilization controller is employed to control a robotic hand to adjust its contact configuration online.

- **Learning synergies-based in-hand manipulation:** We propose a multi-agent reinforcement learning (MARL) algorithm for a multi-fingered robotic hand to learn in-hand manipulation. Manipulation learning with a DRL algorithm is usually slow and unstable, especially when the reward is sparse. For effective manipulation learning, we propose to extract information from both the analysis of hand synergies and the learning history. Two additional rewards are designed to encode the extracted information into the DRL algorithm. The use of additional rewards not only improves the exploration efficiency of the DRL but also provides a way to incorporate domain knowledge. The experimental results demonstrate that exploiting domain knowledge (i.e., grasp type) is effective for learning in-hand manipulation with a multi-finger robotic hand.

1.4 Structure of the Thesis

This section briefly summarizes the remaining contents of this thesis. Figure 1.1 illustrates the structure of this thesis.

- **Chapter 2-Background and related work:** First, we give an overview of the analysis of human grasping behavior. The representation of human grasps and the mechanism of human grasp choices are considered, which inspire the development of robotic grasping techniques. Then, we review related work in the area of robotic grasping and focus on the two main research directions: perception for robotic grasping and robotic grasping. Finally, we introduce related work on in-hand manipulation with multi-fingered robotic hands.

- **Chapter 3-Optimization for reach-to-grasp movements:** In this chapter, we address the problem of the implementation of RTG tasks. The two key tech-

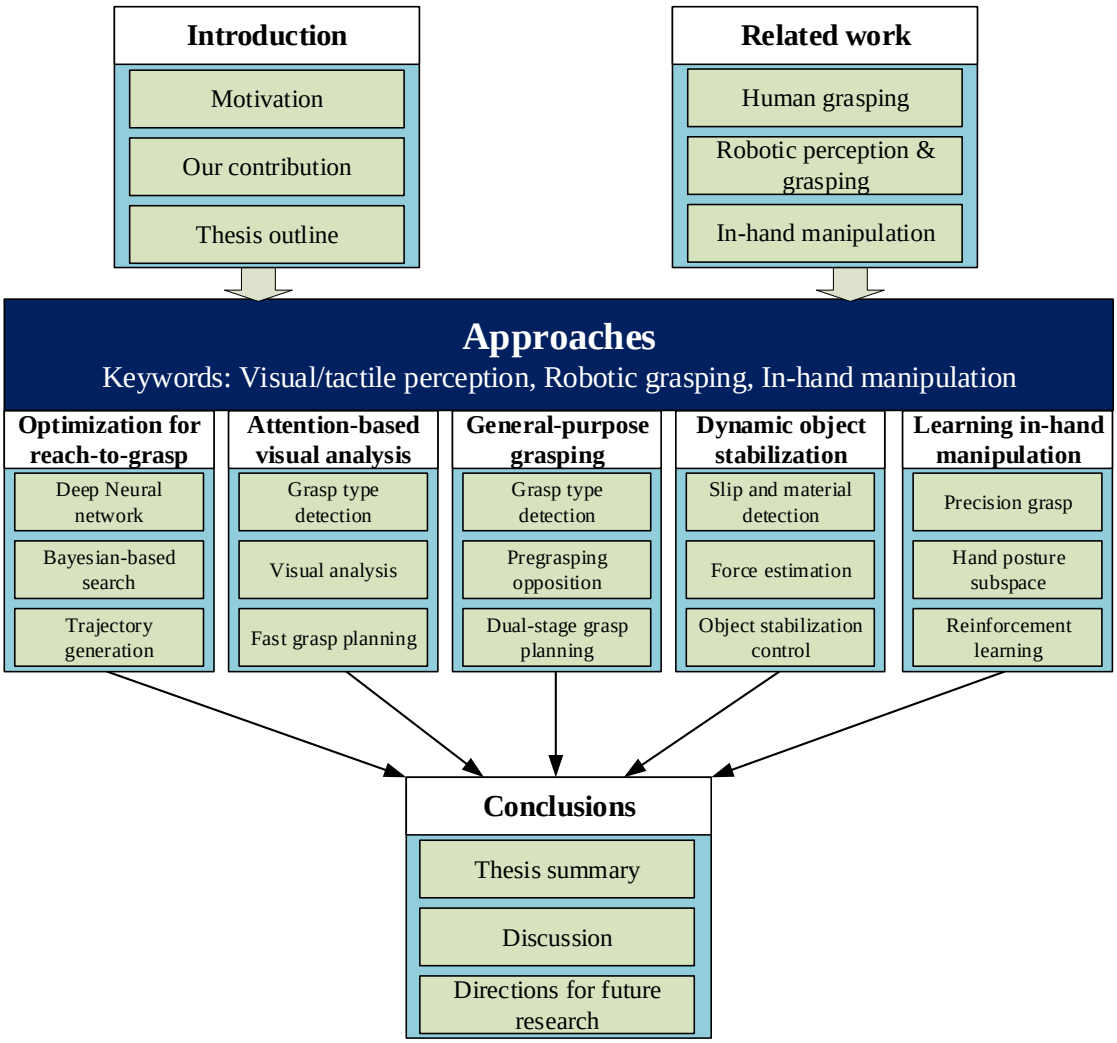


Figure 1.1: The structure of this thesis.

niques, i.e., grasp planning and trajectory generation, are studied. An optimization framework that combines stable grasp planning with trajectory generation is proposed for the implementation of RTG tasks. We show that the grasp quality metric could be learned by using a deep neural network. We also show that the proposed Bayesian search algorithm has the potential to find more stable grasp configurations. A model-based trajectory generation method inspired by human internal model theory is designed to generate reaching movements for the robots. The proposed optimization framework is evaluated with object grasping tasks in real-world experiments.

- **Chapter 4-Grasp type detection for fast grasp planning:** The focus here is visual analysis for fast grasp planning. We introduce an attention-based visual analysis framework to compute grasp-relevant information that is used as guidance for fast grasp planning. A deep convolutional neural network (CNN) is trained for grasp type detection. A search-based grasp planning approach is proposed to

determine feasible grasp configurations. We demonstrate that grasp-relevant information (i.e., grasp type and grasp attention point) can be extracted from visual data. We also show that the extracted information helps to reduce the configuration space of grasp planning and to accelerate the formation of the final grasp. We evaluate the proposed framework with object grasping tasks.

- **Chapter 5-General-purpose grasps planning with pregrasping opposition:** Here we aim to design a general-purpose grasp planning approach suited various robotic hands with an arbitrary number of fingers. Moreover, we expect that the robotic hand can automatically choose a feasible grasp type to operate the target object. We introduce the new concept of *pregrasping opposition* that is taken as a waypoint for grasp formation. A dual-stage grasp planning approach exploiting the *pregrasping opposition* is proposed. We show that the use of the *pregrasping opposition* not only reduces the computational complexity but also provides a way to generalize the grasp planning method to various robotic hands. It can enable a robotic hand to realize different grasp types. A PR2 robotic platform equipped with a five-fingered robotic hand and a two-fingered gripper is used for evaluation.

- **Chapter 6-Dynamic object stabilization through tactile sensing:** In this chapter, we study how to exploit tactile information for object stabilization with multi-fingered robotic hands. We proposed an object stabilization framework that combines tactile sensing with feedback control. We train an online detection model based on a deep neural network to detect the contact event and object material simultaneously from tactile data. We show that the contact information (i.e., the contact force and contact location) can be extracted by using the spatial properties of the tactile data. An object stabilization controller exploiting the detected results is applied to adjust the contact force online to ensure the stability of grasps. The effectiveness of the proposed framework is evaluated in a real-world experiment with a Shadow Dexterous Hand.

- **Chapter 7-Learning in-hand manipulation with reward shaping:** We present a MARL algorithm for a multi-fingered robotic hand to learn in-hand manipulations. The basic idea of the proposed algorithm is that information extracted from both the analysis of hand posture synergies and the learning history could guide in-hand manipulation learning. Two additional reward functions, i.e., a hand-based reward and an uncertainty-based reward, are designed to encode the extracted information. Three independent agents represented by value functions are trained jointly concerning their rewards and then cooperate to optimize a control policy. The performance of the proposed MARL algorithm is evaluated in a physical simulator. The experimental results demonstrate that the proposed MARL algorithm allows multi-fingered robotic hands to learn in-hand manipulation effectively.

- **Chapter 8-Conclusion and future work:** In this final chapter, we conclude the thesis by providing a summary of this work. Furthermore, we discuss the proposed approaches. Finally, we point out what we believe to be the most promising directions for future research.

Chapter 2

Related Work

Grasping is one of the most critical abilities of humans and robots when implementing a variety of manipulation tasks. Humans are capable of performing grasping stably to achieve a variety of manipulation tasks. Currently, object grasping with robotic hands is still limited to certain simple applications where certain simple robotic hands implement some specific tasks in a structured environment. Stable grasping, especially stable dexterous grasping, in a real-world environment is still a challenge for the robots. Many efforts from the neuroscience, physiology, and robotics communities have been conducted to address this challenge from different aspects. In the neuroscience and physiology communities, human grasping behavior has been widely studied to understand the mechanics of grasp formation in recent decades. Many results on the mechanism of human grasp choice and the representation of human grasping posture have been introduced. In the robotics community, the research of robotic grasping mainly focuses on the problem of planning and learning feasible grasp configurations of robotic hands. To implement dexterous grasping tasks successfully, robots are required to perceive their environments, understand task goals, and plan feasible grasp configurations. Many methods of robotic perception and grasping planning have been previously introduced to allow robots to perform stable grasping.

In the rest of this section, we first review the studies on human grasping behavior Section 2.1. The representation of human grasping posture and the mechanism of human grasp choice are introduced. Section 2.2 introduces previous work on robotic perception for grasping. Section 2.3 presents four important research topics in the area of robotic grasping. Finally, we review existing works on in-hand manipulation in Section 2.4.

2.1 Human Grasping

Humans are capable of grasping a variety of unknown objects to achieve manipulation tasks. In recent decades, many efforts have been conducted in the neuroscience and physiology communities to analyze human grasping behavior from different aspects. Previous studies on human grasping behavior help to understand the process of grasp formation. In this section, we mainly review related work on the representation of human grasping posture and the mechanism of human grasp choice.

2.1.1 Representation of Human Grasping Postures

A basic question relevant to the analysis of human grasping behavior is how to represent human grasping postures and reveal their underlying function during grasp formation. Human grasping postures can be basically described from two viewpoints: the geometric viewpoint and the abstraction viewpoint. In terms of the geometric viewpoint, human grasping postures are represented by the joint angles of a hand recorded by a motion tracking system [53]. Owing to the complicated anatomical structure of human hands, the representation with joint angles has high complexity. Moreover, it is easily affected by sensor noises and the tracking performance of a motion tracking system.

Studies from neuroscience suggested that the CNS of humans adopted a simple strategy to coordinate a large number of degrees of freedom in motor control [37]. The concept of the synergies has been introduced as a strategy to describe the movement control involving multiple degrees of freedom. This concept has also been widely used to analyze human grasping behavior. Santello et al. [149] used principal components analysis (PCA) to analyze human grasping postures. They demonstrated that the first two principal components contained more than 80% of the hand posture information. Chen et al. [24] also used the PCA to decompose human hand movements into the primary and secondary motions. The decomposition guided the design of an artificial hand. The primary motion was taken as a free motion controlled by actuators, and the secondary motion was implemented with mechanical compliance. Additionally, Ciocarlie et al. [29] exploited the synergies for grasp planning. The concept of the *Eigengrasp* was introduced to represent a low-dimensional posture subspace for dexterous grasping. Based on the analysis of hand synergies, a few principal components could be used to represent hand grasping postures. The use of the synergies helped to reduce the representation complexity of grasping postures.

The representation methods with joint angles were unable to reveal the underlying functions of human grasps. To illustrate the conceptual functions of human grasping, several concepts are introduced from an abstraction viewpoint. One of the most important concepts was the grasp type. Different taxonomies of grasp types have been proposed, in which human grasping postures were classified into different grasp types. Traditionally, grasps were divided into to two categories: *power* or *precision* grasps [125]. The *power* grasp used the fingers and the palm of a hand to hold an object firmly. The *precision* grasped only used fingertips to stabilize an object. However, this two-category grasp taxonomy was not sufficient to convey information about hand grasping configurations. The most recent work [28] extended the taxonomies [46, 54] and developed a new taxonomy including 33 different grasp types, as shown in Figure 2.1. Grasp type was a way of representing how a hand handles objects. For example, once a feasible grasp type is detected for the target object, robotic systems can make sense of how to used its fingers to contact and grasp the object. These grasp taxonomies help us to develop advanced grasp planning techniques.

In addition, grasping is a process in which a hand applies forces on contact points to stabilize an object. Apart from only considering the appearance of human grasping postures, it was important to analyze the physical characteristic of human grasping. Arbib et al. [4] introduced the concept of virtual fingers (VFs) to represent grasp formation.

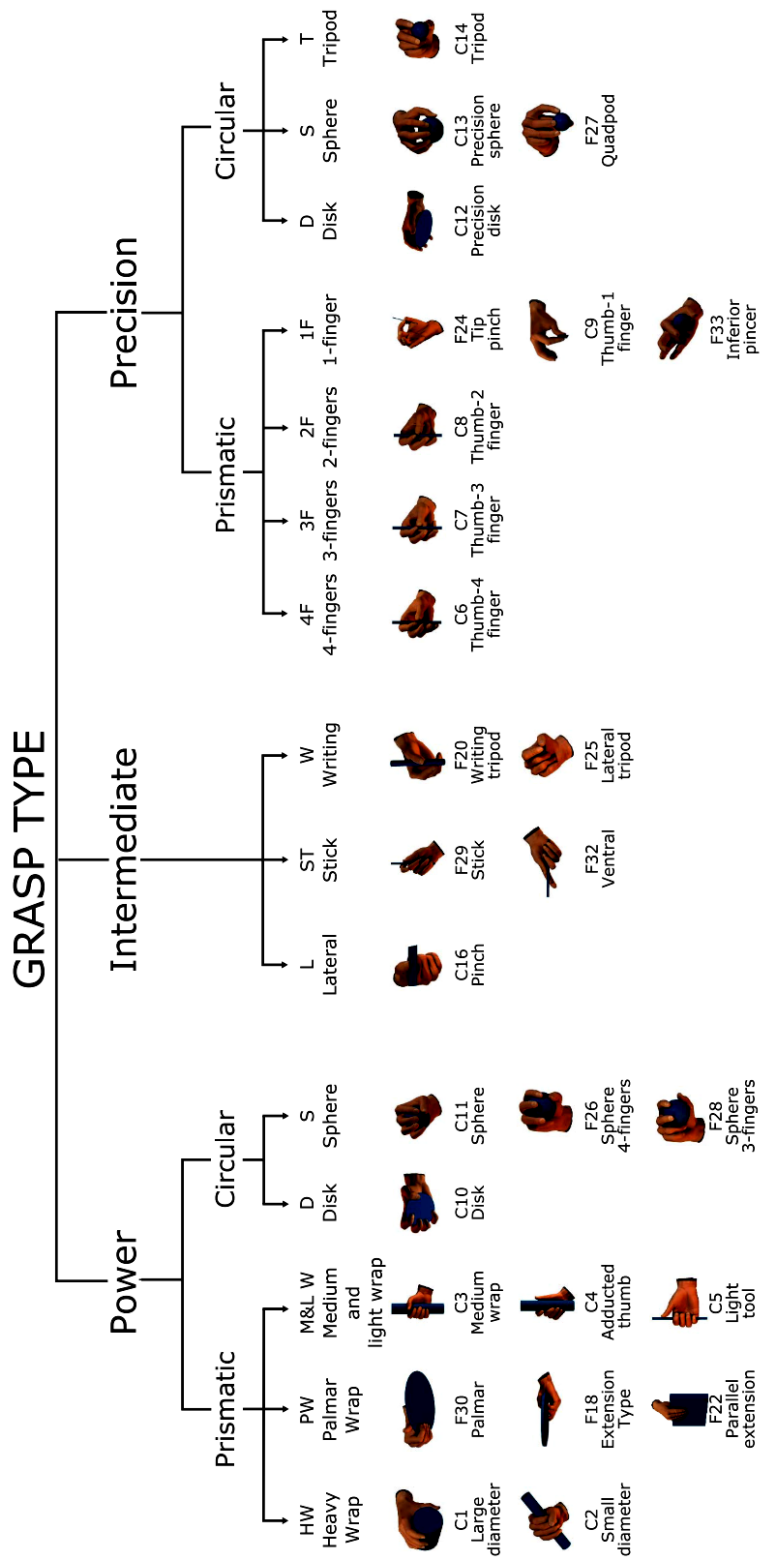


Figure 2.1: Grasp taxonomy introduced by [28] that includes 33 different grasp types.

A VF was a collection of individual hand fingers that worked together to apply an oppositional force. Based on the VFs, a stable grasp was taken as a combination of two VFs. The two VFs were brought together against the object surface to form a grasp configuration. Wimbock et al. [177] exploited the concept of VFs to design an object-level grasp controller for fine manipulation tasks. Furthermore, when grasping an object, human hands are required to apply a pair of opposing forces on the object surface. Based on this hypothesis, Iberall et al. [75] introduced the concept of *opposition* to describe human grasping postures. In that work, three types of *oppositions* primitive (i.e., *Pad opposition*, *Palm opposition*, and *Slid opposition*) were introduced. Additionally, Iberall et al. correlated the *opposition* and the VFs to describe human grasping postures. Each opposing part was taken as a VF, and two VFs formed an opposition. Souza et al. [35] extended the definition of *opposition* in [76] by introducing more opposition primitives. They defined the choice of hand-parts during grasp formation as the *grasp intention*. The *grasp intention* was modeled as a mix of *opposition* between hand parts. Figure 2.2 illustrates human grasping postures grouped by the *opposition* and the virtual finger. The *opposition* and the VF serve as two abstract representations that encode the underlying function of hand grasping.

Overall, human grasping postures can be described from different viewpoints. Several concepts (such as grasp type, *opposition*, VF, and grasp intention) were introduced to represent human grasping postures. The concepts can be used to describe the appearance of human grasping postures or to analyze the physical characteristic of human grasping. These studies on the representation of human grasping postures can benefit research in the robotics community, such as on the design of artificial hands and the development of grasp planning techniques.

2.1.2 Human Grasp Choice

Another fundamental question relevant to hand grasping behavior is how humans choose a grasp to operate an object for implementing manipulation tasks. The study of human grasp choice has also attracted considerable attention from researchers across different communities. Naturally, when humans grasp different objects, the final grasping postures are different. For example, the grasping postures are typically different when a hand grasps a pencil and an apple. Humans can effortlessly choose feasible grasp configurations to operate a variety of objects. There exist several factors that influence grasp formation. It is interesting to study the types of factors that influence the choice of human grasps and to build a relationship between the grasping posture and these factors.

Fei et al. conducted a series of experiments with a wide range of unstructured tasks to analyze the mechanism of human grasp choice [45, 44]. First, Feix et al. [45] investigated the relationship between grasps and object properties. The concept of *grasp location* was defined as the local part of an object surface. Humans first determined the *grasp location* on an object and then inferred a feasible grasp posture based on the object geometry. They also suggested that humans had a high probability of using the same grasp for specific types of objects. Furthermore, Feix et al. [44] correlated the relationship among the task, object properties, and grasps. They suggested that the task requirement and object properties were the two essential factors that influenced human

Figure 2.2: Parts of opposition primitive with which *grasp intention* is interpreted [35].

grasp choices. Different from Fei et al.'s works that were based on human grasping datasets collected in a laboratory setting, some recent works employed computer vision techniques to analyze the mechanism of human grasp choice. Cai et al. [16] recognized hand grasp types, object attributes, and manipulation tasks from RGB images. The relationship among grasp types, object attributions, and manipulation actions was built with a unified computational model.

Previous work mainly demonstrated that task requirements and object properties were the two important factors that influence human grasp choice [31, 44]. However, Cini et al. [28] pointed out that human experience and social environment also influence human grasp choice. They suggested that the following five major factors influence human grasp choice:

- Object properties (e.g., object shape, and object function)
- Task constraints (e.g., object state and contact force)
- Gripper constraints (e.g., hand kinematic and hand size)
- Habits of humans (e.g., experience, preference and social convention)
- Chance (e.g., environmental constraints and the initial position of the object)

In general, the five factors influenced human grasp choice, but they did not have the same contribution to the formation of grasp configurations. The final grasp selected by humans was characterized by a subset of the five factors. The contribution of each factor may change in different situations and vary over time.

2.2 Perception for Robotic Grasping

Perceiving the target object is the first step in robotic grasping. Grasp planning requires information about the shapes and physical properties of objects. Different sensory modalities have been previously applied for object perception, such as visual and tactile sensors.

Visual perception plays an important role in robotic grasping. In recent decades, many visual analysis methods have been proposed to analyze visual data of objects and compute meaningful information for grasp planning. Traditionally, visual data (such as point clouds) were registered by associating it with the point clouds of an object mesh mode. Registration techniques [162], such as Iterative Closest Point (ICP) [117], have been developed to realize the point-to-point association. After object registration, grasp configurations of robotic hands were planned on object mesh modes. These registration-based methods typically required 3D object mesh models that were often not available. Some works proposed to reconstruct object models directly from point clouds of objects. Quispe et al. [137] generated object mesh models from partial point-clouds based on the analysis of symmetry and extrusion patterns in observed shapes. Vezzni et al. [173] used superquadric functions to fit object point clouds to obtain object models. More recent works have used deep learning techniques for object shape completion. Varley et

al. [168] employed a 3D CNN that takes a 2.5 D point clouds as input and performed object shape completion.

In addition, there have been some recent works that detected object grasping affordances from visual data. Object grasping affordance provides certain critical information about the functions of objects and their parts. Myers et al. [124] localized object parts and identified their functions from the geometric features of objects. In that work, seven affordances were considered: *grasp*, *cut*, *scoop*, *contain*, *pound*, *support*, and *wrap-grasp*. Given the results of affordance detection, robots can make sense of how and where to operate objects. Different from Myers et al.'s work, Nguyen et al. [126] hypothesized that an object could have multiple affordances. A CNN was employed to predict the affordances of object parts which was represented by a rectangular box. Detry et al. [41] identified grasp-suitable regions in a depth image by using a CNN-based semantic model. Object grasping affordances detected from visual data help us not only to obtain an understanding of the objects but also to develop novel grasp planning techniques.

Tactile sensors provide additional critical information about interactions between hands and objects. Studies from neuroscience suggested that humans continually monitor contact events and correct any mistakes given the signals in tactile afferent neurons and central processes [81]. Cutkosky et al. [32] introduced that human skins were endowed with fast- and slow-acting (FA and SA) mechanoreceptors. The FA mechanoreceptors responded to high-frequency signals of an event. The SA mechanoreceptors responded to steady skin deformation. Owing to the uncertainties arising from the environmental changes or unknown of object properties, humans could adapt the grasping postures to maintain grasping stability. In the robotics community, many efforts have been made to process and represent tactile information to achieve robotic grasping and manipulations. Tactile information has been used to detect contact events between the hand and object [170, 169], to measure grasp qualities [8], and to recognize object's physical properties [27, 60, 25]. A good survey on the tactile perception of object properties can be found in Luo et al. [111]. The perceived results from tactile sensing can help robots to reduce uncertainties and improve task performance.

Vision perception can provide global information about the geometries of objects and their space states. However, the effectiveness of visual perception is easily affected by lighting conditions and occlusion. Moreover, visual perception is unable to perceive object physical properties. Tactile sensing can be used to detect object properties and contact events between the hand and the objects. Hence, It is necessary for robots to fuse visual and tactile information to perceive objects and environments. To date, visual-tactile fusion for robotic perception has attracted much attention. Sparse coding has been widely used to fuse visual and tactile information [153]. Liu et al. [110] designed a joint group kernel sparse coding (JGKSC) method for object recognition. In that work, visual and tactile modalities were combined to address the recognition of objects. More recent works have used deep learning techniques for visual-tactile fusion. Gao et al. [50] trained a deep neural network to learn and fuse visual and tactile data for object material detection. Li et al. [103] combined tactile and visual information based on a deep neural network for slip detection.

2.3 Robotic Grasping

Robotic grasping is a fundamental ability for robotic hands that interact with their environments. Reviewing previous research on robotic grasping helps us to understand the developments of these techniques. In this section, we mainly focus on four active research topics in the area of robotic grasping: grasp quality measures, grasp planning, grasp learning, and adaptive grasping under uncertainty.

2.3.1 Grasp Quality Measure

Grasp quality measure is an essential component of grasp planning. Grasp configuration of robotic hands is typically determined by optimizing with respect to a user-defined quality metric. A grasp quality metric is an index or function that quantifies the performance of candidate grasps. Grasp quality measure methods can be basically categorized into two groups: analysis-based and data-driven methods. Analysis-based methods considered the physical and mechanical properties involved in grasping. Roa et al. [143] classified geometric methods into three subgroups: measures based on algebraic properties of the grasp matrix G, measures based on geometric relations of the contact points and measure, and measures considering limitations on the finger forces. These methods took into account form- or force-closure constraints during grasp quality measures. These methods were unable to respond to unexpected disturbances. Hence, certain works made use of tactile information to grasp quality measures. Krug et al. [93] predicted the grasp quality by using wrench-based reasoning from tactile information. In that work, the grasp quality metric was approximated with a linear program. Moreover, task constraints should also be considered in grasp quality measures. In addition, Borst et al. [13] proposed the concept of Task Wrench Space (TWS), in which tasks were characterized by a set of wrenches. The TWS could be approximated by a convex centered at the object origin. Analysis-based methods could be quickly transferred to different objects and grippers. However, one shortcoming of these analysis-based methods was that they required prior knowledge of the object model and contact positions.

Recent works have focused more on data-driven methods that tackled the grasp quality measure problem with machine learning. Grasp quality metrics were learned from certain existing grasping examples generated from human demonstrations or simulations. Johns et al. [82] trained a CNN to approximate a grasp quality function. In that work, the CNN took a depth image as input and outputted the scores of candidate grasps. Similarly, Mahler et al. [112] introduced a Grasp Quality CNN (GQ-CNN) that predicted the success probability of grasps directly from depth images. Apart from visual data, there have been some works that predicted grasp quality from tactile data [9, 175]. In these works, sensor features were first extracted from tactile data and then used to train classifiers that approximate the grasp quality functions. Because visual and tactile are two important sensory modalities, some researchers proposed combining visual and tactile modalities to measure grasp qualities [58, 18]. In contrast to analysis-based methods, data-driven methods can predict grasp quality directly from sensory data without accurate object models. However, data-driven methods also suffer certain shortcomings. A major issue of data-driven methods is that they require many grasping examples for

model training. Moreover, data-driven methods are difficult to generalize to new tasks or new robotic hands.

2.3.2 Grasp Planning

The objective of grasp planning is to find feasible grasp configurations of robotic hands to achieve object grasping tasks. Typically, a feasible grasp configuration of a robotic hand consists of four components [11]:

- Approach vector with that a robot hand approaches the target object.
- Hand pose (i.e., hand position and orientation).
- Grasping/contact points on the object surface.
- Hand joint configuration.

Most of the previous works formulated grasp planning as a constrained optimization problem with respect to grasp quality metrics. Classical works on grasp planning focused on finding feasible grasp configurations by considering form- or force-closure as the grasp quality measure [143]. Owing to task requirements and gripper constraints could affect the grasp choice, Gori et al. [55] implemented three-finger precision grasp planning on incomplete 3D point clouds. Those works optimized grasp configurations accounting for both feasibility and stability requirements. EI-Khoury et al. [43] formulated the planning of dexterous grasping as a constrained optimization problem. In that work, the feasible grasp configuration was generated subject to both hand kinematics and grasp stability. Because object grasping is intrinsically complicated, the optimization objective and constraints are typically difficult to construct. To reduce the complexity of grasp planning. The whole configuration space of a robotic hand is usually high-dimensional, especially for multi-fingered robotic hands. Several approaches have been introduced to reduce the search complexity of gasp configurations for fast grasp planning. Ciocarlie et al. [29] analyzed human hand synergies and introduced *eigengrasps* that represented low-dimensional posture subspaces. Their experimental results showed that *eigengrasps* could be instrumental in deriving practical pre-grasp shapes for several complicated robotic hands. The same idea could also be found in [49]. There have also been some works that reduced the complexity of grasp planning by generating a set of hand pre-shapes [119] or detecting the grasp point on an object surface [6]. In addition, Hang et al. [62] proposed the hierarchical fingertip space (HFTS) for multi-fingered precision grasp planning. This space was defined as a hierarchy of surrogate solution spaces of grasping. The final solution was searched in the HFTS (a local space) instead of the high-dimension configuration space.

The performance of grasp planning can be affected by uncertainties arising from sensor noise and environmental changes. There have been previous works considering uncertainties in grasp planning. Hsiao et al. [70] addressed the grasp planning problem considering uncertainties. A Bayesian framework was proposed to evaluate the probability of grasp success, considering the uncertainty in object identification, gripper

positioning, and hand pose. Weisz et al. [176] evaluated the probability of force closure subject to the uncertainty in object pose. Their experimental results demonstrated that the probability of force closure was more robust than that of deterministic wrench space metrics. In terms of the object shape uncertain, Laskey et al. [97] considered 2D grasp planning with uncertainties. In that work, shape uncertainty was represented with a Gaussian process implicit surface. Some works took advantage of tactile sensing to reduce uncertainties. In addition, Hsiao et al. [71] presented a decision-theoretic approach for robust grasping under object pose uncertainty. Tactile sensing was used to update the belief state. The grasp configuration was optimized continuously until a high degree of robustness was reached.

2.3.3 Grasp Learning

Although analysis-based methods have demonstrated their effectiveness in implementing certain specific grasping tasks, analysis-based grasp planning methods largely relied on prior information about objects and robotic hands. Many researchers have explored machine learning methods for robotic grasping. Rapid advances in machine learning bring a lot of progress in robotic grasping. Learning-based methods allowed robots to plan grasp configurations directly from sensory data (such as depth images or point clouds). In recent decades, several machine learning methods (such as supervised learning, self-supervised learning, and RL) have been employed for robotic grasping.

Recently, deep neural networks have achieved massive success in a variety of fields, such as computer vision [52] and audio processing [66]. Some researchers have used deep neural networks to detect grasp configurations. Redmon et al. [139] presented a real-time approach based on CNN for robotic grasp detection. The network approximated a mapping from an RGB image to graspable bounding boxes for object grasping. Guo et al. [57] proposed a shared CNN to implement object discovery and grasp detection simultaneously. Pas et al. [134] detected grasp configurations directly from point clouds by using deep learning techniques. Some other similar detection approaches have also been presented, but they had different representation methods of grasp configurations (such as grasp region [150], grasp rectangle [139], or two-point grasp [57]). These methods relied on a strong assumption that each object contained one single grasp configuration, which limited their practical application. Instead of detecting grasp configurations directly from visual data, some recent work used deep neural networks to approximate grasp quality metrics that were used to guide the research of feasible grasp configurations. Mahler et al. [112] employed deep learning to plan grasps with synthetic points dataset and analytic grasp metrics. The synthetic points dataset was called as Dex-Net 2.0, which included point clouds, grasp configuration, and analytic grasp metrics. Levine et al. [101] used deep learning to learn hand-eye coordination for robotic grasping. In that work, CNN was trained to predict the grasp quality. In this case, one key problem was how to search for the grasp configuration with the highest quality in high-dimensional configuration space. To address this problem, Mahler et al. randomly sampled a set of candidate grasps and ranked them to find the grasp with the highest quality [112]. In addition, Johns et al. learned a quality function considering gripper pose uncertainty and then used a random search algorithm to find a feasible grasp con-

figuration [82].

Supervised learning-based approaches were highly reliant on a large number of labeled samples. However, data collection required either tedious human labeling. The labeling of samples was highly dependent on expert knowledge. Some works used self-supervised learning approaches to build a dataset for grasp planning [113, 123]. Murali et al. [123] presented a curriculum accelerated self-supervised learning approach that detected grasps directly from visual information. This approach ordered the sampling of training data with respect to control dimensions.

RL algorithms have been widely used for grasp planning. Different from supervised learning-based methods that relied on certain existing datasets, RL algorithms allowed robots to learn grasp configurations from interactions with their environments. Gualtieri et al. [56] formulated the grasping problem as a Markov Decision Process (MDP). To solve it, they built a deep RL to train a robot to focus on task-relevant parts of an object and grasp it. Osa et al. [132] presented a hierarchical RL framework to learn grasping policies. Li et al. [107] presented an RL strategy for object grasping with a mobile manipulator. In that work, RL was employed to learn the trajectories that were represented with Dynamic Movement Primitives (DMPs) in the joint space. In addition, Gbrain et al. [136] explored deep RL for vision-based robotic grasping. A variety of RL methods, including Q-learning, have been evaluated in a realistic simulated benchmark. Instead of only learning grasp actions, RL algorithms can be used to multiple manipulation skills simultaneously. Zeng et al. [181] used model-free deep RL to learn synergies between pushing and grasping. Two CNNs were trained to map from visual observations to actions (i.e., pushing and grasping). Mohammadi et al. [120] proposed mixed-reality deep RL for a reach-to-grasp task. In that work, they carried out DRL for grasping in simulation before actual actions were carried out in the physical environment.

Compared with analysis-based methods, learning-based approaches determine feasible grasp configurations directly from sensory data (such as RGB-D or point clouds) without accurate object models. However, current learning-based methods are essentially limited to certain simple parallel-jaw grippers and specific scenes. A computational model trained by supervised learning for grasping is not allowed to be directly used in a new situation and new task. Meanwhile, because RL suffers from the curse of dimensionality, it is still a challenge to learn a grasping skill for a multi-fingered robotic hand with high-dimensionality.

2.3.4 Adaptive Grasping under Uncertainty

In real-world applications, unpredictable disturbances and uncertainties arising from perception may occur when robotic hands implement object grasping tasks. These disturbances and uncertainties may cause a planned-to-be stable grasp to be unstable. Iberall et al. [75] summarized the functional demands of stable grasps as follows:

- The fingers of a hand apply forces on an object surface in order to react to the object's weight and friction.
- The fingers contact the object and impart motion to the object in order to achieve some task (e.g., pick-and-place object.)

- Sensor information is gathered as feedback to describe contact states between the hand and the object in order to enable grasping stability.

Numerous grasp adaptation strategies have been introduced for robots to adjust current grasp configurations to improve grasp stability. Previous works on grasp adaptation can be basically categorized into two groups: grasp adaption by adjusting the contact points and grasp adaption by adapting the contact force between hands and objects.

One of the grasp adaptation strategies is to adjust the contact points to improve the performance of object grasping under uncertainties. Xue et al. [178] searched for the optimal contact points by moving the fingers to their neighboring joint positions until the local maximum grasp quality was reached. Mavrogiannis et al. [116] gradually improved the grasp quality subject to both geometric and mechanical constraints. In that work, they implemented sequential perturbations on the contact points to derive the optimal grasp. Usually, it is difficult to search for the optimal contact point in the high-dimension configuration space. To reduce the complexity of grasp adaptation, some works used prior good grasp examples as guidance for grasp adaptation. Dang et al. [33] introduced a grasp adjustment strategy by utilizing local geometric similarity. The proposed method adjusted the grasp configuration by finding similar local geometries of novel objects. Information from tactile sensors can be used to improve the performance of object grasping. Chebotar et al. [21] used RL to learn re-grasping behavior. In that work, the robotic hand adjusted the unstable grasps based on tactile perception and improved them over time. In addition, Hogan et al. [67] took advantage of tactile information to search for a grasp adjustment to improve grasp stability.

One other way for grasp adaptation is to adapt the contact force between hands and objects. Robots that operate in real-world environments should be able to maintain feasible contact forces on objects without dropping or crushing them. Tactile feedback has been widely used to perceive the interactions between robotic hands and objects. There have been some works making use of tactile information to guide the adjustment of the grasp force. Delgado et al. [36] adapted the contact configuration regarding the position and magnitude of the contact force in order to avoid sliding. The tactile information was represented by a tactile image that was a combination of multiple dynamic Gaussian. Additionally, Romano et al. [146] developed a human-inspired grasp controller that adjusted the contact force according to tactile feedback. In that work, the contact force could also be estimated from tactile data as a feedback signal for grasp control. Some works integrated a slip detector with a force feedback controller to adapt the contact force. Apart from using tactile information to estimate contact force, some works used tactile information for slip detection. Veiga et al. [171] used a random forest classifier to detect the presence or absence of slippage during grasping. Given the detected results, a force feedback controller was employed to adjust the contact force to stabilize the objects. Furthermore, Veiga et al. [169] proposed an independent tactile grip stabilization controller to ensure that slippage did not occur locally. The slip detection was realized by training an SVM. In addition, Su et al. [158] combined force estimation with slip detection for grasp control. They used a force feedback controller, exploiting the estimated results to pick up objects with various weights and texture.

2.4 In-hand Manipulation

In-hand manipulations using multi-fingered robotic hands requires finding a feasible action sequence to change the object pose. Programming a multi-fingered robotic hand to achieve in-hand manipulation is still a challenging problem. Traditionally, trajectory optimization methods have been proposed, which formulate the manipulation planning problem as a constrained optimization problem. Mordatch et al. [122] proposed a contact-invariant optimization method for the synthesis of dexterous hand manipulation. In addition, Kumar et al. [94] employed a model Predictive Control (MPC) algorithm to perform online trajectory optimization for dexterous manipulation. Owing to the high dimensionality of the robotic hand and the non-linearity of the constraints, the optimization formula of in-hand manipulation is difficult to construct accurately.

Some works employed imitation learning methods to learn in-hand manipulation from humans. Jakel et al. [80] used multiple human demonstrations to learn a planning model for dexterous manipulations. In addition, Gupta et al. [59] took human-demonstrated motions as desired motions and used a policy search method to optimize a policy for in-hand manipulations. These methods largely rely on high-quality human demonstrations, which are difficult to obtain. There exists an inconsistency between the human hand and robot hand structures.

More recent works focus on using deep reinforcement learning (DRL) algorithms to learn in-hand manipulation. DRL uses experiences obtained from interactions with the environment to learn manipulation behaviors that maximize a reward function. Zhu et al. used a DRL algorithm for a multi-fingered robotic hand to learn valve rotation, box flipping, and door opening [182]. The OpenAI team demonstrated that a real physical Shadow Hand could learn dexterous manipulation using a DRL algorithm [131]. Compared with trajectory optimization and imitation learning methods, DRL algorithms enabled agents to learn in-hand manipulation directly from interactions with its environment. However, one shortcoming of DRL algorithms is their sample complexity, which limits their application in learning complex manipulations. Some works that exploited external knowledge have been proposed to reduce the sample complexity of DRL algorithms. Rajeswaran et al. [138] incorporated human demonstrations into a model-free DRL. In that work, they pre-trained a control policy with demonstration data and subsequently fine-tuned the policy with DRL. With a number of human demonstrations, the sample complexity of the DRL could be reduced dramatically. Exploration is a key process in manipulation learning with DRL, especially for complex in-hand manipulations. There have been some works that designed heuristic exploration strategies to reduce the sample complexity of DRL. In addition, Achiam et al. [2] designed surprise-based intrinsic motivation for DRL. They formulated surprise as the KL-divergence of the state transition probability distribution, which guides the exploration. The incorporation of external knowledge learned from human demonstration, or its learning history is crucial to improve the learning efficiency of DRL.

Chapter 3

Optimization for Reach-to-grasp Movements

3.1 Introduction

The rapid development in robotic research has enabled robots to work autonomously in unstructured environments. One essential ability for the robots is to implement RTG tasks stably [108, 165]. There are three main sub-tasks required to complete the RTG task: detecting the object of interest from the table, determining a feasible grasp configuration, and generating a collision-free trajectory to reach it. Therefore, the implementation of RTG tasks contains typically three main aspects: object detection, grasp planning, and trajectory generation. Due to the lack of object models and uncertainties from perceptual noise or kinematic errors of the robots, the stable implementation of RTG tasks remains an open problem.

To grasp the object of interest stably, a feasible grasp configuration is required. Typically, grasp planning is formulated as a constrained optimization problem. Several grasp planning methods have been proposed [12, 148, 40, 105]. Most of these methods employ numerical optimization [105] or search algorithms [40] methods to solve it. Most of these methods suffer from two drawbacks. First, these methods rely on accurate object models or good grasping examples. However, objects in unstructured environments are always partially perceived due to visibility constraints. Thus, these methods are limited to grasp known or familiar objects and difficult to generalize to unknown objects. Second, these methods use a handcrafted quality metric to evaluate the performance of candidate grasps. Since the grasping process is intrinsically complicated, the grasp quality metric is typically difficult to construct manually. Their performance decreases when implementing in unstructured environments. In this work, the objective is to design a grasp planning approach, especially considering unknown objects in an unstructured environment.

Recently, there have been several interesting approaches that use deep learning techniques for grasp planning [99, 139, 100]. These approaches formulate grasp planning as a regression problem, where a CNN is trained to build a mapping function from images to grasp configurations. However, most of these methods make a strong assumption that

each object contains a single grasp configuration, which is not robust to uncertainties. Instead of directly predicting a grasp configuration from the image, the proposed grasp planning method first learns a quality function using CNN. A Bayesian-based search algorithm is designed to find a feasible grasp configuration with the highest quality. In this way, the uncertainties are taken into account and the grasp with the highest quality is found. Most similar to us are [82, 112, 134]. However, there are two differences between the mentioned methods and the proposed method. First, the proposed method takes advantage of the information (i.e., object identity and location) computed from object detection to assist the grasp planning. Second, we employ a Bayesian-based search algorithm instead of random searching algorithms [112] to determine the grasp configuration with the highest quality iteratively. That is important to grasp the object of interest stably.

Another critical challenge is to generate a collision-free trajectory that drives a gripper to reach the object of interest. Modeling and reproducing natural human movements have been widely studied in recent years. The most direct approach is to employ function approximation methods to represent human demonstrated trajectories and then generalize it to new environments. The existing approaches for trajectory representation mainly include two categories: dynamic-based and probability-based approaches. Dynamic-based approaches represent human movements as a set of dynamic systems [77, 39]. Probability-based approaches compute the probability distribution of the occurrence of robot state [19, 85]. Most of these methods only suit for simple movements and are easily affected by uncertainties. Some studies learn a cost function from human demonstrations as optimal criteria and then use optimization techniques to produce the desired trajectory. The idea is usually found in inverse reinforcement learning methods [1] or inverse optimal control methods [121]. However, a true cost function is difficult to be learned due to the stochastic of human movements. Although these methods do well in some motion planning applications, these methods are limited to generate simple movements and have a poor generalization ability. In RTG tasks, the trajectory generation method is required to generalize quickly adapt to the changing environment.

Studies of neuroscience demonstrated that human motor control is based on an internal model theory that the human maintains a dynamic model that predicts the consequences of motor commands and employs an inverse model to produce motor commands to achieve the desired movement [84, 78]. Inspired by this theory, a model-based trajectory generation method is proposed in this work. A forward dynamic model in Cartesian space is learned from human demonstrations to reproduce the state transition behavior of the human arm and an inverse model constructed as a linear Gaussian model is on-line learned to produce robot control inputs. Compared with the previous trajectory generation methods [19, 72], the proposed model-based trajectory generation approach can generalize quickly to new tasks.

In this work, we address the problem of the implementation of RTG tasks in an unstructured environment. The objective is to reach to grasp the object of interest stably in an unstructured environment. The two essential techniques (i.e., stable grasp planning and trajectory generation) are studied in this work. The following three contributions are presented.

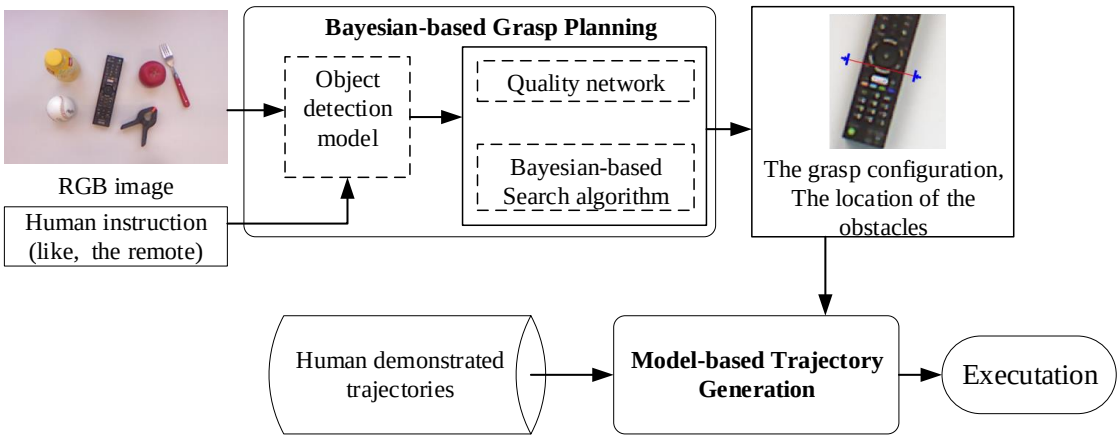


Figure 3.1: Overview of the proposed optimization framework for RTG tasks. Firstly, given a personal instruction and an RGB-D image of the table scene as inputs, the object of interest is recognized and segmented out from the table scene. A Bayesian-based search algorithm determines the grasp configuration with the highest grasp quality computed by a trained network. The planned 2D grasp configuration in the image frame is further projected to a 3D grasp pose in the robot frame. At the same time, the approximated poses of obstacles, i.e., other objects around the object of interest, are computed. Secondly, a model-based trajectory generation approach is employed to generate a collision-free trajectory. Finally, the robot executes the reaching movement and the grasp operation to implement the RTG task.

