A DEVELOPMENTAL GRASP LEARNING SCHEME FOR HUMANOID ROBOTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ASİL KAAN BOZCUOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

A DEVELOPMENTAL GRASP LEARNING SCHEME FOR HUMANOID ROBOTS

submitted by **ASİL KAAN BOZCUOĞLU** in partial fulfillment of the requirements for the degree of
Master of Science in Computer Engineering Department, Middle East Technical University by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** ________________

Assoc. Prof. Dr. Erol Şahin
Supervisor, **Computer Engineering Department** ________________

Asst. Prof. Dr. Erhan Öztop
Co-supervisor, **Computer Science Program, Özyeğin University** ________________

Examining Committee Members:

Prof. Dr. Göktürk Üçoluk
Computer Engineering Department, METU ________________

Assoc. Prof. Dr. Erol Şahin
Computer Engineering Department, METU ________________

Asst. Prof. Dr. Erhan Öztop
Computer Science Program, Özyeğin University ________________

Asst. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU ________________

Asst. Prof. Dr. Buğra Koku
Mechanical Engineering Department, METU ________________

Date: 05.09.2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ASİL KAAN BOZCUOĞLU

Signature :

ABSTRACT

A DEVELOPMENTAL GRASP LEARNING SCHEME FOR HUMANOID ROBOTS

Bozcuoğlu, Asil Kaan
M.Sc., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Erol Şahin
Co-Supervisor : Asst. Prof. Dr. Erhan Öztop

September 2012, 64 pages

While an infant is learning to grasp, there are two key processes that she uses for leading a successful development. In the first process, infants use an intuitional approach where the hand is moved towards the object to create an initial contact regardless of the object properties. The contact is followed by a tactile grasping phase where the object is enclosed by the hand. This intuitive grasping behavior leads an grasping mechanism, which utilizes visual input and incorporates this into the grasp plan. The second process is called *scaffolding*, a guidance by stating how to accomplish the task or modifying its behaviors by interference. Infants pay attention to such guidance and understand the indication of important features of an object from 9 months of age. This supervision mechanism plays an important role for learning how to grasp certain objects in a proper way.

To simulate these behavioral findings, a reaching and a tactile grasping controller was implemented on *iCub* humanoid robot which allowed it to reach an object from different directions, and enclose its fingers to cover the object. With these, a human-like grasp learning for *iCub* is proposed. Namely, the first stage is an unsupervised learning where the robot is experimenting how to grasp objects. The second stage is supervised learning phase where a caregiver modifies the end-effector's position when the robot is mistaken.

By doing several experiments for two different grasping styles, we observe that the proposed methodology shows a better learning rate comparing to the scaffolding-only learning mechanism.

Keywords: affordances, learning by demonstration, scaffolding, grasp learning, developmental robotics

ÖZ

İNSANSI ROBOTLAR İÇİN GELİŞİMSEL BİR KAVRAMA ÖĞRENİM SİSTEMİ

Bozcuoğlu, Asil Kaan
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Erol Şahin
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Erhan Öztop

Eylül 2012, 64 sayfa

Bir bebek kavramayı öğrenirken, başarılı bir gelişim sağlamak için iki ana yöntem kullanır. İlk yöntemde, bebek elini rastgele hareketlerle, nesne özelliklerinden bağımsız olarak nesneye doğru götürerek ilk teması sağlamaya çalışır. Ardından, dokunsal kavrama ile nesnenin yüzeyi sarılır. Bu sezgisel kavrama davranışı daha sonraları kendini bir kavrama mekanizmasına bırakır. İkinci aşamaysa, bir ebeveyn tarafından verilen ve bir hareketin nasıl başarılacağını gösteren rehberlik işlemi olan *ebeveyn iskelesi* adı verilmektedir. Bebekler bu tür rehberliklere ve yol göstermelere 9 aylık olmalarından itibaren dikkat gösterirler. Bu gözetim bebeğin doğru bir şekilde nesne kavramasını öğrenmesinde büyük önem taşır.

Bu davranışsal bulguları benzetim yapmak amacıyla, *iCub* insansı robotu üstünde bir erişim ve dokunsal kavrama sistemi geliştirilmiştir. Bu sistemle, robotun nesneye değişik yönlerden uzanarak, robotun parmaklarıyla nesnenin yüzeyini örtmesine olanak tanımıştır. Bunlarla, insan benzeri kavrama öğrenme metodu *iCub* için tasarlanmıştır. Özetle, ilk aşamada robot bir gözetim olmaksızın kendi çabalarıyla kavramayı öğrenmeye çalışırken ikinci aşamada ebeveyn gözetiminde kavrama yapnası ve hataların ebeveyn tarafından düzeltilmesi öngörülmüştür.

İki kavrama çeşidi için yapılan çeşitli deneylerde, tasarlanan sisteminin sadece ebeveyn iske-

lesi tabanı sisteme oranla daha başarılı olduğu görülmüştür.

Anahtar Kelimeler: sağlarlıklar, gösterme ile öğrenim, ebeveyn iskelesi, kavrama öğrenimi, gelişimsel robotik

to my family...

ACKNOWLEDGMENTS

I want to thank my supervisor, Erol Şahin for giving me the opportunity to work under his supervision during my M.Sc. studies. Being a member of KOVAN Research Lab was a great experience with the latest technology robotics equipments and the cutting-edge research subjects.

I would like to express my deepest gratitudes to Erhan Öztop for being my co-supervisor and supervise this thesis with his deep knowledge on cognitive robotics and grasping. This thesis wouldn't be there without his supervision and mentorship.

I would also like to thank all of the previous and current KOVAN members, my colleagues: Kadir Fırat Uyanık for being a good friend and a co-researcher during the "traversability year" and ROSSI days; Yiğit Çalışkan for being a never-needed-but-always-there "2-D" guy, a great co-allnighter and co-data collector; Mustafa Parlaktuna for his great humour, "machine jokes" and his debugging ability; Onur Yürüten for being at the same "robotic" path with me during both B.S. and M.S. studies; Fatih Gökçe, a helpful senior in both research and teaching assistantship, Erinç İnci for being the other MAC guy in the world of open-source, Güner Orhan for being such an excellent internee, student and friend besides his breakfasts during ROSSI coding all-nights, Hande Çelikkanat for her joy and being a good fellow-traveller during Humanoids'11; and Doruk Tunaoğlu for the exciting Quake 3 Tournaments. I am also thankful to Sinan Kalkan for his supervision and comments during ROSSI project. Last but not least, I gratefully acknowledge the "previous generation" of KOVAN members, Barış Akgün, Nilgün Dağ and İlkay Atıl for leaving a strong research foundation to our generation.

I would like to send my special thanks to my family. Mom, little sis and dad without your love and support, I couldn't accomplish this far.

I also want to acknowledge the support of my PhD supervisor, Michael Beetz for giving me time and place to continue working on this thesis in the last three months, eventhough I started work with him. Without this privilege, I couldn't finish this thesis.

Finally, I acknowledge the support of TÜBİTAK (The Scientific and Technological Research Council of Turkey) BİDEB 2210 graduate student fellowship during my M.Sc education and this thesis is supported by TÜBİTAK under the project, 109E033 and European Commission under ROSSI project, FP7-216125.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Humanoids are envisioned to be our partners in every-day life. Towards this end, they should be able to cope with the dynamically-changing complexity of every-day life and accomplish tasks as we, humans, do in such environments. "Developmental Robotics" suggests a cognitive development for robots like infants do [1]. One approach towards this development is equipping robots with sensors, actuators and a set of core behaviors that are necessary for building a sensorimotor system.

Infants have similar core equipments as a prerequisite of their developments. For example, motor development studies show that during the early stage of grasp learning, which takes place in the period of 3-6 months, infants tries to reach and grab randomly. Their reaches are initiated by the vision of the object but not so much guided by it. After contact with the object, tactile based grasping takes place, molding the hand around the object [2]. When they reach 9 months of age, visual input is incorporated in the grasp planning and execution. This developmental progression suggests that earlier tactile grasping leads to advanced grasp planner that utilizes visual input and incorporates this into the grasp plan. From the age period of 9 months, infants start to pay attention to other humans while they are showing some important features of outer world. This leads to a supervised learning mechanism which plays an important role of infants' development.

An intelligent robot should also be able to accomplish such a developmental learning. Namely, starting from a set of crude behaviors and raw sensory data, it should gather information about itself and things or people around itself that are consistently changing and process this information. By using developmental learning, it should learn how to adapt the environment and how to apply its behaviors to reach a certain goal.

1.1 Problem Definition

From an engineering point of view, we try to make humanoids earn the ability of grasping objects with different sizes and shapes. Although tactile sensors of some robotic arms give a feedback for how much pressure the arm is applying the object, they are not adequate for a successful grasp learning scheme on humanoids. In addition to applying the right amount of force, the robot should also know how to approach objects. As seen on Figure 1.1, the wrong position of the arm can result in unsuccessful grasp performances.

We focus on this problem and propose a learning scheme that is inspired from the findings of psychology and cognitive sciences. We tested this scheme on a 53 DoF humanoid robot, *iCub*, which initially has only 3D perception and a basic grasp controller. This controller is responsible for getting the robot's palm to a certain cartesian coordinate and closes its hand until the tactile feedback reaches the threshold level. From this starting point, our learning scheme matches 3D features with grasping ability in a two-phase learning. With this, the robot can infer how to approach novel objects in order to grasp them successfully. While doing that, we also want to develop a human-like development scheme for humanoids.

1.2 Objectives and Motivation

In this thesis, we investigate how a robot can learn to apply grasping behavior to objects of different size and different shapes in an adaptive manner. Towards this end, a two-stage developmental learning scheme is proposed for *iCub*. In *Stage-1*, *iCub* learns how to approach the objects in the pre-specified locations by assigning random offsets in three dimensions. In *Stage-2*, *iCub* tries to generalize what it learned from *Stage-1* by removing the boundary of the pre-specified locations and with the help of a caregiver.

After learning, the robot successfully grasps set of novel objects in a human-like manner. By accomplishing this, the objective is to show that robots can go through a developmental learning process just like humans, where they can adapt their environment and show intelligence through their sensorimotor system.

While proposing this developmental learning, the *affordances concept* [3] and the *scaffolding theory* [4], which are hypotheses of psychology, supply a theoretical information about how

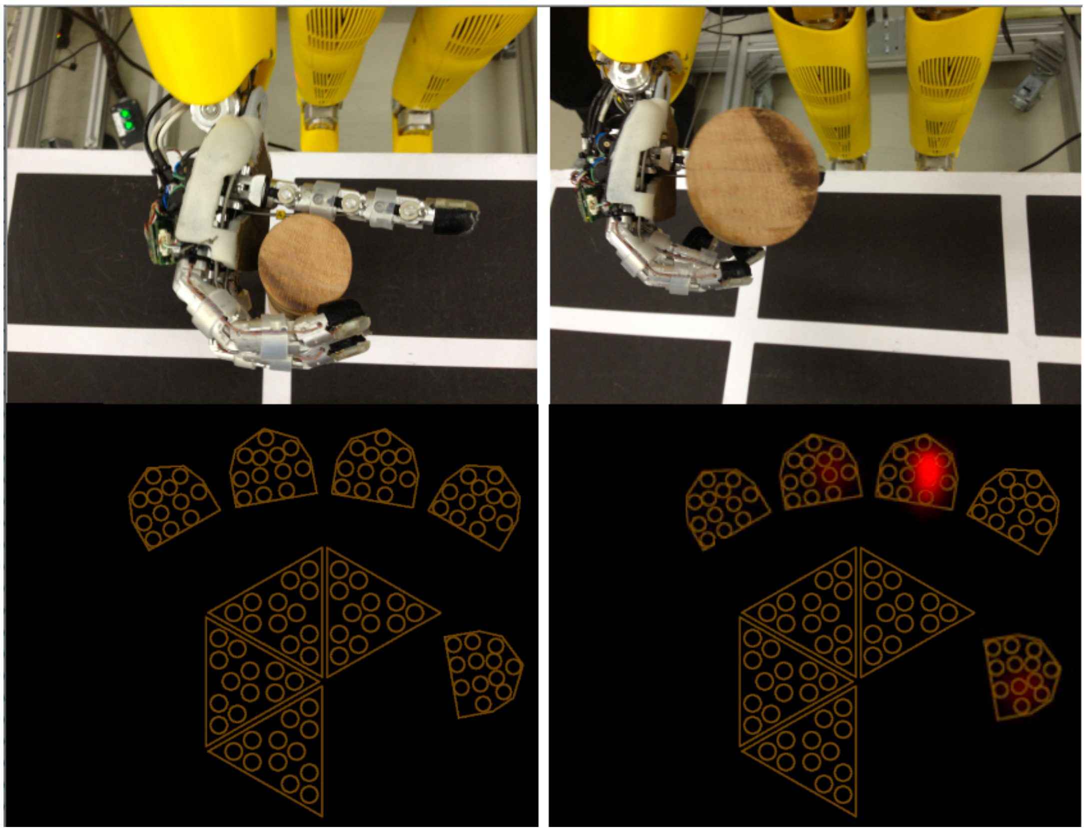


Figure 1.1: On the left, *iCub* cannot grasp the cylindrical object since the object is too close to its palm and fingers can not apply necessary amount of force. Thus, the object slides. On the right, the object is at the level of fingertips. As a result of this, necessary amount of pressure is applied and *iCub* successfully grasp the object. The lower figures show the output of tactile sensors at the time of the grasp.

learning is occurred in humans' life.

1.3 Relation to other work and contributions

This thesis is a result of developmental robotics studies of KOVAN Research Laboratory that have been conducted over years. Since the establishment of KOVAN Research Laboratory, developmental learning of robots has been studied with different kinds of case studies, such as traversability and verb concepts, over the years. In this section, we present these studies and explain the relation between them and this thesis.

The theoretical background of all these studies were the *affordances concept*. In [5], Şahin et al. formalized this concept for developmental robotics. Thereafter, they studied affordances for different behaviors using this formalization.

In the scope of MACS Project [1], the traversability affordance of a mobile robot, KURT 3D, was studied. More specifically, a mobile robot's learning of the features of the environment that affect the robot's traversability between two points was investigated. In [6], Çakmak et al. proposed an affordance-based robot control framework. Doğar et al. used primitive behaviors and learned affordances to learn applying a goal-oriented behavior[7]. Uğur et al. [8] used a curiosity-driven methodology for the traversability learning that identifies the unexplored regions in the search space and makes extra experiments to identify its traversability affordance in these regions. In [9], Uğur et al. used a range camera for learning the traversability affordance. In this thesis, we used same affordance formalization[5] used in these studies.

Learning adjective, noun and verb concepts and the social affordances of a humanoid robot were the focus of ROSSI Project [2]. By using the same affordance formalization, these subjects were investigated. Atıl, Dağ, Kalkan and Şahin published results on learning verb effect prototypes[10] and [11]. In [12], we proposed a methodology for learning adjectives and nouns based on affordances of corresponding objects. After this, we proposed a social affordance learning methodology based on learned verb prototypes, nouns and adjectives [13]. In [14], Uyanık et al. enhanced adjective learning scheme by including the notion of context. Finally, Yürüten's M.Sc. thesis [15] explains the overall framework in detail. In this work,

[1] http://www.macs-eu.org
[2] http://www.rossiproject.net

we used same robotic platform, perception system and experimental setup with these studies.

The work presented in this thesis has partially appeared in [16], [12] and [13].

1.4 Outline of the Thesis

The rest of the thesis is organized as follows: In the following chapter, we present some fundamental concepts and works of the ecological psychology, especially in the notion of *affordances*. Then, the related works in robotics that uses this notion are presented. After that, in Chapter 3, we review the second important psychological notion that is used this study, namely, environmental psychology and the notion of *scaffolding*.

