KTH Computer Science
and Communication

Multi-Modal Scene Understanding for Robotic Grasping

JEANNETTE BOHG

Doctoral Thesis in Robotics and Computer Vision
Stockholm, Sweden 2011

TRITA-CSC-A 2011:17
ISSN-1653-5723
ISRN-KTH/CSC/A–11/17-SE
ISBN 978-91-7501-184-4

Kungliga Tekniska högskolan
School of Computer Science and Communication
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi fredagen den 16 december 2011 klockan 10.00 i Sal D2, Kungliga Tekniska högskolan, Lindstedtsvägen 5, Stockholm.

Tryck: E-Print

Abstract

Current robotics research is largely driven by the vision of creating an intelligent being that can perform dangerous, difficult or unpopular tasks. These can for example be exploring the surface of planet mars or the bottom of the ocean, maintaining a furnace or assembling a car. They can also be more mundane such as cleaning an apartment or fetching groceries.

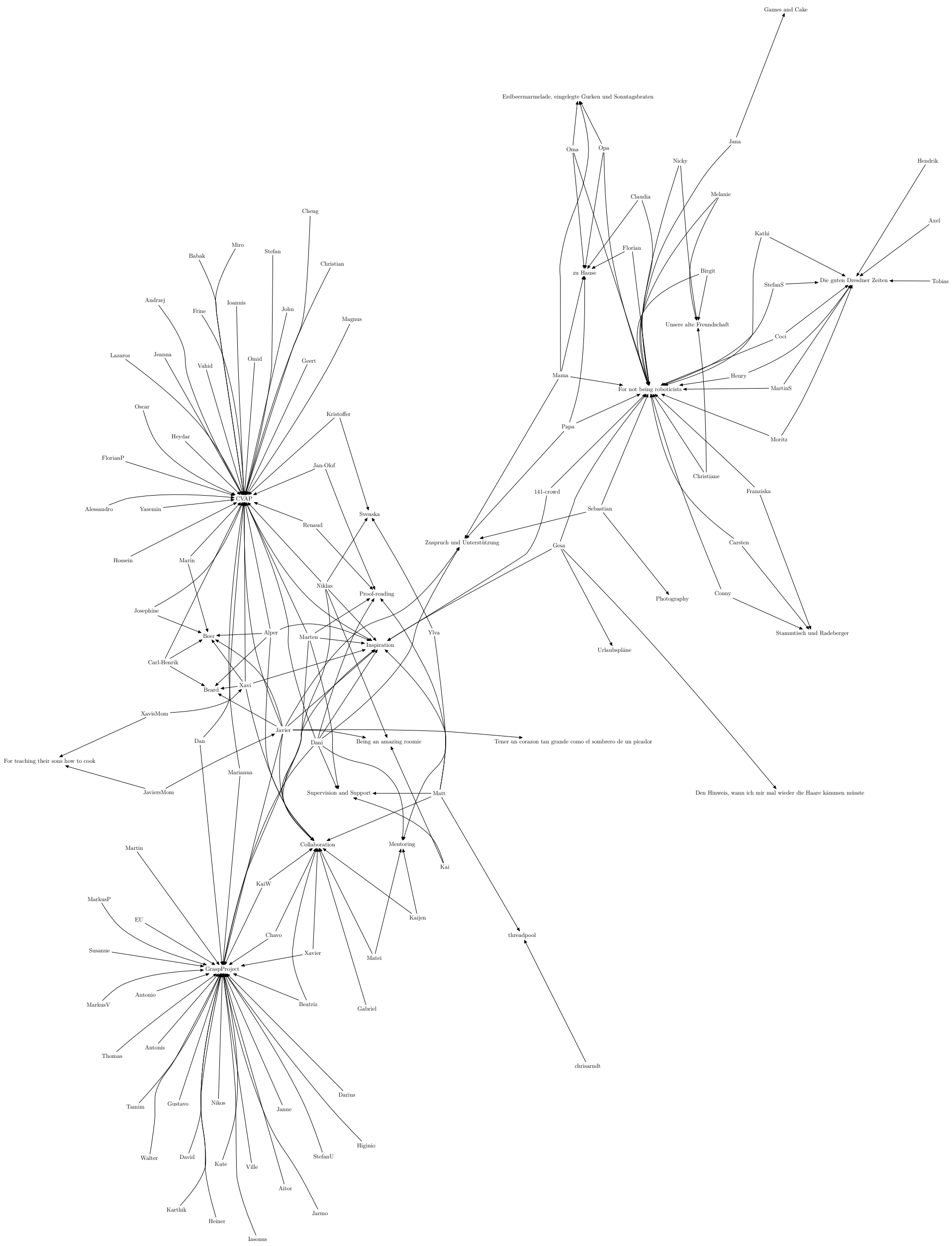
This vision has been pursued since the 1960s when the first robots were built. Some of the tasks mentioned above, especially those in industrial manufacturing, are already frequently performed by robots. Others are still completely out of reach. Especially, household robots are far away from being deployable as general purpose devices. Although advancements have been made in this research area, robots are not yet able to perform household chores robustly in unstructured and open-ended environments given unexpected events and uncertainty in perception and execution.

In this thesis, we are analyzing which perceptual and motor capabilities are necessary for the robot to perform common tasks in a household scenario. In that context, an essential capability is to understand the scene that the robot has to interact with. This involves separating objects from the background but also from each other. Once this is achieved, many other tasks become much easier. Configuration of objects can be determined; they can be identified or categorized; their pose can be estimated; free and occupied space in the environment can be outlined. This kind of scene model can then inform grasp planning algorithms to finally pick up objects. However, scene understanding is not a trivial problem and even state-of-the-art methods may fail. Given an incomplete, noisy and potentially erroneously segmented scene model, the questions remain how suitable grasps can be planned and how they can be executed robustly.

In this thesis, we propose to equip the robot with a set of prediction mechanisms that allow it to hypothesize about parts of the scene it has not yet observed. Additionally, the robot can also quantify how uncertain it is about this prediction allowing it to plan actions for exploring the scene at specifically uncertain places. We consider multiple modalities including monocular and stereo vision, haptic sensing and information obtained through a human-robot dialog system. We also study several scene representations of different complexity and their applicability to a grasping scenario.

Given an improved scene model from this multi-modal exploration, grasps can be inferred for each object hypothesis. Dependent on whether the objects are known, familiar or unknown, different methodologies for grasp inference apply. In this thesis, we propose novel methods for each of these cases. Furthermore, we demonstrate the execution of these grasp both in a closed and open-loop manner showing the effectiveness of the proposed methods in real-world scenarios.

Thank you, $x \xrightarrow{\text{for}} y$!



Contents

Notation

Throughout this thesis, the following notational conventions are used.

- Scalars are denoted by italic symbols, e.g., a, b, c.

- Vectors, regardless of dimension, are denoted in bold lower-case symbols, $\mathbf{x} = (x, y)^T$.

- Sets are indicated by calligraphic symbols, e.g., $\mathcal{P}, \mathcal{Q}, \mathcal{R}$. The cardinality of these sets is denoted by capital letters such as N, M, K. As a compact notation for a set $\mathcal{X}$ containing N vectors $\mathbf{x}_i$, we will use $\{\mathbf{x}_i\}_N$.

- In this thesis, we frequently deal with estimates of vectors. They are indicated by a hat superscript, e.g., $\hat{\mathbf{x}}$. They also can be referred to as *a posteriori* estimates in contrast to *a priori* estimates. The latter are denoted by an additional superscript. This can either be a minus, e.g., $\hat{\mathbf{x}}^-$ for estimates in time, or a plus, e.g., $\hat{\mathbf{x}}^+$ for estimates in space.

- Functions are denoted by italic symbols followed by their arguments in parentheses, e.g., $f(\cdot), g(\cdot), k(\cdot, \cdot)$. An exception is the normal distribution where we adopt the standard notation of $\mathcal{N}(\mu, \mathbf{\Sigma})$.

- Matrices, regardless of dimension, are denoted as bold-face capital letters, e.g., $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

- Frames of reference or coordinate frames are denoted by sans-serif capital letters, e.g., W, C. In this thesis, we convert coordinates mostly between the following three reference frames: the world reference frame W, the camera coordinate frame C and the image coordinate frame I. Vectors that are related to a specific reference frame are annotated with the according superscript, e.g., $\mathbf{x}^{\mathsf{W}}$.

- Sometimes in this thesis, we have to distinguish between the left and right image of a stereo camera. We label vectors referring to the left camera system with a subscript l and analogous with r for the right camera system, e.g., $\mathbf{x}_l, \mathbf{x}_r$.

- A finger on our robotic hand is padded with a tactile sensor matrices. One is on the *distal* and one on the *proximal* phalanx. Measurements from these sensors are either labeled with subscript d or p.

- The subscript t refers to a specific point in time.

- In this thesis, we develop prediction mechanism that enable the robot to make predictions about unobserved space in its environment. We label the set of locations that has already been observed with subscript k for *known*. The part that is yet unknown is labeled with subscript u.

1

Introduction

The idea of creating an artificial being is quite old. Even before the word *robot* got coined by Karel Čapek in 1920, the concept appeared frequently in literature and other artwork. The idea can already be observed in preserved records of ancient automated mechanical artifacts [206]. In the beginning of the 20th century, advancements in technology led to the birth of the *science fiction* genre. Among other topics, the robot became a central figure of stories told in books and movies [217, 82]. The first *physical* robots had to await the development of the necessary underlying technology in the 1960s. Interestingly, already in 1964 Nam June Paik in collaboration with Shuya Abe built the first robot to appear in the field of new media arts: Robot K456 [61, pg. 73].

The fascination of humans with robots stems from a dichotomy in how we perceive them. One the one hand, we strive for replicating our consciousness, intelligence and physical being but with characteristics that we associate with technology: regularity, efficiency, stability and power. Such an artificial being could then exist beyond human limitations, constraints and even responsibility [217]. In a way, this striving reflects the ancient search for immortal life. On the other hand, there is the fear that once such a robot exists we may loose control over our own creation. In fact, we would have created our own replacement [82].

Exactly this conflict is subject of most of the robot stories reaching from the classic ones like *R.U.R.* (Rossum's Universal Robots) by Karel Čapek and Fritz Lang's *Metropolis* to the newer ones such as James Cameron's *Terminator* and Alex Proyas' *I, Robot.* Geduld [86] divided the corpus of stories into material about "god-sanctioned" and "sacrilegious creation" to reflect the simultaneous attraction and recoil. He places the history of real automata completely aside from this.

In current robotics research, the striving for replicating human intelligence is completely embraced. Robots are celebrated as accomplishments that demonstrate human artifice. When reading the introductions of many recent books, PhD theses and articles on robotics research, the arrival of service robots in our everyday life is predicted to happen in the near future [192, 174, 121], already "tomorrow" [206] or is claimed to have already happened [177]. Rusu [192] and Prats [174] emphasize the necessity of service robotics with regard to the aging population of Europe, Japan and the USA. Machines equipped with ideal physical and cognitive capabilities are thought to care for people who became physically and cognitively limited or disabled. Although robotics celebrated its 50th birthday this year, only recently the topic of *roboethics* has been coined and a roadmap published by Veruggio [226]. In that sense, the art community is somewhat ahead of the robotics community in discussing implications of technological advancement.

The question why robotics research shows only little interest in these topics remains. We argue that this is because robots are not perceived by roboticists to have even remotely achieved the level of intelligence that we believe to be necessary to replace us. In reality, we are far away from having a general purpose robot as envisioned in many science fiction stories. Instead we are at the stage of the specific purpose robot. Especially in industrial manufacturing, robots are omnipresent in assembling specific products. Other robots have shown to be capable of autonomous driving in environments of different complexities and constraints. Regarding household scenarios, robots already vacuum-clean apartments [105]. Other tasks like folding towels [141], preparing pancakes [24] or clear a table as in the thesis at hand have been demonstrated on research platforms. However, to let robots perform these task robustly in unstructured and open-ended environments given unexpected events and uncertainty in perception and execution is an ongoing research effort.

1.1 An Example Scenario

Robots are good at many things that are hard for us. This involves the aspects mentioned before like regularity, efficiency, stability and power but also games like chess and jeopardy in which, although not robots in the strict sense, supercomputers have recently beaten human masters. However, robots appear to be not very good in many things that we perform effortlessly. Examples are grasping and dexterous manipulation of everyday objects or learning about new objects for recognizing them later on. In the following, we want to exemplify this with the help of a simple scenario.

Figure 1.1 shows an office desk. Imagine that you ordered a brand-new robotic butler and you want it to clean that desk for you. What kind of capabilities does it need for fulfilling this task? In the following, we list a few of them.

Recognition of Human Action and Speech Just like with a real butler who is new to your place, you need to give the robot a tour. This involves showing it all the rooms and explaining what function they have. Besides that, you may also want to point out places like cupboards in the kitchen or shoe cabinets in the corridor so that it knows where to store certain objects. For the robot to extract information from this tour, it needs to be able to understand human speech but also human actions like walking behavior or pointing gestures. Only then, it can generate the appropriate behavior by itself like for example following, asking questions or pointing at things.

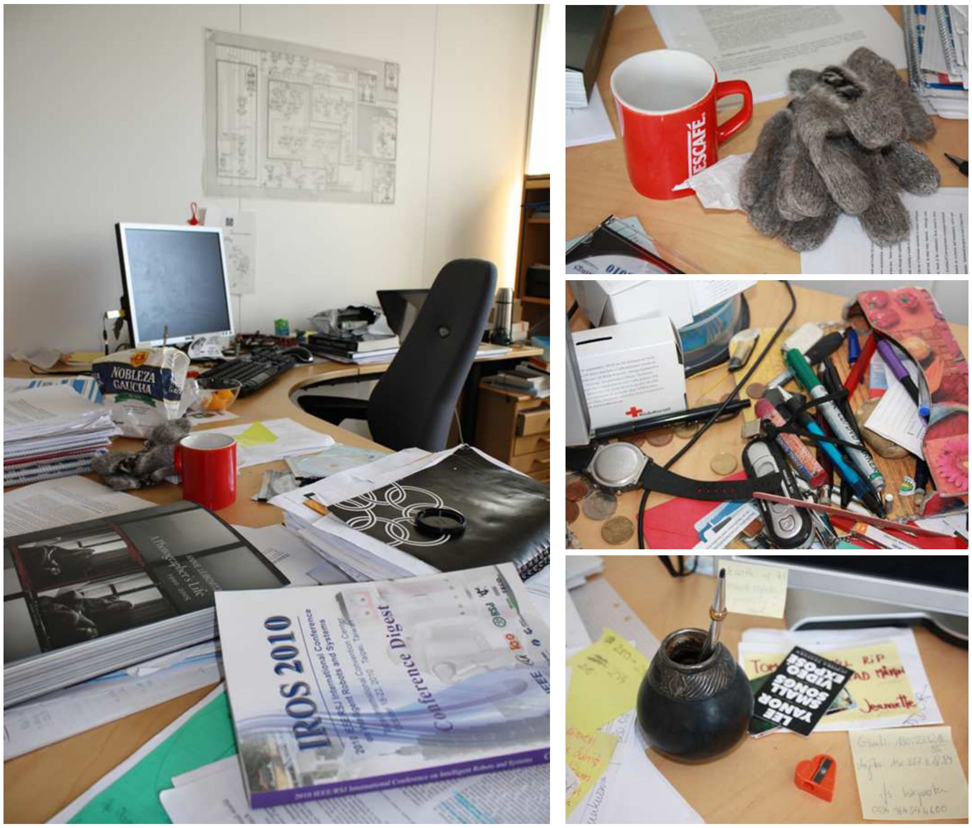


Figure 1.1: Left) Cluttered Office Desk to be Cleaned. Right Top) Gloves. Right Middle) Objects in a Pile. Right Bottom) Mate.

Navigation: Self-Localisation, Mapping and Path Planning During this tour, the robot needs to build a map of the new environment and localise itself in it. It has to store this map in its memory so that it can use it later for path planning or potential further exploration.

Semantic Perception and Scene Understanding You would expect this robot to already have some general knowledge about the world. It should for example know what a kitchen and a bath room is. It should know what shoes or different groceries are and where they usually can be found in an apartment. During your tour, this kind of *semantic* information should be linked to the map that the robot is simultaneously building of your place. For this, the robots needs to possess the ability to recognize and categorize rooms, furniture and objects using its different sensors.

Learning When the robot leaves its factory, it is probably equipped with a lot of knowledge about the world it is going to encounter. However, the probability that it is confronted with rooms of an ambiguous function and objects it has never seen before is quite high. The robot needs to detect these gaps in its world knowledge and attempt to close them. This may involve asking you to elaborate on something or to build a representation of an unknown object grounded in the sensory data.

Higher-Level Reasoning and Planning After the tour, the robot then may be expected to finally clean this desk shown in Figure 1.1. If we assume that it recognized the objects on the table, it now needs to plan what to do with them. It can for example leave them there, bring them to a designated place or throw them into the trash. To make these decisions, it has to consult its world knowledge in conjunction with the semantic map it has just built of you apartment. The cup for example may pose a conflict between its world knowledge (in which cups are usually in the kitchen) and the current map it has of the world. To resolve this conflict, it needs to form a plan to align these two. A plan may involve any kind of interaction with the world be it asking your for assistance, grasping something or keep exploring the environment.

Grasping and Manipulation Once the robot decided to pick up something to bring it somewhere else, it needs to execute this plan. Grasping and manipulation of objects demands the formation of low level motion plans that are robust on the one hand and flexible on the other. They need to be executed while facing noise, uncertainty and potentially unforeseen dynamic changes in the environment.

All of these capabilities in itself are the subject of ongoing research efforts in mostly separate communities. As the title of this thesis suggest, we are mainly concerned with scene understanding for robotic grasping and manipulation. Especially for scenes as depicted in Figure 1.1, this poses a challenging problem to the robot. Although a child would be able to accomplish to grasp the shown objects without any problem, it has yet not been shown that a robot could perform equally well without introducing strong assumptions.

1.2 Towards Intelligent Machines - A Historical Perspective

Since the beginning of robotics, pick and place problems from a table top have been studied [77]. Although the goal remained the same over the years, the different approaches were rooted in the paradigms on *Artificial Intelligence* (AI) of their time. One of the most challenging problems that has been approached over and over again, is to define what *intelligence* actually means. Recently, Pfeifer and Bongard [171] even refuse to define this concept given what they claim to be the mere absence of a good definition. In this section, we will briefly summarize some of the major views on intelligence.

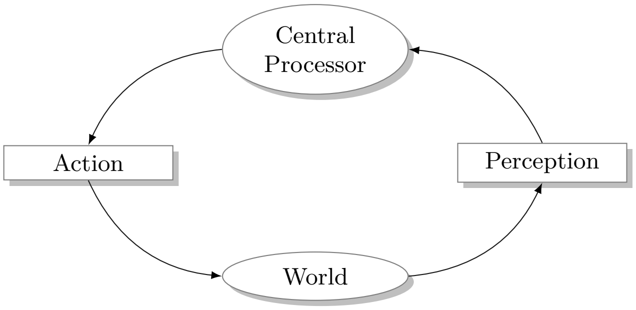


Figure 1.2: Traditional Model of an Intelligent System.

The traditional view is dominated by the *physical symbol system hypothesis*. It states that any system exhibiting intelligence can be proven to be a physical symbol system and any physical symbol system of sufficient size will exhibit intelligent behavior [160, 207]. Figure 1.2 shows a schematic of the traditional model of intelligence exhibited by humans or machines. The central component is a symbolic information processor that is able to manipulate a set of symbols potentially part of whole expressions. It can create, modify, reproduce or delete symbols. Essentially, such a symbol system can be modelled as a Turing machine. Therefore it follows that intelligent behavior can be exhibited by todays computers. The symbols are thought to be *amodal*, i.e., completely independent of the kind of sensors and actuators available to the system. The input to the central processor is provided by perceptual modules that convert sensory input into amodal symbols. The output is generated though action modules that use amodal symbolic descriptions of commands.

Turing [221] proposed an imitation game to re-frame the question "Can machines think?" that is now known as the famous Turing Test. The proposed machines are capable of reading, writing and manipulating text without the need to understand its meaning. The problem of converting between real world percepts and amodal symbols is thereby circumvented as well as conversion between symbolic commands and real world motor commands. The hope was that this will be solved at some later point. In the early days of AI research, a lot of progress was made in for example theorem proofers, blocks world scenarios or artificial systems that could trick humans to believe that they were humans as well. However, early approaches could not be shown to translate to more complex scenarios [191]. In [171], Brooks noted that "Turing modeled what a person does, not what a person thinks". Also the behavioral studies on which Simon's thesis [207] of physical symbol systems rest, can be seen in that light: humans solving cryptarithmetic problems, memorizing or performing tasks that use natural language.

The view of physical symbol system has been challenged on several grounds. In the 1980s connectionist models became largely popular [191]. The most prominent

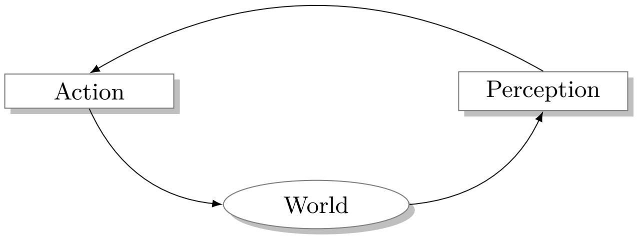


Figure 1.3: Model of an Intelligent System from the Viewpoint of Behavioral Robotics.

representative is the artificial neural network that is seen as a simple model of the human brain. Instead of a central processor, a neural network consists of a number of simple interconnected units. Intelligence is seen as emergent within this network. Connectionist models have been shown to be far less domain specific and more robust to noise. However according to Simon [207], it has yet to be shown that complex thinking and problem solving can be modelled with connectionist approaches. Nowadays, symbol systems and connectionist approaches are seen as complementary [191].

A different case has been argued by the supporters of the *Physical Symbol Grounding Hypothesis*. Instead of criticizing the structure of the proposed intelligent system, it re-considers the assumed amodality of the symbols. Its claim is that to build a system that is truly intelligent, it has to have its symbols grounded in the physical world. In other words, it has to *understand* the meaning of the symbols [96]. However, no perception system has yet been build that robustly outputs a general purpose symbolic representation of the perceived environment. Furthermore, perception has been claimed to be active and task dependent. Given different tasks, different parts of the scene or different aspects of it may be relevant [22]. The the same percept can be interpreted differently. Therefore, no objective truth exists [159].

Brooks [46, 47] has carried this view to its extreme and asks whether intelligence needs any representations at all. Figure 1.3 shows the model of an intelligent machine that completely abondons the need for representations. Brooks coined the sentence: *The world is its own best model*. The more details of the world you store, the more you need to keep up-to-date. He proposes to dismiss any intermediate representation that interfaces between the world and computational modules as well as in between computational modules. Instead, the system should be decomposed into subsystem by activities that describe patterns of interactions with the world. As outlined in Figure 1.3, each subsystems tightly connects sensing to actions, and senses whatever necessary and often enough to accomodate for dynamic changes in the world. No information is transferred between submodules. Instead, they

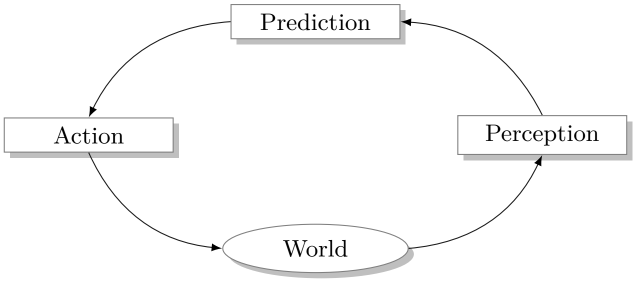


Figure 1.4: Model of an Intelligent System from the Viewpoint of Grounded Cognition

compete for control of the robot through a central system. People have challenged this view as being only suitable for creating insect-like robots without a proof that higher cognitive functions are possible to achieve [51]. This approach is somehow similar to the Turing test in that it is claimed that if the behavior of the machine is intelligent in the eye of the beholder, then it is intelligent. Parallels can also be drawn to Simon [207] in which an intelligent system is seen as essentially a very simple mechanism. Its complexity is a mere reflection of the complexity of the environment.

Figure 1.4 shows the schematic of an intelligent agent as envisioned for example in psychology and neurophysiology. The field of *Grounded Cognition* focuses on the role of simulation as exhibited by humans in behavioral studies [23]. The notion of amodal symbols that reside in semantic memory separate from the brain's modal system for perception, action and introspection is rejected. Symbols are seen as to be grounded and stored in a multi-modal way. Simulation is a re-enactment of perceptual, motor and introspective states acquired during a specific experience with the world, body and mind. It is triggered when knowledge is needed to represent a specific object, category, action or situation. In that sense it is compatible with other theories as for example the *simulation theory* [84] or the *common-coding* theory of action and perception [176]. Through the discovery of the so-called *Mirror neurons* (MNs), these models gained momentum [185]. MNs have first been discovered in the F5 area of a monkey brain. During the first experiments, this area has been monitored during grasping actions. It was discovered that there are neurons that fire both, when a grasping action is performed by the recorded monkey and when this monkey observes the same action performed by another individual. Interestingly, this action has to be goal directed which is in accordance with the common-coding theory. Its main claim is that perception and action share a common representational domain. Evidence from behavioral studies on humans showed that actions are planned and controlled in terms of their effect [176]. There is also strong evidence that MNs exist in the human brain. Etzel et al. [74] tested the simulation theory through standard machine learning techniques. The authors trained a classifier on functional magnetic resonance imaging (fMRI) recorded from humans while hearing either of two actions. They showed that the same classifier can also distinguish between these actions when provided fMRI data that is recorded from the same person performing them.

Up till now, no agreement has been reached on whether a central representational system is used or a distributed one. Also no computational model has yet been proposed in the field of psychology or neurophysiology [23]. However, there are some examples in the field of robotics following the general concept of grounded cognition. Ballard [22] introduces the concept of *Animate Vision* that aims at understanding visual processing in the context of the tasks and behaviors the vision system is currently engaged in. In this paradigm, the need for maintaining a detail model of the world is avoided. Instead, real-time computation of a set of simple visual features is emphasized such that task relevant representations can be rapidly computed on demand. Christensen et al. [51] present a system for keeping multi-modal representations of an entity in memory and refining them in parallel. Instead of one monolithic memory system, representations are shared among subsystems. A binder is proposed for connecting each representation to one entity in memory. This can be related to the *Prediction* box in Figure 1.4. Through this binding, perception of one modality corresponding to an entity could predict the other modalities associated with the same entity in memory. Another example is presented by Kjellström et al. [116]. The authors show that the simultaneous observation of two separate modalities (action and object features) helps to categorize both of them.

1.3 This Thesis

We are specifically concerned with perception for grasping and manipulation. The previously mentioned theories stemming from psychology and neurophysiology are studying the tight link between perception and action. They place the concept of prediction through simulation into their focus. In this thesis, we will equip a robot with mechanisms to predict unobserved parts of the environment and the effects of its actions. We will show how this helps to efficiently explore the environment but also how it improves perception and manipulation.

We will study this approach in a table top scenario populated with either known, unknown or familiar objects. As already outlined in the example scenario in Section 1.1, scene understanding is one important capability of a robot. In specific this involves segmentation of objects from each other and from the background. Once this is achieved, many other tasks become much easier. Configuration of objects on the table can be determined; they can be identified or categorized; their pose can be determined; free and occupied space in the environment can be outlined. This kind of scene model can then inform grasp inference mechanisms to finally grasps objects from the table top.

In Figure 1.1, we can observe why scene understanding is challenging. On the

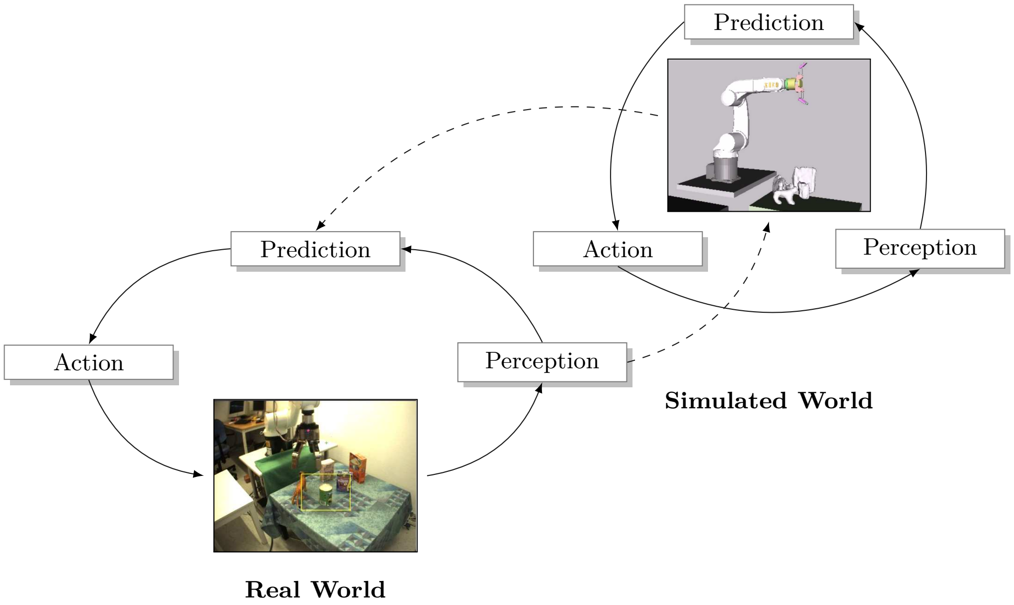


Figure 1.5: Schematic of this thesis. Given some observations of the real world, the state of the world can be estimated. This can be used to predict unobserved parts of the scene (Prediction box) as formalized in Equation 1.1. It can be used to predict action outcomes (Action box) as formalized in Equation 1.2 and to predict what certain sensors should perceive (Perception box) as in Equation 1.3 and 1.4.

depicted office desk, there are piles of clutter for which even humans have problems to separate them visually into single objects. Deformable objects like gloves are shown that come in very many different shapes and styles. Still we are able to recognize them correctly even though we might not have seen this specific pair ever before. And there are objects that are completely unknown to us. Independent of this, we would be able to grasp them.

Figure 1.5 is related to Figure 1.4 by adopting the notion of prediction into the perception-action loop. On the left side, you see the real world. It can be perceived through the sensors the robot is equipped with. This is an active process in which the robot can choose what to perceive by moving through the environment or by changing it using its different actuators. On the left, you see a visualisation of the current model the robot has of its environment.

This model makes simulation explicit in that perception feeds into memory to update a multi-modal representation of the current scene of interest. This scene is estimated at discrete points in time t to be in a specific state $\hat{\mathbf{x}}_t$. This estimate is then used to implement prediction in three different ways.

- Based on earlier observations, we have a partial estimate of the state of the scene $\hat{\mathbf{x}}_t$. Given this estimate and some prior knowledge, we can predict unobserved parts of it to obtain an *a priori* state estimate:

$$\hat{\mathbf{x}}_t^+ = g(\hat{\mathbf{x}}_t) \tag{1.1}$$

- Given the current state estimate, some action u and a process model, we can predict how the scene model will change over time to obtain a different *a priori* state estimate:

$$\hat{\mathbf{x}}_{t+1}^- = f(u, \hat{\mathbf{x}}_t) \tag{1.2}$$

- Given an *a priori* state estimate of the scene, we can predict what different sensor modalities should perceive

$$\hat{\mathbf{z}}_t = h(\hat{\mathbf{x}}_t^+) \tag{1.3}$$

  or

$$\hat{\mathbf{z}}_{t+1} = h(\hat{\mathbf{x}}_{t+1}^-) \tag{1.4}$$

These functions can be associated with the boxes positioned around the simulated environment on the right hand side of Figure 1.5. While the *Prediction* box refers to prediction in space and is related to Equation 1.1, the *Action* box stands for prediction in time and the implementation of Equation 1.2. The *Perception* box represents the prediction of sensor measurements as in Equation 1.3 and 1.4. The scene model serves as a container representation that integrates multi-modal sensory information as well as predictions. In this thesis, we study different kinds of scene representations and their applicability for grasping and manipulation.

Independent of the specific representation, we are not only interested in a point estimate of the state. Instead, we consider the world state as a random variable that follows a distribution $\mathbf{x}_t \sim \mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{\Sigma}_t)$ with the state estimate as the mean and a variance $\mathbf{\Sigma}_t$. This variance is estimated based on the assumed divergence between the output of the functions $g(\cdot), f(\cdot)$ and $h(\cdot)$ and the unknown real value. It quantifies the uncertainty of the current state estimate.

In this thesis, we are proposing approaches that implement the function $g(\cdot)$ in Equation 1.1 to make predictions about space. We will demonstrate that the resulting a priori state estimate $\hat{\mathbf{x}}_t^+$ and the associated variance $\mathbf{\Sigma}_t^+$ can be used for either (i) guiding *exploratory* actions to confirm particularly uncertain areas in the current scene model, (ii) comparing predicted observations $\hat{\mathbf{z}}_t$ or actions outcomes with the real observations $\mathbf{z}_t$ and update the state estimate and variance accordingly to obtain an *a posteriori* estimate or (iii) for *exploiting* the prediction in using it for executing an action to achieve a given goal.

1.4 Outline and Contributions

This thesis is structured as follows:

Chapter 2 – Foundations

Chapter 2 introduces the hardware and software foundation of this thesis. The applied vision systems have been developed and published prior to this thesis and are only briefly introduced here.

Chapter 3 – Active Scene Understanding

The chapter introduces the problem of scene understanding and motivates the approach followed in this thesis.

Chapter 4 – Predicting and Exploring a Scene

Chapter 4 is the first of the three chapters that propose different prediction mechanisms. Here, we study a classical occupancy grid and a height map as a representation for a table top scene. We propose a method for multi-modal scene exploration where initial object hypotheses formed by active *visual* segmentation are confirmed and augmented through *haptic* exploration with a robotic arm. We update the current belief about the state of the map with the detection results and predict yet unknown parts of the map with a Gaussian Process. We show that through the integration of different sensor modalities, we achieve a more complete scene model. We also show that the prediction of the scene structure leads to a valid scene representation even if the map is not fully traversed. Furthermore, we propose different exploration strategies and evaluate them both in simulation and on our robotic platform.

Chapter 5 – Enhanced Scene Understanding through Human-Robot Dialog

While Chapter 4 is more focussed on the prediction of occupied and empty space in the scene, this chapter proposes a method for improving the segmentation of objects from each other. Our approach builds on top of state-of-the-art computer vision segmenting stereo reconstructed point clouds into object hypotheses. In collaboration with Matthew Johnson-Roberson and Gabriel Skantze this process is combined with a natural dialog system. By putting a ‘human in the loop’ and exploiting the natural conversation of an advanced dialogue system, the robot gains knowledge about ambiguous situations beyond its own resolution. Specifically, we are introducing an entropy-based system allowing the robot to predict which object hypotheses might be wrong and query the user for arbitration. Based on the information obtained from the human-to-robot dialog, the scene segmentation can be re-seeded and thereby improved. We analyse quantitatively what features are reliable indicators of segmentation quality. We present experimental results on real data that show an improved segmentation performance compared to segmentation without interaction and compared to interaction with a mouse pointer.

Chapter 6 – Predicting Unknown Object Shape

The purpose of scene understanding is to plan successful grasps on the object hypotheses. Even if these hypotheses are correctly corresponding only to one object, we commonly do not know the geometry of their backside. The approach to object shape prediction proposed in this chapter aims at closing the knowledge gaps in the robot's understanding of the world. It is based on the observation that many objects in a service robotic scenario possess symmetries. We search for the optimal parameters of these symmetries given visibility constraints. Once found, the point cloud is completed and a surface mesh reconstructed. This can then be provided to a simulator in which stable grasps and collision-free movements are planned. We present quantitative experiments showing that the object shape predictions are valid approximations of the real object shape.

Chapter 7 – Generation of Grasp Hypotheses

In this chapter, we will show how the object hypotheses that have been formed through different methods can be used to infer grasps. In the first section, we review existing approaches divided into three categories: grasping known, unknown or familiar objects. Given these different kinds of prior knowledge, the problem of grasp inference reduces to either object recognition and pose estimation, finding similarity metric between functionally similar objects or developing heuristics for mapping grasps to object representations. For each case, we propose our own methods that extend the existing state of the art.

Chapter 8 – Grasp Execution

Once grasp hypotheses have been inferred, they need to be executed. We demonstrate the approaches proposed in Chapter 7 in an open-loop fashion. In collaboration with Beatriz Léon and Javier Felip, experiments on different platform are performed. This is followed by a review of the existing closed-loop approaches towards grasp execution. Although there are a few exceptions, they are usually focussed on either the reaching trajectory or the grasp itself. We will demonstrate in collaboration with Xavi Gratal and Javier Romero visual and virtual visual servoing to execute grasping of known or unknown objects. This helps in controlling the reaching trajectory such that the end effector is accurately aligned with the object prior to grasping.

1.5 Publications

Parts of this thesis have previously been published as listed in the following.

Conferences

[1] Niklas Bergström, Jeannette Bohg, and Danica Kragic. Integration of visual cues for robotic grasping. In *Computer Vision Systems*, volume 5815 of *Lecture Notes in Computer Science*, pages 245–254. Springer Berlin / Heidelberg, 2009.

[2] Jeannette Bohg and Danica Kragic. Grasping familiar objects using shape context. In *International Conference on Advanced Robotics (ICAR)*, pages 1 –6, Munich, Germany, June 2009.

[3] Jeannette Bohg, Matthew Johnson-Roberson, Mårten Björkman, and Danica Kragic. Strategies for Multi-Modal Scene Exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4509 –4515, October 2010.

[4] Jeannette Bohg, Matthew Johnson-Roberson, Beatriz León, Javier Felip, Xavi Gratal, Niklas Bergström, Danica Kragic, and Antonio Morales. Mind the Gap - Robotic Grasping under Incomplete Observation. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[5] Matthew Johnson-Roberson, Jeannette Bohg, Mårten Björkman, and Danica Kragic. Attention-based active 3d point cloud segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1165 –1170, October 2010.

[6] Matthew Johnson-Roberson, Jeannette Bohg, Gabriel Skantze, Joakim Gustafson, Rolf Carlson, Babak Rasolzadeh, and Danica Kragic. Enhanced visual scene understanding through human-robot dialog. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011.

[7] Beatriz León, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisio, Jeannette Bohg, James Kuffner, and Rüdiger Dillmann. OpenGRASP: A toolkit for robot grasping simulation. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots(SIMPAR)*, November 2010. Best Paper Award.

Journals

[8] Jeannette Bohg and Danica Kragic. Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362 – 377, 2010.

[9] Jeannette Bohg, Carl Barck-Holst, Kai Huebner, Babak Rasolzadeh, Maria Ralph, Dan Song, and Danica Kragic. Towards grasp-oriented visual perception for humanoid robots. *International Journal on Humanoid Robotics*, 6(3): 387–434, September 2009.

[10] Xavi Gratal, Javier Romero, Jeannette Bohg, and Danica Kragic. Visual servoing on unknown objects. *IFAC Mechatronics: The Science of Intelligent Machines*, 2011. To appear.

Workshops and Symposia

[11] Niklas Bergström, Mårten Björkman, Jeannette Bohg, Matthew Johnson-Roberson, Gert Kootstra, and Danica Kragic. Active scene analysis. In *Robotics Science and Systems (RSS'10) Workshop on Towards Closing the Loop: Active Learning for Robotics*, June 2010. Extended Abstract.

[12] Jeannette Bohg, Niklas Bergström, and Mårten Björkman Danica Kragic. Acting and Interacting in the Real World. In *European Robotics Forum 2011: RGB-D Workshop on 3D Perception in Robotics*, April 2011. Extended Abstract.

[13] Xavi Gratal, Jeannette Bohg, Mårten Björkman, and Danica Kragic. Scene representation and object grasping using active vision. In *IROS'10 Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*, October 2010.

[14] Matthew Johnson-Roberson, Gabriel Skantze, Jeannette Bohg, Joakim Gustafson, Rolf Carlson, and Danica Kragic. Enhanced visual scene understanding through human-robot dialog. In *2010 AAAI Fall Symposium on Dialog with Robots*, November 2010. Extended Abstract.

2

Foundations

In this thesis, we present methods to incrementally build a scene model suitable for object grasping and manipulation. This chapter introduces the foundations of this thesis. We first present the hardware platform that was used to collect data as well as to evaluate and demonstrate the proposed methods. This is followed by a brief summary of the real-time active vision system that is employed to visually explore a scene. Two different segmentation approaches using the output of the vision system are presented.

2.1 Hardware

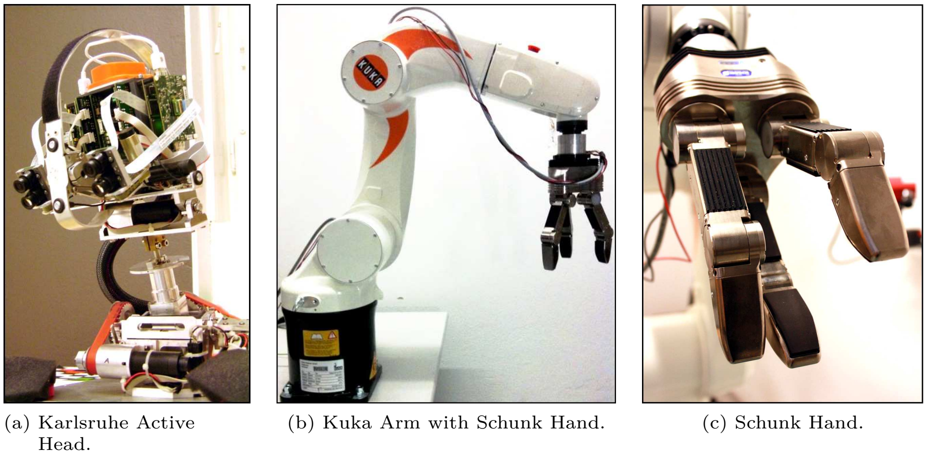
The main components of the robotic platform used throughout this thesis are the Karlsruhe active head [17] and a Kuka arm [126] equipped with a Schunk Dexterous Hand 2.0 (SDH) [203] as shown in Figure 2.1. This embodiment enables the robot to perform a number of actions like saccading, fixating, tactile exploration and grasping.

2.1.1 Karlsruhe Active Head

We are using a robotic head [17] that has been developed as part of the Armar III humanoid robot as described in Asfour et al. [16]. It actively explore the environment through gaze shifts and fixation on objects. The head has 7 DoF. For performing gaze shift, the lower pitch, yaw, roll as well as the eye pitch are used. The upper pitch is kept static. To fixate, the right and left eye yaw are actuated in a coupled manner. Therefore, only 5DoF are effectively used.

The head is equipped with two stereo camera pairs. One has a focal length of $4mm$ and therefore a wide field of view. The other one has a focal length of $12mm$ providing the vision system with a close up view of objects in the scene. For example images, see Figure 2.2.

To model the scene accurately from stereo measurements, the relation between the two camera systems as well as between the cameras and the head origin needs to be determined after each head movement. This is different from stereo cameras with a static epipolar geometry that need to be calibrated only once. In the



(a) Karlsruhe Active Head.

(b) Kuka Arm with Schunk Hand.

(c) Schunk Hand.

Figure 2.1: Hardware Components of the Robotic Platform

case of the active head, these transformations should ideally be obtainable from the known kinematic chain and the readings from the motor encoders. In reality however, these readings are affected by random and systematic errors. The latter are due to inaccuracies in the kinematic model and random errors are induced by noise in the motor movements. In our previous work [10], we describe how through controlled movements of the robotic arm and of the head the whole kinematic chain is calibrated. The arm essentially describes a pattern that is uniformly distributed in image space as well as in the depth of field. We use an LED rigidly attached to the hand for tracking the positions of the arm relative to the camera. Through this procedure, we obtain a set of 3D positions relative to the robot arm and the corresponding projections of these point on the image plane. From these correspondences, we use standard approaches as implemented in [43] to solve for the intrinsic and extrinsic stereo camera parameters. This process is repeated for different joint angles of the head to calibrate the kinematic chain. Thereby, systematic errors are significantly reduced.

Regarding random error, the last five joints in the kinematic chain achieve a repeatability in the range of ±0.025 degrees [17]. The neck pitch and neck roll joints achieve a repeatability in the range of ±0.13 and ±0.075 degrees respectively. How we cope with this noise will be described in more detail in Section 2.2.

2.1.2 Kuka Arm

The Kuka arm has 6 DoF and is the most reliable component of the system. It has a repeatability of less than 0.03 mm [126].

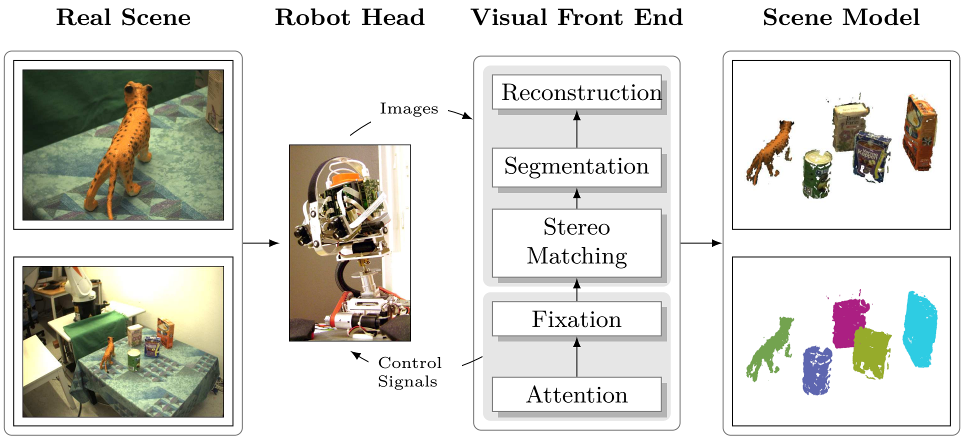


Figure 2.2: Overview of the Input and Output of the Visual Front End. Left) Foveal and Peripheral Images of a Scene. Middle Left) Karlsruhe Active Head. Middle Right) Visual Front End. Right) Segmented 3D Point Cloud.

2.1.3 Schunk Dexterous Hand

The SDH has three fingers each with two joints. An additional degree of freedom executes a coupled rotation of two fingers around their roll axis. Each finger is padded with two tactile sensor arrays: one on the distal phalanges with $6 \times 13 - 10$ cells and one on the proximal phalanges with 6×14 cells [203].

2.2 Vision Components

As input to all the methods that are proposed in this thesis, we expect a 3D point cloud of the scene. To produce such a representation, we use a real-time active vision system running on the Karlsruhe active head. In the following, we will introduce this system and present two different ways in which this representation is segmented into background and several object hypotheses.