- An optimization framework that combines stable grasp planning with trajectory generation is proposed for the implementation of RTG tasks.
- A Bayesian-based search algorithm is proposed for grasp planning. Compared with random search algorithms, the proposed search algorithm has a chance to find a more stable grasp configuration.
- Inspired by human internal model theory, a model-based trajectory generation method is designed to generate collision-free reaching movement for the robots.

In the rest of this section, Section 3.2 introduces the stable grasp planning approach. The model-based trajectory generation is presented in Section 3.3. Experiments and their results are presented in Section 3.4. Finally, the conclusion and future work are discussed in Section 3.5.

3.2 Stable Grasp Planning with a Learned Quality Metric

To implement RTG tasks effectively, we proposed an optimization framework that contains two main components: a stable grasp planning method and a trajectory generation method. Figure 3.1 illustrates an overview of the proposed optimization framework

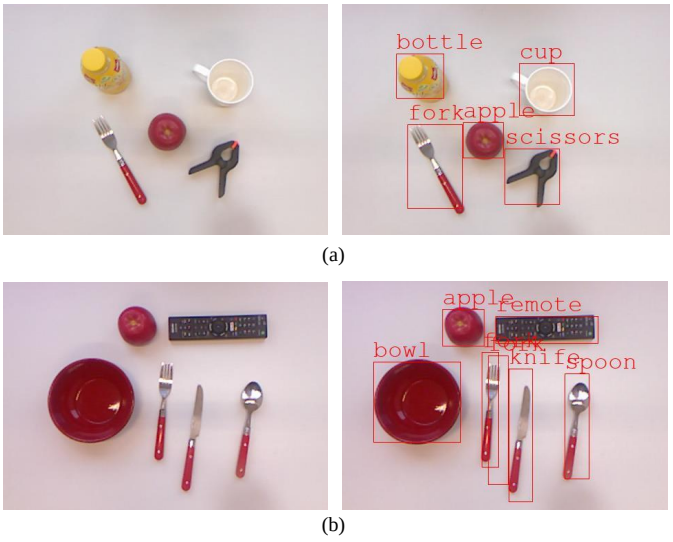


Figure 3.2: Examples of object detection. The texture represents the object identity, and the bounding boxes approximately denote the location of the object of interest.

for RTG tasks. In this section, we introduce the stable grasp planning approach and its components, like object detection, the method for evaluating the grasp quality, the Bayesian-based search algorithm.

3.2.1 Object Detection

Object detection aims not only to recognize the object of interest but also to segment the object from a table for grasp planning. In contrast to previous work that uses feature extraction methods to detect objects [183], a learning-based detection method is employed to localize and recognize the object of interest in an unstructured environment.

The proposed object detection model refers to the architecture proposed by Redmon et al. [140]. Since the public object datasets, like the COCO [1] and VOC [2] dataset, mainly focus on outdoor scenes instead of the table scene and the household objects, it is infeasible to directly use these datasets to train the object detection model for RTG tasks. To meet the requirement of RTG tasks, a Table Object Class (TOC) dataset containing 16 household objects is collected to fine-tune the object detection model further. The details of the TOC dataset are introduced in Section 3.4.

During the model training phase, the COCO dataset is first used to pre-train the proposed object detection model. The TOC dataset is used to fine-tune the model further. The training of the proposed object detection model is detailed in Section 3.4. The object detection model takes an RGB image as an input and outputs the recognized identity and bounding box of objects on the table. Figure 3.2 shows the two examples of object detection.

[1] http://cocodataset.org/
[2] http://host.robots.ox.ac.uk/pascal/VOC/

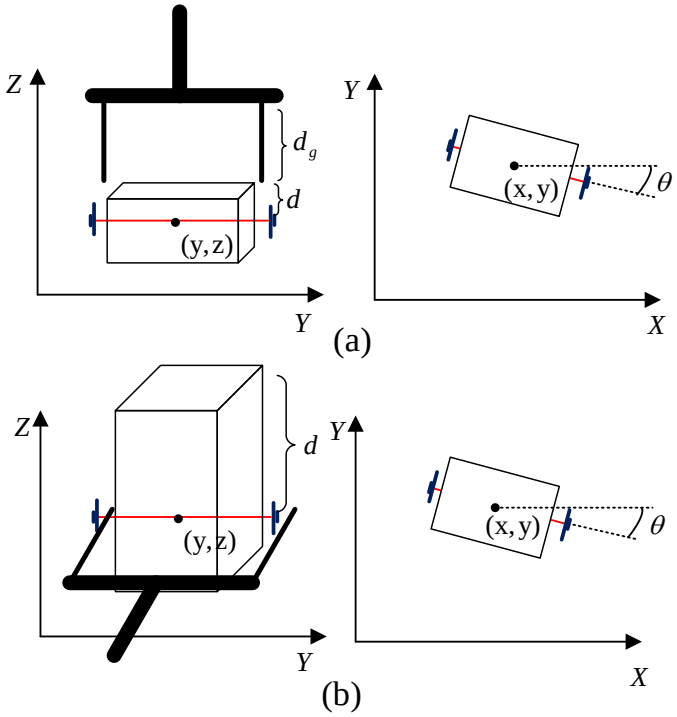


Figure 3.3: Representation of grasp configuration. (a) The gripper grasps the object from top to down when $d <= d_g$. (b) The gripper is parallel to the table plane to grasp the object when $d > d_g$.

3.2.2 Learning Grasp Quality Metric

Evaluating the quality of a grasp configuration is an important aspect of grasp planning. In this work, a deep neural network is employed to learn a quality function $q = Q(g, I)$ that predict a quality q concerning a depth image of an object I and a grasp configuration g.

In this work, the robot uses a parallel-jaw gripper for RTG tasks. Referred to the two-point grasp representation [57], in the work, the grasp configuration is represented by $g = \{x, y, z, \theta\}$, as shown in Figure 3.3. $\{x, y, z\}$ denote the grasp center point relative to the camera frame and the angle θ is the rotation angle in the table plane. The distance between the two points is set to a fixed value in advance according to the gripper mechanism. The grasp configuration specifies the target pose of the gripper before the robot grasps an object. During the grasp execution phase, the gripper is kept perpendicular to or parallel to the table plane. The 2D grasp configuration in the image frame is converted to a 3D grasp pose in the robot frame. The coordinate transformation in RTG tasks is detailed in Section 3.4.

To measure the performance of grasp configurations, a quality network based on CNN is trained. Figure 6.4 shows the architecture of the proposed quality network. The quality network uses four convolutional layers to learn the hierarchical features from depth images. In each convolutional layer, the input of the layer is first multiplied with a set of filter and then an activation function (i.e., a rectified linear unit) is applied to

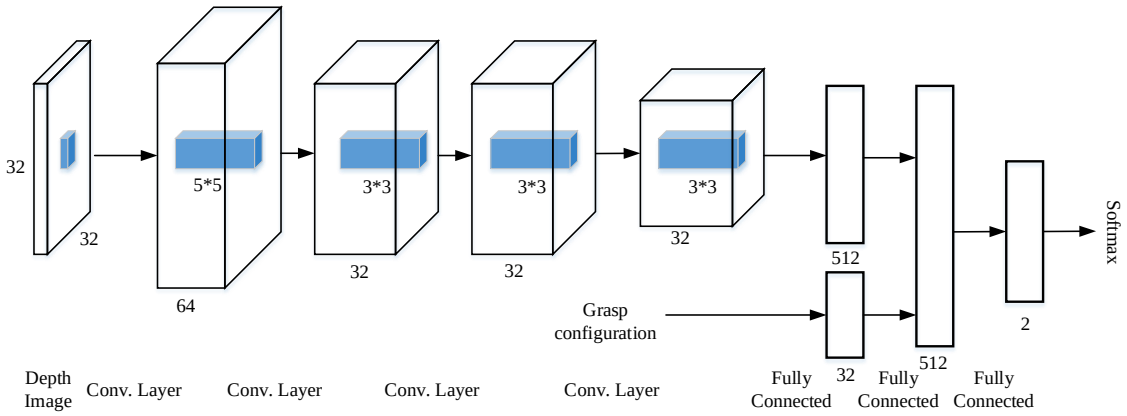


Figure 3.4: The proposed architecture of the quality network. The first four convolution layers are used for image feature extraction. The feature points of the feature maps are concatenated with the grasp configuration and then pass through two fully connected layers to predict the quality score.

compute the output of the layer. Following the four convolutional layers, three fully-connected layers are used to combine the learned feature with a grasp configuration for high-level reasoning. Finally, the last fully-connected layer is passed through a soft-max function to predict the quality score concerning a queried image and a grasp configuration.

$$P(y=1) = \frac{e^{Q}}{e^{P}+e^{Q}} \tag{3.1}$$

Where P and Q denotes the two outputs of the last fully-connected layer. At the training phase of the quality network, the cross-entropy function is used to define the loss function. The quality network is fitted finally to a probability of quality condition on an object image and a candidate grasp. The training and evaluation processes of the proposed quality network are detailed in Section 3.4.

3.2.3 Grasp Configuration Optimization with Bayesian-based Search

Given the trained quality function $Q(I,g)$, the grasp planning problem is formulated as an optimization problem. The objective is to find the best grasp g^* to maximize the quality function $Q(I,g)$.

$$g^* = \underset{g \in G}{\operatorname{argmax}}\, Q(I,g) \tag{3.2}$$

Due to the complexity of solving the grasp optimization problem, previous work mainly uses search algorithms to find the feasible grasp configuration for objects [134]. The Monte Carlo (MC)-based search algorithm is a popular search algorithm, which first samples randomly a large set of candidate grasps and ranks all the candidate grasps to find the grasp configuration with the highest quality [112, 134]. However, for a stable grasp, exploring the quality function directly only is relieve complexity. Therefore, it is essential to employ a more efficient search algorithm to find the feasible grasp configuration, in particular, considering the uncertainties. In this work, a search al-

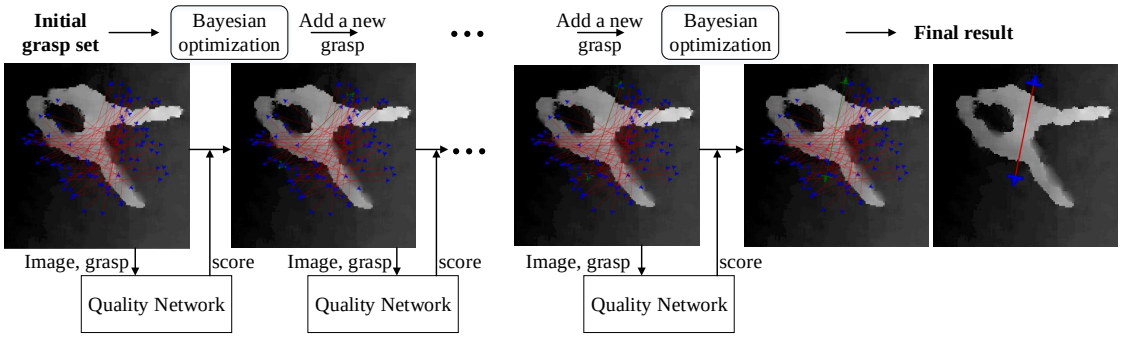


Figure 3.5: Illustration of the proposed Bayesian-based search algorithm. Firstly, an initial set of candidate grasps is first sampled from a depth image to form a grasp dataset. The grasp dataset is used to fit the GP model. Secondly, the local optimum is then determined by maximizing Eq. 3.6. The quality of the determined grasp configuration is computed using the trained network. Then Bayesian optimization determines a new grasp configuration (Green) with a higher quality score for the next query point. The search procedure is iterated for each optimum until convergence.

gorithm based on Bayesian optimization is introduced to determine the feasible grasp configuration. Bayesian optimization is a popular global optimization technique for the black-box function and has been successfully applied in solving non-convex optimization problems [83]. Compared with the MC-based search algorithm, the proposed Bayesian-based search algorithm does not require maintaining a large set of candidate grasps significantly.

During the search phase, Bayesian optimization mainly contains two ingredients [128]. The first one is a surrogate model which represent a distribution over the family of the quality function, and the second one is an acquisition function which select the next query point to achieve the good sampling performance. Hence, instead of exploring the quality function directly, the behaviour of the quality function is first learned by the surrogate model which fitted by a GP model. In the GP model, the distribution of the quality score q_{gp} is fully specified by a mean function $m(g)$ and covariance function $k(g,g')$.

$$q_{gp}(g) \sim \mathscr{GP}(m(g),k(g,g')) \tag{3.3}$$

where a zero mean function is assumed, i.e., $m(g) = 0$. The squared exponential covariance kernel function is defined as the covariance function

$$k(g_i,g_j) = \sigma_f^2 exp(-\frac{\| g_i - g_j \|^2}{2l^2}) + \sigma_n^2 \delta(g_i g_j) \tag{3.4}$$

where l is the bandwidth of the kernel, σ_f^2 and σ_n^2 are the function and noise variances. $\delta(g_i,g_j)$ is the Kronecker delta function. These parameters $\{l,\sigma_f^2,\sigma_n^2\}$ are the hyper-parameters of the GP model, which learned from the training data.

To fit the GP model, an initial set of candidate grasps from the depth image is required to be sampled. In this work, the sampling method presented by Smith et al. [155] is adopted to determine the candidate grasps. These candidate grasps $G^n = \{g_i\}_{i=1}^n$

and its corresponding quality score $Q^n = \{q_i\}_{i=1}^n$ are formed an initial grasp database $D_{grasp}^n = \{(g_i, q_i)\}_{i=1}^n$. Once the GP model is fitted, the quality score $q_{gp}(g_{n+1})$ for a new grasp configuration g_{n+1} given a dataset D_{grasp}^n follows a posterior distribution.

$$p(q_{gp}(g_{n+1})|g_{n+1}, D_{grasp}^n) \sim \mathcal{N}(\mu(g_{n+1}|D_{grasp}^n), \sigma^2(g_{n+1}|D_{grasp}^n)) \tag{3.5}$$

where the mean and the covariance function of the posterior distribution are given by

$$\mu(g_{n+1}) = K(g_{n+1}, G^n)K(G^n, G^n)^{-1}Q^n$$

$$\sigma^2(g_{n+1}) = K(g_{n+1}, g_{n+1}) - K(g_{n+1}, G^n)K(G^n, G^n)^{-1}K(G^n, g_{n+1})$$

After fitting the GP model, we need to determine a new grasp configuration $\hat{g}$ for exploration given the grasp dataset D_{grasp} at each iteration. To trade-off between the exploration and exploitation, we define the Upper Confidence Bound (UCB) policy as the acquisition function a_{UCB} [128]. Hence, the new grasp configuration $\hat{g}$ is selected as the next query point by maximizing the acquisition function, as shown in Eq. 3.6. where β denotes a constant weight that trade-off between the exploration and exploitation.

$$a_{UCB}(g_{n+1}|D_{grasp}^n) = \mu(g_{n+1}|D_{grasp}^n) + \beta\sigma^2(g_{n+1}|D_{grasp}^n) \tag{3.6}$$

Figure 3.5 shows the search process using the proposed Bayesian-based search algorithm. The proposed Bayesian-based search algorithm produces the probability of the quality of each grasp. That allows us to select the grasp configuration that is robust to the uncertainties. The pseudo-code of the proposed stable grasp planning approach is shown in algorithm 1. In the beginning, the object detection model is used to recognize and segment the object of interest based on the individual instruction and the image of the table scene. Then, the Bayesian-based search algorithm is used to find the grasp configuration with the highest probability of success using the quality network.

Algorithm 1 : Stable grasp planning with a learned quality metric

1: **Requires**:the object detection model, the trained quality network $Q(I, g)$.
2: Acquire an RGB-D image I of the table scene.
3: Receive the human instruction, i.e., the name of object identity.
4: Obtain the segmented image I_o of the object o from the image I using the object detection model
5: Sample an initial set of candidate grasps in I_o to form a dataset D_{grasp}
6: **Repeat**
7: Fit the GP model with D_{grasp}
8: Search a local optimum:$\hat{g} = \text{argmax}_g\, a_{UCB}(g|D_{grasp})$
9: Compute the quality with the quality network:$\hat{q} = Q(I, \hat{g})$
10: $D_{grasp} \leftarrow D_{grasp} \cup (\hat{g}, \hat{q})$
11: **until** the grasp configuration with highest quality g^*

3.3 Model-based Trajectory Generation

After determined the grasp configuration for the object, the next step is to generate a trajectory that drives the end-effector to reach the object. Inspired by human internal model theory, a model-based trajectory generation approach is designed, as shown in Figure 3.6. The approach comprises a dynamic model and an inverse model. Moreover, the trajectory adaptation is to makes on-line movement corrections as so to improve its performance. This subsection explains the development of these components.

3.3.1 Learning a Forward Dynamical Model for Optimization

Robot forward dynamic model is learned from human demonstrations to encode the impedance behaviour of the human arm in the Cartesian space. The main assumption underlying the proposed approach is that the forward dynamical model represents the expected state transition of the human arm. This assumption is also made by the trajectory representation method introduced by Ijspeert et al. [77]. Therefore, the following impedance model is defined in Cartesian space to represent human movement behaviour.

$$M\ddot{x}(t) + C\dot{x}(t) + K(x - x_g) = f_t \tag{3.7}$$

where $x(t), \dot{x}(t), \ddot{x}(t) \in \mathbb{R}^3$ are the Cartesian position, velocity and acceleration of the human hand, respectively. $M, C, K \in \mathbb{R}^{3\times3}$ are the virtual mass, damping, and spring matrices, respectively. $x_g \in \mathbb{R}^3$ is the attractor state of the human hand and $f_t \in \mathbb{R}^3$ denotes virtual human controls in the Cartesian space.

Different from the previous methods [77] in which a control policy is directly learned from human demonstrations, this work learn a forward dynamic model from human demonstrations that used to predict the state transition. Based on the above defined human impedance model,the following impedance model is defined in Cartesian space for robots to represent robot movement behavior.

$$M\ddot{x}(t) + C\dot{x}(t) + K(x - x_g) = f_t + u_t \tag{3.8}$$

where $u_t \in \mathbb{R}^3$ is the virtual robot control input in the Cartesian space and $f_t \in \mathbb{R}^3$ is the virtual human control that forces the impedance model to match the demonstrator expectation.

By approximating the human control f, the robot impedance model is able to reproduce the internal behaviour of human movement, i.e., the state transition. Furthermore, the impedance model is used as the forward dynamical model for the robot control. In this work, a function approximation method is used to compute the non-linear virtual human control f. The virtual human control f is defined as a linear combination of Gaussian kernel function.

$$f(x) = \frac{\sum_{i=1}^{k} \omega_i \psi_i(x)}{\sum_{i=1}^{k} \psi_i(x)} \tag{3.9}$$

where $\psi_i(x) = exp(-(x - c_i)^2/2\sigma_i^2)$ represents the Gaussian kernel with a center μ_i and width σ_i. ω_i is the weight of the Gaussian kernels. Hence, the virtual human control f

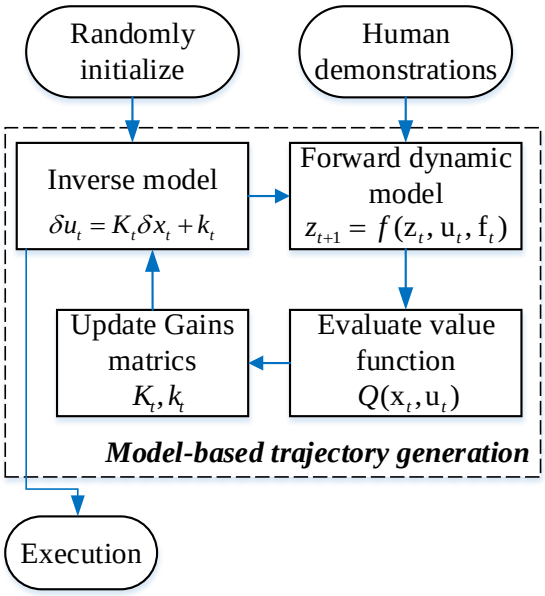


Figure 3.6: Overview of the model-based trajectory generation approach. Combining a forward dynamic model with the inverse model to produce control inputs for robot reaching movements. Trajectory adaptation adjusts the inverse model to correct the movement with respect to new environment.

is fully determined by weights ω. With the recorded human demonstrated trajectories, finding the weights ω is a supervised learning problem. In this work, the Local Weight Regression (LWR) algorithm introduced by Ijspeert et al. [77] is used to a expected sum squared error that computed by using Eq. 3.7 and Eq. 3.9 and then calculate the weights ω. For further analysis, we rewrite Eq. 3.7 into a state-space form of a discrete time system.

$$z_{t+1} = Az_t + B_1 u_t + B_2 f_t + B_3 x_g \tag{3.10}$$

with

$$A = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ M^{-1}K & M^{-1}C \end{bmatrix}, B_1 = B_2 = \begin{bmatrix} 0_{3\times3} \\ M^{-1} \end{bmatrix}, B_3 = \begin{bmatrix} 0_{3\times3} \\ M^{-1}K \end{bmatrix},$$

where $z(t) = [x, \dot{x}]^T$ denotes the robot state that includes the position and velocity of the gripper. $0_{3\times3}$ and $I_{3\times3}$ are the 3×3 zero and identity matrices, respectively. Further, the forward dynamic model is written simply as $z_{t+1} = F(z_t, u_t, f_t)$. During the robot motion control, the reference velocity in the joint space is required, which produces the desired trajectory for the gripper derived from Eq. 3.10. Hence, the inverse kinematics problem needs to be solved, as defined in Eq. 3.11. For the continuity of the motion control, the pseudo-inverse of the Jacobian matrix J is used. q_t denote the joint angle in the joint space and $\dot{x}_t$ is the velocity in the Cartesian space.

$$\dot{q}_t = (J^T J + \gamma I)^{-1} J^T \dot{x}_t \tag{3.11}$$

3.3.2 Optimization of the Inverse Model

After the forward dynamic model is learned, we further construct and optimize the inverse model that produces the robot control u for the reaching movement. In this work, the inverse model is constructed as a time-varying linear-Gaussian controller that defines the robot control u as a function of the robot state $z = [x, \dot{x}]_t$ corrupted by Gaussian noise. The probability of the robot control u is modeled as:

$$p(u_t|x_t) = \mathcal{N}(K_t(x_t - \bar{x}_t) + k_t + \bar{u}_t, \Sigma_t) \tag{3.12}$$

where x_t and u_t are the robot position and the robot control of the current iteration, while $\bar{x}_t, \bar{u}_t$ are sampled from the previous iteration. K_t and k_t are the two important gain matrices of the inverse model that determine the reactive behavior of the robot movement. Σ_t is the variance of robot controls. To adapt to the changing environment, the two gain matrices of the inverse model should be updated according to the new environment. Therefore, a trajectory optimization method is employed to update the two gain matrices of the inverse model iteratively for the desired reaching movements.

In this work, the trajectory optimization is considered in an episodic optimal control setting with episode length T. The Model Predictive Control (MPC) method is employed to adjust the two gain matrices of the inverse model to generate an optimal control sequence $U = \{u_0, u_1, \cdots, u_T\}$. At each time-step t, a H-horizon state-action trajectory $\{\tilde{Z}_t, \tilde{U}_t\}$ is defined as a sequence of state $\tilde{Z}_t = \{z_t, z_{t+1}, \cdots, z_{t+H}\}$ resulted from a control sequence $\tilde{U}_t = \{u_t, u_{t+1}, \cdots, u_{t+H}\}$. $H \leq T$ denotes the MPC horizon. Based on the forward dynamic model introduced in Subsect. 3.3.1, the objective is to find a H-horizon optimal control sequence $\tilde{U}_t^*$ at each time-step t by solving the following H-horizon optimization problem.

$$\begin{aligned} &\underset{\tilde{U}_t}{\operatorname{argmin}} \sum_{i=t}^{t+H} l(x_i, u_i) \\ &\text{s.t.} \quad z_{i+1} = F(z_i, u_i, f_i), \forall i \in t, \cdots, t+H \end{aligned} \tag{3.13}$$

where $l(x_i, u_i)$ is a user-defined cost function. The forward dynamic model $z_{i+1} = F(z_i, u_i, f_i)$ governs the state transition given the robot control u_i, the virtual human control f_i and the robot state z_i.

In RTG tasks, the cost function for trajectory optimization is designed according to the following three criteria. (1) Reaching the target pose at the end of an episode, i.e., the grasp pose of an object in the robot coordinates. (2) The minimum robot controls. (3) Avoiding the obstacles, i.e., the approximated poses of other objects around the object of interest. For the criteria 1) and 3), a quadratic cost function is designed as follows

$$l(x_i, u_i) = (x_i - x_d)^T Q_1 (x_i - x_d) + \dot{x}_i^T Q_2 \dot{x}_i + u_i^T R u_i \tag{3.14}$$

where $x_i, \dot{x}_i \in \mathbb{R}^3$ are the Cartesian position and velocity of the gripper, respectively. x_d is the desired position of the gripper. x_d is the target position when $i = T$. $Q_1, Q_2, R \in \mathbb{R}^{3\times3}$ are the position cost, velocity cost and input cost weighting matrices respectively.

In the optimal control problem, the obstacle avoidance problem is mainly addressed by adding some inequality equations into the optimization problem [51]. To avoid solving a non-convex optimization problem, the obstacle constraints are designed in this work as follows

$$(x_i - x_o)^T Q_1 (x_i - x_o) \sqrt{\frac{\rho_p}{2}} exp(-\frac{\rho_p}{2}(t - t_p)^2) \geq d \tag{3.15}$$

where x_o is the position of an obstacle and d is the allowed shortest distance between the gripper and an obstacle. $Q_2 \in \mathbb{R}^{3\times 3}$ is a weight matrix of the obstacle cost and ρ_p is the temporal spread of the obstacle. This soft obstacle constraint enforces trajectories outside of the obstacles.

Given the learned forward dynamic model and cost function, an iteration Linear-Quadratic-Regulator (iLQR) method is implemented within the MPC context to perform trajectory optimization. The iLQR method updates the inverse model by perform two phases iteratively, i.e., a backward pass and a forward pass, with respect to an initial condition $\{z_0, u_o\}$, where z_0 is set according to the robot initial position and u_0 is set to 0. In the backward pass, the iLQR method compute the Q-value function $Q(x_t, u_t)$ and the value function $V(x_t)$ at each time step t. These functions are quadratic and computing according to the following recurrence:

$$\begin{aligned}
Q_{x,t} &= l_{x,t} + F_{x,t}^T V_{x,t+1} \\
Q_{u,t} &= l_{u,t} + F_{u,t}^T V_{x,t+1} \\
Q_{xx,t} &= l_{xx,t} + F_{x,t}^T V_{xx,t+1} F_{x,t} + V_{x,t+1} F_{xx,t} \\
Q_{ux,t} &= l_{ux,t} + F_{u,t}^T V_{xx,t+1} F_{x,t} + V_{x,t+1} F_{ux,t} \\
Q_{uu,t} &= l_{uu,t} + F_{u,t}^T V_{xx,t+1} F_{u,t} + V_{x,t+1} F_{uu,t} \\
V_{x,t} &= Q_{x,t} - Q_{ux,t}^T Q_{uu,t}^{-1} Q_{u,t} \\
V_{xx,t} &= Q_{xx,t} - Q_{ux,t}^T Q_{uu,t}^{-1} Q_{ux,t}
\end{aligned}$$

where $Q_{x,t}$ and $Q_{u,t}$ denote the first derivations of Q-value function $Q(x_t, u_t)$ with respect to x_t and u_t, respectively. $Q_{xx,t}, Q_{ux,t}$ and $Q_{uu,t}$ are the second derivations. $V_{x,t}$ and $V_{xx,t}$ are the first and second derivations of the value function $V(x_t)$, respectively. $F_{x,t}$ and $F_{xx,t}$ are the first and second derivations of the forward dynamic model $z_{i+1} = F(z_i, u_i, f_i)$, respectively.

In the forward pass phase, given the computed Q-value function, the inverse model is then updated. the optimal update of robot controls is to minimize the Q-value function.

$$\delta u_t^* = \underset{\delta u_t}{\operatorname{argmin}} Q(\delta x_t, \delta u_t) \tag{3.16}$$

where $Q(\delta x_t, \delta u_t)$ denotes the change in the Q-value function at (x_t, u_t) as a result of small perturbations $\delta x_t, \delta u_t$. To update the inverse model, the optimization problem is solved according to [10] and it solution is a linear relationship between δu_t and δx_t,

given by

$$\delta u_t = K_t \delta x_t + k_t \tag{3.17}$$

where $K_t = -Q_{uu,t}^{-1} Q_{ux,t}$ and $k_t = -Q_{uu,t}^{-1} Q_{u,t}$. Furthermore, the inverse model is further rewritten as

$$u_t = \mathcal{N}(-Q_{uu,t}^{-1} Q_{ux,t}(x_t - \bar{x}_t) + -Q_{uu,t}^{-1} Q_{u,t} + \bar{u}_t, Q_{uu,t}^{-1}) \tag{3.18}$$

where the variance of robot controls is set to $Q_{uu,t}^{-1}$ [102]. $\bar{x}_t$ and $\bar{u}_t$ are the robot position and control sampled from the previous iteration.

By iteratively evaluating and updating the inverse model, the two gain matrices of the inverse model, i.e., K and k, are found concerning the current environment. Finally, the robot executes the control inputs U produced by the optimized inverse model to finish the reaching movement. The Pseudocode of the model-based trajectory generation approach is shown in Algorithm 2.

Algorithm 2 : Model-based trajectory generation

Requires: the weights $Q1_1$, Q_2 and R in the cost function; the mini-distance d and the number of Gaussian kernel k; human demonstrated dataset D_{traj}.
Using the LWR algorithm to approximate human control f with the dataset D_{traj}.
Build the forward dynamic model $\dot{z}_t = f(z_t, u_t, f_t)$ for the robot.
for iteration $k = 1 : K$ **do**
    Reset the robot state.
    for time-step $t = 1 : T$ **do**
        Collect H-horizon state-control trajectory $\{\tilde{X}_t, \tilde{U}_t\}$ and human control $f_{t:t+H}$ using Eq. 3.10
        Update the two gain matrices K_t, k_t using Eq. 3.16
        Compute the new action u_t^* using the inverse model.
        Compute the desired velocity $\dot{x}_t$ using Eq. 3.10.
        Compute the desired joint velocity $\dot{q}_t$ using Eq. 3.11
    end for
end for

3.4 Experiments

This section presents a set of experiments to evaluate the proposed optimization framework for RTG movements and discusses the experimental results.

3.4.1 Evaluation of Stable Grasp Planning

Implementation

As introduced in Section 3.2, the TOC dataset that specific for RTG tasks is required. For this purpose, 16 household objects were chosen to create the TOC dataset, as shown in Figure 5.11. During the construction phase of the dataset, multiple objects from the

Figure 3.7: Objects contained in the TOC dataset.

object categories were randomly selected and separately placed on the table. Kinect depth sensor captured an RGB-D image of the table scene. A total of 1051 images of table scenes were collected. Then, objects in each image were labeled with ground-truth bounding boxes and object identities. To further augment the TOC dataset, the rotation and translation transformations were performed to generate more samples. During transformations, the corresponding label of each image was also adjusted. The final augmented TOC dataset resulted in 6306 samples. The TOC dataset was split randomly into a training set (90%) and a testing set (10%).

Training the proposed quality network required a grasping dataset that contains numerous objects and ground-truth labeled grasps. However, it is difficult to construct such grasping dataset, due to the lack of object mesh model and the amount of time required to label the objects. Therefore, an public grasping dataset (Dex-net2.0) proposed by Mahler et al. [112] was utilized. This grasping dataset contained 6.7 million 3D object models that labeled with multiple ground-truth parallel-Jew grasps and the corresponding quality. A subset of 190000 samples was extracted from this grasping dataset. A 3-tuples formed each sample: a raw depth image, a ground-truth labeled grasp, and a quality. Similarly, the subset was randomly split into a training set(90%) and a testing set(10%), respectively.

The training parameters of the proposed quality network are set as follows: the quality network was trained for 20 epochs. One epoch performs one forward pass and one backward pass of all the training examples. The batch size of the training was set as 64, which defines the number of samples that are going to be propagated through the quality network. The initial learning rate was 0.005, and an exponential decay with a decay rate of 0.96 was applied to lower the learning rate as the training progressed. To avoid the over-fitting problem, a drop-out layer with a probability of 0.5 is applied after the first fully-connected layer. The Stochastic Gradient Descent (SGD) method with a momentum rate of 0.9 was employed to optimize the weights of the quality network.

Figure 3.8: Examples of discovering an object of interest on a table. Nine household objects were recognized and localized from multi-objects. The bounding box of each object gives an approximate location of the object, and the texture of the box denotes its identity.

Experimental Results and Discussions

The performance of the object detection model is first evaluated. The fine-tuned detection model is compared with the original detection model that trained on the COCO dataset. During the evaluation phase, the two models predicted the bounding box and object identity of the object in the testing set. The detection accuracy of the original detection model and the fine-tuned detection model are 78.6% and 82%, respectively. After the object detection model is fine-tuned, the detection accuracy obtained a 3.4% improvement. Moreover, the original model had a low probability of detecting the small-scale object correctly, like, the pen and the sports ball. The main reason was that the table scene for RTG tasks was seldom taken into consideration in the COCO dataset. After fine-tuning the object detection model, its performance was improved and can suit for the stable grasp planning for the RTG task. Figure 3.8 shows some examples of object detection.

The prediction accuracy of the proposed quality network is important for grasp planning. Hence, we use the testing dataset to validate its prediction accuracy. Figure 3.9 shows the prediction accuracy. The quality network finally converged to high accuracy (about 93%). The high accuracy of the quality network is a necessary condition for the proposed Bayesian-based search algorithm to find the feasible grasp configuration. Hence, the trained quality network can meet the requirement of grasp planning.

Next, the performance of the proposed Bayesian-based search algorithm was evalu-

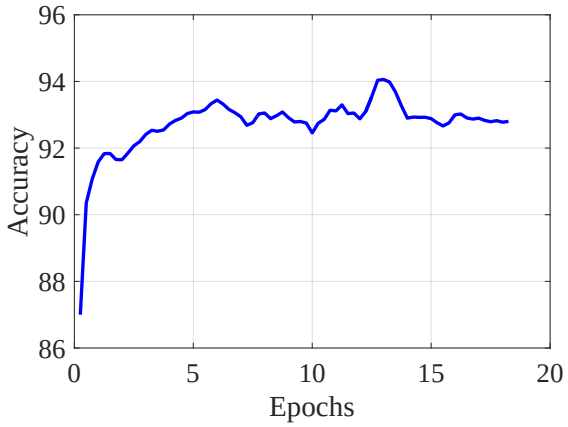


Figure 3.9: The prediction accuracy of the quality network.

ated. The MC-based search algorithm is a popular search algorithm and has been used in several existing grasp planning approaches [112, 134]. Hence, the proposed Bayesian-based search algorithm was compared with the MC-based search algorithm. Figure 3.10 shows the search process of the two different algorithms. In the MC-based search algorithm, a large set of candidate grasps was first sampled from an image and then ranked using the quality function to find the grasp configuration with the highest quality. From figure 3.10.a, it can be seen that there were many bad grasps with low quality among the candidate grasps. Thus, the MC search algorithm only converged to a local optimum due to its sparse sampling. Bayesian optimization is a sequential algorithm for global optimization of the quality function. Although the quality function was considerable complex, the Bayesian optimization explored the behavior of the quality function iteratively to find a feasible solution. From Figure 3.10.b, we find that the new grasp with the higher expected quality than the previously determined grasp is obtained. The search procedure was iterated for each optimum until convergence. Here, six iterations were carried out in the search process. Compared with the MC-based search algorithm, the Bayesian-based search algorithm was able to find the grasp configuration with higher quality. Hence, the Bayesian-based search algorithm was more efficient than the MC-based search algorithm. Figure 3.13 shows some examples produced by the proposed grasp planning method.

To analyze the parameter β, a sensitivity analysis experiment was carried out. In the experiment, five groups of experiments were implemented, and seven different parameter value $\beta \in [1,4]$ were tested in each group. The maximum quality was collected after the fifth search step in each group. Figure 3.11 shows the collected maximum quality under different parameter values in the five groups of experiments. It was observed that when the parameter $\beta \in [2.5,3]$, it was more likely to get the maximum quality score.

Robotic grasping in a cluttered environment is a challenge. The performance of the proposed grasping planning approach in a cluttered environment is also evaluated. In experiments, the clutter levels varied from mild to complex. Figure 3.12 shows the results of the grasp planning for three different objects considering different clutter levels. It can be seen that the proposed planning method performs well in mild clutter environ-

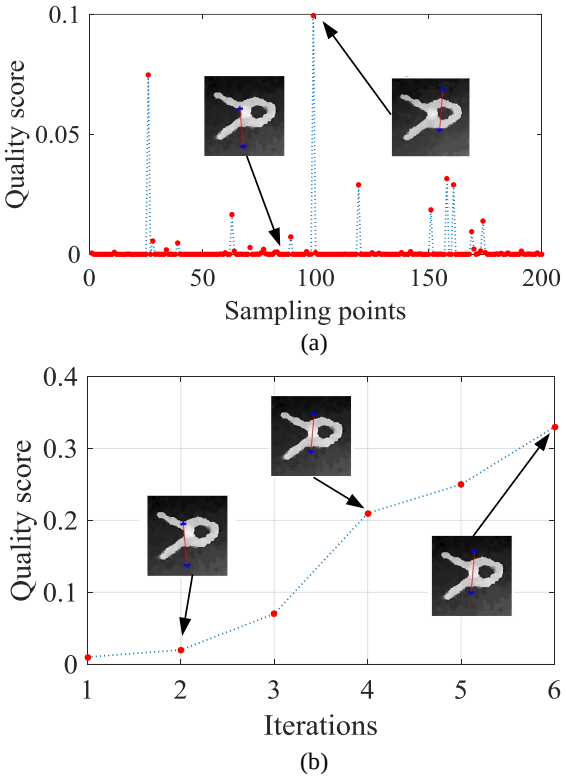


Figure 3.10: A comparison of the results from two different search algorithms. (a) The MC-based search algorithm, (b) the Bayesian-based search algorithm.

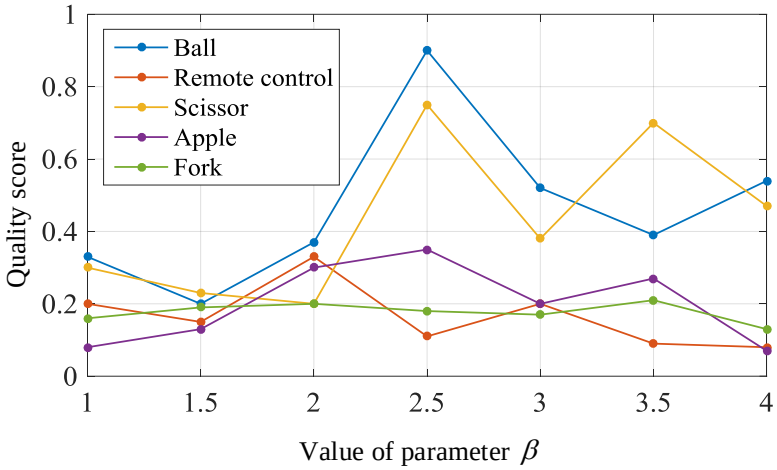


Figure 3.11: The collected maximum quality under different parameter values

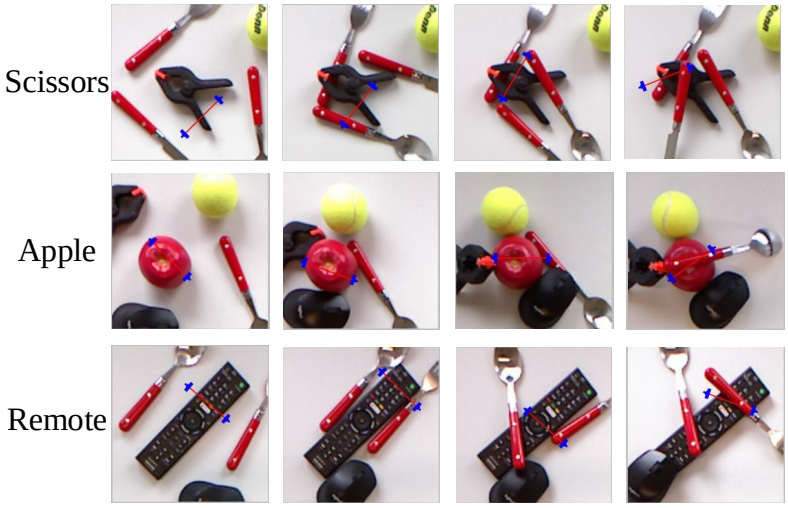


Figure 3.12: Examples of grasp planning in a cluttered environment.

ments. However, in the complex, cluttered environment, the performance of the grasp planning decreases.

The computing time was another ability of the search algorithm. The MC-based search algorithm takes about 3.2 seconds at each time. Bayesian-based search algorithm takes a long time due to the need to solve an optimization problem. A longer search time helped to find a more robust grasp configuration. However, the time required in the Bayesian-based search algorithm could be reduced by enlarging the initial grasping dataset.

Different from the approach introduced in [112, 134] that made a strong assumption that each object contains a single grasp configuration, the proposed approach searches the grasp configuration with the highest quality from the entire grasping space. Moreover, the Probability characteristics of the Bayesian-based search algorithm determines the predicted grasp quality priorities the uncertainty in the prediction. Although the proposed approach successfully finds the feasible grasp, the proposed algorithm still suffered a shortfall that an initial grasp dataset sampled randomly from images was required as a prior. However, the initial grasp dataset usually contained multiple bad grasps that decreased the convergence speed of the search algorithm. The grasp sampling method is important for grasp planning and is the authors' future consideration.

3.4.2 Evaluation of Trajectory Generation

Implementation

To learn the forward dynamic model for the robot, 10 demonstrations of the reaching movements were implemented by the human. As discussed in subsection 3.3.1, the forward dynamic model of the robot was defined by the impedance control Eq. 3.8,

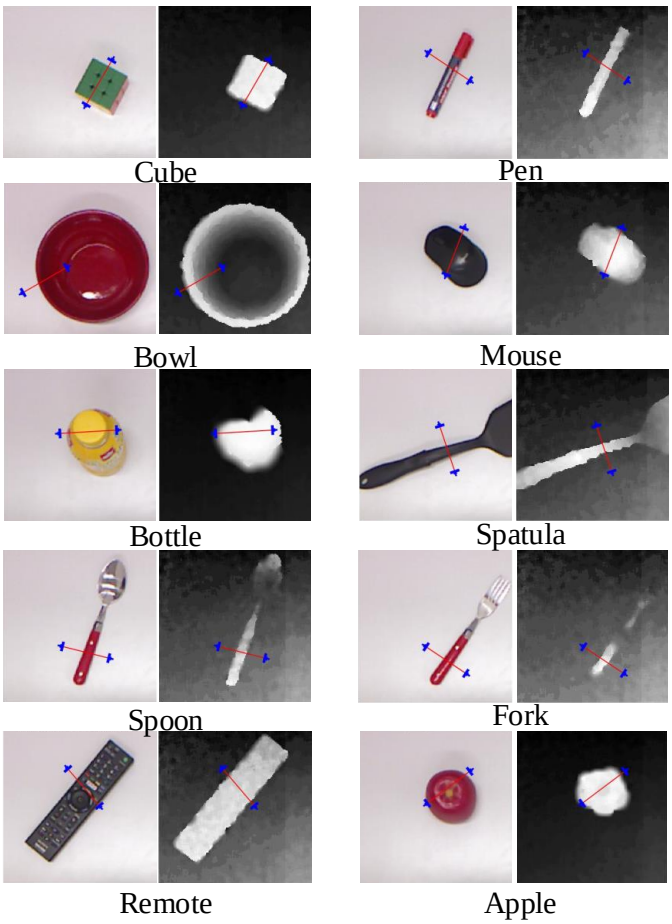


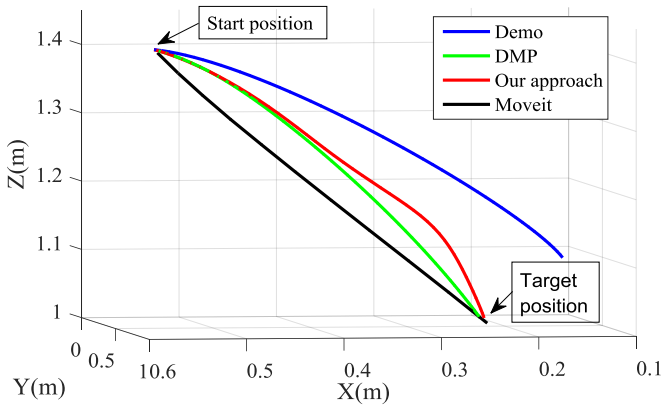
Figure 3.13: Examples of grasp planning. Ten grasp results for ten objects from the proposed object categories were shown. The red line denoted the searched grasp configuration.

where $M = 2I_{3\times3}$, $C = 25I_{3\times3}$ and $K = 156.5I_{3\times3}$. The virtual human control f was approximated by the demonstrated data and the number of Gaussian kernel was selected as $k = 20$.