After giving the psychological background related to this work, we present fundamental works on the robot learning, the affordance development for the robots and grasp learning in Chapter 4. In Chapter 5, the experimental framework is presented. The tools that were used in the experiments, the software components and architecture, the testing methods and the learning methodologies are discussed in detail. Thereafter, in Chapter 6, results of the experiments are given for each grasping type. Finally, in Chapter 7, the outcomes of the experiments are analyzed. In addition, some possible future works are mentioned.

CHAPTER 2

ECOLOGICAL PSYCHOLOGY AND AFFORDANCES

Ecological psychology studies the relation between an agent and its environment. James J. Gibson was one of the pioneers of this area along with Roger G. Barker and Herb Wright. In his work, Gibson mentioned the importance of the outer world, specifically, the perception of the environment supplies the information of possible results of its actions to the organism. Therefore, in order to explain a perceptually-driven behavior, an analysis of the environment is crucial. According to Gibson, the agents are dependent to their environments so that to explain a behavior it is necessary to study the relation between an agent and its environment [17][18][3].

The rest of this chapter is organized as follows: Firstly, Gibson's views on ecological psychology is presented. Then, Gibson's *affordances notion* is analyzed in the scope of ecological psychology. Afterwards, views of Barker, who was another pioneer of Ecological psychology, findings are given. Thereafter, affordances-related studies in developmental robotics are mentioned. Finally, how *affordances notion* is adopted in the proposed learning scheme is presented.

2.1 Gibsonian Ecological Psychology

In [17], Gibson summarized his view on the role of environment in psychology with:

> "A living animal can orient itself in many ways. All of these are orientations to the environment, but to different features of the environment, such as gravity, or the sun in the sky, or a sudden noise, or a mate."

He stated that perception is “information-based” rather than “sensation-based”. This means that information about the environmental features such as affordances is the basis for the ecological psychology. Gibson described the environment as the *surroundings* of organisms that perceive and behave [18]. It consists of objects and animals. Animals are different from objects since they are both the *perceivers* and the *behavers* of the environment. For an animal, every member of the outer world counts as the member of the environment. Furthermore, in [18], he argued that “The words *animal* and *environment* are inseparable pair” and, therefore, the animal cannot be taken and analyzed isolated from the environment.

According to his views, the environment has “substances”. In order to behave accordingly, one should identify these substances. A usual way of identifying is seeing substances’ surfaces. By doing so, some important features such as its layout, its texture and being lighted or shaded can be obtained. On the other hand, they perceive not only the raw physical and mathematical features of substances but they carry also high level information such as action possibilities.

2.2 Affordances

Gibson introduced the term *affordances* to describe things that the environment offers:

> “The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill. The verb to afford is found in the dictionary, but the noun affordance is not. I have made it up. I mean by it something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment.” (J.J. Gibson, 1979/1986, p. 127)

He argued that animals do not perceive the environment by the means of substances’ raw physical properties. Rather, they perceive these properties by taking into account of posture and behavior ability of the self. Different physical properties afford different behaviors for different. For instance, a chair is *sit-able* for a human being while it is *climb-able* for a rat. An *affordance* is what is offered by the environment. In case of the above example, *sit-ability* is the affordance which is offered to the human whilst *climb-ability* is offered to the rat.

According to Gibson, while investigating affordances, animals and the environment they live in, cannot be considered inseparable. While affordances are modifiable by animals after a while, these modifications are constrained by the environment as well. For example, a human can succeed to jump higher by practicing everyday but he/she cannot suddenly start flying while practicing. The law of gravity constrains *fly-ability* affordance of humans.

Gibson stated that affordances are not processed by animals. According to him, they are perceived by the environment as it is:

> "The perceiving of an affordance is not a process of perceiving a value-free physical object to which meaning is somehow added in a way that no one has been able to agree upon; it is a process of perceiving a value-rich ecological object."
> (J.J. Gibson, 1979/1986, p. 140)

Finally, he mentioned that while organisms perceive possible actions from the environment, they only take into account relevant perceptual data which is called as *perceptual economy*.

2.3 Barker's Ecological Psychology

Barker's opinions about how the environment affects animals were less philosophical than Gibson's. In [19], he discussed that human behaviors were dependent to the current conditions. In other words, one cannot predict how a human behaves unless he/she knows the exact context of the corresponding human. For example, a police man's behaviors in his spare time and during the work are completely different. During work, he should obey the certain rules and behave appropriately. Moreover, Barker and Wright stated that children's behaviors were predicted more accurately under known conditions than under known characteristics of children [20]. The experiments reported in study showed that children's behaviors within a day were varied by the immediate outside factors. Also, under similar conditions, different children tended to behave in a similar way. This result is also significant from the developmental robotics' perspective since they reveal the impact of the environment during the development of humans. If most of behavioral developments take place under the influence of outside factors, a similar developmental learning scheme, that takes the environmental factors into account, may also be proposed for robots. Such learning schemes can make robots that have similar core facilities with humans to advance in same progress.

2.4 Applications of Affordances in Developmental Robotics

Affordances, as an inspirational notion from psychology, has received interest from many roboticists in recent years. Fitzpatrick and Metta et al. [21][22] proposed an affordance learning framework for roll-able objects using computational vision. In these studies, the robot learned in which direction the object will roll when it applies a *push* action. Later on, Montesano et al. used Bayesian networks to map the relationship between objects, agents and actions [23].

In [5], Şahin et al. formalized an affordance as a triple relationship:

(effect, entity, behavior)

representing that when the agent applies the *behavior* on a certain *entity*, the given *effect* is generated. The term *entity* stands for an object or any environmental entity that the robot interacts with it. The term *behavior* denotes which action is taken by the robot towards the entity. Finally, the *effect* represents the result of the behavior. An instance of this formalization would be:

(rolled, blue-ball, push)

After certain number of experiments, the robot should learn the color of the ball is not related with the result of the pushing action. Thus, the three-tuple updated as:

*(rolled, *-ball, push)*

Using this formalization, Şahin et al. implemented a traversability learner system for a mobile robot [6][7][8]. Firstly, in [6], they proposed an affordance-based robot control framework. In a later work [7], they presented a formalization for learning how to accomplish a goal from primitive behaviors and learned affordances. Thereafter, they used a curiosity-driven methodology for improving the performance of the traversability learning system [8].

Learning how to use a tool is another application of affordances in robotics. Firstly, in [24], Stoytchev coined this subject in the developmental robotics field. He shows how an affordance representation can be utilized for solving the outcomes of tool-using tasks by dynamically chaining a set of behaviors. In [25][26], Sinapov and Stoytchev studied the learning of affordances of several tools on certain objects. Recently, Çalışkan et al. used 2D visual features for discovering which affordances are provided to the robot by the given tool.

The works related to the object action complexes (OACs) concept, which links the relationship of objects and actions for robots, may also be taken as affordances applications. Huebner et al. used a computer-vision based shape attributes to make the robot learn which results it can get after applying pre-defined actions [27]. In [28], Kraft et al. studied grasping using this concept. By using a selection of visual features for the given object, they make the robot to learn how to grasp. In a later work, Petrick et al. extended this work for the other behaviors such as moving an object to somewhere and filling. Geib et al. used OACs to fill the gap between low-level robot controlling architectures and high-level AI-based control [29]. In [30], Woergoetter used this concept to make the robot being able to predict the results of its action prior to apply.

The literature related to learning grasp affordances are given at the *Chapter 4* in detail.

2.5 The Use of Affordances in This Thesis

In this work, we use slightly modified version of Şahin et al.'s affordance formalization [5]. The original version of the formalization has the following triple relationship:

(effect, entity, behavior)

representing that when the agent applies the *behavior* on a certain *entity*, the given *effect* is generated. The term *entity* stands for an object or any environmental entity that the robot interacts with it. The term *behavior* denotes which action is taken by the robot towards the entity. Finally, the *effect* represents the result of the behavior. In this work, we used a specialized version of this relation which is:

(effect, entity, (x, y, z))

where *effect* can be *grasped* or not grasped and a behavior is represented with a set of parameters.

CHAPTER 3

DEVELOPMENTAL PSYCHOLOGY AND PARENTAL SCAFFOLDING

Developmental psychology focuses on systematic changes and improvements over the life-span of animals. Since most of these changes take place at the period of being infant, children and adolescence, developmental psychologists mostly study these periods.The subjects of this field include motor skills development and cognitive development such as problem solving, emotional development, language learning, and self-awareness.

The rest of this chapter is organized as follows: Firstly, the pioneering works on developmental psychology are introduced. Afterwards, parental scaffolding concept is presented in the scope of developmental psychology. Finally, scaffolding related works in robotics are mentioned.

3.1 Developmental Psychology

The notion of development has been investigated by many psychologists. Many influential theories have built upon these studies such as those by Piaget [31], Vygotsky [4], Bandura [32], Erikson [33] and Kohlberg [34].

Piaget proposed *Stage Theory* [31] as a comprehensive theory about the nature and development of human intelligence. He studied how knowledge is acquired, synthesized, and used by humans emphasizing the constructive structure of intelligence. The activeness was justified with actions. Since the thinker changes the environment by their actions, intelligence should be adaptable to this changes. It is also constructive because mental states of humans are coordinated by inclusive and cohesive systems, which lead to reasoning.

Vygotsky, whose works originated social constructivism theory, claimed that parents' interventions during learning of a new task or concept can make the whole process easier for the child [4]. These interventions were later named as "scaffolding". Scaffolding helps to build a knowledge on children by guidance of a parent. This guidance narrows the domain space of knowledge either by communication or by directly showing. Vygotsky studied the cultural role on the child's pattern of development.

Social learning theory was built upon Bandura's work whose claims were the observational learning depends on three model [32]:

(a) "A live model": a caregiver demonstrates how a behavior is done properly.

(b) "Verbal instructions": a caregiver describes the behavior in detail, instructs the learner how to accomplish it, and tells its outcomes.

(c) "Symbolic modeling": learning is occurred by observing the media, such as movies, television, Internet, literature, and radio.

Erikson's stages of psychosocial development are eight stages which a developing human should pass each from infancy to late adulthood. In each stage the person faces new challenges [33]. Each stage has a "psychosocial crisis" between two conflicting forces . If the person reconciles these forces successfully, he/she emerges from the stage with the corresponding virtue. If the challenges of a particular stage are not overcome, it may cause disorders in the future.

Finally, Kohlberg was the first one who attempted to describe development in moral reasoning. In his theory, there are six distinct developmental stages [34]. In each stage, humans advances themselves at coping with moral dilemmas. These six stages can be grouped into three: pre-conventional, conventional and post-conventional.

3.2 Parental Scaffolding

Scaffolding Theory was initially coined by Jerome Broner in 1950s to describe the language learning process in children. Children in early ages are instructed by their parents. Bed-time stories and read alouds were given as examples of these instructions by Daniels [35].

Later, Vygotsky's (1978) concept of an expert assisting a beginner enhanced the term *scaffolding* [4]. In his concept, he defined the process of learning under the support given to a learner by an experienced adult. In their work, Wood, Bruner, and Ross (1976) supported the idea of *scaffolding* in Vygotsky's works [36].

Semiotic scaffolding has been introduced by Jesper Hoffmeyer as an improvement of Vygotsky's work. In his work, Vygotsky never used the 'scaffolding' word. Hoffmeyer was the first one who used the term *scaffolding* as the interactional support and the process by adults for mediating a certain behavior of a child to grasp the proper way of achieving that behavior. According to him, *scaffolding* describes interactions between adult and child that enable the child to learn something beyond his/her independent efforts.

During a child's development process, the support of his/her parents has an important effect. His later developmental learning and expression of his/her abilities greatly depend on this orientation and assisting given by the parents [36][37].

Zhao and Orey [38] identify these five general features of the scaffolding process as:

(a) "Sharing a specific goal": The caregiver should define a goal for the learner prior to scaffolding. While determining this goal, the child's interests must be taken to the consideration in order to achieve a better learning rate.

(b) "Whole Task Approach": The learning task should be atomic and understandable as a whole by the learner. Instead of a sequence of subgoals, each learning item should have a separate task.

(c) "Intention-assisting": Helping children when they face difficulties during the learning is another essential part of the scaffolding. Providing this immediate help will increase the learning rate.

(d) "Optimal Level of Help": The child should be assisted to overcome the current difficulties. On the other hand, the level of help should not make the learner not to participate in the learning.

(e) "Conveying an Expert Model": An expert model can be used to provide an example of the task of interest.

3.3 Applications of Parental Scaffolding in Developmental Robotics

Parental Scaffolding is a relatively new concept for developmental robotics which has been investigated for nearly half of a decade in psychology. In [39], Nagai et al. proposed a methodology that the caregiver evaluates the robot's performance in a metric while it is accomplishing a task. By looking its current performance metric, the robot also learns which visual features affects learning performance more. Muhl and Nagai assessed the naturality of robot-scaffolding in humans' perspective in [40]. In this study, they compared the results of human-human-interaction with the result of human-robot-interaction while scaffolding. In another study [41], Nagai and Rohlfing pointed out that one of the difficulties in robot scaffolding is that robots do not know where to attend during interaction. For this, they proposed a demonstration method, *motionese*. In [42], Nagai et al. claims that bottom-up attention of robots based on visual salience make partners to exaggerate their body movement and segment their movement frequently, which are so called "homologous to modifications in infant-directed action". Recently, Ugur et al. [43] published a study using this concept for the grasp learning. In this study, the robot learns to identify grasping regions of objects indirectly by modifying reaching trajectory.

3.4 The Use of Parental Scaffolding in This Thesis

We use *Parental Scaffolding* concept in our proposed learning system as an error-correction mechanism. Firstly, the robot itself tries to learn how to grasp objects at fixed locations. After learning to grasp in these locations, it tries to generalize what it learned. In other words, it starts trying to grasp objects at random locations. In such a scenario, it is obvious that the robot will do some errors while grasping. At this point, the caregiver corrects the position of its arm prior to grasping. With such corrections in the training phase, the robot generalize its grasping ability to the random positioning of objects.

CHAPTER 4

LITERATURE SURVEY

4.1 Robot Learning

In this section, studies in robotics using similar learning approaches with this thesis are presented. Firstly, imitation learning studies in robotics is mentioned. Consecutively, works on identifying relevant perceptual regions/cues/features in the environment are presented.

4.1.1 Imitation Learning

In the first half of 1980s, the imitation learning had become popular in robot-learning, especially for the manipulation applications. Instead of programming the manipulators for certain tasks, learning these tasks by *symbolic reasoning* was studied [44][45][46][47]. In these works, there exists a training phase in which many examples of executions of tasks are executed with a manual manipulation. In this phase, feedback from the environment such as sensor readings and position of the arm is stored. In order to imitate given instances, they are segmented into subgoals. While segmentation, all demonstration are taken into considerations so that uncertainties in the environment are formalized into some symbolic rules. Later on, new feedback techniques are applied on this scheme such as *computer vision* [48][49][50] and *marker-based observation* [51].

There are many different demonstration techniques used in literature as well. Firstly, demonstration using human teleoperation via a joystick is used many different areas such as flying an autonomous helicopter [52], robotic arm assembly tasks [53] and obstacle avoidance [54]. Kinesthetic teaching is used for teaching desired motions to a humanoid robot[55].

Instead of a human, hand-written training controllers are also used to teleoperate robots [56][57][58][59].