2.2.1 Real-Time Vision System

In Figure 2.2, we show the structure of the real-time vision system that is capable of gaze shifts and fixation. Segmentation is tightly embedded into this system. In the following, we will briefly explain all the components. For a more detailed description, we refer to Rasolzadeh et al. [182], Björkman and Kragic [35], Björkman and Eklundh [33].

2.2.1.1 Attention

As mentioned in Section 2.1.1, our vision system consists of two stereo camera pairs, a peripheral and a foveal one. Scene search is performed in the wide-field camera by computing a *saliency map* on it. This map is computed based on the Itti & Koch model [106] and predicts the conspicuity of every position in the visual input. An example for such a map is given in Figure 2.3a. Peaks in this saliency map are used to trigger a saccade of the robot head such that the foveal cameras are centered on this peak.

2.2.1.2 Fixation

When a rapid gaze shift to a salient point in the wide-field is completed, the fixation process is immediately started. The foveal images are initially rectified using the vergence angle read from the calibrated left and right eye yaw joint. This rectification is then refined online by matching Harris' corner features extracted from both views and computing an affine essential matrix. The resulting rectified images are then used for stereo matching [43]. A disparity map on the foveal image in Figure 2.2 is given in Figure 2.3c. The vergence angle of the cameras is controlled such that the highest density of points close to the center of the views are placed at zero disparity.

2.2.1.3 Segmentation

For 3D object segmentation we use a recent approach by Björkman and Kragic [35]. It relies on three possible hypotheses: figure, ground and a flat surface. It is assumed that most objects are placed on flat surfaces thereby simplifying segregation of the object from its supporting plane.

The segmentation approach is an iterative two-stage method that first performs pixel-wise labeling using a set of model parameters and then updates these parameters in the second stage. This is similar to Expectation-Maximization with the distinction that instead of enumerating over all combinations of labelings, model evidence is summed up on a per-pixel basis using marginal distributions of labels obtained using belief propagation.

The model parameters consists of the following three parts, corresponding to the foreground, background and flat surface hypothesis:

$$\begin{aligned}
\theta_f &= \{\mathbf{p}_f, \mathbf{\Sigma}_f, \mathbf{c}_f\},\\
\theta_b &= \{d_b, \mathbf{\Sigma}_b, \mathbf{c}_b\},\\
\theta_s &= \{\alpha_s, \beta_s, \delta_s, \mathbf{\Sigma}_s, \mathbf{c}_s\}.
\end{aligned}$$

p_f denotes the mean 3D position of the foreground. d_b is the mean disparity of the background, with the spatial coordinates assumed to be uniformly distributed. The surface disparities are assumed to be linearly dependent on the image coordinates, i.e. $d = \alpha_s x + \beta_s y + \delta_s$. All these spatial parameters are modeled as normal distributions, with $\mathbf{\Sigma}_f$, $\mathbf{\Sigma}_b$ and $\mathbf{\Sigma}_s$ being the corresponding covariances. The last three parameters, $\mathbf{c}_f$, $\mathbf{c}_b$ and $\mathbf{c}_s$, are represented by color histograms expressed in hue and saturation space.

For initialization, there has to be some prior assumption of what is likely to belong to the foreground. In this thesis, we have a fixating system and assume that points close to the center of fixation are most likely to be part of the foreground. An imaginary 3D ball is placed around this fixation point and everything within the ball is initially labeled as foreground. For the flat surface hypothesis, RANSAC [80] is applied to find the most likely plane. The remaining points are initially labeled as background points.

Other approaches to initializing new foreground hypotheses for example through human-robot dialogue or motion of objects induced bei either a person or the robot itself are presented in [28, 29, 140]. Furthermore, these articles present the extension of the segmentation framework to keep multiple object hypotheses segmented simultaneously.

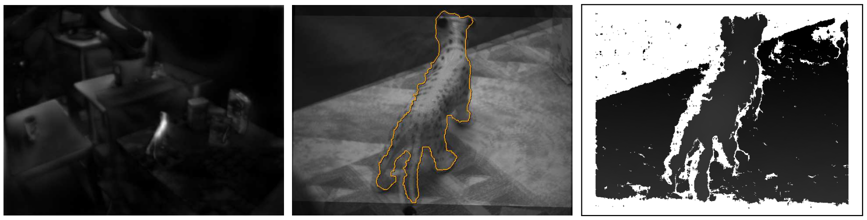
2.2.1.4 Re-Centering

The attention points that are peaks in the saliency map on wide-field images, tend to be on the border of objects rather than on their center. Therefore, when performing a gaze shift, the center of the foveal images does not necessarily correspond to the center of the objects. We perform a re-centering operation to maximize the amount of an object visible in the foveal cameras. This is done by letting the iterative segmentation process stabilize for a specific gaze direction of the head. Then the center of mass of the segmentation mask is computed. A control signal is sent to the head to correct its gaze direction such that the center of the foveal images is aligned with the center of segmentation. After this small gaze-shift has been performed, the fixation and segmentation process are re-started. This process is repeated until the center of the segmentation mask is sufficiently aligned with the center of the images.

An example for the resulting segmentation is given in Figure 2.3b, in which the object boundaries are drawn on the overlayed left and right rectified images of the foveal cameras. Examples for the point clouds calculated from the segmented disparity map are depicted in Figure 2.4. A complete labeled scene is shown on the right of Figure 2.2.

2.2.2 Attention-based Segmentation of 3D Point Clouds

In this section, we present an alternative approach to the segmentation of point clouds into object hypotheses as described in the previous section. It uses a Markov Random Field (MRF) graphical model framework. This paradigm allows for the identification of multiple object hypotheses simultaneously and is described in full detail in [5]. Here, we will only give a brief overview.

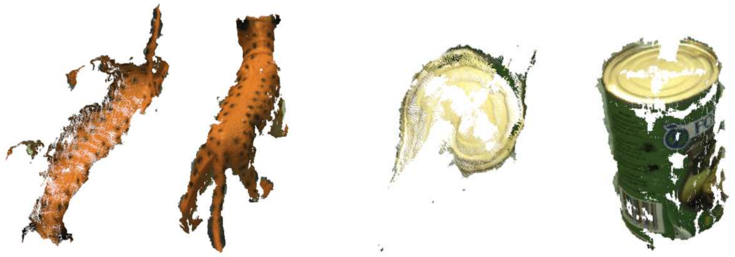


(a) Saliency Map of Wide Field Image.

(b) Segmentation on Overlayed Fixated Rectified Left and Right Images.

(c) Disparity Map.

Figure 2.3: Example Output for Visual Processes run on Wide-Field and Foveal Images shown in Figure 2.2



(a) Point Cloud of a Toy Tiger

(b) Point Cloud of Mango Can

Figure 2.4: Example 3D Point Cloud (from two Viewpoints) generated from Disparity Map and Segmentation as shown in Figure 2.3c and 2.3b.

The active humanoid head uses saliency to direct its gaze. By fully reconstructing each stereo view after each gaze shift and merging these resulting partial point clouds into one, we obtain scene reconstructions as shown in Figure 2.5. The fixation points serve as seed points that we project into the point cloud to create initial clusters for the generation of object hypotheses.

For full segmentation we perform energy minimization in a multi-label MRF. We use the multi-way cut framework as proposed in Boykov et al. [41]. In our application, the MRF is modelled as a graph with two sets of costs for assigning a specific label to a node in that graph: unary costs and pairwise costs.

In our case, the unary cost describes the likelihood of membership to an object hypothesis' color distribution. This distribution is modelled by Gaussian Mixture Models (GMMs) as utilized in GrabCuts by Rother et al. [188]. For each salient region one GMM is created to model the color properties of that object hypothesis.

Pairwise costs enforce smoothness between adjacent labels. The pairwise structure of the graph is derived from a KD-tree neighborhood search directly on the point cloud. The 3D structure provides the links between points and enforces neigh-

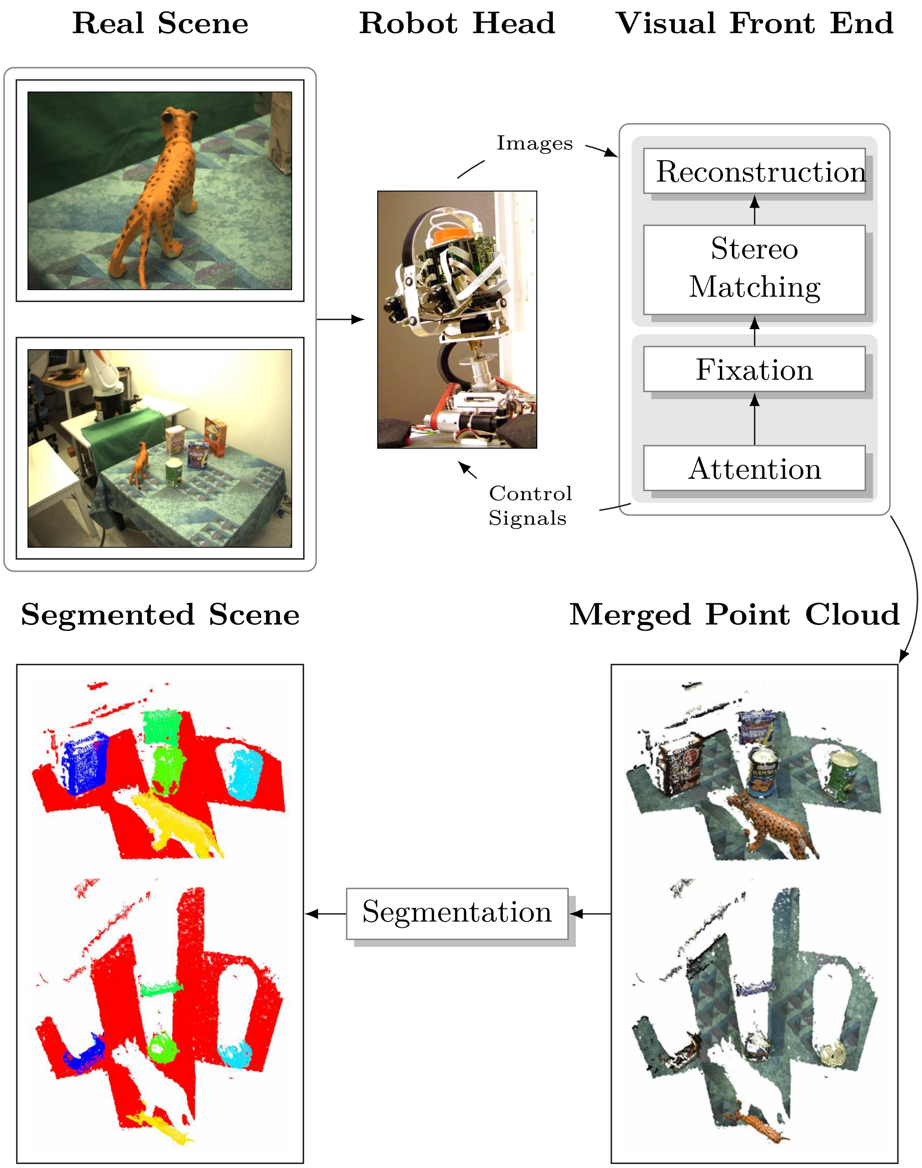


Figure 2.5: Iterative scene modeling through a similar visual front end as shown in Figure 2.2. Segmentation is applied to the point cloud of the whole scene by simultaneously assuming several object hypotheses. Seed points are the fixation points.

bor consistency. Once the pairwise and unary costs are computed, the energy minimization can be performed using standard methods. The α-expansion algorithm with available implementation [42, 119, 40] efficiently computes an approximate solution that approaches the NP-hard global solution.

There are several differences of this approach from the one presented in Section 2.2.1.3. First of all, no flat surface hypothesis is explicitly introduced. This approach is therefore independent of the presence of a table plane. Secondly, object hypotheses are represented by GMMs instead of color histograms and distributions of disparities. And thirdly, several object hypotheses are segmented simultaneously. This has been shown in [5] to be beneficial for segmentation accuracy over just keeping one object hypothesis especially in complex scenes. However, the approach is not real-time. Recently, the previous approach to segmentation has been extended to simultaneously keep several object hypotheses segmented by Bergström et al. [29].

2.3 Discussion

In this chapter, we presented the hardware and software foundations used in this thesis. They enable the robot to visually explore a scene and segment objects from the background as well as from each other. Furthermore, the robot can interact with the scene with its arm and hand.

Given these capabilities, what are the remaining factors that render the problem of robust grasping and manipulation in the scene difficult? First of all, the scene model that is the result of the vision components is only a partial representation of the real scene. As can be observed in Figure 2.5, it has gaps and contains noise. The segmentation of objects might also be erroneous if objects of similar appearance are placed very close to each other. Secondly, given such an uncertain scene model it is not clear how to infer the optimal grasp pose of the arm and hand such that segmented objects can be grasped successfully. And lastly, the execution of a grasp is subject to random and systematic error originating from noise in the motors or an imprecise model of the hardware kinematics and relation between cameras and actuators.

In the following, we will propose different methods to (i) improve scene understanding, (ii) infer grasp poses under significant uncertainty and (iii) demonstrate robust grasping through closed-loop execution.

3

Active Scene Understanding

The term *scene understanding* is very general and means different things dependent on the scale of the scene, the kind of sensory data, the current task of the intelligent agent and the available prior knowledge. In the computer vision community, it refers to the ultimate goal of being able to parse an image (usually monocular), outline the different objects or parts in it and finally label them semantically. A few recent examples for approaches towards this goal are presented by Liu et al. [137], Malisiewicz and Efros [143] and Li et al. [134]. Their common denominator is the reliance on labeled image databases on which for example similarity metrics or category models are learnt.

In the robotics community the goal of scene understanding is similar, however, the prerequisites for achieving it are different. First, a robot is an embodied system that cannot only move in the environment but also interact with it and thereby change it. And second, robotic platforms are usually equipped with a whole sensor suite delivering multi-modal observations of the environment, e.g., laser range data, monocular or stereo images and tactile sensor readings. Depending on the research area, the task of scene understanding changes. In mobile robotics, the scene can be an office floor or even a whole city. The aim is to acquire maps suitable for navigation, obstacle avoidance and planning. This map can be metric as for example in O'Callaghan et al. [162] and Gallup et al. [85]; it can use topological structures for navigation as in Pronobis and Jensfelt [178] and Ekvall et al. [71] or it can contain semantic information as for example in Wojek et al. [232] and Topp [220]. Pronobis and Jensfelt [178], Ekvall et al. [71] and Topp [220] combine these different representations into a layered architecture. The metric map constitutes the most detailed and therefore bottommost layer. The higher up a layer in this architecture, the more abstracted it is from the metric map.

If the goal is mobile manipulation, the requirements for the maps change. More emphasis has to be placed on the objects in the scene that have to be manipulated. These objects can be doors as for example in Rusu et al. [196], Klingbeil et al. [117] and Petersson et al. [170] but also items we commonly find in our offices and homes like cups, books or toys as shown for example in Kragic et al. [122], Ciocarlie et al. [52], Grundmann et al. [95] and Marton et al. [148]. Approaches towards mobile manipulation mostly differ in what kind of prior knowledge is assumed. This ranges from full CAD models of objects to very low level heuristics for grouping 3D points into clusters. A more detailed review on these will be presented in Chapter 7.

Other approaches towards scene understanding exploit the capability of a robot to interact with the scene. Bergström et al. [28] and Katz et al. [110] use motion induced by a robot to segment objects from each other or to learn kinematic model of articulated objects. Luo et al. [140] and Papadourakis and Argyros [164] segment different objects from each other by observing a person interacting with them.

An insight that can be derived from these very different approaches towards scene understanding is that there is yet no representation that has been widely adopted across different areas in robotics. The choice is usually made dependent on task, available sensor data and scale of the scene.

A common situation in robotics is that the scene cannot be captured completely in one view. Instead, several subsequent observations are merged. The task of the robot usually determines what parts of the environment are currently interesting and what parts are irrelevant. Hence, it is desirable for the robot to adopt an exploration strategy that accelerates the understanding of the task-relevant details of the scene as opposed to exhaustively examining its whole surroundings.

The research field of *active perception* as defined by Bajcsy [20] is concerned with planning a sequence of control strategies that minimise a loss function while at the same time maximise the amount of obtained information. Especially in the field of active vision, this paradigm has led to the development of head-eye systems as in Ballard [22] and Pahlavan and Eklundh [163]. They could shift their gaze and fixate on task-relevant details of the scene It was shown that through this approach, visual computation of task-relevant representations could be made significantly more efficient. The active vision system utilized in this thesis is based on the findings in Rasolzadeh et al. [182]. As described in Section 2.2.1, it explores a scene guided by an attention system. The resulting high resolution detail of the scene is then used to grasp objects. Other examples for active vision systems are those concerned with view planning for object recognition like for example Roy [190] or reconstruction like Wenhardt et al. [230]. Aydemir et al. [18] applied active sensing strategies for efficiently searching for objects in large scale environments.

Problem Formulation In the following three chapters of this thesis, we study the problem of scene understanding for grasping and manipulation in the presence of multiple instances of unknown objects. The scenes considered here will be the very common case of table-top environments. We are considering different sources of information that will help us with this task. These can be sensors such as the vision system and tactile arrays described in Section 2.1 or they can be humans that the robot is interacting with through a dialog system.

Requirements The proposed methods have to be able to integrate multi-modal information into a unified scene representation. They have to be robust to noise to estimate an accurate model of the scene suitable for grasping and manipulation.

Assumptions Throughout most of this thesis, we make the common assumption that the dominant plane in the scene can be detected. As already noted by Ballard [22], this kind of context information allows to constrain different optimisation problems or allow for an efficient scene representation. However, we will discuss how the proposed methods are affected when no dominant plane can be detected and how they could be generalized to this case.

We bootstrap the scene understanding process through obtaining an initially segmented scene model from the active vision system. First object hypotheses will be already segmented from the background as visualized in Figure 2.2.

Throughout most of these next three chapters of the thesis, we are assuming that we do not have any prior knowledge on specific object instances or object categories. Instead, we start from assumptions about *objectness*, i.e., general characteristics of what an object constitutes. A commonly used assumption is that objects tend to be homogeneous in their attribute space. The segmentation methods described in Section 2.2.1.3 use color and disparity as these attributes to group specific locations in the visual input to object hypotheses. Other characteristics can be general object shape as for example symmetry.

Approach As emphasized by Bajcsy [20], prediction is essential in active perception. Without being able to predict the outcome of specific actions, neither their cost nor their benefit can be computed. Hence, the information for choosing the best sequence of actions is missing.

To enable prediction, sensors and actuators as well as their interaction has to be modelled with minimum error. Consider for example, the control of the active stereo head in Figure 2.1a. If we had no model of its forward kinematics, we would not be able to predict the direction in which the robot is going to look after sending commands to the motors.

Modeling the noise profiles of the system components also gives us an idea about the uncertainty of the prediction. For example, we are using the haptic sensors of our robotic hand for contact detection. By modeling the noise profile of the sensor pads, we can quantify the uncertainty in each contact detection.

In the following chapters, we will propose different implementations of the function $g(\cdot)$ in Equation 1.1 to make predictions about parts of the scene that have not been observed so far:

$$\hat{\mathbf{x}}_t^+ = g(\hat{\mathbf{x}}_t)$$

in which the current estimate $\hat{\mathbf{x}}_t$ of the state of the scene is predicted to be $\hat{\mathbf{x}}_t^+$. Additionally we are also interested in quantifying how uncertain we are about this prediction, i.e., in the associated variance $\mathbf{\Sigma}_t^+$. We will study different models of the scene state $\mathbf{x}$ but also different kinds of assumptions and *priors* about objectness. These will have different implications on what can be predicted and how accurate. The resulting predictions will help to guide active exploration of the scene, to update the current scene model and to interact with it in a robust manner.

In Chapter 4, we will use the framework of *Gaussian Processes* (GP) to predict the geometric structure of a whole table top scene and guide haptic exploration. The state $\mathbf{x}$ is modelled as a set of 2D locations $\mathbf{x}_i = (u_i, v_i)^T$ on the table. In this chapter, we are studying two different scene representations and their applicability to the task of grasping and manipulation. In an occupancy grid, each location has a value indicating its probability $p(\mathbf{x}_i = occ|\{\mathbf{z}\}_t)$ to be occupied by an object given the set of observations $\{\mathbf{z}\}_t$. In a height map, each location has a value that is equal to the height h_i of the object standing on it. Using the GP framework, we approximate the function $g(\cdot)$ by a distribution $g(\hat{\mathbf{x}}_t) \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ where $\hat{\mathbf{x}}_t^+ = \mu$ and $\mathbf{\Sigma}_t^+ = \mathbf{\Sigma}$. As will be detailed in Chapter 4, this distribution is computed given a set of sample observations and a chosen covariance function. This choice imposes a prior on the kind of structure that can be estimated.

In Chapter 5, we explore how a person interacting through a dialogue system can help a robot in improving its understanding of the scene. We assume that an initial scene segmentation is given by the vision system in Section 2.2. The state of the scene $\mathbf{x}$ is modelled as a set of 3D points $\mathbf{x}_i = (x_i, y_i, z_i)^T$. These are carrying labels l_i which indicate whether they belong to the background ($l_i = 0$) or to one of N object hypotheses ($l_i = j$ with $j \in 1 \dots N$). In this chapter, we are specifically dealing with the case of an under-segmentation of objects that commonly happens when applying a bottom-up segmentation technique to objects of similar appearance standing very close to each other. We utilise the observation that single objects tend to be more uniform in certain attributes than two objects that are grouped together. Given this prior, a set of points $\{\mathbf{x}\}_j$ that are assumed to belong to an object hypothesis j and a feature vector $\mathbf{z}_j$, we can estimate the probability $p(\{\mathbf{x}\}_j = j|\mathbf{z}_j)$ of the set to belong to one object hypothesis j. If this estimate falls below a certain threshold, disambiguation is triggered with the help of a human operator. Based on this information, the function $g(\cdot)$ improves the current segmentation of the scene by re-estimating the labels of each scene point.

In Chapter 6, we focus on single object hypotheses rather than on whole scene structures. We show how we can predict the complete geometry of unknown objects to improve grasp inference. As in the previous chapter, the state of the scene is modelled as a set of 3D points $\mathbf{x}_i = (x_i, y_i, z_i)^T$ and we want to estimate which ones belong to a specific object hypothesis. Different to the previous section, our goal is not to re-estimate the labels of observed points. Instead, the function $g(\cdot)$ has to add points to the scene that are assumed to belong to an object hypothesis. As a prior, we assume that especially man-made objects are commonly symmetric. The problem of modeling the unobserved object part can then be formulated as an optimisation of the parameters of the symmetry plane. The uncertainty about each added point is computed based on visibility constraints.

4

Predicting and Exploring Scenes

The ability to interpret the environment, detect and manipulate objects is at the heart of many autonomous robotic systems as for example in Petersson et al. [170] and Ekvall et al. [71]. These systems need to represent objects for generating task-relevant actions. In this chapter, we present strategies for autonomously exploring a table-top scene. Our robotic setup consists of the vision system presented in Section 2.2 that generates initial object hypotheses using active visual segmentation. Thereby, large parts of the scene are explored in a few glances. However, without significantly changing the viewpoint, areas behind objects are occluded. For finding suitable grasp or for deciding where to place an object once it is picked up, a detailed representation of the scene in the current radius of interaction is essential. To achieve this, parts of the scene that are not visible to the vision system are actively explored by the robot using its hand with tactile sensors. Compared to a gaze shift, moving the arm is expensive in terms of time and gain in information. Therefore, the next best measurement has to be determined to explore the unknown space efficiently.

In this chapter, we study different aspects of this problem. First of all, we need to find an accurate and efficient scene representation that can accomodate multi-modal information. In Section 4.1 and 4.2, we analyse the feasibility of an occupancy grid and a height map for grasping and manipulation.

Second, we need to find efficient exploration strategies that provide maximum information at minimum cost. We compare two approaches from the area of mobile robotics. First, we use *Spanning Tree Covering* (STC) as proposed by Gabriely and Rimon [83]. It is optimal in the sense that every place in the scene is explored just once. Secondly, we extend the approach presented in O'Callaghan et al. [162] where unexplored areas are predicted from sparse sensor measurements by a *Gaussian Process* (GP). Exploration then aims at confirming this prediction and reducing its uncertainty with as few sensing actions as possible.

The resulting scene model is multi-modal in the sense that it i) generates object hypotheses emerging from the integration of several visual cues, and ii) fuses visual and haptic information.

Algorithm 1: Pseudo Code for Scene Exploration

```
Data: Segmented point cloud $\mathcal{S}$ from active segmentation
Result: Fully explored $\mathcal{P}$
begin
    $t = 0,\ j = 0$
    $\mathcal{P} = \texttt{project}(\mathcal{S})$
    while $|\mathcal{P}_u| > 0$ do
        $\hat{\mathcal{P}} = \texttt{predict}(\mathcal{P})$ $\mathbf{p}_j = \texttt{planNextMeasurements}(\mathcal{P}, \hat{\mathcal{P}})$
        repeat
            $t++$
            $z_t = \texttt{observe}(\mathcal{P}, \mathbf{p}_j)$
            $\mathcal{P} = \texttt{update}(\mathcal{P}, z_t)$
        until $z_t \neq occ$
        $j++$
    end
end
```

4.1 Occupancy Grid Maps for Grasping and Manipulation

In this section, we analyse the suitability of a the traditional 2D *occupancy grid* (OG) as a scene representation. It was originally proposed by Elfes [73] and is nowadays widely used for mobile robotics. One of the assets of this representation is that it is well suited for integrating measurements from different sources.

4.1.1 Scene Representation

The grid which is aligned with the table top, uniformly subdivides the scene into N cells $\mathbf{c}_i$ with coordinates (w_i, v_i). Each cell has a specific state $s(\mathbf{c}_i)$. For simplicity, we will refer to it as s_i. It is defined as a binary random variable with two possible values: occupied (occ) or empty (emp). It holds that $p(s_i = occ) + p(s_i = emp) = 1$. We define the set $\mathcal{P} = \{\mathbf{c}_i \mid 0 < i < N\}$, as the whole grid. Our goal is to estimate $p(s_i = occ \mid \{z\}_t)$, the probability for each cell $\mathbf{c}_i$ to be occupied given a set of sensor measurements $\{z\}_t$ up to point t in time. Let $\mathcal{P} = \mathcal{P}_k \cup\ \mathcal{P}_u$ where $\mathcal{P}_k$ is the set of cells whose state has already been estimated based on observations. $\mathcal{P}_u$ is the set of cells that has not been observed yet. Each cell is initialized with a prior probability $p(s_i = occ) = 0.5$. Our approach for scene exploration is summarized in Algorithm 1.

Initially, we project the stereo reconstructed point cloud $\mathcal{S}$ of the scene on the grid as follows. Disparity maps are gathered from several views of the robot head on the scene. They are converted into 3D points and projected into a common reference frame for all observations. Once aggregated, the whole point cloud is cleaned to remove outliers. The labeling from the 3D object segmentation (discussed in Section 2.2.1) is applied to the remaining points identifying objects from the background. These object points are placed into a voxel grid. This voxelized representation is projected down into a 2D occupancy grid $\mathcal{P}$ for planning. Figure 4.1 displays this process and the resulting 2D map.

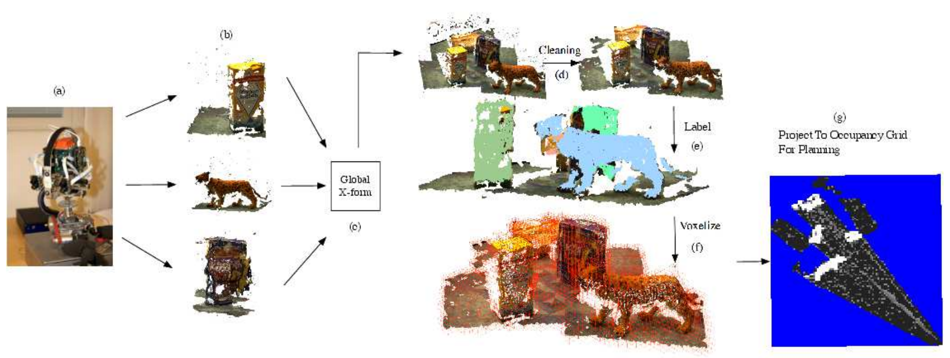


Figure 4.1: Generation of an occupancy grid from individual views. (a) ARMAR robot head that (b) gathers several views. (c) Views are projected into a common reference frame and (d) cleaned to remove noise. (e) Points are labeled according to the 3D object segmentation (Section 2.2). (f) Scene is voxelized. The voxels that belong to objects are projected down into the map (g). Blue labels are unseen cells and gray levels correspond to occupancy probability.

The initial $\mathcal{P}$ contains a large amount of unknown space that needs to be explored with the hand. The single steps in the main loop of Algorithm 1 will be explained in the following sections.

4.1.2 Scene Observation

For visual exploration of the scene, we use the active vision system as presented in Section 2.2.1. It produces an initial scene model segmented in background and several object hypotheses. An example for such a model can be seen in Figure 4.1.

For haptic exploration we use the *proximal* and the *distal* sensor matrices $\mathbf{H}_p$ and $\mathbf{H}_d$ of one of the fingers of the Schunk hand. Our goal is to distinguish between the cases when this finger is in contact with an object and when it is not. For this purpose, we compute a noise profile of the sensor matrices. Let $\mathbf{h}_p$ be the vector of measurements formed through flattening $\mathbf{H}_p$ and similarly, $\mathbf{h}_d$ be the measurement vector corresponding to $\mathbf{H}_d$. We model the distribution of these random variables as multivariate normal distributions $\mathbf{h}_p \sim \mathcal{N}(\mu_p, \mathbf{\Sigma}_p)$ and $\mathbf{h}_d \sim \mathcal{N}(\mu_d, \mathbf{\Sigma}_d)$, respectively. The means and covariance matrices are computed from a number of measurements recorded while the two sensor matrices were in a known non-contact situation. A contact with an object can then be seen as a multivariate *outlier*. For outlier detection, we compute the Mahalanobis distance between the current measurement $\mathbf{h}_{t|p,d}$ and the respective mean $\mu_{p,d}$.

$$d(\mathbf{h}_t) = \sqrt{(\mathbf{h}_t - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{h}_t - \mu)} \tag{4.1}$$

Note that the subscripts p and d are skipped in this equation for simplicity. If $d(\mathbf{h}_t)$ is greater than a threshold ϕ, then we know that this haptic sensor matrix is in contact and $z_t = \mathit{contact}$. Otherwise, $z_t = \neg \mathit{contact}$.

4.1.3 Map Update

For each movement of the haptic sensors along the planned path, we are receiving a measurement z_{t+1}. Based on this and the current estimate $p(s_i = \mathit{occ} \mid \{z\}_t)$ of the state of each cell s_i in the occupancy grid, we want to estimate

$$p(s_i = \mathit{occ} \mid \{z\}_{t+1}) = \frac{p(z_{t+1} \mid s_i = \mathit{occ})\; p(s_i = \mathit{occ} \mid \{z\}_t)}{\sum_{s_i} p(z_{t+1} \mid s_i) p(s_i \mid \{z\}_t)} \tag{4.2}$$

In this recursive formulation, the resulting new estimate $p(s_i = \mathit{occ} \mid \{z\}_{t+1})$ is stored in the occupancy grid. $p(z_{t+1} \mid s_i)$ constitutes the haptic sensor model. To compute $p(z_{t+1} \mid s_i = \mathit{emp})$, the current measurement $\mathbf{h}_{t+1|p,d}$ and Equation 4.1 is used as follows

$$\begin{aligned} p(z_{t+1} \mid s_i = \mathit{emp}) &= \exp(-\frac{1}{2}\; d(\mathbf{h}_{t+1})) & (4.3)\\ &= \exp(-\frac{1}{2}\sqrt{(\mathbf{h}_{t+1} - \mu)^T \Sigma^{-1} (\mathbf{h}_{t+1} - \mu)}) & (4.4)\end{aligned}$$

For the case $s_i = \mathit{occ}$, we empirically determined the following discrete probability values

$$\begin{aligned} p(z_{t+1} = \mathit{contact} \mid s_i = \mathit{occ}) &= 0.9 & (4.5)\\ p(z_{t+1} = \neg\mathit{contact} \mid s_i = \mathit{occ}) &= 1 - p(z_{t+1} = \mathit{contact} \mid s_i = \mathit{occ}). & (4.6)\end{aligned}$$

An alternative to this way of modeling the sensor response (when in contact with an object) is to collect a set of measurements. From this set a distribution could be estimated similar to what is described in Section 4.1.2. However, given the many different shapes of objects and the resulting variety of haptic sensor patterns, modeling the opposite case of a non-contact is the simpler but still effective solution.

4.1.4 Map Prediction

In the traditional occupancy grid, cells that have not been observed yet will have a probability $p(s_i = \mathit{occ} \mid \{z\}_t) = 0.5$, i.e. there is no information available about the state of these cells. However, we know that cells in the grid that are close to occupied spaces but are due to occlusions not directly observable, are likely to be part of the occluding object. By modeling this spatial correlation, we can predict unobserved places from observed ones. Instead of exploring the whole environment exhaustively, we want to confirm the predicted map at specifically uncertain places. Recently, O'Callaghan et al. [162] proposed that the spatial correlation in a 2D occupancy grid can be modeled with a Gaussian Process. The assumption of independence of neighboring cells in traditional occupancy grids is removed.

A GP is used to fit a likelihood function to training data. In our case this is the set of cells $\mathbf{c}_r \in \mathcal{P}_k$ and their estimated state s_r of occupancy. Given the estimated continuous function over the occupancy grid, we can then estimate the state s_j of the cells $\mathbf{c}_j \in \mathcal{P}_u$ that have not been observed yet. We will briefly introduce GPs first for the simpler case of regression and then for the problem of classifying a cell to be either occupied or empty which is the core of this section. For a more detailed explanation, we refer to Rasmussen and Williams [181].

4.1.4.1 Gaussian Process Regression

A GP is defined as a collection of a finite number of random variables with a joint Gaussian distribution. A GP can be seen as a distribution over functions with a mean μ and covariance Σ. Given a set of input values $\mathbf{x}$ and corresponding target values $\mathbf{y}$, the goal in GP regression is to estimate the underlying latent function $g(\cdot)$ that generated these samples. We will also refer to the set $\{\mathbf{x}_i, y_i\}_M$ as the *training set* with cardinality M. Expressed as a linear regression model with Gaussian noise ϵ, the relation between input and target values can be modelled as follows:

$$g(\mathbf{x}_i) = \phi(\mathbf{x}_i)^T\mathbf{w}, \quad y_i = g(\mathbf{x}_i) + \epsilon \tag{4.7}$$

with the noise

$$\epsilon \sim \mathcal{N}(0, \sigma_M^2) \tag{4.8}$$

following an independently and identically distributed Gaussian with zero mean and variance σ_M^2. $\mathbf{w}$ is a vector of weight parameters of the linear model. $\phi(\mathbf{x}_i)$ is a function that maps D-dimensional input data into an N-dimensional feature space. Under this formulation, the likelihood of the training data given the model is computed as a factored distribution over all training samples:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \mathbf{w}) &= \prod_{i=1}^{M} p(y_i|\mathbf{x}_i, \mathbf{w}) && (4.9)\\ &= \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_M} \exp(-\frac{(y_i - \phi(\mathbf{x_i})^T\mathbf{w})}{2\sigma_M^2}) && (4.10)\\ &= N(\phi(\mathbf{x})^T\mathbf{w}, \sigma_M^2\mathbf{I}). && (4.11) \end{aligned}$$

If we use a zero-mean Gaussian prior on the weights $\mathbf{w} \sim \mathcal{N}(0, \Sigma_{\mathbf{w}})$, then the posterior likelihood $p(\mathbf{w}|\mathbf{x}, \mathbf{y})$ is proportional to the product of the likelihood and the prior

$$p(\mathbf{w}|\mathbf{x}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \mathbf{w})p(\mathbf{w}) \tag{4.12}$$

which is again Gaussian.

We are primarily interested in predicting the target value y' for new input values $\mathbf{x}'$. The predictive distribution $p(g(\mathbf{x}')|\mathbf{x}', \mathbf{x}, \mathbf{y})$ is computed by averaging the outputs of all possible linear models weighted with the posterior distribution of the model given the training set:

$$p(g(\mathbf{x}')|\mathbf{x}',\mathbf{x},\mathbf{y}) = \int p(g(\mathbf{x}')|\mathbf{x}',\mathbf{w})p(\mathbf{w}|\mathbf{x},\mathbf{y})d\mathbf{w}. \tag{4.13}$$

Since all the factors over which we want to integrate follow a Gaussian distribution, we can evaluate the integral in Equation 4.13 analytically. The predictive distribution is then Gaussian again and defined as $g(\mathbf{x}') \sim \mathcal{N}(\mu, \Sigma)$ where

$$\mu = k(\mathbf{x}',\mathbf{x})^T[\mathbf{K}(\mathbf{x},\mathbf{x}) + \sigma_M^2 I]^{-1}\mathbf{y} \tag{4.14}$$

$$\Sigma = k(\mathbf{x}',\mathbf{x}') - k(\mathbf{x}',\mathbf{x})^T[\mathbf{K}(\mathbf{x},\mathbf{x}) + \sigma_M^2 I]^{-1}k(\mathbf{x}',\mathbf{x}). \tag{4.15}$$

The entries of the Gram matrix $\mathbf{K}(\mathbf{x},\mathbf{x})_{a,b}$ at row a and column b are defined based on a covariance function $k(\mathbf{x}_a,\mathbf{x}_b)$ with some hyperparameters θ and with $a, b \in \{1 \dots M\}$. This covariance function is related to the function $\phi(\cdot)$ from Equation 4.7 as follows:

$$k(\mathbf{x}_a,\mathbf{x}_b) = \phi(\mathbf{x}_a)^T\Sigma_{\mathbf{w}}\ \phi(\mathbf{x}_b). \tag{4.16}$$

The most widely used covariance function is the squared exponential (SE)

$$k(\mathbf{x}_a,\mathbf{x}_b) = \sigma_l^2 \exp(-((\mathbf{x}_a - \mathbf{x}_b)^T L^{-1}(\mathbf{x}_a - \mathbf{x}_b))/2) \tag{4.17}$$

where the hyperparameters are σ_l, the signal variance, and L, the identity matrix multiplied with the length scale l. There is a large set of other covariance functions proposed in the literature. More details about those applied in this thesis can be found in Appendix A. As will be shown later in Section 4.1.6, the simple SE kernel achieves the best results for scene prediction during exploration.

The kind of covariance function as well as its hyperparameters have to be chosen such that $g(\cdot)$ optimally models the underlying process. As suggested in [181], this *model selection* problem can be treated in a Bayesian way. In this thesis, we want to choose the hyperparameters of a given covariance function such that the *marginal likelihood* $p(\mathbf{y}|\mathbf{x},\theta)$ of the data given the model is maximised. This marginal likelihood is defined as follows:

$$p(\mathbf{y}|\mathbf{x},\theta) = \int p(\mathbf{y}|\mathbf{x},\mathbf{f},\theta)p(\mathbf{f}|\mathbf{x})d\mathbf{f} \tag{4.18}$$

where $\mathbf{f} = g(\mathbf{x})$. A desirable property of $p(\mathbf{y}|\mathbf{x},\theta)$ is that it automatically trades off model fit against model complexity. This can be observed when writing out the logarithm of the marginal likelihood. The integral can be computed analytically since both, the prior and the likelihood in Equation 4.18 are Gaussian:

$$\log p(\mathbf{y}|\mathbf{x},\theta) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{M}{2}\log 2\pi \tag{4.19}$$

where $\mathbf{K}$ is short hand for $K(\mathbf{x}, \mathbf{x}) + \sigma_M^2 I$. The first part of the sum in Equation 4.19 reflects how well the data fits the model. The second part penalizes model complexity and is only dependent on the covariance function and its input. The last part is a normalisation term. By equating the partial derivatives of the log marginal likelihood with respect to the hyperparameters to zero, we can solve for the optimal hyperparameters of a given covariance function and training set.

4.1.4.2 Gaussian Process Classification

Predicting the binary state of a cell in an occupancy grid is a classification rather than a regression problem. Our training set consists of the coordinates $(w, v)_r$ of all the cells $\mathbf{c}_r \in \mathcal{P}_k$ as $\mathbf{x}$ and their observed state s_r as $\mathbf{y}$. This state is binary and can be either occupied or empty. Our goal is to compute $p(s_j = occ \mid \{z\}_t)$, the probability that the state s_j of a cell $\mathbf{c}_j \in \mathcal{P}_u$ is occupied. Note, that this cell has not been observed yet.

The basic idea behind Gaussian Process classification is very simple. Let $\mathbf{x}' = \mathbf{c}_j$ and $y' = s_j$, then we can compute $p(y' = occ \mid \{z\}_t)$ by *squashing* the predictive distribution $g(\mathbf{x}')$ through the cumulative Gaussian function:

$$p(y' = occ \mid \{z\}_t) = 1/2 \cdot (1 + \operatorname{erf}(g(\mathbf{x}')/\sqrt{2})). \tag{4.20}$$

However, the distribution $g(\mathbf{x}')$ cannot be computed analytically as in the case of regression where the posterior $p(\mathbf{w}|\mathbf{x}, \mathbf{y})$ is Gaussian. For classification where the target values are either $+1$ (occupied) or -1 (empty), the likelihood $p(y_i|\mathbf{x}_i, \mathbf{w})$ of a data point given the model is a sigmoid function

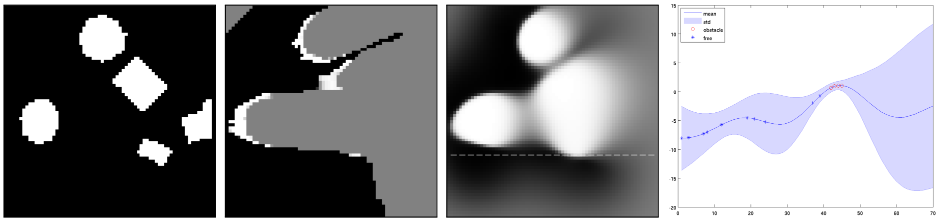
$$p(y_i|\mathbf{x}_i, \mathbf{w}) = \sigma(y_i \mathbf{x}_i^T \mathbf{w}). \tag{4.21}$$

The likelihood $p(\mathbf{y}|\mathbf{x}, \mathbf{w})$ of the whole training data will therefore be non-Gaussian. Even if we use the same prior on the weights $\mathbf{w}$, the posterior as in Equation 4.12 will be non-Gaussian. Therefore, calculating the predictive distribution $g(\mathbf{x}')$ as in Equation 4.13, requires evaluating an integral over a non-Gaussian posterior. Different approximations for this problem have been proposed. We follow the Laplace approach in which the posterior $p(\mathbf{w}|\mathbf{y}, \mathbf{x})$ is approximated as a Gaussian.

An example for this prediction given a 2D map of partially explored scene is given in Figure 4.2. In Section 4.1.6, we will show quantitatively on synthetic data that a GP predicted map is a reasonable estimate of the ground truth.

4.1.5 Action Selection for Exploration

Given a partial map of the environment, we want to efficiently explore the remaining unknown parts with haptic sensors, i.e., we want to minimize the amount of measurement actions needed to reach a sufficient scene understanding. We will present two algorithms for planning a measurement path in the given map. First, we will use Spanning Tree Covering [83]. Second, we propose an active learning scheme based on the predicted map.



(a) Ground truth 2D occupancy grid

(b) Occupancy grid after camera Measurements taken from the left side.

(c) Probability of occupancy for each cell predicted by GP. 50th row labeled.

(d) Predictive mean and variance for 50th row of occupancy grid along with training points.

Figure 4.2: Example for the prediction of a 2D map from camera measurements using GPs.

4.1.5.1 Spanning Tree Covering

STC tackles the *covering problem* that can be formulated as follows. Given the haptic sensor of size d and a planar work-area $\mathcal{P}_u$, the sensor has to be moved along a path such that every point in $\mathcal{P}_u$ is covered by it only once.

STC first defines a graph $G(\mathcal{V}, \mathcal{E})$ on $\mathcal{P}_u$ with cells of size $2d$, the double tool size. In our case where $G(\mathcal{V}, \mathcal{E})$ has uniform edge weights, Prim's algorithm [175] can be used to construct a *Minimum Spanning Tree* (MST) that covers every vertex $\mathcal{V}$ in G at minimum cost regarding the edges $\mathcal{E}$. The haptic measurement path is defined on the original grid with cells of size d such that the MST is circumnavigated in counterclockwise direction. This circular path starts and ends at the current arm position. In case an obstacle is detected along the path, a new spanning tree has to be computed based on the updated grid. An example for such a path is shown in Figure 4.3a and 4.3b

4.1.5.2 Active Learning

Our goal is to estimate the scene structure early in the whole exploration process without exhaustive observation. Thus, we want to support the map prediction by selecting most informative observations. Let us consider a set of measurements made along an MST as described above. These measurements will tend to be very close to each other without leaving any unobserved holes in the map. The GP prediction of the map based on these measurements will not be significantly different from the prediction based on only half of it. By using a GP and thereby exploiting spatial correlation in a map, the probability for a cell to be occupied can be inferred from its neighbors without explicitly observing it.

We will present exploration strategies that follow an active-learning paradigm of selecting new measurement points that maximize the expected information gain. As

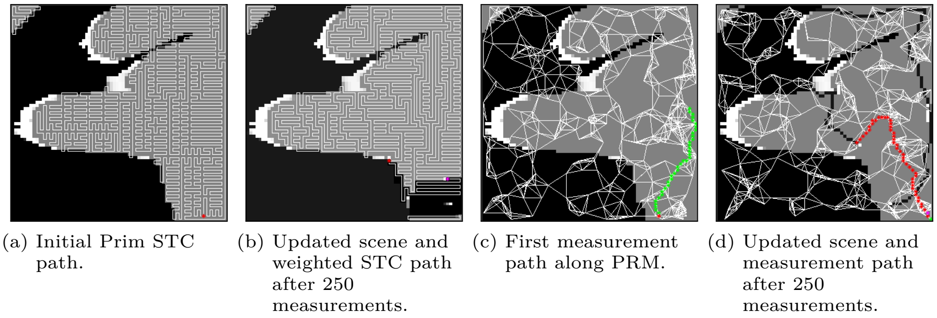

(a) Initial Prim STC path.
(b) Updated scene and weighted STC path after 250 measurements.
(c) First measurement path along PRM.
(d) Updated scene and measurement path after 250 measurements.