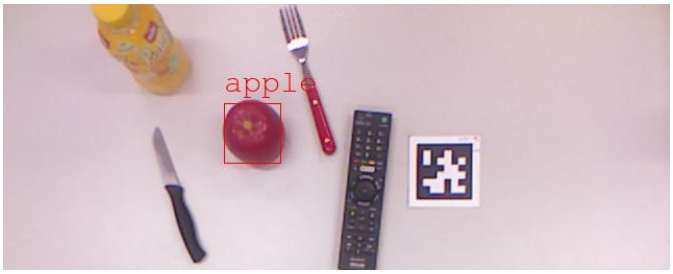
During the trajectory adaptation phase, the inverse model was optimized iteratively by using the MPC method. The control objective was to reach the target position and to avoid a collision with other objects. Hence, the initial weights of the cost function were set as follows: (1) For each waypoint: $Q_1 = 5I_{3\times3}$, $Q_2 = I_{3\times3}$ and $R = 0.5I_{3\times3}$. (2) For obstacles: $Q_1 = 1000I_{3\times3}$, $Q_2 = 0.5I_{3\times3}$ and $R = I_{3\times3}$. (3) For the end-point of the trajectory: $Q_1 = 100I_{3\times3}$, $Q_2 = 10I_{3\times3}$ and $R = 0_{3\times3}$.

Experimental Results and Discussions

The proposed trajectory generation approach was compared two trajectory generation approaches, i.e., the Dynamic Motor Primitives(DMPs) method introduced in [77], and the sampling-based planning method implemented in the MoveIt! software framework [160]. Figure 3.14(b) shows the example scene, where the robot was controlled to



(a)



(b)

Figure 3.14: Comparison of the three different trajectory generation methods. (a) The trajectories produced by the three different methods, (b) The table scene where the robot was controlled to reach the apple. Other objects are considered as obstacles.

pick up the apple on the table. Given the new target pose, i.e., the grasp pose of the apple and the pose of other objects, the trajectories were generated using the three different approaches, as shown in Figure 3.14(a). The blue curve represents the demonstrated trajectory. The green curve represents the trajectory generated by the DMPs method. The DMPs could generalize over different target pose from human demonstrated movements. However, it was unable to avoid the multiple obstacles in the reaching tasks. Here, the trajectory generated by the DMPs collides with the bottle. The black curve denotes the trajectory generated by the sampling-based planning method. This trajectory was searched in a pre-defined collision-free space. Hence, the planning scene was required to be reconstructed before the trajectory planning. In this way, the resulted trajectory was collision-free. However, this method failed easily due to clutter or missing object information. Moreover, this trajectory was less efficient than human demonstrated movement. The red curve represents the trajectory produced by the proposed approach. First, the proposed approach was able to reproduce the natural movement of humans. Similar to the DMPs method, the proposed approach could generalize over different target positions from human demonstrated movements. Different from the DMPs method, the proposed approach considered the constraints of the environment and adapted to the new situation. Second, without constructing a planning scene, the proposed approach effectively avoids multiple obstacles by adding constraints.

Next, we validate the adaptability of the proposed approach. The MPC method is

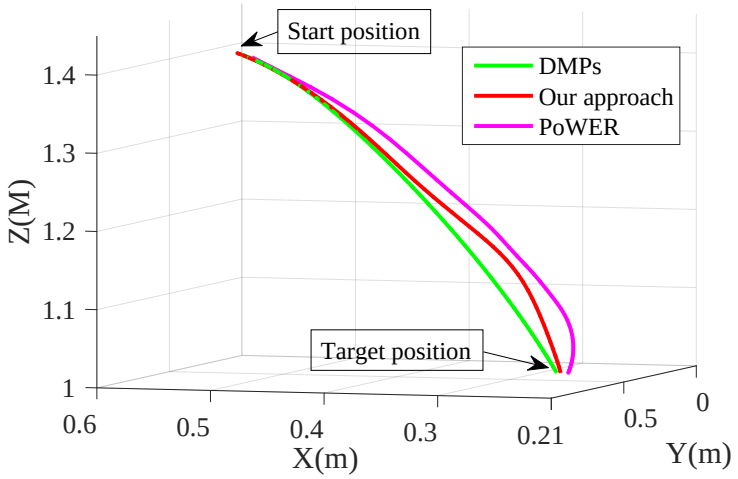


Figure 3.15: Comparison of two different trajectory optimization approaches. The trajectory produced by the DMPs has used a baseline.

employed to optimize the inverse model to adapt to the new environment. For adapting to a new environment, most of learning from human demonstration approaches employ optimization methods, like reinforcement learning, to optimize the parametrized control policy [88]. A comparison between the two different trajectory adaptation approaches was performed. The Policy Learning by Weighting Exploration with the Returns (PoWER) introduced by Kober et al. [88] was used to optimize the DMPs policy learned from human demonstrations. Figure 3.15 shows the two optimized trajectories for the reaching task. It can be seen that the two trajectories reach the target position successfully. However, there were three differences between the two approaches. First, the proposed trajectory adaptation approach required 2 iterations, while the policy search method required 200 iterations to ensure convergence. Typically, the model-free policy search method required a longer computational time than the model-based method. Hence, the proposed model-based trajectory generation method adapted to the new situation. Second, the proposed approach on-line improves trajectory, while the policy search method only optimized one time in an episode. The proposed approach was more similar to the humans who maintain a forward model. Third, the policy-search method optimized all the policy parameters. Hence, its resulted trajectory was quite different from the trajectory generated from human demonstration. While the proposed approach almost maintained the natural movement behavior of humans.

3.4.3 Real-world Robotic Experiment

Implementation

The proposed learning framework was further evaluated in a real-world experiment. Robotic experiments were performed with a 7 DOF Light Weight KUKA robot arm [3]

[3]https://www.kuka.com/de-de/produkte-leistungen/robotersysteme

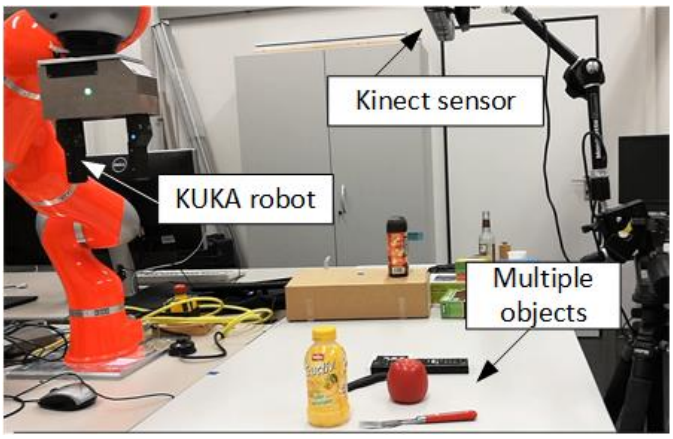


Figure 3.16: Setup of the real-word robotic experiment.

equipped with a WSG 50 parallel-jaw gripper [4], as shown in Figure 3.16. Kinect depth sensor was mounted on the top of the table and used to capture the RGB-D image of the table scene. In the RTG tasks, the target pose was the grasp pose of the object of interest in the robot frame. The poses of the obstacles were computed approximately from the other object on the table. Figure 3.17 shows the coordinates transformation of frames in RTG tasks. Given the dynamic transformation, a grasping pose in the image frame was transformed into a target pose in the robot frame by a sequence of transformation:

$$T_{ro} = T_{rt} T_{ct}^{-1} T_{co} g \tag{3.19}$$

where T_{ro} is the target grasp configuration in the robot frame O_r. T_{rt} is the transformation between the robot frame O_r and the tag frame O_t, which was initially designed by users. T_{ct} is the transformation between the tag frame O_t and the camera frameO_c, which was real-time detected by the Kinect depth sensor. T_{co} is the projection transformation between the 2D image frame and the 3D camera frameO_c. g is a grasp configuration produced by the proposed grasp planning approach.

Experimental Results and Discussions

Figure 3.18 shows three consecutive snapshots which demonstrate the implementation process of RTG tasks. During the implementation of RTG tasks, multiple objects were randomly selected and separately placed in a random position on the table. The human then inputted the object identity hoping to be grasped into the robot. With the captured RGB-D image of objects, the results of object detection, grasp planning, and trajectory generation were obtained sequentially by using the proposed optimization framework. Lastly, the robot was controlled to implement the reaching movement and grasp operation.

To validate the effectiveness of the proposed optimization framework, the above described RTG task was executed three times for each object contained in the TOC

[4]https://www.weiss-robotics.com/en/produkte/gripping-systems/performance-line-en/wsg-50-en/

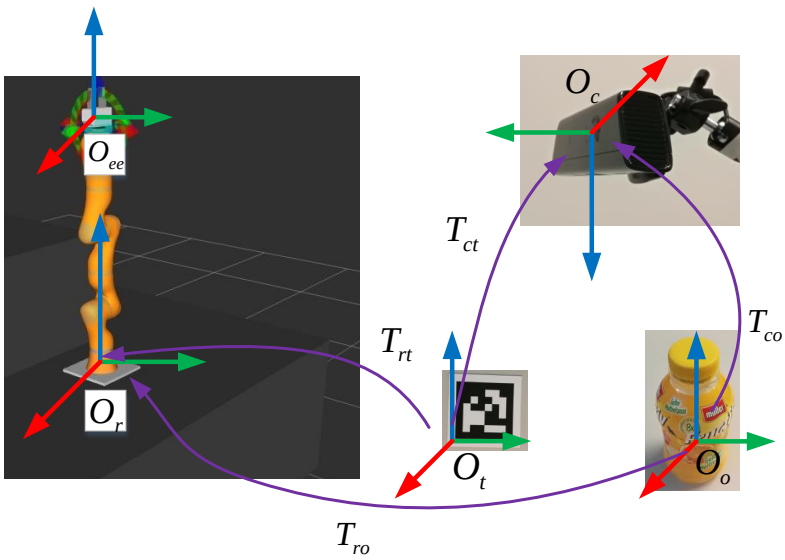


Figure 3.17: Coordinate transformation of the experimental setup in RTG tasks.

Table 3.1: The overall performance of the proposed optimization framework.

Component	Success rate
Object detection	95.8%(46/48)
Robot grasping operation	81.2%(39/48)
Robot reaching movement	100%(48/48)

dataset. At each time, objects were selected and placed at a random position on the table. As a result, 48 RTG tasks were implemented. The experimental results are detailed in Table 3.1. The results show 46 successful object discoveries out of the 48 trials (95.8%). The two failures were due to the recognition error of the object identity. The grasp planning results are 38 successes out of the 48 trials (81.2%). Parts of the failures were due to the infeasible of the detected grasp configuration. The other failure case was due to the object drop out of the gripper during lifting. There were two failures while grasping the bowl, although the detected grasp configuration was feasible. That was because the grasp force was not enough, or the object surface was slippery. The executed trajectories in all 48 trials avoided the obstacles. The experimental results show that the proposed approach can efficiently implement the RTG task in an unstructured environment.

From the experimental results, it can be seen that the proposed approach can find feasible grasp configurations for objects grasping in RTG tasks. The model-based trajectory generation method enables the robot to move to the target subject to constraints successfully. However, the proposed grasp planning approach still suffers a shortcoming. The planned grasp may fail due to the slippage or the irregular shape of the objects, like the bowl. Hence, the grasp closure representing a kind of equilibrium was only a necessary but not sufficient condition for a stable grasp. Bohg et al. [12] pointed out

(a) (b) (c)

Figure 3.18: Three consecutive snapshots during the execution of an RTG task. (a) The object of interest, i.e., the apple, was recognized and localized. (b) A grasp configuration was determined for the apple, and the robot grasps the apple. (c) The robot picks up the apple.

that the stability of grasp can only be defined when considering the grasp as a dynamical system. Hence, it is preferred to employ tactile sensors for grasp adaptation to prevent slippage and increase the stability of the grasps, which will be addressed in future work.

3.5 Conclusion and Future Work

In this work, for stable implementation of RTG tasks, an optimization framework is presented, which combines stable grasp planning with trajectory generation. The proposed optimization framework covers the complete path from perception to decision-making in RTG tasks. By using this framework, the object of interest is first localized and recognized by using a proposed object detection model. The Bayesian-based search algorithm is used to find the grasp configuration with the highest grasp quality computed by a trained network. Then, a model-based trajectory generation approach inspired by human internal model theory is presented to produce robot controls for reaching movements. The effectiveness of the proposed optimization framework is demonstrated and validated in the comparative analysis and on real-world experiments. Results demonstrated that the proposed stable grasp planning method could effectively find the feasible grasp configuration. Moreover, the proposed grasp planning method does not rely on an object mesh model and an accurate pose estimation. The experimental results demonstrate that the proposed optimization framework enables the robots to implement the RTG tasks stably.

Further improvements can be achieved from the following two directions: first, this work only uses the object identified as the semantic information in grasp planning. For a more practical application, it is natural to incorporate the object affordance. Second, determining a possible grasp configuration in multi-finger grasping is more complicated, due to the mathematical complexity of multi-finger gripper. The multi-finger grasp planning requires considering not only the geometry constraints of an object but also kinematics and dynamics of the gripper and contact models between the fingers and the objects. The authors plan to use deep learning techniques to address the multi-fingered grasp planning problem in the future.

Chapter 4

Attention Based Visual Analysis for Fast Grasp Planning

4.1 Introduction

Imagine a toddler is in front of a tabletop with several objects, very likely he or she would interact with those objects by trying to pick up the red mug either by the handle or the rim or trying to grasp the green ball. The ability to rapidly extract relevant information from visual input is an important mechanism and natural behavior for humans to conduct various activities. The majority of visual analysis approaches for grasp planning with multi-fingered robotic hands follow a pipeline containing object localization, recognition, and representation [152]. For most existing approaches, finding a target object in a scene is the first step for robotic grasping. However, reliable object detectors such as deep-learning based approaches require vast amounts of training data, as well as suitable hardware to achieve a reasonable time performance for robotic applications, while handcrafted feature-based approaches cannot handle the dynamics in real-life scenarios.

Many saliency approaches have been proposed in the last two decades. Traditional models are usually based on the feature integration theory (FIT) [163] to compute several handcrafted features which were fused to a saliency map (e.g., the iNVT [79, 174] and the VOCUS system [47]). Frintrop et al. [48] proposed a simple and efficient system which computes multi-scale feature maps using Difference-of-Gaussian (DoG) filters for center-surround contrast and produces a pixel-precise saliency map. Deep learning based saliency detection mostly relies on high-level pre-trained features for object detection tasks. Those learning-based approaches require massive amounts of training data [73, 106, 127]. Kummerer et al. [115] used an AlexNet [91] pre-trained on Imagenet [38] for object recognition tasks. The resulting high-dimensional features are used for fixation prediction and saliency map generation. Since most of the deep-learning based approaches have a central photographer bias which is not desired in robotic applications, we choose to use a handcrafted feature based approach which gathers local visual attributes by combing low-level visual features [48].

Grasp type and grasp attention point convey useful information for planning the con-

figuration of a robotic hand. In the computer vision community, most previous works sample human hand pose with a motion tracking system and use it to detect hand grasp types [145, 17]. In the robotics community, few previous approaches try to integrate grasp type detection into robotic grasp planning [64, 166]. In those works, only two kinds of grasp types, i.e., power and precision [125], are considered, which is not sufficient for exploring the potential of multi-fingered robot hands. Moreover, the desired grasp type is determined manually for robotic hands. In terms of visual analysis, there are approaches which use visual analysis to define heuristics or constraints for grasp planning [69, 3, 166]. In comparison to those approaches, there are three main differences: (1) our approach learns features directly from raw sensor data, while most of the previous approaches use handcrafted features. (2) six grasp types are considered while the previous approaches only consider two grasp types. (3) Most of the previous work only focuses on visual analysis by using computer vision techniques. This work uses the results of the visual analysis for grasp planning with multi-fingered robotic hands. The effectiveness of the proposed framework is evaluated in a real-world object grasping experiment.

Information extracted from the visual analysis can be used to define heuristics or constraints for grasp planning. Previous grasp planning methods can be divided into geometric-based grasping and similarity-based grasping. In geometric-based grasping [69, 96, 166], geometric information of the object is obtained from color or depth images, and it is used to define a set of heuristics to guide grasp planning. Hsiao et al. proposed a heuristic which maps partial shape information of objects to grasp configuration [69]. Aleotti et al. [3] proposed a 3D shape segmentation algorithm which firstly over segments the target object and candidate grasps are chosen based on the shape of the resulted segments [96]. In similarity-based approaches [65, 34, 90], the similarity measure is calculated between the target object and the corresponding object model from human demonstrations or simulation. The candidate grasp is then queried from datasets based on similarity measures. Herzog et al. [65] defined an object shape template as the similarity measure. This template encodes heightmaps of the object observed from various viewpoints. The object properties can also be presented with semantic affordance maps [34] or probability models [90, 92]. Geometric-based approaches usually require a multiple-stage pipeline to gather handcrafted features through visual data analysis. Due to sensor noise, the performance of the geometric-based grasping is often unstable. Meanwhile, similarity-based methods are limited to known objects and can not handle unknown objects. In contrast to previous methods, our method increases grasp stability by extracting more stable features from visual data using deep networks. Meanwhile, it can handle unknown objects.

In this work, we address the problem of visual analysis of natural scenes for grasping by multi-fingered robotic hands. The objective is to compute grasp-relevant information from visual data, which is used to guide grasp planning. We proposes attention based visual analysis framework which directly locates sub-regions of objects as regions of interest (ROIs) and generates grasp-relevant information from visual data inside the ROIs for grasp planning with a multi-fingered robotic hand. In particular, a computational attention model is used to process visual data and outputs a pixel-precise saliency map, from which salient regions are selected for further processing. Inside those salient re-

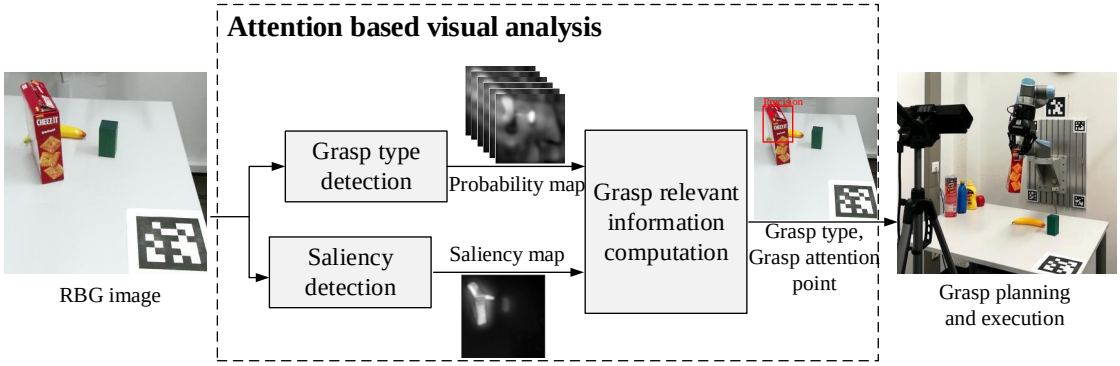


Figure 4.1: The proposed attention based visual analysis framework. With an input RGB image, an ROI is selected using the saliency map produced by a *Saliency detection* model. Inside the ROI, grasp type and grasp attention point are computed based on the six probability maps produced by the *Grasp type detection* network. The obtained information containing grasp type and grasp attention point is then used as a prior to guiding grasp planning. The planned grasp is executed by a robotic hand to verify its quality.

gions, the grasp type and grasp attention point are predicted by a network. The grasp attention point indicates the location on the object surface where the robot plans the grasp. This information is used to guide grasp planning with a multi-fingered robotic hand. A new Grasp Type Dataset (GTD) which considers six commonly used grasp types and contains 12 household objects is also presented.

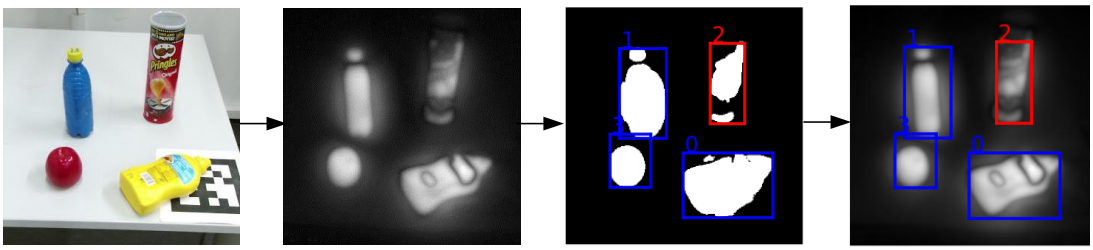
In the rest of this section, Section 4.2 introduces the proposed attention-based visual analysis framework. Grasp planning with grasp-relevant information is described in Section 4.3. Experimental results are presented in Section 4.4. Finally, the conclusion and future work are discussed in Section 4.5.

4.2 Attention Based Visual Analysis

We proposed visual analysis framework contains two main components, a computational visual attention model which gathers low-level visual features and selects ROIs for further processing, and a grasp type detection model which learns higher-level features and produces grasp-relevant information in the ROIs. Figure 4.1 illustrates an overview of the proposed attention-based visual analysis framework.

4.2.1 Computational Visual Attention Model

The pixel-wise saliency map is computed using the computational visual saliency method VOCUS2 [48]. In principle, any saliency system which has a real-time capability and does not have a center-bias could be used. Center bias gives preference to the center of an image, which is not desired in robotics applications. Unfortunately, this excludes most deep-learning based approaches since they are usually trained on large datasets



(a) Input RGB image (b) Pixel-level saliency map (c) Mean shift clustering (d) Output of visual attention

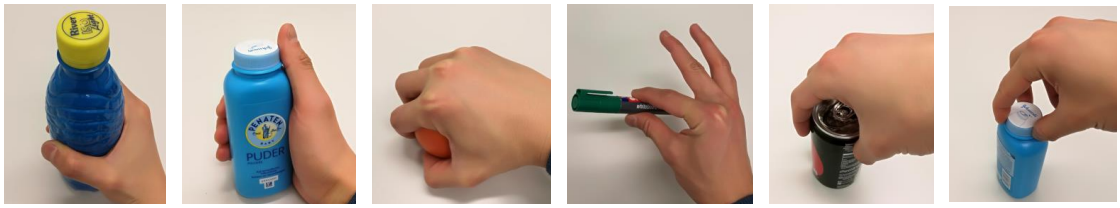
Figure 4.2: Saliency region detection with visual attention model. (a) The input RGB image, (b) the pixel-wise saliency maps, (c) the result after clustering, (d) the output. The red rectangle denotes the selected ROI, which has the highest average saliency value. The blue rectangles denote the candidate ROIs for objects. The numbers are indexing for bounding boxes.

of Internet images, which mostly have a central photographer bias. Therefore, the VOCUS2 system was chosen, which belongs to the traditional saliency systems with excellent performance on several benchmarks. In VOCUS2, an RGB input image is converted into an opponent-color space, including intensity, red-green, and blue-yellow color channels. DoG contrasts are computed with twin pyramids, which consist of two Gaussian pyramids - one for the center and one for the surround of a region - which are subtracted to obtain the DoG contrast. Finally, the contrast maps are fused across multiple scales using the arithmetic means to produce the saliency map.

Given the produced saliency map, the pixels of the saliency map are clustered using Mean Shift [30] to form saliency regions. The salient region, with the highest average salient value, is selected as the ROI, and it is passed to the next stage for further processing. Figure 4.2 shows an example of the saliency region detection. The visual attention model takes the RGB image shown in Figure 4.2(a) as input and produces the saliency map shown in Figure 4.2(b). After clustering, the desired saliency region is determined, as shown in Figure 4.2(c).

4.2.2 Grasp Type Detection

Grasp type is a way of representing how a hand handles objects. Typically, the robotic grasps are divided into power and precision grasp [125]. Power grasp uses the fingers and palm to hold the object firmly, while precision grasp only uses fingertips to stabilize the object. However, this two-category grasp taxonomy is not sufficient to convey information about hand configuration. Feix et al. [46] introduced a GRASP taxonomy in which 33 different grasp types used by humans are presented. All the 33 different grasp types are classified into four groups: prismatic power, circular power, intermediate, prismatic precision, and circular precision. Considering the kinematic limitations of the robotic hand as well as Feix's GRASP taxonomy, we extend the above two-category grasp taxonomy into six commonly used grasp types: *large wrap*, *small wrap*, *power*, *pinch*, *precision* and *tripod*. Figure 4.3 illustrates the proposed grasp taxonomy.



Large wrap | Small wrap | Power | Pinch | Precision | Tripod

Figure 4.3: The proposed six commonly used grasp types.

In order to detect grasp types directly from visual data, we refer to the architecture proposed by [23]. This architecture is based on a deep convolutional neural network (VGG-16) [154]) and uses atrous convolution for signal down sampling. Since an object may support multiple feasible grasp types [46], the grasp type detection is a multi-label detection. Hence, we modify the output layer of the network and do not use the additional fully connected conditional random field. Corresponding to the six grasp types, the modified network predicts six pixel-wise probability maps with the same resolution as the input image. In order to train the modified network for grasp type detection, this work introduces a grasp type detection (GTD) dataset, in which 12 household objects are used, and all the instances are annotated following the proposed six grasp types. The details of the GTD dataset are provided in Section 4.4.1. This work uses a cross-entropy function to define the loss function which is defined as

$$L(\theta) = \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{s \in S} logP(y_{i,j}^{s} | I, \theta) \tag{4.1}$$

where $y_{i,j}^{s} \in \{0,1\}$ indicates if the pixel $y_{i,j}$ belongs to the grasp type $s \in S$ or not. $S = [1,2,\cdots,6]$ is the index of the six grasp types. I denotes an RGB image with height h and width w. θ is the weight of the proposed detection model.In this work, the cross-entropy based on the sigmoid function is defined in Eq. 4.2, where f is the trained network.

$$P(y_{i,j}^{s} | I; \theta) = 1/(1 + exp(-f(y_{i,j}^{s} | I; \theta))) \tag{4.2}$$

Given an RGB image I with height and width $h \times w$ as input, our network outputs pixel-wise probability maps $P(Y|I)$ for each grasp type $s \in S$, where $Y = \{y_{i,j}^{s}\}_{i=1:h, j=1:w}$. The predicted probability of pixel $\{[i,j]_{i=1:h, j=1:w}\}$ belonging to the grasp type s is denoted by $y_{i,j}^{s}$. With the pixel-wise probability maps, the probability $P(Y^{s}|O)$ is computed by summing the predicted probabilities of all the pixels inside the ROI O (defined in Section 4.2.1), as shown in Eq. 4.3. The grasp type with the highest probability is used as the final grasp type s^{*}.

$$P(Y^{s}|O) = \frac{1}{h_O \times w_O} \sum_{i=1}^{h_O} \sum_{j=1}^{w_O} P(y_{i,j}^{s} | x_{i,j}), \forall s \in S. \tag{4.3}$$

After determining the best grasp type s^{*}, we need to localize the grasp attention point for the grasp type s^{*} inside O. In order to find a stable grasp attention point p,

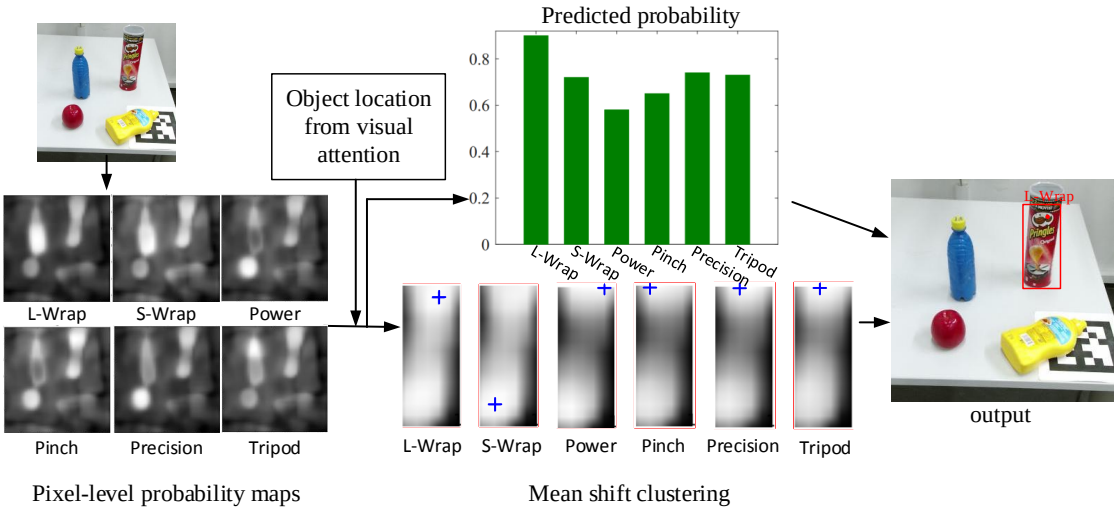


Figure 4.4: The detection process of grasp type and grasp attention point. Six pixel-wise probability maps corresponding to the six grasp types is first computed from the grasp type detection network. Given the object location computed by the visual attention model, these probability maps are clustered. Then the predicted probability of each grasp type and the location of its grasp attention point are computed. Finally, the grasp type with the highest probability and its grasp attention points are determined.

subregions with higher predicted probabilities are clustered. Mean Shift [30] is used to find a grasp attention point p in O. Multiple clusters with multiple centers are produced, and the cluster center with the highest probability is selected as the grasp attention point p. Finally, the grasp relevant information $\Omega = \{O, s^*, p_o\}$, i.e., ROI O, the grasp type s^* and the grasp attention point p_o, are generated from the proposed visual analysis framework. Figure 5.5 illustrates the detection process of grasp type and grasp attention point.

4.3 Grasp Planning with Grasp-relevant Information

The objective of grasp planning is to find the grasp configuration for a stable grasping. Hence, grasp planning in this work is formulated as an optimization problem. A search-based algorithm exploiting grasp-relevant information Ω generated from the proposed visual analysis framework is proposed to find the grasp configuration with high grasp quality. In this work, the search of the feasible grasp configuration is processed from two steps: (1) the formation of the initial grasp configuration based on the grasp-relevant information, (2) the determination of the feasible grasp configuration by the local transformation.

In the first step, we take advantage of the grasp-relevant information $\Omega = \{o, s^*, p_o\}$ to determine the initial grasp configuration and the number of the required finger. The initial grasp configuration of the robotic hand is defined as follows:

1. The number of needed fingers is selected according to the grasp type s^* and the

gripper;

2. The grasp center p_h is set to be a point that deviate a initial offset d_{init} from the 3D grasp attention point p'_o which is obtained from 2D grasp attention point p_0 using frame transformation.

3. The hand palm is controlled to approach the grasp attention point.

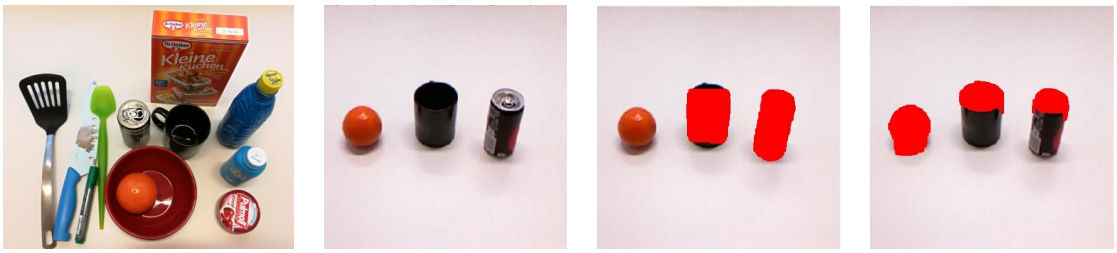
Using a multi-fingered robotic hand to grasp objects typically requires the relative pose between the object and the robotic hand, as well as the hand joint configuration. Due to the high dimensionality of the robotic hand and partially observably of objects, it is challenging to find the optimal contact points on the object surface to form a grasp configuration. In this work, we exploit the concept of *Opposition* introduced by [35] to execute the grasp configuration. The robotic hand is controlled to reach the target pose and close the two-finger groups to grasp an object.

Next, A local search method is used to find the grasp configuration with the highest quality in a grasp search space. Due to the existence of uncertainties, the defined pre-grasp configuration may fail to grasp objects. Hence, a local search is used to find the grasp configuration with higher quality. During searching, the pre-grasp configuration is used as the initial grasp configuration. We sample a set of candidate grasps with coordinate transformation. The search space is a 4 dimensional space, $S = \{d, \alpha, \beta, \gamma\}$, where $d = d_{init} \pm \Delta d$ is the offset of the 3D grasp attention point p'_o. Δd is a pre-defined searching range. $\{\alpha, \beta, \gamma\}$ denote the searching ranges of the rotate angles in the X, Y and Z axes of the hand coordinate respectively. During the search process, all the candidates are evaluated by using force-closure method [159]. The force-closure method has been widely used in grasp planning, which measures the grasp quality through the evaluation of certain geometric relations of the contact points. A grasp is force-closure if a hand can exert arbitrary force on the grasped object through a set of the contact point. After the grasp quality measure, the grasp configuration with the highest quality is chosen for object grasping. Finally, during executing candidate grasps, the fingers move to contact with the object surface and hold it. The robotic arm lifts the object to finish the grasping task.

Algorithm 3 shows the process of the grasp planning procedure.

Algorithm 3 : Attention-based visual analysis for grasp planning

1: **Requires**: a computational saliency model, a grasp type detection model
2: Acquire an RGB image I of the table scene.
3: Visual analysis framework returns the grasp-relevant information $\Omega = \{O, s^*, p_o\}$.
4: Using the information Ω to initialize the pre-grasp configuration of the hand.
5: Using a local search method to find a list of feasible candidate grasps.
6: Find the grasp configuration with the highest quality from all the feasible grasps
7: Execute the grasp operation by using robotic hands.



(a) 12 household objects (b) original image (c) image labeled with *large wrap* (d) image labeled with *precision*

Figure 4.5: Illustration of GTD dataset. (a) Twelve household objects contained in the GTD. (b) The original image. (c) A labeled image with *large wrap*. (d) A labeled image with *precision*. Pixels that belong to a grasp type are marked with color and others are background

4.4 Experiments

4.4.1 Dataset and Implementation

Existing datasets, such as the Yale human grasping datasets [14] and the UT grasp dataset [15], are used for the analysis of human hand behavior. These datasets are not suitable for grasp planning with robotic hands. Hence, we introduce a new grasp type detection (GTD) dataset specified for robot grasping. The GTD dataset contains RGB-D [1] images and ground-truth grasp type labels. There are 11000 annotated images with resolution 640×480. In this dataset, six commonly used grasp types were considered, and 12 household objects with various shape attributes were chosen, as shown in Figure 4.5(a). A MATLAB GUI is designed to annotate grasp types on collected data manually. According to the GRASP taxonomy defined in [46], object parts in images were labeled with different grasp types which enable multi-label detection, as shown in Figure 4.5(b-c). The GTD dataset was split randomly into a training set (90%) and a testing set (10%). The training parameters of the grasp type detection model are set as follows. The initial learning rate was 0.00001, and a step delay policy is used to lower the learning rate as the training progresses. Stochastic gradient descent (SGD) method with a momentum rate of 0.9 is used.

4.4.2 Evaluation of Attention-based Visual Analysis

We first evaluated the accuracy of the grasp type detection on the proposed GTD dataset. For comparison, another network based on the Segnet architecture introduced in [5] is trained and evaluated. Segnet has an encoder-decoder architecture and is widely used for image segmentation. For pixel-wise multi-label detection, we modified the output layer of the Segnet network as introduced in subsection 4.2.2. The same training and testing procedures are used for both networks described in 4.4.1. Table 4.1 shows the

[1] We use only RGB data in this work, and plan to exploit the depth data in the future.

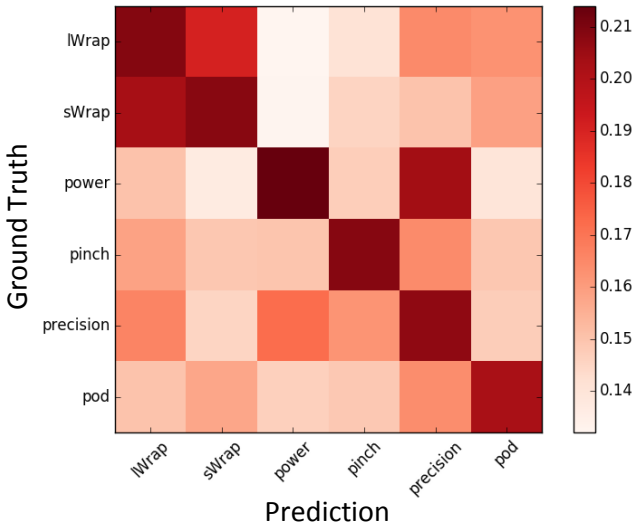


Figure 4.6: The confusion matrix of the six grasp types.

Intersection-over-union (IoU) of the two networks. Our approach achieves a higher average detection accuracy and outperforms the segnet-based network by 10%.

Table 4.1: Performance over GTD dataset (IoU).

	L-wrap	S-wrap	Power	Pinch	Precision	Tripod	Average
Ours	**0.63**	**0.58**	**0.71**	0.56	**0.61**	**0.52**	**0.60**
Segnet-based	0.51	0.56	0.41	**0.61**	0.46	0.48	0.50

A confusion matrix (Figure 4.6) is used to evaluate the overall quality of detected the grasp type. The network predicts six labels corresponding to six grasp types for each pixel. Each row of the matrix shows the predicted probabilities of each grasp type for one ground-truth label. It shows that the proposed method can predict correct grasp types with the highest probability since the diagonal elements have the highest values. It is worth mentioning that several off-diagonal elements also have rather high values. For example, the prediction results for *Power* type also show a high probability for *Precision*, which means those two grasp types are easily mislabeled by the proposed method. The reason is that those two types have a high correlation and share many similar characters. Hence, the confusion matrix can also help to discover the similarity among grasp types.

4.4.3 Grasp Planning in Simulator

The proposed visual analysis framework was further evaluated in object grasping tasks. We implemented a grasping simulation based on the V-REP [2], which is a physical simulator that supports rapid verification, to conduct this experiment. The grasping ex-

[2]http://www.coppeliarobotics.com/

periments were performed on a Shadow Dexterous Hand [3], a five-fingered robotic hand which is an approximation of a human hand. During simulations, the hand configuration and the contact force between the Shadow Dexterous Hand and objects were simulated in real-time, which were used for measuring the qualities of candidate grasps.

In order to evaluate the performance of the visual analysis framework for grasp planning, we compared the proposed planning method with the method proposed by [172]. Veres et al. used a method which randomly samples a set of candidate grasps on the normal of the object surface and then ranked all the candidates to find the best one. Since there is no grasp type provided in this method, we use the commonly used *power* type for the Shadow Dexterous Hand to grasp objects. In this comparison experiment, six objects were selected, as shown in Figure 4.7. Ten trials are tested for each object. For each trial, an object is placed on the tabletop, and a depth sensor is used to capture the RGB-D image of the table scene. Then, the grasp configuration of the Shadow Dexterous Hand is planned in the simulator. The maximum number of search attempts for both methods is limited to forty. For each object, the success rate of object grasping and the average number of search attempts needed for finding a feasible grasp are shown in Table 5.2.

Table 4.2: Performance of the proposed grasp planning.

	Ours		Veres et al. [172]	
object	success rate	search attempt	success rate	search attempt
tomato soup can	8/10	2.5	8/10	20
tuna fish can	9/10	8.7	5/10	23.6
banana	9/10	2.1	5/10	21.6
apple	9/10	2.5	8/10	27.5
orange	8/10	2.8	7/10	19.4
chips can	10/10	2.7	10/10	11.4
Average	88.3%	3.5	71.6%	20.5

It can be seen that the proposed method obtained a higher success rate of grasping than the random search method. Moreover, the number of search attempts by the proposed planning method is only 17.0% of the search attempts by the random search method. It shows that the grasp-relevant information generated helps to reduce the search time needed for grasp planning and to find the feasible grasp configuration in the search space more accurately. It is worth mentioning that the random search method with a *power* type quickly fails at grasping some small objects, such as the banana and the tuna fish can. This limitation does not occur in the proposed planning method since a feasible grasp type is predicted before grasping. Hence, for multi-finger robotic hands, objects with different shape attributes should be handled with different grasp types.

[3] https://www.shadowrobot.com/products/dexterous-hand/

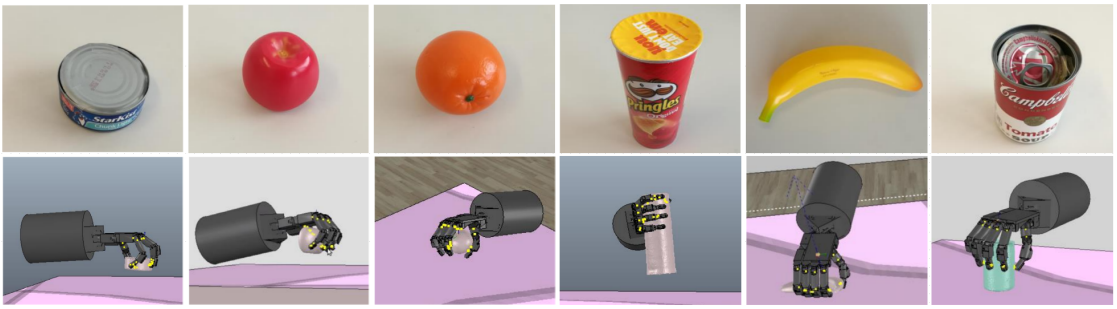
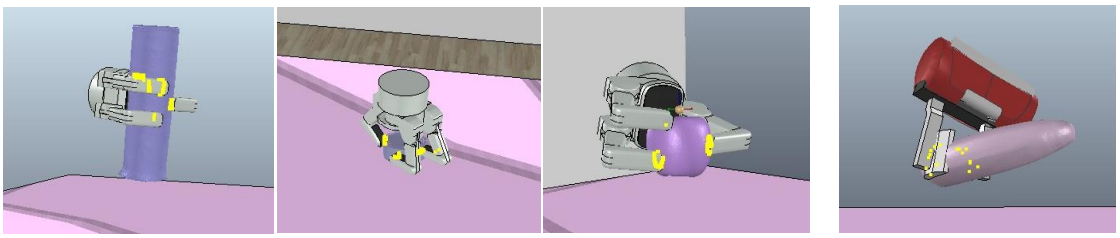


Figure 4.7: Examples of object grasping by the Shadow Dexterous Hand in the simulator.



(a) Barret Hand (b) Baxter gripper

Figure 4.8: Examples of object grasping by the Barrett hand and the Baxter gripper.

We also noticed that there are several failures of object grasps using the proposed planning method. The main reason for the failures is because the predicted grasp attention point on the object surface is too close to the tabletop. Since the environmental constraints are not considered in this work, the Shadow Dexterous Hand will collide with the table and fail to grasp the object. In the future, it will be beneficial also to consider the environment and task constraints.

In order to further evaluate the generalization of the proposed framework, we also tested our framework with a 3-fingered Barrett hand [4] and a 2-fingered Baxter gripper [5], Figure 4.8 shows some results of object grasping. In this experiment, the 2-fingered Baxter gripper only used the *pinch* type to grasp objects. On average, Barrett's hand has 90% success rate with four search attempts while Baxter gripper has 100% success rate with 1.4 search attempts.