Another demonstration technique is the external observation of another agent performing the desired task. Atkeson and Schaal [60] used a stereo vision system to teach a robot arm how to balance itself by observing human. The motion capture systems (mo-cap) are also used for observation. Ude et al. taught basic human motions by recording human motions via a mo-cap [61]. In [62], Pollard and Hodgins proposed a methodology that uses a motion capture system to teach how to manipulate certain objects. There are works on learning based on observing non-human performers as well. In [63], Dillmann et al. presented a system that teaches a robot to a certain manipulation task by observing another identical robot. A generic framework in which a robot learns new behaviors by observing and imitating another physically different agent is introduced by Alissandrakis et al. [64].

4.1.2 Identifying Relevant Features

In this thesis, our learning methodology, firstly, identifies relevant features for grasping, and then, uses them to learn how to grasp. Identifying relevant regions/cues/features in the environment or of an object for a certain task has been investigated by many roboticists as well. In [65], features used for simultaneous localization and map building are filtered out to increase the speed of the process. The features that increase uncertainty in robot localization are filtered out using an entropy-based method. For navigation of the robot in the environment, Bur et al. proposed a system that selects the features that are persistent and reliable over the course of previous runs [66]. Evolutionary algorithms used by Floerano et al. in order to control a mobile robot by motor outputs of the network, which are activated by the weighted sum of perceptual inputs [67]. In his doctoral dissertation [68], Edsinger showed that using only task-relevant features while learning manipulation reduces dramatically the number of training examples. Vlassis et al.[69] proposed a supervised linear feature extraction and identification method for identifying task-relevant features. They claimed that this method is more efficient and many times faster than PCA feature extraction together with a descent feature selection algorithm. In [70], Adaboost algorithm [71], which is a learning algorithm that contains a feature selection process by itself, was used for hand-posture recognition.

4.2 Grasp Learning

Developing the ability of grasping has been a popular subject since the last decade. For this manner, there have been joint efforts in the disciplines of computer-vision, mechatronics and machine learning/artificial learning . Some significant works on this topic include: In [72], Coelho et al. propose a formalization for capturing dynamics of a dexterous hand while grasping and matching them with the visual features of the object-of-interest. A stereo vision-based 6-D object localization prior to grasping is presented [73]. A methodology for a walking robot's vision-based grasp learning is described in [74]. In [75], a gloves with tactile sensors is introduced for further advancement in robot-grasping and manipulation. Huebner et al. present a grasp learning scheme using 3D perception in [76].

Towards the learning of grasp affordances, there are two main approaches in the literature. The first approach is skill acquisition from a human partner. In [77], Öztop et al. presented the adaptive structure of body schema inside the human brain. Towards this end, they controlled a 16 DoF robotic hand by a human experimenter. In a more recent study [78], Uğur et al. proposed a scaffolding framework to learn grasping from a human caregiver. The other type of skill acquisition is through imitation. Goal-oriented sequences of primitive actions for grasping were generated by the robot by imitation [79]. In [80], for each object, the demonstrator shows how the object can be grasped. Then, each of these grasps are clustered together in a similarity metric of the hand's pose with respect to the object. Geng et al. reduced the number of postures that humans make during grasping into three synergies and trained a neural network to match these synergies with object properties [81]. In [82], a common language is established for communication between the robot and the human partner for grasp skill acquisition.

The second approach is robot experimenting grasping by itself without any skill transfer from human. In [83], Salganicoff et al. proposed the *IE-ID3 algorithm* which learns how to approach objects while grasping. Kraft et al. described a cognitive robotic system that explores object definitions and associates them with object-specific grasp behaviors [84]. Detry et al. proposed a framework for developing grasp affordance from 3D experience [85]. They defined a grasp density metric which shows the object's relative grasp affordance. By random grasp attempts, the robot experiments how it should grasp the corresponding object. Then, a batch learning mechanism was used to learn grasp affordance of the robot. Detry et al. also presented a hybrid approach that the grasp affordance can be learned both by imitation and skill acquisition [86].

CHAPTER 5

PROPOSED LEARNING METHODOLOGY

In this chapter, we give the details of our learning scheme. Firstly, we present the feature extraction methodology. Then, we describe the feature selection algorithm used. Finally, we present the used learning algorithm and the proposed learning scheme.

5.1 Feature Extraction

We use 3D perception in the proposed learning scheme. After getting the point cloud of the scene from the perception device, we process it and extract some object features. Since grasping is a behavior that should be adaptive to the object's size, shape and position, we extract features to describe these properties of the object. The process of feature extraction starts with the segmentation. We use PCL's tabletop object segmentation function [87] to identify the points of object-of-interest. This function is based on a simple, computationally-inexpensive methodology. Firstly, it uses SAC segmentation [87] to identify the table plane in the point cloud. After that, it uses Region-Growing algorithm [88] to identify the objects at the top of the table. Then, the features are extracted from the object of interest's point cloud. There are three shape and position related classes of features. Since grasping mainly depends on the shape and the pose of objects, we choose extracting as much features as possible to define these properties of the object in a complete manner.

After segmentation, a bounding box is fitted to the object, such that each of its edges contains a point whose x, y or z coordinate equals to whether the minimum or the maximum in respective coordinate at the object's point cloud. Bounding boxes are used for visualizing whether the segmentation is properly and extracting the first seven features, namely, the size of the bounding box in three axes and the location and the orientation (only yaw angle) of the bounding box's center in space.

The second class consists of normal histograms. Using PCL's integral image normal estimator [89], the surface normals of the object are estimated (Figure 5.2B). After this, each normal histogram's azimuth and zenith angles are calculated. The azimuth angle corresponds to the angle between z-axis and the normal vector's orthogonal projection on x-z plane, whereas the zenith angle is the angle between y-axis and the normal vector. After the zenith and azimuthal angles are computed for each normal, angular histograms, which contains 20 bins, are computed. Each of the bins represents an interval of 18°.

The third class of features consists of shape index histograms. The metric of shape indices is a measure of shape information in the range of $[-1; 1]$ and calculated using principle curvatures, which are the eigenvalues of the extrinsic curvature at the given point:

$$S = \frac{2}{\pi} tan^{-1}(\frac{k_1+k_2}{k_2-k_1})$$

k_1, k_2: principle curvatures

The minimum curvature, k_1 and the maximum curvature, k_2 histograms are also kept as features. Finally, for all histogram features, we also keep minimum, maximum, average and deviation value of the histogram.

A example feature visualization for the given simulated objects is seen in Figure 5.1. Additionally, an example feature histograms can be seen in Figure 5.2 for the real-world case.

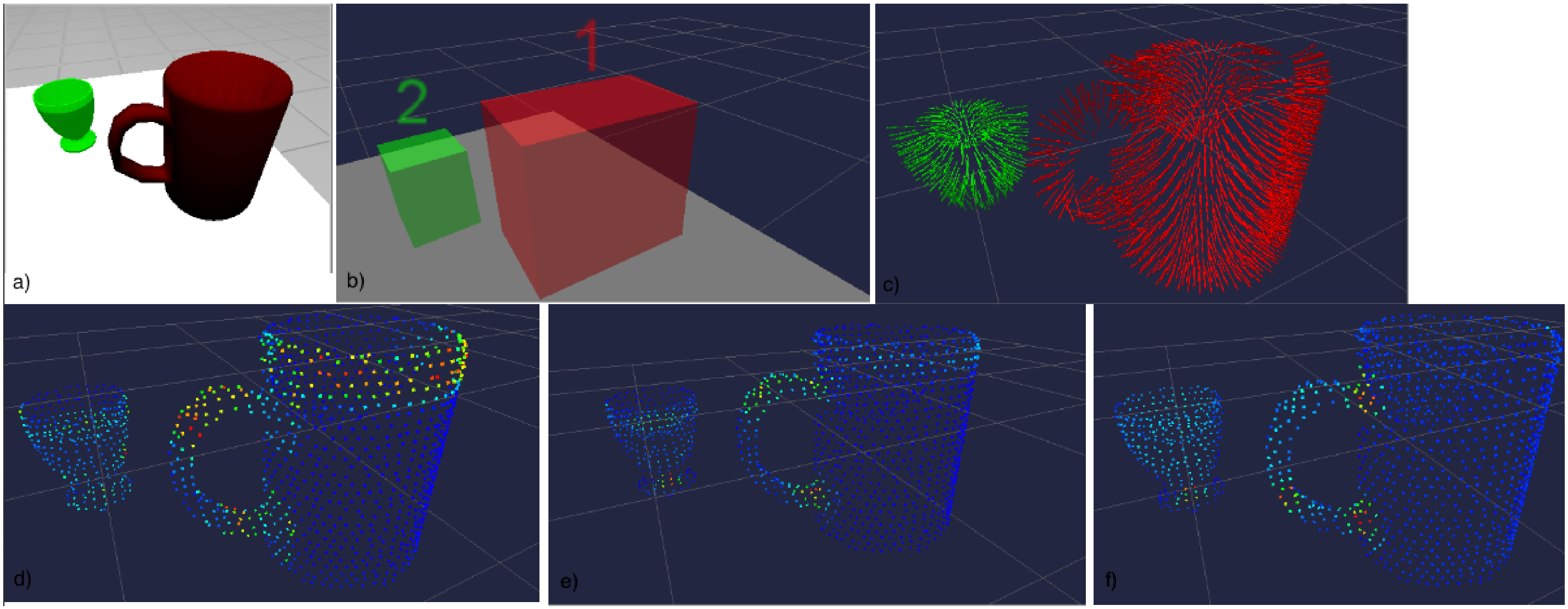
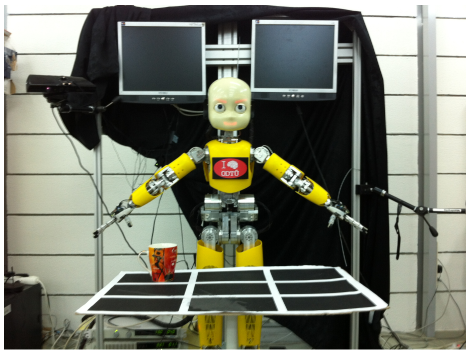
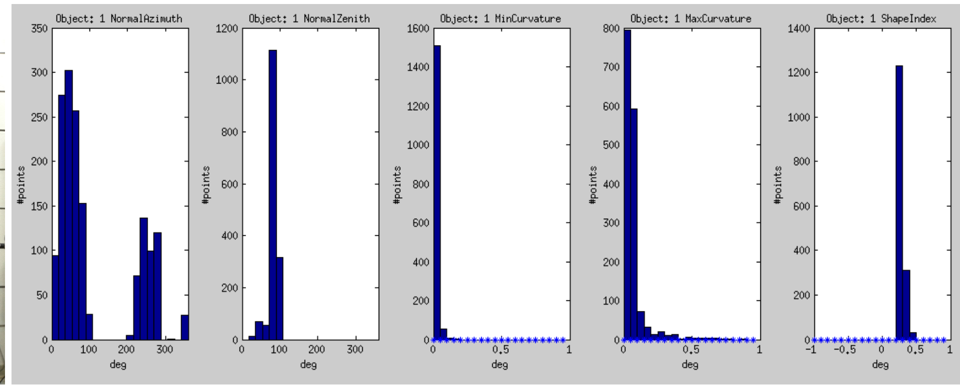


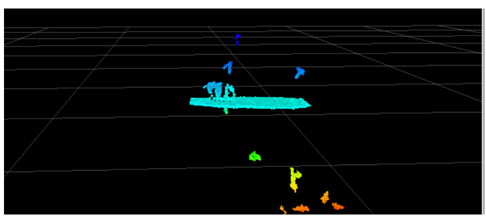
Figure 5.1: a) Simulated objects b) Bounding boxes c) Surface normals d) Maximum curvatures e) Minimum curvatures f) Shape indices.



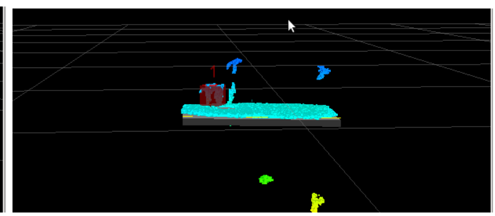
(a) The position of the object



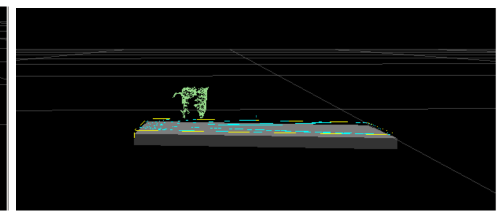
(b) Feature Histograms



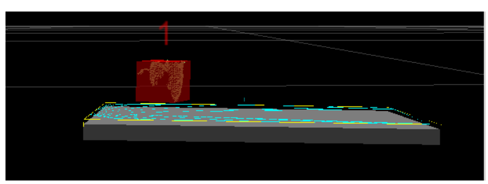
(c) Raw point cloud



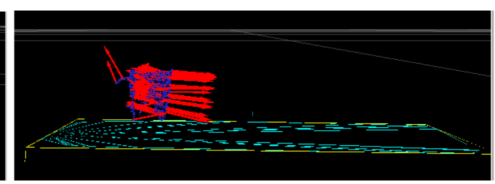
(d) Bounding boxes that are shown in the raw point cloud



(e) Segmented point cloud



(f) Bounding boxes that are shown in the segmented point cloud



(g) Surface normals of the object

Figure 5.2: An example usage of the feature visualization system when the object is at the position of *(a)* .

5.2 Feature Selection

For the selection of relevant features, ReliefF algorithm [90] (Algorithm 1) is used. The algorithm finds the relevance of each feature by observing the impact of its change on the labels (which are offsets in our case). According to their impacts, the algorithm assigns a relevancy weight of each feature.

Algorithm 1 ReliefF Algorithm

n_f: number of features

w_d: weight of d^{th} feature

m: number of iterations

f_l: the feature vector observed in the l^{th} training instance

$f_l[d]$: the normalized value of d^{th} feature in the l^{th} training instance

  $w_d \Leftarrow 0$, where $1 \leq d \leq n_f$

  for $i = 0$ to m **do**

    Select a random feature vector f_l from $\{<r_k^i, f_k, b^i>\}$

    Compute distance of f_l to all $\{f_k\}$

    Find 10 feature vectors closest to f_l with the execution results r_l^i. Put them into set of nearest hits, Γ. ($\Gamma = \{f_{1'}, ..., f_{10'}\}$)

    Find 10 nearest feature vectors with execution results different from r_l^i, i.e. and put them into set of nearest misses, Υ. ($\Upsilon = \{f_{1''}, ..., f_{10''}\}$)

    for $d = 0$ to n_f **do**

      $w_d \Leftarrow w_d - \frac{1}{10m} \sum_{j=1}^{10} |f_l[d] - f_{j'}[d]| + \frac{1}{10m} \sum_{j=1}^{10} |f_l[d] - f_{j''}[d]|$

    end for

  end for

5.3 Learning Algorithm: Gaussian processes for regression

Gaussian processes (GPs) are an extension over multivariate Gaussian distributions (see *Appendix A* for details on Gaussian distributions and Gaussian processes) to infinite dimensionality. A Gaussian process creates data in a given interval such that these data can be represented with a multivariate Gaussian distribution.

Gaussian processes for regression (GPR) is a GPs-based learning algorithm that is used for the learning grasping from the selected features. GPR slightly modifies classical regression approach: Rather than specifying the type of the regression function, a set of Gaussian processes represents the regression function in a loosely-coupled manner.