Figure 4.3: Examples for potential measurement paths generated with different exploration strategies. Red stars label current and previous traversed arm positions, respectively.

it has been shown by Chaloner and Verdinelli [49], this is equivalent to minimizing the predictive variance Σ from Equation 4.15.

$$\mathbf{c}^* = \arg\max_{P_u} U_1(\mathbf{c}_i) \tag{4.22}$$

$$\text{with } U_1(\mathbf{c}_i) = \Sigma_i \tag{4.23}$$

It can occur that the hand is detecting an obstacle along the chosen measurement path. It then has to re-plan and would potentially never reach the initially selected optimal observation point. Instead of considering the predictive variance only, the expected gain in information of the whole measurement path has to be taken into consideration. Different to U_1, the following utility functions take this path into account additionally to the predictive variance. The first one is

$$U_2(\mathbf{c}_i) = \alpha\Sigma_i - (1-\alpha)\ d(\mathbf{c}_s, \mathbf{c}_i) \tag{4.24}$$

where Σ_i is the predictive variance of the cell $\mathbf{c}_i$ and $d(\mathbf{c}_s, \mathbf{c}_i)$ is any distance function of the current position of the arm $\mathbf{c}_s$ and cell $\mathbf{c}_i$. The parameter $0 < \alpha < 1$ is user determined. The closer it is to 1 the more important becomes the value of the predictive variance.

The second utility function uses a discount factor δ.

$$U_3(\mathbf{c}_i) = \sum_{r=1}^{R} \delta^r \Sigma_{p(r)} \tag{4.25}$$

where R is the number of measurement that are needed to reach the final cell $\mathbf{c}_i$ along the path $\mathbf{p} = [\mathbf{c}_{s+1} \dots \mathbf{c}_i]$. Not just the final cell $\mathbf{c}_i$ is considered. Instead, the predictive variance of all the cells along the path contributes to the utility value of

	STC	PRM
OG	Occupancy grid explored through Spanning Tree Covering.	Occypancy grid explored through Probabilistic Roadmap approach.
GP	Gaussian Process-based map explored through Spanning Tree Covering	Gaussian Process-based map explored through Probabilistic Roadmap approach

Table 4.1: Summary of scene models and exploration strategies evaluated in this section.

$\mathbf{c}_i$. The parameter δ has to be chosen by the user. It steers how steep the decrease of influence of the cells in the path are dependent on their distance from the current arm position.

To find the global maxima of Equation 4.22 we have to maximize over all cells $\mathbf{c}_i$ in $\mathcal{P}_u$ and over all the paths through which a cell can be reached from $\mathbf{c}_s$. Since this is a prohibitive number of possibilities to compute, we use sampling techniques to find a local maxima of the utility functions. We are building a *probabilistic road map* (PRM) in the two dimensional C-space of the occupancy grid [111].

A set $\mathcal{T}$ of cells from $\mathcal{P}$ is sampled according to their predictive variance and connected to the PRM. The *Dijkstra* algorithm is used to compute the shortest path from the current arm position to each cell in $\mathcal{T}$ through the PRM. The result is used to compute the utility of each cell in $\mathcal{T}$. An example for the PRM and therefore the possible paths to traverse is shown in Figure 4.3c and 4.3d.

4.1.6 Experiments

In this section, we evaluate the proposed scene models and exploration strategies. We are specifically interested in how accurate the scene estimates are when using an occupancy grid map or a Gaussian Process based prediction. Furthermore, we want to compare the different exploration strategies in terms of how fast an accurate scene estimate is achieved. The different combinations of scene representations and exploration methods are summarized in Table 4.1.

4.1.6.1 Synthetic Data Set

Data Set and Measure of Comparison We generated 100 different 2D occupancy grids (70×70 cells) of which an example appears in Figure 4.2a. Every scene contains ten objects that can either be of circular, elliptical or rectangular shape with a random size, aspect ratio, orientation and position. Overlapping is allowed so that fewer than ten connected components can occur as well as more complex contours.

For every scene, we simulated three camera observations made from a fixed position on their left side with a random direction. As a sensor model, we used a beam model with a Gaussian profile [73]. Given a measurement, the occupancy grid gets updated according to Equation 4.2. An example for the result of this simulation is shown in Figure 4.2b.

We posed the validation problem of occupancy grid estimation as a binary classification into empty or occupied cells. For each estimated grid at any time in the exploration process, the number of false and true positives can be computed for different thresholds resulting in an ROC curve. For evaluating the development of this curve over time, we choose the area under the ROC curve (AUC) as a measure. It corresponds to the probability that the state of a cell is correctly classified.

Covariance Functions Compared O'Callaghan et al. [162] claim that the neural network covariance function is especially suitable for predicting the non-stationary behavior of a typical map data set. That data is recorded from indoor and outdoor environments either with hallways, rooms, walls or streets bounded by buildings. However, evaluations on our experimental data showed superior performance when using the squared exponential covariance function. It poses a better prior for scenes in which blob-like objects are spread on a table.

We predicted each of the 100 occupancy grids with a GP by sampling different numbers of training points from the space observed by the camera and then querying $p(s_i = occ \mid \{z\}_t)$ for $\mathbf{c}_i \in \mathcal{P}_u$. We compared seven different covariance functions with each other. Among these are the already discussed neural network covariance function (NN), the squared exponential function with and without automatic relevance detection (SE or SEARD), the Mátern covariance function with different parameters ν (Mat) and the rational quadratic (RQ). A more detailed explanation of these kernels can be found in Appendix A. While the SE covariance function is determined by only one length scale, the others expose more flexibility. For example SEARD can be parameterized in terms of its hyperparameters and adapt the length scale of each input dimension according to its detected relevancy. An RQ kernel can be seen as a scale mixture of SE kernels of different length scales. And a Mat kernel is characterized by varying degrees of smoothness.

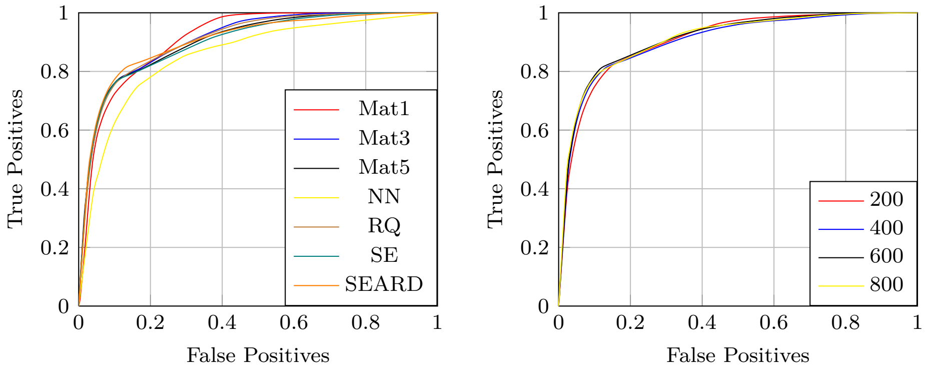
Table 4.4 lists the classification accuracy given different kernels and number of samples from the free space. Since the occupied space is significantly smaller than the free space, we keep the number of samples from there fixed to a maximum of 200 points. Points are sampled according to their confidence.

SEARD always outperforms the other kernels. For visualisation, the corresponding ROC curves of all kernels for 400 samples are shown in Figure 4.5a. They show that the neural network kernel is clearly outperformed by all the other kernels. The difference between the remaining kernels is not large. Furthermore, Figure 4.5b shows that the number of samples from the free space does not have a significant influence on the classification performance.

Prediction vs Occupancy Grid An important question is whether the GP makes a valid prediction of the scene map and how this compares to the traditional occupancy grid. In an OG, only those estimates for $p(s_i = occ \mid \{z\}_t)$ are different

Samples	Mat1	Mat3	Mat5	NN	RQ	SEARD	SE
200	0.9119	0.9167	0.9176	0.9041	0.9158	0.9191	0.9192
400	0.9152	0.9207	0.9220	0.9035	0.9198	0.9226	0.9218
600	0.9156	0.9207	0.9223	0.9023	0.9171	0.9242	0.9234
800	0.9162	0.9230	0.9243	0.9082	0.9206	0.9265	0.9256

Figure 4.4: Evaluation of classification accuracy for different kernels and number of samples. Mat: Mátern Class. NN: Neural Network. RQ: Rational Quadratic. SEARD: Squared Exponential with Automatic Relevance Detection. SE: Squared Exponential.



(a) ROC Curves for different covariance functions and sample size 400.

(b) ROC Curves for SEARD covariance function and different sample sizes.

Figure 4.5: Visualisation of classification accuracy through ROC curves for different covariance functions and sample sizes. While the choice of covariance function can make a significant difference in classification accuracy, the number of samples does not.

from the initial value of 0.5 for which at least one measurement has been obtained. However, the GP predicts occupancy probability for all cells. To confirm that this inference is valid, we calculated the mean AUC for all occupancy grids after they have been observed by the camera and compared it to the mean AUC of the GP predicted maps. While for the unpredicted occupancy grid this value is 0.765205, it is 0.90705 for the predicted maps; a clear increase of 18%. Therefore we conclude in agreement with O'Callaghan et al. [162] that a GP prediction provides a valid inference about regions of the scene in which no measurements are available.

Variance	Trade-Off	Discount
Scene explored through PRM approach with utility function based on variance of goal point (Equation 4.22).	Scene explored through PRM approach with utility function based on variance of goal point traded-off with distance to travel there (Equation 4.24).	Scene explored with a PRM using a utility function that includes variance of all positions along the path to the goal (Equation 4.25)

Table 4.2: Summary of the three PRM-based exploration strategies differing in utility function. These are evaluated on the scene estimate modelled as OG or GP.

Exploration Strategies Compared We compare the different exploration strategies based on the mean and variance of the AUC measure over time and all synthetic scenes. We start from the scenes partially explored with the camera. As a kernel, we use SEARD.

Utility Functions Compared Three functions were proposed in Section 4.1.5.2 that incorporate uncertainty in the map prediction and/or distance to traverse. They are summarized in Table 4.2. Figure 4.6 show the results for the OGs and for the GP predicted maps. There is a clear difference between the utility functions. The discounted version (Equation 4.25) performs best both in terms of mean and variance in the OGs and GP predicted maps. This is due to the explicit consideration of the whole exploration path instead of just a high predictive variance goal point that might never be reached.

However when compared to our previous results in [3] that were derived on a smaller dataset, the difference between the utility functions especially for the GP maps is not as pronounced. In [3], we used the simple squared exponential covariance function that has a fixed length scale. Figure 4.7 shows the development of the map accuracy when using this kernel instead of SEARD. They confirm the results in [3] by also showing a more significant difference between the three utility functions.

The reason for the superior performance of the simple kernel over SEARD lies in how we learn the hyperparameters. Only once in the beginning they are learnt through maximizing the log marginal likelihood given the GP prior and the training data as described in Section 4.1.4.1. During the exploration process, only the Gram matrix **K** is updated with new data. Since the SEARD kernel is more flexible, the point estimate of its hyperparameters leads to an overfitting to the initial map data. This effect is less pronounced when using the simpler SE kernel.

A solution to this problem would be to iteratively learn also the hyperparameters in an online GP approach or to assume an additional prior over the hyperparameters. Instead of computing a maximum likelihood point estimate through Equation 4.19, one could then compute the maximum a posteriori estimate, i.e., a distribution over hyperparameters.

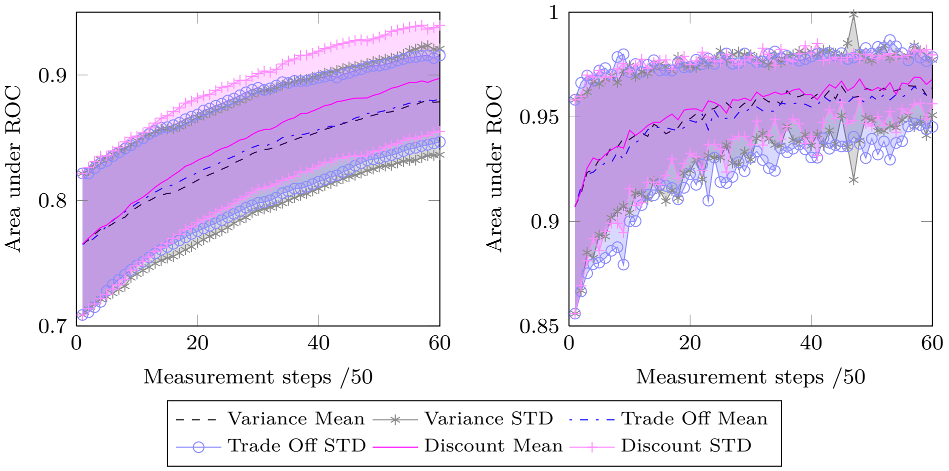


Figure 4.6: Mean and standard deviation of area under ROC curve (AUC) using different utility-based exploration strategies and SEARD kernel. Left) Classification performance averaged over 100 occupancy grids (OG) and shown over time. Right) Classification performance averaged over 100 Gaussian Process (GP) based predicted grids. Discounted predictive variance outperforms pure variance and trading off as utility values both in the OG and GP case.

STC versus PRM We also compared STC based exploration with the one based on PRM. Measurement paths are planned such that each $\mathbf{c}_i \in \mathcal{P}_u$ is traversed only once. Figure 4.8 directly compares the results for the occupancy grids and for the GP predicted maps. The estimation of the occupancy grid converges relatively fast towards ground truth when traversed using STC.

Furthermore, Figure 4.8 shows a comparison of the development of the predictive variance during haptic exploration. Intuitively, this is a measure of uncertainty over the map estimate. We can see that on average the PRM-based approach decreases uncertainty about the scene structure faster.

Summary Figure 4.8 contains the results for the discounted utility function for direct comparison with the STC-based exploration. GP predicted maps traversed based on the discounted utility are more accurate early in the exploration process. This is an expected result since here points of high variance are chosen to be measured first. They will therefore have a high positive influence on the quality of the prediction. However, the occupancy grid does not converge as fast towards ground truth as in the STC-based exploration. This is because the PRM-based exploration might traverse a number of cells more than once.

We conclude that if a good map estimate is needed quickly, active learning based

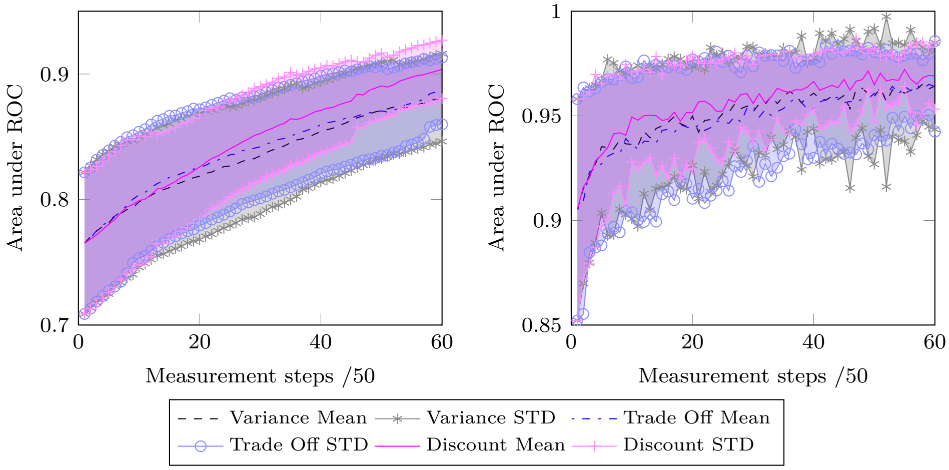


Figure 4.7: Mean and standard deviation of area under ROC curve (AUC) using different utility-based exploration strategies and SE kernel. Left) Classification performance averaged over 100 occupancy grids (OG) and shown over time. Right) Classification performance averaged over 100 Gaussian Process (GP) based predicted grids. Discounted predictive variance outperforms pure variance and trading off as utility values both in the OG and GP case.

exploration is advantageous over systematically traversing the space. If there is time for an exhaustive exploration, STC-based measurement paths are more beneficial.

4.1.6.2 Demonstration in the Real World

In this section, we demonstrate our approach for the scenario of table-top exploration populated with several unknown objects using both, point clouds reconstructed from a stereo camera and haptic data from a robotic hand. As exploration strategy we consider STC and a PRM based scheme with the discounted utility function defined in Equation 4.25.

The real world scene contains three objects. For visual exploration, we manually select the initial fixation point for two of them. This could be replaced by using an attention system. The objects are segmented and stereo reconstructed as described in Section 2.2.1. The resulting point cloud is projected onto an occupancy grid aligned with the table. By excluding the third object from visual observation, we can demonstrate the map update upon finger contact. Examples for the occupancy grid are given in Figure 4.9b and 4.10b.

In this stereo based grid, we can now detect locations that were not observed with the vision system. The reachable unexplored spaces are explored with the haptic sensors on the hand. For doing so, we are using one of the three fingers

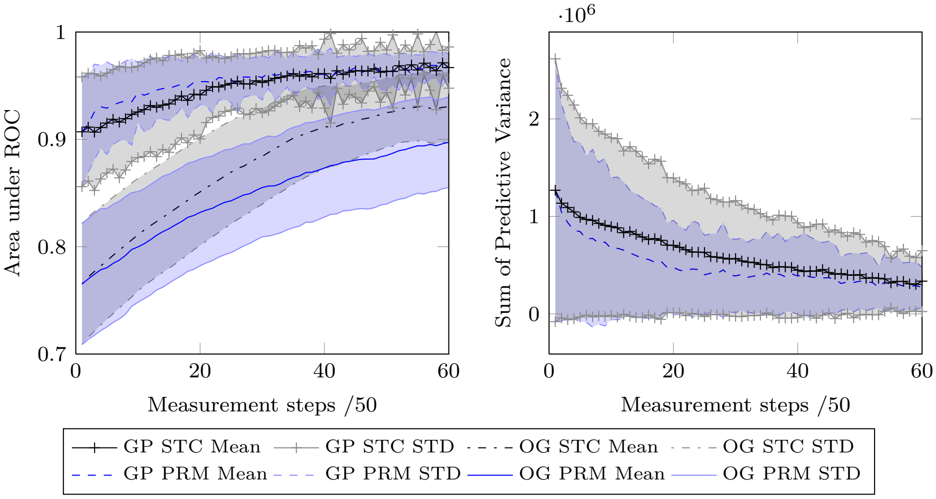
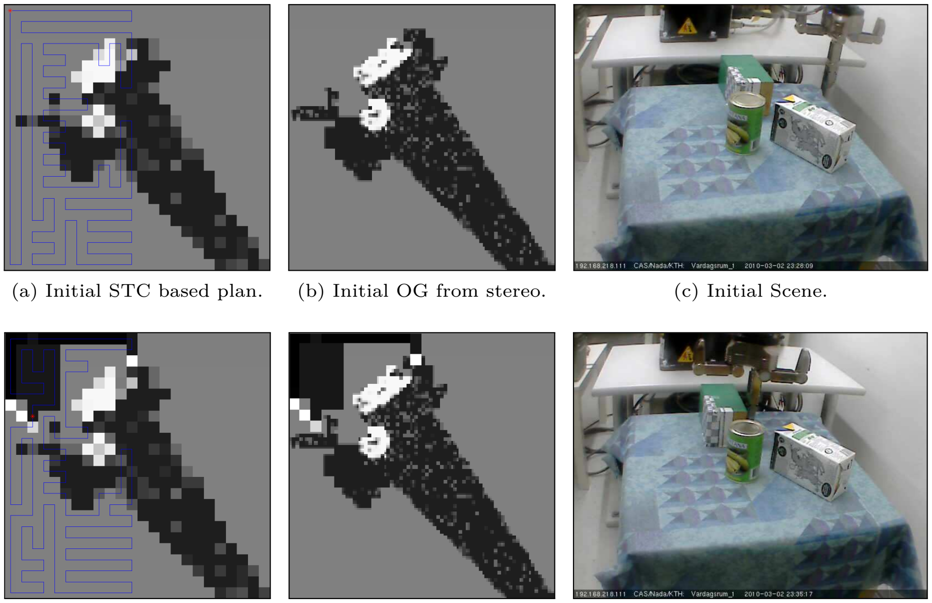


Figure 4.8: Comparison between Spanning Tree Covering (STC) based exploration method and discounted utility based method. Left) Mean and variance of area under ROC curve (AUC) for occupancy grids (OG) as well as Gaussian Process (GP) based prediction. Discounted utility based exploration achieves a more accurate GP prediction early in the process and keeps this accuracy over time as visible in the low standard deviation. STC explores the unknown space in the OG faster. Right) Mean and variance of sum of predictive variance. Discounted utility-based exploration decreases the overall uncertainty about the scene structure faster.

pointing downwards as shown in Figure 4.9c. The hand is moved at a constant height over the table. The haptic sensor arrays on the finger are always pointing in the direction of movement. Planning is done in a slice of the robot task space that is aligned with the table top. This is a reasonable simplification since all the points on the table within a radius of 780mm are reachable with a valid joint configuration. For more complex environments planning has to be done in the six-dimensional C-space of the arm. To this end, other stochastic planning methods besides PRM have been proposed as for example *Rapidly-exploring Random Trees* (RRTs) [125]. To adapt the proposed exploration strategies accordingly is considered as future work.

The fingers of the Schunk Hand are approximately 30×30mm thick. Therefore, one cell in the planning grid has to have the same measures to allow for a rotation around the axis of the finger without entering a neighboring cell. Compared to the stereo based grid with a resolution of approximately 5×5mm this is quite coarse. However, this also reduces the planning space. For every measurement, both grids are updated in parallel: one cell in the planning grid, a set of 6×6 cells in the stereo grid.


192.168.218.111 CAS/Nada/KTH: Vardagsrum_1 2010-03-02 23:28:09
192.168.218.111 CAS/Nada/KTH: Vardagsrum_1 2010-03-02 23:35:17

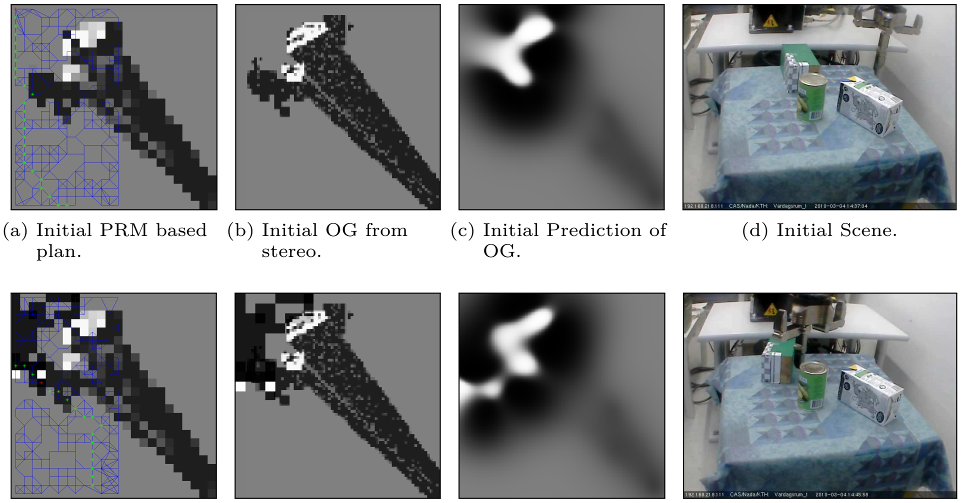
(a) Initial STC based plan. (b) Initial OG from stereo. (c) Initial Scene.

(d) STC based plan after 73 steps. (e) Stereo OG after 73 steps. (f) Scene after 73 steps.

Figure 4.9: Snapshots from an exploration using an STC based plan (covers only reachable workspace). Left Column) Coarse planning grid for the finger. Middle Column) Fine grid for stereo data.

In Figure 4.9, the STC based measurement path planned on the initial occupancy grid is shown as well as the updated grid after 73 measurement steps. As expected from the results on the synthetic data, the area close to the starting position of the hand at the top left corner is explored systematically without leaving holes. In Figure 4.10, the first PRM based measurement path is shown as well as the updated grids after 68 measurements. The area close to the start position is not yet fully explored, but the next measurement path leads towards the lower left of the grid that has a high uncertainty.

Opposed to the synthetic experiments, in the real world, objects can move upon contact with the hand. This situation can be observed when comparing Figure 4.9c and 4.9f or 4.10d and 4.10h. To avoid the map becoming inconsistent one could employ visual tracking.



(a) Initial PRM based plan. (b) Initial OG from stereo. (c) Initial Prediction of OG. (d) Initial Scene.

(e) PRM based plan after 68 steps. (f) Stereo OG after 68 steps. (g) Prediction after 68 steps.. (h) Scene after 68 steps.

Figure 4.10: Snapshots from an exploration using an PRM based plan (covers only reachable workspace). Left Column) Coarse planning grid for the finger. Middle Left Column) Fine grid for stereo data. Middle Right Column) GP Predicted Grid using fine scale.

4.2 Height Map for Grasping and Manipulation

In the previous section, we have proposed a method for multi-modal scene exploration. Initial object hypotheses formed by active visual segmentation are confirmed and augmented through haptic exploration. The current belief about the state of the map is updated with measurements and yet unknown parts of the map are predicted with a Gaussian Process. Through the integration of different sensor modalities, a more complete scene model can be acquired by the robot. We showed that the prediction of the scene structure leads to a valid scene representation even if the map is not fully traversed. Furthermore, different exploration strategies were proposed and evaluated quantitatively on synthetic data. Finally, we showed the feasibility of our scene representation and exploration strategies in a real world scenario.

The proposed method has been studied using the classical occupancy grid as a scene representation. Although this representation is useful for tasks like collision avoidance or detection of free space for placing objects, it does not provide accurate enough information for grasp and manipulation planning.

In this section, we will re-use the previously gained insights and apply them to the problem of representing a table-top scene as a height map. This 2.5D representation is good compromise between a traditional occupancy grid and a full 3D reconstruction. It provides a simplified 3D model and is therefore better suited for manipulation and grasp planning. In [85], the authors showed that a 2.5D representation provides a good surface model and is more efficient than a full 3D reconstruction at the expense of some details.

As opposed to the previous approach in which scene understanding was posed as a classification problem (free or occupied space), here we treat it as a regression problem. As an exploration strategy, we adopt the PRM method using a discounted utility function that we showed to work best for rapidly producing a valid scene prediction. As a covariance function, we use the simple squared exponential kernel that provides the most suitable prior in our scenario.

Additionally, we will study the performance of different inference methods. While in the previous section we have used a sampling approach, here we will also evaluate approximate inference methods.

As an additional *virtual* sensing modality, we will consider object recognition and pose estimation.

4.2.1 Scene Representation

The height map, which is aligned with the table top, uniformly subdivides the scene into N cells $\mathbf{c}_i$ with coordinates (w_i, v_i). Each cell has a state x_i that specifies its height h_i. We model x_i as a normally distributed random variable with mean $\hat{x}_i$ and covariance $\mathbf{P}_i$. Our goal is to obtain an estimate $\hat{x}_i$ for the real state x_i given a sensor measurement z_t at the point t in time. We use the Kalman Filter as an estimator. Since we are assuming a static environment, the process model that predicts how the state of a cell changes over time becomes trivial:

$$x_{i,t+1} = x_{i,t} \tag{4.26}$$

where t and $t+1$ indicate subsequent points in time. We are considering three different sensor modalities s_r, all providing a height measurement directly. We assume that the measurement noise $\epsilon_r \sim \mathcal{N}(0, \mathbf{R}_r)$ of each of them is additive and Gaussian with zero mean and a specific covariance $\mathbf{R}_r$. The details of these models are explained in further detail in Section 4.2.2. The measurement model that relates the true state x_i of a cell $\mathbf{c}_i$ to a noisy measurement $z_{(i,r)}$ is therefore

$$z_{(i,r)} = x_i + \epsilon_r. \tag{4.27}$$

We define $\mathcal{X} = \{\mathbf{c}_i \mid 0 < i < N\}$, as the complete map estimate. Let $\mathcal{X} = \mathcal{X}_k \cup \mathcal{X}_u$ where $\mathcal{X}_k$ is the set of cells whose height value has already been estimated based on observations. $\mathcal{X}_u$ is the set of cells that has not been observed yet.

Additionally to updating the state estimate $\hat{x}_i$ for each cell $\mathbf{c}_i$ with each observation, we also update the covariance $\mathbf{P}_i$. Intuitively, we can see $\mathbf{P}_i$ as a representation of the uncertainty about $\hat{x}_i$. The smaller it is, the more reliable is the current state estimate. Equivalent to $\mathcal{X}$, we define $\mathcal{P} = \{\mathbf{P}_i \mid 0 < i < N\}$ as the complete uncertainty map consisting of $\mathcal{P} = \mathcal{P}_k \cup \mathcal{P}_u$.

Algorithm 2: Pseudo Code for Scene Exploration

```
Data: Point cloud $\mathcal{S}$ segmented into a set of object hypotheses $o_s$
Result: Fully explored $\mathcal{X}$
begin
    $t = 0,\ j = 0$
    $[\mathcal{X},\ \mathcal{P}] = \texttt{generateHeightMap}(\mathcal{S})$
    foreach $o_s \in \mathcal{S}$ do
        $id = \texttt{recognize}(o_s)$
        if $id \neq 0$ then
            $\delta = \texttt{estimatePose}(o_s, id)$
            $[\mathcal{X},\ \mathcal{P}] = \texttt{update}(\mathcal{X}, \mathcal{P}, id, \delta)$
        end
    end
    while $|\mathcal{X}_u| > 0$ do
        $\hat{\mathcal{X}} = \texttt{predict}(\mathcal{X})$
        $p_j = \texttt{planNextMeasurements}(\mathcal{X}, \hat{\mathcal{X}})$
        repeat
            $t++$
            $z_t = \texttt{observe}(\mathcal{X}, p_j)$
            $[\mathcal{X},\ \mathcal{P}] = \texttt{update}(\mathcal{X}, \mathcal{P}, z_t)$
        until $z_t \neq occ$
        $j++$
    end
end
```

Our approach to scene understanding is summarized in Algorithm 2. As a first step, a 3D point cloud $\mathcal{S}$ from the scene is gathered. The function `generateHeightMap` uses this as an input to build an initial scene model consisting of $\mathcal{X}$ and $\mathcal{P}$. The next step is to `recognize` object hypotheses generated through an active segmentation approach and estimate their pose. If this is successful, the initial $\mathcal{X}$ and $\mathcal{P}$ are `updated`. The remaining unexplored space $\mathcal{X}_u$ is `predicted` and then explored haptically with an active learning scheme. In the following, we will introduce the three different sensor modalities in more detail and describe the exploration planning and map update steps.

4.2.2 Sensor Modalities

In this section, we present three different sensing modalities. They provide height measurements of positions in the scene. The sensor models including measurement noise are formalized.

4.2.2.1 Stereo Vision

For acquiring an initial point cloud from the scene, we use the active vision system presented in Section 2.2. An example point cloud can be seen in Figure 4.11a. For creating a height map from stereo measurements, we are applying a similar approach as presented in Gallup et al. [85].

From Point Cloud to Height Map The basis of the height map computation is formed by a probabilistic 3D occupancy grid of the point cloud. We use the technique proposed in Wurm et al. [233] that computes an octree for an efficient decomposition of the space. The method employs the occupancy update technique proposed by Moravec and Elfes [157]. Given a uniform prior ($p(\mathbf{v}) = 0.5$) over the whole grid, all the stereo observations are incorporated to determine the probability of each voxel $\mathbf{v}$ to be occupied.

In detail, the value of the vote ϕ_v for each voxel is computed as follows:

$$\phi_v = \begin{cases} -1 & \text{if } l(\mathbf{v}) < l_{\text{free}}, \\ +1 & \text{if } l(\mathbf{v}) > l_{\text{occ}}. \end{cases}$$

where $l(\mathbf{v})$ returns the current log occupancy value of voxel $\mathbf{v}$. As threshold, we use log occupancy values of $l_{\text{occ}} = 0.85$ and $l_{\text{free}} =$ -0.4, corresponding to probabilities of 0.7 and 0.4 for free and occupied volumes as proposed by Wurm et al. [233].

Given these votes, our goal is to compute the height of all cells $\mathbf{c}_i$ on the table. Let $\{\mathbf{v}_j\}_{N_i}$ refer to the set of voxels that have the same (w, v) coordinates as $\mathbf{c}_i$ but whose voxel center is positioned at different heights h_j. They can also be seen as a column on cell $\mathbf{c}_i$. Then the height value $h_{(i,1)}$ of this cell is chosen to be

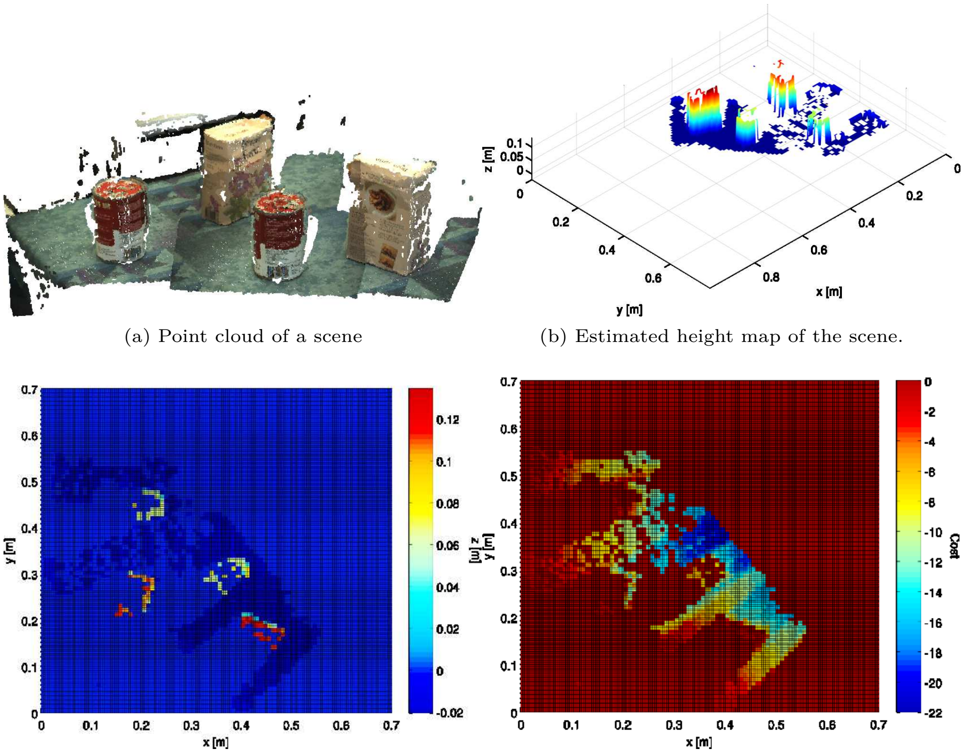
$$c^*_{(i,1)} = \min_{h_j} \sum_{\mathbf{v} \in \mathcal{V}_{>h_j}} \phi_v - \sum_{\mathbf{v} \in \mathcal{V}_{<h_j}} \phi_v \tag{4.28}$$

$$h_{(i,1)} = \arg_{h_j} c^*_{(i,1)}. \tag{4.29}$$

The set $\mathcal{V}_{>h_j}$ contains all those voxels whose height is bigger than h_j. Similarly, $\mathcal{V}_{<h_j}$ contains all those voxels whose height is smaller than h_j. This minimisation results in a height estimate $h_{(i,1)}$ in which the majority of voxels below it are occupied and the majority of voxel above it are empty. An example for the estimated height map of the point cloud in Figure 4.11a is shown in Figure 4.11b and 4.11c. An example for a cost map containing $c^*_{(i,1)}$ as computed in Equation 4.28 is shown in Figure 4.11d. Given N_i voxels with $\phi_v \neq 0$ in the column on cell $\mathbf{c}_i$, $c^*_{(i,1)}$ can have values between $-N_i$ and 0. We therefore normalise and shift this cost to map it to a range between 0 and 1.

$$\bar{c}_{(i,1)} = \frac{c^*_{(i,1)}}{N_i} + 1 \tag{4.30}$$

Measurement Noise Model As discussed above, each measurement modality r has an individual noise model with $\epsilon_r \sim \mathcal{N}(0, \mathbf{R}_r)$. The measurement covariance $\mathbf{R}_1$ for stereo vision and a specific cell $\mathbf{c}_i$ is computed as follows. The normalized cost value $\bar{c}_{(i,1)}$ of choosing $h_{(i,1)}$ will be high for columns that contain many noisy 3D points. For the corresponding cells on the table, we want this uncertainty about



(a) Point cloud of a scene

(b) Estimated height map of the scene.

(c) Height map from above with color scale in meters visualizing $h_{(i,1)}$.

(d) Cost map of the scene visualizing $c^*_{(i,1)}$.

Figure 4.11: From a 3D point cloud to a height map.

their true height value to be reflected in the measurement covariance. Therefore, we parameterize $\mathbf{R}_1$ with $\bar{c}_{(i,1)}$ as follows

$$\mathbf{R}_1(\bar{c}_{(i,1)}) = \bar{c}_{(i,1)} * \sigma_1^2 \tag{4.31}$$

where σ_1 is a user selected fixed noise variance. This parameterization has the effect, that height values determined with a higher cost also have a higher measurement noise variance.

4.2.2.2 Recognition and Pose Estimation of Known Objects

The second measurement modality recognizes objects and estimates their pose. This gives us information beyond the directly visible parts of the scene like object height estimates in occluded parts. Different from the other two sensing modalities, object recognition and pose estimation can be considered as a *virtual sensor*.

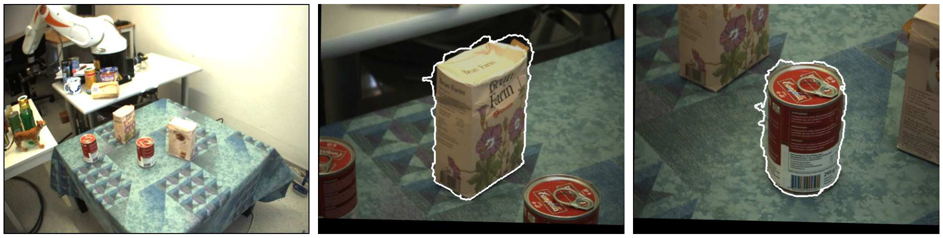


Figure 4.12: Wide-field camera view and segmentation results of objects in the scene.

In Persson et al. [169], this is defined as a collection of several physical sensors dedicated to a specific recognition task of real world concepts. The weights of the different data streams can be learned by for example AdaBoost as in [169] or by soft cue integration using an SVM as proposed by Björkman and Eklundh [33]. For simplicity, here we empirically determined values that result in robust object recognition.

For each visited view point, the object in the center of the image is segmented from its background using the vision system described in Section 2.2.1. Figure 4.12 shows an example. The identity of the foveated object is sought in a database of 106 known objects. Two complementary cues are used for recognition; scale invariant features (SIFT) [139] and color co-occurrence histograms (CCH) [88]. The cues are complementary in the sense that SIFT features rely on objects being textured, while CCH works best for objects of a few, but distinct, colors. An earlier version of this system, including a study on the benefits of segmentation for recognition, can be found in Björkman and Eklundh [33].

If an object is identified and it is known to be either rectangular or cylindrical, the pose is estimated using the cloud of extracted 3D points, with object dimensions given by a lookup in the database. However, if object identification fails, *i.e.* if the object is unknown or the segmented region does not correspond to a real physical object, the recognition modality is ignored.

The 3D point cloud is initially projected down to the 2D supporting surface, where fitting of points to object models is done using random sampling (RANSAC) [80]. For cylinders, pairs of sample points are drawn from the point cloud. For each such pair, two hypothetical 2D circles are found with the radius provided by the object identity. These hypotheses are then tested against the full set of points. The one with the minimum median distance from each point to the corresponding circle is selected as an initial estimate. The selected hypothesis is then refined as follows. The full point cloud is moved so as to be aligned with a distance map based on an ideal circle of the known radius, using sums of distances as fitting criteria.

For rectangular object, the fitting procedure is similar. Pairs of points are drawn and six hypotheses are given for each pair, assuming that they fit either side of the known object. Unlike the cylinders, additional to the position also the orientation

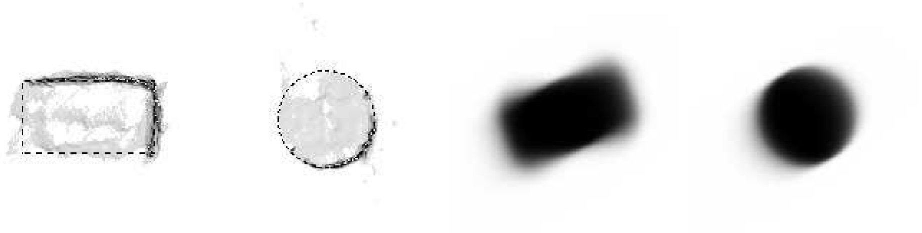


Figure 4.13: Results of the pose estimation approach applied to the segmented point clouds shown in Figure 4.12. Left) Pose estimation results. Right) Error probability of the pose estimation.

needs to be estimated. The most likely orientation is defined as the one that leads to the smallest rectangle that encloses at least 90% of the 2D points in the projected point set. The position of the rectangle is refined with a distance map similar to the circle. The combination of random sampling, least median fitting, distance maps and enclosing distances is chosen to ensure maximum robustness in the presence of a large amount of outliers.

Once the position and orientation of an object is estimated, a probabilistic height map is computed. This is done by anisotropic blurring of an ideal height map. The amount of blur is assumed to be proportional to the errors of the individual 3D points which is dependent on their depth relative to the camera and a given amount of image noise. Blur is also introduced to account for errors in each object parameter. This is approximated by the shape of the distance map fitness function, when each parameter is varied. Example results of the pose estimation and error modeling are shown in Figure 4.13 and 4.14.

From Pose to Height Map Given a probability map that contains for each grid cell $\mathbf{c}_i$ the probability $p(\mathbf{c}_i == o_s)$ of how likely it is occupied by the recognized object o_s, we generate a height map as follows. For each cell with $p(\mathbf{c}_i == o_s) > 0.5$, we set the corresponding height measurement $h_{(i,2)}$ to the known height of the recognized object o_s.

Measurement Noise Model The less likely a cell is to be occupied by an object, the higher its uncertainty should be to have the corresponding height value. We compute the measurement noise variance $\mathbf{R}_2$ by parameterizing it with $p(\mathbf{c}_i == o_s)$ as follows

$$c_{(i,2)} = \sqrt{p(\mathbf{c}_i == o_s). * (1 - p(\mathbf{c}_i == o_s))} \tag{4.32}$$

$$\mathbf{R}_2(c_{(i,2)}) = c_{(i,2)} * \sigma_2^2 \tag{4.33}$$

where σ_2^2 is again a user chosen fixed noise variance.

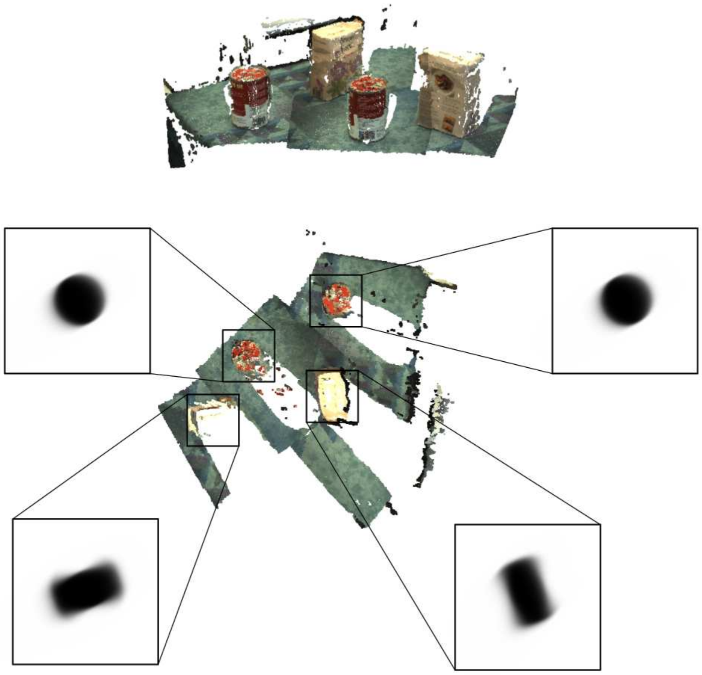


Figure 4.14: Error of the pose estimation shown in the context of the point cloud.

4.2.2.3 Haptic Measurements

As the third sensor modality, we consider haptic measurements that can be actively acquired by moving the robot hand through the scene. These observations are complementary to stereo reconstruction and object recognition. They can provide the robot with height measurements of visually occluded positions in the scene occupied by unknown objects.

In this section, we will evaluate haptic exploration only in simulation. We therefore assume that we can directly measure the height $h_{(i,3)}$ of a cell $\mathbf{c}_i$. The measurement noise $\mathbf{R}_3$ is assumed to be constant for all haptic measurements. For height measurements with the real system, haptic exploration primitives have to be developed. These could be based on the method presented before in Section 4.1.2.

4.2.3 Scene Update

As mentioned previously, we model the state of each cell, i.e. its height, as a random variable with Gaussian distribution. For estimating its mean and covariance, we use the Kalman Filter. Following the process and measurement model from Equation 4.26 and 4.27, we update the state estimate $\hat{x}_i$ and covariance $\mathbf{P}_i$ for cell $\mathbf{c}_i$ at time $t+1$ as follows:

$$\begin{aligned}
\hat{x}^-_{i,t+1} &= \hat{x}_{i,t} \\
\mathbf{P}^-_{i,t+1} &= \mathbf{P}_{i,t} \\
\mathbf{K}_{i,t+1} &= \frac{\mathbf{P}^-_{i,t+1}}{\mathbf{P}^-_{i,t+1} + \mathbf{R}_r} \\
\hat{x}_{i,t+1} &= \hat{x}^-_{i,t+1} + \mathbf{K}_{i,t+1}(z_{(i,t+1,r)} - \hat{x}^-_{i,t+1}) \\
\mathbf{P}_{i,t+1} &= (\mathbf{I} - \mathbf{K}_{i,t+1})\mathbf{P}^-_{i,t+1}
\end{aligned} \tag{4.34}$$

given an observation $z_{(i,t+1,r)}$ made with modality r.

4.2.4 Scene Prediction

In traditional exploration techniques, no assumptions are made about places in space that have not been observed yet. In accordance with related work [162, 85], we have shown in Section 4.1 that we can make use of spatial correlation to predict the structure of an environment based on already measured parts of it.

Spatial correlation in our case means, we know that cells in the grid that are close to occupied spaces but are due to occlusions not directly observable, are likely to be part of the occluding object. By modeling this spatial correlation between physically close locations in the environment, we can predict unobserved places from observed ones. Then instead of exploring the whole environment exhaustively we can select to confirm the predicted map at specifically uncertain places. In Section 4.1 we have shown the spatial correlation in a 2D occupancy grid can be modeled with a Gaussian Process. The assumption of independence of neighboring cells in a traditional occupancy grid is then removed.