To further verify the effectiveness of the grasp planning with prior information, we compared with the work from [29]. This work searches a grasp configuration for dexterous robotic hands in a hand posture subspace, which is determined by using grasp synergies. In their work, the grasp planner only results in a *power* type, which means their grasp planner may fail to grasp small objects. Their planning method needs a long search time for finding a feasible solution, with over 70,000 attempts for each plan, and

[4] https://www.barrett.com/about-barrethand/
[5] https://www.rethinkrobotics.com/baxter/

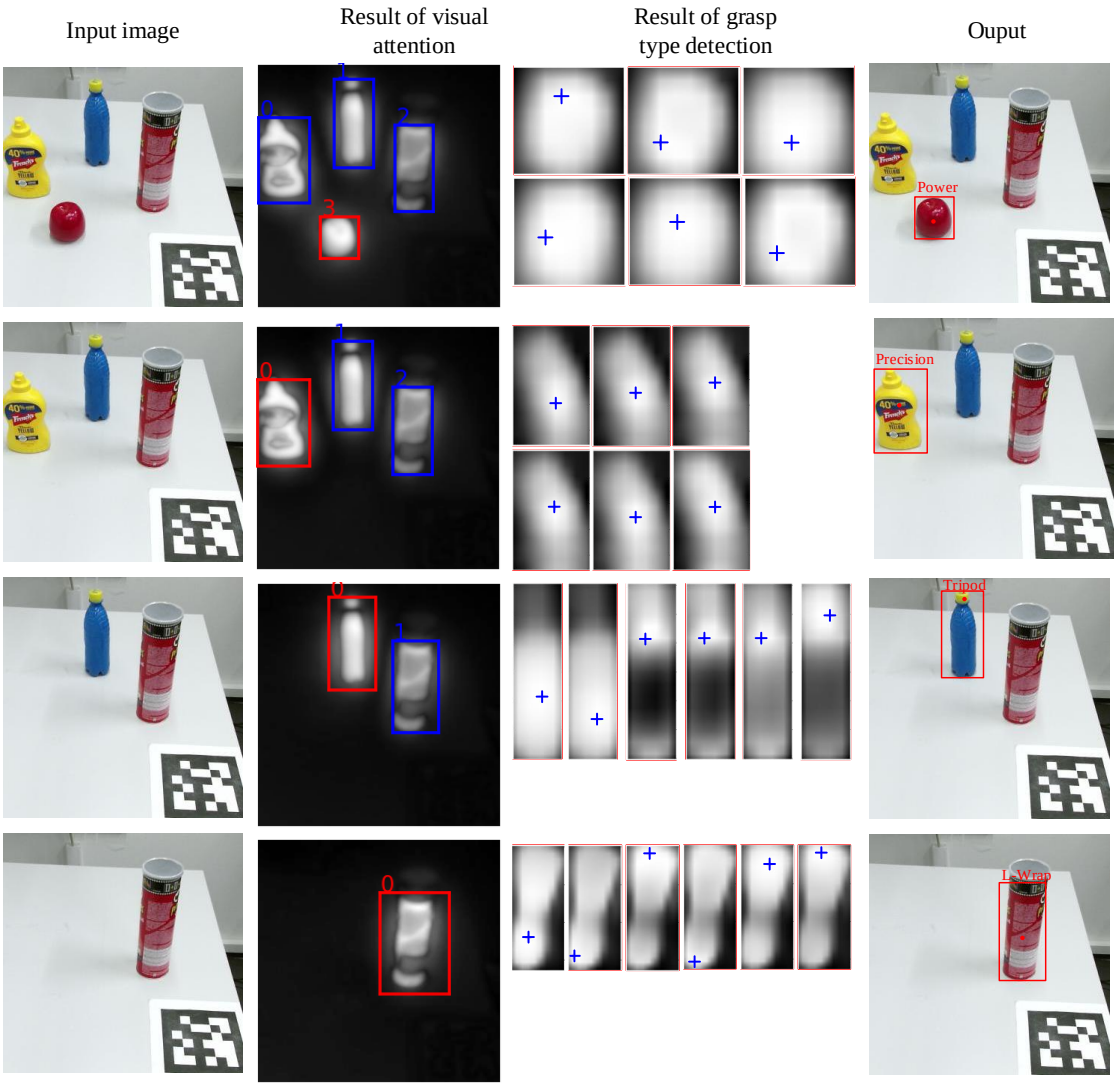


Figure 4.9: Example of the visual analysis on various objects. The first column is the input RGB image. The second column is the pixel-wise saliency map, in which the red rectangle denotes the selected ROI. The third column is six pixel-wise probability maps. The images from top left to bottom right corresponds to the six grasp types(*large wrap*, *small wrap*, *power*, *pinch*, *precision* and *tripod*).The cross in the probability maps denote the cluster centers which is considered as the grasp attention point. Last column is the output.

an average running time of 158 seconds [29]. Compared with their work, our method requires fewer search attempts and enables the robotic hand to grasp objects with different grasp types.

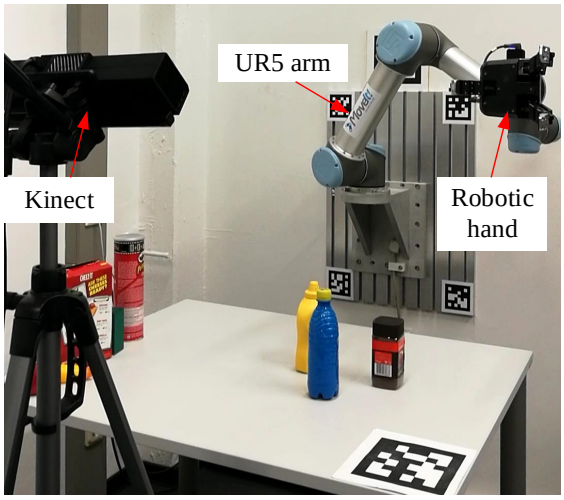


Figure 4.10: Experimental setup with a UR5 arm and a three-fingered robotic hand.

Figure 4.11: Eight different objects for robotic experiments.

4.4.4 Real-world Robotic Experiment

The robotic experiments were conducted using the six DOF UR5 robot [6] and the three-fingered Robotiq gripper [7]. Figure 4.10 shows the experimental setup for the object grasping tasks. A Kinect sensor was used to capture the RGB-D image of the table scenes. Eight objects selected from YCB object set [20] were used for the evaluation, as shown in Figure 4.11. It contains six unknown objects comparing to our dataset (Figure 4.5). In the object grasping experiments, we adopted the following procedure. Multiple objects were randomly selected and placed on the table. The proposed visual analysis framework took the image captured by Kinect as input and outputted the grasp-relevant information. Then, the grasp configuration was planned by taking advantage of this computed information and sent to the UR5 robot for grasping.

Figure 4.9 shows the process of attention based on visual analysis. Given an input RGB image, the ROI denoted by a rectangle in the saliency map is firstly selected by

[6] https://www.universal-robots.com/products/ur5-robot/
[7] https://robotiq.com/products/3-finger-adaptive-robot-gripper

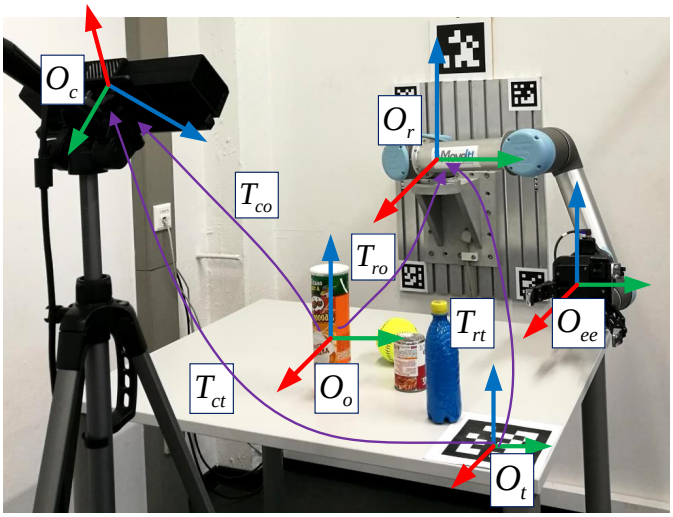


Figure 4.12: The relationship between coordination frame.

the attention model. Meanwhile, six pixel-wise probability maps are obtained from the grasp type detection model. The grasp attention point denoted by the cross in each probability map is obtained by clustering. Finally, the grasp type with the highest probability in the ROI is selected. As it is shown in Figure 4.9, our system is also able to produce grasp type and grasp attention point results on unknown objects.

The performance of the whole system is evaluated based on object grasping tasks. Four trails were tested for each object, and a total of 32 trials were implemented. Because the robotic gripper only had three-finger, we consider *large wrap* and *small wrap* equivalent, and consider *precision* and *tripod* equivalent. So the numbers of the used finger for *precision* and *tripod* were same. The experimental results were that 28 successful grasps out of 32 trails (87.5%). The proposed method enabled the robotic hand to find the feasible grasp configuration and successfully grasp it. Figure 4.13 shows some examples of the object grasping using the proposed framework. As we can see, the grasp-relevant information generated from the proposed framework was used as prior information to guide the grasp formation. For each frame, ROI localization takes 1.8 seconds, grasp type detection takes 6.5 seconds, and the complete process takes 8.5 seconds on average. The proposed framework is implemented in python and runs on a 2.50GHz Intel i5 CPU.

It is worth mentioning that several failures of object grasping have occurred. As in simulation experiments, when grasping the small object (e.g., apple), the planned grasp pose was too close to the table, the UR5 robot failed to find a feasible kinematic solution. Another cause was that the proposed visual attention method sometimes only locate a small region of an object and a possible grasp configuration cannot be found. This is caused by low color contrast between the object and its background. It also occurred that the object fell out of the gripper during lifting. The uncertainty from the object weight caused it. In the future, it will also be beneficial to incorporate grasp adaptation into the proposed framework.

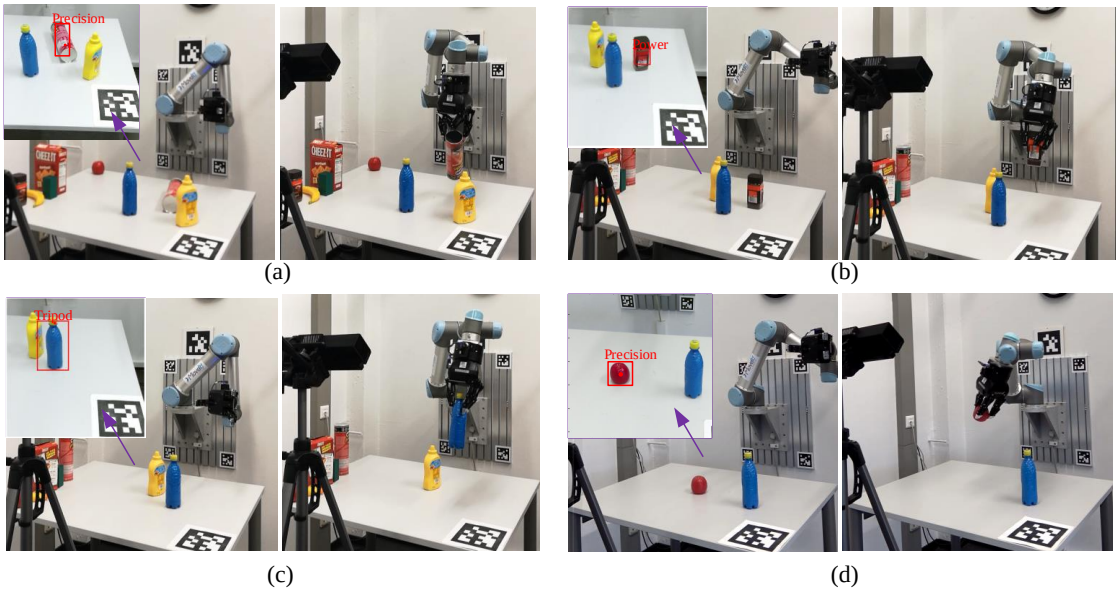


Figure 4.13: Examples of object grasping using the UR5 robot. In each sub-figure, the left showed the analyzed results and the right showed the robot grasped the object.

4.5 Conclusion and Future Work

This work proposes an attention-based visual analysis framework, which computes grasp-relevant information directly from visual data for multi-fingered robotic grasping. By using the visual framework, an ROI is firstly localized by a computational attention model. The grasp type and grasp attention point on object segment presented in the ROI is then computed using a grasp type detection model, which is used as prior information to guide grasp planning. We demonstrated that the proposed method could give the right prediction of grasp type and grasp attention point. Furthermore, the performance of the proposed visual analysis framework has been evaluated in object grasping tasks. Compared to previous methods without prior, the information generated from the visual analysis can significantly speed up grasp planning. Moreover, by using a feasible grasp type, the success rate of the grasping is also improved. Results show that the proposed framework helps the robotic systems to know how and where to grasp objects according to attributes of sub-regions of objects. Since our method does not rely on object detection, it can also handle unknown objects.

For future work, several aspects will be considered. First, the current framework is goal-driven, and it only learns how to grasp an object, so it will be interesting to extend the proposed framework into a task-driven framework (e.g., grasping in human-robot handover task). Second, currently, the choice of grasp type and grasp attention point only depends on the attributes of sub-regions of objects. Since grasp planning is also affected by environment and task constraints, those constraints will be taken into consideration.

Chapter 5

General-purpose Grasp Planning with Pregrasping Opposition

5.1 Introduction

Autonomous grasping is one of the most fundamental abilities required for a robot to handle manipulation tasks. Many grasp planning approaches, such as analysis-based [61, 43] and data-driven approaches [12], have been proposed to allow robotic hands to achieve some specific grasping tasks. Most of the previous methods are designed for one specific robotic hand. Few works address the general-purpose problem of grasping planning where the designed grasp planning methods should be suitable for various hands with an arbitrary number of fingers. Moreover, previous grasp planning methods only allow robotic hands to execute two grasp types (i.e., *power* and *precision* grasps), which were introduced by Napier et al. [125]. It is a natural behavior for humans to choose a feasible grasp type from multiple grasp types to grasp objects. The grasp type could guide a robot on how to operate an object. Grasp type is still an underexplored concept in robotic grasp planning. The generality of grasp planning methods has not been addressed well. This work proposes a novel general-purpose grasp planning approach that allows various robotic hands with an arbitrary number of fingers to stably grasp objects.

Grasp type conveys useful information for planning grasp configurations of a robotic hand. However, previous grasp planning methods [142, 55, 63] mainly choose grasp types manually for object grasping. Only two kinds of grasp types (i.e., *power* and *precision* grasps) are considered. These two grasp types are not sufficient for exploring the potential of a multi-fingered robotic hand. Previous work [145, 17] on grasp type detection mainly focuses on the analysis of human grasping behavior rather than robotic grasping application. Hence, in this work, six commonly used grasp types extracted from the grasp taxonomy [46] are considered in grasp planning. The grasp-relevant information is computed based on the detected grasp type, which is further used as guidance for grasp planning. There are also some approaches that use visual analysis to define heuristics as guidance for grasp planning [69, 3, 166]. In comparison to those approaches, there are two main differences: (1) Six commonly used grasp types

are considered in the proposed method, whereas the previous approaches only consider two grasp types. (2) The proposed approaches extract the grasp-relevant information directly from raw sensory data, whereas most of the previous approaches use handcrafted features to estimate the grasp-relevant information.

Understanding grasp formation is important to design effective grasp planning approaches. Different concepts have been introduced to analyze human grasping behavior. They include *opposition* [75, 35], VFs [74], *caging* [144] and grasp type [125, 46]. These concepts convey some important information about grasp formation. For example, the *opposition* concept suggested that a prehensile posture can be interpreted as an opposition between hand-parts. Meanwhile, the hand-part could be understood as a VF. Based on the *opposition* concept, Iberall et al. [76] introduced three opposition primitives (i.e., *pad opposition*, *palm opposition* and *side opposition*). Souza et al. [35] extended the definition of opposition primitives in [76] and referred to the choice of hand-parts during grasp planning as the grasp intention. In this way, a complex grasp configuration is understood as a simple *opposition* primitive. These concepts have been widely used for the analysis of human grasping behavior. However, little previous work has incorporated these concepts into grasp planning. In this work, the *opposition* configuration is taken as a starting grasp configuration that is further used to form more complex grasp configurations. Specifically, this work introduces a new concept called *pregrasping opposition* for grasp planning by exploiting two concepts (i.e., the *opposition* and the grasp type). The *pregrasping opposition* provides a way to generalize the grasp planning approach across various robotic hands with an arbitrary number of fingers.

One challenge in grasp planning is to find a set of feasible contact points on an object surface for stable object grasping. Classical analysis-based methods [61, 43, 64] typically formulated grasp planning as a constrained optimization problem. Owing to the high dimensionality of robotic hands and the non-linearity of the constraints, the computation of the optimization problem was intractable. Some works used data-driven methods [12, 34, 90] to learn feasible grasp configurations from prior experience. The performance of data-driven methods largely relied on a huge number of grasp examples. Moreover, these grasp planning methods were limited to handling known objects using a specific hand. They failed to generalize to new robotic hands and unseen objects. In this work, a dual-stage grasp planning is designed to suit various robotic hands with an arbitrary number of fingers. The use of *pregrasping opposition* helps to reduce the computational complexity of grasp planning.

The objective of this work is to achieve general-purpose robotic grasping in a real-world environment. We present a general-purpose grasp planning framework that allows various robotic hands with an arbitrary number of fingers to grasp objects by considering the use of grasp types. The novel concept of *pregrasping opposition* is introduced to encode grasp-relevant information (i.e., grasp type and grasp attention point). A dual-stage grasp planning method beginning with *pregrasping opposition* is designed to find feasible grasp configurations involving a set of contact points. The *pregrasping opposition* is taken as a waypoint for the formation of feasible grasp configurations. The performance of the proposed grasp planning methods is evaluated in both simulations and real-world experiments.

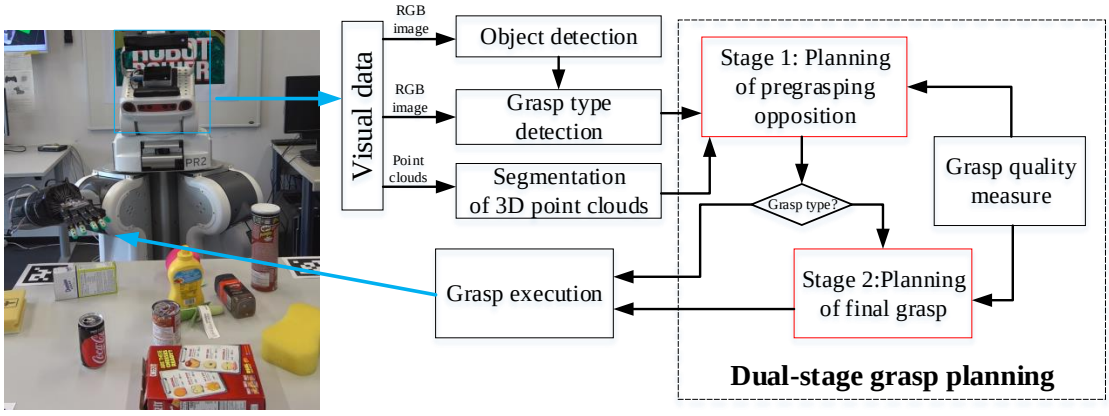


Figure 5.1: Overview of the proposed general-purpose grasp planning framework. Firstly, a depth sensor is used to capture visual data (i.e., the RGB image and point clouds) of the table scene. Then, the grasp-relevant information is computed from visual data. A *pregrasping opposition* is then planned based on the computed information, which used as a waypoint for the grasp formation. The final grasp configuration is optimized based on the *pregrasping opposition*. At final, the robotic hand executes the planned grasp configuration.

.

The rest of this section is organized as follows. An overview of the proposed grasp planning framework is introduced in Section 5.2. Section 5.3 introduce the representation of the *pregrasping opposition* and Section 5.4 present the dual-stage grasp planning method. Experiments are presented in Section 5.5. Finally, the conclusion and future works are discussed in Section 5.7.

5.2 Overview

In this section, we give an overview of the proposed grasp planning framework that contains two main components: the representation of *pregrasping opposition* and a dual-stage grasp planning method. *Pregrasping opposition* is introduced to encode the grasp-relevant information (i.e., grasp type, grasp attention point, and object location) computed from visual data. The dual-stage grasp planning method is used to determine a feasible grasp configuration for object grasping. *Pregrasping opposition* is taken as a waypoint for grasp formation. Figure 7.4 shows an overview of the proposed multi-fingered grasp planning framework.

5.3 Definition of Pregrasping Opposition

This section introduces the novel concept of *pregrasping opposition* to encode the grasp-relevant information extracted from visual data.

5.3.1 Information Extraction Based on Grasp Type Detection

This subsection introduces the extraction of grasp-relevant information (including the grasp type, the grasp attention point, and object class) for object grasping. Grasp type is a way of representing how a hand handles objects. Grasp attention point is a location on the object surface that a hand hopes to approach. The grasp-relevant information can be used as guidance for grasp planning. Typically, the robotic hands are planned with two types (i.e., the power and precision type) to grasp objects. Power grasp uses the fingers and palm to hold the object firmly, and precision grasp only uses fingertips to stabilize the object. However, this two-categories grasp taxonomy is not sufficient to convey information about hand configuration, especially for multi-fingered robotic hands. Six commonly used grasp types extracted from Feix's GRASP taxonomy [46] are considered in this work. They are *large wrap*, *small wrap*, *power*, *pinch*, *precision* and *tripod*. The detail of the grasp type detection method has been presented in Chapter 4. Different from our previous work, this work combines grasp type detection with a learning-based object detection method instead of saliency detection. The main computation of grasp-relevant information extraction is introduced as follows.

Given an input RGB image, the object region O denoted by a rectangle and the object class are first determined. In this work, YOLO-2, a state-of-the-art object detection method, is used to find the object region O. Then, a grasp type detection model trained by using a Convolutional Neural Network(CNN) takes an RGB image as an input and output the probability for each grasp type in pixel-wise. Corresponding to the six grasp types, the detection model predicts six pixel-wise probability maps with the same resolution as the input image. Given the predicted probability maps, the grasp-relevant information is computed in two steps: (1) the predicted probabilities of all the pixels in the object region O are summed. The predicted probability $P(s|O)$of each grasp type s is obtained. The most appropriate grasp type s^* is chosen according to the highest predicted probability, i.e., $s^* = \text{argmax}\, P(s|O), \forall s \in [1,2,...,6]$. (2) After determining the best grasp type s^*, we can localize a grasp attention point p_{att} for the grasp type s^* inside the object region O. The Mean shift algorithm is applied to the probability maps. The cluster center with the highest probability is selected as the grasp attention point p_{att}. Finally, the grasp relevant information $\Omega = \{O, s^*, p_{att}\}$, i.e., the object region O, the grasp type s^* and the grasp attention point p_{att}, are computed, which are used as guidance for guiding the grasp planning. Figure 7.1 shows some examples of the extraction of the grasp-relevant information.

5.3.2 Pregrasping Opposition

The introduction of *pregrasping opposition* drives from the *opposition* concept introduced in [35] and the grasp taxonomy presented by Feix et al. [46]. According to the *opposition* concept, a final grasp posture of a robotic hand is viewed as an opposition between two hand-parts. Hence, this work makes a hypothesis that a complex grasp configuration can generate from a simple opposition configuration. This work introduces the concept of *pregrasping opposition* to represent this opposition configuration. Moreover, the *pregrasping opposition* is taken as a waypoint for grasp formation. Dif-

Figure 5.2: Examples of grasp-relevant information extraction. The red bounding box denotes object locations. The text denotes grasp types (here, the lWrap denotes the *large-wrap*). The red cross denotes the location of the grasp attention point.

ferent from the opposition primitives introduced in [76, 35], the introduced *pregrasping opposition* consider not only the opposing hand-parts and the opposing points, but also the requirement of grasp types. The concept of *pregrasping opposition* is defined as follows.

Definition 1. *(pregrasping opposition) is an opposition configuration of robotic hands, in which a pair of opposing hand-parts exerts a pair of opposing force on the grasped object within a hand-centric coordinate frame to realize the desired grasp type.*

According to the above definition, this work introduces six different *pregrasping opposition* specified for six different grasp types, which are defined as follows

$$\begin{aligned} O^{s}_{h1-h2}(C_o) = \{&O^{1}_{TS-FS}(C_o), O^{2}_{P-FS}(C_o), O^{3}_{TS-FS}(C_o), \\ &O^{4}_{TT-FT}(C_o), O^{5}_{TT-FT}(C_o), O^{6}_{TT-FT}(C_o)\} \end{aligned} \tag{5.1}$$

where $O^{s}_{h1-h2}(C_o)$ denotes a *pregrasping opposition* that determines an opposition between the hand-parts $h1$ and $h2$ under the grasp type s. Each opposing hand-part is abbreviated by its starting letter (e.g., Thumb Surface = TS). The $C_o = (oc_1, oc_2)$ denotes a pair of opposing points. This work takes the regional center of hand-parts as the opposing point. In addition, the grasp type $s = \{1, 2, ..., 6\}$ represents the *large-wrap*, *small-wrap*, *power*, *pinch*, *precision* and *tripod*, respectively. Figure 5.3 shows the six *pregrasping opposition* specified for six different grasp types. From the figure, it can be seen that the opposing hand-parts in the *pregrasping opposition* is a collection of fingers or hand surface.

The construction of the *pregrasping opposition* requires to determine two ingredients, i.e., the grasp type s and a pair of opposing points C_o on an object surface. The desired grasp type s is computed with the proposed information extraction method introduced in Section 5.3.1. Given the grasp type s, the hand-parts for object grasping are specified. The next problem is to compute the C_o from point clouds of an object. Section 5.4 presents the detail of the computation of the grasp configuration. Once the desired grasp type and the C_o are determined, the *pregrasping opposition* is fully constructed, which is further used to guide the formation of the final grasp configuration.

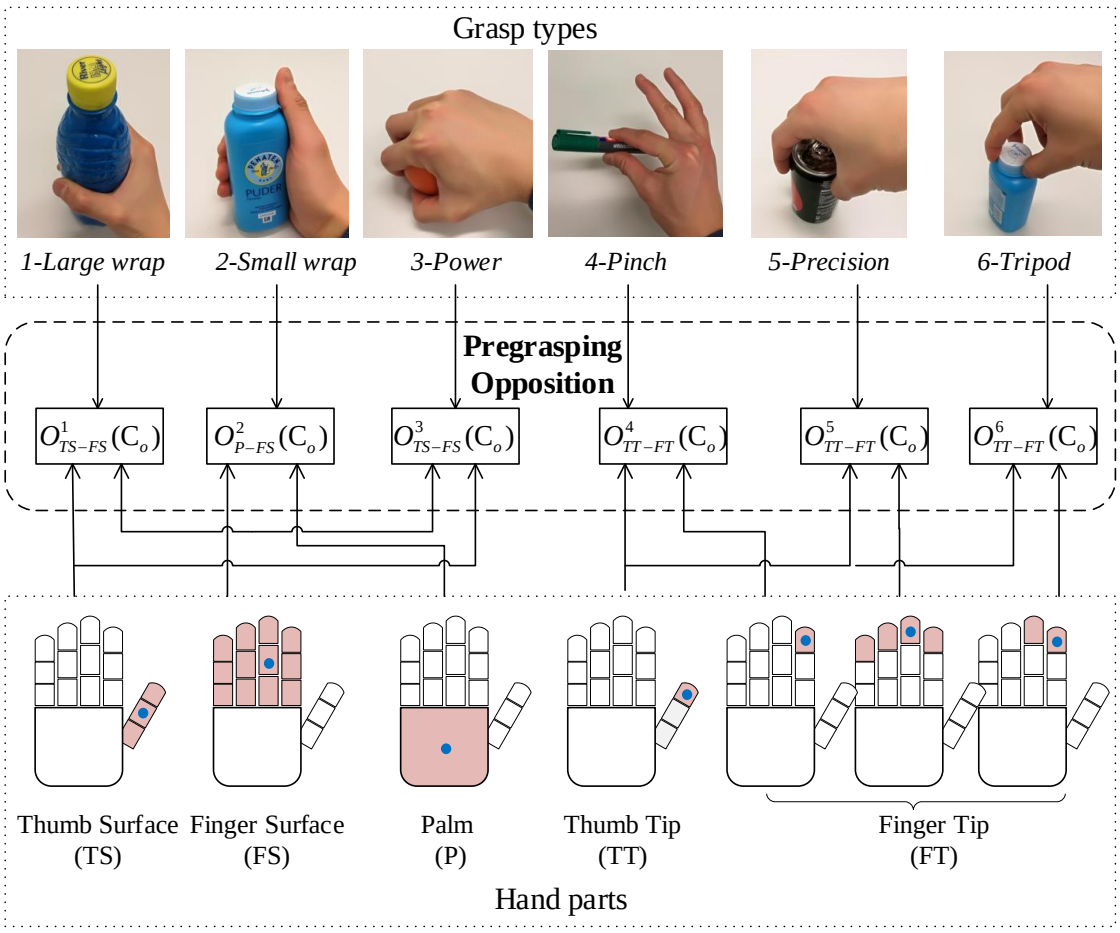


Figure 5.3: The top figure shows six different grasp types and the corresponding human grasp examples. The bottom figure shows the hand parts, which includes five categories. The blue dot at the regional center of the hand-part denotes the opposing points. The middle figure shows the *pregrasping opposition*. The O^4_{FT-TT}, O^5_{FT-TT}, and O^6_{FT-TT} have the same hand-part, but the number of used fingers is different due to the different grasp type.

5.4 From Pregrasping Opposition to Grasping

This section presents a dual-stage grasp planning method beginning with the *pregrasping opposition*. The proposed method finds feasible grasp configurations involving a set of contacts for a robotic hand. The *pregrasping opposition* is taken as a waypoint for grasp formation. The computation of the final feasible grasp configuration for object grasping is achieved from the following two stages.

- At stage 1, we construct the *pregrasping opposition* $O^s_{h1-h2}(C_o)$ by determining a pair of opposing points $C_o = \{c^1_o, c^2_o\}$ on an object surface. The computation of the $O^s_{h1-h2}(C_o)$ is guided by the desired grasp type s and the grasp attention point p_{att}.
- At stage 2, we plan the final grasp configuration $g = (C_g, Joint_g, Pose_g)$ by search-

ing a set of contact points $C_g = \{c_g^1, ..., c_g^m\}$ in a bounded searching space. The *pregrasping opposition* $O^s_{h1-h2}(C_o)$ serves as the starting configuration for grasp planning at stage 2.

The objectives of the two planning stages are to find a configuration $\hat{g}$ that maximizes a grasp quality. Specifically, the $\hat{g}$ is a *pregrasping opposition* $O^s_{h1-h2}(C_o)$ at stage 1 and is a final grasp configuration g at stage 2. Grasp quality measure in this work includes three components: grasp stability measure $S(\hat{g})$, grasp reachability measure $R(\hat{g})$ and grasp attention measure $G(\hat{g})$. The maximizing of the composition quality is a multi-objective optimization problem. This work use the ε-constraint method to scalarize the multi-objective optimization problem.

$$\begin{aligned} \max_{\hat{g}} \quad & \omega_R R(\hat{g}) + \omega_G G(\hat{g}) \\ \text{s.t.} \quad & S(\hat{g}) \geq \varepsilon \\ & \hat{g} \in \Phi_{\hat{g}} \end{aligned} \tag{5.2}$$

where the grasp objective is denoted as $Q = \omega_R R(\hat{g}) + \omega_G G(\hat{g})$. ω_R and ω_G are the weight of the $R(\hat{g})$ and $G(\hat{g})$ respectively. ε is a lower bounds for the grasp stability. $\Phi_{\hat{g}}$ is the solution space of the configuration $\hat{g}$.

5.4.1 Grasp Quality Measure

Grasp quality measure is to evaluate the performance of candidate grasps. This subsection presents the choice of the grasp quality measure. In this work, grasp quality measure includes three components: grasp stability measure $S(\hat{g})$, grasp reachability measure $R(\hat{g})$ and grasp attention measure $G(\hat{g})$

- Grasp stability measure $S(\hat{g})$: The grasp stability is usually analyzed based on force-closure methods [86, 159]. A grasp is force-closure if a hand can exert arbitrary force and moment on the grasped object through a set of contacts. This work applies the Grasp Isotropic Index (GII) reported in [86] to evaluate the grasp stability, which is defined in Eq. 5.3. The GII tries to obtain a uniform contribution of contact forces to the total wrench exerted on the grasped object.

$$S(\hat{g}) = \frac{\beta_{min}(G)}{\beta_{max}(G)} \tag{5.3}$$

  where $\beta_{min}(G)$ and $\beta_{max}(G)$ are the minimum and maximum singular value of grasp matrix G, respectively.

- Grasp reachability measure $R(\hat{g})$: Grasp reachability measure mainly considers the kinematic constraint of a robotic hand. It is time-consuming to computing an inverse kinematic solution of the robotic hand in each optimization step. Most of the previous work [55, 159] defined some heuristics to measure the grasp reachability. These previous methods adopted a hypothesis that the bigger the area covered by a grasp polygon is, the better the grasp will be. However, these methods

ignored a fact that the higher the height from the centroid of the grasp polygon to the hand palm is, the smaller the available workspace of the robotic hand is. This work proposes a new reachability measure $R(\hat{g})$ that takes into account both the area covered by the grasp polygon and the height from the centroid of the grasp polygon to the hand palm. The reachability measure $R(\hat{g})$ is defined in Eq. 5.4. The $R(g)$ tries to drive robotic hands to maximize the feasible area covered by the grasp polygon.

$$R(\hat{g}) = \begin{cases} cos(K(a - 0.5a_{max}))e^{-2h} & \text{if } h < h_{max} \text{ or } a < a_{max} \\ cos(-0.5K \cdot a_{max})e^{-2h_{max}} & \text{Otherwise} \end{cases} \tag{5.4}$$

where a is the area covered by a grasp polygon. When only two contact points are required for a grasp, a is a line length formed by the two contact points. h denotes the height from the centroid of a grasp polygon to a hand palm. a_{max} and h_{max} are the maximum area and the maximum height determined by robotic hands. K is a weight.

- Grasp attention measure $G(\hat{g})$: When a robotic hand grasps an object, we expect that the approaching direction of a robotic hand is preferred to be consistent with the normal direction n_{att} of object surface at the p_{att}. The p_{att} defines the desired approaching location of robotic hands. This work measures the grasp attention quality to filter all the candidate grasps. Grasp attention measure $G(\hat{g})$ is computed based on the relationship between the p_{att} and the pair of the opposition points (oc_1, oc_2), as defined in Eq. 5.5. The $G(\hat{g})$ is to encourage the line $oc_1 - oc_2$ perpendicular to the normal direction n_{att} of object surface at the p_{att}.

$$G(\hat{g}) = -\left| arccos(\frac{n_{att} \cdot n_{oc_1 - oc_2}}{|n_{att}| \cdot |n_{oc_1 - oc_2}|}) - \frac{\pi}{2} \right| \tag{5.5}$$

where n_{att} is the the normal direction of an object surface at the grasp attention point p_{att}. The range of the $G(\hat{g})$ is set to $[-\frac{\pi}{2}, 0]$. Figure 5.4 shows a example of the grasp attention measure under the *tripod* grasp. The $G(\hat{g})$ tries to drive robotic hands to grasp objects with a constrained approaching motion. In this figure, the two contact points (c_2, c_3) is searched around the point oc_2. These two points then cooperate with the point oc_1 to form the final set of the contact points.

5.4.2 Grasp Configuration Optimization

As defined in Eq. 5.2, given point clouds P of an object, we aim at finding a set of contact points $\hat{C}$ to form a grasp configuration $\hat{g}$ that maximize the objective Q. The $\hat{C}$ is a pair of opposing points $C_o = \{c_o^1, c_o^2\}$ at stage 1 and is the final contact points C_g at stage 2. This work applies a simulated annealing algorithm [29] to solve the optimization problem defined in Eq. 5.2.

Different from previous simulated annealing algorithms [29] which start from a random initial state, this work exploits grasp-relevant information to define some reference

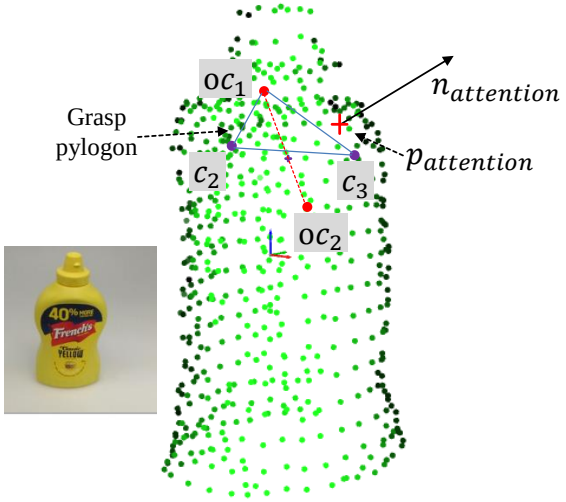


Figure 5.4: Illustration of grasp attention measure under the *tripod* grasp.

points p_{ref} which are used to guide the search of configurations. In this work, the p_{ref} is the grasp attention point p_{att} at stage 1 and is the pair of opposing points C_o at stage 2. The neighborhood $\Phi_{p_{ref}}$ of the p_{ref} is a set of points closed to the p_{ref} as defined in Eq. 5.6. During optimization, this algorithm searches candidate grasp points in the neighborhood $\Phi_{p_{ref}}$ as the stages for configuration optimization.

$$\begin{aligned} \Phi_{p_{ref}} &= \{p_i \in P \mid\mid\mid p_i - p_{ref} \mid\mid < \lambda\} \\ \text{with} \quad p_{ref} &= \begin{cases} p_{att} & \text{at stage 1} \\ C_o & \text{at stage 2} \end{cases} \end{aligned} \tag{5.6}$$

where p_{ref} denotes a reference point. p_i is a candidate contact point. P is the point clouds of the target object. λ is a threshold which controls the size of the neighborhood. The use of p_{att} and C_o in this algorithm helps to reject infeasible contact points and reduce the sampling complexity.

Given a new configuration $\hat{g}_{new}$ generated by the neighbor-generating function, this algorithm uses criteria defined in Eq. 5.7 to accept or reject the $\hat{g}_{new}$. It can be seen that this algorithm accepts all new configurations that raise the grasp objective Q and still accepts the $\hat{g}_{new}$ with a certain probability which lower the Q. By accepting points that lower the Q, this algorithm can explore globally for more feasible grasp configurations.

$$P(\hat{g}_{new}) = \begin{cases} 1 & \text{if } Q(\hat{g}_{new}) > Q\prime \\ e^{-\frac{Q\prime - Q(\hat{g}_{new})}{t}} & \text{otherwise} \end{cases} \tag{5.7}$$

where $P(\hat{g}_{new})$ is the acceptance probability of the $\hat{g}_{new}$. $Q(\hat{g}_{new})$ is the grasp objective obtained by the $\hat{g}_{new}$ and $Q\prime$ denotes the current highest sampled objective. t is the current temperature of the simulated annealing.

In the proposed two-stage grasp planning method, at stage 1, the p_{att} is used to guide

the configuration search. The optimized result C_o of this stage is used to construct the *pregrasping opposition* $O^n_{h1-h2}(C_o)$. At stage 2, the $O^n_{h1-h2}(C_o)$ is used to guide the solution search. The optimized result C_g is used to further form the final grasp configuration $g = (C_g, Joint_g, Pose_g)$. In the proposed grasp planning method, because the *pregrasping opposition* is taken as a waypoint for grasp formation, the computation of complex grasp configuration is reduced. Moreover, the proposed grasp planning method is general purpose because it can determine feasible contact points for various robotic hands. The pseudocode of the proposed grasp planning framework is shown in Algorithm 4.

Algorithm 4 : General-purpose grasp planning with pregrasping opposition.

1: **Input**: a grasp type detection model,
2: **Output**: grasp configuration g
3: Acquire an RGB image I and point clouds P of a table scene.
4: Performance grasp type detection to obtain six probability maps $P(Y^s|I)$.
5: Extracts the grasp-relevant information $\Omega = \{O, s^*, p_{att}\}$.
6: Plan the *pregrasping opposition* $O^s_{h1-h2}(C_o)$ given the guidance of the information Ω.
7: Plan the final grasp configuration $g = (C_g, Joint_g, Pose_g)$ given the guidance of the $O^s_{h1-h2}(C_o)$.
8: Grasp execution.

5.5 Experiments

In this section, we evaluate the effectiveness of the proposed grasp planning framework both in a simulation experiment and in a real-world experiment. The experimental setup, result, and discussion are provided in the following each subsection, respectively.

5.5.1 Evaluation of Grasp-relevant Information Execution

The performance of the grasp-relevant information execution is first evaluated. Chapter 4 has evaluated the performance of grasp type detection. Here, we mainly demonstrated the result of the grasp type detection when combining with a learning-based object detection method. At each trial, multiple objects were randomly selected and separately placed on the table. Kinect depth sensor captured an RGB-D image of the table scenes. Figure 5.5 shows some results of the grasp-relevant information execution on various objects. Given an input RGB image, the object region denoted by a rectangle and its id are first determined. YOLO-2 is used to find the object region O is an RGB image. Based on the detected ids, a human chooses the desired object to be grasped. Six pixel-wise probability maps are obtained from the grasp type detection model. The grasp attention points denoted by a blue cross in each probability map are obtained by clustering. The grasp type with the highest probability in the object region is selected. For each trial, the object detection takes 2.5 seconds, grasp type detection takes 4.2 sec-

onds, and the whole computation of grasp-relevant information takes about 7.4 seconds. The proposed method is implemented in python and runs on a 2.5GHz Intel i5 CPU.

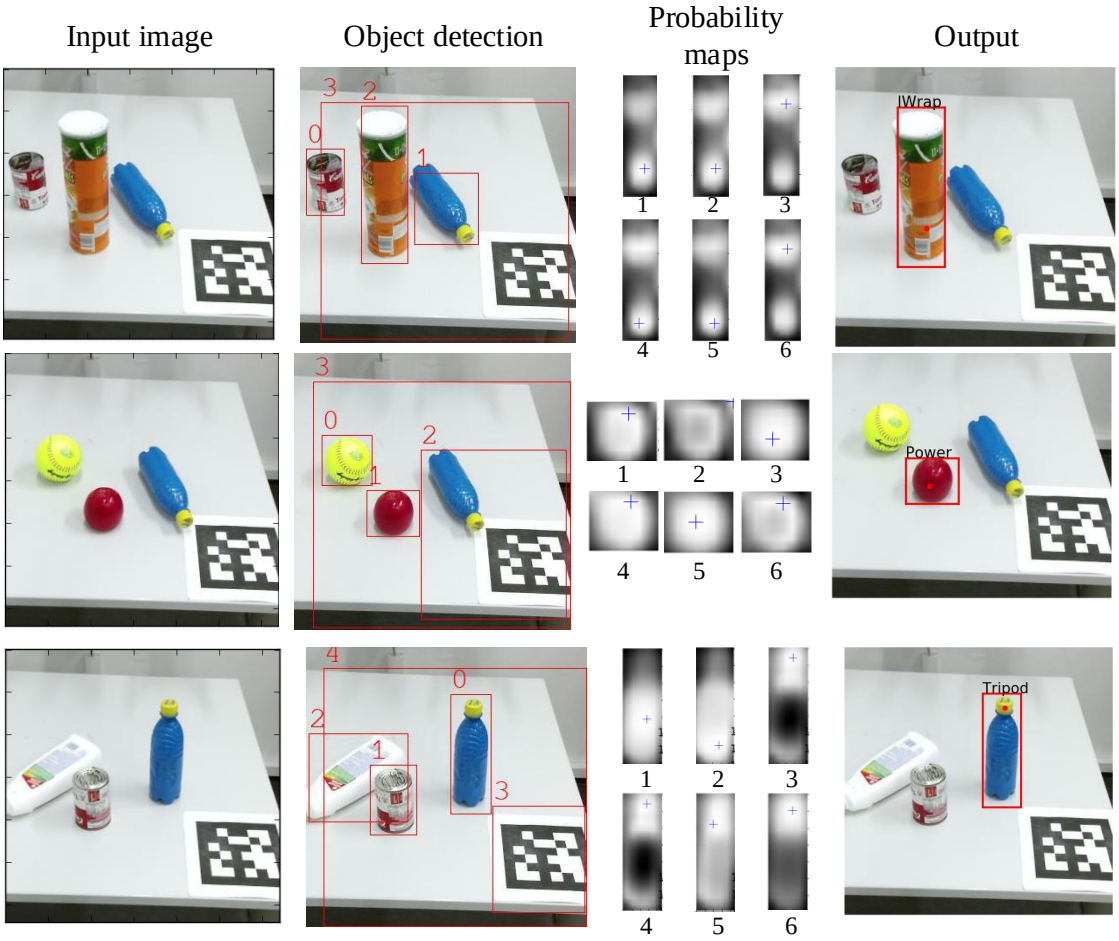


Figure 5.5: Example of of the grasp-relevant information extraction. First column is the input RGB images captured by a Kinect sensor. Second column is the results of object detection. The red rectangle denotes the object region and the texture above the rectangle is the object id number. Third column is six pixel-wise probability maps in the object region. From left-top to right-bottom, the probability maps corresponds six grasp types: *large wrap*, *small wrap*, *power*, *pinch*, *precision* and *tripod*. The blue cross denotes the grasp attention point for each grasp type. Last column is the output.