Overall, GPR has following steps for learning according to [91]:

(a) After observing n data instances, each of these instances can be treated as a single point sampled from a multivariate Gaussian distribution. Then, each instance can be associated with a Gaussian process. It is assumed that mean of this GP is zero everywhere. The relationship of two instances in the process is defined by *the squared covariance function*:

$$k_w(x, x') = \sigma_f^2 \, exp(\frac{(-(x-x')^2)}{2l^2}),$$

where x denotes an observed instance and the maximum allowed covariance is σ_f^2. Thus, if x is distant from x', we have $k_w(x, x') \approx 0$, i.e. the two instances do not affect each other. During the prediction of new outcomes, distant observations will have negligible effect. How much effect two instances will have on each other also depends on the length parameter, l.

(b) The data collected during the training is noisy. Therefore, a Gaussian noise is added the regression function, $f(x)$.

$$k(x, x') = k_w(x, x') + \mathcal{N}(0, \sigma_n^2),$$
$$\mathcal{N}(0, \sigma_n^2) = \sigma_n^2 \delta(x, x')$$

Then we will have the following covariance function:

$$k(x, x') = \sigma_f^2 \, exp \, (\frac{(-(x-x')^2)}{2l^2}) + \sigma_n^2 \delta(x, x'),$$

where $\delta(x, x')$ is the Kronecker delta function (see *Appendix A*).

(c) In order to calculate GPR, we calculate the covariance function among all possible combinations. Then, these calculations lead to three matrices where x_* stands for a new data instance that is not in the training set:

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \cdots & k(x_*, x_n) \end{bmatrix} \text{ and } K_{**} = \begin{bmatrix} k(x_*, x_*) \end{bmatrix}$$

Since the key assumption for GPs is fitting the data to a multivariate Gaussian, we have:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} K & K^T \\ K_* & K_{**} \end{bmatrix})$$

Thus, the conditional probability $p(y_*|y)$ is a Gaussian distribution,
$\mathcal{N}(K_* K^{-1} y, K_{**} - K_* K^{-1} K^T)$.

(d) The best estimate for y_* is the mean of this Gaussian:

$$\overline{y_*} = K_* K^{-1} y.$$

5.4 Learning

5.4.1 Learning with Offsets

In the proposed learning methodology, we do not explicitly show the robot where the grasping regions are located at the objects. Instead we are using *offsets* (Figure 5.3) that refer where the palm should be located from the relative surface of object-of-interest prior to grasping. The arm reaches this surface with respect to the given/learned offsets and closes its hands to grasp the object. Thus, we have a following equation for reaching:

$$\text{reach}(x_c + x_o, y_c + y_o, z_c + z_o, \theta, \phi, \gamma)$$

where (x_c, y_c, z_c) is the surface's center point, (x_o, y_o, z_o) is offsets given to the system and, finally, $(\theta,\ \phi,\ \gamma)$ is the orientation of the palm which changes according to different grasp

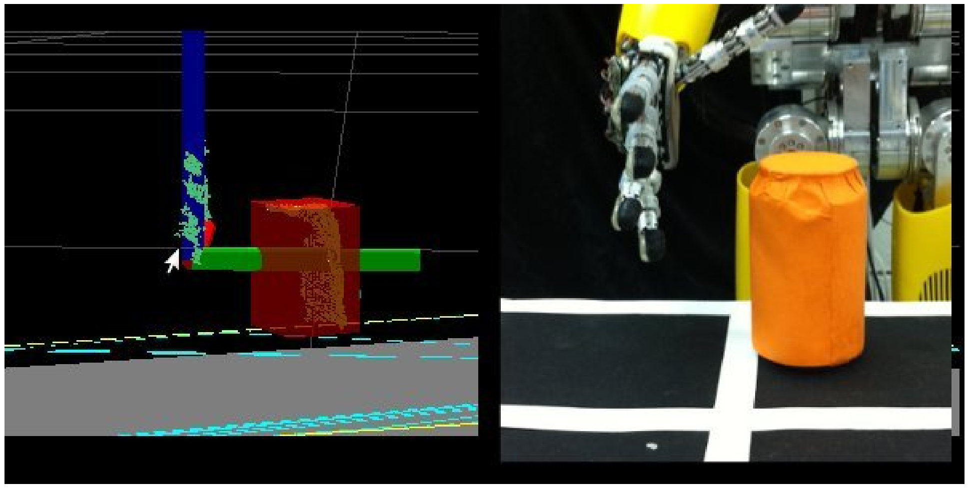


Figure 5.3: The bounding box of the object of interest and the reference frame of the hand. The red line represents the *x-axis* of the object, the green line represents the *y-axis* and the blue one is the *z-axis*. By giving different offsets in these axes from the bounding box's right surface's center, grasping attempts are made by the robot.

styles. For example, the relative surface would be the top surface for *top-grasp* style, grabing the object from its top. Thus, the orientation of the palm would be $(0°, 0°, 0°)$. On the other hand, for *side-grasping* from the right surface, the palm should turn counter-clock wise 90° since the palm should be parallel to the right surface as in Figure 5.3. Thus, the orientation should be $(90°, 0°, 0°)$.

5.4.2 Learning Scheme

In this thesis, a two-stage learning is proposed. In the first stage(*Stage-1*), an affordance-learning based grasp learning is implemented. In the proposed framework, the robot itself tries to grasp objects at the pre-determined locations by approaching random offsets at this stage. Then, the robot detects whether each grasping attempt is successful by looking its tactile sensors. Finally, the offsets that leads to the successful grasping together with the object features are stored for training. On the other hand, in order to reduce the robot running, we emulate this setup by inputing these offsets manually.

The second stage(*Stage-2*) is a parental-scaffolding based learning in which we remove the assumption of the object-of-interest being at the fixed positions. In this stage, the robot moves its hand to the object-of-interest by learned offsets in the first stage. In such a scenario, the robot makes errors while approaching since the objects are not in the fixed positions. After

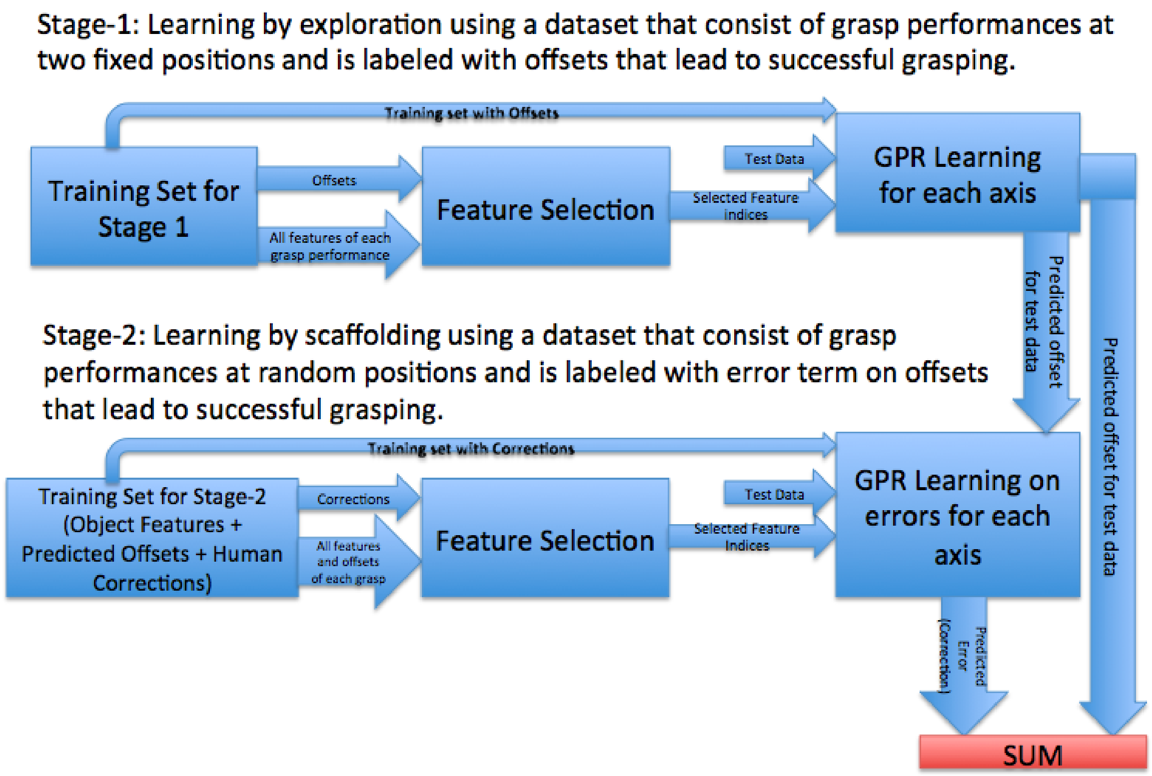


Figure 5.4: Overall learning scheme where summation of two GPRs gives how to approach the object for the corresponding axis.

that, the caregiver corrects the robot's end-effectors position such that a stable grasp can be achieved. Overall scheme can be seen at Figure 5.4

5.4.3 Stage 1: Learning by Exploration

Stage-1 is based on affordance learning and mimics the self-learning of an infant. In this stage, we use slightly modified version of Şahin et al.'s affordance formalization [5]:

$$(\textit{effect, entity, } (x, y, z))$$

representing that when the agent applies the *grasp* with offsets *(x, y, z)* on a certain *entity*, the given *effect* is generated. The term *entity* stands for an object or any environmental entity that the robot interacts with it. The *effect* represents the result of the behavior. In this work, we used a specialized version of this relation where *effect* can be *grasped* or not grasped.

In this stage, we mainly aim to learn how grasping behavior should change with respect to the object's size and shape-related features. Thus, objects can only be located at two pre-determined locations. A dataset, which contains successful grasping attempts at these locations, is collected for training. For each attempt, the feature set of the corresponding object labeled with the offsets in three dimensions. After that, the relevant features that affects the offsets were determined. In this case, 7 features were selected since their weights are in the top-20 highest weights for each offset. Finally, using these features, a GPR is trained for offsets in each axis.

5.4.4 Stage 2: Learning by Scaffolding

Stage-2 mimics the scaffolding theory of the developmental psychology by taking into account of human's guidance. The main aim in this stage is to learn the relationship between the position and orientation of the object and grasping behavior. Thus, we remove the assumption of being at the pre-determined behavior in this stage.

The training is similar to the training in *Stage-1*. A dataset, which contains successful grasping attempts at random locations, is collected for learning. In this dataset, there are the successful grasp performances, corrected by human guidance at random positions. For each attempt, the feature set of the corresponding object and predicted offsets (using the trained GPRs in *Stage-1*) stored with the offset corrections in three dimensions. These corrections were determined by the caregiver's intervention to the end-effector's position. After that, the relevant features that affects the offsets were determined. Similar to *Stage-1*, 7 features were selected since their weights are in the top-20 highest weights for each offset for comparing the learning rate with the control learning methodology. In addition to these 7 relevant features, the offset that the corresponding Stage-1 GPR returns for that axis is added into each relevant feature set. For analyzing how the learning rate change with the different number of relevant features, the relevancy threshold is parametrized. By changing the value of this parameter, we asses the learning rate and try to find the optimum threshold. Finally, using these features, a GPR is trained for the errors on each offset.

Overall, we have the following affordance formalization after *Stage-2*:

$$\textit{(effect, entity, } (x, y, z) + (x_c, y_c, z_c)\textit{)}$$

where x_c, y_c and z_c represents the corrections made by human caregiver in the corresponding axis.

CHAPTER 6

EXPERIMENTAL FRAMEWORK AND METHODOLOGY

6.1 Robotic Platform

A 53 DoF humanoid robot platform, *iCub* [92] is used as the main robotic platform in our studies. *iCub* is an experimental framework for cognitive studies which was the result of European Commission FP6 project *RobotCub*. The arm of *iCub* has 7 DoF. Each hand has 108 tactile sensors. The reference frame of *iCub* can be seen at (Figure 6.2). On each fingertip, there are 12 tactile sensors and rest of them are located in the palm. In addition, iCub is also equipped with four 6-axis force-torque sensors (FT sensors) mounted on its arms.

Being the product of an academic study brings two main advantages to its users. Firstly, both its software design and hardware design are publicly-accessible. This makes possible to modify the robot according to your needs. Secondly, there is a strong academic community that deals with the robot simultaneously. You can communicate with them while encountering a problem and use their software modules and solutions to accelerate your research.

6.2 Perception

We use Microsoft's Kinect sensor for perception (Figure 6.3). The device, originally developed for the company's XBOX 360 video game console, mainly contains a range sensor and a RGB camera. The range sensor consists of an infrared laser projector combined with a monochrome CMOS sensor. Overall, the sensor has 640×480 resolution and 30 Hz frame rate.

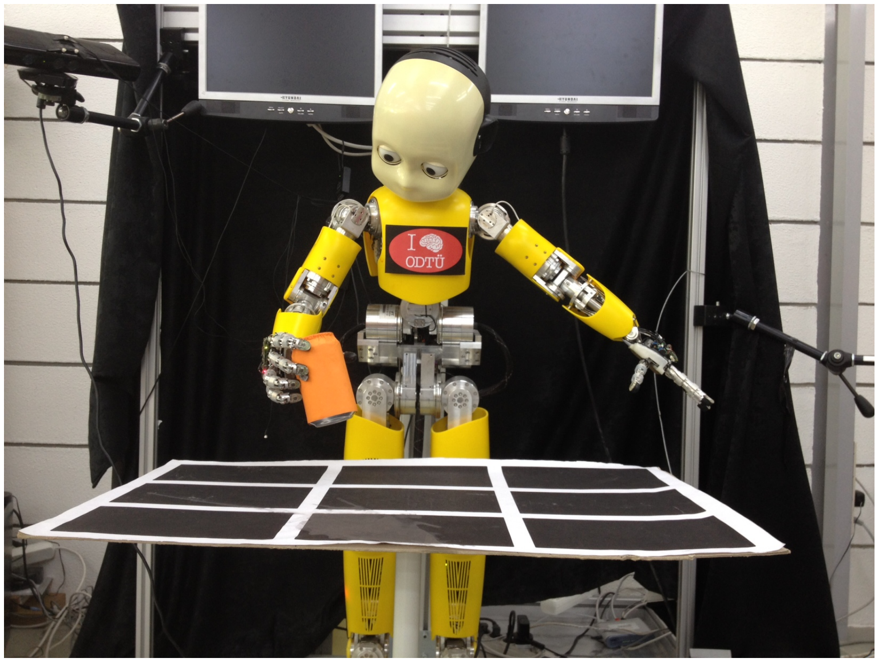


Figure 6.1: *iCub* Humanoid Platform.

6.3 Conversion between Reference Frameworks

In order to transform the camera's reference frame to the robot's, the 3D motion capture system, *VisualEyez VZ4000* is used (Figure 6.4). VZ4000 uses active markers to find the pose of the objects. The device is able to capture motion within a volume of $190m^3$. We place active markers at the center of both *iCub* and Kinect and obtain a $4x4$ conversion matrix by the device's API.

Although same transformation can be done using regression on data which contains information of where a certain point is located in the robot's reference frame (forward-kinematics) and in Kinect's reference frame, using VZ4000 provides a quicker and less error-prune method for this calibration.