In this section, we are using a height map instead of an occupancy grid. We will show that also in this representation spatial correlation can be similarly modeled with a GP. Compared to the previous section, scene prediction is conducted through *Gaussian Process Regression* instead of *Classification* of cells into *occupied* or *empty*. Given a set of input values $\mathbf{x}$ and the corresponding target values $\mathbf{y}$, the goal of GP regression is to estimate the latent function $g(\cdot)$ that generated these samples. In our case, $\mathbf{x}$ is constituted by the set of observed cells $\mathbf{c}_k \in \mathcal{P}_k$. The target values $\mathbf{y}$ are the corresponding noisy height estimates h_k. Together they form the training set $\{\mathbf{x}_i, y_i\}_M$ with cardinality M. As has already been detailed in Section 4.1.4.1, the relation between the input and target values assuming Gaussian additive noise can be modelled as

$$g(\mathbf{x}_i) = \phi(\mathbf{x}_i)^T\mathbf{w}, \quad y_i = g(\mathbf{x}_i) + \epsilon$$

$\mathbf{w}$ is a vector of weight parameters of the linear model. $\phi(\mathbf{x}_i)$ is a function that maps D-dimensional input data into an N-dimensional feature space.

From this formulation, we can derive the posterior likelihood of the weight parameters given the training data. Primarily, we are interested in predicting the state $y' = h_j$ of an unobserved cell $\mathbf{c}_j \in \mathcal{P}_u$ referred to as $\mathbf{x}'$ in the following. In Section 4.1.4.1, we have shown that this prediction $g(\mathbf{x}') \sim \mathcal{N}(\mu, \Sigma)$ follows a Gaussian distribution with

$$\begin{aligned}\mu &= k(\mathbf{x}', \mathbf{x})^T [\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_M^2 I]^{-1} \mathbf{y} \\ \Sigma &= k(\mathbf{x}', \mathbf{x}') - k(\mathbf{x}', \mathbf{x})^T [\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_M^2 I]^{-1} k(\mathbf{x}', \mathbf{x}).\end{aligned}$$

The Gram matrix $\mathbf{K}(\mathbf{x}, \mathbf{x})$ is computed based on a covariance function $k(\cdot, \cdot)$. In Section 4.1, we showed that the squared exponential covariance function is a good prior for our scenario. Therefore, we are using it for the height map representation as well. An example for this prediction given a 2D map of a partially explored scene is given in Figure. In Section 4.2.6, we will show quantitatively on synthetic data that a GP predicted map is a reasonable estimate of the ground truth.

Inference Methods To predict the height value of an unobserved cell $\mathbf{x}'$ and its uncertainty, we need to evaluate the mean and variance of $g(\mathbf{x}')$. Especially, the computation of the variance is expensive since it involves inverting the covariance matrix $\mathbf{K}(\mathbf{x}, \mathbf{x})$. If the set $\mathcal{P}_k$ of previously observed grid cells is of cardinality M, then $\mathbf{K}$ is of dimension $M \times M$. Hence, inverting $\mathbf{K}$ is of complexity $\mathcal{O}(M^3)$.

Ideally, we would like the robot to be able to represent, predict and explore a scene of its current radius of action in real-time. Therefore, we not only need an efficient scene representation that scales well to different resolutions, but also an efficient inference method for prediction. In this section, we will evaluate and compare three different inference methods.

Confidence-based Sampling In Section 4.1, we used a sampling approach in which data points are sampled according to their confidence. In this section, we will use the uncertainty map $\mathcal{P}_k$ containing the covariance values $\mathbf{P}_i$ of all observed grid cells $\mathbf{c}_i$ in $\mathcal{X}_k$. Cells with a smaller covariance, i.e., a smaller uncertainty about their height estimate, are more likely to be part of the training set $\mathbf{x}$. This set will contain a fixed number of samples from free and from occupied space. The number of free samples will be higher than the number of occupied samples, thereby reflecting structure of the scene.

Uniform Sampling Additionally to the confidence-based sampling, we will also use uniform sampling to assemble the training set $\mathbf{x}$. The same number of free and occupied samples are drawn as for the confidence-based case.

Subset of Regressors The aforementioned sampling methods only consider the selected subset of the observed grid cells for constructing the Gram matrix $\mathbf{K}$. A different approximation to the full GP regression problem is proposed in [181] which we will briefly summarize here. It also relies on a *subset of regressors* (SR). However, it takes the correlation of this subset to the remaining data in the training set into account.

It has been shown that the mean predictor of $g(\mathbf{x}')$ can be obtained by formulating it as a finite-dimensional generalized linear regression model:

$$g(\mathbf{x}') = \sum_{i=1}^{M} \alpha_i k(\mathbf{x}', \mathbf{x}_i) \quad \text{with } \alpha \sim \mathcal{N}(0, \mathbf{K}^{-1}). \tag{4.35}$$

If we consider only a subset $\{\mathbf{x}_j, y_j\}_N$ of regressors of cardinality N, then Equation 4.35 becomes

$$g_{SR}(\mathbf{x}') = \sum_{j=1}^{N} \alpha_j k(\mathbf{x}', \mathbf{x}_j) \text{ with } \alpha_j \sim \mathcal{N}(0, \mathbf{K}_{jj}^{-1}). \tag{4.36}$$

Given the subset of regressor as $\mathbf{x}_{SR}$, it follows that $g_{SR}(\mathbf{x}') \sim \mathcal{N}(\mu_{SR}, \Sigma_{SR})$ with

$$\mu_{SR} = k(\mathbf{x}', \mathbf{x}_{SR})^T [\mathbf{K}_{NM}\mathbf{K}_{MN} + \sigma_M^2 \mathbf{K}_{NN}]^{-1} \mathbf{y} \tag{4.37}$$

$$\Sigma_{SR} = \sigma_M^2 k(\mathbf{x}', \mathbf{x}_{SR})^T [\mathbf{K}_{NM}\mathbf{K}_{MN} + \sigma_M^2 \mathbf{K}_{NN}]^{-1} k(\mathbf{x}', \mathbf{x}_{SR}). \tag{4.38}$$

To keep the notation uncluttered, $\mathbf{K}_{NM}$ refers to $\mathbf{K}(\mathbf{x}_{SR}, \mathbf{x})$, $\mathbf{K}_{MN}$ to $\mathbf{K}(\mathbf{x}, \mathbf{x}_{SR})$ and equivalently $\mathbf{K}_{NN}$ to $\mathbf{K}(\mathbf{x}_{SR}, \mathbf{x}_{SR})$. The matrix computations in Equation 4.37 and 4.38 are then of complexity $\mathcal{O}(N^2 M)$. Evaluating the mean takes time $\mathcal{O}(M)$ and the variance $\mathcal{O}(M^2)$.

Note that under the SR model, the covariance function of two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ becomes $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_{SR})^T K_{NN}^{-1} k(\mathbf{x}_j, \mathbf{x}_{SR})$ and is therefore only dependent on the correlation with the so called *active set*. This leaves the question, how this set is chosen. Many different methods have been proposed in the literature. Here, we will use the result from the confidence-based sampling as the subset of regressors.

4.2.5 Active Exploration

Given a partial map of the environment, we want to efficiently explore the remaining unknown parts with haptic sensors. Efficiently here means that we want to minimize the number of measurement actions needed to reach a sufficient understanding about the scene as quickly as possible. In Section 4.1, we found that an active learning scheme based on PRM path planning in combination with a discounted utility function performs best when the goal is to reach a valid scene prediction already early on in the exploration process. This scheme supports the Gaussian Process based map prediction by selecting most informative observations, i.e., measurements that maximise the expected gain in Shannon information.

4.2.6 Experiments

We evaluate the proposed approach quantitatively on synthetic data. We are specifically interested in analyzing how well a Gaussian Process can predict a height map from partial observations compared to different baselines. In that context, we will also study the three inference methods we proposed.

Furthermore, we will explore how much the different sensor modalities improve the prediction. Additionally to stereo vision, we consider object recognition in combination with pose estimation and haptic exploration.

4.2.6.1 Data Set

We generated 50 different synthetic environments (1000 × 1000 mm) of the kind shown in Figure 4.15g. Every scene contains ten objects that can either be of cylindrical or box shape and are spread on a table. Their intrinsic attributes like height, aspect ratio or radius as well as their extrinsic attributes like position and orientation were chosen randomly within pre-determined ranges. Overlapping of objects was allowed so that fewer than ten connected components can occur as well as more complex shapes.

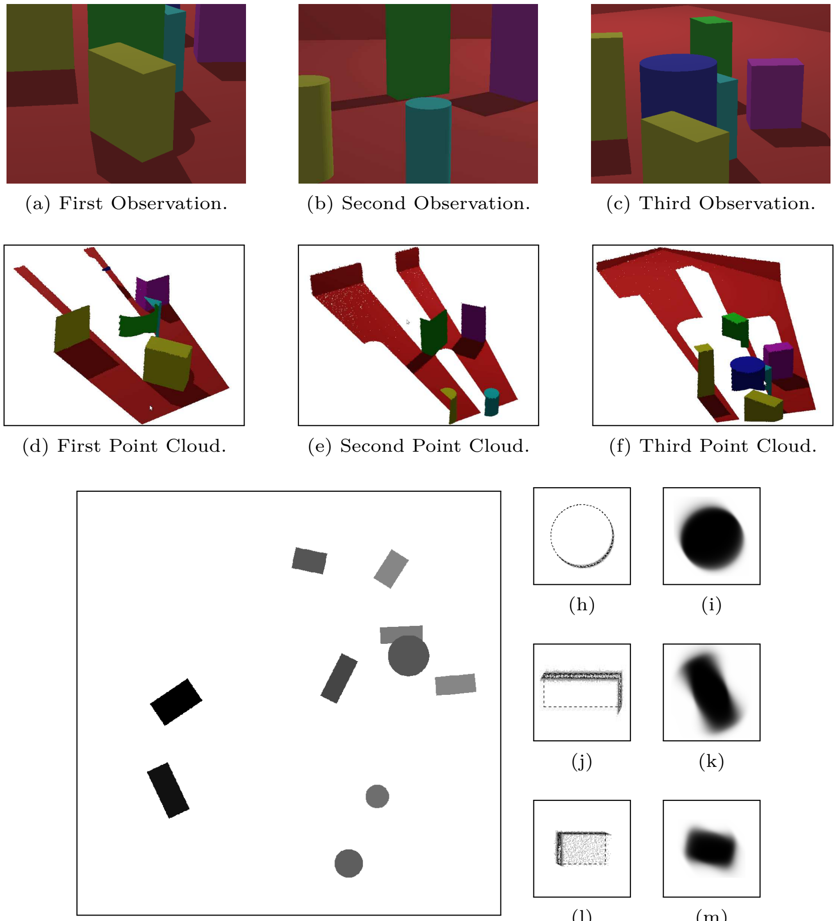
For every scene, we simulated three camera observations such that in each of them one randomly chosen object is in fixation. This is close to our real active vision system in which stereo reconstruction is performed once an object is properly fixated. For each observation, we reconstruct a point cloud using the ray-tracer YafaRay [234]. We model the noise from the stereo system by assuming zero-mean Gaussian noise of a specific covariance in the image. This results in a point cloud with stronger noise in locations that are farer away from the camera. An example for the camera observation as well as the point clouds can be seen in Figures 4.15a to 4.15f. The height maps corresponding to the observations were computed as described in Section 4.2.2.1. For obtaining an initial height map estimate $\mathcal{X}$ as well as a corresponding uncertainty map $\mathcal{P}$, the update Equations 4.34 with the corresponding noise covariance in Equation 4.31 are applied.

We assumed that each of the three fixated objects got segmented and recognized so that their pose can be estimated from the segmented point cloud as described in Section 4.2.2.2. Examples for the recognition results on the synthetic data can be seen in Figure 4.15h to 4.15m. This results in three additional height and cost maps that can be used to update the current $\mathcal{X}$ and $\mathcal{P}$. The same update equations as for the height maps from stereo measurements are applied with the noise covariance corresponding to the recognition sensor modality.

The difference between the state estimate and corresponding covariance with and without the recognition results can be observed in Figure 4.16. The height map that includes additional recognition results is more complete. The covariance of this map shows more certainty about height estimates at the center of recognized objects and more uncertainty at their boundaries.

4.2.6.2 Measure of Comparison

We considered the problem of height map estimation as a regression problem. As a measure of divergence between ground truth and the estimated scene, we use the



(a) First Observation.

(b) Second Observation.

(c) Third Observation.

(d) First Point Cloud.

(e) Second Point Cloud.

(f) Third Point Cloud.

(g) Ground Truth Height Map.

(h)

(i)

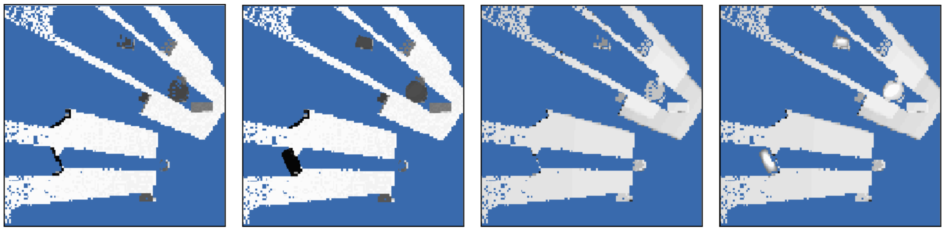
(j)

(k)

(l)

(m)

Figure 4.15: Example for synthetic dataset. 4.15a-4.15c Camera observations. 4.15d-4.15f Corresponding point clouds. 4.15h, 4.15j, 4.15l Object model fitted to segmented point cloud. 4.15i, 4.15k, 4.15m Probability for each grid cell to belong to recognized object.



(a) State Estimate without Recognition.

(b) State Estimate with Recognition.

(c) Covariance over State Estimate without Recognition.

(d) Covariance over State Estimate with Recognition.

Figure 4.16: 4.16a and 4.16c $\mathcal{X}$ and $\mathcal{P}$ without including results from recognition and pose estimation. 4.16b and 4.16d $\mathcal{X}$ and $\mathcal{P}$ after including results from recognition and pose estimation. Unexplored areas are marked in blue. In $\mathcal{X}$ the darker a pixel, the larger its estimated height. In $\mathcal{P}$ the darker a pixel, the more uncertain its height estimate.

standardized mean squared error (SMSE) that is defined as follows

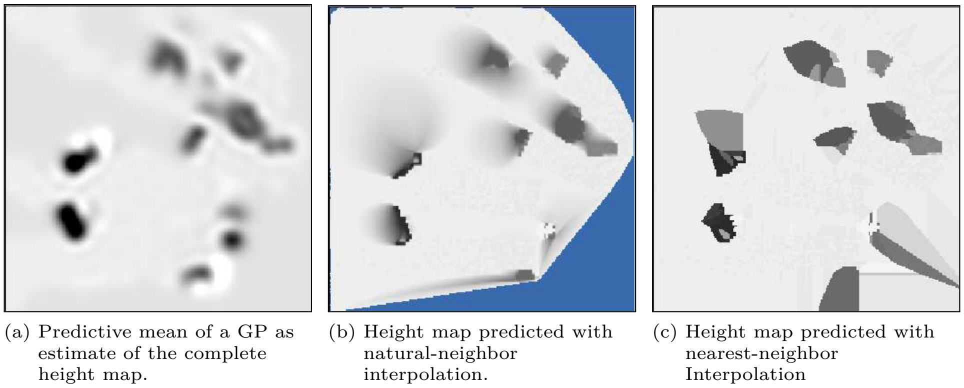
$$e = \frac{1}{\sigma_y^2} \mathbb{E}[(y - g(\mathbf{x}))^2] \tag{4.39}$$

where σ_y^2 is the variance of the target values of all test cases and the remainder is the expected value of the squared residual between the target value y and mean prediction $g(\mathbf{x})$ for each cell $\mathbf{c} \in \mathcal{X}$.

As a baseline for the height map prediction we use three methods. The first is *nearest-neighbor interpolation* in which each grid cell that has not been observed yet will be assigned the value of its nearest observed data point. Second, we apply *natural-neighbor interpolation* as proposed in [75] and implemented in Sakov [199]. This method is based a Voronoi tessellation of the space and is claimed to provide a smoother interpolation than the simpler nearest neighbor interpolation. We only use interpolation values within the convex hull of the dataset. And third, we use the trivial model which predicts each unknown data point using the mean height value of the whole test data. Figure 4.17 shows the resulting predicted height maps of the reference scene from Figure 4.15g. Different from the proposed approach based on a Gaussian Process, the baseline methods do not provide information about the expected quality of the prediction.

4.2.6.3 Predicting Height Maps with a GP

In this section, we will evaluate the accuracy of the predicted height map given different sensor modalities and using three different inference methods. The performance will be compared to the proposed baselines.



(a) Predictive mean of a GP as estimate of the complete height map.

(b) Height map predicted with natural-neighbor interpolation.

(c) Height map predicted with nearest-neighbor Interpolation

Figure 4.17: Example results of the different methods for predicting a height map. Unpredicted areas are marked in blue. Only stereo-vision based map data is used as input.

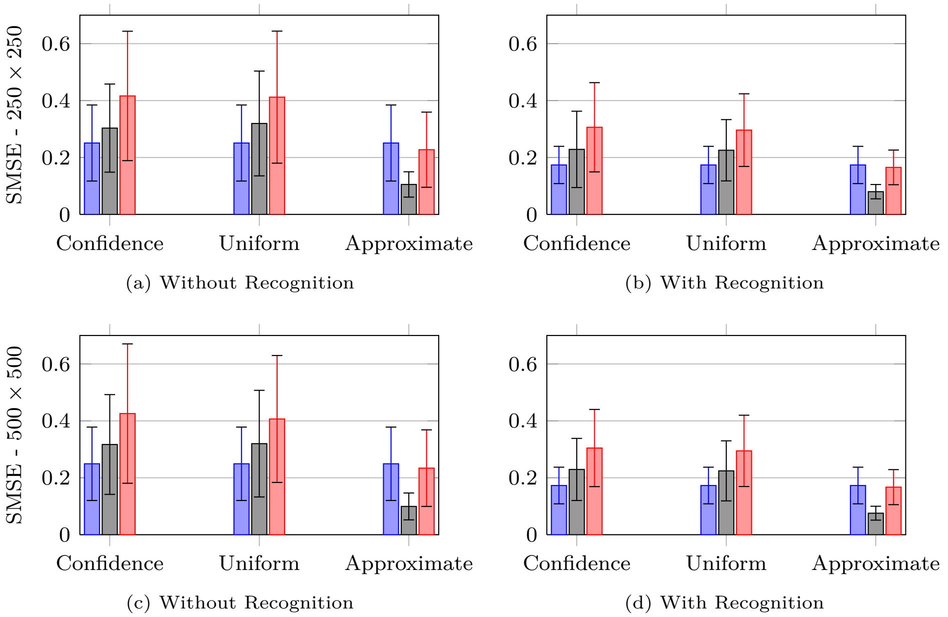
Inference Methods Compared In Section 4.2.4, we described three different approximate inference methods. While the first two form the Gram matrix only on a subset that is sampled in different ways (uniform or confidence-based), the third one also takes the correlation between the subset and the rest of the data into account. In the following, we will refer to them as *Confidence*, *Uniform* or *Approximate*.

In Figure 4.18a, we compare the standardized mean squared error (SMSE) between the *observed* regions of all 50 synthetic environments and the ground truth for the three different inference methods. As an orientation, we plot the SMSE between ground truth and the observations. This shows how high the SMSE is, induced only by the noise in the observations. Ideally, we would like the GP to predict the scene structure with a similar or even lower SMSE. Furthermore, we plotted the SMSE between the predictive mean of the GP and all observations in $\mathcal{P}_k$. This visualises how well the GP prediction complies to the observations.

In Figure 4.18a, we see that sampling of the training points according to their confidence leads to similar results as if uniform sampling is applied. In each of them, the observations are closer to the ground truth than the GP based prediction. Applying the approximate method leads to improved performance in predicting the ground truth. We can also observe a clear difference in the SMSE between prediction and observations for all the three inference methods. For the approximate method, the mean of this SMSE is less than half of those achieved by the sampling based methods. It also has a significantly lower variance over the 50 environments.

The reason for this is that the approximate method takes all the data in $\mathcal{P}_k$ into account while the sampling based approaches are just using a subset of this to compute an exact solution.

Furthermore, when comparing Figure 4.18b with Figure 4.18a we can observe



(a) Without Recognition

(b) With Recognition

(c) Without Recognition

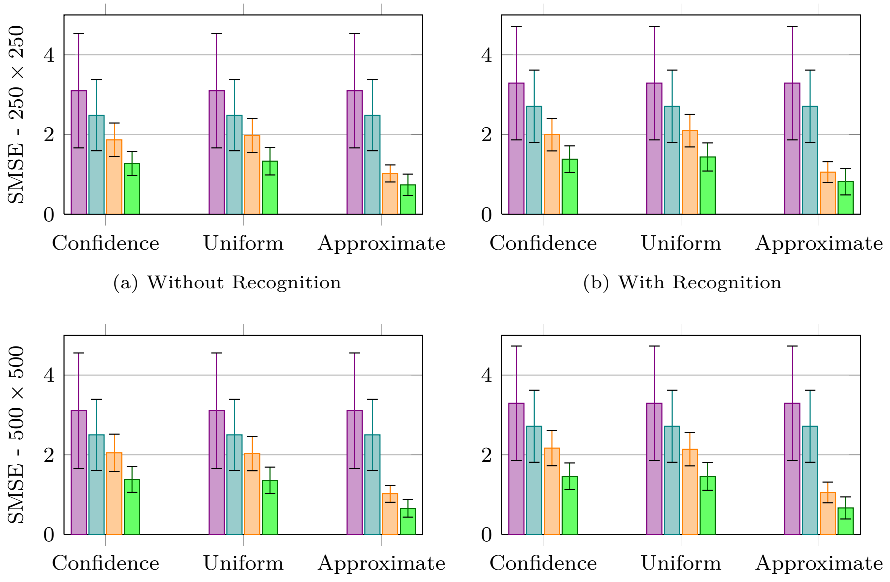
(d) With Recognition

Figure 4.18: Standardized Mean Squared Error (SMSE) in $\mathcal{P}_k$ of the scene between (i) 🟦 Observation and Ground Truth (reflects the overall noise in the system), (ii) ◼ Prediction and Observation and (iii) 🟥 Prediction and Ground Truth. Sampling of the training points according to their *Confidence* leads to similar results as if *Uniform* sampling is applied: The actual observations are closer to the ground truth than the GP based prediction. Applying an *Approximate* method to obtain the GP predictor leads to improved performance in predicting the ground truth. Furthermore, there is a clear overall improvement in mean and standard deviation when predicting ground truth including recognition information over the case when the recognition modality is not available. When increasing the resolution of the environment, no significant improvement in accuracy of the prediction is achieved.

a clear overall improvement in both mean and standard deviation when predicting ground truth including recognition information over the case when the recognition modality is not available.

In Figure 4.18, we also compare the prediction performance when using a scene resolution of 250 × 250 cells (top row) to a resolution of 500 × 500 (bottom row). We did not find a significant difference between the two.

Prediction Methods Compared Furthermore, we are interested in the performance of the GP in predicting the whole scene, i.e. all cells in $\mathcal{P}$. For this purpose, we compared it to three baselines, nearest-neighbor interpolation, natural-neighbor



(a) Without Recognition

(b) With Recognition

(c) Without Recognition

(d) With Recognition

Figure 4.19: Standardized Mean Squared Error (SMSE) in $\mathcal{P}$ between Ground Truth and (i) 🟪 Nearest-Neighbor Interpolation, (ii) 🟦 Natural-Neighbor Interpolation, (iii) 🟧 Trivial Prediction, (iv) 🟩 Gaussian Process Prediction which performs always best. As in Figure 4.18, the two sampling methods for selecting training points (*Confidence* and *Uniform*) perform worse in predicting the ground truth than when using the *Approximate* method to obtain a GP predictor. Confidence based sampling performs slightly better than uniform sampling.

interpolation and trivial prediction. Figure 4.19 shows that the GP based approach outperforms the three baselines. It also confirms the results, that the approximate method achieves a higher accuracy than the sampling approaches.

The trivial prediction performs surprisingly well, which indicates that the area of the scenes covered by objects is relatively small compared to free spaces. It delivers a significantly better prediction of the scene in the approximate case, because the mean object height is estimated on all observations not just on a sampled subset.

As opposed to Figure 4.18, there is no clear improvement in the prediction of the ground truth when including information from the recognition modality. This is due to the smoothing characteristic of the GP. Although, it has more data points to model the objects, it has none for the space surrounding their backside. At these places a false smoothing of the boundary is occurring.

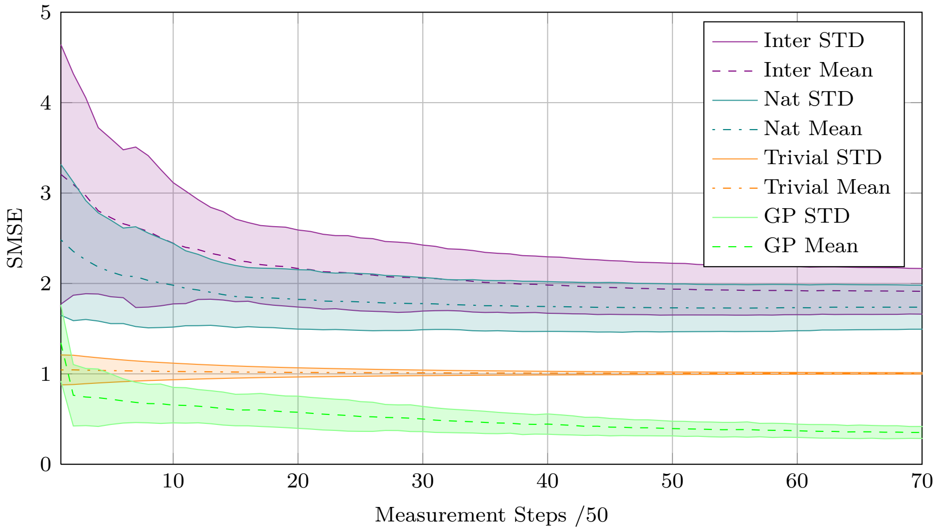


Figure 4.20: The development of the standardized mean squared error (SMSE) during exploration time averaged over 50 different synthetic environments.

Prediction Accuracy over Time The previous evaluations have been generated on the scene observations from stereo vision or recognition and pose estimation. To increase the accuracy of scene prediction, we want to enhance the current scene model with haptic data. We adopt the best exploration strategy from Section 4.1: a PRM based method using a discounted utility function. In Figure 4.20, we evaluate the development of the SMSE during this exploration process for all three prediction methods. For reducing the exploration time, we used the 50 example scenes with a resolution of 100 × 100 cells. Here, no recognition information is taken into account. Please note that the exploration actions are chosen based on the predictive variance of the GP. In general, the GP predictor produces a result that is much closer to the ground truth than the trivial model or the predictor based on nearest-neighbor and natural-neighbor interpolation. The error for the trivial model stays as expected nearly constant at an $SMSE = 1$ constant with a decreasing variance. The error achieved through interpolation increases quickly in the beginning, but then stagnates at a high error level. Natural-neighbors performs better than nearest-neighbor interpolation.

In Figure 4.21, we compared the SMSE between ground truth and GP prediction when taking recognition information into account or not. There is no significant difference between the two error means. When using recognition information, the error shows a slightly larger variance and higher mean over the 50 environments during the beginning of the exploration process. Again, this is due to the smoothing characteristic of the GP in the absence of data suggesting otherwise.

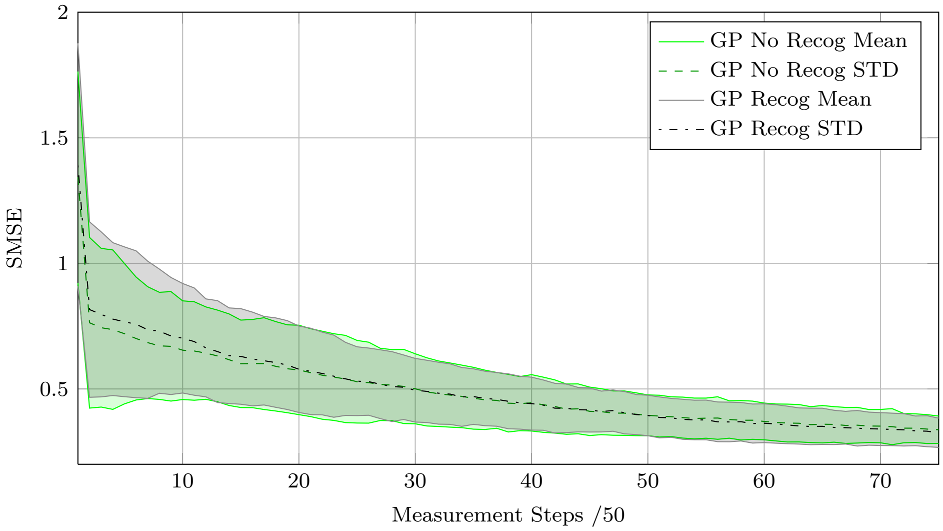


Figure 4.21: Comparison of the development of the SMSE during exploration time averaged over 50 different synthetic environments with and without including recognition information. There is no significant gain in including recognition information.

4.2.6.4 Summary

In this section, we have shown that a scene represented by a height map can be accurately predicted by a Gaussian Process. It consistently outperforms other prediction methods such as nearest and natural neighbor-based interpolation or by just simply taking the mean of all observations. We also found that approximate inference in which all the observations are taken into account is preferable over sampling methods. Additional recognition information does not increase accuracy of map prediction in the proposed GP framework. For better modeling a scene containing a mixture of known and unknown objects, clustering methods might be interesting to study. In that case, object hypotheses have to be taken explicitly into account. Clustering should be treated in a Bayesian framework to provide exploration with a measure of uncertainty.

4.3 Discussion

In this chapter, we proposed two approaches to scene representation for grasping and manipulation. One is based on an occupancy grid and the other on a height map. Their state $\mathbf{x}$ consists of a set of cells on the table indexed by their coordinates $\mathbf{x}_i = (u_i, v_i)^T$ on the table. In case of the occupancy grid, each cell has a value indicating its probability to be occupied. In the case of the height map, the cell value corresponds to the height of the object that is occupying this cell. We have also demonstrated how multiple modalities such as stereo vision, object recognition and pose estimation as well as haptic sensor data can be integrated into either of these representations. Each representation has different utilities in a grasping and manipulation scenario. The occupancy grid map outlines free and occupied spaces and is therefore suitable for deciding for example on where to place objects that the robot has picked up. A height map on the other hand carries additional information about the object height and therefore facilitates grasp planning.

We have shown that a Gaussian Process can be used as an implementation of the function $g(\cdot)$ for accurately predicting the value of unobserved cells in the scene.

The remaining challenge is to make this approach capable of real-time processing. The map of the scene should be predicted and updated rapidly during interaction. In this chapter, we have proposed to use knowledge about the table plane in the scene to reduce the dimensionality of the representation from 3D to 2D. Furthermore, we experimented with approximate inference methods. However, these approximate techniques still do not scale well enough to allow for real-time computation in large environments. Furthermore, a table plane may not be detectable in every scene.

An approach inspired by the ideas by Brooks [46, 47] and from the field of Active Vision [22, 20] would be to restrict the proposed high-resolution prediction to the task-relevant parts of the scene. Through this limitation, the map could be computed more often. Changes in the currently interesting part of the scene could be taken into account more rapidly.

To illustrate this idea, consider the following scenario. The goal is to pick up an object from a table-top scene. The active head fixates on this object and stereo reconstructs this part of the scene. The object is segmented from the background. However, the scene is only partially known due to occlusions.

Using the proposed approach of height map prediction through a Gaussian Process, we can predict the whole scene in the current view This information can then be used to plan a *safe* grasp. In this case this means, that the grasp is chosen such that the probability of the hand colliding with any other object on the table is minimal.

Furthermore, during the grasp, the environment might change due to moving objects. By having restricted the map to the region in the current viewpoint, a real-time update of the map that takes these changes into account becomes possible. This would then allow for online adaptation of the grasp parameters.

Once the object is picked up, we might want to put it down at another free spot in the environment. Given an approximate position in space, where this object should be put down, the head can start exploring the environment from there until a suitable place is found. Instead of using a fixed size of the scene as in this chapter, it could start from an initial size and extend it on demand.

This application of the scene prediction is in the spirit of active vision and behavioral robotics in that perception is done rapidly and only on demand in the context of certain behavior.

5

Enhanced Scene Understanding through Human-Robot Dialog

A full segmentation of all objects in a scene is desirable for many tasks such as recognition [32], grasp planning [8] or learning models for previously unseen objects that are re-usable for later re-recognition. This is a different subproblem of scene understanding from what has been proposed in Chapter 4. In the previous chapter, geometric scene structure was predicted independent of correct object segmentation.

A common problem of vision systems employed for object segmentation is that they might incorrectly group objects together or split them into several parts. This is commonly referred to as under- or over-segmentation. The two vision systems presented in Section 2.2 suffer from this limitation as well. The sensitivity to this problem depends on how close objects are to each other, how similar they are in appearance and on whether the correct number of objects in the scene is known in advance.

Interactive segmentation methods that aim at refining the robot's current scene model have recently gained a lot of attention. One approach addressing this problem, is to let the robot interact with the scene, e.g., through pushing movements and then use motion cues for verifying or refining object hypotheses [114, 151, 110, 28]. These papers focus on the question of what knowledge can be gained from observing the outcomes of robotic actions. The majority of them leaves the question open of how to select the action that provides the greatest amount of information.

In this chapter, we propose to put a "human in the loop" to achieve the objective of correct scene segmentation. This has the advantage of the robot being capable of learning new labels or symbols online and ground them in the current sensory data. We move towards this goal by combining state-of-the-art computer vision with a natural dialog system allowing a human to rapidly refine the model of a complex 3D scene. Through that approach, we harness the strengths of 'bottom up' automated processing with the 'top down' decision making power of a human operator. The resulting refined scene model forms the basis for a general symbol grounding problem [96].

Human input is not a new concept in object segmentation and scene understanding. Several previous systems have used it to guide automated processing or restrict the search space of algorithms both improving efficiency and decreasing false positives [109, 227, 115]. *GrabCut* [188] and *Lazy Snapping* [136] are the most related works on interactive segmentation. They require the user to operate a mouse either for selecting a region containing the object in the image or for drawing its rough outline. In this chapter, we aim for human-robot interaction using only natural dialog and no additional tools. Thereby, we are moving closer to how humans naturally interact with one another in such situations.

Several unique challenges exist when designing a framework for human robot interaction using computer vision. One of them is to minimize the iteration time of the vision component of such a system. This processing must be rapid enough as to not induce a large lag in the system increasing the mental load on the user [212]. As most state-of-the-art vision algorithm for scene understanding run in tens of minutes or hours per scene [215, 143, 134], they would be unacceptable for any "human-in-the-loop" system. Here, we propose to use the multi-object segmentation algorithm presented in Section 2.2.2, which is capable of providing acceptable responsiveness in the proposed algorithm.

Another design challenge is to minimize the necessary interaction between the autonomous system and the human operator. This means maximizing the value of each interaction to obtain the greatest discriminatory information. A very common approach to this kind of problem is to make a decision that maximises the expected gain in information or, equivalently, minimises the uncertainty in the system [109][3]. In this chapter, we propose the concept of entropy to characterize the quality of any object hypothesis. This helps in guiding the dialog system to resolve the most challenging and ambiguous segments first.

5.1 System Overview

In the following, we will give a brief system overview. Figure 5.1 visualises the different modules and their interconnections. As mentioned earlier, we use the vision system as proposed in Section 2.2.2 to acquire a point cloud of the scene. An initial segmentation is performed that groups close points with similar color traits [5]. The dialog system allows a human operator to provide responses to the robot's questions in a natural manner. The robot can be interrupted and corrected in mid-utterance allowing the operator to avoid the latencies associated with traditional dialog call and response conversation patterns.

The main contributions of this chapter lie in the scene analysis module that seeks to bridge the vision and dialog module. It fulfills two tasks: (i) determine areas of the scene that are the poorest object hypotheses and seek human arbitration, (ii) translate the human input to a re-seeding of the segmentation.

5.2 Dialog System

To facilitate interaction in the proposed system, a dialog system was implemented as shown in Figure 5.1. It is based on Jindigo – a Java-based open source dialog

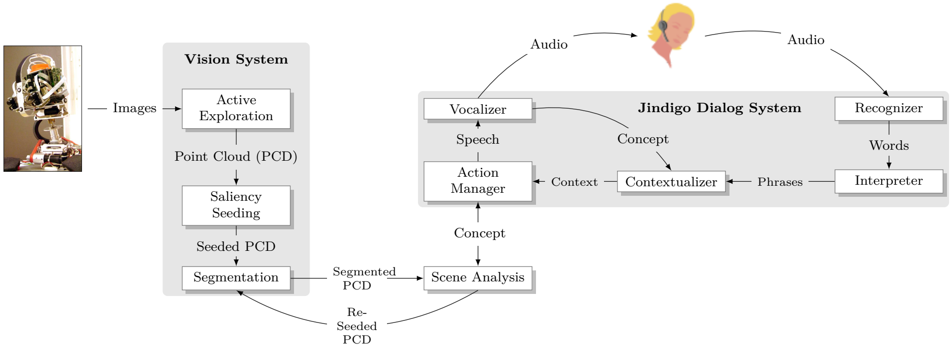


Figure 5.1: Overview of the System. Left) Karlsruhe Head. Middle) Vision System that actively explores the scene, reconstructs and segmentes a point cloud as also visualized in Figure 2.5. Right) Architecture of the Jindigo Dialog System. Vision and Dialog System are communicating through a Scene Analysis Module.

framework by Skantze [208]. This system overcomes a typical limitation of other dialog system: strict turn-taking between user and system. The minimal unit of processing in such systems is the *utterance*, which is processed in whole by each module of the system before it is handed on to the next. Contrary to this, Jindigo processes the dialog incrementally, word by word, allowing the system to respond more quickly and to give and receive feedback in the middle of utterances [202]. All components of the system are outlined in the right side of Figure 5.1. Here, we will only briefly describe the parts that are relevant to this thesis. For more details on Jindigo, we refer to Skantze [208] and [6].

When the user is saying something, the utterance is recognized and interpreted in the context of the whole dialog history and sent to the Action Manager (AM). The task of the AM is to decide which step to take next. To make these decisions, the AM communicates directly with the Scene Analysis (SA) component. If the AM or SA decides that the robot should say something, the semantic representation of this utterance is sent to the Vocalizer. In Section 5.3.1, this will be clarified with the help of an example dialog.

5.3 Refining the Scene Model through Human-Robot Interaction

From the visual scene segmentation as described in Section 2.2.2, we obtain an initial bottom-up analysis of the scene resulting in several object hypotheses.

The quality of the initial segmentation can vary due to occlusions, objects that are very close to each other or when the position and number of seed points does

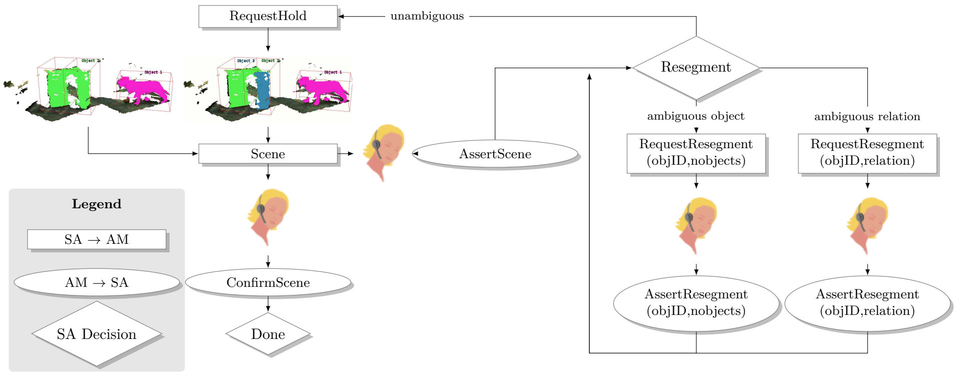

RequestHold
unambiguous
Resegment
ambiguous object
ambiguous relation
Scene
AssertScene
RequestResegment (objID,nobjects)
RequestResegment (objID,relation)
Legend
SA → AM
AM → SA
SA Decision
ConfirmScene
Done
AssertResegment (objID,nobjects)
AssertResegment (objID,relation)

Figure 5.2: Graph of the dialog loop showing the messages being passed between the action manager (AM) and the scene analysis (SA) module.

not correspond to the actual objects. The demonstrated system was configured to handle the most common configurations and given a corresponding vocabulary for verbal disambiguation. However, we will show later how our system can easily be extended to deal with more complex cases.

In the following sections, we will show how user utterances are interpreted and how natural language is generated using the example dialog shown in Table 5.1. This dialog is an instantiation of the general dialog loop shown in Figure 5.2. Furthermore in Section 5.3.2 and 5.3.3, we propose a method to (i) rank the object hypotheses in terms of their uncertainty and (ii) re-segment the incorrect hypotheses.

5.3.1 Contextual Interpretation and Generation of Natural Language

In a natural dialog about the structure of a scene, objects may be referred to in different ways. One way is to use their properties (e.g, *the red object* or *the leftmost object*). To be able to ground these descriptions into the actual objects that are being talked about, the dialog system must have a representation of the robot's hypothesis of the set of objects. Therefore, the initial segmentation of the scene is sent from the SA component to the AM, including the properties of objects and their IDs (see Table 5.1, Row 1).

Another way of referring to objects in dialog is by anaphoric expressions. For example a speaker may say *it* or *the segment* to refer to an object that has just been mentioned. A dialog system must be able to both understand and generate such expressions with the help of the dialog context. Another phenomena where the dialog context needs to be taken into account is elliptical (or fragmentary) expressions. These are expressions that lack propositional content, such as *the red one* or *to the left*. The dialog system needs to resolve these into full propositions,

1	SA → AM	Scene(objects:[Object(id:32 color:Green ...) Object(id:47 color:Yellow ...)])
2	AM→Voc	CheckScene(Scene(nobjects:2))
3	Voc→User	"I think there are two objects"
4	User→Rec	"No there are three objects"
5	Int→Con→AM→SA	AssertScene(Scene(nobjects:3))
6	SA→AM	RequestResegment(object:Object(id:32) nobjects:2)
7	AM→Voc	RequestResegment(object:Object(color:Green) nobjects:2)
8	Voc→User	"So, should I split the green segment?"
9	User→Rec	"No, the yellow one!"
10	Int→Cont	NegFeedback Object(color:Yellow)
11	Cont→AM	AssertResegment(object:Object(color:Yellow) nobjects:2))
12	AM→SA	AssertResegment(object:Object(id:47) nobjects:2)
13	SA→AM	RequestResegment(object:Object(id:47) relation:TopOf)
14	AM→Voc	RequestResegment(object:Object(ref:Anaphora) relation:TopOf))
15	Voc→User	"Are the objects on top of each other?"
16	User→Rec	"No, next to each other"
17	Int→Cont	NegFeedback NextTo
18	Cont→AM	AssertResegment(object:Object(ref:Anaphora) relation:NextTo)
19	AM→SA	AssertResegment(object:Object(id:47) relation:NextTo)
20	SA→AM→Voc	RequestHold
21	Voc→User	"Okay, just wait a moment"
22	SA→AM	Scene(...)
23	AM→Voc	CheckScene
24	Voc→User	"Is this correct?"
25	Voc→User	"Yes"
26	Int→Con	PosFeedback
27	Con→AM→SA	ConfirmScene

Table 5.1: Example dialog between human and robot and the information flow in the system. The second column indicates the currently active system modules. The third column shows the messages being send between the modules. SA = Scene Analysis. AM = Action Manager. Voc = Vocalizer. Rec = Recognizer. Cont=Contextualizer.

so called ellipsis resolution (e.g., *the red one* should perhaps be understood as *the red segment should be split into two objects* given the dialog context).

When the AM receives the initial scene, it starts to describe the scene, allowing the user to correct anything that is incorrect. It starts by stating the number of objects using the `CheckScene` action (Table 5.1, Row 2). This is transformed into text and speech by the Vocalizer (Row 3). Since the robot's hypothesis is incorrect, the user now corrects the robot. This utterance is unambiguous and therefore sent as it is all the way to the SA (Row 5). Notice how the semantic representation in Row 2 and 5 and the textual representation in Row 3 and 4 reflect each other. This illustrates how large parts of the domain grammars may be used for both interpretation and generation of utterances.

The SA requests help with refining the segment that is most likely to be incorrect (see the next section for how this choice is made). As can be seen in the example, the SA only refers to objects by their IDs. The AM then chooses the best way of referring to the object. It may for example be by the color (as in Row 7: *the green segment*) or by using an anaphoric expression (as in Row 14: *the objects*) depending on the current dialog history. The user now replies with an elliptical (fragmentary) expression (Row 9). Note that this utterance cannot be fully understood unless the dialog context is taken into account. This is precisely what the Contextualizer does - it re-interprets the ellipsis into a full proposition (Row 11). Thus, the `RequestResegment` from the robot combined with the `Object` from the user is transformed into an `AssertResegment` with this `Object` as argument. The same principles are applied in the ellipsis resolution in Row 18 (but where the spatial relation is replaced instead of the object).

As the scene is refined, the SA sends new scene hypotheses to the AM (Row 22). Finally, it will receive a `ConfirmScene` message from the AM (Row 27).

5.3.2 Identifying Poor Object Hypotheses through Entropy

To bootstrap the process of scene model refinement, the robot should be able to identify the most ambiguous object hypotheses and query the user about these first. We propose an entropy-based system to rank the object hypotheses according to their uncertainty. In general, the entropy $H[p(\mathbf{x})]$ of a distribution over a discrete random variable $\mathbf{x}$ with $\{\mathbf{x}\}_{j=0}^{M}$ is defined as follows:

$$H[P(\mathbf{x})] = -\sum_{j=0}^{M} p(\mathbf{x}_j) \ln p(\mathbf{x}_j). \tag{5.1}$$

and equivalently for a continuous variable with the integral. Intuitively, this means that sharply peaked distributions will have a low entropy while rather flat distribution will have a relatively high entropy. The maximum entropy is reached with a uniform distribution and is computed as $H_{max}[p(\mathbf{x})] = \ln(1/M)$.

Our observation is that single objects tend to be relatively homogeneous in an appropriate attribute space. Consider for example a correctly segmented single object and a segment in which two objects are contained as in Figure 5.3. The hue histogram of the correct hypothesis will show a narrower distribution than for the segment containing two objects. The same holds for the distribution of 3D points which will vary more in the second segment.