Extracting grasp-relevant information from visual data is important for robotic grasping and manipulation. Different from previous methods, this work takes advantage of deep learning techniques to detects the grasp type and grasp attention point directly from visual data. It is a natural behavior for humans to choose a feasible grasp type based on the object image to grasp an object. Hence, the proposed grasp type detection method enables a robotic system with similar grasping behavior as humans. We also noticed that previous methods also used CNN to detection some information, for example, the grasp affordance, the grasp region, and the grasp rectangle. Most of the previous methods focus on visual analysis and do not be applied in robotic grasping applications. They

(a) (b) (c)

Figure 5.6: Three robotic hands used in the simulation experiment. (a) The P-Grip[4]. (b) The Barrett hand[5]. (c) The Shadow Dexterous hand[6]

Figure 5.7: Eight objects used for the simulation experiment.

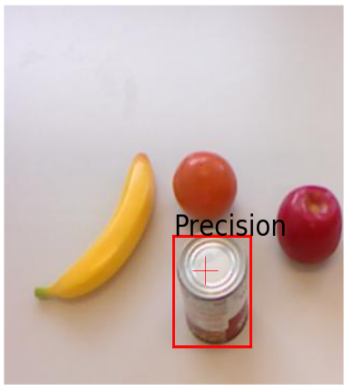
use a simpler parallel-jaw gripper to realize the detection result. This work aims to extract grasp-relevant information for general-purpose grasp planning. The grasp-relevant information is exploited into the general-purpose grasp planning method. In this way, the extracted information helps the robotic system to implement more complex grasping tasks.

5.5.2 Evaluation of General-purpose Grasp Planning

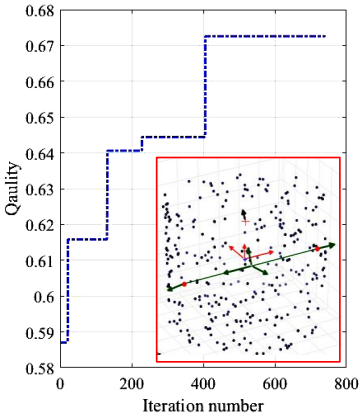
Implementation

The evaluations of the proposed general-purpose grasp planning method were conducted in a V- REP [7] simulator. Three robotic hands with a different number of fingers were used, as shown in Figure 5.6. They were the P-Grip [8] with two-fingers, the Barrett hand [9] with three fingers and the Shadow hand [10] with five fingers. The six grasp types are considered in this evaluation. Eight objects with various shape attributes were chosen

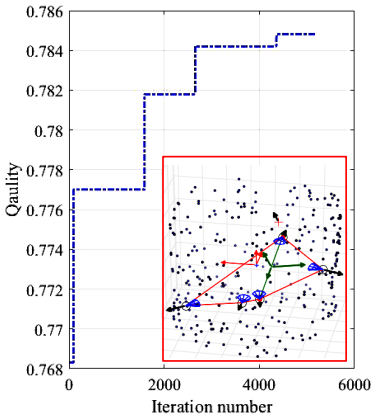
[7]http://www.coppeliarobotics.com/
[8]https://www.fp-robotics.com/de/technology/
[9]https://www.barrett.com/about-barrethand/
[10]https://www.shadowrobot.com/products/dexterous-hand/



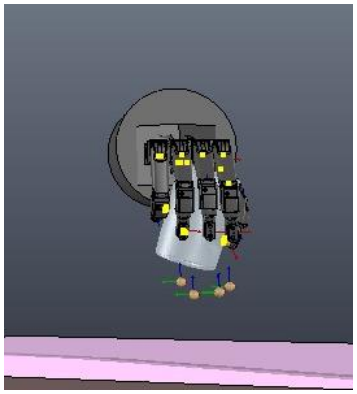
(a) Grasp type detection



(b) The first planning stage



(c) The second planning stage



(d) Grasp execution by the Shadow hand

Figure 5.8: An example of object grasping by using the Shadow hand. (a) The grasp-relevant information extraction. The text was the detected grasp type, and the red cross denoted the location of the grasp attention point. (b) the *pregrasping opposition* involving a pair of opposing points was constructed at stage 1. The curve describes the highest quality sampled in each iteration using simulated annealing. (c) The final grasp configuration involving five contact points was generated at stage 2. The curve describes the highest quality sampled in each iteration using simulated annealing. (d) The Shadow hand executes the planned grasp configuration.

from YCB object set [11] for this evaluation, as shown in Figure 5.7. YCB object set provides the point clouds of these objects. At each grasping trail, multiple objects were placed on a table. A depth sensor was used to capture RGB images of a table scene. Then, the proposed planning approach generated a feasible grasp configuration for a robotic hand. Finally, the grasp configuration was executed by a robotic hand in the V-REP simulator.

The setting of the proposed planning approach was as follows. A hard contact model

[11] http://www.ycbbenchmarks.com/

was adopted, and the friction coefficient was chosen as 0.6. During grasp optimization, the lower bound was set to $\varepsilon = 0.7$ in Eq. 5.2. At stage 1, the two weights in Eq. 5.2 were set as $\omega_R = 0.3$ and $\omega_G = 0.7$. At stage 2, the two weights in Eq. 5.2 were set as $\omega_R = 0.7$ and $\omega_G = 0.3$. In this way, the grasp attention made a bigger contribution in the first stage, and the grasp reachability made a bigger contribution in the second stage. The maximum iteration number (i.e., $maxInter_1$ and $maxInter_2$) were set as 10% of the number of candidate grasps and the maximum temperature (i.e., $maxTemp_1$ and $maxTemp_2$) of the optimization method were set to be 25000.

Experimental Results

We first demonstrated the grasp planning process using the proposed approach. Taking the test of the object grasping with the Shadow hand as an example, as shown in Figure 5.8. In this example, the proposed approach planed a feasible grasp configuration on the tomato soup can. The Shadow hand grasped and lifted the can. It can be seen that the *Precision* type was selected for the can and the p_{att} on the object surface was chosen. The p_{att} also specifies the desired approaching location on the object surface. The proposed dual-stage grasp planning approach was directly performed on point clouds of the can. At stage 1, the *pregrasping opposition* $O^5_{FT-TT}(C_o)$ involving a pair of opposing points C_o was constructed. The opposing points C_o was optimized to get a high grasp quality. At stage 2, the final grasp configuration involving five contact points was generated with the guidance of the $O^5_{FT-TT}(C_o)$. Finally, the resulted grasp configuration was successfully executed by the Shadow hand in the V-REP simulator. Figure 5.8 (b,c) shows the curve of the highest objectives Q sampled during configuration optimization. The Q were increasing during optimization at stage 1 and 2. It can be seen that the complexity of configuration optimization has an exponential relationship with the number of candidate contact points on the object. The use of the *pregrasping opposition* reduced the complexity of precision grasp planning.

Next, the use of *pregrasping opposition* in grasp planning was also evaluated. We compared the grasp planning approach with and without *pregrasping opposition*. The optimization algorithm and the parameters of the two approaches kept same. The two approaches were used to plan grasp configuration under the *tripod* grasp type for object grasping with the Shadow hand. The final grasp configuration involved three contact points. Table 5.1 shows the comparison results of the two approaches. It can be seen that the number of possible solutions by the approach with *pregrasping opposition* (i.e., the size of the solution space Φ_g) is only 3.6% compared to that of the approach without *pregrasping opposition*. The use of *pregrasping opposition* was able to ignore infeasible solutions and reduces the feasible solution space. The results demonstrate that the use of *pregrasping opposition* helps to reduce the complexity of the grasp planning. Meanwhile, it can be seen that the average consuming time required by the proposed approach is 3.0% of that of the approach without *pregrasping opposition*. It also demonstrates that the grasp planning with *pregrasping opposition* is more efficient in comparison to the approach without *pregrasping opposition*.

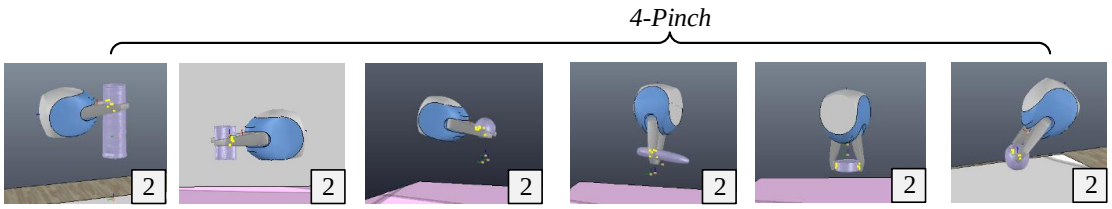
To further evaluate the effectiveness of the proposed grasp planning approach, a set of object grasping tasks was performed using the three robotic hands. Table 5.2 records

Table 5.1: Evaluation of the use of *pregrasping opposition* in grasp planning.

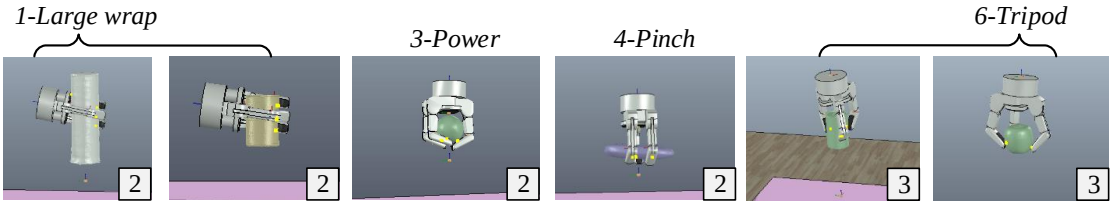
object (#points)	With O^s_{H1-H2}		Without O^s_{H1-H2}	
	size of Φ_g	Time	size of Φ_g	Time
tomato soup can (120)	6256	1.8	120^3	61.1
potted meat can (120)	7518	2.5	120^3	59.3
mustard bottle (120)	4976	1.15	120^3	57.5
Average	**6253**	**1.81**	120^3	59.3

the statistic of the evaluations. Six objects from the YCB object data set were used for the evaluations, as detailed in Table 5.2. A grasp is a success only if a hand can stably grasp and lift an object to a distance from the table. From the table 5.2, it can be seen that the P-Grip successfully grasped all the objects and obtained a 100% success rate. The grasp configuration of the P-Grip only involved two contact points. We used the *pregrasping opposition* that is determined at stage 1 as the final grasp configuration of the P-grip. The results demonstrate that the *pregrasping opposition* can be used as the stable grasp configuration for the two-fingered robotic hand. Meanwhile, the success rates of Barrett's hand and the Shadow hand are 90.0% and 88.7%. The proposed planning approach was able to find feasible grasp configurations for the three and five-fingered robotic hands. The results demonstrate that the *pregrasping opposition* can provide a way to generalize the grasp planning approach to various robotic hands. Moreover, a complex grasp configuration can be generated from *pregrasping opposition*. We also notice that the success rates of Barrett's hand and the Shadow hand have decreased in comparison to the P-Grip. Most of the failures were due to the grasp force was not enough to support the object weight. Although the planed grasp configurations were feasible, the hands still failed to lift objects. For each grasping experiment, the grasp-relevant information extraction takes 7.4 seconds, the grasp planning takes 2.4 seconds, and the complete process takes 11 seconds on average. The proposed approach is implemented by Python in the Ubuntu laptop with a 2.50GHz Intel i5 CPU.

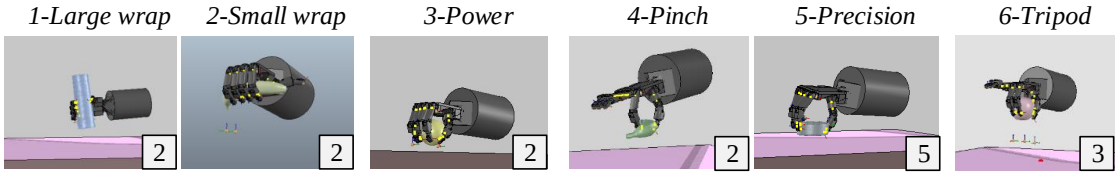
Figure 5.9 shows some examples of object grasping using the three robotic hands. It can be seen that objects with different shape attributes have different desired grasp types. Different grasp types required a different number of contact points. The number in each sub-figure denoted the required number of contact points that was determined by the desired grasp type and the robotic hand. For example, the precision grasping with the Shadow hand includes five contact points. While the precision grasping with the Barrett hand only contains three contact points. The *precision* grasp was most difficult because it required finding more contact points. The results show that the proposed planning approach allows various robotic hands to realize feasible grasp types during object grasping. We also see that the P-Grip only uses the *pinch* type to grasp objects. For the Barrett hand, the *large wrap* and the *small wrap* are same and the *precision* and the *tripod* are also same. The Shadow Hand can execute all six grasp types. A robotic hand with a more number of fingers is able to complete more dexterous grasping tasks,



(a) object grasping with the P-Grip



(b) object grasping with the Barrett hand



(c) object grasping with the Shadow hand

Figure 5.9: Examples of object grasping using three different grippers considering six different grasp types. The number in the rectangle of each image showed the required number of the contact points.

Table 5.2: Performance of the proposed grasp planning.

object	P-Grip	Barret hand	Shadow hand
tomato soup can	10/10	10/10	9/10
tuna fish can	10/10	8/10	8/10
plotted meat can	10/10	9/10	9/10
mustard bottle	10/10	8/10	10/10
banana	10/10	8/10	7/10
apple	10/10	10/10	9/10
orange	10/10	9/10	9/10
chips can	10/10	10/10	10/10
Average success rate	100%	90.0%	88.7%

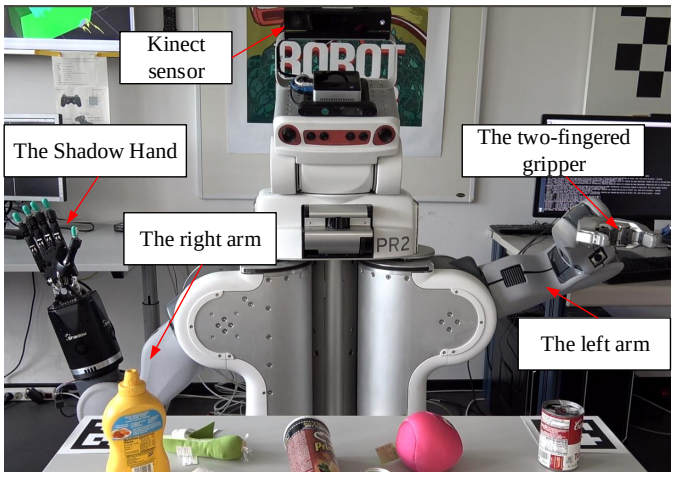


Figure 5.10: The set-up of the real-world robotic experiment with a PR2 robotic platform.

Figure 5.11: Ten objects used in the real-world robotic experiment.

such as the *precision* grasp. The experimental results also show that the proposed grasp planning approach can find feasible grasp configurations for various robotic hands with an arbitrary number of fingers.

5.5.3 Real-world Robotic Experiment

Implementation

The robotic experiments were conducted on a PR2 robotic platform that has two robotic arms, as shown in Figure 5.10. The right arm equips a Shadow Dexterous Hand, i.e., a five-fingered robotic hand. The end-effector of the left arm is a two-fingered gripper. A Kinect sensor was used to capture the visual data (i.e., RGB-D image and point clouds) of the table scenes. Ten objects selected from the YCB object set was used for the

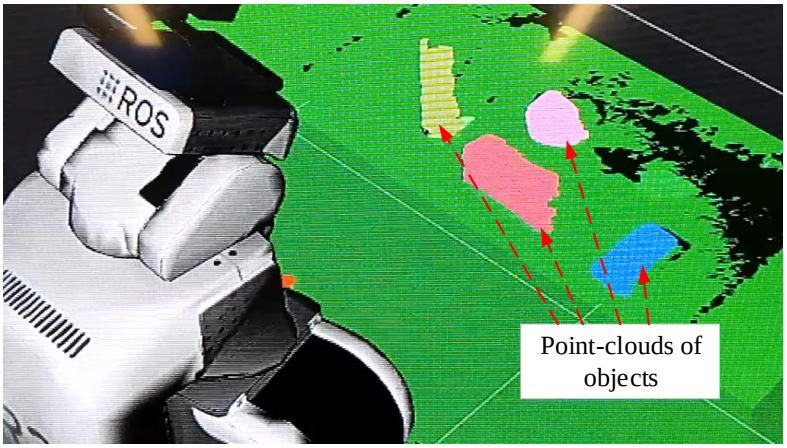
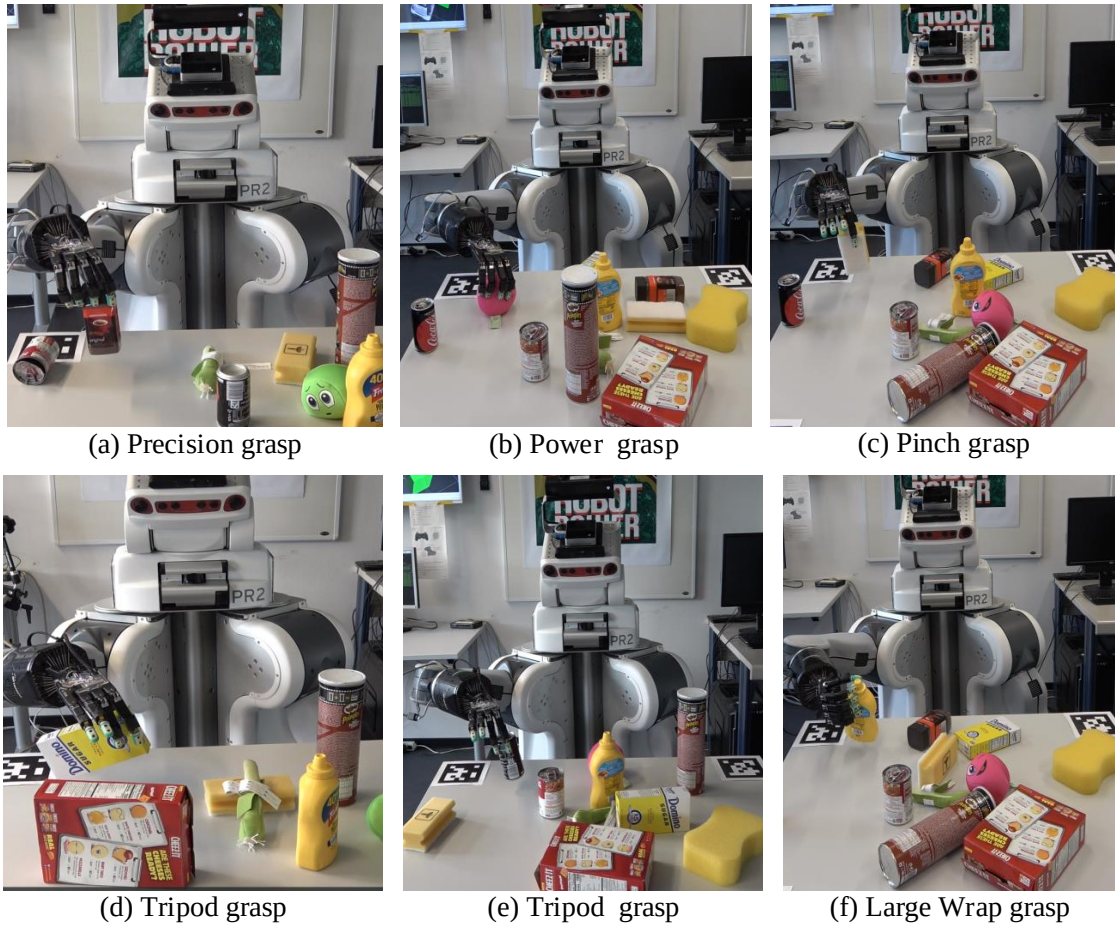


Figure 5.12: Example of the Segmented point clouds of objects on a table.

evaluation, as shown in Figure 5.11. In the object grasping experiments, we adopted the following procedure. At each trial, multiple objects were randomly selected and placed on the table. A human chose a robotic hand (i.e., the Shadow Dexterous Hand or the two-fingered gripper) for object grasping. The point clouds of the objects on the table are segmented from the table. The grasp-relevant information (i.e., the grasp type and grasp attention point) is computed based on grasp type detection. Then the proposed dual-stage grasp planning approach determines the feasible grasp configuration considering the computed information. At final, the PR2 robotic platform executed the planned grasp configuration.

Experimental Results

Different from the simulation experiments mentioned above, the object model is unknown in the real-world robotic experiment. We use the Kinect sensor to capture the point clouds of the table scene. The grasp planning is performed on the partially observed point clouds of the target objects, which were segmented from the point clouds of the scene. Figure 5.12 shows an example of the segmented point clouds of objects on a table. It can be seen that the point clouds of multiple objects were segmented from the table scene. Figure 5.13 shows some examples of the object grasping using the Shadow Dexterous Hand. Although the point cloud of the target objects is not complete, the feasible grasp configuration could still be determined by using the proposed framework. The planing of the *pregrasping opposition* determines a pair of opposing points that could ensure the stable of grasping. The detected grasp attention point guides the hand to search the contact points on the observed point clouds of the target objects. The use of the *pregrasping opposition* helps to reduce the sample complexity of grasp planning. As shown in the experiments, different grasp types were used to grasp the objects with different shape attributes. In the real-world application, it is an important ability for the robotic hand to choose a feasible grasp type from different possible grasp types to operate the objects. The Shadow hand can execute multiple grasp types. A robotic hand with a more number of fingers could accomplish more sophisticated grasping tasks.



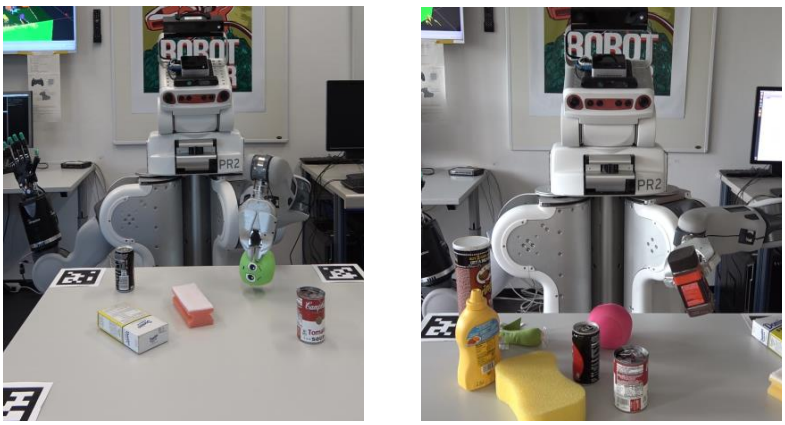
(a) Precision grasp (b) Power grasp (c) Pinch grasp

(d) Tripod grasp (e) Tripod grasp (f) Large Wrap grasp

Figure 5.13: Examples of object grasping using the Shadow Dexterous Hand. In each sub-figure, the shadow hand grasps the object using the detected grasp type.

Figure 5.14 shows some examples of the object grasping using the two-fingered gripper. For the two-fingered gripper, only two contact points are required for a grasp. Hence, we take the *pregrasping opposition* planned by the proposed framework as the final grasp configuration. From the Figure 5.13 and Figure 5.14, it can be seen that the *pregrasping opposition* can encode not only the grasp-relevant information and provide a way to generalize the grasp planning method across different robotic hands.

It is worth mentioning that several failures of object grasping have occurred. The main reason for the failure is because there are no complete point clouds of the target object. The segmented point clouds of the target object contain noise, especially for the small object. Another cause was that the robotic arms failed to find a feasible kinematic solution. Because the workspace of the robotic arms is limited, the robotic hand is unable to grasp the distant objects. It also occurred that the object fell out of the gripper during lifting. The uncertainty from the object weight caused it. In the future, it will also be beneficial to incorporate grasp adaptation into the proposed framework.



(a) Power grasp (b) Large Wrap grasp

Figure 5.14: Examples of object grasping with two-fingered gripper.

5.6 Discussions

Grasping is a fundamental ability for a robotic hand that has to interact with its environments. Different types of robotic hands (such as two- or five-fingered robotic hands) are required in the industrial and home-assistance applications. This work explores the general-purpose problem of grasp planning, which is important for intelligent robots. The motivation of this work includes two points: (1) To enable robotic hands to grasp objects using a feasible grasp type as humans do. (2) To design a grasp planning method that suits various robotic hands with an arbitrary number of fingers. In terms of motivation 1, previous grasp planning methods only consider two grasp types, i.e., the *Power* and the *Precision*. Moreover, the grasp types are determined manually for the robotic hands. However, humans are capable of operating objects by choosing a feasible grasp type from multiple possible grasp types. Feix et al. introduced a novel grasp taxonomy in which 33 different grasp types are presented [28]. Hence, we proposed taking advantage of deep learning techniques to detect grasp types directly from visual data. A novel concept of *pregrasping opposition*, is introduced to encode the detected information, which is used to guide the grasp planning. The experimental results showed that the proposed methods enable a robotic hand to grasp objects with different grasp types. In terms of motivation 2, few works have considered the general-purpose problem of grasp planning. Most previous grasp planning methods were designed for one specific hand. To address this problem, a dual-stage grasp planning method beginning with the *pregrasping opposition* is designed. Complex grasp configuration, such as the *precision* grasp configuration, are generated from a simple *pregrasping opposition*. The performance of the proposed framework is evaluated both in simulation and real-world experiments. The proposed framework is quite suitable for the scenarios in which the robots equipped with different robotic hands work together. Instead of designing different grasp planning methods for each robotic hand, the proposed framework can plan grasp configurations for various hands with an arbitrary number of fingers.

5.7 Conclusion and Future Work

This work presented a general-purpose grasp planning framework that suits various robotic hands with an arbitrary number of fingers. By using the proposed framework, grasp-relevant information (i.e., the grasp type and grasp attention point) is extracted using a trained neural network. Given the extracted information as guidance, the *pregrasping opposition* involving a pair of opposing points is constructed. The final grasp configuration involving a set of contact points is planned with the guidance of the *pregrasping opposition*. The use of the *pregrasping opposition* not only incorporates grasp-relevant information into grasp planning but also provides a way to generalize the grasp planning method to various robotic hands with an arbitrary number of fingers. The dual-stage grasp planning method is also able to plan feasible grasps for various robotic hands effectively. Furthermore, the performance of the proposed framework has been evaluated both in simulation and real-world robotic experiments.

Improvements can be achieved in the following directions. First, the use of grasp types is quite essential for multi-fingered robotic hands to implement object grasping tasks. In this work, six commonly used grasp types are considered. It would be interesting to consider more grasp types with reference to the grasp taxonomy of Feix et al. [46] for multi-fingered grasp planning. Second, the choice of grasp type currently only depends on the object attributes. Because grasp planning is also affected by task constraints, those constraints should also be taken into consideration. One approach to address this problem would be to use Bayesian methods to build the relationship among the grasp type and the object attributes and task variables. Finally, we also plan to use tactile information obtained from tactile sensors to avoid slippage and improve grasp stability.

Chapter 6

Dynamic Object Stabilization through Tactile Sensing

6.1 Introduction

Stable grasping of unknown objects is one of the fundamental abilities for robots implementing manipulation tasks in real-world environments. External disturbances caused by environmental changes and uncertainties arising from perception may occur when a robotic hand grasps a target object. This may cause a planned-to-be stable grasp into an unstable grasp. Maintaining the in-hand stability of grasped objects over time is critically important for further robotic manipulations. Humans are capable of reacting to quickly instabilities through tactile sensing. Studies from neuroscience have demonstrated that tactile perception provides critical information about the physical properties of the object and the contact event between the object and the hand [81]. Through tactile sensing, humans take tactile information as feedback to adjust their grasp configurations to improve the stability of objects. However, dynamic object stabilization through tactile sensing for robotic hands is still unexplored. To effectively stabilize a grasped object with a robotic hand, three related problems should be addressed: (1) How to detect the properties (such as material) of the grasped object. (2) How to effectively perceive the contact event (i.e., slippage or not). (3) How to correct the contact configuration of a robotic hand online through tactile sensing to avoid slippage.

The goal of object stabilization is to maintain the contact between the robotic hand and the object, and to avoid slippage. The instability of a grasped object may be caused by the slipperiness of the object or an external disturbance. A slip occurs when the contact force applied by the hand is insufficient. Online slip detection is of critical importance for robots to stabilize grasped objects. Once slippage is detected, the robotic hand updates the contact configuration to avoid slippage. Tactile information has been used as the main modality for slip detection. Previous methods usually formulated slip detection as a classification problem, in which a classifier could be built with a SVM [22], random forest [171] or neural network [167, 180]. However, these work requires to design handcrafted features for slip detection. It is also useful for robots to know the physical properties of the grasped object. These are usually unknown to robots, but they are essential

for object stabilization. Information on the object material could help to decide how the robotic hand interacts with the object in advance. Although some works [27, 50, 60] on material detection have been introduced, few works have incorporated material detection into object stabilization. There are two main differences between the proposed method and these previous methods: (1) In contrast to previous methods that consider slip detection and material detection separately, this work trains an online detection module to detect the contact event and the material of the grasped object simultaneously. (2) Our approach uses deep learning techniques to learn a latent representation of tactile data, which captures the temporal property of tactile data and is a low dimension. In this way, it is not necessary to design features of tactile data manually for classification.

Object stabilization relies on a feedback controller that drives the robotic hand to adjust the contact configuration in order to track the desired contact configuration or to avoid slippage. Tactile sensors could provide critical contact information between the robotic hand and the object and have been widely used for object stabilization [146, 158, 36]. One important technique in object stabilization is to estimate the contact force from tactile data. The estimated force is then taken as the feedback signal for the feedback controller. Because tactile data is usually high-dimensional, most of the previous methods define the contact force as the weighted sum of the tactile data. These methods do not consider the spatial property of tactile data. Moreover, these methods ignore the computation of the contact region on the fingertip. This work proposed forming a tactile image that captures the spatial property of tactile data. We use a GMM to determine the contact region on the tactile image. The contact force and location are calculated based on the tactile data features in the contact region. The proposed force estimation method makes the tactile data more suitable for feedback control. Meanwhile, previous methods mainly used a fixed desired contact force that was manually predefined as the reference signal for the feedback controller. These methods were limited to handling certain similar or known objects and were unable to grasp unknown objects. To address this limitation, we integrate a force feedback controller with material detection stably. The controller sets the desired contact force automatically according to the material detection results.

In this work, we jointly address the problems of slip detection, object material detection, and force feedback control. The objective is to enable the robot to make use of tactile sensing for object stabilization. We propose a tactile sensing method for a multi-fingered robotic hand to stably grasp unknown objects without prior knowledge of their shape or physical properties. The proposed framework consists of three components: an online detection module, a force estimation method and an object stabilization controller. The online detection module detects the contact event and the object material directly from tactile data. The online detection module consists of three components: a Long Short-Term Memory (LSTM)-based encoder-decoder architecture that learns a latent representation of tactile data, a slip classifier that predicts the contact event, and a material classifier that predicts the object material. Here, the temporal property of the tactile data is fully analyzed and used for detection. Next, by exploiting the detection results, the force feedback controller is employed to drive the robotic hand to adjust its contact configuration online. To represent tactile data as feedback signals for the feedback controller, a novel force estimation method based on GMM is proposed to cal-

culate the contact force and location. The force estimation method takes advantage of the spatial property of tactile data. Hence, this work makes use of the spatio-temporal property of tactile data for tactile sensing. The performance of the proposed framework is demonstrated with a Shadow Dexterous Hand equipped with BioTac sensors. The following contributions are made in this work.

- An online detection module is introduced, which trains a deep neural network to detect contact events and object materials directly from tactile data. An LSTM-based encoder-decoder is employed to reduce the dimension of tactile data and capture the temporal property of tactile data. A new tactile dataset is introduced for tactile sensing. The dataset contains twelve household objects with four different materials.

- A novel force estimation method based on GMM is proposed to calculate the contact force and location when a robotic hand interacts with an object. The proposed estimation method uses a tactile image to capture the spatial property of tactile data.

- An object stabilization framework is introduced for a multi-fingered robotic hand to stabilize the grasped object. In the framework, the results of slip and material detection are exploited as guidance and an object stabilization controller is employed to adjust the contact force between the robotic hand and the object online.

The rest of the section is organized as follows. An overview of the proposed framework is introduced in Section 6.1. Section 6.3 introduces the online detection module. The force estimation method is presented in Section 6.4. Section 6.5 introduces the object stabilization controller. Experiments and their results are presented in Section 6.6. Finally, the conclusion and future work are discussed in Section 6.8.

6.2 Overview

In this section, we give an overview of the proposed object stabilization framework for a multi-fingered robotic hand to grasp unknown objects stably. The proposed framework consists of three components: an online detection module, a force estimation method, and an object stabilization controller. The detection module based on a deep neural network samples tactile sequences online from the tactile readings as inputs and predicts the object material and contact event simultaneously. A force estimation method based on a GMM is employed to process the tactile data online and calculate the contact information (i.e., normal contact force and contact location) of fingers. In the object stabilization controller, by exploiting the processed results of the tactile data, an object stabilization controller is employed to drive the robot hand to adjust its contact configuration online in order to track the desired contact configuration or to avoid slippage. Figure 6.1 illustrates an overview of the proposed object stabilization framework.

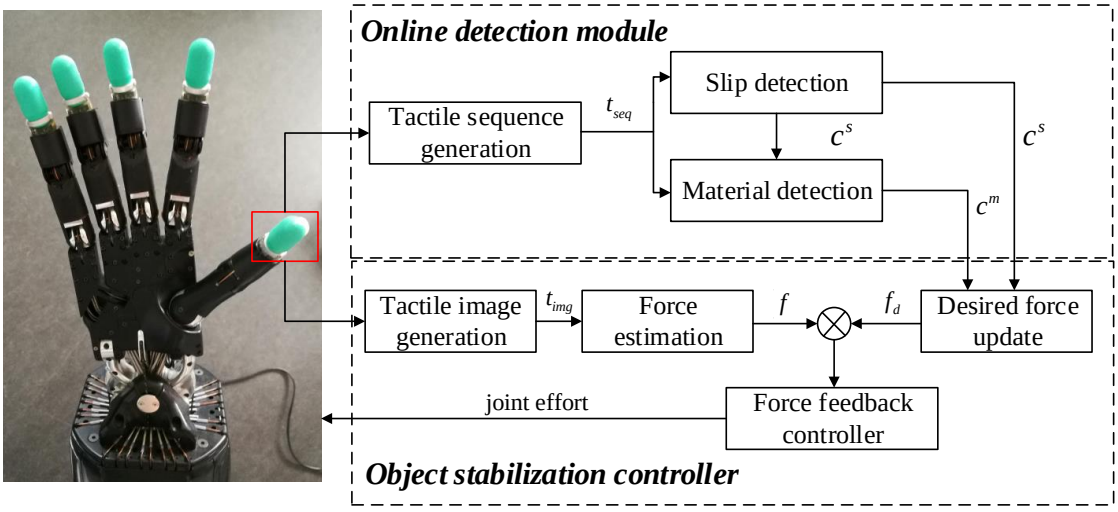
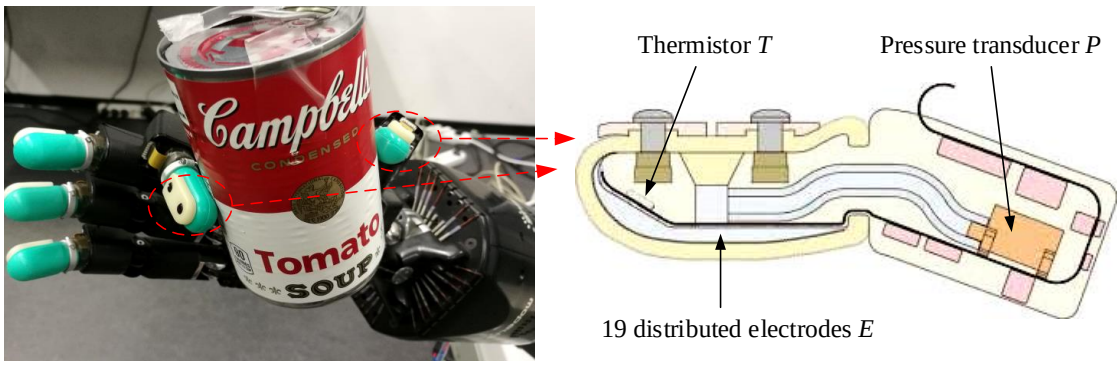


Figure 6.1: Overview of the proposed object stabilization framework. Firstly, the tactile sequence t_{seq} and the tactile image t_{img} are sampling from the tactile readings. The contact event c^s and the object material c^m are detected based on the t_{seq}. Next, the desired contact force f_d is updated based on the detection results. And the current contact force f is estimated based on the t_{img} . Finally, the object stabilization control is employed to drive the robotic hand to track the desired contact force f_d.



(a) Shadow Dexterous Hand

(b) Biotac tactile sensor

Figure 6.2: Shadow Dexterous Hand with BioTac sensors.

6.3 Online Detection Module for Slip and Material Recognition

In this subsection, we present an online detection module that takes a tactile sequence sampled from tactile readings as input and predicted the contact event and object material simultaneously. In the following, we first introduced the sampling process of the tactile sequence and then detailed the proposed online detection module.

In this work, we use a Shadow Dexterous Hand equipped with BioTac sensors for tactile data collection and experiments, as shown in Figure 6.2. The BioTac sensor is a multi-channel tactile sensor and it includes: (1) a pressure transducer P for pressure vi-

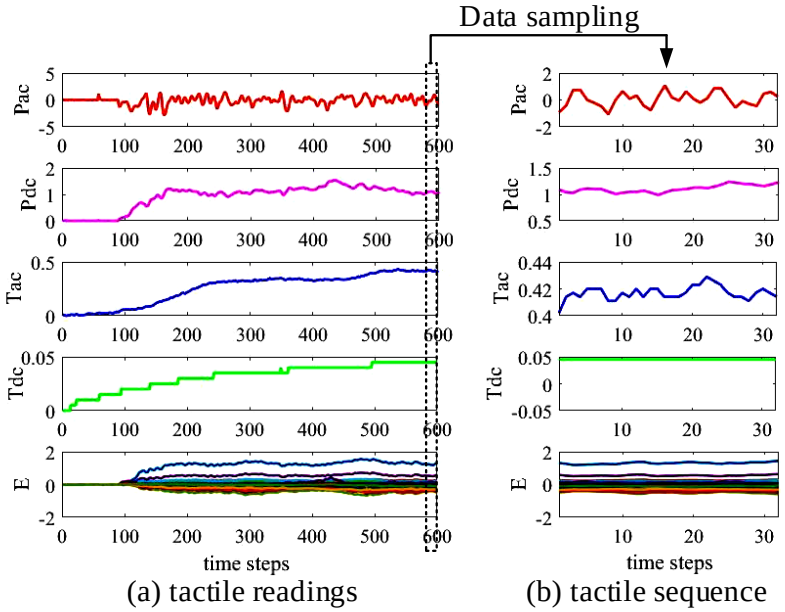


Figure 6.3: Example of a tactile sequence sampled from a tactile reading. (a) tactile readings: it is recorded from the BioTac sensor, which has 23 dimension tactile features. (b) tactile sequence: it has a dimension $32 * 23$. The window size is set to 32.

brations sensing, (2) a thermistor T for measuring heat flow, (3) 19 distributed electrodes E for detecting the local pressure at different locations of the sensor. The BioTac sensor has multi-modal sensory capabilities and provides 23 dimension tactile features which includes low-frequency fluid pressure (P_{dc}), high-frequency fluid pressure (P_{ac}), core temperature (T_{dc}), core temperature change(T_{ac}) and 19 electrodes ($E_1, ..., E_{19}$). Tactile data is recorded from the BioTac sensor at 100 Hz. To obtain a input for the detection model, we sample a tactile sequence t_{seq} with a window size l from the tactile readings, i.e., $t_{seq} = [P_{dc}, P_{ac}, T_{dc}, T_{ac}, E_1, ..., E_{19}]$. Thus, the tactile sequence t_{seq} has a dimension $l * 23$ where l is the window size and 23 is the number of tactile features. Figure 6.3 shows that the sampling process of the tactile sequence t_{seq} from the tactile readings. The tactile sequence t_{seq} capture the temporal property of tactile data which is taken as the input of the proposed online detection module.

This work aims to solve a multi-task classification problem where the contact event and object material are detected simultaneously based on tactile data. Instead of training two complex networks for the two detection tasks, this work proposed an online detection module. The architecture of the proposed detection module is illustrated in Figure 6.4. The proposed model first to learn a low-dimension latent representation of the tactile sequence and take it as the input for slip and material detection. The use of the low-dimension latent representation of the tactile sequence help to reduce the model complexity and enable different classifiers to share the tactile feature.

An LSTM-based encoder-decoder is employed to learn the low-dimension latent representation of a tactile sequence. The encoder-decoder consists of three parts: an encoder E, a latent representation h, and a decoder D. The encoder takes the tactile

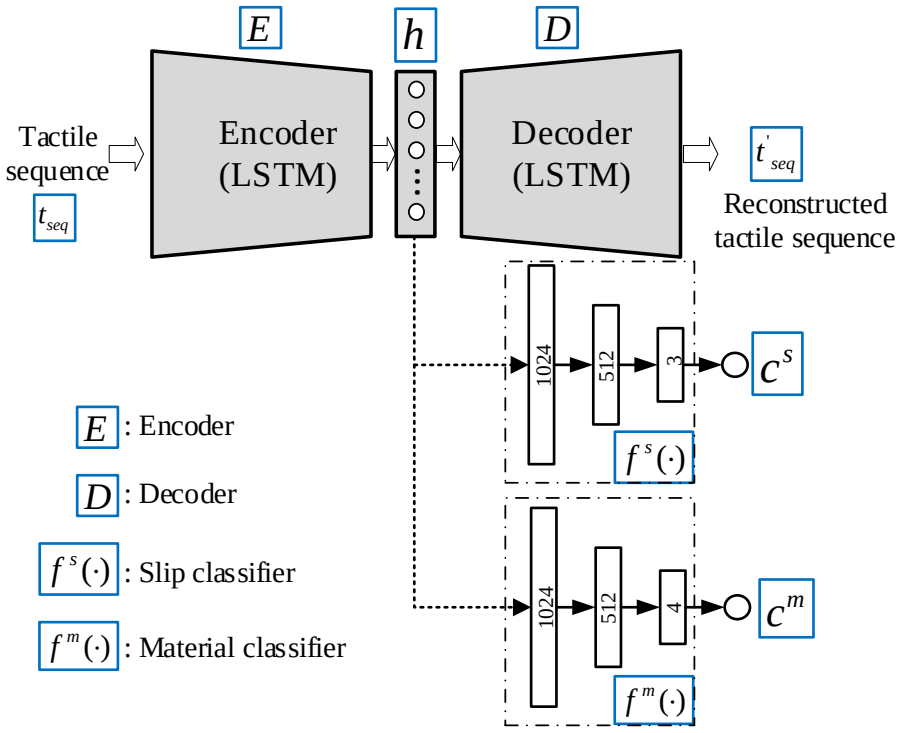


Figure 6.4: The architecture of the online detection module. The detection model consists of three components: an LSTM-based encoder-decoder, a slip classifier, and a material classifier.

sequence t_{seq} as input and outputs the latent representation h that is a feature vector with a fixed length. The decoder takes the latent representation h as input and reconstructs the inputted tactile sequence. Because the tactile sequence t_{seq} is time series, we take advantage of the LSTM to capture the temporal characteristics of the tactile sequence. The LSTM is a kind of recurrent neural network which excels at time series processing. Hence, this work employs LSTM recurrent units in the encoder and decoder. The LSTM unit takes the current frame x_t and the previous hidden states h_{t-1} as inputs and produces its hidden state h_t and the output o_t. The forward pass of the LSTM unit is summered as follows

$$
\begin{aligned}
i_t &= \sigma(W_{x,i}x_t + W_{hi}h_{t-1} + W_{c,i}c_{t-1} + b_i),\\
f_t &= \sigma(W_{x,f}x_t + W_{h,f}h_{t-1} + W_{c,f}c_{t-1} + b_f),\\
g_t &= tanh(W_{x,c}x_t + W_{h,c}h_{t-1} + b_g),\\
c_t &= f_t c_{t-1} + i_t g_t,\\
o_t &= \sigma(W_{x,o}x_t + W_{h,o}h_{t-1} + W_{c,o}c_t + b_o),\\
h_t &= o_t tanh(c_t),
\end{aligned}
\tag{6.1}
$$

where σ and $tanh$ are the sigmoid function and the hyperbolic tangent function, respectively. c_t is the memory unit that stores the temporal information. The LSTM unit has four gates, i.e., the input gate i_t, the forget gate f_t, the cell gate g_t and the output gate o_t.