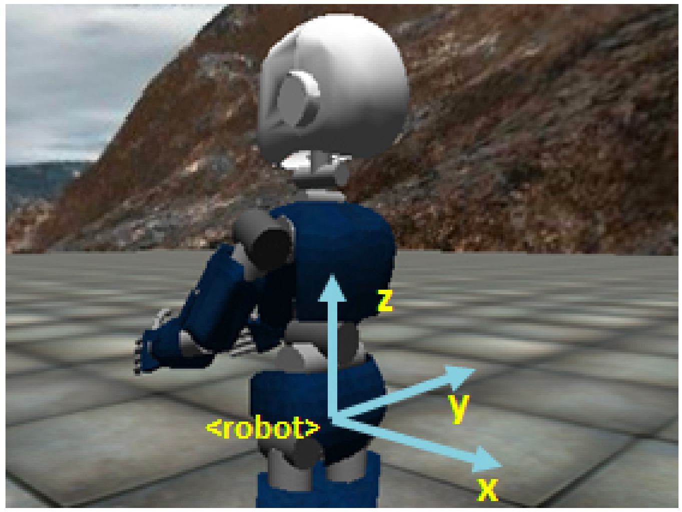


Figure 6.2: The reference frame of *iCub*, the image is taken from [93]

Figure 6.3: Microsoft Kinect is used to get the point cloud of the environment during experiments.

Figure 6.4: VisualEyez VZ4000 is used to transform point clould obtained from Kinect to *iCub*'s reference framework.

6.4 Software Platforms Used

6.4.1 Yet Another Robot Platform (YARP)

YARP [94] is the middleware of *iCub*'s software system. All of the drivers and low level control modules of *iCub* were implemented using YARP. By the design of its *port* facility, YARP makes easy to communicate between nodes of a multi-threaded (and/or multi-computer) robotic application. Via *ports*, nodes pass informative messages each other.

In this work, we use YARP platform for communicating with the robot-related hardware/software components.

6.4.2 Robot Operating System (ROS)

ROS [95] is another alternative middleware for robotic platforms and contains many state-of-the-art robotic algorithms as built-in libraries. Our software mainly uses ROS-module structure and information passing between software components are done via ROS communication tools such as topics, service-calling and *actionlib* structure. Only communication via *iCub* nodes are done using YARP.

One of ROS' useful libraries is Point Clould Library (PCL), which is an open-source library for 3D point cloud processing. It contains many important segmentation and feature extraction algorithms used this thesis such as table-top segmentation, surface normal estimation and principle curvature analysis.

6.5 Software Components

The software system that is developed for this study consists of 7 main modules (Figure 6.5). *Experiment Manager* is responsible for coordinating different robot and perception modules during the experiments. It has three modes. In *Stage-1 Training Mode*, the experimental is prompted to input three offsets which describe how the robot should approach the object-of interest. In *Stage-2 Training Mode*, the robot tries to approach the object via learned offsets in Stage-1. After approaching complete, the module fixed the hand at given position prior to grasping. Then, the human modifies the hand's position so as the robot makes successful grasping. Finally, in *Testing Mode*, the robot approaches the object with learned offsets and error corrections in two-stage learning, then it starts grasping behavior automatically.

The second module is *Grasping Manager*. It gets directions from *Experiment Manager* by ROS Service calls. For each direction, it communicates with robot modules using YARP Ports. The third module is *Perception Manager*. It connects and segments Kinect's online perceptual data via Point Cloud Library (PCL) API. It also connects ROS' *tf* module, which contains the transition matrix between reference frames. According to this information, it is responsible for converting the segmented point cloud to the robot's framework.

6.6 Methodology

6.6.1 The Grasping Behavior

The behavior set used in experiments consists of *reach*, *grasp* and *lift*. Firstly, the *reach* behavior just consists of moving the palm to the given cartesian coordinate. Secondly, *grasp* moves each of the fingers towards the palm until the respective tactile sensors of each finger

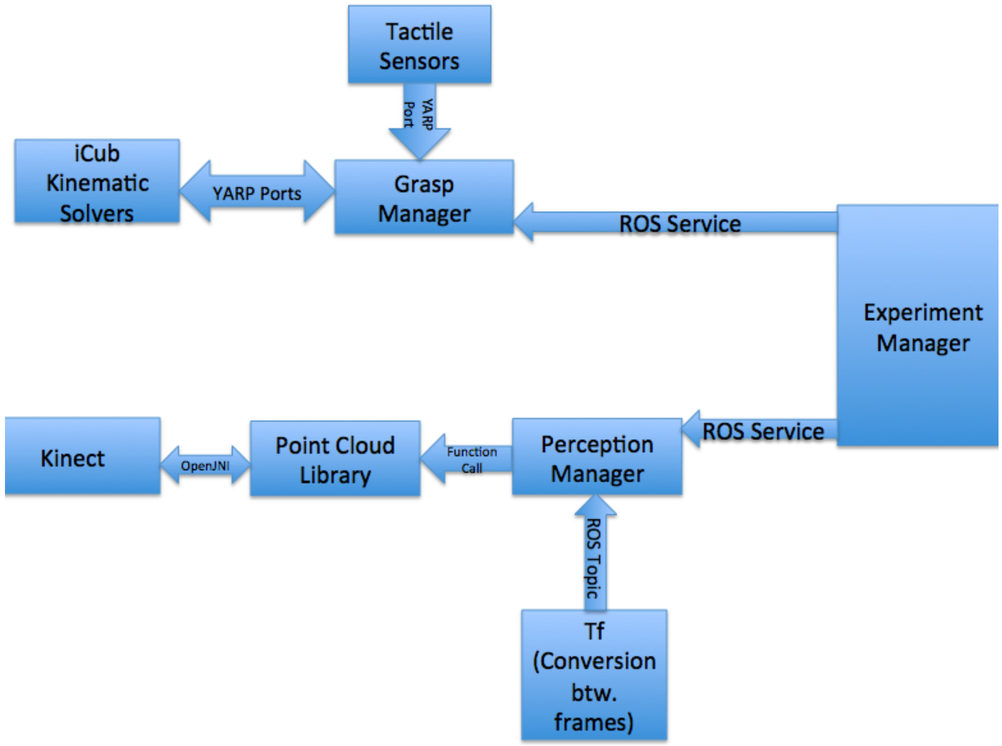


Figure 6.5: The component diagram of the software

reaches the threshold value. Finally, *lifting* moves the palm to 10 cm upwards. This behavior is used to check whether grasping is stable.

In the proposed framework, *iCub* (Figure 6.1) applies a chain of *reach-grasp-lift* actions for two different grasping styles, namely top-grasp and side-grasp. The illustrations of these grasp styles can be seen in Figure 6.6. The object of interest is perceived by 3D perception system and enclosed by a bounding box.

The robot reaches the corresponding surface (right surface for side-grasp and top surface for top-grasp) of the bounding box of the object following a trajectory that is computed by a minimum-jerk controller, which moves the arm to the desired pose while keeping acceleration changes minimum. Then, the hand gets closed to the object according to given/human-modified/learned offsets. After approaching is completed, the robot closes its fingers with a tactile sensor-based grasp controller. The grasp controller executes fingers such that the movement of each finger stops if sum of its tactile reading reaches a threshold or the corresponding finger's position becomes unsafe. Finally, the robot tries to lift the object to assess if the grasping is stable

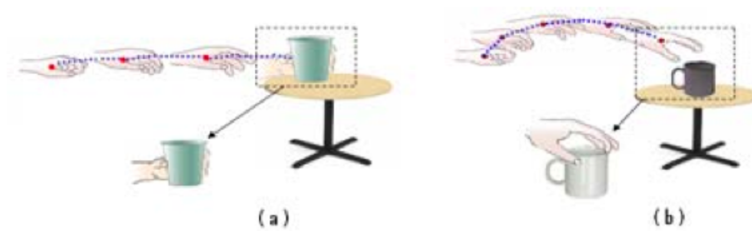


Figure 6.6: a) Side-grasp b) Top-grasp (Figure is taken from [96]).

6.6.2 Training

At *Stage-1* , the robot tries to grasp the object in a certain grasping style (either top-grasp or side-grasp) with the random offsets. These offsets can be given by the experimenter or created by a pseudo-random algorithm. The experiment with each object is repeated many times until grasping is successful and the robot can lift the object. In our experiments, the operator gives offsets to the system for a successful grasp but this is for achieving the successful grasp performance fast. Prior to grasping, the object's features, along with given offsets, are extracted and stored by *Perception Manager*. In this stage, the positions of objects are pre-determined (Figure 7.6a and 7.6b) positions. After experimenting, each successful performance together with offsets is stored in a file.

Using this file, firstly, we use a feature selection algorithm to obtain relevant features for each offset. The algorithm gives a relevancy weight to each feature. By giving a threshold weight, we filter out irrelevant features. In other words, we have 3 set of relevant features and each set corresponds to an offset. After that, for each offset, we train the learning algorithm with its corresponding relevant features only (Overall, we have 3 trained algorithm).

At *Stage-2*, we remove the assumption of pre-determined locations. The robot tries to grasp the objects which are outside of these regions with the trained algorithm instances in *Stage-1*. Obviously, in such a scenario, it makes errors while grasping. This errors are fixed by the human partner which modifies end-effector's positions prior to grasping.

Similar to *Stage-1*, initially, the object is featurized by *Perception Manager*. Thereafter, using this features, the trained algorithms of stage-1 are queried by the system for obtaining the predicted offsets. After this, the robot moves its arms towards the object with these offsets. Prior to grasping, the human modifies the end-effector's position, with the help of FT-sensors, to lead a successful grasp. Then, the robot grasps and lifts the object.

After experimenting, each successful performance together with the predicted offsets and the corrections for offsets is stored in a file. The predicted offsets are given to the learning mechanism as features. After experimenting is completed, the learning is same as the stage-1. For each offset, firstly, the relevant features are obtained through the feature selection algorithm. Then, the learning algorithm is trained for each offset's correction amount using the relevant features and the predicted offsets.

6.6.3 Testing

In order to grasp different objects, the hand should approach each object with the learned offsets (Figure 5.3). For deriving these learned offsets, *Experiment Manager* should query first the Stage-1 trained learning algorithm, then, the Stage-2 trained learning algorithm. For each offset, the sum of Stage-1's predicted offset and Stage-2's correction amount will give the learned offset for the current object.

In the online-testing mode, features of the object-of-interest are extracted. Then, the learning algorithms are queried to get the predicted offsets using relevant features. Finally, the robot applies *reach-grasp-lift* behavior with these offsets.

For analysis the performance of our learning system, we train and use a second learning methodology. This system is based a simple scaffolding methodology that is similar to the stage-2 learning of the proposed learning system. The overall experimental setup is presented in detail at *Chapter 6*.

An overview of the grasping framework is illustrated in Figure 6.7.

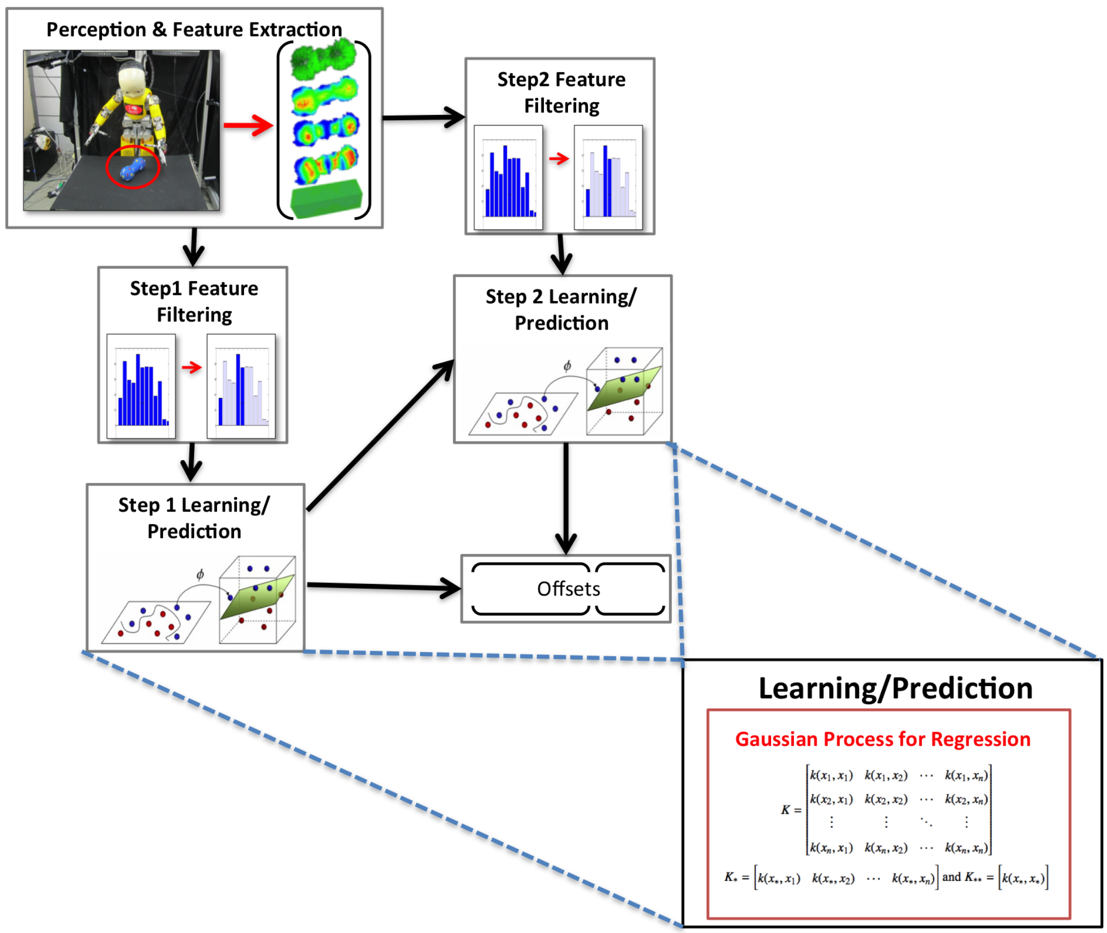


Figure 6.7: Overview of the system. iCub perceives the environment and learns offsets in a two-stage learning. Figure is adapted from [12].

CHAPTER 7

EXPERIMENTAL RESULTS

In this section, firstly, the experimental setup is presented. Then, the experimental results of the proposed learning methodology is presented for two grasping styles, namely side-grasp and top- grasp. As a control group, another learning mechanism, which contains just the second stage of proposed learning, is used for comparison. Thereafter, we analyze how the number of relevant features effect the learning rate and find out what the optimal threshold for the relevancy weight is. After that, we analyze how the correction of Stage-2 changes over different locations. Finally, we conclude with an example success scenario

7.1 Experimental Setup

The robot uses a behavior set of *reach*, *grasp* and *lift* to move its hand to the center of the bounding box's left surface or its top surface (depending on the grasp style). Thereafter, it encloses fingers to grasp the object. Finally, it moves its palm upwards to check whether grasping is stable. In *Stage-1*, the destination of the right hand modified with offsets in x, y and in z axes in the robot's reference framework. In the second stage of the learning, the end-effector's position is modified by the human partner prior to enclosing fingers.

10 objects were chosen for training and 5 novel objects were chosen for testing. All objects are placed at two different pre-determined positions and tested with which offsets they can be successfully grasped (Figure 7.6). For the 3-fold cross validation, training data, which contains object features and given offsets for each successful grasping attempt, is split into three and one slice is used for testing. The difference between predictions and given offsets denotes the error. For the results on novel objects, training data was used as whole to train the

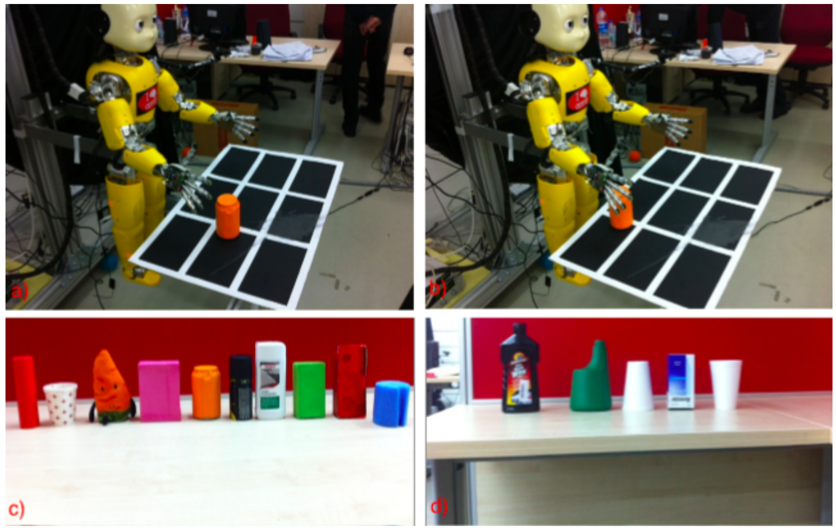


Figure 7.1: a,b) Pre-determined object positions for *Stage-1* experiments c) Objects used for training d) Novel objects used for testing

GPRs. Then, testing data is used to evaluate performance.