To test the hypothesis of entropy being a good indicator for the quality of object segmentation, we formalise it as binary classification problem. We compute the entropy for each initial object hypothesis over a number of different attribute histograms. Let N be the set of points in a segment and let us define a set of normalized histograms as follows:

1. 1D histogram $p(c)$ with 30 bins over color hue c.
2. 3D histogram $p_{BB}(c)$ over the voxelized BB.
3. 3D histogram $p(\mathbf{v})$ over a voxelized grid (10mm side length) of fixed size carrying the number of objects points in each voxel $\mathbf{v}$. The grid is aligned with the oriented object bounding box BB.
4. One normalized 1D histogram $p_{BB_x}(\mathbf{s}), p_{BB_y}(\mathbf{s}), p_{BB_z}(\mathbf{s})$ for each axis x, y and z of BB.
5. And similarly, one normalized 1D histogram $p_x(\mathbf{s}), p_y(\mathbf{s}), p_z(\mathbf{s})$ for each axis x, y and z of the grid of fixed size counting the number of 3D points falling into 10mm wide slices $\mathbf{s}$ along the respective axis.

For computing the oriented bounding box, we project the segmented point cloud onto the table as for example shown in Figure 4.13. The axes of the bounding box will be aligned with the two eigenvectors of this projection where $\mathbf{e}_a$ refers to the longer and $\mathbf{e}_b$ to the shorter one. The fixed size grid that is aligned with this bounding box can for example be chosen dependent on the visible portion of the scene in the foveal cameras.

Both 3D histograms, $p(\mathbf{v})$ and $p_{BB}(\mathbf{v})$, will be relatively sparse compared to the 1D histograms. While $p(\mathbf{v})$ reflects on the size of an object segment relative to what is visible in the foveal cameras, $p_{BB}(\mathbf{v})$ is normalized to the dimensions of the oriented bounding box and instead measures the complexity of the 3D distribution of points. The one dimensional histograms break down this complexity to the single axes of either the bounding box or the grid.

The entropy values of these histograms are normalized with their respective maximum entropy. We use these values to form a feature vector $\mathbf{f}_i$ containing for example the following values:

$$\mathbf{f}_i = (H[p(C)], H[p(\mathbf{v})], H[p_{BB}(\mathbf{v})], H[p_x(\mathbf{s})], H[p_y(\mathbf{s})], H[p_z(\mathbf{s})], H[p_{BB_y}(\mathbf{s})]) \tag{5.2}$$

for each object hypothesis i. From a set of initially segmented example scenes, we can extract labeled data to train an SVM with an RBF kernel for classifying an object hypothesis as either being correct or not. The probability estimates provided by [50] are used to rank the object hypotheses according to their uncertainty.

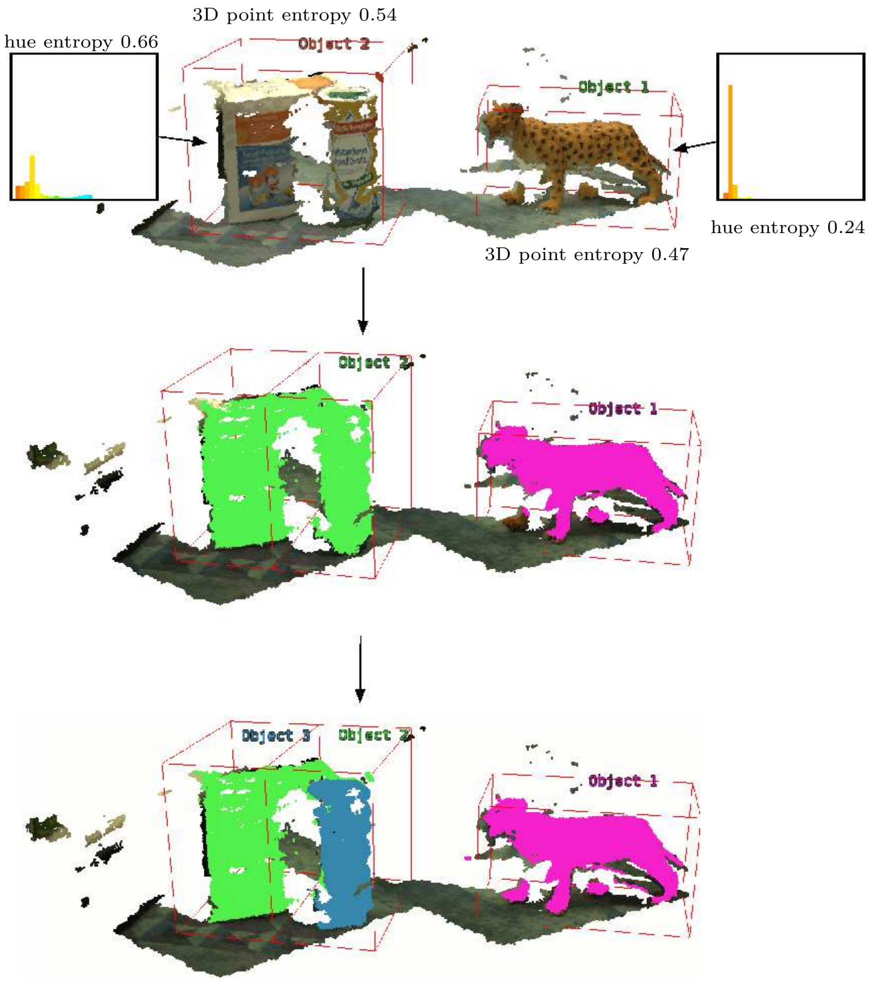


Figure 5.3: Visualization of the different stages of the scene understanding. Top: Point cloud from stereo matching with bounding boxes (BBs) around segmented objects and their hue histograms. Left segment containing two objects has a higher entropy in hue and 3D point distribution and is therefore considered more uncertain. Middle: Initial labeling of the segments. Left segment is re-seeded by splitting BB in two parts based on human dialog input. Bottom: Re-segmented objects. Three objects are correctly detected.

5.3.3 Re-Segmentation Based on User Input

Once a possible candidate for refinement has been identified, the algorithm proceeds to query the user to determine two things: (i) Is the current segmentation correct? And (ii) if not, what is the relative relationships of the objects in the current incorrect segment. We limit the user's options for spatial relationships to two objects being either next to, in front of or on top of one another. We propose this limited set of configuration to simplify the interaction. Furthermore, the robustness of the *Markov Random Field* (MRF) segmentation algorithm eliminates the need for tight boundaries. Despite the simplicity of these three rules, quite challenging scenes can be resolved in practice.

To begin the re-segmentation, the extents of the original segment are calculated to produce an object aligned bounded box. This bounding box is then divided in half along one of the three axes based upon the human operator's input. Once the bounding box is divided, the initially segmented points are relabeled and new Gaussian Mixture Models for the region are iteratively calculated as in [188]. A new graph is constructed based on the probability of membership to these new models. Energy minimization is performed for the new regions and repeated until convergence.

In a similar manner, the more rare situations in which two object hypotheses have to be merged can be dealt with. The 3D points initially labeled with two different labels would be re-labeled to carry just one.

5.4 Experiments

In this section, we will show that through the interaction of a robot with a human operator, the resulting segmentation is improved and that this interaction is made more efficient through the use of an entropy based selection mechanism.

There exist only a few databases containing RGB-D data, e.g. in [129]. They are targeted at object recognition and pose estimation instead of segmentation and usually contain scenes with single objects only. Therefore, we created our own database by recording 19 scenes containing two to four objects in specifically challenging configurations, see Figure 5.4. Reconstructing the point cloud and obtaining the seeding for the initial segmentation is done through the active vision system described in Section 5.1. An example point cloud for the first scene is shown in Figure 5.3.

In the recorded scenes, objects are hard to separate from each other due to similar appearance or close positioning. There is one incorrectly segmented pair of objects in each of the 19 scenes. Each scene is labeled with human ground truth, thereby separating each object perfectly.

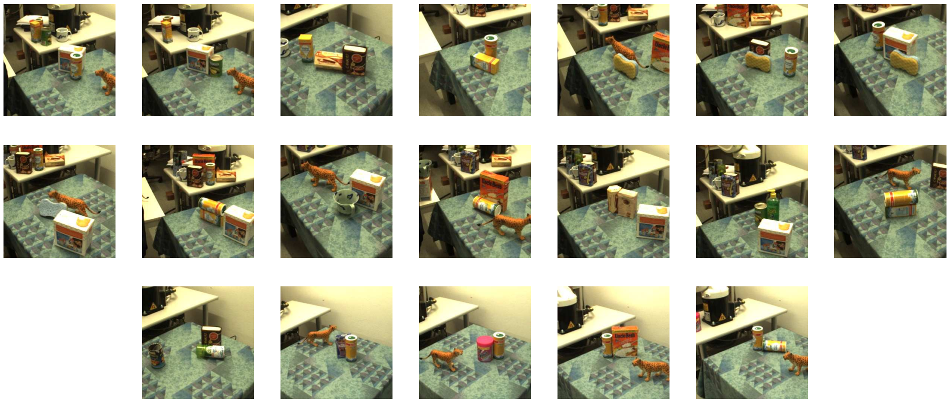


Figure 5.4: Pictures of the 19 scenes in the data set on which the experiments are performed. Ordering from left to right and then top to bottom.

5.4.1 Identification of Incorrect Object Hypotheses

We proposed an entropy-based system to select most uncertain object hypotheses to be confirmed by the user. For evaluation, we captured and labeled 303 examples 128 incorrect (positive) and 175 correct (negative) hypotheses. The latter consists mainly of the data presented in Chapter 6: 16 different objects each in 8 different orientations as shown in Figure 6.5 and 6.6. Additionally, we include correct segments from scenes similar to those shown in Figure 5.4. Thereby we ensure that the positive examples from the dataset contain objects in a variety of different orientations or with parts missing due to occlusions.

We were interested in i) which entropy features are important for distinguishing between correct and incorrect segments and ii) how well we can perform this classification task when using a combination of the best features in a feature vector.

For both experiments, we randomly divided the data into a training and test set. The training set contains 25 positive and 25 negative examples, the test set 253 examples. The complete set was randomly divided five times and the results were combined into an average receiver operating characteristic (ROC) curve.

Figure 5.5 Left) shows a comparison of different single or combined entropy features in terms of area under the ROC curve (AUC). First of all, we can observe that the entropy features computed on the fixed size grid provide enough information to achieve a classification performance of $AUC > 0.8$. This shows that the size of the segment relative to the size of the visible portion of the scene in the foveal cameras is a good indicator of segment quality.

On the other hand, the entropy features computed on the tight oriented bounding box achieve a significantly lower AUC. Especially when looking at the single dimensions, the classification performance of a feature vector containing $H[p_{BB_x}(\mathbf{s})]$

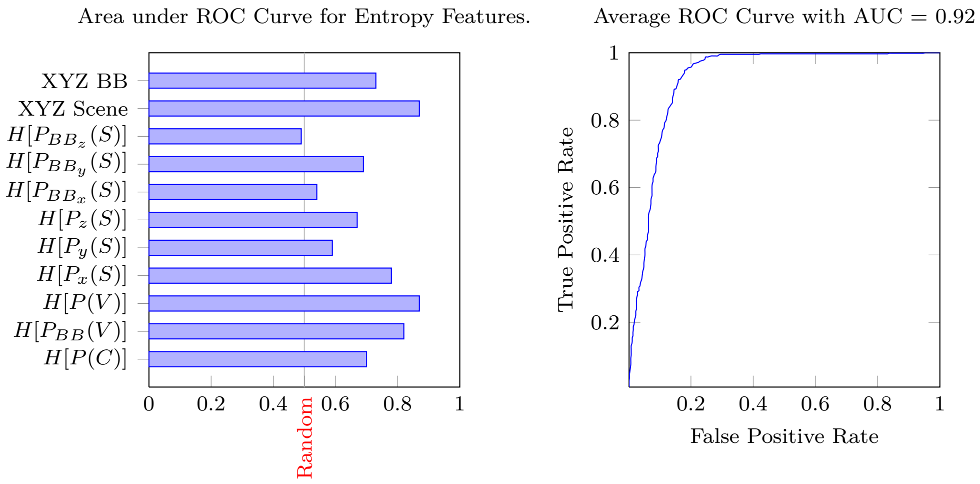
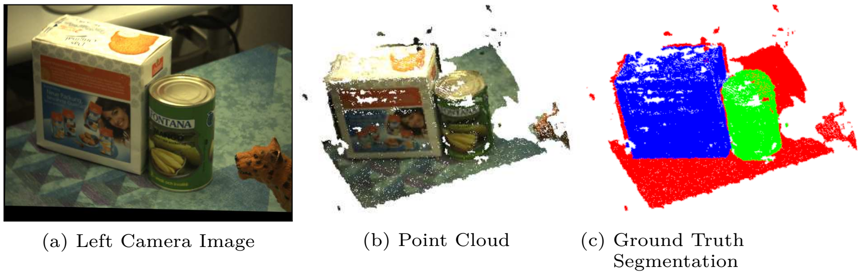


Figure 5.5: Entropy-based Detection of Under-segmentation. Left) Performance of Single and Subgroups of Features. XYZ Scene and XYZ BB indicate the three-dimensional feature vectors consisting of $H[p_x(\mathbf{s})]$, $H[p_y(\mathbf{s})]$, $H[p_z(\mathbf{s})]$ and $H[p_{BB_z}(\mathbf{s})]$, $H[p_{BB_y}(\mathbf{s})]$, $H[p_{BB_z}(\mathbf{s})]$,respectively. Right) ROC Curve for Classifier Using a Combination of the best Features as in Equation 5.2.

or $H[p_{BB_z}(\mathbf{s})]$ is essentially random. $H[p_{BB_y}(\mathbf{s})]$ however, reaches an AUC of 0.7. This is due to the way, we compute the oriented bounding box whose y-axis will always be aligned with $\mathbf{e}_b$, the smallest eigenvector of the projection of the point cloud. That means that the entropy values along the x and z axis for correct and incorrect segments will be very similar. However, for segments containing two objects standing in front of one another, the entropy value over the y axis will be significantly different from segments containing single objects. This case can for example be observed in Figure 5.3. Also the hue of an object hypothesis reaches an AUC of 0.7 for separating correct from incorrect object hypotheses.

Figure 5.5 right) shows the ROC curve for a classifier using a feature vector as in Equation 5.2. This is the combination of the best single features: (i) entropy over the color hue histogram $H[p(c)]$, (ii) entropy over the point histogram of the bounding box $H[p_{BB}(\mathbf{v})]$, (iii) entropy over the point histogram of the fixed size box $H[p_{BB}(\mathbf{v})]$, (iv) entropy over each of the one-dimensional histograms of 3D points in slices along the three axis $H[p_x(\mathbf{s})]$, $H[p_y(\mathbf{s})]$, $H[p_z(\mathbf{s})]$ and (v) entropy over the histogram of 3D points in the slices along the y-dimension of the bounding box $H[p_{BB_y}(\mathbf{s})]$. As can be seen from the plot and the AUC=0.92 , the classifier is easily able to distinguish between correctly and incorrectly labeled examples in this dataset.


(a) Left Camera Image

(b) Point Cloud

(c) Ground Truth Segmentation

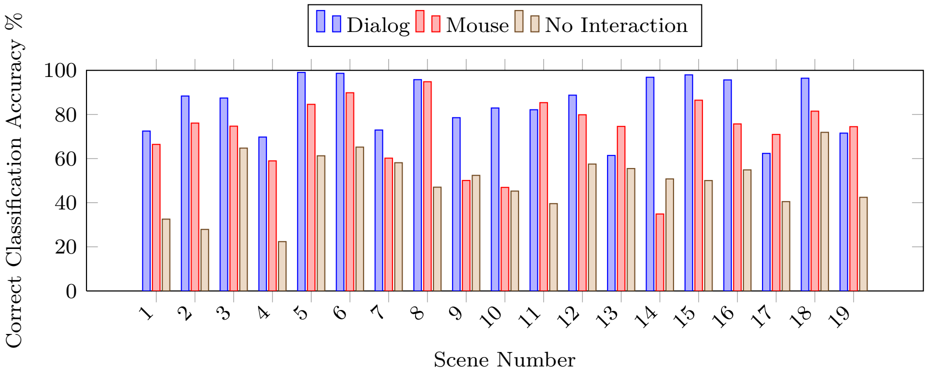
Figure 5.6: Image of Left Foveal Camera, Point Cloud and Ground Truth Labeling of Scene 2.

5.4.2 Improvement of Segmentation Quality

To validate the technique, we present experimental results designed to display its performance on the 19 scenes in the data set as displayed in Figure 5.4. They contain one incorrect segment for which we have the hand labeled ground truth segmentation. Figure 5.6 shows scene 2 as an example. We compared the resulting total segmentation accuracy before and after interaction with the human operator on the mislabeled segments.

Baseline As a baseline, we use an interactive segmentation method in which mouse clicks of the user are the seed points of the re-segmentation. In detail, the user sees an image of the part of the scene that is detected to contain a wrong object hypothesis. An example for such an image is shown in Figure 5.6a. The user is then asked to click with a mouse on the center of each object that he or she can identify. For each click position (u_i, v_i), we know the corresponding 3D point (x_i, y_i, z_i) from prior stereo reconstruction. These points are then used as seeds for re-initializing segmentation by placing a ball around them with a certain radius.

Results The graph in Figure 5.7a displays three bars for each segment. They display the total classification rate for all object points in the segment as compared to ground truth. The first bar in each set shows the performance achieved with the aid of human dialog input in the proposed framework; the second bar shows the results based on mouse clicks, while the third is without any human interaction. The proposed method achieves an increase in performance of 33.25% on average. This is attributable to the correct labeling of an object initially missing from the segmentation. A confusion matrix for a typical under-segmentation can be seen in Figure 5.7d and the resulting correction achieved through dialog interaction in Figure 5.7b. The mouse based interaction only achieves an increase of 21.26% in segmentation performance as compared to the initial segmentation.



(a) Average of approx. 33.25% increase in overall performance when dialog information is used for re-segmentation. With segmentation seed points from mouse clicks, we achieve only an average increase of 21.26%. Note that in some cases when there is quite a poor initial segmentation even with the human input the total accuracy remains low. This suggests interaction is most useful when the automated technique has at least some understanding of the scene.

		Actual		
		Bkrd.	Obj 1	Obj 2
Predicted	Bkrd.	**23706**	57	121
	Obj 1	1073	**9793**	3823
	Obj 2	743	114	**21447**
	F1:	0.96	0.79	0.90

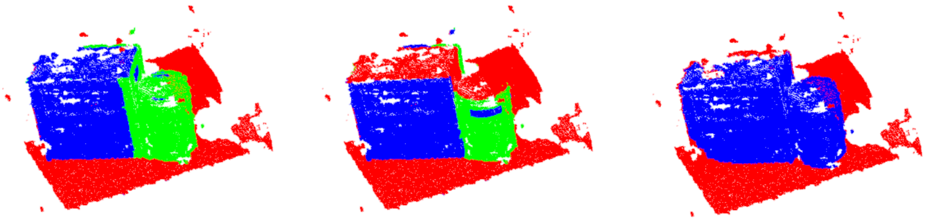
(b) Dialog CM

		Actual		
		Bkrd.	Obj 1	Obj 2
Predicted	Bkrd.	**24079**	1951	5417
	Obj 1	729	**7585**	657
	Obj 2	714	428	**19317**
	F1:	0.85	0.80	0.84

(c) Mouse CM

		Actual		
		Bkrd.	Obj 1	Obj 2
Predicted	Bkrd.	**24449**	100	1450
	Obj 1	1073	**9864**	23941
	Obj 2	0	0	**0**
	F1:	0.95	0.44	0.00

(d) No Interaction CM



(e) Dialog PC (f) Mouse PC (g) No Interaction PC

Figure 5.7: The results of a comparison to hand labeled ground truth across an aggregate set (a) and the confusion matrices (CMs) for scene 2. (b) With dialog interaction. (c) With mouse interaction. (d) Without interaction. Note the superior performance of the mouse-based over the dialogue-based re-segmentation at the object boundaries. This effect is due to the accurate placing of the seed points through the user as compared to the splitting of a potentially slightly miss-aligned bounding box. However, the mouse based interaction fails at including whole objects into one segment.

To clarify this result, Figure 5.7e and 5.7f show the re-segmented point clouds for Scene 2. For the dialog-based interaction, segmentation errors occur at the boundary of the two objects and are due to the inflexible splitting of the bounding box in the middle. The mouse-based method fails to include the whole objects into the new segments. This is because in this simple seeding method, the initial segmentation is not being re-used for the new object models. Other possibilities for re-seeding would be to group together points from the initial segmentation that are closest to one of the click positions. We would expect the segmentation quality of the dialog and mouse based re-segmentation methods to be comparable.

5.5 Discussion

We presented a novel human-robot interaction framework for the purpose of robust visual scene understanding. A state-of-the-art computer vision method for performing scene segmentation was combined with a natural dialog system. Experiments showed that by putting a human in the loop, the robot could gain improved models of challenging 3D scenes when compared with pure bottom-up segmentation. The proposed method for dialog-based interaction outperformed the simple mouse-based interaction. Furthermore, through an entropy-based approach the interaction between human and robot could be made more efficient.

As future work, one could address the problem of how to represent objects efficiently in the working memory and label them with symbols that are convenient for a human operator. These could be attributes including positional state, dominant color and relative size enabling more complex dialogs.

Furthermore, user studies could be conducted with non-expert users to gain more insight into interaction patterns for this specific task. Specifically, it would be interesting to analyse how users describe scenes and object relations spatially. Are these descriptions dependent on viewpoint differences between the user and the robot? Which object attributes are chosen for reference resolution? Is this choice dependent on the whole scene structure?

In the scenario used in this chapter, the robot first exhaustively explores the scenes by shifting its gaze to different salient points in the scene. Then it describes to the user what it has seen. A more realistic scenario is that the robot describes the scene already during exploration. Scene understanding could then be performed iteratively. The model for distinguishing correct from incorrect object hypotheses could also be extended during interaction with the user.

To improve reference resolution, the robot could point to the parts of the scene it is currently talking about. In that context, recognition of pointing gestures performed by the user would also be of use.

6

Predicting Unknown Object Shape

Many challenging problems addressed by the robotics community are currently studied in simulation. Examples are motion planning [125, 66, 225] or grasp planning [53, 90, 153] in which the knowledge of the complete world model or of specific objects is assumed to be known. However, on a real robotic platform this assumption breaks down due to noisy sensors or occlusions.

Consider for example the point cloud in Figure 6.1 that was reconstructed with an active stereo head from a table-top scene. Due to occlusions, no information about the backside of the object or about the region behind it is available from pure stereo reconstructed data. There are *gaps* in the scene and object representations. Additionally, noise causes the observed 3D structure of an object or scene to deviate from its true shape. Although the previously mentioned planners might be applied on real robotic platforms, a mismatch between the real world and the world model is usually not taken into account explicitly.

Exceptions are reactive grasping strategies that adapt their behavior based on sensor data generated during execution time [78, 98, 99].

In this section, we consider the problem of picking up objects from a table-top scene containing several objects. We assume that the robot has reconstructed a point cloud from the scene using the approach from Section 2.2.1. Additionally, we

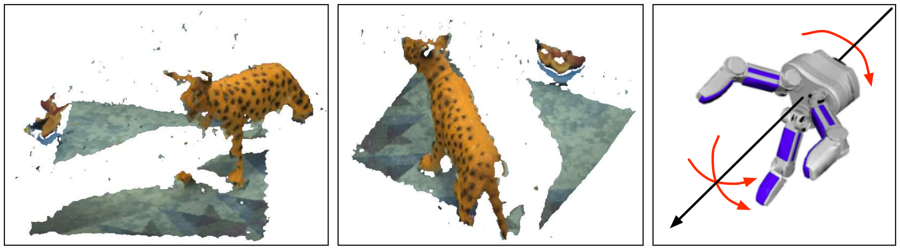


Figure 6.1: How can a suitable grasp pose be determined based on an incomplete point cloud? Left and Middle) Example views of a raw stereo reconstructed point cloud. Right) Model of the Schunk Hand with some of its degrees of freedom.

assume that this point cloud is correctly segmented which could be ensured through the additional use of the human-robot dialog framework presented in the previous chapter.

To accomplish stable grasping executed through collision-free motion, we follow the idea of filling in the gaps in the scene representation through predicting the full shape of objects. We make use of the observation that many, especially man-made objects, possess one or more symmetries. Given this, we can provide the simulator with an estimated complete world model under which it can plan actions or predict sensor measurements. Furthermore, we propose a method to quantify the uncertainty about each point in the completed object model. We study how much perception and manipulation can be improved by taking uncertainty explicitly into account.

The contributions presented in this chapter are (i) a quantitative evaluation of the shape prediction on real-world data containing a set of objects observed from a number of different viewpoints. This is different from related work by Thrun and Wegbreit [219] in which only qualitative results in form of point clouds are presented rather than quantitative results on polygonal meshes. (ii) We propose to estimate the most general symmetry parameters. This is different from Marton et al. [147, 145] in which prior to reconstruction, the kind of symmetry an object possesses has to be detected. It can either be box-shaped, cylindrical or have a more complex radial symmetry. Our approach is also different to Blodow et al. [36] in which only rotationally symmetry is considered. (iii) We reduce the search space for the optimal symmetry parameters through a good initialization. And (iv), we will demonstrate in Chapter 7 how prediction of object shape helps perception as well as manipulation.

6.1 Symmetry for Object Shape Prediction

Estimating the occluded and unknown part of an object has applications in many fields, e.g. 3D shape acquisition, 3D object recognition or classification. In this chapter, we are looking at this problem from the perspective of service robotics and are therefore also interested in the advantages of shape completion for collision detection and grasp planning.

Psychological studies suggest that humans are able to predict the portions of a scene that are not visible to them through *controlled scene continuation* [44]. The expected structure of unobserved object parts are governed by two classes of knowledge: i) Visual evidence and ii) completion rules gained through prior visual experience. A very strong prior that exists in especially man-made objects is symmetry. This claim is supported by Marton et al. [149]. The authors analysed five different databases containing models of objects of daily use. Only around 25% of these were not describable by simple geometric primitives such as cylinders, boxes or rotationally symmetric objects. And even this comparably small set contains objects that posses planar symmetry.

Thrun and Wegbreit [219] have shown that this symmetry can be detected in partial point clouds and then exploited for shape completion. The authors developed a taxonomy of symmetries in which *planar reflection symmetry* is the most general one. It is defined as the case in which each surface point P can be uniquely associated with a second surface point Q by reflection on the opposite side of a symmetry plane. Furthermore, in a household environment, objects are commonly placed such that one of their symmetry planes is perpendicular to the supporting plane. Exceptions exist such as grocery bags, dishwashers or drawers.

Given these observations, for our table-top scenario we can make the assumption that objects commonly possess one or several planar symmetries of which one is usually positioned perpendicular to the table from which we are grasping. By making these simplifications, we can reduce the search space for the pose of this symmetry plane significantly. As it will be shown in the experimental section, our method produces valid approximations of the true object shape in very different viewpoints and for varying levels of symmetry.

6.1.1 Detecting Planar Symmetry

Since we assume the symmetry plane to be perpendicular to the table plane, the search for its pose is reduced to a search for a line in the 2 1/2D projection of the partial object point cloud. This line has 3 degrees of freedom (DoF): the 2D position of its center and its orientation. We follow a generate-and-test scheme in which we create a number of hypotheses for these three parameters and determine the *plausibility* of the resulting mirrored point cloud based on visibility constraints. Intuitively, the fewer points of a mirrored point cloud are positioned in visible free space the more plausible is this point cloud. This is visualized in Figure 6.3 and will be detailed in Section 6.1.1.4.

We bootstrap the parameter search by initializing it with the major or minor eigenvector $\mathbf{e}_a$ or $\mathbf{e}_b$ of the projected point cloud. As will be shown in Section 6.2, this usually yields a good first approximation. Further symmetry plane hypotheses are generated from this starting point by varying the orientation and position of the eigenvectors as outlined in Figure 6.2. We therefore further reduce the search space to 2 DoF. In the following, we will describe the details of this search.

6.1.1.1 Initial Plane Hypothesis

The first hypothesis for the symmetry axis is either one of the two eigenvectors of the projected point cloud. Our goal is to predict the unseen object part. We therefore make the choice dependent on the inverse viewing direction $\mathbf{v}$: the eigenvector that is most perpendicular to $\mathbf{v}$ is used as the symmetry plane $\mathbf{s}$.

$$\mathbf{s} = \begin{cases} \mathbf{e}_a \texttt{ if } \mathbf{e}_a \cdot \mathbf{v} \leq \mathbf{e}_b \cdot \mathbf{v} \\ \mathbf{e}_b \texttt{ if } \mathbf{e}_a \cdot \mathbf{v} > \mathbf{e}_b \cdot \mathbf{v} \end{cases} \tag{6.1}$$

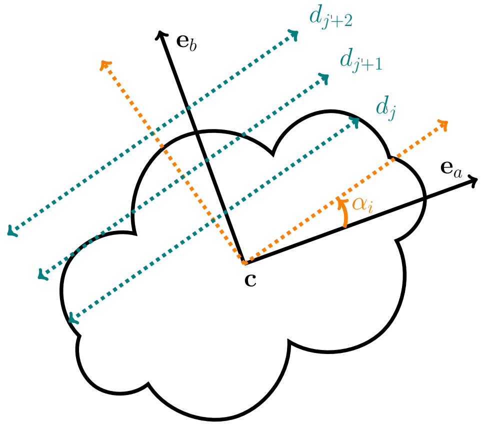


Figure 6.2: A set of hypotheses for the position and orientation of the symmetry plane. $\mathbf{e}_a$ and $\mathbf{e}_b$ denote the eigenvectors of the projected point cloud. $\mathbf{c}$ is its center of mass. α_i denotes one of the variations of line orientation along which the best pose of the symmetry plane is searched. The blue lines translated about d_j to d_{j+2} yield three further candidates with orientation α_i.

where $\mathbf{e}_a$ and $\mathbf{e}_b$ denote the major and minor eigenvectors of the projected point cloud, respectively.

6.1.1.2 Generating a Set of Symmetry Hypothesis

Given this initial approximation of the symmetry plane $\mathbf{s}$ of the considered point cloud, we sample n line orientations α_i in the range between $-20°$ and $20°$ relative to the orientation of $\mathbf{s}$. The 2D position of these n symmetry planes is varied based on a shift d_j in m discrete steps along the normal of the symmetry plane. Together this yields a set $\mathcal{S} = \{\mathbf{s}_{(1,1)} \cdots \mathbf{s}_{(1,j)} \cdots \mathbf{s}_{(i,j)} \cdots \mathbf{s}_{(n,m)}\}$ of $n \times m$ hypotheses.

6.1.1.3 Mirroring the Point Cloud

Let us denote the original point cloud as $\mathcal{P}$ containing points $\mathbf{p}$. Then given some symmetry plane parameters $s_{(i,j)}$, the mirrored point cloud $\mathcal{Q}_{(i,j)}$ with points $\mathbf{q}$ is determined as follows:

$$\mathbf{q} = \mathbf{R}_{\alpha_i}^{-1}(-\mathbf{R}_{\alpha_i}(\mathbf{p} + \mathbf{d}_j - \mathbf{c})) + \mathbf{c} \tag{6.2}$$

with $\mathbf{R}_{\alpha_i}$ denoting the rotation matrix corresponding to α_i. $\mathbf{d}_j$ is the translation vector formed from the shift d_j .

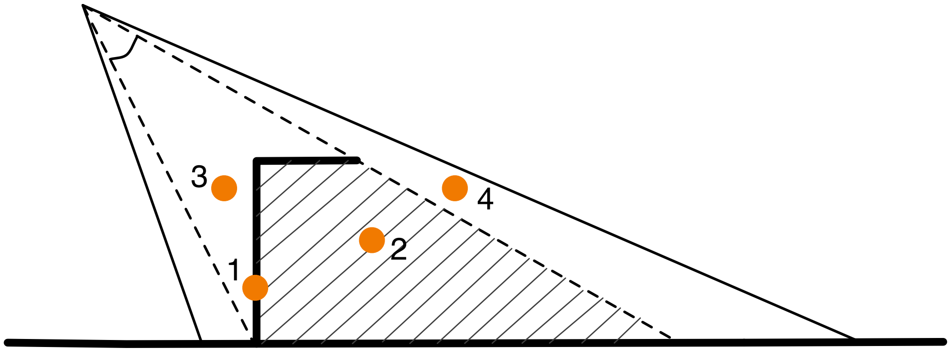


Figure 6.3: Free, occupied and unknown areas. 1. Mirrored point on original cloud. 2. Mirrored point in occluded space. 3. Mirrored point in front of segmented cloud. 4. Mirrored point in free space but not in front of segmented cloud.

6.1.1.4 Computing the Visibility Score

The simplest source of information about visibility constraints are the binary 2D segmentation masks $\mathcal{O}$ that separate an object from the background in combination with the viewpoint of the camera. The mirroring process aims at adding points $\mathbf{q}$ to the scene that were unseen from the original viewpoints.

Let us assume a mirrored point cloud $\mathcal{Q}_{(i,j)}$ has been generated, then there are four cases for where a reflected point $\mathbf{q}$ can be positioned. These cases are visualized in Figure 6.3. i) If it coincides with a point in $\mathcal{P}$ (the original point cloud), then it supports the corresponding symmetry hypothesis. ii) If $\mathbf{q}$ falls into previously occluded space, it provides information about potential surfaces not visible from the original viewpoint. And finally iii) and iv), if $\mathbf{q}$ is positioned into the free space that has been visible before, then it contradicts the symmetry hypothesis. For selecting the best mirrored point cloud, we want to maximise the number of points that support the symmetry hypothesis while minimizing the number of points contradicting it.

Using visibility contraints, we can compute a score for each point $\mathbf{q}$ in a mirrored point cloud $\mathcal{Q}_{(i,j)}$ but also for the corresponding symmetry plane hypothesis itself. We back project each mirrored point cloud $\mathcal{Q}_{(i,j)}$ into the original image giving us a set of pixels. For all these pixels $\mathbf{u}_q$, we search its nearest neighbor pixel $\mathbf{u}_p$ that lies inside the original object segmentation mask $\mathcal{O}$ and is a projection of a point in $\mathcal{P}$. For each pair $(\mathbf{u}_q, \mathbf{u}_p)$ we can then compute the absolute value of the depth difference δ between their corresponding 3D points $\mathbf{q}$ and $\mathbf{p}$.

Plausibility of a Point As plausibility $\texttt{plaus}(\mathbf{q})$ of a point $\mathbf{q}$, we define its probability to belong to the original object. This can be computed dependent on to which case in Figure 6.3 it belongs and how big the depth divergence is to its nearest neighbor in $\mathcal{P}$. For all points $\mathbf{q}$ falling into the occluded space or coincide with the segmented object as in case one and two of Figure 6.3, we compute

$$\texttt{plaus}_1(\mathbf{q}) = \frac{\delta \times (0.5 - \texttt{plaus}_{max})}{\delta_{max}} + \texttt{plaus}_{max} \tag{6.3}$$

where $\texttt{plaus}_{max}$ refers to the maximum plausibility and δ_{max} to the maximum depth divergence. Each point in the occluded space will have a minimum probability to belong to the real object of 0.5. The higher the depth divergence δ of a point to its nearest neighbor, the lower will be its plausibility. This relationship is defined as a linear fall-off.

For all points $\mathbf{q}$ falling into previously visible space (case three and four in Figure 6.3), we similarly compute their plausibility as

$$\texttt{plaus}_2(\mathbf{q}) = \frac{-\delta \times \texttt{plaus}_{max}}{\delta_{max}} + \texttt{plaus}_{max}. \tag{6.4}$$

In contrast to Equation 6.3, Equation 6.4 can assign 0 as the minimum probability of a point to belong to the object. All original points $\mathbf{p}$ of the point cloud will have $\texttt{plaus}_{max}$ assigned which can be lower than 1 to account for noise.

Plausibility of a whole point cloud The plausibility score $p_{(i,j)}$ for the whole mirrored point cloud $\mathcal{Q}_{(i,j)}$, i.e., the symmetry plane hypothesis $\mathbf{s}_{(i,j)}$ consists of two parts each reflecting the third or fourth case as visualized in Figure 6.3. Let $\mathbf{q} \notin \mathcal{O}$ refer to all points whose projection does not fall into the original segmentation mask $\mathcal{O}$. And let $\mathbf{q} \in \mathcal{O}$ refer to the set of points whose projections falls into $\mathcal{O}$. Then $p_{(i,j)}$ is computed as the sum of expected values of the plausibilities in these two sets:

$$p_{(i,j)} = \mathbb{E}_{\mathbf{q}\notin\mathcal{O}}\left[\texttt{plaus}_2(\mathbf{q})\right] + \mathbb{E}_{\mathbf{q}\in\mathcal{O}}[\texttt{plaus}_2(\mathbf{q})]. \tag{6.5}$$

In case the plane parameters are chosen such that there is a large overlap between the original point cloud $\mathcal{P}$ and the mirrored cloud $\mathcal{Q}_{(i,j)}$, the second part of Equation 6.5 will be relatively large compared to the first part. The bigger d_j, i.e. shift of the symmetry plane as visualized in Figure 6.2, $\mathbb{E}_{\mathbf{q}\in\mathcal{O}}\left[\texttt{plaus}_2(\mathbf{q})\right]$ will increase and $\mathbb{E}_{\mathbf{q}\notin\mathcal{O}}[\texttt{plaus}_2(\mathbf{q})]$ decrease. We are searching for symmetry plane parameters $\hat{\mathbf{s}}_{(i,j)}$ that produce a mirrored point cloud with a global maximum in the space of the plausibilities scores

$$\mathbf{s}^*_{(i,j)} = \arg\max_{\mathcal{S}} p_{(i,j)}. \tag{6.6}$$

The corresponding $\mathcal{Q}_{(i,j)}$ will have the smallest amount of points contradicting the symmetry hypothesis and therefore is of a maximum plausibility.

6.1.2 Surface Approximation of Completed Point Clouds

After the prediction of the backside of an object point cloud, we want to create a surface mesh approximation to support grasp planning and collision detection.

We use Poisson surface reconstruction for this purpose proposed by Kazhdan et al. [113]. This method is a global method considering all data at once. It produces smooth watertight surfaces that robustly approximate noisy data.

6.1.2.1 Introduction to Poisson Surface Reconstruction

Poisson surface reconstruction approaches the problem by computing an indicator function $\mathcal{X}$ that takes the value 1 at points inside the object and 0 outside of it. The reconstructed surface then equals an appropriate iso-surface. The key insight used for formulating surface reconstruction as a poisson problem is that there is a close relationship between a set of oriented points sampled from the real surface and $\mathcal{X}$. Specifically, the gradient $\nabla\mathcal{X}$ of the indicator function is 0 almost everywhere except from close to the surface where it equals the inward surface normals. The set of oriented points $\mathcal{S}$ can therefore be seen as samples from $\nabla\mathcal{X}$. Computing the indicator function can then be formulated as finding $\mathcal{X}$ whose gradient best approximates a vector field $\vec{\mathcal{V}}$ as defined by the points in $\mathcal{S}$.

$$\hat{\mathcal{X}} = \min_{\mathcal{X}} ||\nabla\mathcal{X} - \vec{\mathcal{V}}|| \tag{6.7}$$

To convert this into a standard poisson problem, Kazhdan et al. [113] propose to find the least-squares approximate solution of the indicator function $\mathcal{X}$ whose second derivative equals the first derivative of the vector field $\vec{\mathcal{V}}$.

$$\Delta\mathcal{X} \equiv \nabla\nabla\mathcal{X} = \nabla\vec{\mathcal{V}} \tag{6.8}$$

For details on how this is efficiently solved, we refer to [113]. Here, we will concentrate on the aspects that are relevant to this thesis.

Given point samples $\mathcal{S}$ and a desired tree depth D, an adaptive octree is defined in which each sample falls into a leaf node at D. A function $f_o(\cdot)$ is associated with each node o in that tree. It maps a sample point $\mathbf{q}$ from the set $\mathcal{S}$ to a scalar value such that the indicator function can be precisely and efficiently evaluated near the sample points. In [113] f_o is chosen to be an approximation of a unit-variance Gaussian centered about the node o and stretched by its size.

To achieve subnode precision in the vector field, trilinear interpolation is used to distribute a sample across its nearest neighbor nodes at depth D.

$$\vec{\mathcal{V}}(\mathbf{q}) \equiv \sum_{\mathbf{s}\in\mathcal{S}} \sum_{o\in\mathrm{Ngbr}_D(\mathbf{s})} \alpha_{o,s} f_o(\mathbf{q})\mathbf{n}_s \tag{6.9}$$

where $\alpha_{o,s}$ are the appropriate weights for interpolation and $\mathbf{n}_s$ the normal of the sample $\mathbf{s}$.

To choose the appropriate iso-value γ at which the iso-surface $\delta\tilde{\mathcal{M}}$ is extracted, $\mathcal{X}$ is evaluated at all the samples from set $\mathcal{S}$. The average of the resulting values is then used as the iso-value.

$$\delta\tilde{\mathcal{M}} \equiv \{\mathbf{q} \in \mathbb{R}^3 | \tilde{\mathcal{X}}(\mathbf{q}) = \gamma\} \tag{6.10}$$

with

$$\gamma = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathcal{X}}(\mathbf{s}). \tag{6.11}$$

$\delta\tilde{\mathcal{M}}$ is then extracted by using marching cubes adapted to octrees as proposed in Lorensen and Cline [138] and Kazhdan et al. [113].

6.1.2.2 Taking Non-Uniform Sampling into Account

The aforementioned formulation assumes that samples are uniformly distributed. However, this case seldom applies to real-world situations. In our case, where point clouds are extracted through stereo vision, sampling is non-uniform due to for example textureless areas on the real objects.

Kazhdan et al. [113] propose to scale the magnitude as well as the width of the kernel associated with each sample point according to the local sampling density. Through that approach, areas of high sampling density will maintain sharp features in their surface approximation while areas of sparse sampling will be characterized by smoothly fitting surfaces. The local sampling density $w_{\hat{D}}(\mathbf{q})$ at a point $\mathbf{q}$ is computed as the sum of the node functions of the nearest neighbor nodes at a specific depth $\hat{D} \leq D$

$$w_{\hat{D}}(\mathbf{q}) \equiv \sum_{\mathbf{s} \in \mathcal{S}} \sum_{o \in \mathrm{Ngbr}_{\hat{D}}(s)} \alpha_{o,s} f_o(\mathbf{q}). \tag{6.12}$$

$w_{\hat{D}}(\mathbf{q})$ is then used to modify Equation 6.9 such that each sample point $\mathbf{q}$ contributes proportional to the surface area associated with it. Since the local sampling density is inversely related to the surface area, then

$$\vec{\mathcal{V}}(\mathbf{q}) \equiv \sum_{\mathbf{s} \in \mathcal{S}} \frac{1}{w_{\hat{D}}(\mathbf{q})} \sum_{o \in \mathrm{Ngbr}_{\mathrm{Depth}(s)}(\mathbf{s})} \alpha_{o,s} f_o(\mathbf{q}) \mathbf{n}_s. \tag{6.13}$$

By interpolating the vector field over the neighbors of $\mathbf{s}$ at depth

$$\mathrm{Depth}(\mathbf{s}) \equiv \min(D, D + \log_4(w_{\hat{D}}(\mathbf{s})/W)) \tag{6.14}$$

not only the magnitude of the smoothing filter is adapted. Also its width is chosen proportional to the radius of the associated surface patch of point $\mathbf{s}$. W refers to the average sampling density of all samples.

Similarly, Equation 6.11 is adapted such that points with a higher local sampling density have a higher influence on the choice of the isovalue:

$$\gamma = \frac{\sum_{\mathbf{s} \in \mathcal{S}} w_{\hat{D}}(\mathbf{s}) \tilde{\mathcal{X}}(\mathbf{s})}{\sum_{\mathbf{s} \in \mathcal{S}} w_{\hat{D}}(\mathbf{s})}. \tag{6.15}$$

6.1.2.3 Plausibility for Improved Robustness of the Surface Reconstruction

Kazhdan et al. [113] propose to adapt their approach to take confidence value of the sampling points into account which is implemented in [112]. In this thesis, we want to show that different prediction mechanisms are useful for improving perception but also manipulation and grasping. Additionally, we have already shown in Section 4 that having an idea about the uncertainty in a prediction helps to efficiently explore a scene. In this section, we want to show that using the probability of a point in the mirrored point cloud to be from the original object improves the robustness of surface reconstruction. This probability is defined as the plausibility $\texttt{plaus}(\mathbf{q})$ of a point $\mathbf{q}$ according to visibility constraints as in Equation 6.3 and 6.4.

The intuition behind the following adaptation of the poisson equations is that sample points with low plausibility do not need to be represented with fine detail. Equation 6.13 is therefore adapted as follows

$$\vec{\mathcal{V}}(\mathbf{q}) \equiv \sum_{\mathbf{s}\in\mathcal{S}} \frac{1}{w_{\hat{D}}(\mathbf{q}) \cdot \texttt{plaus}(\mathbf{q})} \sum_{o\in \mathrm{Ngbr}_{\mathrm{Depth}(s)}(\mathbf{s})} \alpha_{o,s} f_o(\mathbf{q}) \mathbf{n}_s. \tag{6.16}$$

where

$$\mathrm{Depth}(\mathbf{s}) \equiv \min(D, D + \log_4(w_{\hat{D}}(\mathbf{s}) \cdot \texttt{plaus}(\mathbf{s})/W)) \tag{6.17}$$

Equation 6.11 is adapted as follows:

$$\gamma = \frac{\sum_{\mathbf{s}\in\mathcal{S}} w_{\hat{D}}(\mathbf{s}) \cdot \texttt{plaus}(\mathbf{s}) \tilde{\mathcal{X}}(\mathbf{s})}{\sum_{\mathbf{s}\in\mathcal{S}} w_{\hat{D}}(\mathbf{s} \cdot \texttt{plaus}(\mathbf{s}))}. \tag{6.18}$$

As will be shown in Section 6.2, this is specifically beneficial at the *seams* of the completed point cloud where old and new points overlap. These places are prone to noise and potentially contradictory normal information. New points that tend to have a lower plausibility than old points should therefore be down-weighted to achieve smooth boundaries.