The four gates control the reading or modifying of the memory unit c_t. For more detail, please refer to [156].

The training of the LSTM-based encoder-decoder is in an unsupervised setting. Hence, we only requires tactile sequence t_{seq} contained in the proposed tactile dataset. In this work, the encoder-decoder run through four tactile sequences $\{t_{seq,i}\}_{i=1:4}$ and produce four reconstructed tactile sequences $\{t^{'}_{seq,i}\}_{i=1:4}$. The squared loss between the inputted tactile sequence and the reconstructed tactile sequence is calculated as the loss function for the training of the encoder-decoder, as defined in Eq. 6.2.

$$L_1 = \sum_{i}^{4} ||t_{seq,i} - t^{'}_{seq,i}||_2^2 \tag{6.2}$$

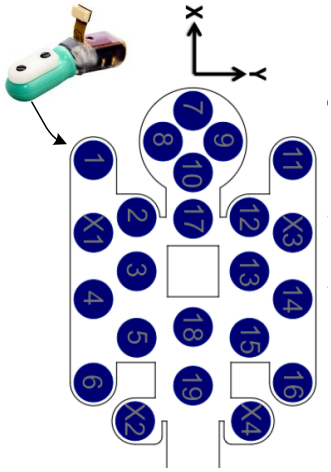
Next, we employ a slip classifier $f^s(\cdot)$ to achieve slip detection task, which classifies the latent representation h of the tactile sequence produced by the encoder-decoder as one of three different contact event c^s, as defined in Eq. 6.3. In this work, the three different contact events are the non-contact c^s_{non-c}, the contact $c^s_{contact}$ and the slip c^s_{slip}. This work uses a feed-forward network to form the slip classifier $f^s(\cdot)$. The network consists of three full-connected layers. We use a rectified linear unit (ReLU) nonlinearity at the two layers. The last full-connected layer is passed through a soft-max function to predict the class of the contact event. Figure 6.3 show the architecture of the proposed slip classifier. A cross-entropy function is used to define the loss function for training the slip classifier.

$$c^s = f^s(h), \quad c^s \in \{c^s_{non-c}, c^s_{contact}, c^s_{slip}\} \tag{6.3}$$

This work also trains a material classifier $f^m(\cdot)$ for object material detection, which classifies the latent representation h of the tactile sequence as one of the four different materials c^m, as defined in Eq. 6.4. The four different materials considered in this work are the paper c^m_{paper}, the foam c^m_{foam}, the plastic $c^m_{plastic}$ and the metal c^m_{metal}. The material classifier is trained with a feed-forward network which consists of three full-connected layers. We use a rectified linear unit (ReLU) nonlinearity at the two layers. The last full-connected layer is passed through a soft-max function to predict the contact event class. A cross-entropy function is used to define the loss function for training the material classifier. Figure 6.4 show the architecture of the proposed material classifier.

$$c^m = f^m(h), \quad c^m \in \{c^m_{paper}, c^m_{foam}, c^m_{plastic}, c^m_{metal}\} \tag{6.4}$$

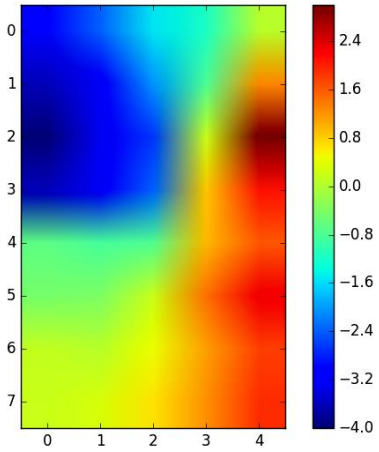
The online detection module is trained with the tactile dataset proposed in this work. The training and evaluation process of the detection module are detailed in Section 6.6. The online detection module takes the tactile sequence that online sampled from tactile readings as an input and output the detection results, i.e., the contact event and the object material. These detection results are used in the object stabilization control.



(a)

		7		
	8		9	
1		10		11
	2	17	12	
	3		13	
4				14
	5	18	15	
6		19		16

(b)



(c)

Figure 6.5: Definition of the tactile image. (a) 19 electrodes that distributes at different locations of the BioTac sensor. (2) tactile image that consist of a $8 * 5$ matrix. The number denotes the corresponding electrodes. (3) tactile image after filling the empty pixel in the tactile image. The pixel value of the tactile image represent the electrode data.

6.4 Force Estimation Based on GMM

One important technique in object stabilization is to estimate contact force from tactile data, which is then taken as a feedback signal for feedback control. This work proposes a force estimation method based on GMM that computes the contact information (i.e., contact force and contact point) of the robotic hand based on the tactile reading obtained from the tactile sensor. In the previous subsection, the tactile sequence t_{seq} that captures the temporal property of tactile data is sampled as the input of the proposed detection model. In term of force estimation, we focus on the spatial property of tactile data. We take advantage of the electrodes data and their connectivity to compute the contact information. As introduced in Section 6.3, the Biotec sensor has 19 electrodes that are distributed at different locations of the sensor surface. Figure 6.5(a) illustrates the layout of the ninth electrodes on the surface of the BioTac sensor. To capture the spatial relationship among the 19 electrodes, we defines a tactile image t_{img} consists of a $8 * 5$ matrix in x-y plane according to the locations of the 19 electrodes, as shown in Figure 6.5(b). The electrodes are arranged in matrix-like distribution within the surface of the BioTac sensor. The electrodes data is used to fill the pixel of the t_{img}. In this work, the pixels of the t_{img} are filled as follows. When a pixel of the t_{img} corresponds to an electrode, we take the electrode data as the pixel value. Otherwise, the pixel value is filled with the mean of its surrounding non-empty pixel values. Figure 6.5(c) shows an example of the tactile image t_{img} after filling the empty pixels.

The tactile image t_{img} is further used to calculate the contact information (i.e., contact location and contact force) between the fingertip and the object. From the tactile image shown in Figure 6.5(c), it can be seen that there is only a region with relative high pixel value of the fingertip that contact the object during the robotic hand interacts with the

object. Previous methods mainly defined the contact force as the weighted sum of all the electronic impedance. These methods made a hypothesis that all the electronics were supposed to contact the grasped object. In these works, the electronic values at the non-contact region were considered in the computation of the contact force. These methods may cause uncertainty from the non-contact region in object stabilization.

Different previous methods, we first determine the contact region on the sensor surface and then computes the contact information based on the contacted electrodes. This work makes a hypothesis for the segmentation of the contact region from the sensor surface. The hypothesis is that the contact region contains the electrodes with a relatively high electrode value. The hypothesis is reasonable because the contact between the fingertip and the object increases the pressure in the contact region. The segmentation of the contact region is processed as follows. The electronic impedance in the tactile image t_{img} is first clustered into several clusters. Then the electrodes with a related high electrode data are grouped to form the contact region Ω. In this work, we used GMM to cluster the tactile image. We use all the pixel value in the tactile image t_{img} to fit the GMM model denoted as Ω. The GMM is modeled as a mixture of K Gaussian distribution. The likelihood of an input e under a GMM is defined as

$$P(e|t_{img}) = \sum_{k=1}^{K} \pi_k N(e|\mu_k, \sigma_k) \tag{6.5}$$

Where K is the number of the Gaussian components used in the GMM Ω. The K is set to 3 in this work. e is a pixel value in the tactile image t_{img} . π_k is the prior of the k th Gaussian component. $N(x|\mu_k, \sigma_k)$ represents a cluster with mean μ_k and covariance σ_k. The parameters $\{\pi_k, \mu_k, \delta_k\}$ are estimated by maximum likelihood on the t_{img}.

After the GMM fitting, each cluster is described with its mean and covariance. The probability of a pixel belonging to each of the clusters can be calculated. We denote the cluster with related high electronic value as $P_c(e) = N(e|u_c, \delta_c)$. A pixel e is said to be contacted if the predicted probability $P_c(e)$ is more than a threshold α, as defined as

$$f(e) = \begin{cases} contact & P_c(e) >= \alpha, \\ noncontact & P_c(e) < \alpha, \end{cases} \tag{6.6}$$

All the pixels that predicted as a contact are grouped to form the contact region denoted as C_{img}. After segmenting the contact region C_{img}, we then calculate the contact information (i.e., contact location and normal contact force). In term of the computation of the contact location, we use a Gaussian component to fit the C_{img}. The mean of the fitted Gaussian component is used as the contact center denoted as $p = \{p_x, p_y\}$. In term of the computation of the normal contact force, we discard the influence from the pixel that predicted as non-contact during force estimation. We define the normal contact force f as a weighted sum of the value of the pixels that belong in the C_{img}, as defined in Eq. 6.7. Where n is the number of pixels in the C_{img}. $P_c(e_i)$ is the predicted probability

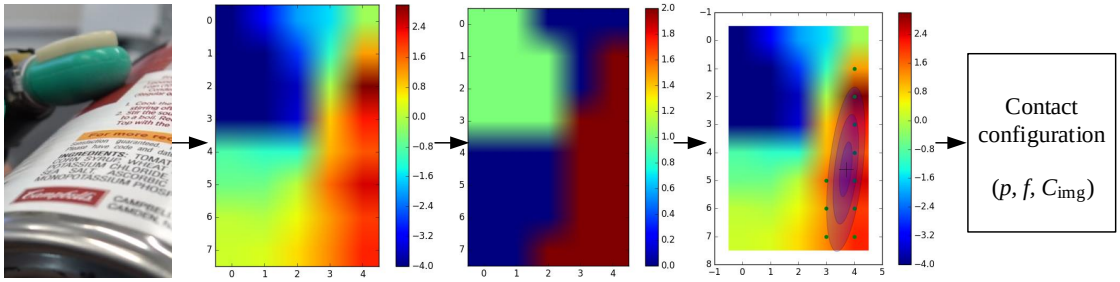


Figure 6.6: The process of the estimation of the contact information. (a) The BioTac sensor contacts with the object to produce tactile data. (b) The inputted tactile image t_{img}. (c) The tactile image is clustered based on GMM to obtain three different clusters denoted by different colors. (d) The contact region C_{img} is segmented and a Gaussian distribution is used to fit the contact pixels. (e) The contact information between the fingertip and the object is calculated.

of the pixel e_i which reflects the degree of contact of the pixel e_i.

$$f = \sum_{i}^{n} P_c(e_i) e_i \tag{6.7}$$

At final, The contact information, i.e., $\Psi = \{p, f, C_{img}\}$, between a robotic fingertip and the object is computed from tactile data. Figure 6.6 shows the process of the estimation of the contact information. The estimated contact information is then used for feedback control.

6.5 Object Stabilization Controller

Object stabilization controller is employed to drive the robotic hand to keep a feasible contact force to stabilize an object without damaging or dropping it. In this work, the object stabilization controller is built based on force feedback control. The controller controls the joint torque τ_i of each joint q_i of the robotic hand to track the desired contact force f_d in the Cartesian space. The object stabilization controller is independently performed in each joint of the robotic hand. Hence, each finger of the robotic hand is controlled to maintain a desired contact with an object to ensure the stable of the grasped object.

In this work, the desired contact force f_d is online updated autonomously according to the results of slip and material detection. The desired contact force is updated by taking the following steps:

1. A initial contact force f_0 is manually set for each fingers of the robotic hand.

2. We then update an initial contact force f_0 as the f_d^m according to the result of the material detection, as shown in Eq. 6.8. Objects with different materials require a

different contact forces exerted by the robotic hand to ensure the grasp stability.

$$f_d^m = \begin{cases} f_0 & c^m = c^m_{paper} \\ f_0 - \delta f_m & c^m = c^m_{foam} \\ f_0 & c^m = c^m_{plastic} \\ f_0 + \delta f_m & c^m = c^m_{metal} \end{cases} \tag{6.8}$$

3. Once a slippage is detected during the grasping process, we increase the desired contact force f_d^m applied by a fixed amount of δf_s resulting in the f_d^s. Otherwise the desired contact force f_{ds} is set to equal to the f_d^m, as shown in Eq. 6.9.

$$f_d^s = \begin{cases} f_0 & c^s = c^s_{noncontact} \\ f_d^m & c^s = c^s_{contact} \\ f_d^m + \delta f_s & c^s = c^s_{slip} \end{cases} \tag{6.9}$$

4. The desired contact force f_d is finally obtained by clipping the f_d^s into a safety region with the maximum and minimum value (i.e., $f_{d,max}$ and $f_{d,min}$) for control safety. During the grasping process, the desired contact force f_d is updated continually, the feedback controller is employed to driven the robotic hand to track the f_d.

This work employ a force feedback controller for the robotic hand to stabilize an object without damaging or dropping them. Figure 6.1 shows the diagram of the control architecture. The controller first takes the estimated contact force f as a feedback signal. The current contact force is compared with the desired contact force to calculate the force error, i.e., $\Delta f = f_d - f$. This work use a ProportionalIntegralDerivative (PID) control for force control, which take the force error Δf as input and computed the target torque. The target torque is then send to low-level torque controller to control the robotic fingers. Algorithm 5 shows the process of object stabilization through tactile sensing.

6.6 Experiments

This section first introduces the proposed tactile dataset for tactile sensing. We evaluate the performance of the proposed tactile sensing method. The effectiveness of the object stabilization method is also demonstrated in a real-world robotic experiment. The experiment results and their discussions are introduced.

6.6.1 Evaluation of Online Detection Module

Dataset and Implementation

A tactile dataset that suits for material recognition and slip detection simultaneously is still missing in the robotic community. Hence, this work introduces a new tactile

Algorithm 5 : Object stabilization through tactile sensing.

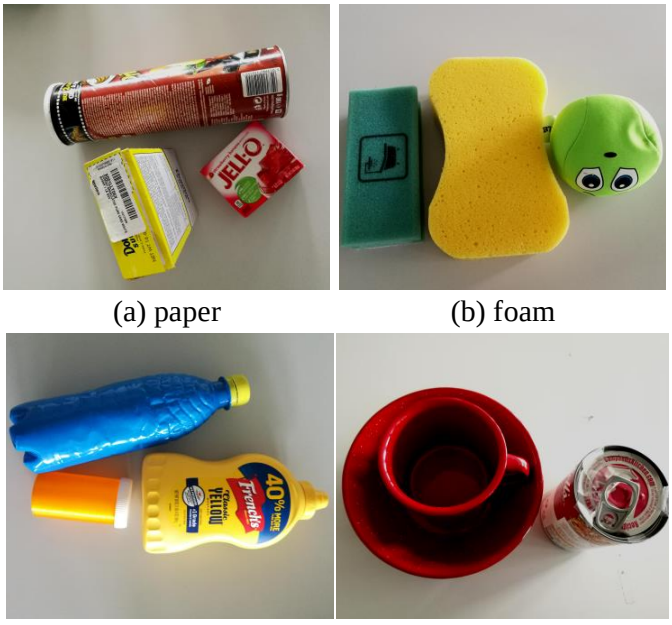
Requires: a trained online detection module, Tactile readings, a initial contact force f_0,
Repeat:
  Sample a tactile sequence t_{seq} and a tactile image t_{img} from tactile readings.
  Perform online detection based on t_{seq} and output contact event c^s and object material c^m.
  Perform force estimation based on t_{img} to calculate current contact information $\Psi = \{p, f, C_{img}\}$.
  If $c_s = c^s_{non-contact}$ **then**
    Perform position controller to control the joints of the robotic hand to reach desired joint position.
  else
    Update the desired contact force f_d based on c^s and c^m.
    Perform force feedback control to tracking the desired contact force f_d .

dataset [1] that includes the tactile data, the ground-truth of material and the ground-truth of contact event. In this dataset, three contact events (i.e., non-contact, contact, and slip) and four different materials (i.e., paper, foam, plastic, and metal) are considered.

For tactile data collection, a Shadow Dexterous Hand equipped with BioTac sensors on its fingers was used. We used the three fingers (i.e., the thumb finger, first finger, and middle finger) of the Shadow Dexterous Hand for data collection. The tactile data was recorded from the BioTac sensor mounted on these three fingers. Twelve household objects were selected and divided into four groups according to their different materials, as shown in Figure 6.7. The data collection was performed to record the tactile data across all the selected objects under three different contact events. We control these objects manually to interact with the fingertip of the Shadow Dexterous Hand. The tactile data are recorded at 100Hz and saved into the ROS bag files. The recorded tactile readings have 23 dimension tactile features. As introduced in Section 6.3, the tactile sequence t_{seq} with a dimension $l * 23$ is extracted from the tactile reading and taken as the sample of the tactile dataset. Meanwhile, all the tactile sequences t_{seq} are labeled with the contact event (0: non-contact, 1: contact, 2: slip) and the material class (0: paper, 1: foam, 2: plastic, and 3: metal).

The training of the proposed online detection module is performed as follows: this work first used the proposed tactile dataset to train the LSTM-based encoder-decoder. Then training of the encoder-decoder only requires the samples contained in the tactile data set since the training process is in an unsupervised learning setting. The trained encoder-decoder is used to compute the latent represents of all the samples (i.e., the t_{img}). In the encoder-decoder, the dimension of the representation of the samples is chosen as 256. Next, exploiting the latent representation of the samples and their corresponding labels, we trained the slip classifier and material classifier jointly. The training parameters of the online detection module are set as follows: the number of epochs was

[1] https://tams.informatik.uni-hamburg.de/people/deng/index.php?content=research



(a) paper (b) foam

(c)plastic (d) metal

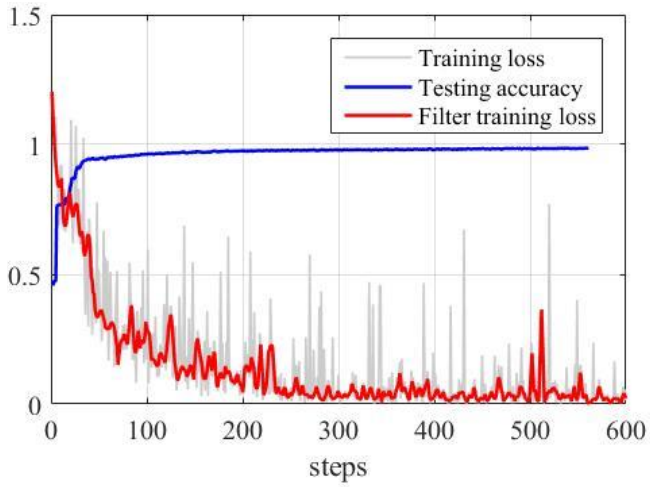
Figure 6.7: Twelve household objects contained in the tactile dataset. These objects are divided into four groups according to their different material.

10, and the batch size was 5. The learning rate was set as 0.00001. We use an Adam optimization method to optimize these two classifiers.

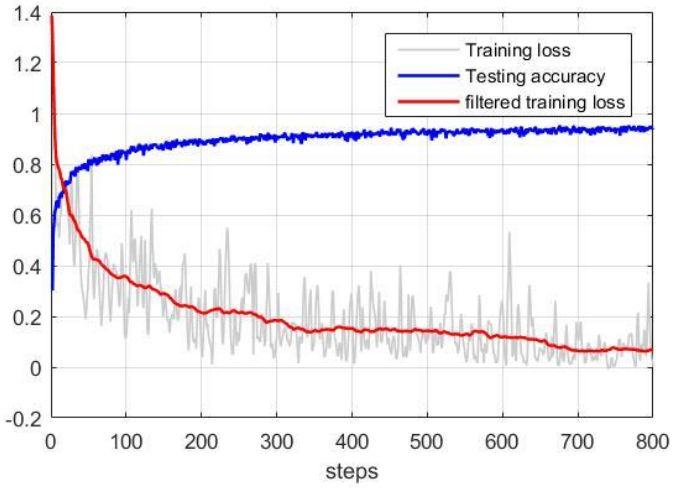
Experimental Results

The performance of the proposed online detection module is first evaluated based on the proposed tactile dataset. The tactile dataset was split randomly into a training set (90%) and a testing set (10%). This work uses the training loss and the test accuracy for this evaluation. Figure 6.8 illustrates the training loss and testing accuracy of the slip and material detection. In term of slip detection, the training loss is reduced as the training steps increase and converge a minimum. The slip classifier achieves a high accuracy of about 98% at final. That means the slip classifier can correctly classify the samples in the tactile data. The high accuracy of the slip detection is of critical importance to the object stabilization since the accurate prediction of slippage guides the force feedback control. In term of material detection, the final detection accuracy is about 95%. The proposed material classifier predicts the right material basically for the samples in the tactile data.

We make use of a confusion matrix to evaluate the overall quality of the proposed detection models. Figure 6.9 shows the two fusion matrix that visualizes the performance of the proposed two classifiers. In each confusion matrix, each row of the confusion matrix shows that the predicted probabilities for each ground-truth label. In term of slip detection, the proposed slip classifier could predict the contact event correctly since the diagonal elements have the highest values. The three contact events can be easily clas-



(a) slip detection

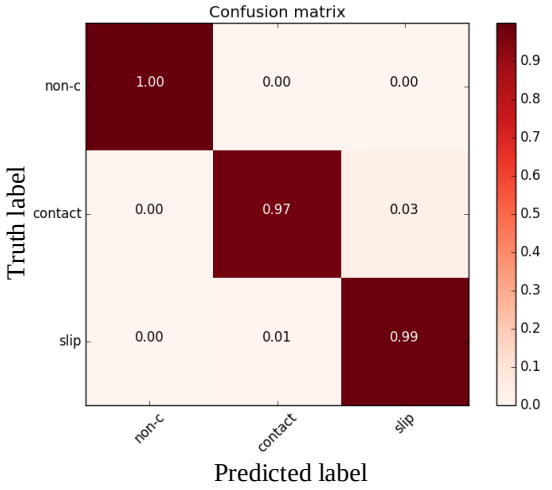


(b) material detection

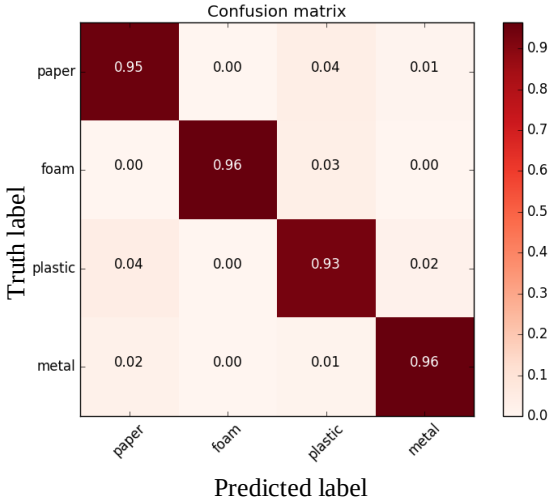
Figure 6.8: Performance of the proposed slip and material detection models.

sified. The proposed material classifier is also able to predict correct material for the object since the diagonal element has higher values than that of the other element in a small row. The foam is the most easily classified material. It is worth mentioning that several off-diagonal elements also have rather high values. For example, the prediction results of the plastic material show a relatively high probability for the paper. In this case, the proposed model may incorrectly predict both object materials (i.e., the paper and the plastic). There is a confusion between the plastic and paper maybe because the two material has similar physical properties, like the hardness.

Next, we analyse the sensitivity of the proposed online detection module concerning two important parameters (i.e., the window size l and the sample rate f). As introduced in Section 6.3, this work sample the tactile sequence t_{seq} with a fixed window size l and sampling rate f from tactile readings for detection. The window size determines the



(a) slip detection



(b) material detection

Figure 6.9: Confusion matrices of slip detection (a) and material detection (b).

number of sequential sensor readings that are taken as the input of the detection module. The sampling rate determines how fast the sensor readings are taken. We evaluate how the performance of the detected module changes with respect to the two parameters. In this evaluation, three different window sizes $l = \{16, 32, 48\}$ and three different sampling rates $f = \{50, 33.3, 25\}$ Hz are selected. That means the tactile data is sampled every 2, 3, or 4 points to from the t_{seq}. Before the sensitivity analysis, we first construct different tactile datasets by using the different combinations with the two parameters. The proposed detection model is then trained on these constructed datasets. The detection accuracy is used as the performance metric. Table 6.1 shows the detection accuracy obtained from these comparison experiments concerning different parameters. First, it can be seen that the accuracy of the slip detection and material detection are improved as increasing the window size l and keeping a fixed sampling rate. The highest accuracy is obtained when $l = 48$. That may be because the bigger window size contains

Table 6.1: Performance of the proposed detection model with respect to different parameters.

	Sampling rate		
Window size	50	33.3	25
16	96.9%	96.0%	97.6%
32	98.6%	98.6%	97.4%
48	99.0%	99.0%	98.0%

(a) Slip detection

	Sampling rate		
Window size	50	33.3	25
16	90.2%	90.7%	92.0%
32	94.5%	95.0%	97.2%
48	94.5%	97.2%	93.2%

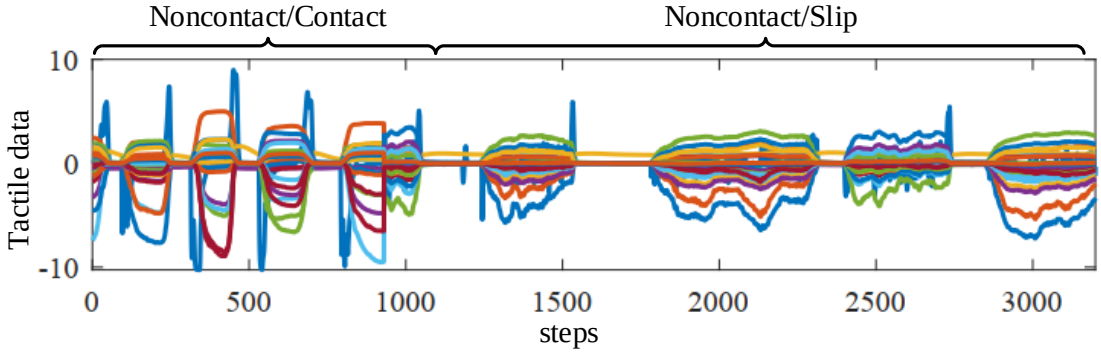
(b) Material detection

more tactile information that promotes detection. Second, the detection accuracy does not improve when the sampling rate increases. When $f = 33.3$ Hz, the detection performance is better than that of the other two sampling rates. At final, the highest slip detection accuracy of about 99% is obtained when the window size $l = 48$ and sampling rate $f = 50$ Hz or $f = 33.3$ Hz. The highest material detection accuracy of about 97.2% is obtained when the $l = 32$ and $f = 25$ or the $l = 48$ and $f = 33.3$. Since the tactile data is sampled at 100 Hz in this work, the tactile sequence t_{seq} has a 160 ms, 320 ms, and 480ms of consecutive reading under the window size is 16, 32, 48. As the window size increase, the detection delay is increased. To trade-off the real-time of detection and the high accuracy of detection, we chose the window size $l = 32$ and the sample rate $f = 33.3$Hz for the following real-world experiments.

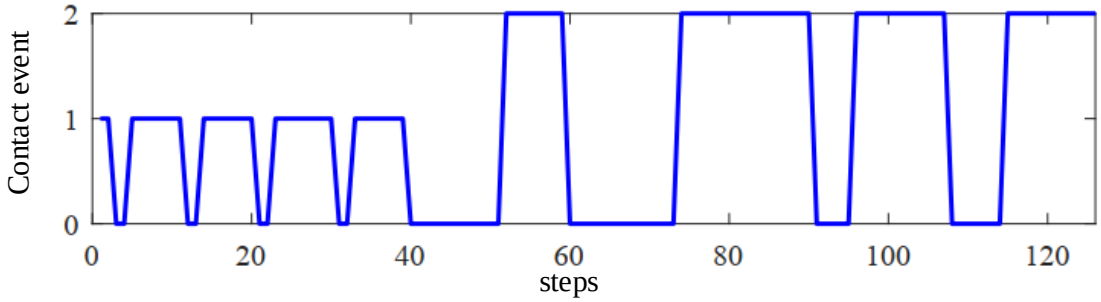
We further evaluate the performance of the proposed detection model by comparing the proposed method with a set of machine learning methods. In this work, we use linear SVM, Random forest and a multi-layer perceptron (MLP) for this comparison. These machine learning methods have been widely used to solve classification problems [22, 171]. The three machine learning methods are trained with the proposed tactile data which is constructed under the window size $l = 32$ and the sampling rate $f = 33.3$ Hz. The detection accuracy is used as the performance metric. Table 6.2 shows the detection accuracy of those methods. It can be seen that the proposed detection module achieves a high slip detection accuracy of 98.6% compared with the other three methods. The highest material detection accuracy of 95% was achieved by the proposed method. From the comparison results, we can find that the proposed method obtained a better performance on slip and material detection. Moreover, the proposed detection model can detect the contact event and object material simultaneously from tactile data.

Table 6.2: A comparison of the object and material detection accuracy from different methods.

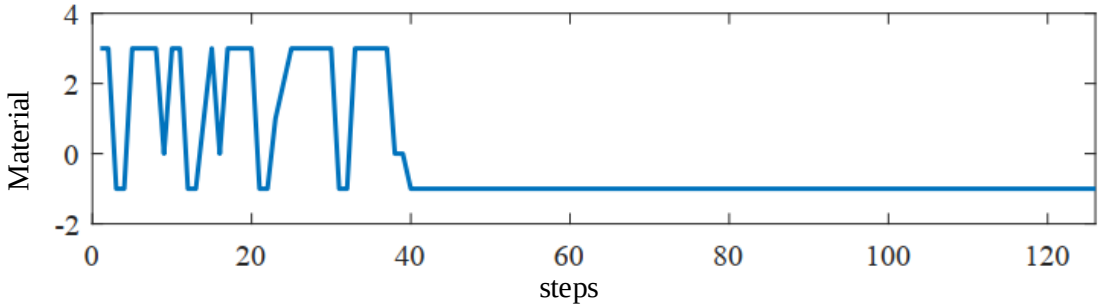
Method	Slip detection	Material detection
Linear SVM	98.1%	67.0%
Random forest	96.2%	59.2%
MLP	99%	78.4%
Ours	98.6%	95.0%



(a) Tactile reading



(b) Results of online slip detection



(c) Results of online material detection

Figure 6.10: Examples of online slip detection and material detection using the tomato bottle. (a) The tactile reading is collected from the BioTac sensor. The first part of the tactile reading only consists of two contact events, i.e., the non-contact and contact. The second part is the slippage or non-contact. (b) and (c) show the results of slip detection and material detection, respectively.

Finally, we show an example where the proposed detection module is used to detect contact events and object materials online. In this experiment, the tomato bottle is controlled to interact with the thumb finger of the Shadow Dexterous Hand to produce tactile streaming. During the online detection process, the tactile sequence is continually sampled from the tactile readings and is sent to the proposed online detection module. The proposed module outputs the predicted results. Figure 6.10 shows the tactile readings and the detection results. First, Figure 6.10(a) shows the tactile readings recorded from the BioTac sensor. The truth label of the contact event is shown in Figure 6.10(a) and the truth label of the object material is 3, i.e., the metal. Figure 6.10(b) illustrates the results of slip detection. It can be seen that the proposed slip detection method predicts a correct contact event at each step. It is worth mentioning that there is a delay in the slip detection. The degree of delay is affected by the window size and sampling rate. The bigger window size and lower sampling rate would increase the delay of slip detection. At final, the work performs material detection only when the slip classifier predicts the contact between the hand and the object. Otherwise, the material detection model output a value (-1). Figure 6.10(c) shows the results of the material detection. The times that the object is predicted as the metal is highest. Hence, the proposed detection model can correctly classify the object. The proposed detection model may predict the wrong material for the object since the effect of the sensor noise. A good performance of online detection is of critical importance for online object stabilization. The proposed detection module is implemented in python and runs on a 2.50GHz Intel i5 CPU.

6.6.2 Evaluation of Force Estimation

The force estimation method is also evaluated. The force estimated method is used to compute the contact information (i.e., normal contact force and contact location) from tactile data. Figure 6.11 illustrates the results of force estimation. In this evaluation, we control a object manually to contact the fingertip of the shadow hand in order to produce tactile data. Figure 6.11(a) shows the tactile reading sampling from the BioTac Sensor on the fingertip. From the Figure 6.11(b), it can be seen that the normal contact force clearly changes during the robotic hand contacts the object. The contact location is also computed based on tactile data, as shown in Figure 6.11(c,d). It is worth to mention that the estimated normal contact force describes the change of the contact state between the hand and the object.

6.6.3 Real-world Object Stabilization Experiment

We use the Shadow Dexterous Hand with five fingers for this evaluation. BioTac sensors were mounted on the fingertips of the Shadow Hand. Eight different objects are used, as shown in Figure 6.12. Each joint of the hand has its independent low-level torque controller. In this experiment, we considered grasp configuration involving three fingers (i.e., the thumb finger, the first finger, and the middle finger) of the hand across all the test objects. The BioTac sensor on each finger can provide tactile data when the hand contacts an object. The goal of object stabilization control is to stable grasp an object under physical uncertainties. Hence, in the object stabilization experiment, we

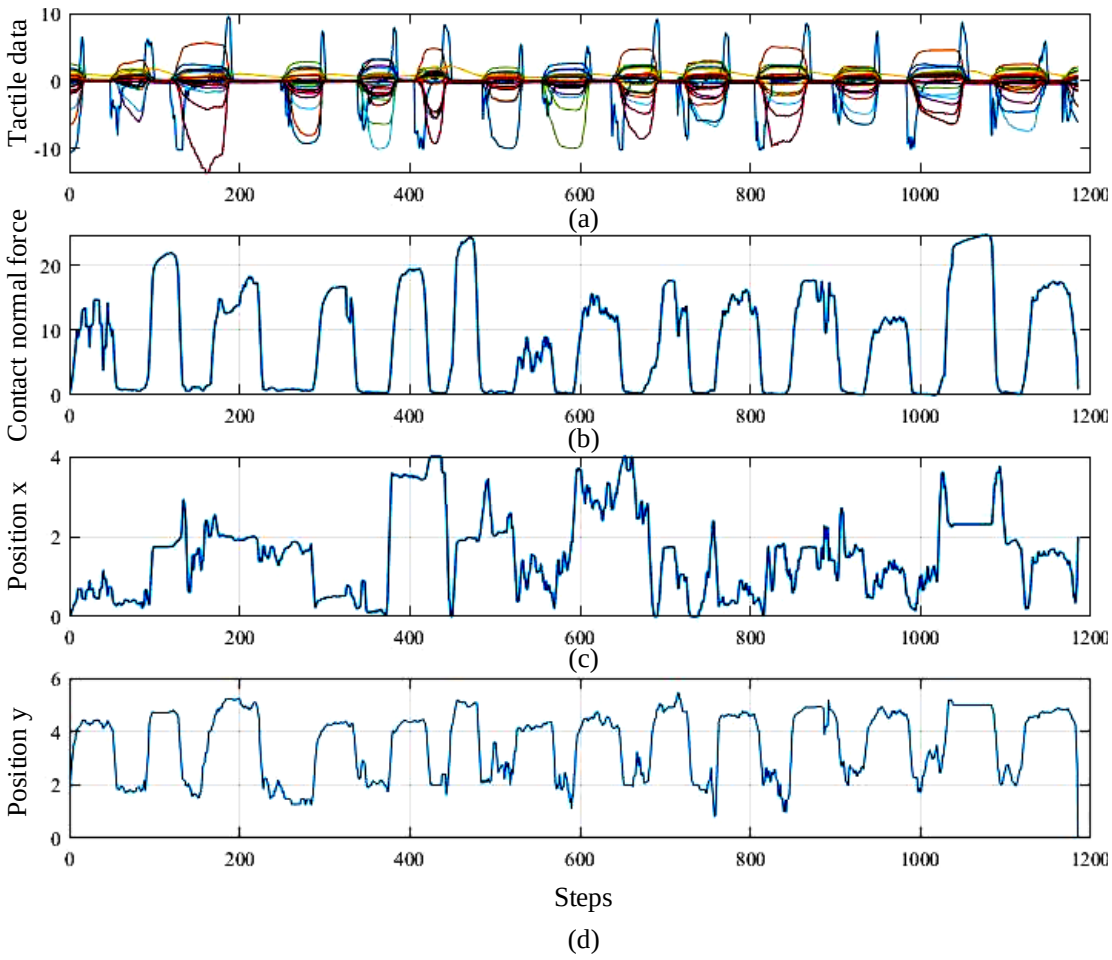


Figure 6.11: Results of force estimation

adopted the following procedures. (1) The robotic hand in an open-hand configuration is controlled under position control to reach a specified position. (2) Once contact between a finger and the object is detected by the online detection module, the object stabilization controllers were activated.

In a real-world application, a robotic hand may fail to grasp and hold an object by using a planned configuration due to the slippage and unknown weight of the object. Hence, we first validate the proposed stabilization strategy on the real robotic hand that implements object grasping tasks. The eight objects have different materials and weights. The Shadow hand is controlled to grasp and hold an object positioned by a human. The grasp configuration with three contact points on the object surface was not enforced. We recorded ten trails for the eight objects. A grasp was considered stable if the object was not dropped. The proposed stabilization strategy was able to stabilize all the eight objects, as shown in Figure 6.12. By using the proposed stabilization strategy, the contact force between the fingertip and the object was online adjusted to ensure the grasp stable. The experimental results demonstrated that the proposed stabilization strategies allow a multi-fingered robotic hand to stably grasp unknown objects without prior knowledge of their weight and physical properties.

Figure 6.12: Object grasping with the Shadow hand. The three fingers (i.e., the thumb finger, the first finger and the middle finger) are used.

6.7 Discussion

This paper aims to explore an object stabilization strategy that combines tactile sensing technique with feedback control to ensure a stable grasp with a multi-fingered robotic hand. Object stabilization with a robotic hand is essential for a robot that implements a manipulation task. However, it is still a challenge owing to the uncertainties arising from the objects and environments. This work takes advantage of tactile information to address the contact event detection, material detection and force estimation problems jointly. In this way, the uncertainties during object stabilization are reduced. The robotic hand can update the desired contact force automatically and track it. There has been some previous work that uses tactile data to predict the object material [27] and contact event [167]. In contrast to these previous methods, we trained a unified detection model to detect contact event and object material simultaneously from tactile data. Moreover, the use of the low-dimensional latent representation of the tactile sequence in the detection model helps to reduce the model complexity and enables different classifiers to share the tactile feature. In real-world robotic applications, it is of critical importance to build a unified detection model instead of multiple separate detection models for sensory perception.

Grasp stability under external perturbations can be maintained by adjusting the contact force [36] or updating the contact point of the fingers on the object surface [104]. In this work, we consider adjusting the contact force for object stabilization, because a tactile sensor provides local contact information. In the future, it will be beneficial to simultaneously adjust the contact point and force of the fingers for the stable grasp. In this case, the global and local information of the objects and environment is required.

Although tactile sensors can provide critical information about the interactions between the robotic hand and the objects, tactile sensors only perceive local contact information. Moreover, tactile information processing must be implemented in real-time to enable fast responses. Visual sensors are one of the most mainstream sensors in the robotic community. Visual sensors can provide global information about the target object and the environments. For example, the visual data can be used for grasp adaptation. Therefore, it is interesting to fuse visual and tactile modalities for robotic perception.

6.8 Conclusion and Future Work

In this work, we address the tactile sensing problem during object stabilization using a multi-fingered grasping considering dynamic uncertainties. We propose an object stabilization method for the multi-fingered robotic hand to perceive an unknown object online. The proposed object stabilization frameworks consist of three components: an online detection module, a force estimation approach and a force feedback controller. Our method takes advantage of the spatio-temporal characteristics of tactile data to perceive the interaction with the grasped object. We use a deep learning technique to learn features from tactile data for detection. The detection module samples a tactile sequence online from the tactile readings as input and predicts the object material and contact event simultaneously. A force estimation method exploiting the spatial property of tactile data is proposed to extract the contact information (i.e., normal contact force and contact location). A feedback controller is employed to adjust the grasp force to enable grasp stability. The effectiveness of the proposed framework was evaluated with a Shadow Dexterous Hand equipped with BioTac sensors.

Tactile sensors only perceive local contact information. Visual sensors can provide global information with respect to objects. However, the effectiveness of visual perception is easily affected by lighting conditions or occlusion. Therefore, it is necessary to explore novel methods to fuse visual and tactile information to improve perception capability in the future.

Chapter 7

Learning Synergies-based In-hand Manipulation with Reward Shaping

7.1 Introduction

Implementing in-hand manipulations is an important ability for multi-fingered robotic hands. To this end, a robotic hand uses its fingers to hold and manipulate an object within the hand. Owing to the high dimensionality of a robotic hand and intermittent contact dynamics, effectively programming in-hand manipulation for a robotic hand is still a challenging problem [130]. Many methods, such as trajectory optimization [122] and learning from demonstration [59, 138, 179], have been proposed to address this problem. The performance of these previous methods heavily relies on accurate dynamic models or high-quality human demonstrations that are difficult to obtain. Recently, DRL algorithms have achieved state-of-the-art performance in a set of continuous control tasks [109, 42]. DRL has the potential for learning manipulations directly from interactions with the environment. However, the sample complexity of DRL has become a key issue in learning complex manipulations [161], such as in-hand manipulation with a multi-fingered robotic hand.

In-hand manipulation can be understood as a process in which a robotic hand takes an action sequence using its fingers to move an object from a start pose to a goal pose. Palli et al. proposed understanding in-hand manipulation as a derivation of a reference grasping posture [133]. In robotics, the grasping postures of robotic hands are typically divided into two grasp types, i.e., *power* and *precision* grasps, which were first introduced by Napier et al. [125]. Grasp type is a way of representing the manner in which a hand handles objects. *Power* grasps use the fingers and palm to hold an object firmly, and *precision* grasps only use the fingertips to stabilize an object. Feix et al. extended Napier's grasp taxonomy and introduced a novel grasp taxonomy in which 33 different grasp types are presented [28]. Humans are naturally capable of operating objects by choosing a feasible grasp type from multiple possible grasp types. Based on the analyses of grasping behavior [133, 46], some works have used the *precision* grasp as guidance to plan in-hand manipulation [118, 129]. In these works, the *precision* grasp is first sought in the configuration space of a robotic hand and then used as guidance for the plan-

ning and control of in-hand manipulation. Using the *precision* grasp as guidance helps to reduce the sample complexity of manipulation planning. In this work, we exploit grasp types as guidance to assist in-hand manipulation learning with DRL. In contrast to previous work that requires high-quality human demonstrations [95, 59], this work only requires some grasping postures obtained from human demonstrations. Moreover, grasp type is exploited as guidance for in-hand manipulation learning. In this way, the robot is encouraged to explore a specific posture subspace, which reduces the sample complexity of the DRL.

DRL algorithms learn manipulations in the context of rewards received from interactions with the environment. Reward functions play a central role in specifying how agents should act. However, the learning of complex manipulations with DRL algorithms is usually slow and unstable when it only uses an extrinsic reward. Reward shaping methods [98] have been widely proposed to assist manipulation learning by introducing additional reward functions to augment the extrinsic reward. Various additional reward functions have been previously introduced according to different learning objectives (such as exploiting human demonstrations [135], human advice [26], or intrinsic motivation [68]). In this work, the agent is expected to achieve two sub-objectives: (1) exploring a grasping posture subspace under a specific grasp type (such as *precision* grasp) instead of the whole configuration space of the robotic hand, and (2) fully exploring this specific posture subspace to avoid local convergence. Accordingly, two additional reward functions are designed. The additional rewards help to not only improve the exploration efficiency of DRL but also provide a way to integrate domain-specific concepts (i.e., the grasp type) into DRL. Optimizing a control policy concerning multiple rewards is a multi-objective optimization problem. In contrast to previous methods that directly train a control policy with respect to a composite reward [135, 26], this work proposes a MARL algorithm to scalarize the multi-objective optimization problem [141].

The objective of this work is to allow a multi-fingered robotic hand to learn in-hand manipulation effectively. To this end, we propose a MARL algorithm that exploits grasp types as guidance for in-hand manipulation learning. To exploit grasp types as guidance for manipulation learning, we first construct a low-dimensional posture subspace based on the analysis of hand synergies considering six different grasp types. Given the hand posture subspace, the robotic hand is encouraged to explore a specific posture subspace fully under a specific grasp type. To realize this goal, we designs three different rewards, i.e., an extrinsic reward r^{ext}, a hand-based reward r^{hand} and an uncertainty-based reward r^{unc}. The r^{hand} is defined based on the analysis of hand synergies, and the r^{unc} is computed based on the uncertainty of the state prediction. The three reward functions allow the robot to know more about its environment and task. Meanwhile, three independent agents are trained jointly with respect to their rewards and then cooperate to optimize a control policy. Training each value agent with respect to a single reward helps the policy model to learn faster and better. The experimental results demonstrated that exploiting a specific grasp type as guidance helps to improve the exploration efficiency of DRL and increases the success rate of the task execution.