After *Stage-1*, we have a trained GPR for each axis. Then, the same 10 objects were used for training and the other 5 objects were used for testing. All objects were put into 4 different random positions in *Stage-2*. For each position, the robot tries to grasp object with its prior knowledge. Then, human modifies the arm position and makes the robot to grasp the object. After that, the data is stored with errors (which we call as *correction*) that the robot made in three axes. Similar to *Stage-2*, training data, which contains object features, predicted offsets by Stage-1 GPRs and correction rates, is split into three and one slice is used for testing for the 3-fold cross validation. The difference between predictions and labeled correction rates denotes the error. For the results on novel objects, training data was used as whole to train the GPRs. Then, testing data was used to examine whether the robot can grasp the object.

7.2 Side-Grasp Results

7.2.1 Relevant Features

For *Stage-1* and *Stage-2* learning, the features whose weights are in the top-20 for all axes are given in Table 6.1A and Table 6.1B respectively. Also, for Stage-2, the relevancy weight of Stage-1 offsets are included in Table 6.1B.

As illustrated in the tables, the relevant features of Stage-1 are mostly features that describe the shape properties of the object. On the other hand, since two pre-determined locations are located in different areas at the $x - axis$, the position of the object at that axis is also found relevant.

For the Stage-2 learning, mostly size and position features are found as relevant. This is also expected since we do not have any position constraints in Stage-2. Thus, it is normal that the key learning factors on this stage are positions and sizes at different axes.

The offset that is returned by Stage-1 learning algorithms has a higher relevancy weight in $x - axis$ and $z - axis$. On the other hand, this offset in $y - axis$ is not as important as in other axes.

(a)

Feature	Weight in X	Weight in Y	Weight in Z
$pos - x$	0.142	0.15235	0.0724
$azimuth - histogram - 2$	0.1108	0.28174	0.072032
$minimum - curvature - 3$	0.097885	0.06905	0.203645
$max - curvature - 7$	0.116914	0.15235	0.189726
$size - z$	0.0118924	0.0173586	0.05498
$shapeIndexHist - 13$	0.021176	0.08417	0.101555
$zenith - histogram - 5$	0.01691	0.05498	0.0835

(b)

Feature	Weight in X	Weight in Y	Weight in Z
$pos - y$	0.027663	0.014653	0.026193
$size - x$	0.1806534	0.010793	0.004329
$size - z$	0.0363765	0.01945	0.020156
$pos - x$	0.0452528	0.019771	0.028391
$max - curvature - min$	0.0575996	0.038201	0.027957
$max - curvature - max$	0.05441	0.030992	0.036341
$zenith - hist - 0$	0.0919471	0.05406	0.050457
$Stage1 - offset - x$	0.0781851	-	-
$Stage1 - offset - y$	-	0.001955	-
$Stage1 - offset - z$	-	-	0.058499

Table 7.1: Relevant Features and their weights for (a) Stage-1 and (b) Stage-2 for Side-Grasp

7.2.2 Stage 1 Evaluation

3-Fold cross validation is used to see the system's performance in similar objects. In Table 6.2, the mean and standard deviations are given.

Table 7.2: The mean and the standard deviation of the relative errors of predicting offsets in terms of centimeters (3-Fold Cross Validation, 500 Repeats) for Stage-1 only learning of side-grasping.

	μ (cm)	σ (cm)
X	0.3754	0.6902
Y	0.2582	0.5168
Z	0.298	0.1287

Table 7.3: The mean and the standard deviation of the relative errors of predicting offsets in terms of centimeters (3-Fold Cross Validation, 500 Repeats) for overall learning of side-grasping.

	μ (cm)	σ (cm)
	The proposed system	
X	0.7462	0.8018
Y	0.6594	0.6871
Z	0.7011	0.4287
	The control group	
X	1.4172	1.0759
Y	1.2161	1.1098
Z	0.9863	1.1203

For testing with novel objects, we use the training set for learning as whole (20 successful grasp attempts on the objects in Figure 7.6A). Then, we use the testing set (10 grasp attempts on the objects in Figure 7.6D) to analyze the learning performance.

We have 90% success rate on autonomous grasp tests (which consist of 10 grasping attempts for 5 novel objects in Figure 5D) on novel objects. Only unsuccessful attempt was occurred when the green bottle was at position 2 (Figure 5B). In this case, slipping occurred while lifting due to a slight misplacement of the thumb.

7.2.3 Stage 1 & Stage 2 Evaluation

Similar to *Stage-1*, 3-fold cross validation is applied both for the presented system and the control group. In order to see the benefit of the two-stage incremental learning, the control group's learning scheme consists of a single GPR-based scaffolding learning with the same amount of training data.

For testing with novel objects, we use the testing set (40 grasp attempts on the objects at the 4 different random positions in Figure 7.6D) to analyze the learning performance. We have 87.5% success rate on autonomous grasp tests (which consist of 10 grasping attempts for 5 novel objects in Figure 5D) on novel objects.

7.3 Top-Grasp Results

7.3.1 Relevant Features

For the Stage-1 and Stage-2 learning, the features whose weights are in the top-20 for all axes are given in Table 6.4A and Table 6.4B respectively. Also, for Stage-2, the relevancy weight of Stage-1 offsets are included in Table 6.4B.

Half of the relevant features of Stage-1 are the shape-related features. On the other hand, since two pre-determined locations are located in different areas in the $x - axis$, the position of the object at that axis is also found relevant. Moreover, since the arm is approaching to the top surface of the object while doing top-grasping, size and position in the $z - axis$ are also important. On the other hand, since the table lies on the $x - y$ plane, size and position in the $z - axis$ describe the same object property. Thus, one of these is unnecessary.

Similar to *side-grasp*, for the Stage-2 learning, mostly size and position features are found as relevant. This is also expected since we do not have any position constraints in Stage-2.

The offset that is returned by Stage-1 learning algorithms has a higher relevancy weight in the $y - axis$ on the contrary to *side-grasp*.

7.3.2 Stage 1 Evaluation

3-Fold cross validation is used to see the system's performance in similar objects. As described above, the training set was split into three in each repeat. In Table 6.5, the mean and standard deviations are given.

Knowing the fact that iCub's inverse kinematic modules have much less resolution than $1cm$, relative errors are relatively small for both grasping styles.

(a)

Feature	Weight in X	Weight in Y	Weight in Z
$azimuth - histogram - 2$	0.042261	0.043715	0.081257
$posX$	0.074493	0.1487792	0.092685
$sizeZ$	0.041469	0.0371963	0.032757
$zenith - max$	0.021964	0.0227086	0.015426
$shapeIndexHist - 13$	0.026946	0.0101919	0.01148
$posZ$	0.031739	0.0464673	0.144823

(b)

Feature	Weight in X	Weight in Y	Weight in Z
$sizeY$	0.0100158	0.0485481	0.027101
$sizeX$	0.071398	0.0222245	0.034275
$MaxCurvatureHist - 18$	0.034256	0.0263685	0.009715
$posX$	0.026482	0.0088496	0.004509
$posY$	0.009244	0.0134686	0.00556
$MinCurvatureHist - 3$	0.054685	0.0239432	0.058273
$MaxCurvatureHist - 17$	0.042723	0.0172533	0.01709
$Stage - 1 - offset - x$	0.013118	-	-
$Stage - 1 - offset - y$	-	0.0485481	-
$Stage - 1 - offset - z$	-	-	0.014581

Table 7.4: Relevant Features and their weights for (a) Stage-1 and (b) Stage-2 for Top-Grasp

Table 7.5: The mean and the standard deviation of the relative errors of predicting offsets in terms of centimeters (3-Fold Cross Validation, 500 Repeats) for Stage-1 only learning of top-grasping.

	μ (cm)	σ (cm)
X	0.2431	0.4453
Y	0.6140	0.8185
Z	0.3606	0.4066

For testing with novel objects, we use the training set for learning as whole (20 successful grasp attempts on the objects in Figure 7.6A). Then, we use the testing set (10 grasp attempts on the objects in Figure 7.6D) to analyze the learning performance. We have 100% success rate on autonomous grasp tests (which consist of 10 grasping attempts for 5 novel objects in Figure 5D) on novel objects.

Table 7.6: The mean and the standard deviation of the relative errors of predicting offsets in terms of centimeters (3-Fold Cross Validation, 500 Repeats) for overall learning of top-grasping.

	μ (cm)	σ (cm)
	The proposed system	
X	0.6713	0.8885
Y	0.8317	0.6759
Z	0.6667	0.5098
	The control group	
X	1.3926	0.9759
Y	1.2356	1.1238
Z	1.0210	1.2087

7.3.3 Stage 1 & Stage 2 Evaluation

Similar to Stage 1 testing, 3-fold cross validation is applied both for the presented system and the control group. In order to see the benefit of the two-stage incremental learning, the control group's learning scheme consists of a single GPR-based scaffolding learning with the same amount of training data.

As you can see from the Table 6.2 and Table 6-4, the two-stage incremental learning approach outcomes the control group for both grasping performances. The main reason behind that is having the information of offset prediction from the Stage-1. This information is identified as highly relevant by the ReliefF algorithm and is used in the training of the Stage-2 GPRs.

For testing with novel objects, we use the same testing set as the *Section 6.2.5*. Similarly, we have 87.5% success rate on autonomous grasp tests (which consist of 10 grasping attempts for 5 novel objects in Figure 5D) on novel objects.

7.4 Number of Relevant Features versus Learning Rate

In this section, we evaluate how the number of relevant features affects the learning performance of *Stage-2* using 3-fold cross validation.

In Figure 7.2 and Figure 7.3, we illustrate how errors are changing with respect to the relevancy threshold and how many features are included to the learning with the respective threshold.

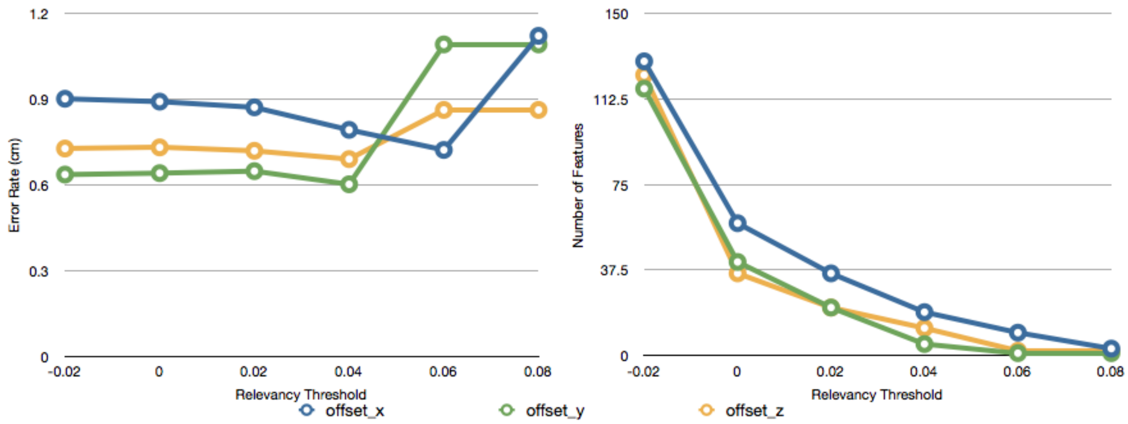


Figure 7.2: The analysis of relevant features for side-grasp. The first graph shows the relation between error rate and relevancy threshold. The second one shows the relation between number of relevant features and relevancy threshold.

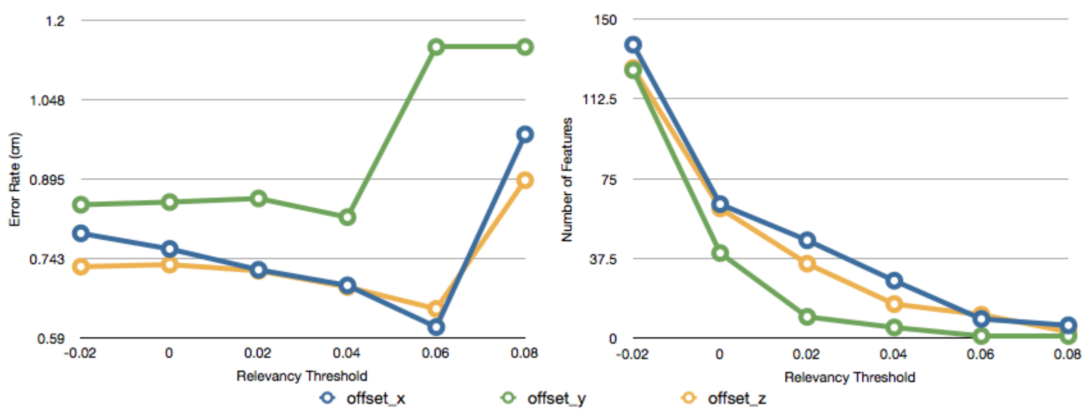
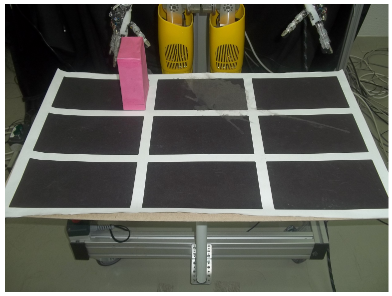
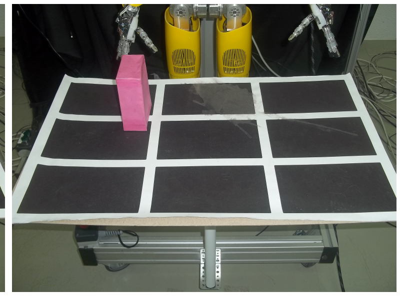


Figure 7.3: The analysis of relevant features for top-grasp. The first graph shows the relation between error rate and relevancy threshold. The second one shows the relation between number of relevant features and relevancy threshold.

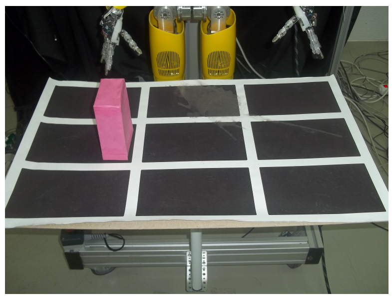
As seen from the figures, each offset exhibits a similar characteristics on the relevancy threshold variation for both grasping styles. Although a better learning can be achieved by choosing the relevancy threshold as 0.06 for *offset_x* and *offset_z*, the overall best learning rate is achieved by fixing threshold to 0.04. This rate nearly same with the learning rate in the previous performances in which we choose 7 features, whose weights are in the top-20 highest weights for each offset, as the relevant features.