6.1.2.4 Computing the Normals of the Completed Point Cloud

The point cloud data, that we use as sample set $\mathcal{S}$ is not originally oriented. To determine the normals, we use a kd-tree as proposed by Hoppe et al. [97]. A local plane fit is estimated for the k-nearest neighbors of the target point. This plane is assumed to be a local approximation of the surface at the current point. More advanced normal estimation techniques as for example by Cole et al. [54] and Mitra and Nguyen [154] have been proposed which could perhaps increase the performance of the surface reconstruction at the cost of computation speed. Following normal estimation, we ensure that the normals of the mirrored points are consistent with the mirrored viewing direction reflected across the plane of symmetry.

It should be noted while Poisson surface reconstruction is robust to some noise, it is sensitive to normal direction. We therefore filter outliers from the segmented

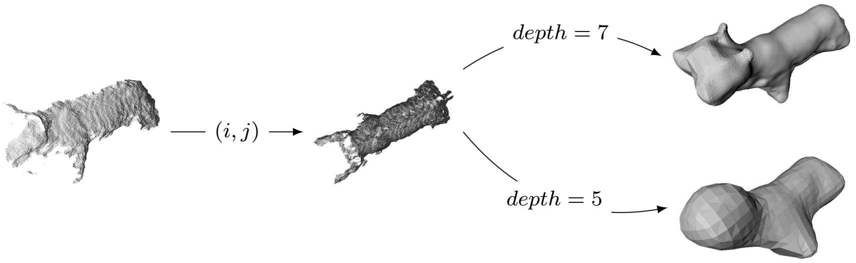


Figure 6.4: From a segmented point cloud to meshes of different resolutions. (i, j) is the index of a symmetry plane hypothesis with an angle α_i and a position d_j as visualized in Figure 6.2. The two different meshes are reconstructed based on different octree depths.

point cloud prior to the surface reconstruction step. Example results using a mirrored point cloud for different depth values D of the octree are shown in Figure 6.4.

6.2 Experiments

In this section, we will present quantitative experiments showing that the completion of incomplete object point clouds based on symmetry produces valid object models. Furthermore, we will evaluate how much the robustness of registration improves when using completed point clouds. Furthermore, we will show how the estimated complete object model helps when using a simple grasp strategy based on the center of an object.

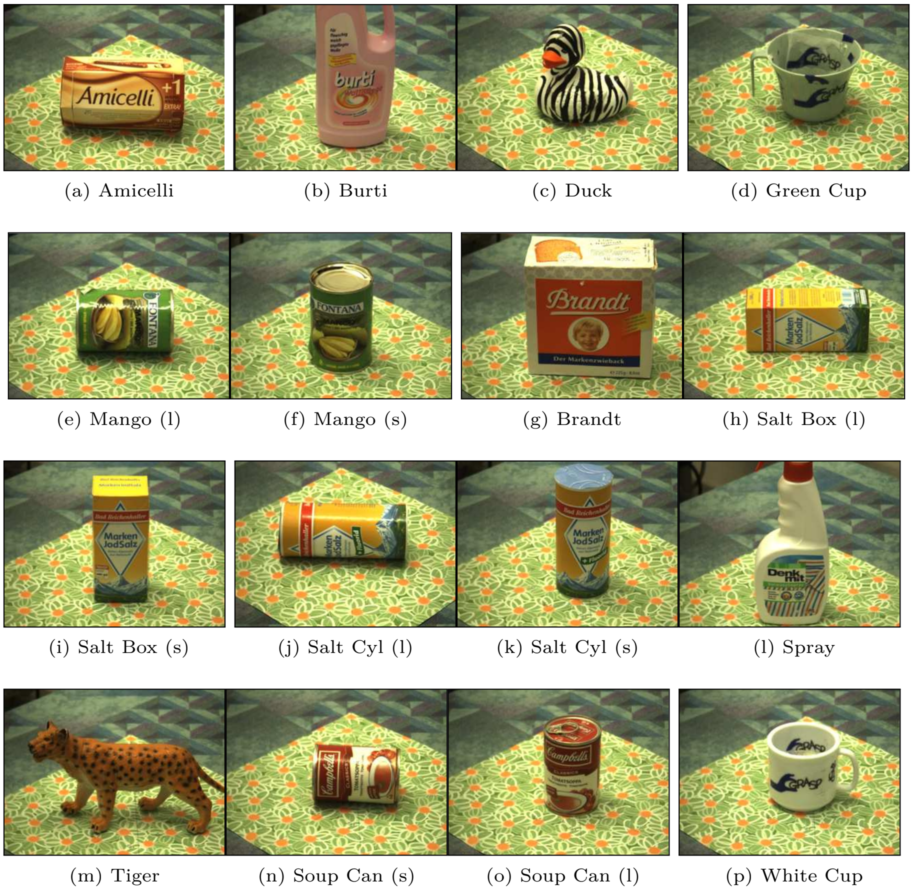
6.2.1 Evaluation of the Mesh Reconstruction

In this section, we evaluate quantitatively how accurate the surface of objects is reconstructed with the proposed method.

6.2.1.1 Dataset

The database we used for evaluating the point cloud mirroring method is shown in Figure 6.5. For each of the objects in this database, with the exception of the toy tiger and rubber duck, we have laser scan ground truth[1]. The test data was captured with the active vision system (see Section 2.2.1) and contains 12 different household or toy objects. Four of them are used both, when standing upright or lying on their side. Thereby, the database contains 16 different data sets. Each set

[1] The ground truth object models were obtained from `http://i61p109.ira.uka.de/ObjectModelsWebUI/`



(a) Amicelli (b) Burti (c) Duck (d) Green Cup

(e) Mango (l) (f) Mango (s) (g) Brandt (h) Salt Box (l)

(i) Salt Box (s) (j) Salt Cyl (l) (k) Salt Cyl (s) (l) Spray

(m) Tiger (n) Soup Can (s) (o) Soup Can (l) (p) White Cup

Figure 6.5: The 12 Objects of the Mesh Reconstruction Database in partially varying poses yielding 16 Data Sets. Objects are shown in their 0° position, i.e, with their longest dimension when projected on the table parallel to the image plane. Ground truth meshes are existing for all objects except the toy tiger and the rubber duck.

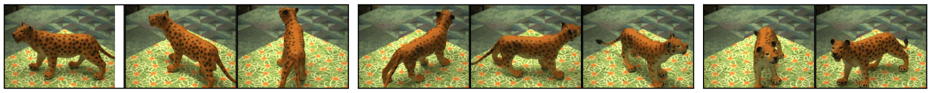


Figure 6.6: One of the Datasets from Figure 6.5 shown in Orientation 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°.

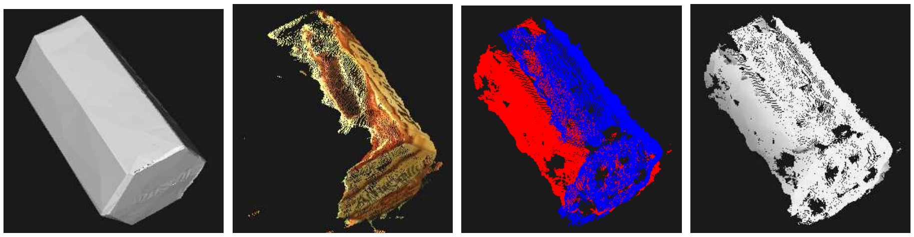


Figure 6.7: Left) Ground Truth Mesh. Middle Left) Original Point Cloud. Middle Right) Mirrored Cloud with Original Points in Blue and Additional Points in Red. Right) Mirrored Cloud with Plausibility Information Mapped to Gray Scale.

contains 8 stereo images showing the object in one of the following orientations: 0°, 45°, 90°, 135°, 180°, 225°, 270° or 315°. An example for the toy tiger is shown in Figure 6.6. As an orientation reference we used the longest object dimension when projected down to the table. 0° then means that this reference axis is parallel to the image plane of the stereo camera. This can be observed in Figure 6.5 in which all objects are shown in their 0° pose. Therefore, the database contains 128 stereo images along with their point clouds.

From all point clouds, we reconstructed the complete meshes based on the methods described in Section 6.1.2 with and without taking plausibility information into account. As octree depth parameter of the Poisson surface reconstruction we use 5. In our previous work [4], we also used a tree depth of 7 and did not find a significant difference in performance. By limiting this parameter, we enable mesh reconstruction in near real-time. With a tree depth of 5, the meshes are more coarse and blob-like but less sensitive to noise in the point cloud and normal estimation. With a depth of 7, the reconstructed surface is closer to the original point set. However, outliers strongly affect the mesh shape and it is more sensitive to noise.

To obtain the ground truth pose for each item in the database, we applied the technique proposed in [165]. It allows to register ground truth object meshes to the incomplete point clouds. We manually inspected the results to ensure correctness.

Figure 6.7 shows an example point from our dataset along with the aligned ground truth mesh, the completed point cloud and the plausibility values.

6.2.1.2 Baseline

As a baseline, we generated a mesh only based on the original point cloud. To do this, we applied a Delaunay triangulation as implemented in [43] to the projection of a uniformly sampled subset of 500 points from the original point cloud. Spurious edges are filtered based on their length in 2D and 3D. Furthermore, we extract the outer contour edges of this triangulation and span triangles between them which produces a watertight mesh. Figure 6.8 shows an example Delaunay graph with color-coded edges. Figure 6.9 shows example results of this Delaunay-based mesh

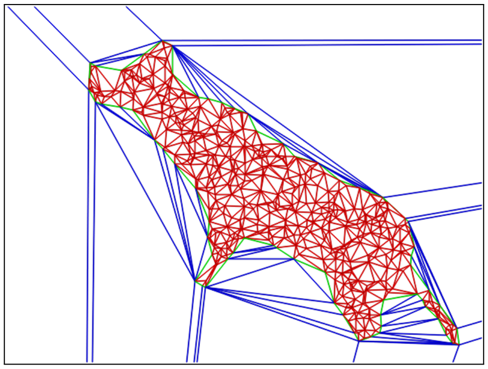


Figure 6.8: Delaunay graph as the basis for the baseline mesh reconstruction. Red edges belong to the mesh. Green edges are contour edges that are additionally triangulated separately to create a mesh backside. Blue edges are filtered out based on length and connectivity to the outer triangle.

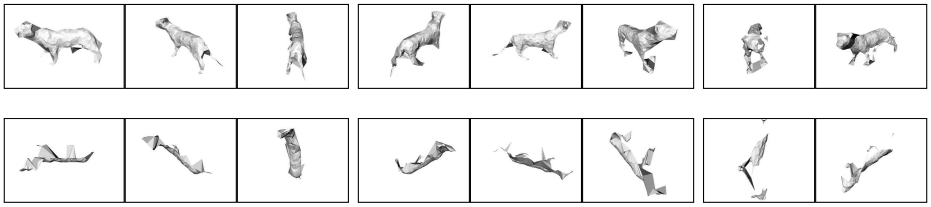


Figure 6.9: Delaunay based Meshes of Toy Tiger. Top: Front View. Bottom: Top View.

reconstruction.

6.2.1.3 Mesh Deviation Metric

To assess the deviation of the Delaunay meshes and the mirrored meshes from the ground truth we use MeshDev [189]. As a metric we evaluated geometric deviation, i.e., the distance between each point on the reference mesh to the nearest neighbor on the other mesh. We applied the uniform sampling of the surface of the reference mesh as proposed in [189] to calculate this deviation.

In Figure 6.10, meshes are visualized that were reconstructed using the baseline approach, the Poisson reconstruction without plausibility and with plausibility information. Figure 6.11 shows the heat maps of the geometric deviation between the meshes in Figure 6.10 and the ground truth mesh in Figure 6.7. In [4], we compared the performance of the different mesh reconstruction results using the mean and variance over the geometric deviation histogram. The resulting Gaussians are

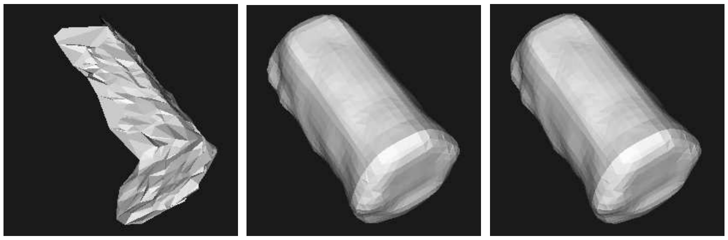


Figure 6.10: Reconstructed Meshes. Left) Delaunay based. Middle) Poisson reconstruction on uniformly weighted points. Right) Poisson reconstruction on points weighted with plausibility.

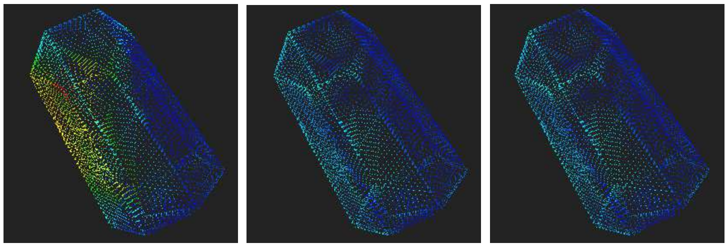


Figure 6.11: Heat Map of Geometric Deviation to Ground Truth Mesh.

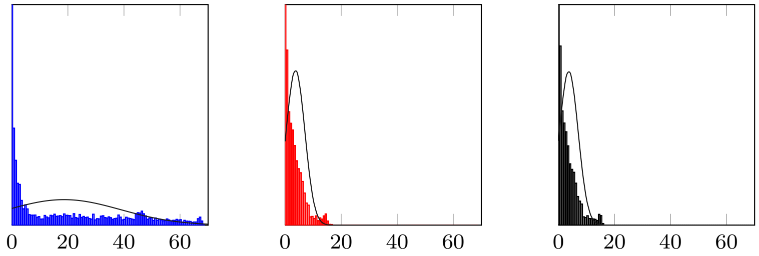


Figure 6.12: Gaussian distribution fit to histograms of geometric deviation [mm].

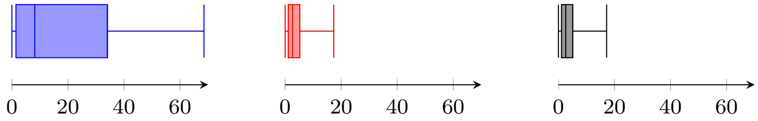


Figure 6.13: Median, First and Third Quartille of Histograms.

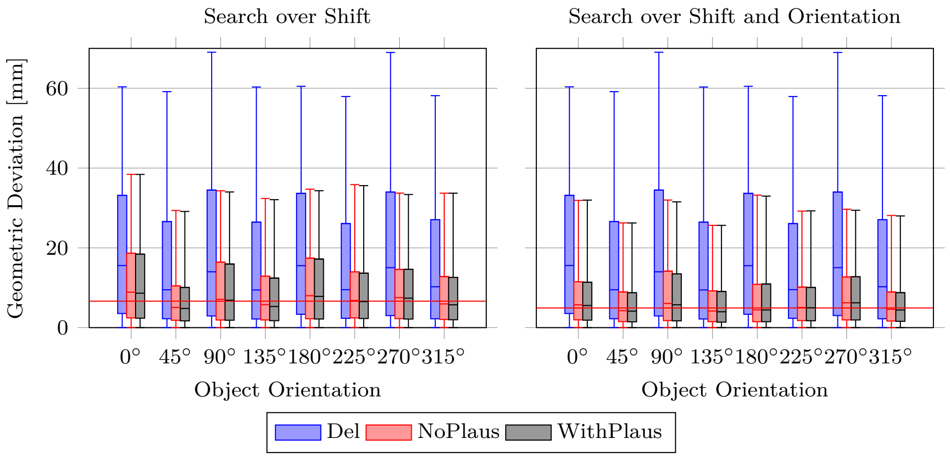


Figure 6.14: Evaluation of the Deviation between the Ground Truth Mesh and i) Mesh based on 3D Point Cloud only (*Del*), ii) Mesh based on mirroring and Poisson Surface Reconstruction without using plausibility information(*WOPlaus*) or iii) with using plausibility information(*WPlaus*). The red line indicates the average median deviation over all orientations.

visualized in Figure 6.12. It clearly shows that this model does not sufficiently fit the data. To better visualise the distribution of the data, we use a *box-and-whiskers* plot as shown in Figure 6.13. The box contains the first, second and third quartile while the whiskers show the minimum and maximum of the distribution.

6.2.1.4 Results

Geometric Deviation of Reconstructed Meshes Figure 6.14 shows the box-and-whiskers plots of the mesh deviation between the ground truth mesh and the reconstructed meshes for all object orientations over all objects. The left and right graph present results for two different categories of symmetry planes. On the left, only those with the same orientation as the selected eigenvector were considered. The planes were translated along the normal of the eigenvector. On the right of Figure 6.14, additional hypotheses with varying orientation are considered.

On average, the mirrored point clouds are always closer to the ground truth than the Delaunay-based meshes. Furthermore, when searching over different positions and orientations of the symmetry plane, the average median deviation is lower ($4.93mm$) than when only taking shifts into account ($6.64mm$).

Figure 6.15 shows the same error measure for each object independently averaged over all its orientations. The deviation measure is not normalized to the overall object size and shows deviation in millimeters. For bigger objects, like the

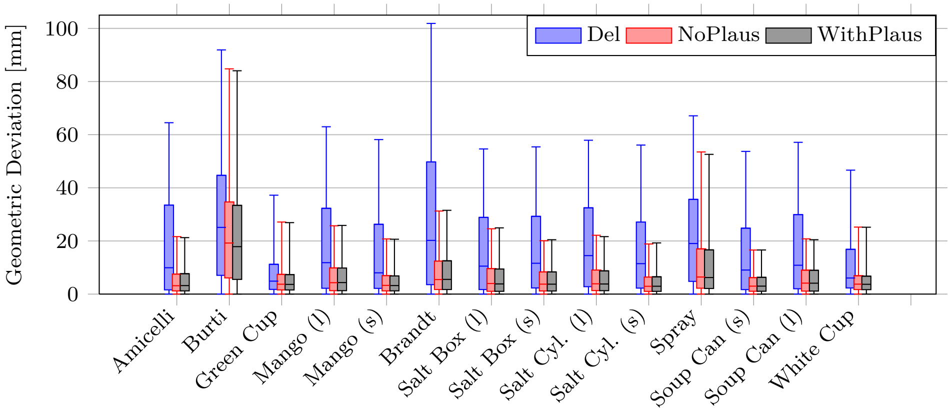
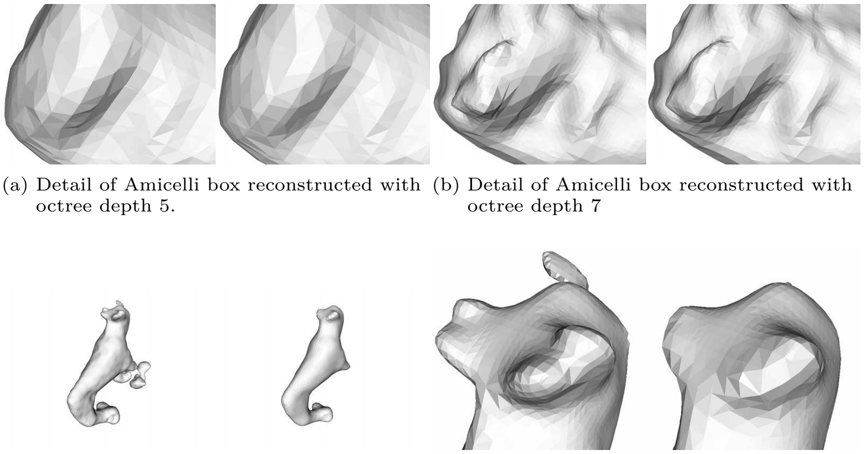


Figure 6.15: Evaluation of the Deviation between the Ground Truth Mesh and i) Mesh based on 3D Point Cloud only (*Del*), ii) Mesh based on mirroring and Poisson Surface Reconstruction without using plausibility information(*WOPlaus*) or iii) with using plausibility information(*WPlaus*).

Brandt box or the *Burti* and *Spray* bottle, the median geometric deviation between the Delaunay meshes and the ground truth exceeds 20mm. The mirroring yields a significant improvement for most objects. The green and white cup pose a challenging problem to the Poisson surface reconstruction because they are hollow. Modeling the void is difficult due to viewpoint limitations. On the other hand, holes in the point clouds due to non-uniform texture are usually closed by the surface reconstruction. For the *Burti* bottle, the original point clouds are very sparse due to this object being only lightly textured. This is reflected by the very high median of the geometric deviation of these object meshes shown in Figure 6.15. This indicates that given input of very low quality, the proposed completion algorithm cannot achieve a significantly better reconstruction of the whole object.

Plausibility for Surface Reconstruction Figure 6.14 and 6.15 also compare the geometric deviation for meshes being reconstructed with and without taking the point-wise plausibility information into account. They show that on average there is no significant difference between them regarding the median of geometric deviation. However for some example meshes, we can observe a significantly better surface reconstruction when using the plausibility information. In Figure 6.16, show some of these examples and illustrate the differences between the two reconstruction methods. Details are shown of the *seams* between the original and the mirrored point cloud. Figure 6.16a, 6.16b and 6.16d show specifically, how the lower plausibility values of the mirrored points at these regions lead to stronger local smoothing of the mesh. Figure 6.16c shows how using plausibility informa-



(a) Detail of Amicelli box reconstructed with octree depth 5.

(b) Detail of Amicelli box reconstructed with octree depth 7

(c) Reconstructed tiger with octree depth 7

(d) Details of reconstructed tiger head.

Figure 6.16: Comparison of Poisson reconstruction results of mirrored point cloud with and without using plausibility information. Left) Without Plausibility. Right) With Plausibility. Direct comparison shows that at places where mirrored cloud consists of contradicting oriented points, including plausibility information results in coarser meshes at those uncertain places.

tion for the choice of the isovalue can result in a more plausible mesh that does not comply with potential noise introduced by the point cloud completion. The details of the meshes suggest that fewer faces should be used for the smoother surfaces reconstructed using plausibility information. As shown in Figure 6.17, there is however no significant difference in the number of faces between the two methods. This suggests that the meshes differ only marginally at small parts as for example at the seams between the original and mirrored point clouds.

Plausibility as Quality Indicator Since, we do not have the ground truth models available for the toy tiger and the rubber duck, we show their reconstructed meshes with tree depth 7 in Figure 6.18. The overall shape of the quite irregular toy objects is well reconstructed. However, because of the complexity of the objects if an incorrect mirroring plane is chosen, we obtain toy animals with either two heads or two tails. In such cases, there are strong violations of the visibility constraints. Therefore, these symmetry plane hypotheses will have a low plausibility value as computed in Equation 6.5. By thresholding this value, we should be able to remove these erroneous hypotheses.

Figure 6.19 shows a scatterplot of the best mirrored point clouds for each object and angle combination. Their overall plausibility is plotted against the median

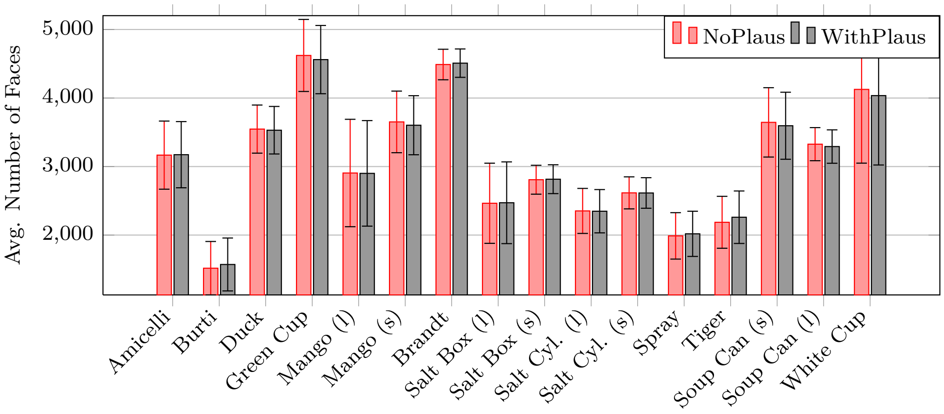


Figure 6.17: Comparison of the number of faces in a mesh reconstructed with and without considering plausibility information. Mean and standard deviation are shown for one object in the dataset averaged over all orientations.

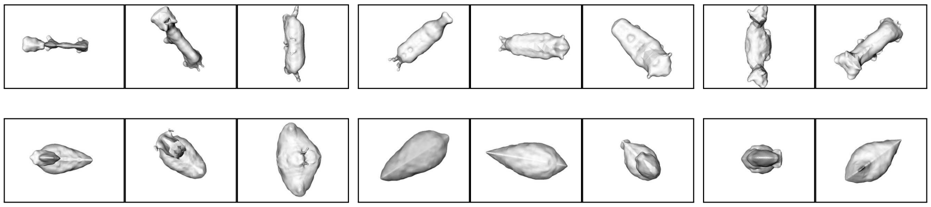


Figure 6.18: Meshes based on mirroring using only shifts of the symmetry plane. First Row: Toy Tiger. Second Row: Rubber Duck.

of the normalized geometric deviation from the ground truth meshes. We used Bayesian line fitting as described in Bishop [31] to reveal the inverse relation of plausibility to geometric deviation: The lower the plausibility of a mirrored point cloud, the higher the geometric deviation from the ground truth. This shows, that thresholding plausibility can be used to reject completions of low quality.

6.3 Discussion

In this chapter, we proposed a method that estimates complete object models from partial views. No knowledge about the specific objects is assumed for this task. Instead, we introduce a symmetry prior. We validated the accuracy of the completed models using laser scan ground truth. The results show effectiveness of the technique on a variety of household objects in table top environments.

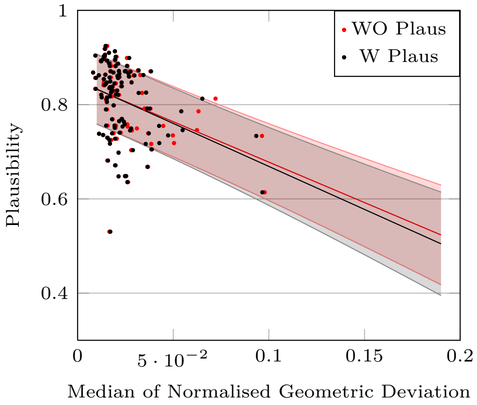


Figure 6.19: Scatterplot of the best mirrored point clouds for each object and angle combination showing their plausibility against the median of the normalized geometric deviation from the ground truth meshes. Line fit to this data in a Bayesian manner reveals the inverse relation of plausibility to geometric deviation: The lower the plausibility of a mirrored point cloud, the higher the geometric deviation from the ground truth.

We argue that the proposed object prediction technique is another step towards bridging the gap between simulation and the real world. This will be further discussed in Section 7.3 where we show how the predicted meshes can provide grasp and motion planners with more accurate object models.

As described earlier, we reduced the search space for the optimal symmetry parameters by making the assumption that objects are standing on a table top with the symmetry plane perpendicular to the table. This reduces the search for the position and orientation of a plane to the 3 DoF of a line on the table. By furthermore initializing the search from the eigenvectors of the projection of the object point cloud, only an orientation and shift of the line has to be considered. These restrictions limit the possible object poses under which we can estimate their full shape. However, the generate-and-test scheme can be easily parallelized. Therefore, search in a space covering more DoF of a symmetry plane can be efficiently performed.

As mentioned by Breckon and Fisher [44], a symmetry prior can easily fail in reconstructing a surface when the object is strongly occluded. Also wrong segmentations can lead to erroneous results. The dataset on which we quantitatively evaluated the proposed method contains only single objects. We assume they are correctly segmented with only few outliers from the background or table. For this case, we have shown the effectiveness of our method. In Section 8.1.1, we will demonstrate the proposed approach in a grasping scenario with several objects on a table top. Although minor occlusions occur, the method still succeeds in providing models on which successful grasps are planned and finally executed.

However, in the presence of strong occlusions the proposed method will most likely fail to estimate the correct object surface. Among others, this is due to using the eigenvectors of the point cloud projection as an initialization for the underlying symmetry plane. This choice has been shown on the dataset to yield reasonable estimates for the orientation of the symmetry plane. However, if the point cloud represents only a minor part of the real object, then the eigenvector of this part will not be close to the optimal symmetry plane of the real object.

We see two possibilities to overcome this limitation. The first one is based on using plausibility of a symmetry plane hypothesis as an absolute quality indicator. That this is possible has been shown earlier in this chapter. If only a hypothesis can be found that strongly violates visibility constraints, i.e., has a low plausibility, it can be rejected completely by thresholding this value. Then, a different method for motion and grasp planning has to be used.

A second possibility is to keep the predicted object shape but explicitly take the uncertainty about the reconstructed surface into account for grasp and motion planning. This could be done by for example avoiding placements of the hand in which fingertip positions coincide with points of low plausibility. Another option is to use the proposed method in combination with haptic exploration such that especially uncertain parts of the estimated object surface are confirmed with the hand first.

7

Generation of Grasp Hypotheses

Autonomous grasping of objects in unstructured environments remains an open problem in the robotics community. Even if the robot achieves a good understanding of the scene with well-segmented objects, the decision on how to grasp or manipulate them depends on many factors, e.g., the identity and geometry of the object, the scene context, the embodiment of the robot or its task. The number of potentially applicable grasps to a single object is immense. In the following, we will refer to this set as *grasp candidates*. Two main challenges exist: (i) planning good grasps. This can be seen as choosing a set of promising *grasp hypotheses* among the candidates. And (ii) executing these robustly under uncertainties introduced by the perception or motor system. In Chapter 8, different approaches towards robust grasp execution will be discussed. In this chapter, we will propose approaches to generate good grasp hypotheses given different prior object knowledge. In our case, one hypothesis is defined by

- the grasping point on the object with which the *tool centre point* (TCP) should be aligned [1],
- the *approach vector* [70] which describes the 3D angle that the robot hand approaches the grasping point with and
- the wrist orientation of the robotic hand.

Although humans master grasping easily, no suitable representations of the whole process have yet been proposed in the neuro-scientific literature, making it difficult to develop robotic systems that can mimic human grasping behavior. However, there is some valuable insight. Goodale [91] proposes that the human visual system is characterized by a division into the dorsal and ventral pathway. While the dorsal stream is mainly responsible for the spatial vision targeted at extracting action-relevant visual features, the ventral stream is engaged in the task of object identification. This dissociation also suggests two different grasp choice mechanisms dependent on whether a known or unknown object is to be manipulated. Support for this can be found in behavioral studies by Borghi [38] and Creem

[1] In this thesis, the TCP is at the centre of palm of the robotic gripper.

and Proffitt [57]. The authors claim that in the case of novel objects, our actions are purely guided by affordances as introduced by Gibson [89]. In the case of known objects, semantic information (e.g., through grasp experience) is needed to grasp them appropriately according to their function. However as argued in [91, 48, 228] this division of labour is not absolute. In the case of objects that are similar to previously encountered ones, the ventral system helps the dorsal stream in the action selection process by providing information about prehensile parts along with their afforded actions.

Grasp synthesis algorithms have previously been reviewed by Sahbani et al. [198]. The authors divide the approaches into *analytical* and *empirical* methods. With analytic, they refer to all those methods that find optimal grasps according to some criteria. This can be based on geometric, kinematic or dynamic formulations as for example force or form closure. Empirical approaches on the other hand avoid computing these mathematical or physical models. Instead, a set of example grasps is sampled from the set of candidates, their quality evaluated and the best subset is chosen as the final grasp hypotheses. Kamon et al. [108] also refer to this as the *comparative* approach. Existing algorithms differ in the source of grasp candidates and the definition of their quality.

In [198] the division is made dependent on whether the method is based on object features or observation of humans during grasping. We believe that this falls short of capturing the diversity of the empirical approaches. In this thesis, we instead group existing empirical grasp synthesis algorithms according to the assumed prior knowledge about object hypotheses. The three categories are:

- *Known Objects*: These approaches consider grasping of *a-priori* known objects. Once the object is identified, the goal is to estimate its pose and retrieve a suitable grasp, e.g., from an experience database.

- *Unknown Objects*: Approaches that fall into this category commonly represent the shape of an *unknown* object and apply heuristics to reduce the number of potential grasps.

- *Familiar Objects*: These approaches try to transfer pre-existing grasp experience from similar objects. Objects can be *familiar* on different levels. Low level similarity can for example be defined in terms of shape, color or texture. Higher level similarity could be defined based on object category. A underlying assumption is that new objects similar to the old ones can be grasped in a similar way.

This kind of division is inspired by the research in the field of neuropsychology. It is suitable because the assumed prior object knowledge determines the necessary perceptual processing but also what empirical method can be used for generating grasp candidates. This will be discussed in more detail in Section 7.1. Following related work, we will propose different grasp inference methods for either known, familiar or unknown objects. Our requirements on the employed object representations are that they have to be rich enough to allow for the inference of the aforementioned grasp parameters.

7.1 Related Work

Figure 7.1 outlines the different aspects that influence the generation of grasp hypotheses for an object. In this section, we focus on the significant body of work proposing empirical grasp synthesis methods. For an in-depth review of analytical approaches, we refer to [198]. We use the division proposed in the previous section to review the related work. Given the kind of prior object knowledge, we will discuss the different object representations that are partially dependent on the kind of sensor the robot is equipped with. Furthermore, we are also interested in which empirical method is employed as outlined in Figure 7.2. There is a number of methods that learn either from humans, labeled examples or trial and error. Other methods use heuristics to prune the search space for appropriate grasp hypotheses.

There is relatively little work on task-dependent grasping. Also, embodiment is usually not in the focus of the approaches discussed here. These two aspects will therefore be mentioned but not discussed in depth.

7.1.1 Grasping Known Objects

If the robot knows the object that it has to grasp, then the problem of finding a good grasp reduces to the online estimation of the object's pose. Given this, a set of grasp hypotheses can be retrieved from an experience database and filtered by reachability.

7.1.1.1 Offline Generation of a Grasp Experience Database

In this section, we will look at approaches that set up an experience database. They differ in how candidate grasps are generated and in how their quality is evaluated. These two aspects are usually tightly connected and also depend on the object representation.

3D Mesh Models and Contact-Level Grasping The main problem in the area of grasp synthesis is the huge search space from which a *good* grasp has to be retrieved. Its size is due to the large number of hand configurations that can be applied to a given object. In the theory of contact-level grasping [161, 204] a good grasp is defined from the perspective of forces, friction and wrenches. Based on this, different criteria are defined to rate grasp configurations, e.g., force closure, dexterity, equilibrium, stability and dynamic behavior.

Several approaches in the area of grasp synthesis exist that apply these criteria to rank grasp candidates for an object with a given 3D mesh model. They differ in the heuristics they use to generate the candidates. Some of them approximate the object's shape with a constellation of primitives such as spheres, cones, cylinders and

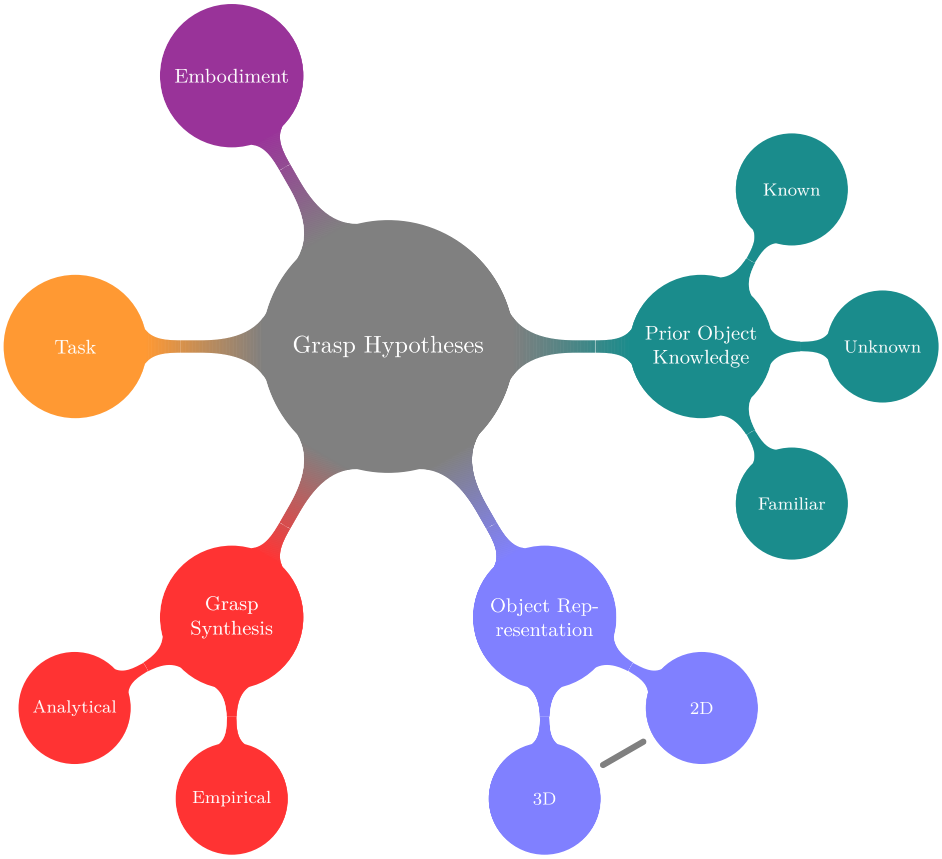


Figure 7.1: Aspects Influencing the Generation of Grasp Hypotheses.

boxes as in Miller et al. [153], Hübner and Kragic [101] and Przybylski and Asfour [179] or superquadrics (SQ) as in Goldfeder et al. [90]. These shape primitives are then used to limit the amount of candidate grasps and thus prune the search tree for finding the most stable set of grasp hypotheses. Borst et al. [39] reduce the number of candidate grasps by randomly generating a number of them dependent on the object surface and filter them with a simple heuristic. The authors show that this approach works well if the goal is not to find an optimal grasp but instead a fairly good grasp that works well for " everyday tasks". Diankov [65] proposes to sample grasp candidates dependent on the objects bounding box in conjunction with surface normals. Grasp parameters that are varied are the distance between the palm of the hand and the grasp point as well as the wrist orientation. This scheme is implemented in the OpenRave Simulator [66]. The authors find that usually a relatively small amount of 30% from all grasp samples is in force closure.

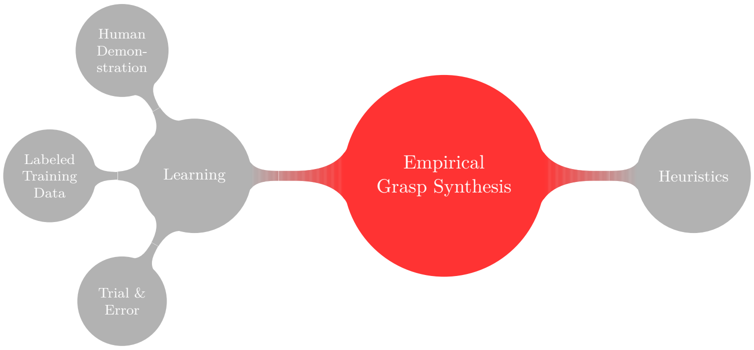


Figure 7.2: Different Approaches towards Empirical Grasp Synthesis.

All these approaches are developed and evaluated in simulation. As also claimed in [65], the biggest criticism towards ranking grasps based on force closure computed on point contacts is that relatively *fragile grasps* might be selected. A common approach to filter these, is to add noise to the grasp parameters and keep only those grasps in which a certain percentage of the neighboring candidates also yield force closure. Balasubramanian et al. [21] question classical grasp metrics in principle. The authors systematically tested a number of grasps in the real world that were stable according to classical grasp metrics. Compared to grasps planned by humans through kinesthetic learning on the same objects, these grasps underperformed significantly. The authors found that humans optimise a *skewness* metric, i.e., the divergence of alignment between hand and principal object axes.

Learning from Humans A different way to generate grasp candidates is to observe how humans grasp an object. Ciorcarlie et al. [53] exploit results from neuroscience that showed that human hand control takes place in a much lower dimension than the actual number of its degrees of freedom. This finding was applied to directly reduce the configuration space of a robotic hand to find pre-grasp postures. From these so called *eigengrasps* the system searches for stable grasps. Li and Pollard [135] treat the problem of finding a suitable grasp as a shape matching problem between the human hand and the object. The approach starts off with a database of human grasp examples. From this database, a suitable grasp is retrieved when queried with a new object. Shape features of this object are matched against the shape of the inside of the available hand postures.

Kroemer et al. [124] learn to grasp specific objects through a combination of a high-level reinforcement learner and a low level reactive grasp controller. The object is represented by a constellation of local multi-modal contour descriptors.

Four elementary grasping actions are associated to specific constellations of these features resulting in an abundance of grasp candidates. The learning process is bootstrapped through imitation learning in which a demonstrated reaching trajectory recorded with a VICON system is converted into an initial policy. Similar initialization of an object specific grasping policy is used in Pastor et al. [167] and Stulp et al. [214] through kinesthetic learning. Detry et al. [62, 64] use the same sparse object model as Kroemer et al. [124]. The set of associated grasp hypotheses is modelled as a non-parametric density functions in the space of 6D gripper poses, referred to as a *bootstrap* density. In [62] human grasp examples are used to build an object specific *empirical* grasp density from which grasp hypotheses can be sampled.

Faria et al. [76] use a multi-modal data acquisition tool to record humans grasping specific objects. Objects are represented as probabilistic volumetric maps that integrate multi-modal information such as contact regions, fixation points and tactile forces. This information is then used to annotate object parts as graspable or not. However, no results are provided on how a robot can transfer this information to new objects and to its own embodiment. Romero et al. [186] present a system for observing humans visually while they interact with an object. A grasp type and pose is recognized and mapped to different robotic hands in a fixed scheme. For validation of the approach in the simulator, 3D object models are used. Examples of real-world executions are shown. The object is not explicitly modelled. Instead, it is assumed that human and robot act on the same object in the same pose.

Learning through Self-Experience Instead of adopting a fixed set of grasp candidates given for example by a person, the following approaches try to refine them by *trial and error*. In [124, 214] reinforcement learning is used to improve an initial human demonstration. Kroemer et al. [124] uses a low-level reactive controller to perform the grasp that informs the high level controller with reward information. Stulp et al. [214] increase the robustness of their non-reactive grasping strategy by learning shape and goal parameters of the motion primitives that are used to model a full grasping action. Through this approach, the robot learns reaching trajectories and grasps that are robust against object pose uncertainties. Also Detry et al. [64], adopt an approach in which the robot learns to grasp specific objects by experience. Instead of human examples as in [62], successful grasping trials are used to build the object-specific empirical grasp density. This non-parametric density can then be used to sample grasp hypotheses.

7.1.1.2 Online Object Pose Estimation

In the previous section, we reviewed different approaches towards grasping known objects regarding their way to generate candidate grasps and rank them to form a set of grasp hypotheses. During online execution, an object has to be first recognized and its pose estimated before the offline trained grasps can be executed. Furthermore, from the set of hypotheses not all grasps might be applicable in the current scenario. An appropriate selection has to be made.

Several of the aforementioned grasp generation methods [124, 62, 64] use the probabilistic approach towards object representation and pose estimation proposed by Detry et al. [63]. As described before, grasps are either selected by sampling from densities [62, 64] or a grasp policy refined from a human demonstration is applied [124].

The following approaches use offline generated experience databases through simulation of grasps on 3D models in either GraspIt! [152] or OpenRave [66]. Berenson et al. [27] consider this experience database then as one factor in selecting an appropriate grasp. The second factor in computing the final grasp score is the whole scene geometry. The approach was demonstrated on a real robot using a motion capture system to determine the robot and the object position.

In the online stage of the method presented by Ekvall and Kragic [70], a human demonstrator wearing a magnetic tracking device is observed while manipulating a specific object. The grasp type is recognized and mapped through a fixed schema to a set of robotic hands. Given the grasp type and the hand, the best approach vector is selected from an offline trained experience database. Unlike Detry et al. [62] and Romero et al. [186], the approach vector used by the demonstrator is not adopted. Ekvall and Kragic [70] assume that the object pose is known. However, experiments are conducted in which pose error is simulated. No demonstration on a real robot is shown. Morales et al. [156] use the method proposed by Azad et al. [19] to recognize an object and estimate its pose from a monocular image. Given that information, an appropriate grasp configuration can be selected from a grasp experience database that has been acquired offline. The whole system is demonstrated on the robotic platform described in Asfour et al. [16]. Huebner et al. [102] demonstrate grasping of known objects on the same humanoid platform and the same method for object recognition and pose estimation. The offline selection of grasp hypotheses is based on a decomposition into boxes. Ciocarlie et al. [52] propose a robust grasping pipeline in which known object models are fitted to point cloud clusters using standard ICP [30]. Knowledge about the table plane and the assumption that objects are rotationally symmetric and are always standing upright helps in reducing the search space of potential object poses.

All of the aforementioned methods assume prior knowledge on a rigid 3D object model. Jared Glover [107] consider known deformable objects. For representing them, probabilistic models of their 2D shape are learnt. The objects can then be detected in monocular images of cluttered scenes even when they are partially occluded. The visible object parts serve as a basis for planning a stable grasp under consideration of the global object shape. Collet Romea et al. [55] use a combination of 2D and 3D features as an object model. The authors estimate the object's pose in a scene from a single image. The accuracy of their approach is demonstrated through a number of successful grasps on 4 different objects in somewhat cluttered scenes. Other robust methods for estimating the pose of single or multiple objects simultaneously are presented in [95, 68, 165, 166]. Although these approaches are motivated in a grasping scenario, no actual grasping has been shown.

7.1.1.3 Learning Object Models

All of the above mentioned approaches are dependent on an a-priori known dense or detailed object model either in 2D or in 3D that may be difficult to obtain.

Different methods have been proposed to build a model of a new object by combining a sequence of observations. Foix et al. [81] present a method for merging partial object point clouds when facing strong noise in the data and uncertainty in the estimated camera trajectory. The approach is based on standard ICP for registering point clouds of two consecutive views. The uncertainty in the resulting camera pose estimates is estimated by propagating the sensor covariance through the ICP minimization. Collet Romea et al. [55] proposes to acquire object models offline consisting of an image, a sparse 3D model and a simplified mesh model. A method towards autonomous learning of object surface models is demonstrated by Krainin et al. [123]. The robot holds an object in its hand and plans the next best views to acquire a surface mesh. An alternative way to learn object models is taken by Tenorth et al. [218]. Here, web-enabled robots are proposed that can retrieve appearance and 3D models from online databases.

7.1.2 Grasping Unknown Objects

If the currently considered object hypothesis could not be recognized as a known object and is not similar to one either, then it is considered as *unknown*. In this case, the previously listed approaches are not applicable since in practice it is very difficult to infer object geometry fully and accurately from measurements taken from sensor devices such as cameras and laser range finders. There are various ways to deal with this sparse, incomplete and noisy data. We divided the approaches roughly into methods that try to approximate the shape of an object, methods that generate grasps based on low-level features and a set of heuristics and methods that rely mostly on the global shape of the object hypothesis.