The rest of this section is organized as follows. Section 7.2 introduces the necessary background of the DRL. Section 7.3 introduces the construction of hand posture space

considering six different grasp types. The proposed MARL algorithm is presented in Section 7.4. The experimental evaluations and results are shown in Section 7.5. Finally, the conclusion and future works are presented in Section 7.7.

7.2 Background

In this section, we briefly introduced preliminaries for RL. A RL setting is typically modeled as a Markov Decision Process (MDP) given by a tuple: $M = \{S, A, R, P, \rho_0\}$. $S \in R^n$ and $A \in R^m$ denote the state and action, respectively. $R : s \times a \rightarrow r$ is a reward function. $P : s \times a \rightarrow s'$ is a transition dynamics model that outputs the probability distribution of the next state s' given the current state s and action a. ρ_0 is the initial probability distribution of the state. At the time step t, the agent executes an action a_t given a current state s_t. Then the agent observes the next state s_{t+1} and receives a reward r_t. The objective of the RL is to optimize a control policy $\pi : s \rightarrow a$ by maximizing an expected discounted return, as defined in Eq. 7.1. Where $\tau = \{s_0, a_0, ..., s_T, a_T\}$ denotes a rollout data obtained from the interaction with an environment.

$$\pi^* \leftarrow \max_{\theta_\pi} E_{\tau \sim \pi}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)] \tag{7.1}$$

RL defines a state-value $V(s)$, a Q-value $Q(s,a)$ and an advantage $A(s,a)$ function as follows:

$$V(s) = E_{\tau \sim \pi}[\sum_{t=0}^{T} \gamma^t r_t | s_0 = s]$$

$$Q(s,a) = E_{\tau \sim \pi}[\sum_{t=0}^{T} \gamma^t r_t | s_0 = s, a_0 = a]$$

$$A(s,a) = Q(s,a) - V(s)$$

In this work, we use Proximal Policy Optimization (PPO) algorithm [151], a state-of-the-art policy gradient method, as a baseline to learn in-hand manipulation. The PPO algorithm maintains two functions: a value function $V_{\theta_v}(s)$ and a policy function $\pi_{\theta_\pi}(s)$. Policy gradient methods typically suffer from catastrophically large updates. To stably update control policies, the PPO limits the magnitude of updates to the policy weight θ_π by imposing constrains on the difference between the new policy π_θ and old policy $\pi_{\theta_{old}}$. The policy function is optimized by maximizing the clipped surrogate objective as defined in Eq. 7.2. The objective constructs a trust region around the old policy $\pi_{\theta_{old}}$ by posing a lower bound on the improvement induced by an update.

$$L^{clip}(\theta_\pi) = \hat{E}_t[min(r_t(\theta_\pi)\hat{A}_t, clip(r_t(\theta_\pi), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{7.2}$$

where $\hat{E}_t$ denotes the empirical expectation over time steps. $\hat{A}_t$ represents the estimated advantage at time step t. ε is a clip parameter which clips the estimated advantage $\hat{A}_t$. $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}(a_t|s_t)}}$ is a ratio between the probability of action under current policy π_θ

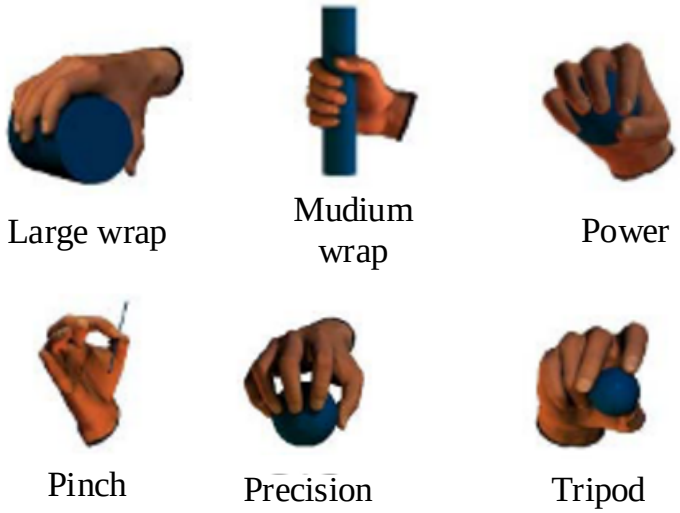


Figure 7.1: Six commonly used grasp types [28] for the construction of hand posture space.

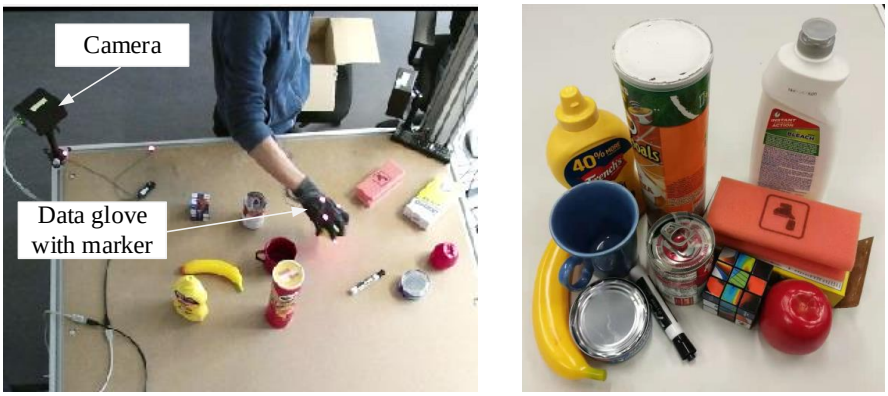
and the probability of the action under previous policy $\pi_{\theta_{old}}$.

7.3 Construction of Low-dimension Hand Posture Space

Humans are capable of operating objects by choosing a feasible grasp type from multiple possible grasp types. Grasp type is a way to represent how a hand handles objects. Once a feasible grasp type is selected, the robotic hand can make sense of how to used its fingers to operate the object. Feix et al. [28] introduced a GRASP taxonomy in which 33 different grasp types used by humans are presented. Considering the kinematic limitations of the robotic hand as well as Feix's GRASP taxonomy, this work consider six commonly used grasp types (i.e., *large wrap*, *small wrap*, *power*, *pinch*, *precision* and *tripod*), as shown in Figure 7.1. *Precision* grasp is defined as the operation of a object using fingertip contacts alone. Moreover, this work exploit the *precision* grasp type as guidance for in-hand manipulation learning. Hence, the robotic hand is encouraged to explore the hand posture under the *precision* grasp type in the configuration space of the robotic hand.

To construct the hand posture space, we need to collect a set of grasping postures, considering the six different grasp types. During data collection, a user is supposed to perform object grasping tasks considering the six grasp types. This work utilizes a tracking system to record human grasping posture, as shown in Figure 7.2(a). The tracking system uses a data glove with active markers to record 3D positions of the markers. These 3D marker positions are used to compute the grasping posture of the human hand. Twelve household objects are selected to collect the grasping postures, as shown in Figure 7.2(b).

After data collection, the grasping postures of the human hand is further mapped to a robotic hand. In this work, we use a Shadow Dexterous Hand, i.e., a five-fingered



(a) Collection of human grasping data (b) Twelve objects for data collection

Figure 7.2: Collection of human grasping data. (a) Data collection with a tracking system. (b) Twelve household objects

robotic hand, for in-hand manipulation experiments. The mapping process is achieved by using the inverse kinematic method from [147]. The Shadow Dexterous Hand has 20 actuated degrees of freedom and a further four under-actuated movements for a total of 24 joints. Hence, the whole configuration space of the Shadow Hand for in-hand manipulation is a high dimension. Studies from neuroscience suggested that the central nerves system (CNS) adopts a simplified strategy to coordinate a large number of degrees of freedom in motor control [37]. Synergies have been introduced as a strategy to understand the control of movement involving multiple degrees of freedom. In the past decade, synergies have also been widely used to analyze the behavior of the human hand [114] and design grasp planning methods [29]. Ciocarlie et al. [29] used Principal Components Analysis (PCA) algorithm to process hand poses recorded during human hand grasping. This work demonstrated that the first two principal components contain more than 80% of hand posture information. This work exploits the concept of synergies to construct a low-dimension hand posture space for a robotic hand.

This work also employs PCA to analyze the hand posture synergies and build the hand posture space for the Shadow hand. The PCA algorithm is employed to compute the principal components for all the mapped hand postures. This work defines the hand posture of the robotic hand as a vector of joint angles, i.e., $c = \{\theta_i\}_{i=1:24}$. Using the PCA, a hand posture c is expressed as a linear combination of uncorrelated variables $\{e_i\}_{i=1:n}$, as defined in Eq. 7.3. Here, we use the first two principal components to form the hand posture space, i.e., $n = 2$. As a result, the hand posture of Shadow Hand could be represented completely with the amplitude vector $A = [a_1, a_2]$.

$$c = c_m + \sum_{i=1}^{n} a_i e_i \tag{7.3}$$

where c_m denotes a nominal hand posture. $\{e_i\}_{i=1:n}$ are the principal components of the PCA algorithm, which defines the principal directions of variance. n is the number of principal components.

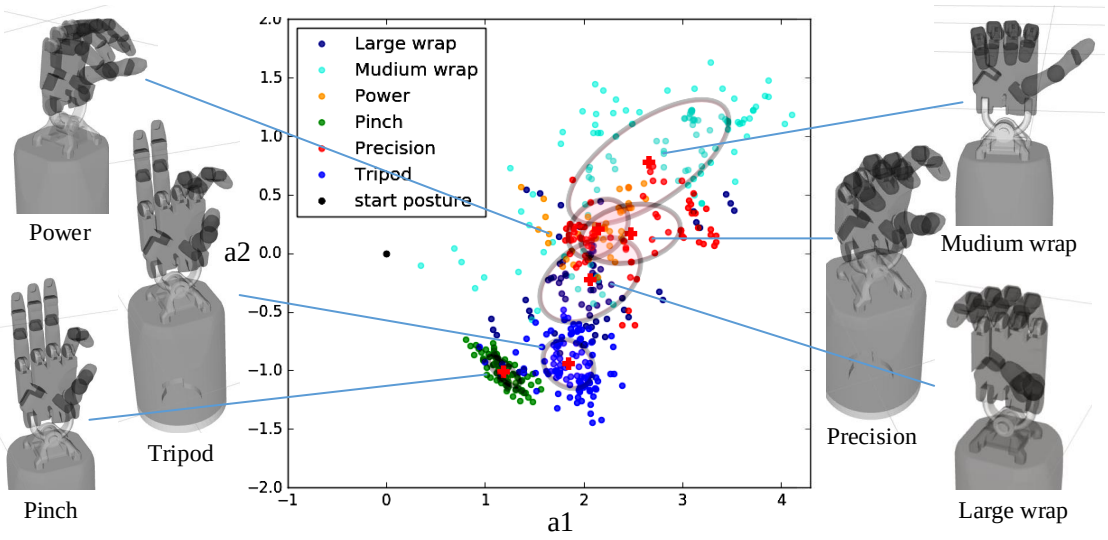


Figure 7.3: Low-dimension hand posture space under the six grasp types. The first two principal components (a_1, a_2) of the PCA is used to represent the posture space. The red cross denotes the cluster center of grasp posture under each grasp type.

At final, a low-dimension posture space of the robotic hand could be constructed through hand synergies. Figure 7.3 shows the low-dimension hand posture space considering six different grasp types. The start posture is an open-hand configuration. From this figure, it can be seen that the hand postures under different grasp types locate in the different subspace of the hand posture space. The hand postures under the same grasp type cluster together. For effective manipulation learning, the robotic hand is encouraged to explore the posture subspace under the *precision* grasp type.

7.4 Multi-agent Reinforcement Learning with Reward Shaping

This section presents the MARL algorithm exploiting a specific grasp type as guidance for a multi-fingered robotic hand to learn in-hand manipulation. Two additional reward functions are first introduced based on the analysis of hand posture synergies and its learning history. In-hand manipulation is then learned using a proposed MARL algorithm. Figure 7.4 shows the schematic diagram of the MARL algorithm.

7.4.1 Reward Shaping for Efficiency Exploration of DRL

This work takes advantage of the information extracted from both the analysis of hand posture synergies and the learning history for assisting manipulation learning. Two additional reward functions are designed to encode the extracted information. In this work, the proposed MARL algorithm use three different reward functions: (1) An extrinsic reward r^{ext} that specifies the task goal. (2) A hand-based reward r^{hand} that encourages the agent to explore the state subspace under a specific grasp type (such as *precision*). (3)

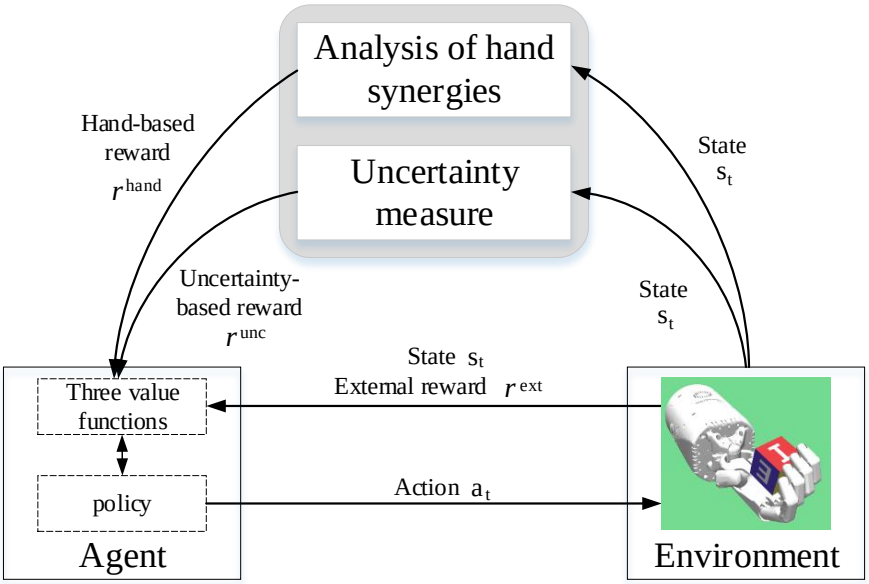


Figure 7.4: The schematic diagram of the proposed MARL algorithm.

An uncertainty-based reward r^{unc} that used to balance the trade-off between exploration and exploitation of the MARL algorithm. The extrinsic reward is generated according to the task requirement and the environment, which is defined in Section 7.5. The following subsections present the computation of the hand-based reward r^{hand} and the uncertainty-based reward r^{unc}.

$$r := \{r^{ext}, r^{hand}, r^{unc}\} \tag{7.4}$$

Hand-based Reward Based on the Analysis of Hand Synergies

This work exploits the *precision* grasp type as guidance for in-hand manipulation learning. Hand postures required for implementing in-hand manipulations are understood as a derivation from the *precision* grasp. As shown in Figure 7.3, the hand posture space is divided into different regions according to six different grasp types. The posture subspace under the *precision* grasp type is a partial region of the whole posture space. To reduce the sample complexity of the DRL, the robot is preferred to explore the specific posture subspace under the *precision* grasp type and ignore some infeasible states in other posture subspace (such as the subspace under the *pinch* grasp type) .

This work use similarity measure to define the hand-based reward r^{hand}, which drives the robot to explore the specific posture subspace under the *precision* grasp. The r^{hand} is defined based on the Euclidean distance d^{hand} between the amplitude vector A of the explored configuration c and the amplitude vector $A^{precision}_{center}$ of the reference grasp configuration $c^{precision}_{center}$, as shown in Eq.7.5. Where the $A^{precision}_{center}$ is obtained by clustering in the hand posture space, as shown in Figure 7.3. Eq. 7.5 shows the computation of the r^{hand}. We take the explored hand posture as a *precision* grasp if $d^{hand} < 0.4$ and set a

high value to encourage the agent to execute the *precision* grasps.

$$r^{hand} = \begin{cases} 1.5 & \text{if } d^{hand} < 0.4 \\ -\frac{d^{hand}-0.4}{1.4-0.4} & \text{if } 0.4 \leqslant d^{hand} \leqslant 1.4 \\ -10 & \text{otherwise} \end{cases} \\ \text{with} \quad d^{hand} = ||A - A^{precision}_{center}||^2_2, \quad A = [a_1, a_2] \tag{7.5}$$

Uncertainty-based Reward Based on Uncertainty Measure

The r^{hand} encourages the robot to explore a specific hand posture subspace under the *precision* grasp. At the same time, the agent is also preferred to explore this posture subspace fully to avoid a local convergence. Inadequate exploration could result in failing to find an effective control policy. Although this work adds random Gaussian noise to action selections, random exploration is not efficiency, particularly for learning complex manipulations. Different heuristic exploration approaches have been previously proposed to improve exploration efficiency. Intrinsic motivated RL [157, 68, 2] is a popular method that defines different intrinsic motivation rewards to improve the exploration efficiency of the DRL. This work proposes to measure the uncertainty of the explored state as an intrinsic motivation of the agent. The uncertainty measure guides the robot to explore new states.

The uncertainty of explored states is approximated by the prediction error of a transitional dynamic model. This work approximates the transitional dynamic model $f_{\theta_f}(s_t, a_t)$ by training a neural network, where the θ_f denotes the weights of the network. The dynamic model $f_{\theta_f}(s_t, a_t)$ takes current state s_t and action a_t as inputs and predicts the next state s_{t+1}, i.e., $f_{\theta_f} : s_t \times a_t \rightarrow \hat{s}_{t+1} \approx s_{t+1}$. The dynamic model f_{θ_f} is trained by minimize a mean squared error loss, as defined in Eq. 7.6. Where $||\cdot||_2$ is the L2 norm. D is a minibatch dataset received from the interactions with its environment.

$$L(\theta_f) = \frac{1}{|D|} \sum_{(s_t, a_t, s_{t+1}) \in D} ||s_{t+1} - f_{\theta_f}(s_t, a_t)||^2_2 \tag{7.6}$$

In this work, the uncertainty of the state is measured based on the prediction error of the dynamic model. At each time step t, given the collected state $s_t$and action a_t, the dynamic model f_{θ_f} predict the next state s_{t+1}. We measure the Euclidean distance d^{unc} between the explored next state s_{t+1} and the predicted next state $\hat{s}_{t+1}$ to estimate the state uncertainty. Eq. 7.7 shows the computation of the uncertainty-based reward r^{unc}. The learning history described by the transitional dynamic model is used to design the r^{unc}. The r^{unc} encourages the robot to try unexplored states with high prediction error of the dynamic model.

$$r^{unc} = \begin{cases} 1.5 & \text{if } d^{unc} < 0.2 \\ \frac{d_{unc}-0.2}{25-0.2} & \text{if } 0.2 \leqslant d^{unc} \leqslant 25 \\ -10 & \text{otherwise} \end{cases} \\ \text{with} \quad d^{unc} = ||s_{t+1} - f_{\theta_f}(s_t, a_t)||^2_2 \tag{7.7}$$

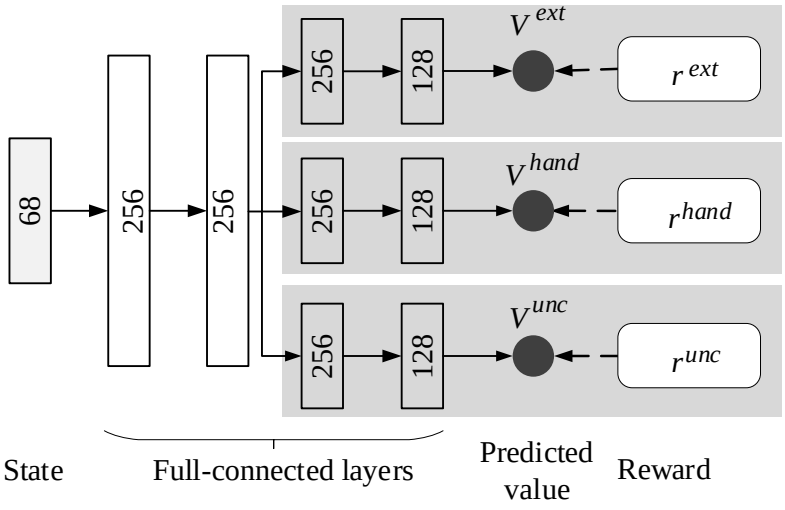


Figure 7.5: Network architecture of the value functions. Three independent functions are trained jointly concerning their reward function.

7.4.2 Multi-agent Reinforcement Learning with Multiple Rewards

Different rewards could drive the robot to achieve different learning behavior. Manipulation learning concerning multiple rewards is usually formulated as a multi-objective optimization problem. Different from previous methods that train the robot directly concerning a composition reward, this work formulates manipulation learning as a MARL problem. In the MARL setting, three independent agents represented by value functions are trained jointly concerning their reward, and then cooperate to optimize a control policy. In this way, each value agent captures different learning behavior concerning its reward. For example, a hand-based agent learns the formation of a *precision* grasp for in-hand manipulation; at the same time, and the uncertainty-based agent aims to try unexplored states to avoid local convergence.

In the MARL setting, according to the three rewards $r := \{r^{ext}, r^{hand}, r^{unc}\}$, this work jointly trains three agents represented by three value functions $V_{\theta_v} := \{V^{ext}, V^{hand}, V^{unc}\}$. The value function V_{θ_v} is approximated by training a deep neural network, where the θ_v denotes the weight of the network. The value function V_{θ_v} takes current state s_t and predicted the state value $v_t = \{v^{ext}, e^{hand}, v^{unc}\}$, i.e., $V_{\theta_v} : s_t \rightarrow \hat{v}_t \approx v_t$. In this work, each value function shares the low-level layers of neural network and owns a separate high-level layers. Figure 7.5 shows the network architecture of the value functions. The first two fully-connected layers is used to extract the state feature and the last two independent fully-connected layer is trained to achieve different learning behavior. The parameters $\theta_v := \{\theta_{v,0}, \theta_{v,ext}, \theta_{v,hand}, \theta_{v,unc}\}$ of the value function are optimized by minimizing the following objective that is a linear combination of the three loss functions concerning the three value functions.

$$\begin{aligned} L(\theta_v) &= L_v^{ext} + L_v^{hand} + L_v^{unc} \\ \text{with} \quad L^o &= E_\pi[||y^o - \hat{v}^o||_2^2], \quad o := \{ext, hand, unc\} \end{aligned} \tag{7.8}$$

where L^{ext}, L^{hand} and L^{unc} denote the loss functions of the three value agents, respectively. y^o denote the estimated value of a value agent computed with the TD-λ method. $\hat{v}^o$ is the predicted value of a value agent.

Then, the three value functions cooperate to compute the advantage A to optimize the control policy. The composition value V_{comp} is first computed based on the three value functions. The V_{comp} is a linear combination of the predicted values of the three value functions, as defined in Eq. 7.9. Where β_1, β_2 and β_3 denote the weights that determine the important of the value functions during manipulation training.

$$V_{comp}(s) = \beta_1 V^{ext}(s) + \beta_2 V^{hand}(s) + \beta_2 V^{unc}(s) \tag{7.9}$$

We also can demonstrate that the policy optimization with respect to the V_{comp} is consistent to the optimization with respect to the composition reward function $r_{comp} = \beta_1 r^{ext} + \beta_2 r^{hand} + \beta_3 r^{unc}$, as defined in Eq. 7.10. However, in the multi-agent setting, the multi-objective optimization of the value function is scalalized and the there value functions are easily trained.

$$\begin{aligned} V_{comp}(s) :&= \beta_1 V^{ext}(s) + \beta_2 V^{hand}(s) + \beta_2 V^{unc}(s) \\ &= \beta_1 E[\sum_{t=1}^{T} \gamma^t r^{ext}(s_t)] + \beta_2 E[\sum_{t=1}^{T} \gamma^t r^{hand}(s_t)] \\ &\quad + \beta_3 E[\sum_{t=1}^{T} \gamma^t r^{unc}(s_t)] \\ &= E[\sum_{t=1}^{T} \gamma^t (\beta_1 r^{ext}(s_t) + \beta_2 r^{hand}(s_t) + \beta_3 r^{unc}(s_t))] \\ &= E[\sum_{t=1}^{\infty} \gamma^t r_{comp}(s_t) | s_t = s] \end{aligned} \tag{7.10}$$

Next, this work uses the GAE-λ method to estimate the advantage A. Given the composition value V_{comp} and composition reward r_{comp}, the advantage A is estimated by

$$A_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{k-1} (\gamma\lambda)^l (r_{comp,t} + \lambda V_{comp}(s_{t+1}) - V_{comp}(s_t))$$

In the proposed MARL setting, the control policy π_{θ_π}, the value function V_{θ_v} and the transitional dynamic model f_{θ_f} are jointly optimized by minimize the following loss, as defined in Eq. 7.11. The $L^{clip}(\theta_\pi)$, $L(\theta_v)$ and $L(\theta_f)$ denote the loss function of the policy function, the value function and the dynamic model, respectively.

$$\min_{\theta_\pi, \theta_v, \theta_f} L^{clip}(\theta_\pi) + L(\theta_v) + L(\theta_f) \tag{7.11}$$

The pseudocode of the proposed MARL algorithm with reward shaping is shown in Algorithm 6. Initially, the agent collects rollout data by running the current policy π_{θ_π}. The two additional rewards (i.e., r^{hand} and r^{unc}) together the extrinsic reward r^{ext}

are computed at each time step. After collecting the data, the policy function, value function, and dynamic model are jointly updated with the sampled rollout data.

Algorithm 6 : Leaning synergies-based in-hand manipulation

Input: Number of epochs N, Length of epoch N_{epc}, Number of optimization steps N_{opt}
Output: policy $\pi_{\theta_\pi}(a|s)$
Initialize the agent state as $s_0 \sim \rho_0(s_0)$
for $i = 1$ to N **do**
  for $t = 1$ to N_{epc} **do** Training data collecting
    Sample and execute the action $a_t \sim \pi_{\theta_\pi}(a|s)$
    Receive the next state s_{t+1} and the r_t^{ext}
    Compute the r_t^{hand} using Eq. 7.5 and the r_t^{unc} using Eq. 7.7.
    Add the tuple $< s_t, a_t, s_{t+1}, r_t^{ext}, r_t^{hand}, r_t^{unc} >$ to replay memory M
    $t += 1$
  end for
  for $j = 1$ **to** N_{opt} **do** Optimization
    Sample a minibatch B from replay memory M
    Optimize the policy π_{θ_π}, value function V_{θ_v} and dynamic model f_{θ_f} using Eq. 7.11
    $j += 1$
  end for
  $i += 1$
end for

7.5 Experiments

This section first introduces the experiment setup and then presents the experiment results and its discussion.

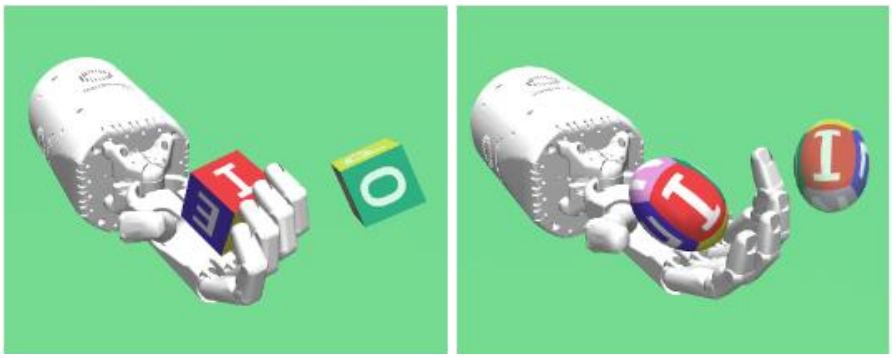
7.5.1 Implementation

Environments: To evaluate the proposed MARL algorithm, we used the robotic environments from OpenAI Gym [1]. The robotic environments are built based on the MuJoCo physic simulator [2], as shown in Figure 7.6. Object rotation tasks were used for the evaluation. In this experiment, two objects (i.e., a block and an egg) were used. A Shadow Dexterous Hand [3] was used to manipulate an object from an initial pose to a target pose by using its fingers. The Shadow Hand is a 24-DoF manipulator which five fingers with 22 joints and a wrist with two joints.

[1] https://gym.openai.com/envs/classic$_control$

[2] http://www.mujoco.org/

[3] https://www.shadowrobot.com/products/dexterous-hand/



a) Block rotation task b) Egg rotation task

Figure 7.6: Two object rotation tasks in the OpenAI Gym robotic environment. Two objects, i.e., a block and an egg, are used for the evaluation.

In the MARL setting, the state of the robotic hand is 68 dimension variable which includes the hand joint angles, the joint velocity, the object pose, the object velocity and the object target pose. The action was a 24-dimensional variable which was the relative angle of hand joints w.r.t. the current angle. The initial state of the agent and the target object pose were randomly chosen at the beginning of each episode. The objective is to manipulate the object to reach the desired orientation. Hence, the external reward r^{ext} was computed based on the difference d^{ext} between current object orientation and the target object orientation, as defined in Eq. 7.12. Where the object orientation was represented as a quaternion q. The $q_{curr} \ominus q_{target}$ denoted the different between the current quaternion q_{curr} and the target quaternion q_{target}. The two rotation tasks used the same r^{ext} which encourage the agent to manipulate an object to reach the target pose. We take that the object reaches the goal pose if $d^{ext} > -0.1$ and set a high value to encourage the robot to implement the object rotation tasks successfully.

$$r^{ext} = \begin{cases} 50 & \text{if } d^{ext} > -0.1 \\ -\frac{d^{ext}-(-0.1)}{-5-(-0.1)} & \text{if } -5 \leqslant d^{ext} \leqslant -0.1 \\ -10 & \text{Otherwise} \end{cases} \tag{7.12}$$

$$\text{with} \quad d^{ext} = -q_{curr} \ominus q_{target}$$

Training details: The control policy π_{θ_π} was also approximated by training a neural network, where the θ_π denotes the weight of the network. The policy π_{θ_π} taken the current state s_t and outputted the predicted action a_t, i.e., $\pi_{\theta_\pi} : s_t \rightarrow a_t$. The policy network used three fully-connected layers. Each layer had 256 hidden units and used the tanh activation function for all the hidden units. The training parameters of the proposed MARL algorithm were set as follows. The number of epochs N was 2500. The number of optimization steps N_{opt} was 10. The length of rollout $N_{rollout}$ was 2048. The length of each episode N_{epi} was 100. During the policy training, the batch size was set as 256. The initial learning rate was set as 0.0001, and it was linearly decreased as the learning process. The discount factor γ was 0.99, and the clip parameter ε was

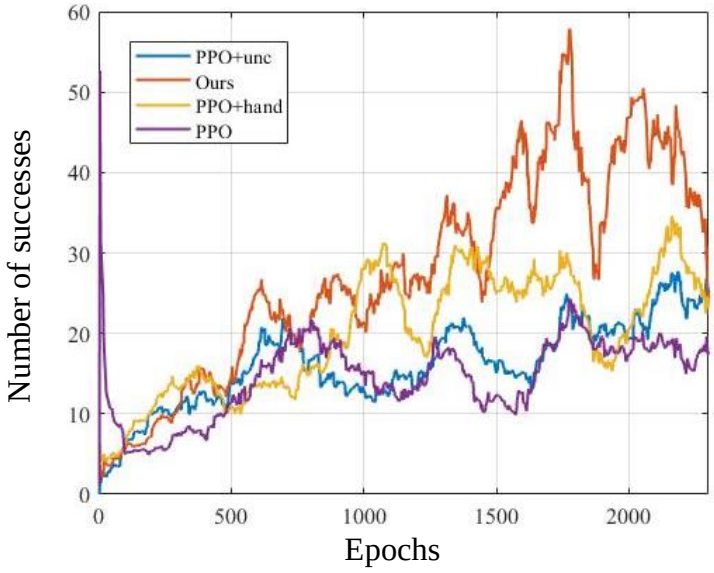
0.2. The two parameters of the GAE method are set as $\lambda = 0.95$ and $\gamma = 0.99$. The Stochastic Gradient Descent (SGD) method with a momentum rate of 0.9 was employed to optimize the weights of the neural networks. The three parameters were set as $\beta_1 = 1$, $\beta_2 = 0.5$ and $\beta_3 = 0.1$, which weighted the importance of the r^{ext}, r^{hand} and r^{unc}, respectively.

7.5.2 Experimental Results

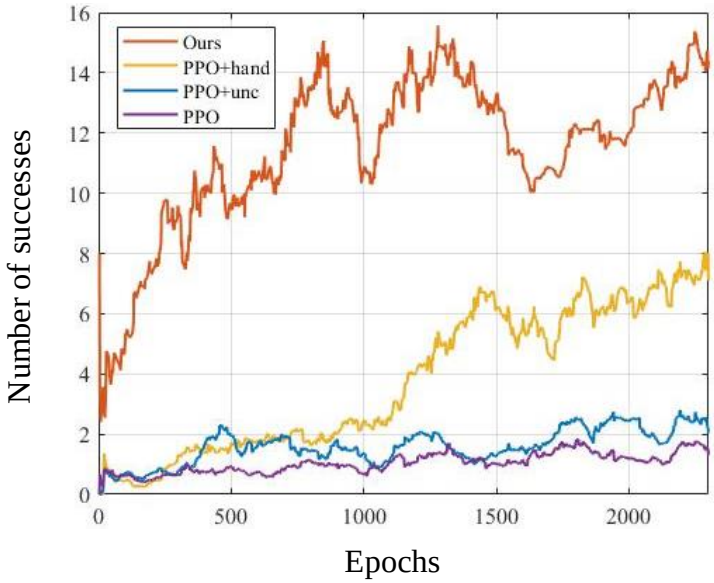
Firstly, the overall performance of the proposed MARL algorithm concerning object rotation tasks was evaluated. We compare our results with four categories of baselines, as defined as follows.

- PPO algorithm: The PPO algorithm was used as a baseline for this evaluation. This algorithm only used the extrinsic reward r^{ext} for policy optimization.
- ppo + *hand* algorithm: This algorithm used the hand-based reward r^{hand} and the extrinsic reward r^{ext}. Two independent value agents trained jointly and cooperated to optimize the control policy.
- ppo + *unc* algorithm: This algorithm used the uncertainty-based reward r^{unc} and the extrinsic reward r^{ext}. Two independent value agents trained jointly and cooperated to optimize the control policy.
- Our proposed MARL algorithm: This algorithm used all the three reward function.

The four algorithms were evaluated in the two rotation tasks mentioned above. Figure 7.7 shows the performance curve of the four algorithms in the object rotation tasks. The learning performance of these algorithms was evaluated through the number of successes in each rollout. The high number of successes meant that the robot has a high chance of achieving object rotation tasks successfully. Firstly, from the learning results, it can be seen that the PPO algorithm failed to obtain a high success rate in the block rotation task. The manipulation learning with the PPO is relatively slow. Secondly, it can be seen that the ppo + *hand* and ppo + *unc* algorithm performed better than the PPO algorithm because the two algorithms obtained a higher success rate. The two additional reward functions all helped to improve exploration efficiency of the PPO algorithm. Meanwhile, we also noticed that the learning speed of the ppo + *hand* algorithm was faster than that of the ppo + *unc* algorithm. The reason was that the hand-based reward r^{hand} drives the robot to use the feasible hand configurations for rotating the objects. Hence, the learning time required by the ppo + *hand* algorithm was reduced. Finally, compared with the three other algorithms, the proposed algorithm obtained the best learning performance because it obtained the highest success rate. The robot with the guidance of the r^{hand} had a high chance to try the states with high probability success. The action selection in the exploration process was adjusted according to the predicted uncertainty of the dynamic model. As a result, the learning agent explored new states to balance the trade-off between exploration and exploitation. The incorporating of the



(a) Block rotation task

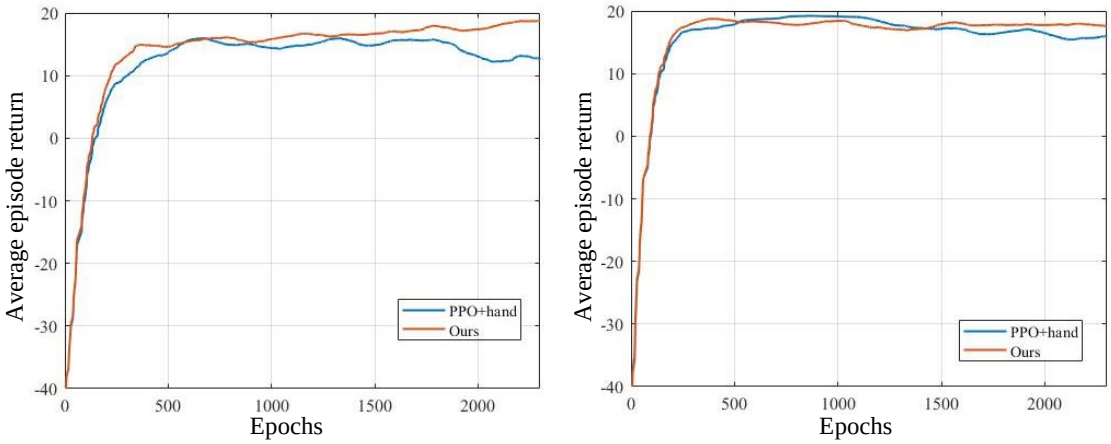


(b) Egg rotation task

Figure 7.7: A comparison of the results from four algorithms concerning object rotation tasks.

domain knowledge and information extracted from the learning history helped to reduce sample complexity and improve the exploration efficiency. The comparison results demonstrated that the proposed MARL algorithm enabled the Shadow Hand to learn in-hand manipulations effectively.

In this work, the *precision* grasp configuration was taken as guidance for in-hand manipulation learning. The hand-based reward r^{hand} is designed to encode the grasp type information to assist the manipulation learning. The average episode returns R^{hand}



(a) Block rotation task

(b) Egg rotation task

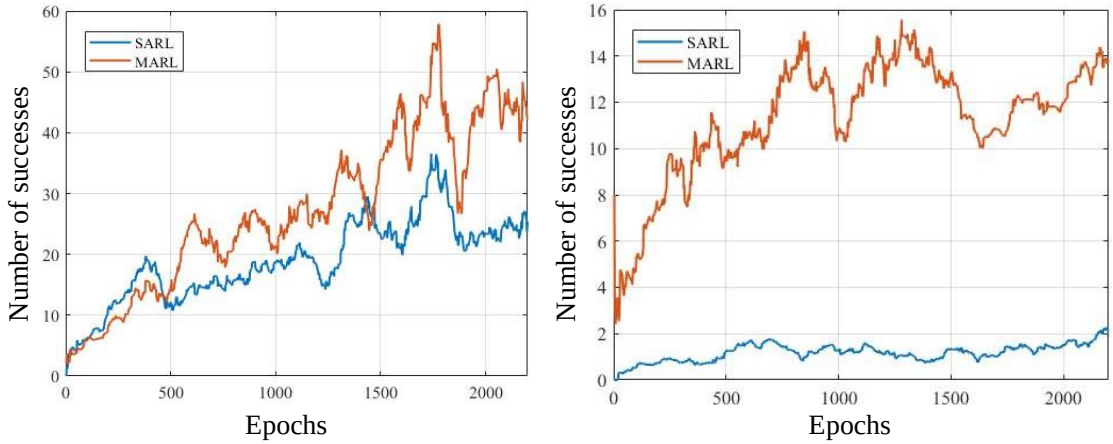
Figure 7.8: The collected average episode return concerning the hand-based reward during running the two algorithms.

concerning the r^{hand} could be used to evaluate the manipulation ability of the robotic hand. The high r^{hand} meant that the agent had a strong ability to form the *precision* hand configuration. Figure 7.8 shows the average episode return R^{hand} during running the two algorithms (i.e., the ppo + *hand* and the proposed algorithm). The return R^{hand} was increasing as the learning progress. That meant the robot was to form and use the *precision* hand configuration which produced a high success rate of object rotation. It can be seen that our proposed algorithm obtained a faster learning speed than the ppo + *hand* algorithm. It showed that the r^{hand} encouraged the agent to explore the state space where the states had a high hand-based reward opportunity. It also can understand that the use of domain knowledge helped the agent to quickly learn how to control its fingers to achieve the object rotation tasks successfully. Hence, given the hand-based reward r^{hand} as a guidance, the Shadow Hand had a high chance to find feasible hand configuration for in-hand manipulation.

Next, we further evaluated the effectiveness of the MARL setting for in-hand manipulation learning. In the proposed algorithm, three independent agents represented by value functions were jointly trained concerning their reward function and then cooperated to optimize the control policy. We want to demonstrate the necessity of learning independent value agents. This work compared the proposed algorithm with another baseline.

- Single Agent Reinforcement Learning (SARL) algorithm: This algorithm is also based on the PPO algorithm, but it used a composition reward that is a linear combination of the three rewards defined in Section 7.4. Previous methods mainly used a composition reward computed from multiple rewards to learn a control policy [87].

Figure 7.9 shows the performance curve of the two algorithms in the two object rotation tasks. The learning performance of the two algorithms was evaluated through the number of successes in each rollout. From the figure, it can be seen that the proposed al-



(a) Block rotation task (b) Egg rotation task

Figure 7.9: Comparison between the proposed MARL algorithm and the SARL algorithm.

gorithm obtained a clear performance boost over the SARL algorithm. The optimization concerning a composition reward was a multi-objective optimization problem. Training three independent value functions jointly helped to scalarize the multi-objective optimization. Each reward function typically represented an individual desirable learning behavior of the agent. Experimental results showed that the training of the agent (i.e., the optimization of the value function) concerning individual rewards could be fast.

7.6 Discussion

This work aims to explore a learning method that exploits meaning abstraction to assist manipulation learning (such as object rotation with a multi-fingered robotic hand). Manipulation learning using DRL algorithms is usually a time-consuming process. DRL algorithms in the context of an extrinsic reward are usually slow and unstable, especially when the extrinsic reward is sparse. Previous work has demonstrated that the use of domain knowledge helps to improve the performance of manipulation learning using DRL algorithms [7, 89]. However, it is still a challenging problem to represent and use domain knowledge in DRL algorithms. In this work, we exploited an abstraction of hand grasping posture (i.e., grasp type) as guidance to assist in-hand manipulation learning. Grasp types convey important information that represents the manner in which a hand handles objects. We extract information from both the analysis of hand posture synergies and the learning history. Two additional rewards were designed to encode the extracted information. The two additional reward functions encourage the robot to achieve multiple learning behaviors (such as the formation of a *precision* grasp and the execution of an unexplored state). A reward shaping method is used to integrate these rewards into a DRL algorithm. Domain knowledge then encourages the agent to attempt reasonable actions. The experimental results showed that the use of such domain knowledge helped the agent to improve the performance of in-hand manipulation learning.

7.7 Conclusion and Future Work

This work presents a MARL algorithm for a multi-fingered robotic hand to learn in-hand manipulation. Information extracted from both the analysis of hand posture synergies and the learning history is used to guide in-hand manipulation learning. Two additional reward functions (i.e., a hand-based reward and an uncertainty-based reward) are designed to encode the extracted information into a DRL algorithm. The hand-based reward drives the robot to explore the specific hand posture subspace with high success probability. The uncertainty-based reward encourages the robot to try unexplored states to balance the trade-off between exploration and exploitation. Meanwhile, three independent agents represented by value functions are trained jointly with respect to their reward and then cooperate to optimize a control policy. The two additional rewards not only improve the exploration efficiency of the DRL algorithm, but also provide a way to incorporate domain knowledge. The performance of the proposed MARL algorithm was evaluated with object rotation tasks. The experimental results showed that the proposed MARL algorithm enabled the robot to achieve the object rotation tasks effectively. Moreover, this work demonstrated that exploiting domain knowledge (such as grasp type) is powerful for learning in-hand manipulations with a multi-fingered robotic hand.

Effectively learning in-hand manipulation is far from a solved problem. In this work, the robot is allowed to explore all the states in the configuration space. However, this is impractical for a real physical robotic system, because some unreasonable state may break the system. Safety constraints should be considered in the DRL algorithm, which restricts the agent in exploring the reasonable states.

Chapter 8

Conclusion and Future Work

In this chapter, we first present a summary of the thesis. Furthermore, the advantages and limitations of the proposed approaches are discussed. Finally, we introduce some potential directions for future research.

8.1 Summary of the Thesis

The overall objective of this thesis was to empower a robot to achieve human-like grasping and manipulation abilities. Throughout this thesis, we proposed a variety of approaches for the robotic hands to implement dexterous grasping and manipulations. The development of these approaches aimed to address four open challenges relevant to dexterous grasping and manipulation. The four challenges were *stable grasping during manipulation tasks*, *generalization of grasp planning*, *stability of grasps under uncertainties* and *complexity of in-hand manipulation learning*. In relation to the four challenges, we summarize the thesis as follows.