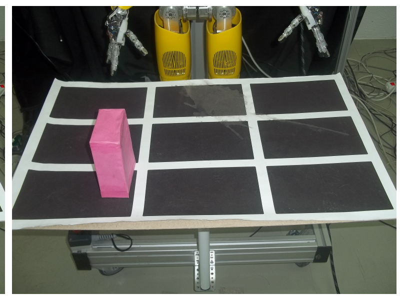
(a) The first position



(b) The second position

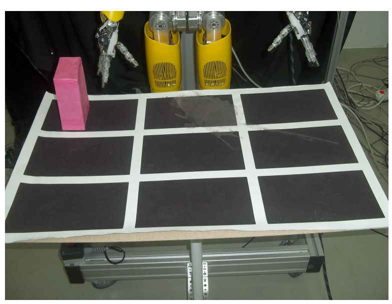


(c) The third position

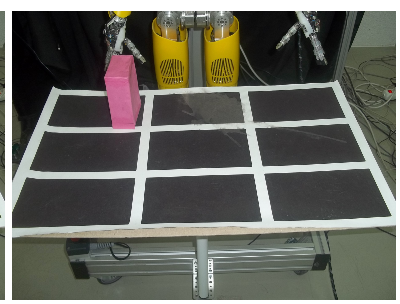


(d) The final position

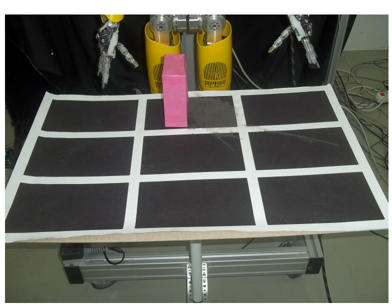
Figure 7.4: The positions for assessing Stage-2 Correction in *x-axis*.



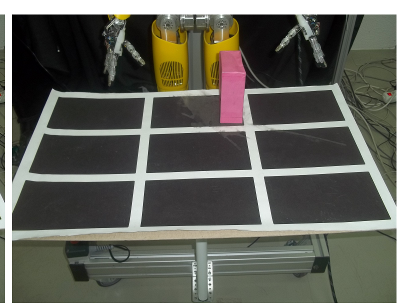
(a) The first position



(b) The second position



(c) The third position



(d) The final position

Figure 7.5: The positions for assessing Stage-2 Correction in *y-axis*.

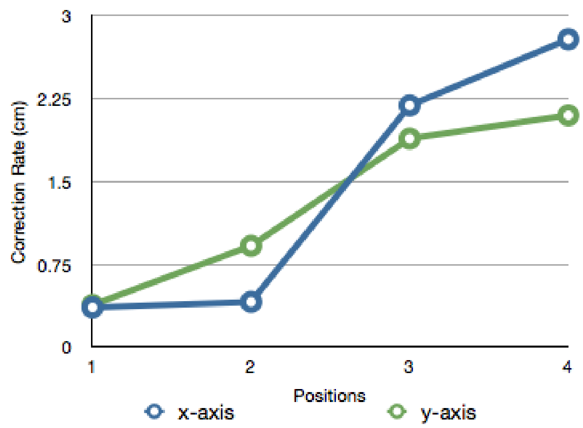


Figure 7.6: The variation of correction rate in Stage-2. For *x-axis*, the positions are shown in Figure 7.4. For *y-axis*, the positions are shown in Figure 7.5.

7.5 Analysis of the Importance of Stage-2 Correction

We assess the impact of Stage-2 learning by evaluating how a particular axis' Stage-2 correction rate changes when the object is moved away from a pre-determined location of Stage-1 in the direction of the corresponding axis.

For *x-axis*, we first locate the object at the position shown in Figure 7.6B then move towards $x = -\infty$. All positions of the object are shown in Figure 7.4. The variation of correction rate in *x-axis* is illustrated in Figure 7.6A. As seen in this figure, the correction rate is relatively small at the starting point and the point between two pre-determined locations. After these two points, the correction rate becomes higher while the object is becoming distant from the pre-determined locations.

Similarly, in the case of *y-axis*, we first locate the object at the position shown in Figure 7.6B then move towards $y = -\infty$. All positions of the object are shown in Figure 7.5. The variation of correction rate in *y-axis* is illustrated in Figure 7.6. Similar to the $x - axis$ case, the correction rate is relatively small when the object is at the starting point. On the other hand, after this point, the object starts to be distant from the pre-determined locations. Thus, the correction increases.

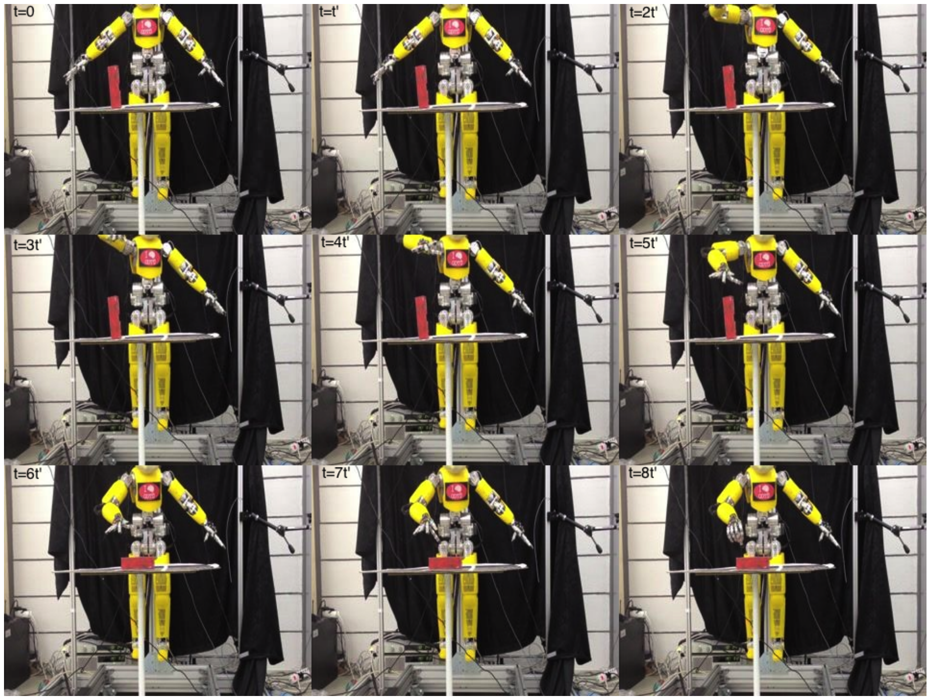


Figure 7.7: The hand-coded grasping behavior leads a failure of grasping.

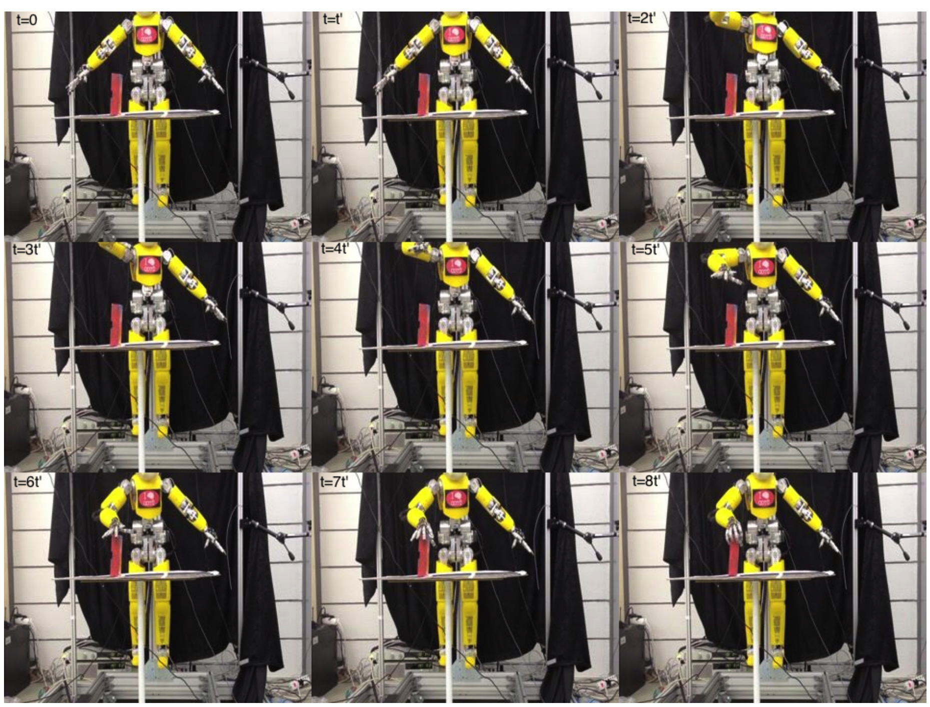


Figure 7.8: The successful grasping performance with the proposed learning system.

7.6 An Example Success Scenario

We conclude this chapter with a scenario in which the robot with the proposed learning system successfully grasp the object (Figure 7.7) while the hand-coded grasping behavior does not lead to a successful grasp behavior (Figure 7.8). This example emphasizes the importance of approaching the object with the correct offsets leads the successful grasping.

CHAPTER 8

DISCUSSION AND CONCLUSION

In this thesis, a developmental scheme for humanoid robot's grasp learning is proposed. Our scheme takes inspiration from two psychological notion, *affordance notion* of ecological psychology and *scaffolding theory* of developmental psychology.

Stage-1 mimics humans' self-exploration and self-learning. This stage was implemented on the fundamentals of Gibsonian ecological psychology. We used a formalization of affordances[5] to make the robot develop its adaptive grasping behavior on different objects that can only be located at two predetermined locations. The robot tries to approach the object with random objects. Then, using the data of the successful, the robot tries to generalize how to approach given objects in order to grasp them successfully. In our case, this corresponds to training GPRs.

In this stage, there is an indirect and optional supervision. Although we are using a supervised learning in this stage, understanding whether grasp performances are successful is easy due to tactile sensors. Thus, the robot can detect the result of each grasping attempt automatically by looking its tactile sensors. Furthermore, the offsets of grasp performances are not necessarily given by the humans in this stage. The robot, by trial-and-error, can also find offsets that leads a successful grasp performance. In the second stage, there is a direct supervision in which the human partner modifies the final position of the arm so as to lead the robot to a successful grasping.

Since we remove the assumption of objects being at two predetermined locations, the error the robot makes becomes higher in *Stage-2*. Therefore, a human partner helps the robot to correct its mistakes while grasping at random positions. The partner modifies the end-effector's position to correct the errors. After enough samples, new GPRs are trained in order to make the robot generalize these error terms.

We tested the proposed scheme on two different grasp types; namely top-grasp and side-grasp. After completing two-stage learning, the results of automated grasp performances on novel objects have a success rate of 87.5% for both types.

8.1 Future Work

The main disadvantage of the current system is that it does not take into account of how articulation affects grasping behavior. For example, The usual way of grasping a mug is by its handle. Thus, an improvement can be achieved by supplying articulation information to the system and training the algorithm such that it can adjust the offsets according to this information.

Additionally, *grasp* behavior is hand-coded in the proposed system. We learn how to grasp by modifying *reach* behavior prior to *grasp* with learned offsets. Some performance improvements can be gained by making *grasp* behavior adaptive to different objects.

Moreover, the caregiver currently corrects only the final position of the arm prior to grasping. An immediate improvement would be, instead of just the final position of the arm, correcting the robot's whole reaching part. In the current scheme, the robot reaches the object with a simple *minimum-jerk controller*. With this improvement, the robot would have the capability of how to reach the different objects with different properties. Obviously, in the end, the robot will learn more natural grasping behavior with the learned reaching behavior.

Another possible improvement would be the inclusion of tactile-data as features for the learning algorithm. By having such features, the robot would learn how its tactile sensors return after grasping. Thus, it can locate its arm with this information. For example, some small objects are grasped with only two fingers. By learning such a fact, the robot will locate its arm such that only two fingers have the contact with the object.

REFERENCES

[1] "http://eris.liralab.it/wiki/simulator_readme retrieved on sep 9, 2012,"

[2] J. Bohg and D. Kragic, "Grasping familiar objects using shape context," in *Int. Conf. on Advanced Robotics*, 2009.

[3] O. Yürüten, K. F. Uyanik, Y. Çaliskan, A. K. Bozcuoglu, E. Sahin, and S. Kalkan, "Learning adjectives and nouns from affordances on the icub humanoid robot," in *Proceedings of the 2012 International Conference on Adaptive Behaviour*, SAB '12.

[4] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, pp. 151–190, Dec. 2003.

[5] R. K. Clifton, D. W. Muir, D. H. Ashmead, and M. G. Clarkson, "Is visually guided reaching in early infancy a myth?," *Child Development*, vol. 64, pp. 1099–1110, 1993.

[6] J. Gibson, "The theory of affordances," *Perceiving, Acting, and Knowing*, 1977.

[7] L. Vygotski and M. Cole, *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1978.

[8] E. Şahin, M. Çakmak, M. R. Doğar, E. Uğur, and G. Üçoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, vol. 15, pp. 447–472, Dec. 2007.

[9] M. Cakmak, M. R. Dogar, E. Ugur, and E. Sahin, "Affordances as a framework for robot control," in *Proc. of International Conference on Epigenetic Robotics, EpiRob 2007*, Nov. 2007.

[10] M. R. Dogar, M. Cakmak, E. Ugur, and E. Sahin, "From primitive behaviors to goal-directed behavior using affordances," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2007*, pp. 729–734, 2007.

[11] E. Ugur, M. R. Dogar, M. Cakmak, and E. Sahin, "Curiosity-driven learning of traversability affordance on a mobile robot," in *IEEE 6th International Conference on Development and Learning (ICDL' 07)*, pp. 13–18, July 2007.

[12] E. Ugur, M. R. Dogar, M. Cakmak, and E. Sahin, "The learning and use of traversability affordance using range images on a mobile robots," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1721–1726, Apr. 2007.

[13] N. Dag, I. Atil, S. Kalkan, and E. Sahin, "Learning affordances for categorizing objects and their properties," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, (Washington, DC, USA), pp. 3089–3092, IEEE Computer Society, 2010.

[14] I. Atil, N. Dag, S. Kalkan, and E. Sahin, "Affordances and emergence of concepts," in *Proceedings of the 2010 International Conference on Epigenetic Robotics*, EPIROB '10, pp. 11–18, 2010.

[15] K. Uyanik, Y. Caliskan, A. Bozcuoglu, O. Yuruten, S. Kalkan, and E. Sahin, "Learning social affordances and using them for planning," in *Submitted to the 2012 International Conference on Humanoid Robots*, Humanoids '12.

[16] K. Uyanik, O. Yuruten, S. Kalkan, and E. Sahin, "Towards context-aware adjective learning," in *ICRA 2012 Workshop: The Future of HRI - Paving the way to next generation of HRI*, ICRA '12.

[17] O. Yuruten, "Incremental formation of verb and object concepts through affordances," *M.Sc., Dept. of Computer Engineering, Middle East Technical University*, 2012.

[18] A. Bozcuoglu and E. Sahin, "Traversability on a simple humanoid: What did i just trip over?," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pp. 701 –706, oct. 2011.

[19] J. Gibson, *The Senses Considered as Perceptual Systems*. Houghton Mifflin Company, 1966.

[20] J. Gibson, *The ecological approach to visual perception*. Lawrence Erlbaum, 1979.

[21] R. G. Barker, *Ecological Psychology: Concepts and Methods of Studying the Environment of Human Behavior*. Stanford: University Press, 1968.

[22] R. Barker and H. Wright, *One Boy's Day*. Archon Books, 1966.