7.1.2.1 Approximating Unknown Object Shape

One approach towards generating grasp hypotheses for unknown objects is to approximate them with shape primitives that provide cues for potential grasps. Dunes et al. [67] approximate the rough object shape with a quadric whose minor axis is used to infer the wrist orientation. The object centroid serves as the approach target and the rough object size helps to determine the hand pre-shape. The quadric is estimated from multi-view measurements of object shape in monocular images. Opposed to the above mentioned techniques, Bone et al. [37] make no prior assumption about the shape of the object. They apply shape carving for the purpose of grasping with a parallel-jaw gripper. After obtaining a model of the object, they search for a pair of reasonably flat and parallel surfaces that are best suited for this kind of manipulator.

7.1.2.2 From Low-Level Features to Grasp Hypotheses

A common approach is to map low-level features to an abundance of grasp hypotheses and then rank them dependent on a set criteria. Kraft et al. [120] use a stereo camera to extract a representation of the scene. Instead of a raw point cloud, they are processing it further to obtain a sparser model consisting of local multi-modal contour descriptors already mentioned above in conjunction with the work by Detry et al. [62, 64] and Kroemer et al. [124]. Four elementary grasping actions are associated to specific constellations of these features. With the help of heuristics the huge number of resulting grasp hypothesis is reduced. Popović et al. [172] present an extension of this system that also uses local surfaces and their interrelations to propose and filter two and three-fingered grasp hypotheses. The feasibility of the approach is evaluated in a mixed real-world and simulated environment.

Hsiao et al. [99] employ several heuristics for generating grasp hypotheses dependent on the shape of the segmented point cloud. These can be grasps from the top, from the side or applied to high points of the objects. The generated hypotheses are then ranked using a weighted list of features such as for example number of points within the gripper, distance between the fingertip and the center of the segment or whether the object fits into the whole hand. This method is integrated into the grasping pipeline proposed in [52] for the segmented point cloud clusters that did not get recognized as a specific object.

The main idea presented by Klingbeil et al. [118] is to search for a pattern in the scene that is similar to the 2D cross section of the robotic gripper interior. A depth image serves as the input to the method and is sampled to find a set of grasp hypotheses. These are ranked according to an objective functions that takes pairs of these grasp hypotheses and their local structure into account.

7.1.2.3 From Global Shape to Grasp Hypothesis

Other approaches use the global shape of an object to infer one good grasp hypothesis. Richtsfeld and Vincze [184] use a segmented point cloud from a stereo camera. They are searching for a suitable grasp with a simple gripper based on the shift of the top plane of an object into its centre of mass. A set of heuristics is used for selecting promising fingertip positions. Maldonado et al. [142] model the object as a 3D Gaussian. For choosing a grasp configuration, it optimises a criterion in which the distance between palm and object is minimised while the distance between fingertips and the object is maximised. Stückler et al. [213] generate grasp hypotheses based on eigenvectors of the object's *footprints* on the table. With footprints, we refer to the 3D object point cloud projected onto the supporting surface. A similar approach has been taken in our previous work [10]. This approach is related to [21] in which it is shown that people are optimizing the aforementioned skewness measure.

7.1.2.4 Assuming Symmetry

Many of the man-made objects surrounding us possess one or more symmetries. Exploiting this, allows us to predict unobserved parts of an object hypothesis. Marton et al. [147] show how grasp selection can be performed based on such an approximation of unobserved object shape. Using footprint analysis, objects with a box and a cylinder shape are reconstructed. The remaining object hypotheses are analysed in terms of more complex rotational symmetry and reconstructed by fitting a curve to a cross section of the point cloud. For grasp planning, the reconstructed object is imported to a simulator. Grasp candidates are generated through randomisation of grasp parameters on which then the force-closure criteria is evaluated. Rao et al. [180] sample grasp points from the surface of a segmented object. The normal of the local surface at this point serves as a search direction for a second contact point. This is chosen to be at the intersection between the extended normal and the opposite side of the object. By assuming symmetry, this second contact point is assumed to have a contact normal in the direction opposite to the normal of the first contact point.

In Chapter 6, we presented a related approach that does not rely on selecting the kind of symmetry prior to reconstruction. Furthermore, it takes the complete point cloud for reconstruction into account and not only a local patch. In Section 7.3, different methods to generate grasp candidate on the resulting completed object models are presented.

7.1.3 Grasping Familiar Objects

A promising direction in the area of grasp planning is to re-use experience to grasp *familiar* objects. Many of the objects surrounding us can be grouped together into categories of common characteristics. There are different possibilities what these commonalities can be. In the computer vision community, objects within one category usually share characteristic visual properties. These can be, e.g., a common texture [205] or shape [79, 26], the occurrence of specific local features [60, 127] or their specific spatial constellation [133, 131]. These categories are usually referred to as *basic level categories* and emerged from the area of cognitive psychology [187].

In robotics however, and specifically in the area of manipulation, the goal is to enable an embodied, cognitive agent to interact with these objects. In this case, objects in one category should share common affordances [211]. More specifically, this means that they should also be graspable in a similar way. The difficulty then is to find a representation that can encode this common affordance and is grounded in the embodiment and cognitive capabilities of the agent. Given this representation, a similarity metric has to be found under which different objects can be compared to each other.

The following approaches try to learn from experience how objects can be grasped. This is different from the aforementioned systems in which objects are assumed to be unknown, i.e, dissimilar to any object encountered previously. There the difficulty lies in finding appropriate rules and heuristics. In the following, we will present related work that tackles the grasping of familiar objects and specifically focus on the applied representations.

7.1.3.1 Based on 3D Data

First of all, there are approaches that rely on 3D data only. For example, El-Khoury and Sahbani [72] segment a given point cloud into parts and approximate each part by a SQ. An artificial neural network (ANN) is used to classify whether or not the part is prehensile. The ANN has been trained beforehand on labeled SQs. If one of the object parts is classified as prehensile, an n-fingered force-closure grasp is computed on this object part. Pelossof et al. [168] use a single SQ to find a suitable grasp configuration for a Barrett hand consisting of the approach vector, wrist orientation and finger spread. An SVM is trained on data consisting of feature vectors containing the parameters of the SQ and of the grasp configuration. They were labelled with a scalar estimating the grasp quality. When feeding the SVM only with the shape parameters of the SQ, their algorithm searches efficiently through the grasp configuration space for parameters that maximise the grasp quality. Curtis and Xiao [59] build upon a database of 3D objects annotated with the best grasps that can be applied to them. To infer a good grasp for a new object, very basic shape features, e.g., the aspect ratio of the object's bounding box, are extracted to classify it as similar to an object in the database. The assumption made in this approach is that similarly shaped objects can be grasped in a similar way. Song et al. [209] learn a Bayesian Network from labeled training data to model the joint distribution between different variables describing a grasp. These variables can be object or action related (object size and class as well as grasp position and final hand configuration); they can describe constraints or be the task label. Given the learnt joint distribution, different information can be inferred as for example the task after having observed the object and the action features.

All these approaches were performed and evaluated in simulation where the central assumption is that accurate and detailed 3D models are available in a known pose. There are only very few approaches, that only use 3D data from real-world sensors to infer suitable grasps. One of them has been presented by Hübner and Kragic [101]. A point cloud from a stereo camera is decomposed into a constellation of boxes. The simple geometry of a box reduces the number of potential grasps significantly. To decide which of the sides of the boxes provides a good grasp, an ANN is trained offline. The projection of the point cloud inside a box to its sides provides the input to the ANN. The training data consists of a set of these projections from different objects labeled with the grasp quality metrics. These are computed by the GraspIt! Simulator while performing the according grasps. Although, grasp selection is demonstrated on one objects, it remains unclear how well the ANN can generalize from the synthetic training set to real-world data.

3D Descriptors of Objects for Categorization Recently, we have seen an increasing amount of new approaches towards pure 3D descriptors of objects for categorization. Although, the following methods look promising, it has not ben shown yet that they provide a suitable base for generalizing grasps over an object category. Rusu et al. [197, 195] provide extensions of [194] for either recognizing or categorizing objects and estimating their pose relative to the viewpoint. While in [195] quantitative results on real data are presented, [197] uses simulated object point clouds only. Wohlkinger and Vincze [231] use a combination of different shape descriptors of partial object views to retrieve matching objects of a specific category from a large database. The retrieval problem is formulated as a nearest-neighbor approach. Robustness is achieved by exploiting intra class view similarities.

Although 3D descriptor of object hypotheses are used in Marton et al. [148] and Marton et al. [146], they are combined with 2D features to increase the robustness of detection. These approaches will therefore be discussed in more detail in Section 7.1.3.3 together with other methods integrating several sensor modalities.

7.1.3.2 Based on 2D Data

As mentioned previously, the assumption of having a detailed and accurate 3D object model available for learning to generalize grasps may not always be valid. This is particularly the case with real-world data gathered from sensors like laser-range finders, the kinect, stereo or time-of-flight cameras. However, there are experience-based approaches that avoid this difficulty by relying mainly on 2D data.

Kamon et al. [108] propose one of the first approaches towards generalizing grasp experience to novel objects. The aim is to learn a function $f : Q \rightarrow G$ that maps object- and grasp-candidate-dependent quality parameters Q to a grade G of the grasp. An object is represented by its 2D silhouette, its center of mass and main axis. The grasp is represented by two parameters f_1 and f_2 from which in combination with the object features the fingertip positions can be computed. The authors found that the normals of the object surface at the contact points and the distance of the line connecting the contact points to the object's center of mass are highly reliable quality features for predicting the grade of a grasp. Learning is bootstrapped by the offline generation of a knowledge database containing grasp parameters along with their grade. This knowledge database is then updated while the robot collects experience by grasping new objects. The system is restricted to planar grasps and visual processing of top-down views on objects. It is therefore questionable how robust this approach is to more cluttered environments and strong pose variations of the object.

Saxena et al. [200] proposed a system that infers a point at where to grasp an object directly as a function of its image. The authors apply machine learning to train a grasping point model on labelled synthetic images of a number of different objects. The classification is based on a feature vector containing local appearance cues regarding color, texture and edges of an image patch in several scales and of its 24 neighboring patches in the lowest scale. The system was used successfully to pick up objects from a dishwasher after it has been specifically trained for this scenario. However, if more complex goals are considered that require subsequent actions, e.g., pouring something from one container into another, semantic knowledge about the object and about suitable grasps regarding their functionality becomes necessary [38, 57, 94]. Then, to only represent graspable points without the conception of *objectness* [120][9] is not sufficient.

Another example of a system involving 2D data and grasp experience is presented by Stark et al. [211]. Here, an object is represented by a composition of prehensile parts. These so called *affordance cues* are obtained by observing the interaction of a person with a specific object. Grasp hypotheses for new stimuli are inferred by matching features of that object against a codebook of learnt *affordance cues* that are stored along with relative object position and scale. However, how exactly to grasp these detected prehensile parts is not yet solved since hand orientation and finger configuration are not inferred from the affordance cues. More successful in terms of the inference of full grasp configurations are Morales et al. [155] who use visual feedback to even predict fingertip positions. The authors also take the hand kinematics into consideration when selecting a number of planar grasp hypotheses directly from 2D object contours. To predict which of these grasps is the most stable one, a KNN-approach is applied in connection with a grasp experience database. The experience database is built through a trial-and-error approach applied in the real world. Grasp hypotheses are ranked dependent on their outcome. However, the approach is restricted to planar objects.

7.1.3.3 Integrating 2D and 3D Data

In Figure 7.1, one aspect that influences the selection of grasp hypotheses is the employed modality of the object representation. We believe that systems in which both 2D and 3D data are integrated are most promising in terms of dealing with sensor noise and removing assumptions about object shape or applicable grasps. In Figure 7.1, this case is represented by the edge connecting 2D and 3D object representations. In the following, we will review two different kinds of these approaches. The first ones employ relatively low-level features for grasp knowledge transfer. The second kind try to generalize over an object category.

Grasping using Low-Level Representations Saxena et al. [201] apply two depth sensors to obtain a point cloud of a tabletop scene with several objects. The authors extend their previous work on inferring 2D grasping point hypotheses. Here, the shape of the point cloud within a sphere centered around a hypothesis is analysed with respect to hand kinematics. This enhances the prediction of a stable grasp and also allows for the inference of grasp parameters like approach vector and finger spread. In earlier work by Saxena et al. [200], only downward or outward grasp were possible with the manipulators in a fixed pinch grasp configuration.

Speth et al. [210] showed that their earlier 2D based approach [155] is also applicable when considering 3D objects. The camera is used to explore the object to retrieve crucial information like height, 3D position and pose. However, all this additional information is not applied in the inference and final selection of a suitable grasp configuration. We introduced the work by Rao et al. [180] already in the previous section about grasping unknown objects. We referred to their strategy of choosing fingertip positions given a segmented object hypotheses. Here, we want to emphasize how the authors distinguish between graspable and non-graspable object hypotheses in a machine learning framework. Using a combination of 2D and 3D features, an SVM is trained on labeled data of segmented objects. Interestingly, among those features are for example the variance in depth and height as well as variance of the three channels in the Lab color space. This is similar to the entropy features used in Chapter 5, for classifying object hypotheses as being correct or not. Rao et al. [180] achieve good classification rates on object hypotheses formed by segmentation on color and depth cues. Similar to the work in [180], Le et al. [132] consider grasp hypotheses as consisting of two contact points. They however, apply a learning approach to rank a sampled set of fingertip positions according to graspability. The feature vector consists of a combination of 2D and 3D cues such as gradient angle or depth variation along the line connecting the two grasping points.

In Section 7.4, we are also proposing an approach that integrates 2D and 3D information. Similar to [201], we see the result of the 2D based grasp selection as a way to search in a 3D object representation for a full grasp configuration. Here, we will focus on the development of the 2D method and demonstrate its applicability for searching in a minimal 3D object representation. The proposed approach from Section 7.4 was extended to work on a sparse edge based representation of objects [1]. We showed that by integrating 3D and 2D based methods for grasp hypotheses generation leads to a sparser set of grasps with a good quality.

Category-based Grasping The previous approaches do not explicitly consider any high level information about for example the object category. In the following, we review method that try to categorize objects and are motivated in a grasping scenario. Marton et al. [146] use different 3D sensors and a thermo camera for object categorization. Features of the segmented point cloud and the segmented image region are extracted to train a Bayesian Logic Network for classifying object hypotheses as either boxes, plates, bottles, glasses, mugs or silverware. In [146] no grasping results are shown. A modified approach is presented in [149]. A layered 3D object descriptor is used for categorization and a SIFT-based approach is applied for view based object recognition. To increase robustness of the categorization, the examination methods are run iteratively on the object hypotheses. A list of potential matching objects are kept and re-used for verification in the next iteration. Objects for which no matching model can be found in the database are labeled as novel. Given that an object has been recognized, associated grasp hypotheses can be re-used. These have been generated using the technique presented in [147].

Lai et al. [128] perform object category and instance recognition. The authors learn an instance distance using the database presented in [129]. A combination of 3D and 2D features is used. However, the focus of this work is not grasping.

There is one interesting concluding remark that we can make. Recently, an increasing amount of approaches has been developed that try to recognize or categorize objects in point clouds. However, none of these approaches have yet been shown to be a suitable representation for transferring grasping experience between objects of the same class. Instead this is solved by finding the (best) matching object instance on which grasp hypotheses have been generated offline. Generalizing from already seen objects to familiar objects in the real-world has up till now only been shown to be possible for low-level object representations.

7.1.4 Hybrid Systems

There are a few empirical grasp synthesis systems that cannot clearly be classified as using only one kind of prior knowledge. One of these approaches has been proposed in Brook et al. [45]. Different grasp planners are integrated to reach a consensus on how to grasp a segmented point cloud. The authors show results using the planner presented in [99] for unknown objects in combination with grasp hypotheses generated through fitting known objects to point cloud clusters as described in [52].

Another example for a hybrid approach is the work by Marton et al. [149]. A set of very simple shape primitives like boxes, cylinders and more general rotational objects are considered. They are reconstructed from segmented point clouds by analysis of their footprints. Parameters such as circle radius and the side lengths of rectangles are varied; curve parameters are estimated to reconstruct more complex rotationally symmetric objects. Given these reconstructions, a look-up is made in a database of already encountered objects for re-using successful grasp hypotheses. In case no similar object is found, new grasp hypotheses are generated using the technique presented in [147]. For object hypotheses that cannot be represented by the simple shape primitives mentioned above, a surface is reconstructed through triangulation. Grasp hypotheses are generated using the planner presented in [99].

7.1.5 Summary

In this section, we reviewed empirical grasp synthesis methods. We subdivided them according to their assumed prior knowledge about an object hypothesis. In the following, we propose grasp synthesis methods that go beyond the current state of the art.

For grasping known objects, we demonstrate object recognition and pose estimation methods as already partially referred to in this section. This is followed by a study on whether completed point clouds can improve the accuracy and robustness of these approaches. For grasping unknown objects, we present an approach based on point cloud completion through a symmetry assumption as presented in Chapter 6. We show different methods for generating grasp hypotheses for these.

And finally, for grasping familiar objects, we present a method in which global object shape in a 2D image is used as feature vector in a learning scheme. Detected 2D grasping points in conjunction with 3D shape cues are used for inferring a full grasp pose.

7.2 Grasping Known Objects

In this section, we will show four example systems that are based on the active vision system presented in Section 2.2.1. The objects in front of it are considered to be known. Their goal is to determine where these objects are in the scene and in what pose. The first two use a scene model as input in which a 3D point cloud is segmented into a number of hypotheses. The segments are then further analysed to obtain the object's identity and pose. In the third example system, no prior segmentation of the point cloud into object hypotheses is necessary. Object recognition and pose estimation is performed simultaneously.

We extend these approaches in the fourth method by considering not only the sensory data from the scene. Instead, we will also study how a completed point cloud can help to increase robustness of object recognition and pose estimation.

Given this information, the scene model containing known models can be fed into a simulator where grasps from an experience database are tested for reachability. This will be detailed in Chapter 8.

7.2.1 Recognition

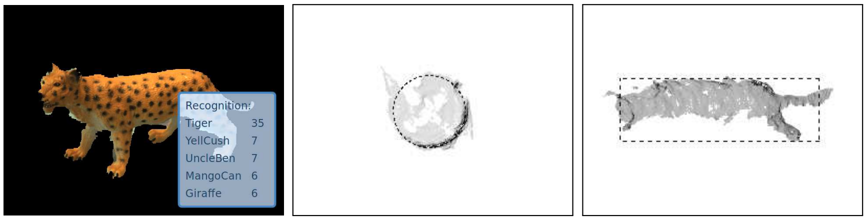
The identities of the object hypotheses detected in a scene are sought in a database of 25 known objects. Two complementary cues are used for recognition; SIFT and CCH. The cues are complementary in the sense that SIFT features rely on objects being textured, while CCH works best for objects of a few, but distinct, colors. An earlier version of this system with experiments in a significantly larger database and including a study on the benefits of segmentation for recognition, can be found in [33]. Here, the total recognition score is a weighted sum of the single scores. An example of the recognition results is shown in Figure 7.3a.

7.2.2 Pose Estimation

Once we know the identity of an object hypotheses, we can use the associated model to estimate its pose. In the following, we demonstrate a system that uses a simplified object model and a system that retrieves the full 6D object pose of an object using a 3D mesh model.

7.2.2.1 Simplified Pose Estimation

If an object is identified and it is known to be either rectangular or cylindrical, the pose is estimated using the cloud of extracted 3D points projected onto the



(a) Segmented Image and Recognition Result. Five best matching objects are shown in the table.

(b) Estimated Pose of Mango Can in Figure 2.4b.

(c) Estimated Pose of Toy Tiger in Figure 2.4a.

Figure 7.3: Example Output for Recognition and Pose Estimation.

2D table plane. Object dimensions are looked up in the database. This means that instead of searching for the full 6DoF pose, we reduce the search space to the position (x, y) of either a rectangle or a circle on the table. In case of the rectangle the additional orientation parameter θ is sought for. This is done using the method already described in Section 4.2.2.2. Essentially it is based on RANSAC in combination with least-median optimisation for either a rectangular or circular shape. Figure 7.3b and 7.3c show an example for each shape.

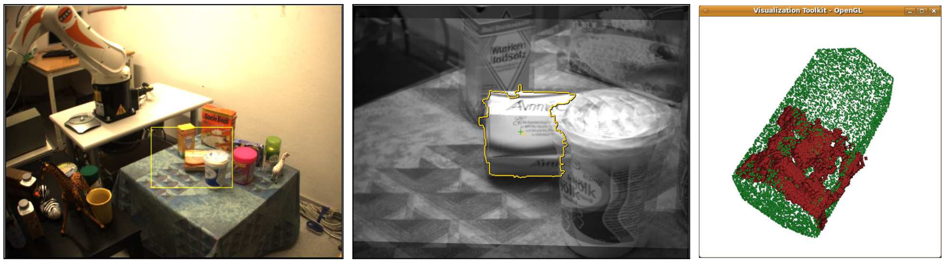
There are several drawbacks of this simple method. First of all, it will not be able to correctly estimate the pose of an object that does not rest on one of its sides. These kinds of situations occur when objects are lying in a pile or in a grocery bag. Secondly, this approach can probably not deal very well with the case of strong occlusions. And lastly, because the object models are very simple, motion planning for collision avoidance will also not be very accurate. However, as will be shown in Section 8.2.2.2 this simple approach works well in combination with position-based visual servoing.

7.2.2.2 Full Pose Estimation

For estimating the full 6 DoF pose of an object given a segmented point cloud and the object's identity, we applied the approach presented in Papazov and Burschka [165]. A 3D mesh model is assumed to be available for each object in the database[2].

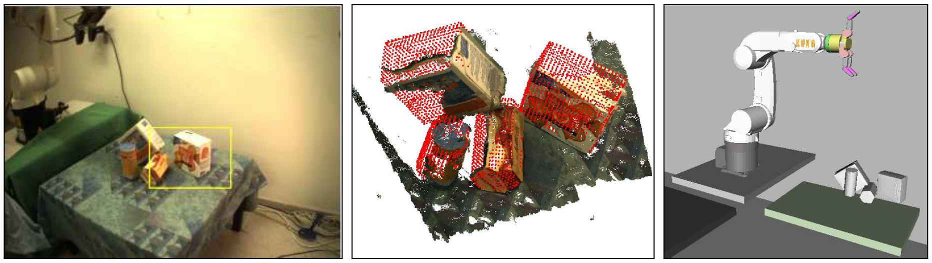
The problem of pairwise rigid point set registration is approached with a stochastic method for global minimisation of a proposed cost function. The method has been shown to be robust against noise and outliers. Although it does not rely on a good initial alignment of the source to the target, prior segmentation of object hypotheses improves its performance. Figure 7.4 shows an example result of this method given a segmented point cloud from the real-time vision system described in Section 2.2.1.

[2] The models were obtained from `http://i61p109.ira.uka.de/ObjectModelsWebUI/`



(a) Peripheral Image of the Scene (b) Segmented Object in the Foveal Image. (c) Model Point Cloud (green) Fit to Target Point Cloud (red).

Figure 7.4: Example Output for 6DoF Pose Estimation.



(a) Peripheral Image of the Scene (b) Point Cloud of the Scene with Fitted Models (red). (c) Representation of the Scene in OpenRave.

Figure 7.5: Example Output for Simultaneous Object Recognition and Pose Estimation in Point Cloud.

7.2.3 Simultaneous Recognition and Pose Estimation of Several Targets

We also demonstrated the approach by Papazov and Burschka [166] for simultaneous recognition and pose estimation of rigid objects in a 3D point cloud of a scene. No prior segmentation is necessary.

It relies on the combination of a robust geometric descriptor of oriented 3D points, a hashing technique and a RANSAC-based sampling technique. It is robust to noise, outliers and sparsity. Example results of the approach applied to merged point clouds from the real-time vision system are shown in Figure 7.5.

7.2.4 Using Completed Point Clouds for Object Recognition and Pose Estimation

The previous methods towards object recognition and pose estimation use as an input the raw stereo reconstructed point cloud that is potentially segmented into object hypotheses. In this section we are going to analyse if a completed point cloud helps to improve the robustness of registering known object models to it. This completion will be performed through the method proposed in Chapter 6. The hidden or occluded part of an object is estimated based on a symmetry assumption. Visibility constraints are employed to assign a plausibility measure to each point in the completed point cloud.

In the following, we will pose the registration problem as the classic *Orthogonal Procrustes* problem introduced in Hurley and Cartell [103]. We will briefly introduce the *iterative closest point* algorithm and its weighted formulation following Besl and McKay [30] and Maurer et al. [150]. The latter will be useful to explicitly consider the plausibility information of each point.

7.2.4.1 Orthogonal Procrustes Problem

The registration of an object model to a point cloud can usually be described as a rigid body transformation

$$\mathcal{T}(\mathbf{a}) = \mathbf{R} \cdot \mathbf{a} + \mathbf{t} \tag{7.1}$$

where $\mathbf{R}$ is a 3×3 rotation matrix, $\mathbf{t}$ a three-dimensional translation vector and $\mathbf{a}$ a three-dimensional point of a point cloud. Let $\mathcal{A}$ be a set containing points $\mathbf{a}_j$ with $j = 1 \ldots N_a$ to be registered to a different set $\mathcal{B}$ with points $\mathbf{b}_j$ with $j = 1 \ldots N_b$ where $N = N_a = N_b$. Each point $\mathbf{a}_j$ corresponds to a point $\mathbf{b}_j$ with the same index. The Orthogonal Procrustes Problem is described as finding the rigid-body transformation $\mathcal{T}$ that minimises the following cost function:

$$d(\mathcal{T}) = \sqrt{\frac{1}{N}\sum_{j=1}^{N} ||\mathbf{b}_j - \mathcal{T}(\mathbf{a}_j)||^2} \tag{7.2}$$

Aarun et al. [15] have shown that this problem can be solved in closed form using for example the Singular Value Decomposition (SVD).

7.2.4.2 ICP and Weighted ICP

The problem of registering an object model to a point cloud usually suffers from the problem of not knowing the correspondences between the two point sets. In that case, there is no closed-form solution available. The general approach is then to search iteratively for the rigid-body transformation that minimises

$$d(\mathcal{T}) = \sqrt{\frac{1}{N_a}\sum_{j=1}^{N_a} ||\mathbf{y}_j - \mathcal{T}(\mathbf{a}_j)||^2} \tag{7.3}$$

where

$$\mathbf{y}_j = c(\mathcal{T}(a_j), \mathcal{B}) \tag{7.4}$$

is a point in the point set $\mathcal{B}$ that corresponds to point $\mathbf{a}_j$. This relation is determined through the correspondence function $c(\cdot)$

Besl and McKay [30] proposed ICP in which $c(\cdot)$ is the closest point operator. They showed that their method converges to the local minimum of the cost function in Equation 7.3.

Maurer et al. [150] proposed a weighted extension of the original ICP method. Let $\mathcal{W}$ be a set of weights w_j with $j = 1 \dots N_a$ associated with the points in $\mathcal{A}$. The authors show that their weighted ICP formulation minimises the following cost function:

$$d(\mathcal{T}) = \sqrt{\frac{1}{N_a} \sum_{j=1}^{N_a} w_j ||\mathbf{y}_j - \mathcal{T}(\mathbf{a}_j)||^2} \tag{7.5}$$

where the function $c(\cdot)$ is defined as in Equation 7.4.

ICP for Completed Point Clouds In this section, we propose to register the true object model to the completed point cloud instead of to the partial one. Our hypothesis is that this increases the robustness of fitting because more points that constrain the pose of the true object are available.

We use ICP as implemented in [193] that iteratively minimises Equation 7.3. The set $\mathcal{B}$ consist of the completed point cloud. $\mathcal{A}$ contains point samples from the surface of the ground truth object. The mesh is sampled using the method described in Roy et al. [189].

Using Plausibility Information for Registration The quality of the ICP-based registration of the object model to the completed point cloud depends on the quality of the symmetry plane estimation. The resulting point cloud might diverge strongly from the true shape. Therefore, we want to ensure that the registered model complies more accurately to the original reconstructed points than to the hypothesized ones.

We approach this, by using a weighted ICP scheme proposed in Maurer et al. [150] that minimises the cost function in Equation 7.5. $\mathcal{B}$ consists of the completed point cloud and $\mathcal{A}$ of the sampled ground truth mesh. As weights w_j, we choose the plausibility $\texttt{plaus}(\mathbf{b}_j)$ of each point $\mathbf{b}_j$ in the set $\mathcal{B}$. This is defined in Equation 6.3 and 6.4.

We implemented this approach by adapting ICP from [193] as shown in Algorithm 3.

7.2.4.3 Experiments

In this section, we want to evaluate the proposed approach of using the completed point clouds as targets for the registration of ground truth object models. Our

Algorithm 3: One Loop in the Weighted ICP Algorithm

```
Data: Initial Transformation $\mathcal{T}$
Subset $\mathcal{P} = \{\mathbf{p}_i\}$ of $\mathcal{A} = \{\mathbf{a}_j\}$ (Target Point Cloud)
Subset $\mathcal{X} = \{\mathbf{x}_i\}$ of $\mathcal{B} = \{\mathbf{b}_j\}$ (Model Point Cloud) where $\mathbf{x}_i = C(\mathcal{T}(\mathbf{p}_i), \mathcal{B})$
Subset $\mathcal{V} = \{v_i\}$ of $\mathcal{W}$ with $v_i = \texttt{plaus}(\mathbf{p}_i)$
$i \in \{1 \dots N\}$
Result: New $\mathcal{T}$ minimizing $d(\mathcal{T})$ in Equation 7.5
begin
    // Compute Weighted Centroids
    $\bar{\mathbf{p}} = \frac{\sum_{i=1}^{N} w_i \mathbf{p}_i}{\sum_{j=1}^{N} w_i}$
    $\bar{\mathbf{x}} = \frac{\sum_{i=1}^{N} w_i \mathbf{x}_i}{\sum_{j=1}^{N} w_i}$
    // Translate each point
    foreach $i \in \{1 \dots N\}$ do
        $\mathbf{p}'_i = \mathbf{p}_i - \bar{\mathbf{p}}$
        $\mathbf{x}'_i = \mathbf{x}_i - \bar{\mathbf{x}}$
    end
    // $3 \times 3$ covariance matrix
    $\mathbf{H} = \sum_{i=1}^{N} w_i \mathbf{p}'_i \mathbf{x}'^{T}_i$
    // Factorization of $H$ through Singular Value Decomposition
    $\mathbf{U} \mathbf{\Lambda} \mathbf{V}^T = \mathrm{SVD}(\mathbf{H})$
    // Computation of Rotation Matrix $\mathbf{R}$
    $\mathbf{R} = \mathbf{V} \mathbf{U}^T$
    if $\det \mathbf{R} == -1$ then
        $\mathbf{R} = \mathbf{V}' \mathbf{U}^T$
    end
    // Computation of Translation Vector
    $\mathbf{t} = \bar{\mathbf{x}} - \mathbf{R} \bar{\mathbf{p}}$
    // Whole Transformation
    $\mathcal{T} = [\mathbf{R} | \mathbf{t}]$
end
```

hypothesis is that this increases the robustness of pose estimation.

For registration, we compare two methods: i) standard ICP in which all points in the completed point cloud are taken as equally important and ii) weighted ICP in which the plausibility of each point in the completed cloud is explicitly taken into account. As a baseline, we use standard ICP in which a ground truth model is fitted to the original partial point cloud.

For evaluation, we use the same database as in Chapter 6. It consists of 10 distinct objects in different poses yielding 14 datasets for which a ground truth object pose is available. Each dataset contains segmented 3D point clouds reconstructed from the objects in 8 different orientations (see Figure 6.5 and 6.6). The completed point clouds, that are the result of applying the method proposed in Chapter 6, form the input to the different registration methods. In the following, we will test our hypothesis in two different scenarios.

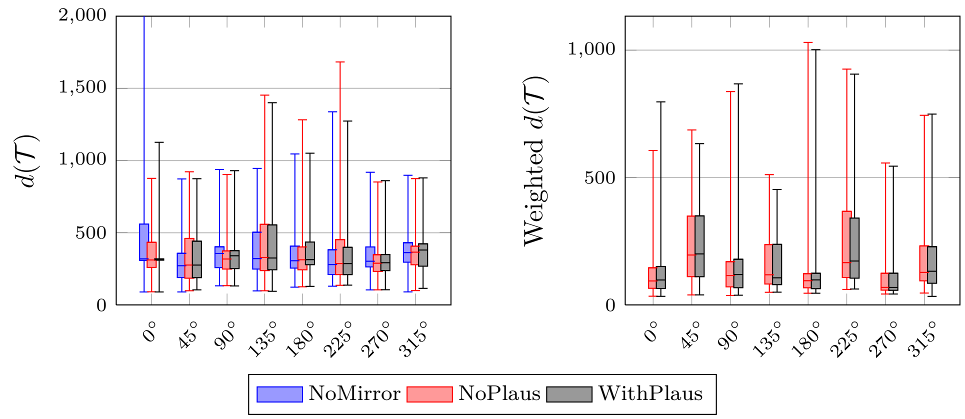


Figure 7.6: Error Measures after Fitting Object Models to Point Cloud. Left) Comparison of the mean squared error for the best fit averaged over all objects for each orientation. The performance for the different methods are not significantly different. Right) Comparison of the weighted mean squared error for the best fit. The weighted ICP (withPlaus) optimises this measure and should therefore produce lower values than the standard ICP (NoPlaus). However, the data does not show a significant difference between the two approaches.

Pose Estimation In the first scenario, we assume that we know the identity of the segmented object. The goal is to find its 6 DoF pose.

Since ICP is a method that converges to a local minimum, we need to initialise it with an object pose that is relatively close to ground truth, i.e., the global minimum. This could be achieved by methods such as for example proposed by Rusu et al. [194]. Here, we make use of the fact that in our dataset the object pose with a $-45°$ difference to the ground truth pose is available.

We compare the three different registration methods in terms of mean squared error as defined in Equation 7.3. The results are shown in Figure 7.6. They do not show a significant difference between the proposed methods. This is probably due to the good segmentation of the object models, the low number of outliers and the good initial pose.

Recognition through Registration In the second scenario, we want to analyse, whether using the completed point cloud facilitates a more robust *joint* recognition and pose estimation.

Experimental Setup Given a point cloud $\mathcal{B}$, we want to find the object model o^* in pose $\mathcal{T}$ that minimises

$$o^* = \arg\min_{\mathcal{O}} d(\mathcal{T}) \tag{7.6}$$

Divergence from Ground Truth	Original	Unweighted	Weighted
±45° and 0°	50.42%	49.58%	52.08%
0°	55.00%	47.50%	51.25%
±45°	48.12%	50.62	52.50 %

Table 7.1: Summary of total recognition results for joint recognition and pose estimation. Original: Standard ICP on original partial point clouds. Unweighted: Standard ICP on completed point clouds. Weighted: Weighted ICP on completed point clouds with plausibility information.

where the set $\mathcal{O}$ of objects is depicted in Figure 6.5.

Dependent on the input, $d(\mathcal{T})$ is defined in two ways. If $\mathcal{B}$ is a completed point cloud with plausibility information, then it is computed as in Equation 7.5. If $\mathcal{B}$ is a completed point cloud without plausibility information or the original point cloud, $d(\mathcal{T})$ is defined as in Equation 7.3.

As model point cloud $\mathcal{A}$, we again use the sampled meshes in initial poses of ±45° or 0° deviation in orientation from ground truth. As in the previous experiment, we compare standard ICP using the original point cloud with ICP on the completed point cloud and weighted ICP on the completed point cloud with plausibility information.

Results Averaging over all objects, orientations and initial model poses, we achieve the following total recognition results: Orignal 50.42%. Unweighted 49.58%. Weighted 52.08%. When using the weighted ICP method that takes plausibility information explicitly into account, we achieve the best recognition results. The standard ICP on the completed point cloud performs on average even worse than the approach on partial point clouds. This indicates that inaccuracies in the object shape prediction through the symmetry assumption affect the registration result negatively.

We can gain more insights, if we split these results into the case when the initial pose is the same as the ground truth pose and the case when the initial pose has ±45° deviation in orientation. The corresponding full confusion matrices are given in Figure 7.7 and 7.8. An overview is given in Table 7.1.

For the case where the initial pose is equal to the ground truth, the standard method on partial point clouds performs clearly superior over the methods using the completed point clouds. However, when the initial pose of the object models is different from ground truth, using the completed point clouds in combination with weighted ICP results in a better recognition accuracy than the standard approach. This indicates that in the common case of the initial pose being only approximately known, completed point clouds provide more robustness in the task of joint object recognition and pose estimation.

The alignment method we used in this section is based on the widely applied ICP. This method is guaranteed to converge to a local minimum. However, there is a set of alignment method that aim at finding the global minimum in the space of

Actual

Predicted	amicelli	burti	greenCup	mangoLy	mangoStand	rusk	saltBoxLy	saltBoxStand	saltCylLy	saltCylStand	spray	tomatoSoup	tomatoSoupLy	whiteCup
amicelli	**0.00**	0.00	0.00	0.12	0.00	0.00	0.25	0.12	0.00	0.00	0.00	0.00	0.50	0.00
burti	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.62	0.00	0.25	0.00	0.12	0.00	0.00
greenCup	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mangoLy	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
mangoStand	0.00	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
rusk	0.00	0.00	0.00	0.25	0.25	**0.00**	0.00	0.00	0.25	0.00	0.00	0.25	0.00	0.00
saltBoxLy	0.00	0.00	0.00	0.00	0.00	0.00	**0.62**	0.12	0.00	0.00	0.00	0.00	0.25	0.00
saltBoxStand	0.00	0.00	0.00	0.00	0.00	0.00	0.12	**0.62**	0.00	0.00	0.00	0.00	0.25	0.00
saltCylLy	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	**0.25**	0.25	0.00	0.00	0.25	0.00
saltCylStand	0.00	0.00	0.00	0.00	0.00	0.00	0.38	0.12	0.12	**0.38**	0.00	0.00	0.00	0.00
spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.25	**0.00**	0.00	0.00	0.00
tomatoSoup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.75**	0.25	0.00
tomatoSoupLy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00
whiteCup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	**0.75**
F1	0.00	0.00	1.00	0.00	0.00	0.00	0.48	0.37	0.31	0.36	0.00	0.48	0.42	0.86

Total Recognition Rate: 55.00

(a) NoMirror

Actual

Predicted	amicelli	burti	greenCup	mangoLy	mangoStand	rusk	saltBoxLy	saltBoxStand	saltCylLy	saltCylStand	spray	tomatoSoup	tomatoSoupLy	whiteCup
amicelli	**0.12**	0.00	0.00	0.12	0.12	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.38	0.00
burti	0.00	**0.00**	0.00	0.00	0.00	0.00	0.12	0.38	0.12	0.12	0.00	0.25	0.00	0.00
greenCup	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mangoLy	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.12	0.75	0.00
mangoStand	0.00	0.00	0.00	0.00	**0.12**	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.12	0.12
rusk	0.00	0.00	0.00	0.25	0.25	**0.25**	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.12
saltBoxLy	0.00	0.00	0.00	0.00	0.00	0.00	**0.75**	0.00	0.00	0.00	0.00	0.00	0.25	0.00
saltBoxStand	0.00	0.00	0.00	0.00	0.00	0.00	0.12	**0.12**	0.12	0.00	0.00	0.38	0.25	0.00
saltCylLy	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.12	**0.25**	0.00	0.00	0.25	0.25	0.00
saltCylStand	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.12	0.00	**0.00**	0.00	0.62	0.12	0.00
spray	0.00	0.00	0.00	0.00	0.12	0.00	0.12	0.00	0.00	0.00	**0.00**	0.62	0.12	0.00
tomatoSoup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00
tomatoSoupLy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00
whiteCup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**
F1	0.22	0.00	1.00	0.00	0.15	0.40	0.61	0.14	0.34	0.00	0.00	0.41	0.47	0.89

Total Recognition Rate: 47.50

(b) Unweighted

Actual

Predicted	amicelli	burti	greenCup	mangoLy	mangoStand	rusk	saltBoxLy	saltBoxStand	saltCylLy	saltCylStand	spray	tomatoSoup	tomatoSoupLy	whiteCup
amicelli	**0.12**	0.00	0.00	0.25	0.00	0.00	0.00	0.12	0.12	0.12	0.00	0.00	0.25	0.00
burti	0.00	**0.00**	0.00	0.00	0.00	0.00	0.12	0.38	0.12	0.12	0.00	0.25	0.00	0.00
greenCup	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mangoLy	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.12	0.25	0.00	0.25	0.38	0.00
mangoStand	0.00	0.00	0.00	0.00	**0.12**	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.00	0.00
rusk	0.12	0.00	0.12	0.00	0.12	**0.38**	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.12
saltBoxLy	0.00	0.00	0.00	0.00	0.00	0.00	**0.62**	0.12	0.00	0.00	0.00	0.00	0.25	0.00
saltBoxStand	0.00	0.00	0.00	0.00	0.00	0.00	0.12	**0.38**	0.00	0.00	0.00	0.50	0.00	0.00
saltCylLy	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	**0.25**	0.12	0.00	0.12	0.38	0.00
saltCylStand	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.12	0.00	**0.00**	0.00	0.62	0.12	0.00
spray	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.12	**0.00**	0.50	0.12	0.12
tomatoSoup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.75**	0.12	0.12
tomatoSoupLy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	**0.75**	0.00
whiteCup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**
F1	0.20	0.00	0.94	0.00	0.19	0.56	0.53	0.36	0.31	0.00	0.00	0.29	0.45	0.85

Total Recognition Rate: 51.25

(c) Weighted

Figure 7.7: Confusion matrices for the task of recognition of objects represented by point clouds. Source point clouds are 0° from ground truth. Total recognition rate is computed by taking into account object duplicates, e.g., standing or lying mango can. Using the original point cloud as target yields the best recognition rate.

Actual

Predicted	amicelli	burti	greenCup	mangoLy	mangoStand	rusk	saltBoxLy	saltBoxStand	saltCylLy	saltCylStand	spray	tomatoSoup	tomatoSoupLy	whiteCup
amicelli	**0.00**	0.00	0.00	0.00	0.06	0.00	0.31	0.25	0.06	0.00	0.00	0.00	0.19	0.12
burti	0.00	**0.00**	0.00	0.00	0.00	0.00	0.19	0.50	0.06	0.19	0.00	0.06	0.00	0.00
greenCup	0.00	0.00	**0.94**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06
mangoLy	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.62	0.06
mangoStand	0.00	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
rusk	0.00	0.00	0.06	0.06	0.31	**0.00**	0.00	0.12	0.12	0.00	0.00	0.12	0.12	0.06
saltBoxLy	0.00	0.00	0.00	0.00	0.00	0.00	**0.56**	0.12	0.00	0.00	0.00	0.00	0.31	0.00
saltBoxStand	0.00	0.00	0.00	0.12	0.00	0.00	0.06	**0.50**	0.06	0.06	0.00	0.00	0.19	0.00
saltCylLy	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.19	**0.19**	0.25	0.00	0.06	0.06	0.00
saltCylStand	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.44	0.06	**0.19**	0.00	0.25	0.00	0.00
spray	0.00	0.00	0.00	0.00	0.06	0.00	0.06	0.44	0.00	0.12	**0.00**	0.25	0.06	0.00
tomatoSoup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.81**	0.19	0.00
tomatoSoupLy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	**0.69**	0.06
whiteCup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	**0.81**
F1	0.00	0.00	0.94	0.00	0.00	0.00	0.45	0.28	0.25	0.21	0.00	0.39	0.38	0.75

Total Recognition Rate: 48.12

(a) NoMirror

Actual

Predicted	amicelli	burti	greenCup	mangoLy	mangoStand	rusk	saltBoxLy	saltBoxStand	saltCylLy	saltCylStand	spray	tomatoSoup	tomatoSoupLy	whiteCup
amicelli	**0.00**	0.00	0.00	0.38	0.06	0.00	0.00	0.12	0.12	0.12	0.00	0.06	0.12	0.00
burti	0.00	**0.00**	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.25	0.00	0.19	0.06	0.00
greenCup	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mangoLy	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.38	0.50	0.00
mangoStand	0.00	0.00	0.00	0.00	**0.25**	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.12	0.00
rusk	0.06	0.00	0.12	0.00	0.31	**0.25**	0.06	0.00	0.06	0.00	0.00	0.00	0.12	0.00
saltBoxLy	0.00	0.00	0.00	0.06	0.00	0.00	**0.56**	0.19	0.00	0.00	0.00	0.00	0.19	0.00
saltBoxStand	0.00	0.00	0.00	0.00	0.06	0.00	0.12	**0.25**	0.00	0.00	0.00	0.50	0.06	0.00
saltCylLy	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.25	**0.25**	0.25	0.00	0.06	0.00	0.06
saltCylStand	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	**0.00**	0.00	0.69	0.06	0.00
spray	0.00	0.00	0.00	0.06	0.06	0.00	0.00	0.06	0.00	0.12	**0.00**	0.56	0.12	0.00
tomatoSoup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.88**	0.06	0.06
tomatoSoupLy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00
whiteCup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	**0.94**
F1	0.00	0.00	0.94	0.00	0.29	0.41	0.53	0.21	0.35	0.00	0.00	0.36	0.58	0.91

Total Recognition Rate: 50.62

(b) Unweighted

Actual

Predicted	amicelli	burti	greenCup	mangoLy	mangoStand	rusk	saltBoxLy	saltBoxStand	saltCylLy	saltCylStand	spray	tomatoSoup	tomatoSoupLy	whiteCup
amicelli	**0.06**	0.00	0.00	0.25	0.12	0.00	0.12	0.12	0.19	0.00	0.00	0.00	0.06	0.06
burti	0.00	**0.00**	0.00	0.00	0.00	0.00	0.31	0.31	0.06	0.06	0.00	0.25	0.00	0.00
greenCup	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mangoLy	0.00	0.00	0.00	**0.00**	0.00	0.00	0.06	0.00	0.06	0.00	0.00	0.31	0.56	0.00
mangoStand	0.00	0.00	0.00	0.00	**0.31**	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.06	0.00
rusk	0.12	0.00	0.19	0.06	0.06	**0.25**	0.06	0.00	0.00	0.00	0.00	0.12	0.06	0.06
saltBoxLy	0.00	0.00	0.00	0.00	0.00	0.00	**0.62**	0.12	0.00	0.00	0.00	0.00	0.25	0.00
saltBoxStand	0.00	0.00	0.00	0.00	0.00	0.00	0.06	**0.44**	0.00	0.00	0.00	0.50	0.00	0.00
saltCylLy	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.12	**0.25**	0.12	0.00	0.12	0.25	0.00
saltCylStand	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.00	**0.00**	0.00	0.62	0.06	0.00
spray	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.12	0.00	0.12	**0.00**	0.50	0.12	0.06
tomatoSoup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.88**	0.12	0.00
tomatoSoupLy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	**0.94**	0.00
whiteCup	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	**0.94**
F1	0.10	0.00	0.91	0.00	0.42	0.41	0.52	0.35	0.32	0.00	0.00	0.35	0.53	0.89

Total Recognition Rate: 52.50

(c) Weighted

Figure 7.8: Confusion matrices for the task of simultaneous recognition and pose estimation of objects represented by point clouds. Source point clouds are ±45° off from ground truth. Total recognition rate is computed by taking into account object duplicates, e.g., standing or lying mango can. In the very common case of the object pose being only approximately known, using the plausibility information leads to increased robustness of the recognition task.

all object poses [165, 166, 194]. It would be interesting to evaluate whether these approaches could benefit from predicted object shapes.