First, we studied stable grasping when implementing manipulation tasks. RTG tasks were considered in this work, which is an essential ability of a robot. For a stable implementation of RTG tasks, Chapter 3 proposed an optimization framework that combines stable grasp planning with trajectory generation. The proposed optimization framework covered the complete path from perception to decision-making in RTG tasks. In this framework, the object of interest was first localized and recognized by using the proposed object detection model. A Bayesian-based search algorithm was used to find the grasp configuration with the highest grasp quality computed by a trained quality network. A model-based trajectory generation approach inspired by the human internal model theory was presented to compute constraint-satisfied trajectories for reaching movements. The effectiveness of the proposed framework was demonstrated and validated through comparative analysis and in real-world experiments. Experimental results demonstrated that the proposed learning framework allows robots to implement RTG tasks effectively.

Second, we addressed the challenge of *generalization of grasp planning*. In Chapter 5, we proposed a general-purpose grasp planning approach that suits various robotic hands with an arbitrary number of fingers. The novel definition of *pregrasping opposi-*

tion was introduced, which was taken as a waypoint for grasp formation. Additionally, in Chapter 4, we proposed an attention-based visual analysis framework that computed grasp-relevant information (i.e., grasp type and grasp attention point) directly from visual data. The grasp type and grasp attention point on an object segment presented in the ROI were then computed using a grasp type detection model. The *pregrasping opposition* was then used to encode the computed grasp-relevant information. Finally, a dual-stage grasp planning method was proposed to plan feasible grasps for various robotic hands with an arbitrary number of fingers. The final grasp configuration involving a set of contact points was planned with the guidance of the *pregrasping opposition*. In this framework, the use of the *pregrasping opposition* not only incorporated the grasp-relevant information into grasp planning but also provided a way to generalize the grasp planning approach to various robotic hands with an arbitrary number of fingers. Compared to previous methods without grasp-relevant information as guidance, the information generated from the visual analysis can significantly accelerate grasp planning . Moreover, the success rate of the grasping was also increased by exploiting the detected grasp type as guidance. The performance of the proposed framework was evaluated on a real-world robotic platform. The experimental results demonstrated that the proposed framework allowed various robotic hands to implement grasping tasks effectively.

Third, we proposed an object stabilization framework for a multi-fingered robotic hand to enable the grasp stability under uncertainties. The object stabilization framework combined tactile sensing with feedback control. We used deep learning to learn features from tactile data. A detection module sampled a tactile sequence online from the tactile readings as input and predicted the object material and contact event simultaneously. A force estimation method exploiting the spatial property of tactile data was proposed to compute the contact information (i.e., contact normal force and contact location). The spatio-temporal characteristics of tactile data were exploited for perception. An object stabilization controller was employed to adjust the grasp force for object stabilization. The effectiveness of the proposed framework was evaluated using a Shadow Dexterous Hand equipped with BioTac tactile sensors.

Fourth, in-hand manipulation was also an important ability for a multi-fingered robotic hand interacting with its environment. In Chapter 7, we proposed a MARL algorithm for a multi-fingered robotic hand to learn in-hand manipulations. The information extracted from both the analysis of hand posture synergies and the learning history was used to guide in-hand manipulation learning. Two additional reward functions (i.e., a hand-based reward and an uncertainty-based reward) were designed to encode the extracted information into the DRL. Three independent value agents represented by value functions were trained jointly with respect to their reward and then cooperate to optimize a control policy. In this way, the two additional rewards not only improved the exploration efficiency of the DRL algorithm but also provided a way to incorporate domain-specific knowledge. The performance of the proposed MARL algorithm was evaluated in a physical simulator with object rotation tasks. The experimental results showed that the proposed algorithm allowed a robotic hand to achieve object rotation tasks effectively. Moreover, this work demonstrated that exploiting domain-specific knowledge (e.g., grasp type) was effective for learning in-hand manipulations with multi-fingered robotic hands.

8.2 Discussion

The research presented in this thesis aimed at addressing the performance of robotic grasping and manipulation systems. To this end, we have proposed a variety of approaches in this thesis. Detailed discussions of these proposed approaches were introduced in each chapter. Here, we summarize them as follows.

8.2.1 Visual/tactile Perception for Robotic Grasping

In this thesis, we proposed an attention-based visual analysis framework that allows the robots to choose the most feasible grasp type from multiple grasp types to operate objects. In this work, six commonly used grasp types are considered, while the previous approaches only consider two grasp types. Moreover, the proposed framework learns features directly from raw sensor data, while most of the previous approaches use handcrafted features. This work only considered object attributes during grasp type detection. Because grasp planning is also affected by human preferences and task requirements, it is interesting to consider these constraints in grasp and manipulation planning.

In addition, we trained a deep neural network to learn a grasp quality metric that evaluates the performance of grasps from depth images. Different computational models based on deep neural networks have been trained for different applications (such as grasp quality measure and grasp type detection). However, humans typically use the two-stream hypothesis for visual analysis where a ventral stream recognizes object properties while a dorsal stream simultaneously interprets the geometric relationship to implement object grasping [164]. Hence, it is interesting to make use of deep learning techniques to train a unified model that measures the quality of grasps as well as detecting the grasp type.

In this thesis, we developed a tactile sensing method for robotic hands to detect contact events and object materials, and to estimate the contact force between a robotic hand and a grasped object. Although tactile sensors can provide critical information about interactions between the robotic hand and the object, they only perceive local contact information. Moreover, tactile information must be processed in real-time to enable fast responses. Visual sensors are among the most mainstream sensors in the robotics community. Visual sensors can provide global information with respect to objects and environments. However, the effectiveness of visual perception is easily affected by lighting conditions and occlusion. Therefore, it is necessary to explore novel methods to fuse visual and tactile information to improve perception capability.

8.2.2 Robotic Grasp Planning

This thesis addressed the stable grasping problem when implementing RTG tasks. We proposed an optimization framework that solved the grasp planning and trajectory generation problem when implementing RTG tasks. By using the optimization framework, the grasp planning module determined the feasible grasp configuration. The trajectory generation module took the planned grasp configuration as inputs and computed the constraint-satisfied reaching movement. In the proposed framework, the grasp planning

and trajectory generation were considered separately. It will be possible to develop a unified framework that jointly learns the grasping skill and motion moving. One possible way is to use a hierarchical deep RL algorithm to learn multiple control policies simultaneously. This way, we could achieve complex tasks that involve multiple subtasks.

Generalization of grasp planning is important for intelligent robots. This work also studied the general-purpose problem of grasp planning. The motivation of this work includes two points: (1) To enable robotic hands to grasp objects using a feasible grasp type as humans do. (2) To design a grasp planning method that suits various robotic hands with an arbitrary number of fingers. In terms of motivation 1, previous grasp planning methods only consider two grasp types (i.e., *Power* and *Precision*). Moreover, the grasp types are determined manually for robotic hands. We proposed taking advantage of deep learning techniques to detect grasp types directly from visual data. The experimental results demonstrated that the proposed methods enabled robotic hands to grasp objects with different grasp types. In terms of motivation 2, most of the previous grasp planning methods were designed for one specific hand. We designed a dual-stage grasp planning method beginning with the *pregrasping opposition*. Complex grasp configurations, such as *precision* grasps, were generated from a simple *pregrasping opposition*. In the proposed general-purpose grasp planning approach, we designed an analysis-based quality metric to evaluate the performance of candidate grasps. However, the performance of the analysis-based quality metric could easily be affected by sensor noise. For stably measuring grasp quality, we hope to use deep learning to learn a quality metric for dexterous grasping. Hence, a dataset including the object's point clouds and exampled grasps should be collected.

8.2.3 Effective In-hand Manipulation Learning

The problem of effective learning in-hand manipulation is far from solved. DRL algorithms in the context of an extrinsic reward are usually slow and unstable, especially when the extrinsic reward is sparse. Previous work has demonstrated that the use of prior knowledge helped to improve the performance of manipulation learning using DRL algorithms [7, 89]. However, it is still a challenge to represent and use prior knowledge in DRL algorithms. In this work, we exploited an abstraction of hand grasping posture (i.e., grasp type) as guidance to assist in-hand manipulation learning. We extracted the information from both the analysis of hand posture synergies and the learning history. Two additional rewards were designed to encode the extracted information. The experimental results demonstrated that this information helped robotic hands to improve the performance of in-hand manipulation learning. However, the proposed algorithm does not consider safety constraints to restrict the robot to explore promising states. It is essential for a real physical robotic system to consider safety constraints in DRL algorithms, because certain unreasonable states may break robotic systems.

8.3 Directions for Future Research

The implementation of dexterous grasping and manipulations with a multi-fingered robotic hand in daily-life environments still requires further investigation. Here we identify several interesting directions for future researches as follows.

- **Crossmodal perception for robotic manipulation:** Humans are capable of perceiving objects using multiple sensory modalities, such as vision, tactility, audio, and smell. Based on rich sensory information, humans can effectively interact with their environment. Previous work has used sensory data collected from a single modal sensor to accomplish various tasks (such as pose estimation, shape reconstruction, and material recognition). In Chapter 6, we used tactile data to detect a contact event and recognize the object material. However, a single modal sensor does not always provide enough information to perceive and understand the objects. Naturally, it is interesting to endow robots with the ability of human-like crossmodal perception. Currently, with the development of sensor technologies, robots could be equipped with multiple sensors that capture multimodal information (such as vision, sound, and tactile information). In the future, it will be important to explore fusion methods to integrate features from multiple sources effectively. In recent years, deep learning methods have achieved good results in a set of tasks, such as object recognition, audio separation, and texture detection. It is interesting to make use of deep learning techniques to develop multimodal feature learning methods to address complex perception tasks. Moreover, it is also an important ability for robots to transfer knowledge across different sensor modalities. For example, the effectiveness of visual perception is easily affected by lighting conditions and occlusion. The tactile modality can provide useful information about the contact between the robots and their environment. When one sensory modality (e.g., vision) is missing, we expect to use another (e.g., tactility) to carry out perception tasks. One of the possible methods is to formulate the transfer across different sensory modalities as a domain adaptation problem.

- **Learning abstractions from sensory data:** Humans construct some high-level abstractions (such as object affordance, task goal, and human preference) from sensory data to ground an object and tool in daily-life environments. The constructed abstractions can account for the interactive dynamic and continuous nature of the everyday human environment. Low-level sensory is are not enough to describe manipulation tasks because they cannot capture the invariant of manipulations. Robots need to reason on a high-level to efficiently understand the object and the manipulation tasks. In the future, it will be interesting to solve two problems: (1) How to learn high-level abstractions from multi-sensory data. (2) How to exploit the learned abstractions to assist in robotic manipulation learning. Recently, some works learn these abstractions (such as object affordance and human preference) separately from sensory data. In Chapter 4, we predicted the grasp type (i.e., a grasping affordance) directly from visual data. In the future, it is essential to learn multiple abstractions from sensory data jointly. Meanwhile,

we expect that the learned abstractions can be further exploited to guide the implementation of robotic grasping and manipulation in new situations.

- **Manipulation learning with active and safe exploration:** The efficiency of manipulation learning usually can be improved by exploiting prior knowledge or integrating active exploration. Prior knowledge is difficult to obtain in real-world applications. Active exploration during manipulation learning could help robots to acquire new manipulation skills effectively. Exploration can be understood as the act of gathering information about the environment. Through passive exploration, robots gather information by executing random policy/actions (such as ε-greedy policy) to interact with the environment. Passive exploration is usually time-consuming and not safe for robots and their environment. As robots need to address increasingly complex tasks, active exploration is of critical importance for the robots to learn manipulations. During active exploration, robots take advantage of their own learning experiences to adjust their exploration policy actively. Active exploration could be achieved in the following two ways: (1) By learning an intrinsic reward function that measures learning progress to guide the computation of exploration actions. This method is typically called as intrinsically motivated exploration. (2) By selecting the task goal for active exploration. In this way, robots can avoid spending too much time on unreachable or trivial goals and progressively optimize control policies.

  The problem of effective learning complex manipulations is far from solved. The robots are usually allowed to explore all the states in the configuration space during the process of manipulation learning. However, this is impractical for a real physical robotic system, because some unreasonable state may break the system. Safety constraints should also be considered in the DRL algorithm, which restricts the agent in exploring the reasonable states. Hence, it is essential to develop a active and safe exploration algorithm for manipulation learning.

- **Sim-to-Real domain adaptation for dexterous manipulation learning:** Robots should have the ability to perform dexterous manipulation to achieve complex manipulation tasks, such as writing and tool usage. However, owing to the high dimensionality of robotic hands and intermittent contact dynamics, effectively programming a robotic hand for in-hand manipulations remains a challenge. Manipulation learning using DRL algorithms is usually a time-consuming process. Simulations are attractive environments for training agents because they provide an abundant source of samples for model training. In Chapter 7, we developed a MARL algorithm for a robotic hand to learn in-hand manipulation. The performance of the proposed algorithm was only evaluated in a physical simulation. In the future, it is essential to explore methods that transfer the control policies learned in a simulator to a real-world environment. The control policies are first learned in a physics simulator and then deployed on real robots. It is also essential to ensure the safety of real-world robotic systems during manipulation learning. Safe reinforcement learning algorithms could be designed by modifying the optimization objectives of the DRL or by modifying the exploration policies of the

DRL by incorporating external knowledge.

8.4 Conclusion

This thesis contributes to the field of dexterous grasping and manipulation by exploring approaches aiming to improve the performance of robotic grasping and manipulation systems. We integrate visual/tactile perception with optimization methods to address four open challenges relevant to dexterous grasping and manipulation.

In conclusion, the proposed optimization framework achieves a stable grasp during manipulation tasks. The optimization framework combines stable grasp planning and model-based trajectory generation. Then, we propose a general-purpose grasp planning approach to improve the generalization of grasp planning. The proposed general-purpose grasp planning approach suits various robotic hands with a different number of fingers. Additionally, we address the challenge of how to enable grasp stability under uncertainties. An object stabilization framework through tactile sensing is proposed. Finally, we introduce a multi-agent reinforcement learning algorithm based on the analysis of hand synergies to reduce the complexity of in-hand manipulation learning. All these proposed approaches have been evaluated on simulation experiments or real-word robotic experiments.

Appendix A

List of Abbreviations

CNN Convolutional Neural Network

CNS Central Nervous System

DMP Dynamic Movement Primitive

RL Reinforcement Learning

DRL Deep Reinforcement Learning

GMM Gaussian Mixture Model

GP Gaussian Process

GTD Grasp Type Dataset

ICP Iterative Closest Point

IoU Intersection-over-Union

iLQR iteration Linear-Quadratic Regulator

LSTM Long Short Term Memory

MARL Multi-agent Reinforcement Learning

MC Monte Carlo

MDP Markov Decision Process

MPC Model Predictive Control

PCA Principal Components Analysis

PID Proportional Integral Derivative

PPO Proximal Policy Optimization

ROI Region of Interest

RTG Reach-to-grasp

SGD Stochastic Gradient Descent

SVM Support Vector Machine

TWS Task Wrench Space

UCB Upper Confidence Bound

VF Virtual Finger

Appendix B

Publications

The following publications formed part of this research. The list is ordered by date.

- Deng, Z., Gao, G., Frintrop, S., Sun, F., Zhang, C., Zhang, J. (2019). Attention based visual analysis for fast grasp planning with a multi-fingered robotic hand. Frontiers in neurorobotics, 13, 60.

- Yan, W., Deng, Z., Chen, J., Nie, H., Zhang, J. (2019). Precision Grasp Planning for Multi-Finger Hand to Grasp Unknown Objects. Robotica, 37(8), 1415-1437.

- Mi, J., Tang, S., Deng, Z., Goerner, M., Zhang, J. (2019). Object affordance based multimodal fusion for natural Human-Robot interaction. Cognitive Systems Research, 54, 128-137.

- Deng, Z., Zheng, X., Zhang, L., Zhang, J. (2018). A learning framework for semantic reach-to-grasp tasks integrating machine learning and optimization. Robotics and Autonomous Systems, 108, 140-152.

- Han, D., Nie, H., Chen, J., Chen, M., Deng, Z., Zhang, J. (2018). Multi-modal haptic image recognition based on deep learning. Sensor Review, 38(4), 486-493.

- Deng, Z., Guan, H., Huang, R., Liang, H., Zhang, L., Zhang, J. (2017). Combining Model-Based Q-Learning With Structural Knowledge Transfer for Robot Skill Learning. IEEE Transactions on Cognitive and Developmental Systems, 11(1), 26-35.

- Liu, J., Deng, Z., Sun, Y., Hu, Y. (2017, October). An Eye Movement Simulator Based on 3-DOF Parallel Mechanism With Flexure Joint. In 2017 Design of Medical Devices Conference. American Society of Mechanical Engineers Digital Collection.

- Deng, Z., Mi, J., Chen, Z., Einig, L., Zou, C., Zhang, J. (2016, December). Learning human compliant behavior from demonstration for force-based robot manipulation. In 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 319-324). IEEE.

- Zou, C., He, B., Zhang, L., Zhang, J., Deng, Z. (2016, December). An automatic calibration between an omni-directional camera and a laser rangefinder for dynamic scenes reconstruction. In 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 1528-1534). IEEE.

- Mi, J., Sun, Y., Wang, Y., Deng, Z., Li, L., Zhang, J., Xie, G. (2016, December). Gesture recognition based teleoperation framework of robotic fish. In 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 137-142). IEEE.

Appendix C

Acknowledgements

In performing this research, I have greatly benefited from the supports and encouragements of many people. I would like to express my very great appreciation to all of them.

First and foremost, I would like to thank my advisor Prof. Dr. Jianwei Zhang for all his support throughout my doctoral studies. He gives me the opportunity to study at Universität Hamburg. His assistance helped me in the development of this research work and writing of the thesis.

Besides my advisor, I would like to thank my co-supervisor Prof. Dr. Stefen Wermter, for his valuable and constructive suggests and advice. His willingness to give his time has been a great help in creating this thesis.

I am also grateful to all the members of our TAMS group. It was wonderful working with you. In particular, I owe many thanks to Tatjana Lu Tetsis and Wiebke Noeske for their time and helping me a lot during the past four years. I would like to thank Dr. Norman Hendrich, Dr. Andreas Mäder, Yannick Jonetzko, Michael Görner, Philipp Ruppel, Florens Wasserfall, Lasse Einig, Dennis Krupke for discussions. Furthermore, I thank my Chinese friends in Hamburg as Dr. Junhu He, Dr. Kunpeng Wang, Dr. Xiaozhi Qi, Lin Cong, Ge Gao, Hongzhuo Liang, Jinpeng Mi, Yuyang Tu, Di Zhang, Xinqiang Pan, Chao Zeng, Shuang Li, Dan Liu et al. I wish to thank especially to Jie Wu for all her love and support.

In addition, I gratefully acknowledge the support of the German Science Foundation (DFG) and National Science Foundation of China (NSFC) in project Crossmodal Learning under contract Son-derforschungsbereich Transregio 169. I would like to thank the ones I had the opportunity to collaborate with as Prof. Dr. Simone Frintrop, Prof. Dr. Fuchun Sun, and Dr. Bing Fang.

Last but not least, I would like to thank my parents Huiguang Deng and Airong Jiang, and my brother Le Deng for unconditional love and support of all kinds. This accomplishment would be impossible without them. Thank you.

Thanks for all your encouragement!

Bibliography

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[2] Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

[3] Jacopo Aleotti and Stefano Caselli. A 3D shape segmentation approach for robot grasping by parts. *Robotics and Autonomous Systems*, 60(3):358–366., 2012.

[4] Michael Anthony Arbib. Coordinated control programs for movements of the hand. *Experimantal Brain Research*, 10:111–129, 1985.

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495., 2017.

[6] Benjamin Balaguer and Stefano Carpin. Efficient grasping of novel objects through dimensionality reduction. In *2010 IEEE International Conference on Robotics and Automation*, pages 1279–1285. IEEE, 2010.

[7] Somil Bansal, Roberto Calandra, Kurtland Chua, Sergey Levine, and Claire Tomlin. Mbmf: Model-based priors for model-free reinforcement learning. *arXiv preprint arXiv:1709.03153*, 2017.

[8] Yasemin Bekiroglu, Danica Kragic, and Ville Kyrki. Learning grasp stability based on tactile data and hmms. In *19th International Symposium in Robot and Human Interactive Communication*, pages 132–137. IEEE, 2010.

[9] Yasemin Bekiroglu, Dan Song, Lu Wang, and Danica Kragic. A probabilistic framework for task-oriented grasp stability assessment. In *2013 IEEE International Conference on Robotics and Automation*, pages 3040–3047. IEEE, 2013.

[10] Richard Bellman. *Dynamic programming*. Courier Corporation, 2013.

[11] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesisa survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2013.

[12] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesisa survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014.

[13] Ch Borst, Max Fischer, and Gerd Hirzinger. Grasp planning: How to choose a suitable task wrench space. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 1, pages 319–325. IEEE, 2004.

[14] Ian M Bullock, Thomas Feix, and Aaron M Dollar. The Yale human grasping dataset: Grasp, object, and task data in household and machine shop environments. *The International Journal of Robotics Research*, 34(3):251–255., 2015.

[15] Minjie Cai, Kris M Kitani, and Yoichi Sato. A scalable approach for understanding the visual structures of hand grasps. In *IEEE International Conference on Robotics and Automation*, pages 1360–1366., 2015.

[16] Minjie Cai, Kris M Kitani, and Yoichi Sato. Understanding hand-object manipulation with grasp types and object attributes. In *Robotics: Science and Systems*, volume 3. Ann Arbor, Michigan;, 2016.

[17] Minjie Cai, Kris M Kitani, and Yoichi Sato. An ego-vision system for hand grasp analysis. *IEEE Transactions on Human-Machine Systems*, 47(4):524–535., 2017.

[18] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.

[19] Sylvain Calinon, Florent D'halluin, Eric L Sauser, Darwin G Caldwell, and Aude G Billard. Learning and reproduction of gestures by imitation. *IEEE Robotics & Automation Magazine*, 17(2):44–54, 2010.

[20] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22(3):36–52., 2015.

[21] Yevgen Chebotar, Karol Hausman, Oliver Kroemer, Gaurav Sukhatme, and Stefan Schaal. Regrasping using tactile perception and supervised policy learning. In *2017 AAAI Spring Symposium Series*, 2017.

[22] Yevgen Chebotar, Karol Hausman, Zhe Su, Gaurav S Sukhatme, and Stefan Schaal. Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1960–1966. IEEE, 2016.

[23] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848., 2018.

[24] Wenrui Chen, Caihua Xiong, Mingjin Liu, and Liu Mao. Characteristics analysis and mechanical implementation of human finger movements. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 403–408. IEEE, 2014.

[25] Lillian Chin, Jeffrey Lipton, Michelle C Yuen, Rebecca Kramer-Bottiglio, and Daniela Rus. Automated recycling separation enabled by soft robotic material classification. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 102–107. IEEE, 2019.

[26] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.

[27] Vivian Chu, Ian McMahon, Lorenzo Riano, Craig G McDonald, Qin He, Jorge Martinez Perez-Tejada, Michael Arrigo, Trevor Darrell, and Katherine J Kuchenbecker. Robotic learning of haptic adjectives through physical interaction. *Robotics and Autonomous Systems*, 63:279–292, 2015.

[28] F Cini, V Ortenzi, P Corke, and M Controzzi. On the choice of grasp type and location when handing over an object. *Science Robotics*, 4(27):eaau9757, 2019.

[29] Matei T Ciocarlie and Peter K Allen. Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7):851–867, 2009.

[30] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619., 2002.

[31] Mark R Cutkosky and Robert D Howe. Human grasp choice and robotic grasp analysis. In *Dextrous robot hands*, pages 5–31. Springer, 1990.

[32] Mark R Cutkosky and John Ulmen. Dynamic tactile sensing. In *The Human Hand as an Inspiration for Robot Hand Development*, pages 389–403. Springer, 2014.

[33] Hao Dang and Peter K Allen. Grasp adjustment on novel objects using tactile experience from similar local geometry. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4007–4012. IEEE, 2013.

[34] Hao Dang and Peter K Allen. Semantic grasping: planning task-specific stable robotic grasps. *Autonomous Robots*, 37(3):301–316, 2014.

[35] Ravin De Souza, Sahar El-Khoury, José Santos-Victor, and Aude Billard. Recognizing the grasp intention from human demonstration. *Robotics and Autonomous Systems*, 74:108–121, 2015.

[36] Angel Delgado, Juan Antonio Corrales, Youcef Mezouar, Laurent Lequievre, Carlos Jara, and Fernando Torres. Tactile control based on gaussian images and its application in bi-manual manipulation of deformable objects. *Robotics and Autonomous systems*, 94:148–161, 2017.

[37] Ioannis Delis, Bastien Berret, Thierry Pozzo, and Stefano Panzeri. Quantitative evaluation of muscle synergy models: a single-trial task decoding approach. *Frontiers in computational neuroscience*, 7:8, 2013.

[38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255., 2009.

[39] Zhen Deng, Jinpeng Mi, Zhixian Chen, Lasse Einig, Cheng Zou, and Jianwei Zhang. Learning human compliant behavior from demonstration for force-based robot manipulation. In *Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on*, pages 319–324. IEEE, 2016.

[40] Renaud Detry, Carl Henrik Ek, Marianna Madry, Justus Piater, and Danica Kragic. Generalizing grasps across partly similar objects. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3791–3797. IEEE, 2012.

[41] Renaud Detry, Jeremie Papon, and Larry Matthies. Task-oriented grasping with semantic and geometric scene understanding. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3266–3273. IEEE, 2017.

[42] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338, 2016.

[43] Sahar El-Khoury, Miao Li, and Aude Billard. On the generation of a variety of grasps. *Robotics and Autonomous Systems*, 61(12):1335–1349, 2013.

[44] Thomas Feix, Ian M Bullock, and Aaron M Dollar. Analysis of human grasping behavior: Correlating tasks, objects and grasps. *IEEE transactions on haptics*, 7(4):430–441, 2014.

[45] Thomas Feix, Ian M Bullock, and Aaron M Dollar. Analysis of human grasping behavior: Object characteristics and grasp type. *IEEE transactions on haptics*, 7(3):311–323, 2014.

[46] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77., 2016.

[47] Simone Frintrop. VOCUS : A visual attention system for object detection and goal-directed search. *Lecture Notes Artificial Intelligence*, 3899:1–197., 2006.

[48] Simone Frintrop, Thomas Werner, and German Martin Garcia. Traditional saliency reloaded: A good old model in new shape. In *IEEE Conference on Computer Vision and Pattern Recognition.*, 2015.

[49] Marco Gabiccini, Antonio Bicchi, Domenico Prattichizzo, and Monica Malvezzi. On the role of hand synergies in the optimal choice of grasping forces. *Autonomous Robots*, 31(2-3):235, 2011.

[50] Yang Gao, Lisa Anne Hendricks, Katherine J Kuchenbecker, and Trevor Darrell. Deep learning for tactile understanding from visual and haptic data. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 536–543. IEEE, 2016.

[51] Mathieu Geisert and Nicolas Mansard. Trajectory generation for quadrotor based systems using numerical optimal control. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2958–2964. IEEE, 2016.

[52] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[53] Georg Goldenberg. Imitation and matching of hand and finger postures. *Neuroimage*, 14(1):S132–S136, 2001.

[54] Franck Gonzalez, Florian Gosselin, and Wael Bachta. Analysis of hand contact areas and interaction capabilities during manipulation and exploration. *IEEE transactions on haptics*, 7(4):415–429, 2014.

[55] Ilaria Gori, Ugo Pattacini, Vadim Tikhanoff, and Giorgio Metta. Three-finger precision grasp on incomplete 3d point clouds. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5366–5373. IEEE, 2014.

[56] Marcus Gualtieri and Robert Platt. Learning 6-dof grasping and pick-place using attention focus. *arXiv preprint arXiv:1806.06134*, 2018.

[57] Di Guo, Tao Kong, Fuchun Sun, and Huaping Liu. Object discovery and grasp detection with a shared convolutional neural network. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2038–2043. IEEE, 2016.

[58] Di Guo, Fuchun Sun, Bin Fang, Chao Yang, and Ning Xi. Robotic grasping using visual and tactile sensing. *Information Sciences*, 417:274–286, 2017.

[59] Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3786–3793. IEEE, 2016.

[60] Dong Han, Hong Nie, Jinbao Chen, Meng Chen, Zhen Deng, and Jianwei Zhang. Multi-modal haptic image recognition based on deep learning. *Sensor Review*, 38(4):486–493, 2018.

[61] Li Han, Jeffrey C Trinkle, and Zexiang X Li. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*, 16(6):663–674, 2000.

[62] Kaiyu Hang, Miao Li, Johannes A Stork, Yasemin Bekiroglu, Florian T Pokorny, Aude Billard, and Danica Kragic. Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation. *IEEE Transactions on robotics*, 32(4):960–972, 2016.

[63] Kaiyu Hang, Johannes A Stork, and Danica Kragic. Hierarchical fingertip space for multi-fingered precision grasping. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1641–1648. IEEE, 2014.

[64] Kensuke Harada, Kenji Kaneko, and Fumio Kanehiro. Fast grasp planning for hand/arm systems based on convex model. In *IEEE International Conference on Robotics and Automation*, pages 1162–1168., 2008.

[65] Alexander Herzog, Peter Pastor, Mrinal Kalakrishnan, Ludovic Righetti, Jeannette Bohg, Tamim Asfour, and Stefan Schaal. Learning of grasp selection based on shape-templates. *Autonomous Robots*, 36(1-2):51–65., 2014.

[66] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[67] Francois R Hogan, Maria Bauza, Oleguer Canal, Elliott Donlon, and Alberto Rodriguez. Tactile regrasp: Grasp adjustments via simulated tactile transformations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2963–2970. IEEE, 2018.

[68] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016.

[69] Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E Gil Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228–1235., 2010.

[70] Kaijen Hsiao, Matei Ciocarlie, Peter Brook, and Willow Garage. Bayesian grasp planning. In *ICRA 2011 Workshop on Mobile Manipulation: Integrating Perception and Manipulation*, 2011.

[71] Kaijen Hsiao, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Robust grasping under object pose uncertainty. *Autonomous Robots*, 31(2-3):253, 2011.

[72] Bidan Huang, Miao Li, Ravin Luis De Souza, Joanna J Bryson, and Aude Billard. A modular approach to learning manipulation strategies from human demonstration. *Autonomous Robots*, 40(5):903–927, 2016.

[73] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. SALICON: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *IEEE International Conference on Computer Vision*, 2015.

[74] Thea Iberall. The nature of human prehension: Three dextrous hands in one. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 396–401. IEEE, 1987.

[75] Thea Iberall. Human prehension and dexterous robot hands. *The International Journal of Robotics Research*, 16(3):285–299, 1997.

[76] Thea Iberall and Christine L MacKenzie. Opposition space and human prehension. In *Dextrous robot hands*, pages 32–54. Springer, 1990.

[77] Auke J Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in neural information processing systems*, pages 1547–1554, 2003.

[78] Masao Ito. Control of mental activities by internal models in the cerebellum. *Nature Reviews Neuroscience*, 9(4):304–313, 2008.

[79] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259., Nov 1998.

[80] Rainer Jäkel, Sven R Schmidt-Rohr, Steffen W Rühl, Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. Learning of planning models for dexterous manipulation based on human demonstrations. *International Journal of Social Robotics*, 4(4):437–448, 2012.

[81] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345, 2009.

[82] Edward Johns, Stefan Leutenegger, and Andrew J Davison. Deep learning a grasp function for grasping under gripper pose uncertainty. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4461–4468. IEEE, 2016.

[83] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[84] Mitsuo Kawato and Daniel Wolpert. Internal models for motor control. *Sensory Guidance of Movement*, 218:291–307, 1998.

[85] Mohi Khansari and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[86] Byoung-Ho Kim, Sang-Rok Oh, Byung-Ju Yi, and Il Hong Suh. Optimal grasping based on non-dimensionalized performance indices. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 2, pages 949–956. IEEE, 2001.

[87] W Bradley Knox and Peter Stone. Reinforcement learning from simultaneous human and mdp reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 475–482. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[88] Jens Kober and Jan R Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856, 2009.

[89] George Konidaris and Andrew Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 489–496. ACM, 2006.

[90] Marek Kopicki, Renaud Detry, Maxime Adjigble, Rustam Stolkin, Ales Leonardis, and Jeremy L Wyatt. One-shot learning and generation of dexterous grasps for novel objects. *The International Journal of Robotics Research*, 35(8):959–976., 2016.

[91] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012.

[92] Oliver Kroemer and Jan Peters. Predicting object interactions from contact distributions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3361–3367., 2014.

[93] Robert Krug, Achim J Lilienthal, Danica Kragic, and Yasemin Bekiroglu. Analytic grasp success prediction with tactile feedback. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 165–171. IEEE, 2016.

[94] Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6808–6815. IEEE, 2014.

[95] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 378–383. IEEE, 2016.

[96] Hamid Laga, Michela Mortara, and Michela Spagnuolo. Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes. *ACM Transactions on Graphics (TOG)*, 32(5):150:1–150:16., 2013.

[97] Michael Laskey, Jeff Mahler, Zoe McCarthy, Florian T Pokorny, Sachin Patil, Jur Van Den Berg, Danica Kragic, Pieter Abbeel, and Ken Goldberg. Multi-armed bandit models for 2d grasp planning with uncertainty. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 572–579. IEEE, 2015.

[98] Adam Daniel Laud. Theory and application of reward shaping in reinforcement learning. Technical report, 2004.

[99] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[100] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, page 0278364917710318, 2016.

[101] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[102] Sergey Levine, Nolan Wagener, and Pieter Abbeel. Learning contact-rich manipulation skills with guided policy search. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 156–163. IEEE, 2015.

[103] Jianhua Li, Siyuan Dong, and Edward Adelson. Slip detection with combined tactile and visual information. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7772–7777. IEEE, 2018.

[104] Miao Li, Yasemin Bekiroglu, Danica Kragic, and Aude Billard. Learning of grasp adaptation through experience and tactile sensing. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3339–3346. Ieee, 2014.

[105] Miao Li, Kaiyu Hang, Danica Kragic, and Aude Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364, 2016.

[106] X. Li, L. Zhao, L. Wei, M. H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. DeepSaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing*, 25(8):3919–3930., Aug 2016.

[107] Zhijun Li, Ting Zhao, Fei Chen, Yingbai Hu, Chun-Yi Su, and Toshio Fukuda. Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator. *IEEE/ASME Transactions on Mechatronics*, 23(1):121–131, 2017.

[108] Minas V Liarokapis, Panagiotis K Artemiadis, Pantelis T Katsiaris, Kostas J Kyriakopoulos, and Elias S Manolakos. Learning human reach-to-grasp strategies: Towards emg-based control of robotic arm-hand systems. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2287–2292. IEEE, 2012.

[109] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[110] Huaping Liu, Yuanlong Yu, Fuchun Sun, and Jason Gu. Visual–tactile fusion for object recognition. *IEEE Transactions on Automation Science and Engineering*, 14(2):996–1008, 2016.

[111] Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017.

[112] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[113] Tanis Mar, Vadim Tikhanoff, Giorgio Metta, and Lorenzo Natale. Self-supervised learning of grasp dependent tool affordances on the icub humanoid robot. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3200–3206. IEEE, 2015.

[114] Carolyn R Mason, Jose E Gomez, and Timothy J Ebner. Hand synergies during reach-to-grasp. *Journal of neurophysiology*, 86(6):2896–2910, 2001.

[115] Kümmerer Matthias, Theis Lucas, and Bethge Matthias. Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet. In *International Conference on Learning Representations Workshop.*, 2015.

[116] Christoforos I Mavrogiannis, Charalampos P Bechlioulis, and Kostas J Kyriakopoulos. Sequential improvement of grasp based on sensitivity analysis. In *2013 IEEE International Conference on Robotics and Automation*, pages 1094–1099. IEEE, 2013.

[117] Hao Men, Biruk Gebre, and Kishore Pochiraju. Color point cloud registration with 4d icp algorithm. In *2011 IEEE International Conference on Robotics and Automation*, pages 1511–1516. IEEE, 2011.

[118] Paul Michelman. Precision object manipulation with a multifingered robot hand. *IEEE Transactions on Robotics and Automation*, 14(1):105–113, 1998.

[119] Andrew T Miller, Steffen Knoop, Henrik Iskov Christensen, and Peter K Allen. Automatic grasp planning using shape primitives. 2003.

[120] Hadi Beik Mohammadi, Mohammad Ali Zamani, Matthias Kerzel, and Stefan Wermter. Mixed-reality deep reinforcement learning for a reach-to-grasp task. In *International Conference on Artificial Neural Networks*, pages 611–623. Springer, 2019.

[121] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotionan inverse optimal control approach. *Autonomous robots*, 28(3):369–383, 2010.

[122] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144. Eurographics Association, 2012.

[123] Adithyavairavan Murali, Lerrel Pinto, Dhiraj Gandhi, and Abhinav Gupta. Cassl: Curriculum accelerated self-supervised learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6453–6460. IEEE, 2018.

[124] Austin Myers, Angjoo Kanazawa, Cornelia Fermuller, and Yiannis Aloimonos. Affordance of object parts from geometric features. In *Workshop on Vision meets Cognition, CVPR*, volume 9, 2014.

[125] John R Napier. The prehensile movements of the human hand. *The Journal of bone and joint surgery. British volume*, 38(4):902–913., 1956.

[126] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Detecting object affordances with convolutional neural networks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2765–2770. IEEE, 2016.

[127] Liu Nian and Han Junwei. DHSNet: Deep hierarchical saliency network for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[128] José Nogueira, Ruben Martinez-Cantin, Alexandre Bernardino, and Lorenzo Jamone. Unscented bayesian optimization for safe robot grasping. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1967–1972. IEEE, 2016.

[129] Lael U Odhner and Aaron M Dollar. Stable, open-loop precision manipulation with underactuated hands. *The International Journal of Robotics Research*, 34(11):1347–1360, 2015.

[130] Allison M Okamura, Niels Smaby, and Mark R Cutkosky. An overview of dexterous manipulation. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 255–262. IEEE, 2000.

[131] Bowen; Chociej Maciek; Jozefowicz Rafal; McGrew Bob; Pachocki Jakub; Petron Arthur; Plappert Matthias; Powell Glenn; Ray Alex; Schneider Jonas; Sidor Szymon; Tobin Josh; Welinder Peter; Weng Lilian; Zaremba Wojciech OpenAI; :; Andrychowicz, Marcin; Baker. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.

[132] Takayuki Osa, Jan Peters, and Gerhard Neumann. Experiments with hierarchical reinforcement learning of multiple grasping policies. In *International Symposium on Experimental Robotics*, pages 160–172. Springer, 2016.

[133] Gianluca Palli, Fanny Ficuciello, Umberto Scarcia, Claudio Melchiorri, and Bruno Siciliano. Experimental evaluation of synergy-based in-hand manipulation. *IFAC Proceedings Volumes*, 47(3):299–304, 2014.

[134] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *arXiv preprint arXiv:1706.09911*, 2017.

[135] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *arXiv preprint arXiv:1810.03599*, 2018.

[136] Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6284–6291. IEEE, 2018.

[137] Ana Huamán Quispe, Benoît Milville, Marco A Gutiérrez, Can Erdogan, Mike Stilman, Henrik Christensen, and Heni Ben Amor. Exploiting symmetries and extrusions for grasping household objects. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3702–3708. IEEE, 2015.

[138] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[139] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1316–1322. IEEE, 2015.

[140] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[141] Yara Rizk, Mariette Awad, and Edward W Tunstel. Decision making in multi-agent systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 2018.

[142] Maximo A Roa, Max J Argus, Daniel Leidner, Christoph Borst, and Gerd Hirzinger. Power grasp planning for anthropomorphic robot hands. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 563–569. IEEE, 2012.

[143] Máximo A Roa and Raúl Suárez. Grasp quality measures: review and performance. *Autonomous robots*, 38(1):65–88, 2015.

[144] Alberto Rodriguez, Matthew T Mason, and Steve Ferry. From caging to grasping. *The International Journal of Robotics Research*, 31(7):886–900, 2012.

[145] Grégory Rogez, James S Supancic, and Deva Ramanan. Understanding everyday hands in action from RGB-D images. In *IEEE International Conference on Computer Vision*, pages 3889–3897., 2015.

[146] Joseph M Romano, Kaijen Hsiao, Günter Niemeyer, Sachin Chitta, and Katherine J Kuchenbecker. Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics*, 27(6):1067–1079, 2011.

[147] Philipp Ruppel, Norman Hendrich, Sebastian Starke, and Jianwei Zhang. Cost functions to specify full-body motion and multi-goal manipulation tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3152–3159. IEEE, 2018.

[148] Anis Sahbani, Sahar El-Khoury, and Philippe Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012.

[149] Marco Santello, Martha Flanders, and John F Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998.

[150] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

[151] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[152] Max Schwarz, Anton Milan, Christian Lenz, Aura Munoz, Arul Selvam Periyasamy, Michael Schreiber, Sebastian Schüller, and Sven Behnke. NimbRo picking: Versatile part handling for warehouse automation. In *IEEE International Conference on Robotics and Automation*, pages 3032–3039., 2017.

[153] Sumit Shekhar, Vishal M Patel, Nasser M Nasrabadi, and Rama Chellappa. Joint sparse representation for robust multimodal biometrics recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 36(1):113–126, 2013.

[154] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *international conference on learning representations*, 2015.

[155] Gordon Smith, Eric Lee, Ken Goldberg, Karl Bohringer, and John Craig. Computing parallel-jaw grips. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 3, pages 1897–1903. IEEE, 1999.

[156] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.

[157] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[158] Zhe Su, Karol Hausman, Yevgen Chebotar, Artem Molchanov, Gerald E Loeb, Gaurav S Sukhatme, and Stefan Schaal. Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 297–303. IEEE, 2015.

[159] Raúl Suárez, Jordi Cornella, and Máximo Roa Garzón. *Grasp quality measures*. Citeseer, 2006.

[160] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[161] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 2011.

[162] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2012.

[163] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97 – 136., 1980.

[164] Leslie G Ungerleider and James V Haxby. whatand wherein the human brain. *Current opinion in neurobiology*, 4(2):157–165, 1994.

[165] Nikolaus Vahrenkamp, Martin Do, Tamim Asfour, and Rüdiger Dillmann. Integrated grasp and motion planning. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2883–2888. IEEE, 2010.

[166] Nikolaus Vahrenkamp, Eduard Koch, Mirko Waechter, and Tamim Asfour. Planning high-quality grasps using mean curvature object skeletons. *IEEE Robotics and Automation Letters*, 2018.

[167] Karl Van Wyk and Joe Falco. Slip detection: Analysis and calibration of univariate tactile signals. *arXiv preprint arXiv:1806.10451*, 2018.

[168] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2442–2447. IEEE, 2017.

[169] Filipe Veiga, Benoni B Edin, and Jan Peters. In-hand object stabilization by independent finger control. *arXiv preprint arXiv:1806.05031*, 2018.

[170] Filipe Veiga, Jan Peters, and Tucker Hermans. Grip stabilization of novel objects using slip prediction. *IEEE transactions on haptics*, 11(4):531–542, 2018.

[171] Filipe Veiga, Herke Van Hoof, Jan Peters, and Tucker Hermans. Stabilizing novel objects by learning to predict tactile slip. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5065–5072. IEEE, 2015.

[172] Matthew Veres, Medhat Moussa, and Graham W Taylor. An integrated simulator and dataset that combines grasping and vision for deep learning. *arXiv preprint arXiv:1702.02103.*, 2017.

[173] Giulia Vezzani, Ugo Pattacini, and Lorenzo Natale. A grasping approach based on superquadric models. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1579–1586. IEEE, 2017.

[174] Dirk Walther and Christof Koch. Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407., 2006.

[175] Qian Wan, Ryan P Adams, and Robert D Howe. Variability and predictability in tactile sensing during grasping. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 158–164. IEEE, 2016.

[176] Jonathan Weisz and Peter K Allen. Pose error robust grasping from contact wrench space metrics. In *2012 IEEE international conference on robotics and automation*, pages 557–562. IEEE, 2012.

[177] Thomas Wimböck, Christian Ott, Alin Albu-Schäffer, and Gerd Hirzinger. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1):3–23, 2012.

[178] Zhixing Xue, J Marius Zoellner, and Ruediger Dillmann. Automatic optimal grasp planning based on found contact points. In *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1053–1058. IEEE, 2008.

[179] Chenguang Yang, Chuize Chen, Ning Wang, Zhaojie Ju, Jian Fu, and Min Wang. Biologically-inspired motion modeling and neural control for robot learning from demonstrations. *IEEE Transactions on Cognitive and Developmental Systems*, 2018.

[180] Brayan S Zapata-Impata, Pablo Gil, and Fernando Torres. Learning spatio temporal tactile features with a convlstm for the direction of slip detection. *Sensors*, 19(3):523, 2019.

[181] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.

[182] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. *arXiv preprint arXiv:1810.06045*, 2018.

[183] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.

Eidesstattliche Versicherung

Hiermit erklre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Ort, Datum Unterschrift