[23] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, G. Sandini, and S. Giulio, "Learning about objects through action - initial steps towards artificial cognition," in *In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA*, pp. 3140–3145, 2003.

[24] P. Fitzpatrick and G. Metta, "Grounding vision through experimental manipulation," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, pp. 2165–2185, 2003.

[25] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *Robotics, IEEE Transactions on*, vol. 24, pp. 15 –26, feb. 2008.

[26] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3071–3076, 2005.

[27] J. Sinapov and A. Stoytchev, "Learning and generalization of behavior-grounded tool affordances," in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pp. 19 –24, july 2007.

[28] J. Sinapov and A. Stoytchev, "Detecting the functional similarities between tools using a hierarchical representation of outcomes," in *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, pp. 91 –96, aug. 2008.

[29] K. Huebner, M. Björkman, B. Rasolzadeh, M. Schmidt, and D. Kragic, “Integration of visual and shape attributes for object action complexes,” in *Proceedings of the 6th international conference on Computer vision systems*, ICVS’08, (Berlin, Heidelberg), pp. 13–22, Springer-Verlag, 2008.

[30] D. Kraft, N. Pugeault, E. Baseski, M. Popovic, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krüger, “Birth of the object: Detection of objectness and extraction of object shape through object action complexes,” *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 247–265, 2008.

[31] C. Geib, K. Mourão, R. Petrick, N. Pugeault, M. Steedman, N. Krueger, and F. Wörgötter, “Object action complexes as an interface for planning and robot control,” in *Workshop: Towards Cognitive Humanoid Robots at IEEE RAS Int Conf. Humanoid Robots*, (Genova), IEEE RAS, Dec. 2006.

[32] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, “Cognitive agents - a procedural perspective relying on the predictability of object-action-complexes (oacs),” *Robot. Auton. Syst.*, vol. 57, pp. 420–432, Apr. 2009.

[33] J. Piaget, “Piaget’s theory in PH mussen,” in *Carmichael’s manual of child psychology* (L. Carmichael and P. H. Mussen, eds.), pp. 703–732, Wiley, 1970.

[34] A. Bandura, *Social learning theory*. Prentice-Hall series in social learning theory, Prentice Hall, 1977.

[35] E. Erikson, *Childhood and society*. Norton, 1993.

[36] L. Kohlberg, *Education, Moral Development and Faith*. Journal of moral education, 1974.

[37] H. Daniels, *Literature Circles: Voice and Choice in the Student-Centered Classroom*. Pembroke Publishers, 1994.

[38] D. Wood, J. Bruner, and G. Ross, “The role of tutoring in problem solving,” *Journal of Child Psychiatry and Psychology*, vol. 17, no. 2, pp. 89–100, 1976.

[39] C. A. Haden, “Joint encoding and joint reminiscing: Implications for young children’s understanding and remembering of personal experiences,” in *Autobiographical memory and the construction of a narrative self: Developmental and cultural perspectives* (R. Fivush and C. Haden, eds.), pp. 49–69, NJ: Lawrence Erlbaum Associates, 2003.

[40] R. Zhao and M. Orey, *The scaffolding process: Concepts, features, and empirical studies*. Unpublished manuscript, University of Georgia, 1999.

[41] Y. Nagai, M. Asada, and K. Hosoda, “Learning for joint attention helped by functional development,” *Advanced Robotics*, vol. 20, no. 10, pp. 1165–1181, 2006.

[42] C. Muhl and Y. Nagai, “Does disturbance discourage people from communicating with a robot?,” in *The 16th IEEE International Symposium on Robot and Human Interactive Communication*, (Jeju, Korea), 26/08/2007 2007.

[43] Y. Nagai and K. Rohlfing, “Computational analysis of motionese toward scaffolding robot action learning,” *Autonomous Mental Development, IEEE Transactions on*, vol. 1, pp. 44 –54, may 2009.

[44] Y. Nagai, A. Nakatani, and A. M., "How a robot's attention shapes the way people teach," in *Proceedings of the 10th International Conference on Epigenetic Robotics* (R. Fivush and C. Haden, eds.), pp. 81–88, 2010.

[45] E. Ugur, H. Celikkanat, E. Sahin, Y. Nagai, and E. Oztop, "Learning to grasp with parental scaffolding.," in *Humanoids*, pp. 480–486, IEEE, 2011.

[46] B. Dufay and J. Latombe, "An approach to automatic robot programming based on inductive learning," *International Journal of Robotics Research, Winter 1984*, vol. 3, no. 4, pp. 3–20, 1984.

[47] A. Levas and M. Selfridge, "A user-friendly high-level robot teaching system," in *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, vol. 1, pp. 413–416, 1984.

[48] A. Segre and G. DeJong, "Explanation-based manipulator learning: Acquisition of planning ability through observation," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 555 – 560, mar 1985.

[49] A. M. Segre, *Machine learning of robot assembly plans*. Norwell, MA, USA: Kluwer Academic Publishers, 1988.

[50] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Teaching by showing: Generating robot programs by visual observation of human performance," in *Proceedings of the International Symposium of Indus- trial Robots*, vol. 2, pp. 119 – 126, mar 1989.

[51] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 799–822, 1994.

[52] K. Ikeuchi, M. Kawade, and T. Suehiro, "Assembly task recognition with planar, curved, and mechanical contacts.," in *ICRA (2)*, pp. 688–694, 1993.

[53] C. P. Tung and A. C. Kak, "Automatic learning of assembly tasks using a DataGlove system," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1, pp. 1–8 vol.1, 1995.

[54] A. Ng, H. Kim, M. Jordan, and S. Sastry, "Inverted autonomous helicopter flight via reinforcement learning," in *In International Symposium on Experimental Robotics*, MIT Press, 2004.

[55] J. Chen and A. Zelinsky, "Programing by demonstration: Coping with suboptimal teaching actions," 2003.

[56] T. Inamura, M. Inaba, and H. Inoue, "Acquisition of probabilistic behavior decision model based on the interactive teaching method," in *In Proceedings of the Ninth International Conference on Advanced Robotics (ICAR'99)*, 1999.

[57] A. Billard, S. Calinon, and G. F., "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[58] B. D. Argall, B. Browning, and M. Veloso, "Learning robot motion control from demonstration and human advice."

[59] D. H. Grollman and O. C. Jenkins, "Dogged learning for robots," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2483–2488, 2007.

[60] M. T. Rosenstein and B. A. G., "1 supervised actor-critic reinforcement learning."

[61] W. D. Smart, *Making reinforcement learning work on real robots*. PhD thesis, Providence, RI, USA, 2002. AAI3050968.

[62] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proceedings of the International Machine Learning Conference 1997*, ICML'97, pp. 12–20, 1997.

[63] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, pp. 93–108, June 2004.

[64] N. S. Pollard and J. K. Hodgins, "Generalizing demonstrated manipulation tasks," in *In Proceedings of the Workshop on the Algorithmic Foundations of Robotics (WAFR'02*, 2002.

[65] R. Dillmann, M. Kaiser, and A. Ude, "Acquisition of elementary robot skills from human demonstration," in *In International Symposium on Intelligent Robotics Systems*, pp. 185–192, 1995.

[66] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, "Towards robot cultures?: Learning to imitate in a robotic arm test-bed with dissimilarly embodied agents," *Interaction Studies*, pp. 3–44, 2004.

[67] S. Zhang, L. Xie, and M. D. Adams, "Entropy based feature selection scheme for real time simultaneous localization and map building.," in *IROS*, pp. 1175–1180, IEEE, 2005.

[68] A. Bur, A. Tapus, N. Ouerhani, R. Siegwart, and R. Hügli, "Robot navigation by panoramic vision and attention guided fetaures," in *ICPR (1)*, pp. 695–698, 2006.

[69] D. Floreano, "Complex Visual Behavior with Simple Brains: Co-evolutionary Active Vision and Feature Selection," in *Proceedings of European Cognitive Science Conference*, 2003.

[70] A. L. Edsinger, *Robot manipulation in human environments*. PhD thesis, Cambridge, MA, USA, 2007. AAI0818368.

[71] N. Vlassis, R. Bunschoten, and B. Krose, "Learning task-relevant features from robot data," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, pp. 499 – 504 vol.1, 2001.

[72] C. Wang and K. Wang, "Hand posture recognition using adaboost with sift for human robot interaction," in *Recent Progress in Robotics: Viable Robotic Service to Human* (S. Lee, I. Suh, and M. Kim, eds.), vol. 370 of *Lecture Notes in Control and Information Sciences*, pp. 317–329, Springer Berlin / Heidelberg, 2008.

[73] Y. Freund and R. E. Schapire, *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting*. 1995.

[74] J. Coelho, J. Piater, and R. Grupen, "Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot," 2000.

[75] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6d object localization for grasping with humanoid robot systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 919–924, 2007.

[76] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, "Visually-guided grasping while walking on a humanoid robot," in *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, (Roma, Italy), pp. 3041–3047, April 2007.

[77] G. Grisetti, G. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, p. 2007, 2007.

[78] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping known objects with humanoid robots: A box-based approach," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, 2009.

[79] E. Oztop, L. Lin, M. Kawato, and G. Cheng, "Dexterous skills transfer by extending human body schema to a robotic hand," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 82–87, dec. 2006.

[80] E. Ugur, H. Celikkanat, E. Sahin, Y. Nagai, and E. Oztop, "Learning to grasp with parental scaffolding," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pp. 480–486, oct. 2011.

[81] W. Erlhagen, A. Mukovskiy, E. Bicho, G. Panin, C. Kiss, A. Knoll, H. Vanschie, and H. Bekkering, "Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 353–360, 2006.

[82] C. Granville and J. Southerland, "Learning grasp affordances through human demonstration."

[83] T. Geng, M. Lee, and M. Hülse, "Transferring human grasping synergies to a robot," *Mechatronics*, vol. 21, no. 1, pp. 272–284, 2011.

[84] M. Ralph and M. Moussa, "Toward a natural language interface for transferring grasping skills to robots," *Robotics, IEEE Transactions on*, vol. 24, pp. 468–475, april 2008.

[85] M. Salganicoff, L. H. Ungar, and R. Bajcsy, "Active learning for vision-based robot grasping," *Machine Learning*, vol. 23, pp. 251–278, 1996. 10.1007/BF00117446.

[86] D. Kraft, R. Detry, N. Pugeault, E. Baseski, F. Guerin, J. Piater, and N. Kruandger, "Development of object and grasping knowledge by robot exploration," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, pp. 368–383, dec. 2010.

[87] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Kruger, O. Kroemer, J. Peters, and J. Piater, "Learning object-specific grasp affordance densities," in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, pp. 1–7, june 2009.

[88] R. Detry, D. Kraft, A. Buch, N. Kruger, and J. Piater, "Refining grasp affordance models by experience," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2287–2293, may 2010.

[89] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1–4, IEEE, 2011.

[90] A. K. Jain, *Fundamentals of digital image processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[91] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments," in *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, IROS'09, (Piscataway, NJ, USA), pp. 1–6, IEEE Press, 2009.

[92] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," pp. 171–182, Springer Verlag, 1994.

[93] M. Ebden, "Gaussian processes for regression: A quick introduction," *Available on Internet http://www.robots.ox.ac.uk*, vol. 33, no. August, pp. 137–157, 2008.

[94] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pp. 50–56, ACM, 2008.

[95] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal of Advanced Robotics Systems, special issue on Software Development and Integration in Robotics*, vol. 3, no. 1, 2006.

[96] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[97] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

APPENDIX A

Normal Distribution and Gaussian Processes

A.1 Normal Distribution

In the probability theory, normal distribution or Gaussian distribution is is a continuous distribution that has a density function, called as Gaussian function:

$$f(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

μ is the mean (or expectation) and σ^2 is the variance. σ is called as the standard deviation.

The multivariate Gaussian distribution is a generalization of the one-dimensional normal distribution to higher dimensions. In such a distribution, the variable x has more than one dimension. Thus, the density function becomes:

$$f_x(x_1, x_2, x_3, ..., x_n) = \frac{1}{2\pi^{\frac{k}{2}} |\Sigma|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$

where Σ is the covariance matrix of the multivariate normal distribution. Furthermore, $|\Sigma|$ is the determinant of the covariance matrix.

A.2 Gaussian Fitting for a Certain Dataset

Multivariate Gaussian fitting can be executed for training data which includes features and labels. Expectation Maximization (The EM algorithm) [97] can be used for this purpose.

In this algorithm, given a given function model such as Gaussian model or linear model and a training dataset X, a set of testing data without labels Z, and unknown labels Θ, the maximum likelihood estimate (MLE) of the labels is determined by the marginal likelihood of the training dataset.

The EM algorithm finds the estimate by applying the following two steps in an iterative manner:

Expectation step (E step): Calculate the expected value of the likelihood function, with respect to the conditional distribution of Z given X under the current estimate of the parameters:

$$Q(\Theta|\Theta^{(t)}) = E_{Z|X,\Theta^{(t)}}[log\ L(\Theta; X, Z)]$$

Maximization step (M step): Find the labels that maximizes the belief:

$$Q^{(t+1)} = argmax_{\Theta}\ Q(\Theta|\Theta^{(t)})$$

In our case, when a new data without labels is come in the testing phase (M step), the following procedure is done in order to predict the offsets:

If the fitted Gaussian's mean, μ and variance, Σ can be partitioned as follows (In our case, q is the number of offset labels, $N - q$ is the number of features):

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \text{ where } \mu_1 \text{ has } q \text{ rows and } \mu_2 \text{ has } N - q \text{ rows.}$$

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times q & q \times (N - q) \\ (N - q) \times q & (N - q) \times (N - q) \end{bmatrix}$$

then, the distribution of offsets which can be defined as X_1 conditional on features, X_2 is a multivariate normal $(X_1|X_2 = a) = N(\mu^{'}, \Sigma^{'})$ where

$$\mu^{'} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}\ (a - \mu_2)$$

$$\Sigma^{'} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

In such a case, $\mu^{'}$ denotes the prediction of offsets whereas $\Sigma^{'}$ is an indicator of quality of the prediction. Lower the variance, $\Sigma^{'}$ is, more certain the prediction is.

A.3 Gaussian Processes

The methodology of *Gaussian processes* (GPs) is a stochastic technique which associates random values with every data instance in a range of times (or of space) such that each such random variable together with the corresponding data instance has a normal distribution. Additionally, every collection of those random variables has a multivariate normal distribution.

A Gaussian process contains every set of instances of data $t_1, t_2, ..., t_n$ in T:

$$X_{t_1,t_2,...,t_k} = (X_{t_1}, ..., X_{t_k})$$

is a set of multivariate Gaussian random variable. The Gaussian property is shown as follows: $X_t; t\epsilon T$ is Gaussian if and only if, for every set of indices $t_1, t_2, ..., t_n$, there are σ_{lj} with $\sigma_{ii} > 0$ and μ_l such that

$$E(exp(i\sum_{l=0}^{k} t_l X_{t_l})) = -1\frac{1}{2}\sum_{l,j}\sigma_{l_j} t_l t_j + i\sum_{l}\mu_l t_l$$

Gaussian processes are widely-used in different areas because of its normal distribution-based roots. In nature, it is believed that most of phenomena and ingredients has normal distribution. Using GPs, some information can also be reached easily. For example, the distributions of various features can be obtained explicitly in a GP. Such features include: the average value of the process over a certain interval; the error in estimating the average using sample values at a small set of times.

A.4 Kronecker Delta Function

The Kronecker delta function is a function of two parameters, usually integers. The function is 1 if the parameters are equal to each other. Otherwise, the result is 0:

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j. \\ 0, & \text{otherwise.} \end{cases} \tag{A.1}$$