7.2.5 Generation of Grasp Hypotheses

In the previous section, we studied different methods towards estimating the identity and pose of a set of objects in the scene. Given this information, grasp hypotheses can be selected for each of them. For the approaches relying on 3D mesh models, experience databases can be generated offline using for example OpenRave [66]. Figure 7.5c shows an example scene as recognized by the method in [166] visualized in OpenRave. For each of these objects, the best grasps from the database can be retrieved and tested for reachability using motion planners like for example *Rapidly-exploring random trees* (RRTs) [130] in combination with Jacobian-based gradient descent methods.

For the simplified object models consisting either of a box or a cylinder with a specific height, we choose to perform top grasps that approach the objects at their center. The wrist is aligned with the smallest dimension of the box. This approach is not combined with a motion planner. In Section 8.2.2.2, we will demonstrate and discuss this approach for table-top scenes in combination with position-based visual servoing.

7.3 Grasping Unknown Objects

For grasping unknown objects, i.e., objects for which no matching 3D model is available, no database of good grasps can be set up a priori. Grasp hypotheses have to be generated online and then tested for reachability.

In Chapter 6, we proposed a method that estimates the full object shape from a partial point cloud of an unknown object. We showed that the estimated object shape accurately estimates its true shape. In this section, we want to evaluate how the prediction of object shape can help grasping and manipulation.

In the first part, we consider a very simple grasp strategy that approaches the object centroid. Estimating this centroid is difficult when only a partial object model is known. In the next section, we evaluate quantitatively how much better the centroid can be estimated when using completed point clouds. In the second part, we demonstrate approaches that generate a whole set of grasp hypotheses and rank them based on simulated grasp planning.

7.3.1 Evaluation of Object Centroid

A very simple but effective grasping strategy of unknown objects is to approach the object at its center. However, estimating the centroid is not a trivial problem when the object is unknown. Reactive grasping strategies are proposed to cope with the uncertainty in the object information during the grasp as for example by Hsiao et al. [99] and Felip and Morales [78]. However, contact between the object

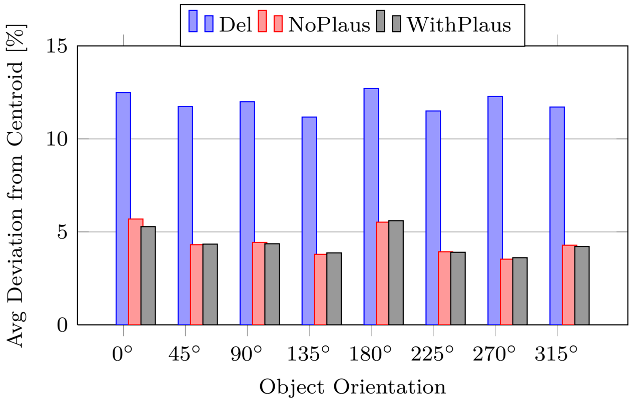


Figure 7.9: Deviation of Estimated Object Centroid from Ground Truth Centroid.

and the manipulator can move the object and change what kind of grasp is possible to apply. The more accurate the initial estimate of the object centroid, the fewer unnecessary contacts with the object occur.

In this section, we therefore evaluated the accuracy of object center estimation as a simple proxy for grasp quality. We compared the center of mass of the Delaunay mesh (baseline as described in Section 6.2.1.2) and of the mirrored mesh with the ground truth center. To render this comparison independent of the distribution of vertices (especially for the ground truth meshes), we applied the same uniform sampling of the mesh surface as in [189]. The center of mass is then the average over all the samples.

Figure 7.9 shows the error between the estimated and real centroid of an object per viewing direction and averaged over all objects. The deviation is normalized with the length of the diagonal of the oriented object bounding box. We can observe that the deviation ranges from approximately 5% to 10% of the total object size. However, using plausibility information does not improve the estimation of the centroid significantly.

These results indicate that the simple top-grasp strategy would have a larger success rate due to a more accurate estimation of the object centroid. In combination with reactive grasping approaches, we expect that fewer corrective movements would be necessary.

7.3.2 Generating Grasp Hypotheses

In this section, we want to propose two methods for generating a set of grasp hypotheses on the estimated surface mesh. In Chapter 8 we will show how these are used in conjunction with motion planning to select a suitable grasp and a reaching trajectory for execution by the robot.

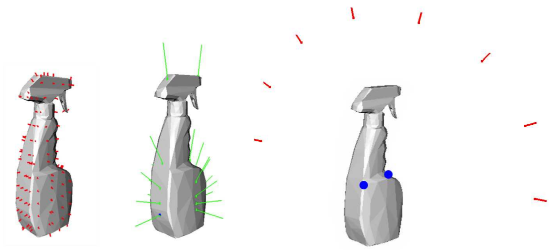


Figure 7.10: Example of the approach vectors generated for a spray bottle by (Left) the OpenRAVE grasper plugin, (Middle) the UJI proposed algorithm using the object's centroid and (Right) the KTH proposed algorithm using the grasp points in blue.

OpenRAVE [66] has a default algorithm to generate a set of approach vectors. It first creates a bounding box of the object and samples its surface uniformly. For each sample, a ray is intersected with the object. At each intersection, i.e. the grasping point, an approach vector is created that is aligned with the normal of the object's surface at this point. An example output is shown in Figure 7.10 (left). Dependent on the choice of parameters like distance between hand palm and grasping point, wrist alignment or pre-shape, time to simulate all the corresponding grasps can vary from few minutes to more than an hour. These execution times are acceptable for objects that are known beforehand because the set of grasp candidates can be evaluated off-line. When the objects are unknown, this process has to be executed on-line and long waiting times are not desirable.

For this reason, we use two methods to reduce the number of approach vectors. The first one, computes two contact points as described in Richtsfeld and Vincze [184]. To grasp the object at these points, there are an infinite number of approach vectors on a circle with the vector between the two contact points as its normal. We sample a given number, typically between 5 and 10 of these between 0° and 180° degrees. Figure 7.10 (Right) shows the detected grasping points along with the generated approach vectors. The second method, calculates the approach vectors in a similar way only that the center of the circle is aligned with the object's centroid and the major eigenvector e_a of its footprint, i.e., the projection of the point cloud onto the table. Another circle, perpendicular to the first one, is added in order to compensate the possible loss of vector quality due to the lack of grasp points. Figure 7.10 (middle) shows an example.

We applied these grasp hypotheses in a real-world scenario on different robotic platforms. For details on the execution, we refer to Chapter 8. For quantitative evaluation, one could compare grasp hypotheses generated on the completed point clouds to those planned on partial point clouds. This would involve executing these grasps on the corresponding ground truth meshes in simulation and evaluating common grasp stability measures.

7.4 Grasping Familiar Objects

As already outlined in Section 7.1, another approach towards grasp inference is to re-use prior grasping experience on familiar objects. A similarity metric between already encountered objects and new ones has to be developed that takes grasp-relevant features into account.

In the approach proposed in this section, a grasping point is detected based on the global shape of an object in a single image. This object feature serves as the basis of comparison to previously encountered objects in a machine learning framework. Research in the area of neuropsychology emphasizes the influence of global shape when humans choose a grasp [92, 58, 87]. Matching between stereo views is then used to infer the approach vector and wrist orientation for the robot hand. We further demonstrate how a supervised learning methodology can be used for grasping of familiar objects.

The contributions of our approach are:
i) We apply the concept of shape context as described in Belongie et al. [26] to the task of robotic grasping which to the best of our knowledge has not yet been applied for that purpose. The approach is different from the one taken in Saxena et al. [200] or Stark et al. [211] where only local appearance is used instead of global shape.
ii) We infer grasp configurations for arbitrarily shaped objects from a stereo image pair. These are the main difference to the work presented in Morales et al. [155] and Speth et al. [210] where either only planar objects are considered or three views from an object have to be obtained by moving the camera.
iii) We analyse how stable our algorithm is in realistic scenarios including background clutter without trained scenario examples as in Saxena et al. [200].
iv) We apply a supervised learning algorithm trained using synthetic labelled images from the database provided by [200]. We compare the classification performance when using a linear classifier (logistic regression) and a non-linear classifier (*Support Vector Machines* (SVMs)).

In the following, we will first describe the method of applying shape context to grasping is introduced. We also describe and comment on the database that we used for training and give some background knowledge on the two different classification methods. The section concludes with a presentation on how a whole grasp configuration can be derived. In Section 7.4.2 we evaluate our method both on simulated and real data.

7.4.1 Using Shape Context for Grasping

A detailed flow chart of the whole system and associated hardware is given in Figure 7.11. First, scene segmentation is performed based on a stereo input resulting in several object hypotheses. Shape context is then computed on each of the object hypotheses and 2D grasping points are extracted. The models of grasping points are computed beforehand through offline training on an image database. The points

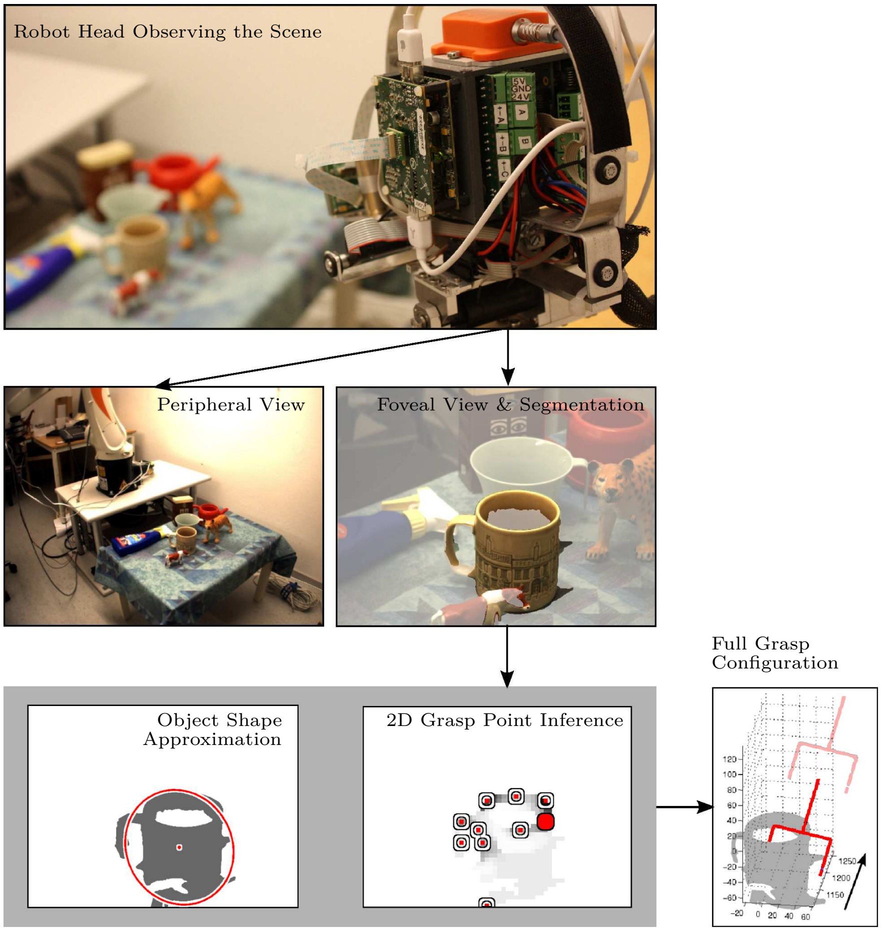
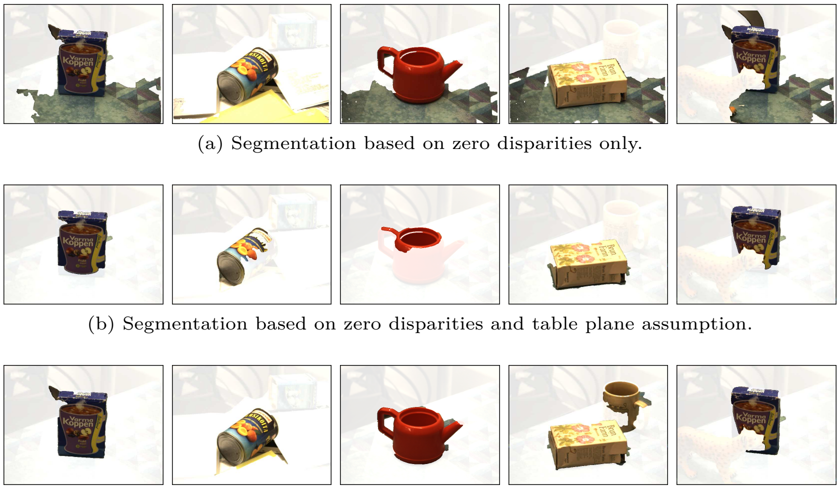


Figure 7.11: Flow Chart of Stereo Vision based Grasp Inference System

in the left and in the right image are associated to each other to infer a 3D grasping point via triangulation. In parallel to the grasping point detection, the segments are analysed in terms of rough object pose. By integrating the 3D grasping point with this pose, a full grasp configuration can be determined and then executed. In the following sections, the individual steps of the system are explained in more detail.



(a) Segmentation based on zero disparities only.

(b) Segmentation based on zero disparities and table plane assumption.

(c) Segmentation based on zero disparities, table plane assumption and known hue.

Figure 7.12: Segmentation results for: 1st column) One textured object. 2nd column) Cluttered table scene. 3rd column) Non-textured object. 4th column) Two similarly colored objects. 5th column) Occlusion.

7.4.1.1 Scene Segmentation

The system starts by performing the *figure-ground segmentation*. Although the problem is still unsolved for general scenes, we have demonstrated in our previous work how simple assumptions about the environment help in segmentation of table-top scenes, [34, 122, 32]. The segmentation is based on integration of stereo cues using foveal and peripheral cameras. This system is a predecessor of the real-time vision system shown presented in Section 2.2.1. In the below, we briefly outline the different cues used for segmentation and how they influence the result.

Zero-Disparity The advantage of using an active stereo head lies in its capability to fixate on interesting parts of the scene. A system that implements an attentional mechanism has been presented by Rasolzadeh et al. [182]. Once the system is in fixation, zero-disparities are employed as a cue for figure-ground segmentation through different segmentation techniques, e.g., water-shedding , Björkman and Eklundh [32]. The assumption is that neighboring pixels with a similar reconstructed depth belong to the same objects. However, Figure 7.12 shows that such a simple assumption results in bad segregation of the object from the plane on which it is placed.

Planar Surfaces The environment in which service robots perform their tasks are dominated by planar surfaces. In order to overcome the above segmentation problem, we use the assumption of a dominant plane. In our examples, this plane represents the table top on which objects are placed on. For that purpose, we fit a planar surface to the disparity image. The probability for each pixel in the disparity image to belong to that plane depends on its distance to it. In that way, objects on a plane are well segmented. Problems can arise with non-textured objects when the disparity image has large hollow regions. When the table plane assumption is violated through, e.g., clutter, the segmentation of the object is more difficult. Examples are shown in Figure 7.12.

Uniform Texture and Color An additional assumption can be made on the constancy of object appearance properties, assuming either uniformly colored or textured objects. When introducing this cue in conjunction with the table plane assumption, the quality of the figure-ground segmentation increases. The probability that a specific hue indicates a foreground object depends on the foreground probability (including the table plane assumption) of pixels in which it occurs. This holds equivalently for the background probability of the hue. The color cue contributes to the overall estimate with the likelihood ratio between foreground and background probability of the hue. The examples are shown in Figure 7.12. Judging from the examples and our previous work, we can obtain reasonable hypotheses of objects in the scene. In Section 7.4.2 and Section 7.4.2.3 we analyse the performance of the grasp point detection for varying quality of segmentation.

7.4.1.2 Representing Relative Shape

Once the individual object hypotheses are formed, we continue with the detection of grasping points. In Section 7.4.1.5 we show how to further infer the approach vector and wrist orientation. Grasping an object depends to a large extent on its global shape. Our approach encodes the global property of an object with a local, image-based representation. Consider for example elongated objects such as pens. A natural grasp is in its middle, roughly at the centre of mass. The point in the middle divides the object in two relatively similar shapes. Hence, the shape *relative* to this point is approximately symmetric. In contrast to that, the shape *relative* to a point at one of the ends of the object is highly asymmetric. Associating a point on the object with its *relative shape* and the natural grasp is the central idea of our work. For this purpose we use the concept of shape context commonly used for shape matching, object recognition and human body tracking, [93, 69, 158]. In the following, we briefly summarize the main ideas of shape context. For a more elaborate description, we refer to [26].

The basis for the computation of shape context is an edge image of the object. N sample points are taken uniformly from the contours, considering both inner and outer contours. For each point we compute the vectors that lead to all other sample points. These vectors relate the global shape of the object to the considered reference point. We create a compact descriptor comprising this information for each point by a two dimensional histogram with angle and radius bins. In [26] it is

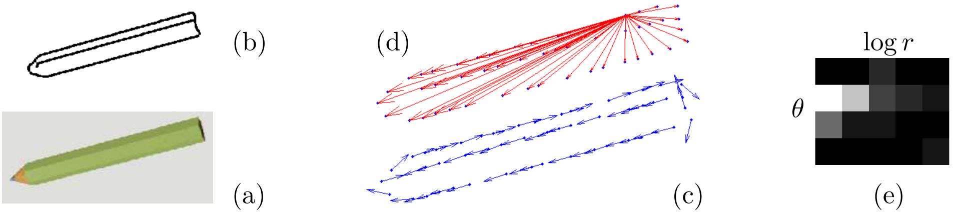


Figure 7.13: Example of deriving the shape context descriptor for the image of a pencil. (a) Input image of the pencil. (b) Contour of the pencil derived with the Canny operator. (c) Sampled points of the contour with gradients. (d) All vectors from one point to all other sample points. (e) Histogram with four angle and five log-radius bins comprising the vectors depicted in (d).

proposed to use a log-polar coordinate system in order to emphasize the influence of nearby samples. An example for the entire process is shown in Figure 7.13.

A big advantage of shape context is that it is invariant to different transformations. Invariance to translation is intrinsic since both the angle and the radius values are determined relative to points on the object. To achieve scale invariance, [26] proposed to normalise all radial distances by the median distance between all N^2 point pairs in the shape. Also rotation invariance can be easily achieved by measuring the angles relative to the gradient of the sample points. In the following, we will describe how to apply the *relative shape* representation to form a feature vector that can later be classified as either graspable or not.

Feature Vector In the segmented image, we compute the contour of the object by applying the Canny edge detector. This raw output is then filtered to remove spurious edge segments that are either too short or have a very high curvature. The result serves as the input for computing shape context as described above. We start by subdividing the image into rectangular patches of 10 × 10 pixels. A descriptor for each patch serves as the basis to decide whether it represents a grasping point or not. This descriptor is simply composed of the accumulated histograms of all sample points on the object's contour that lie in that patch. Typically, only a few sample points will be in a 10 × 10 pixel wide window. Furthermore, comparatively small shape details that are less relevant for making a grasp decision will be represented in the edge image. We therefore calculate the accumulated histograms in three different scales centered at the current patch. The edge image of the lowest scale then contains only major edges of the object. The three histograms are concatenated to form the final feature descriptor of dimension 120.

7.4.1.3 Classification

The detection of grasping points applies a supervised classification approach utilizing the feature vector described in the previous section. We examine two different classification methods: a linear one (logistic regression) and a non-linear one (SVMs), [31]. We describe these briefly below.

Logistic Regression Let g_i denote the binary variable for the ith image patch in the image. It can either carry the value 1 or 0 for being a grasping point or not. The posterior probability for the former case will be denoted as $p(g_i = 1|\mathbf{f}_i)$ where $\mathbf{f}_i$ is the feature descriptor of the ith image patch. For logistic regression, this probability is modelled as the sigmoid of a linear function of the feature descriptor:

$$p(g_i = 1|\mathbf{f}_i) = \frac{1}{1 + e^{-w\mathbf{f}_i}} \tag{7.7}$$

where w is the weight vector of the linear model. These weights are estimated by maximum likelihood:

$$w^* = \arg\max_{w'} \prod_i p(g_i = 1|\mathbf{f}_i, w') \tag{7.8}$$

where here g_i and $\mathbf{f}_i$ are the labels and feature descriptors of our training data, respectively.

Support Vector Machines SVMs produce arbitrary decision functions in feature space by a linear separation in a space of higher dimension compared to the feature space. The mapping of the input data into that space is accomplished by a non-linear kernel function k. In order to obtain the model for the decision function when applying SVMs, we solve the following optimisation problem:

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j g_i g_j k(\mathbf{f}_i, \mathbf{f}_j) \tag{7.9}$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i g_i = 0$ with the solution $w = \sum_i^{N_s} \alpha_i g_i \mathbf{f}_i$. As a kernel we have chosen a *Radial Basis Function* (RBF):

$$k(\mathbf{f}_i, \mathbf{f}_j) = e^{-\gamma||\mathbf{f}_i - \mathbf{f}_j||^2}, \gamma > 0 \text{ and } \gamma = \frac{1}{2\sigma^2} \tag{7.10}$$

The two parameters C and σ are determined by a grid search over parameter space. In our implementation, we are using the package libsvm [50].

Training Database For training the different classifiers we will use the database developed by Saxena et al. [200] containing ca. 12000 synthetic images of eight object classes depicted along with their grasp labels in Figure 7.14. One drawback of the database is that the choice of grasping points is not always consistent with the object category. As an example, a cup is labelled at two places on its rim but all the points on the rim are equally well suited for grasping. The eraser is quite a symmetric object. Neither the local appearance nor the relative shape of its grasping point are discriminative descriptors. This will be further discussed in Section 7.4.2 where our method is compared to that of [200].

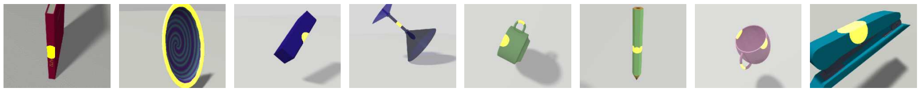


Figure 7.14: One example picture for each of the eight object classes used for training along with their grasp labels (in yellow). Depicted are a book, a cereal bowl, a white board eraser, a martini glass, a cup, a pencil, a mug and a stapler. The database is adopted from Saxena et al. [200].

7.4.1.4 Approximating Object Shape

Shape context provides a compact 2D representation of objects in monocular images. However, grasping is an inherently three-dimensional process. Our goal is to apply a pinch grasp on the object using a 3D coordinate of the grasping point and known position and orientation of the wrist of the robot. In order to infer the 6D grasp configuration, we integrate 2D grasping point detection with the 3D reconstruction of the object that results from the segmentation process.

We approach the problem by detecting the dominant plane $\hat{\Pi}^{\mathsf{D}} : d^{\mathsf{D}} = a^{\mathsf{D}}x + b^{\mathsf{D}}y + c^{\mathsf{D}}d$ in the disparity space $d = I^{\mathsf{D}}(x, y)$. The assumption is that the object can be represented as a constellation of planes in 3D, i.e. a box-like object has commonly three sides visible to the camera while for a cylindrical object, the rim or lid generates the most likely plane. We use RANSAC to estimate the dominant plane hypothesis $\hat{\Pi}^{\mathsf{D}}$ and we also determine its centroid μ^{D} by calculating the mean over all the points in the plane $\{(x, y)|e(x, y) > \theta\}$ where

$$e(x, y) = I^{\mathsf{D}}(x, y) - \frac{(d^{\mathsf{D}} - a^{\mathsf{D}}x - b^{\mathsf{D}}y)}{c^{\mathsf{D}}}, \tag{7.11}$$

the error between estimated and measured disparity of a point, and θ a threshold. By using the standard projective equations between image coordinates (x, y), disparity d, camera coordinates $(x^{\mathsf{C}}, y^{\mathsf{C}}, z^{\mathsf{C}})$, baseline b and the focal length f

$$x = f\frac{x^{\mathsf{C}}}{z^{\mathsf{C}}},\ y = f\frac{y^{\mathsf{C}}}{z^{\mathsf{C}}},\ d = \frac{bf}{z^{\mathsf{C}}}, \tag{7.12}$$

we can transform $\hat{\Pi}^{\mathsf{D}}$ into the camera coordinate frame:

$$\hat{\Pi}^{\mathsf{C}} : -c^{\mathsf{D}}bf = a^{\mathsf{D}}fx^{\mathsf{C}} + b^{\mathsf{D}}fy^{\mathsf{C}} - d^{\mathsf{D}}z^{\mathsf{C}}. \tag{7.13}$$

The normal of this plane is then defined as

$$\mathbf{n}^{\mathsf{C}} = (a^{\mathsf{C}}, b^{\mathsf{C}}, c^{\mathsf{C}}) = (a^{\mathsf{D}}f, b^{\mathsf{D}}f, -d^{\mathsf{D}}). \tag{7.14}$$

Equations 7.12 are also used to convert the plane centroid μ^{D} from disparity space to μ^{C} in the camera coordinate frame.

7.4.1.5 Generation of Grasp Hypotheses

In the following, we describe how the dominant plane provides the structural support for inferring a full grasp configuration.

3D Grasping Point The output of the classifier are candidate grasping points in each of the stereo images. These then need to be matched for estimation of their 3D coordinates. For this purpose we create a set $\mathcal{B}_l = \{\mathbf{b}_{(i,l)} | i = 1 \cdots M\}$ of M image patches $\mathbf{b}_{(i,l)}$ in the left image representing local maxima of the classifier $p(g_i = 1|\mathbf{f}_i)$ and whose adjacent patches in the 8-neighborhood carry values close to that of the centre patch. We apply stereo matching to obtain the corresponding patches $\mathcal{B}_r = \{\mathbf{b}_{(i,r)} | i = 1 \cdots M\}$ in the right image. Let $p(\mathbf{b}_{(i,l)}|\mathbf{f}_{(i,l)})$ and $p(\mathbf{b}_{(i,r)}|\mathbf{f}_{(i,r)})$ be the probability for each image patch in set $\mathcal{B}_l$ and $\mathcal{B}_r$ to be a grasping point given the respective feature descriptors $\mathbf{f}_{(i,l)}$ or $\mathbf{f}_{(i,r)}$. Assuming naïve Bayesian independence between corresponding patches in the left and right image, the probability $p(\mathbf{b}_i|\mathbf{f}_{(i,l)}, \mathbf{f}_{(i,r)})$ for a 3D point $\mathbf{b}_i$ to be a grasping point is modelled as

$$p(\mathbf{b}_{(i,l)}|\mathbf{f}_{(i,l)}, \mathbf{f}_{(i,r)}) = p(\mathbf{b}_{(i,l)}|\mathbf{f}_{(i,l)})\ p(\mathbf{b}_{(i,r)}|\mathbf{f}_{(i,r)}). \tag{7.15}$$

As already mentioned in the previous section, the approach vector and wrist orientation are generated based on the dominant plane. Therefore, the choice of the best grasping point is also influenced by the detected plane. For this purpose, we use the error $e(\mathbf{b}_{(i,l)})$ as defined in Equation 7.11 as a weight w_i in the ranking of the 3D grasping points. The best patch is then

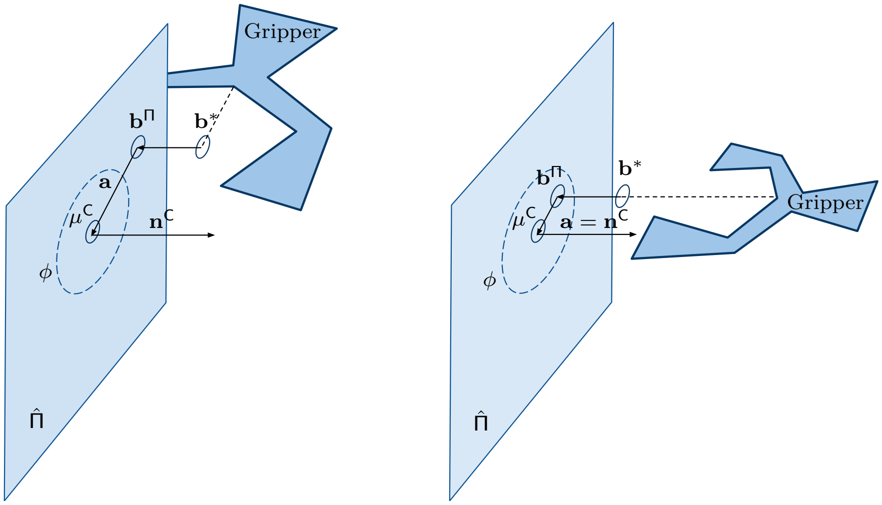
$$\mathbf{b}^* = \arg\max_i w_i\ p(\mathbf{b}_{(i,l)}|\mathbf{f}_{(i,l)}, \mathbf{f}_{(i,r)}). \tag{7.16}$$

Orientation of the Schunk Hand Given a 3D grasping point $\mathbf{b}^*$, the dominant plane $\hat{\Pi}^{\mathsf{C}}$ with its centroid μ^{C} and normal $\mathbf{n}^{\mathsf{C}}$, there are two possibilities for the approach vector $\mathbf{a}$:

1. $\mathbf{a} = \mathbf{v}^{\mathsf{C}}$ where $\mathbf{v}^{\mathsf{C}} = \mu^{\mathsf{C}} - \mathbf{b}^{\Pi}$ and $\mathbf{b}^{\Pi}$ is the projected grasping point on $\hat{\Pi}^{\mathsf{C}}$ along $\mathbf{n}^{\mathsf{C}}$

2. $\mathbf{a} = \mathbf{n}^{\mathsf{C}}$.

This is illustrated in Figure 7.15. Which of them is chosen depends on the magnitude of $\mathbf{v}^{\mathsf{C}}$. If $|\mathbf{v}^{\mathsf{C}}| > \phi$, the wrist of the hand is chosen to be aligned with the normal of the plane. If $|\mathbf{v}^{\mathsf{C}}| < \phi$, such that $\mathbf{b}^{\Pi}$ is very close to the centroid of the plane, we chose $\mathbf{n}^{\mathsf{C}}$ as the approach vector, i.e., the hand is approaching the dominant plane perpendicularly. In this case, we choose the wrist orientation to be parallel to the ground plane. In Section 8.1.2 we present results of object grasping on our hardware.



(a) Approach vector $\mathbf{a}$ is equal to $\mu^{\mathsf{C}} - \mathbf{b}^{\Pi}$. The wrist is aligned with $\mathbf{n}^{\mathsf{C}}$.

(b) Approach Vector $\mathbf{a}$ is equal to the normal. The wrist is aligned with the horizon.

Figure 7.15: Visualisation of the 6 DoF Grasp Configuration with respect to the estimated dominant plane $\hat{\Pi}$

7.4.2 Experimental Evaluation

We start by comparing our method to the one presented in [200]. The goal is to show the performance of the methods on synthetic images. This is followed by an in-depth analysis of our method. Finally, we investigate the applicability of our method in real settings.

7.4.2.1 Evaluation on Synthetic Images

In this section, we are especially interested in how well the classifiers generalize over global shape or local appearance given synthetic test images. For this purpose we applied four different sets of objects to train the classifiers.

- *Pencils* are grasped at their centre of mass.
- *Mugs & cups* are grasped at handles. They only differ slightly in global shape and local grasping point appearance.
- *Pencils, white board erasers & martini glasses* are all grasped approximately at their centre of mass at two parallel straight edges. However, their global shape and local appearance differ significantly.
- *Pencils & mugs* are grasped differently and also differ significantly in their shape.

We divided each set into a training and test set. On the training sets we trained four different classifiers.

- *Shape context & SVM* (SCSVM). We employed twelve angle and five log radius bins for the shape context histogram. We sample the contour with 300 points. The same parameters were applied by [26] and have proven to perform well for grasping point detection.
- *Local appearance features & logistic regression* (OrigLog) is the classifier by [200].
- *Local appearance features & SVM* (OrigSVM) applies an SVM instead of logistic regression.
- *Shape context, local appearance features & SVM* (SCOrigSVM) integrates shape context features with local appearance cues. The resulting feature vector is used to train an SVM.

Accuracy Each model was evaluated on the respective test sets. The results are shown as ROC curves in Figure 7.16 and as accuracy values in Table 7.2. Accuracy is defined as the sum of true positives and true negatives over the total number of examples. Table 7.2 presents the maximum accuracy for a varying threshold.

The first general observation is that SVM classification outperforms logistic regression. On average, the classification performance for each set of objects rose about 4.32% when comparing OrigSVM with OrigLog. A second general observation is that classifiers that employ global shape (either integrated or not integrated with appearance cues) have the best classification performance for all training sets. In the following we will discuss the results for each set.

- *Pencils.* The local appearance of a pencil does not vary a lot at different positions along its surface whereas relative shape does. Therefore, local appearance based features are not discriminative enough. This is confirmed for the models that are only trained on images of pencils. SCSVM performs slightly better than OrigSVM. The classification performance grows when applying an integrated feature vector.
- *Mugs & Cups.* These objects are grasped at their handle which is characterized by a local structure that is rather constant even when the global shape changes. Thus, OrigSVM outperforms slightly the classifier that applies shape context only. However, an integration of both features leads to an even better performance.
- *Pencils, white board erasers & martini glasses.* For this set of objects the position of the grasp is very similar when considering their global shape whereas the local appearance of the grasping points differs greatly. Also here, the models based on shape context performs best. Feature integration degrades the performance.

- *Pencils & mugs.* The performance of the different classifiers for the previous set of objects is a first indication for a weaker generalization capability of OrigSVM and OrigLog over varying local appearance compared to SCSVM and SCOrigSVM. This is further confirmed for the last set where not just local appearance but also global shape changes significantly. SCSVM improves the performance of OrigSVM about 6.75% even though the grasping points are very different when related to global object shape. Feature integration increases the performance only moderately.

Repeatability Our goal is to make a robot grasp arbitrary and novel objects. Thus, we are also interested in discovering whether the *best* grasping point hypotheses correspond to points that in reality afford stable grasps. Thus, our second experiment evaluates whether the best hypotheses are located on or close to the labelled points. We constructed a set of 80 images from the synthetic image database with ten randomly selected images of each of the eight object classes (Figure 7.14). Thus, also novel objects that were not used for training the different classifiers are considered. On every image we run all the aforementioned models and for each one picked out the best ten grasping points $\mathbf{b}_i$. In the database, a label is not a single point, but actually covers a certain area. We evaluated the Euclidean distance d_i of each of the ten grasping points measured from the border of this ground truth label at position $\mathbf{p}_j$ and normalized with respect to the length l_j of its major axis. This way, the distance is dependent on the scale of the object in the image. In case there is more than one label in the image, we choose the one with the minimum distance. If a point $\mathbf{b}_i$ lies directly on the label, the distance $d_i = 0$. If a point lies outside of the label, the distance d_i gets weighted with a Gaussian function $(\sigma = 1, \mu = 0)$ multiplied with $\sqrt{2\pi}$. The number of hits h_m of each model m on the picture set is counted as follows:

$$\begin{aligned} h_m &= \sum_{k=1}^{K} \sum_{i=1}^{N_k} e^{-\frac{d^2_{(i,k)}}{2}} \\ \text{with } d_{(i,k)} &= \min_{j=1}^{M_k} \frac{dist(\mathbf{b}_{(i,k)}, \mathbf{p}_{(j,k)})}{2 l_{(j,k)}} \end{aligned}$$

where K is the number of images in the set, M_k is the number of grasp labels in that picture and N_k is the number of detected grasping points. Grasping points whose distance d_i exceed a value of $3 * \sigma$ are considered as outliers. Figure 7.17 shows the number of hits, that is, the amount of good grasps for each model.

Apart from the model trained on cups and mugs, the SVM trained only on shape context always performs best. The performance drop for the second object set can be explained in the same way as in the previous chapter: handles have a very distinctive local appearance and are therefore easily detected with features that capture this. In general, this result indicates that classifiers based on shape context detect grasping points with a better repeatability. This is particularly important

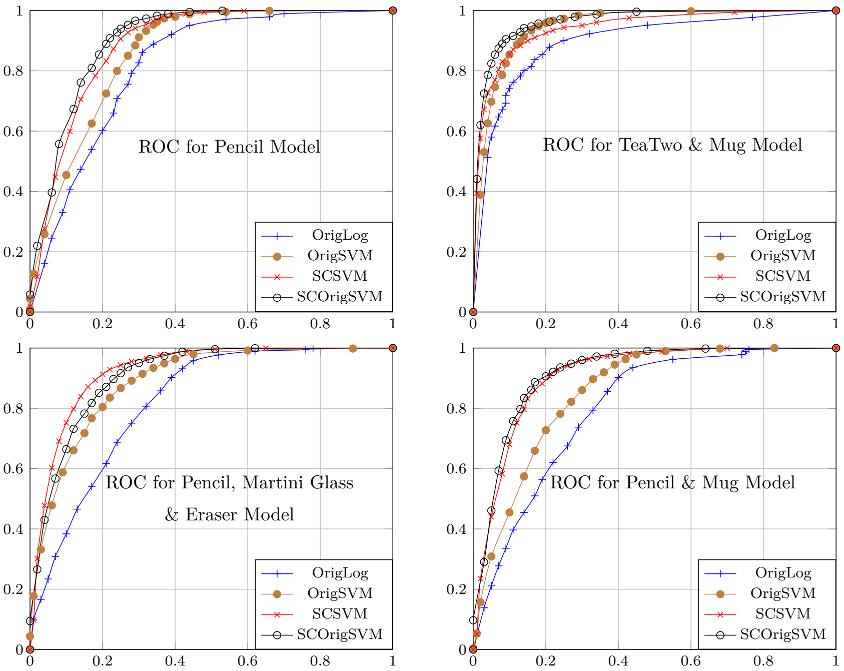


Figure 7.16: ROC curves for models trained on different objects.

Table 7.2: Accuracy of the models trained on different objects.

	SCOrigSVM	SCSVM	OrigSVM	OrigLog
Pencil	**84.45%**	82.55%	80.16%	77.07%
Cup & Mug	**90.71%**	88.01%	88.67%	83.85%
Pencil, Martini & Eraser	84.38%	**85.65%**	80.79%	74.92%
Pencil & Mug	**85.71%**	84.64%	77.80%	74.32%

for the inference of 3D grasping points in which two 2D grasping points in the left and right image of a stereo camera have to be matched.

Summary of Results We draw several conclusions regarding experimental results on synthetic images. First, independent of which feature representation is chosen, SVM outperforms logistic regression. Secondly, our simple and compact

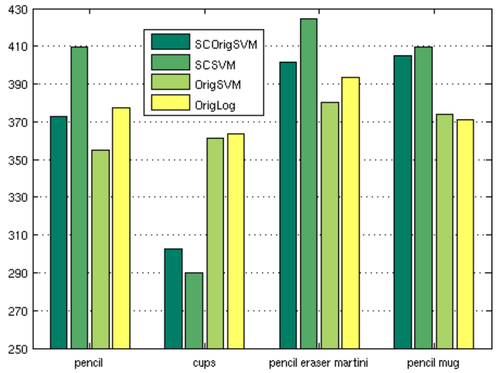


Figure 7.17: Evaluation of the best ten grasping points of each model on a picture test set containing in total 80 pictures of *familiar* and novel objects (see Figure 7.14).

feature descriptor that encodes relative object shape improves the detection of grasping points both in accuracy and repeatability in most cases. In case of very distinct local features, both representations are comparable. Integration of the two representations leads only to moderate improvements or even decreases the classification performance.

7.4.2.2 Opening the Black Box

In the previous section, we presented evidence that the shape-context–based models detect grasping points more accurately than the models trained on local appearance features. As argued in Section 7.4.1, we see relative shape as a better cue for graspability than local appearance. In this section, we would like to confirm this intuition by analyzing what the different grasping point models encode. We are especially interested in whether there are prototypical visual features on which the two different models fire. We conduct this analysis by applying the Trepan Algorithm by Craven and Shavlik [56] to the learnt classifiers. This algorithm builds a decision tree that approximates a concept represented by a given *black box* classifier. Although originally proposed for neural networks, Martens et al. [144] showed that it is also applicable for SVMs.

We use the same sets of objects as mentioned in the previous section. The extracted trees are binary with leafs that are classifying feature vectors as either graspable or non-graspable. The decisions at the non-leaf nodes are made based on either one or more components of the feature vector. We consider each *positive* leaf node as encoding a prototypical visual feature that indicates graspability. As previously mentioned, the extracted trees are only approximations of the actual learnt models. Thus, the feature vectors that end up at a specific leaf of the tree will be of three different kinds:

- *Ground truth.* Features that are graspable according to the ground truth labels in the database.
- *False positives by model.* Features that are not graspable according to the labels but are so according to the classifier.
- *False positives by tree.* Features that are neither labelled in the database nor classified by the model to be graspable, but are considered to be so by the tree.

We will analyse these samples separately and also rate the trees by stating their *fidelity* and *accuracy*. *Fidelity* is a measure of how well the extracted trees approximate the considered models. It states the amount of feature vectors whose classification is compliant with the classification of the approximated model. *Accuracy* measures the classification rate for either the tree or the model when run on a test set.

The analysis of these samples is conducted using PCA. The resulting eigenvectors form an orthonormal basis with the first eigenvector representing the direction of the highest variance, the second one the direction with the second largest variance, etc. In the following sections we visualise only those eigenvectors whose *energy* is above a certain threshold and at maximum ten of these. The *energy* e_i of an eigenvector $\mathbf{e}_i$ is defined as

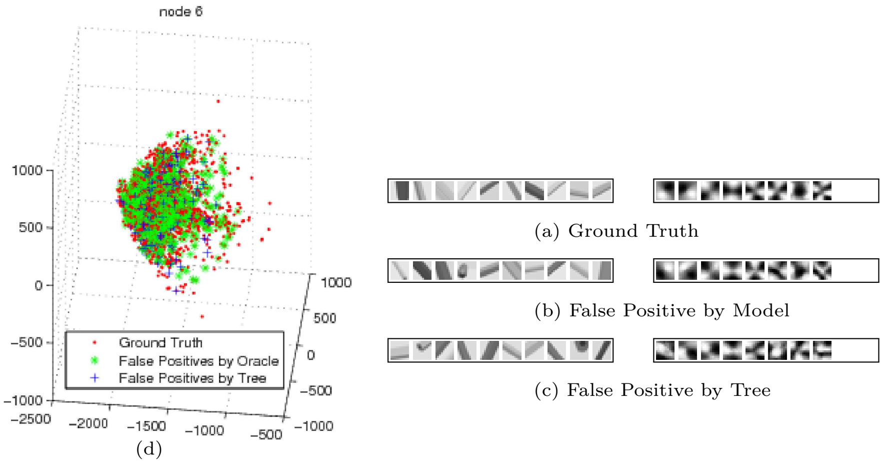
$$e_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{k} \lambda_j} \tag{7.17}$$

where λ_j is the eigenvalues of eigenvector $\mathbf{e}_j$ with K eigenvectors in total. As a threshold we use $\theta = 0.9$.

The remainder of this section is structured as follows. In Section 7.4.2.2, we visualise the prototypical features for the local appearance method by applying PCA to the samples at positive nodes. In Section 7.4.2.2 we do the same for the relative shape-based representation.

Local Appearance Features Saxena et al. [200] applied a filter bank to 10×10 pixel patches in three spatial scales. The filter bank contains edge, texture (Law's masks) and color filters. In this section, we depict samples of these 10×10 pixel patches in the largest scale. They are taken from every positive node of each tree trained for a specific object set. All feature vectors that end up at one of these positive nodes are used as an input to PCA.

The first set we present consists of images of a pencil (see Figure 7.14) labelled in its centre of mass. The built tree is rather shallow: it has only four leaf nodes of which one is positive. The decisions on the non-leaf nodes are made based on the output of the texture filters only. Neither color nor edge information are considered. This means that this part of the feature vector is not necessary to achieve a classification performance of 75.41% (see Table 7.3). Ten random samples from the positive node are shown in Figure 7.18(a)-(c) subdivided dependent on



(a) Ground Truth

(b) False Positive by Model

(c) False Positive by Tree

(d)

Figure 7.18: Pencils:(a)-(c) Ten samples and PCA components of the positive node of the decision tree. (d) Feature vectors projected into the three-dimensional space spanned by the three eigenvectors of the sample set of true grasping points with the highest variance.

whether they are graspable according to the ground truth labels from the database or only according to the model and tree, respectively.

In order to visualise to which visual cues these grasping point models actually respond, we run PCA on the set of feature vectors that ended up at that node. The resulting principal components selected according to Equation 7.17 are also depicted in Figure 7.18 (a)-(c). Encoded are close-ups of the body of the pencil and perspective distortions.

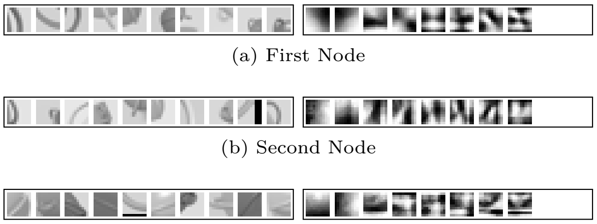
However, the majority of the pencil complies with these components. Because of that, the samples from the set of false positives are very similar to the ground truth samples. The appearance of the centre of mass is not that different from the rest of the pencil. This is further clarified by Figure 7.18 where the false positives by the model and tree are projected into the space spanned by the first three principal components from the ground truth: they are strongly overlapping. We will show later that given our relative shape based representation these three first principal components are already enough to define a space in which graspable points can be better separated from non-graspable points.

For the other sets of objects we applied the same procedure. The principal components of the samples at each positive node are shown in Figure 7.19, 7.20 and 7.21. In Table 7.3, the fidelity of the respective trees in relation to the model and their accuracies are given.

Relative Shape In this section we evaluate the performance of the shape context in the same manner as the local appearance features were tested in the previous

Table 7.3: Accuracy of the models trained on different objects given the local appearance representation.

	Pencil	Cups	Elongated	Pencil & Mug
Fidelity	86.78%	83.97%	87.55%	89.29%
Accuracy Tree	75.41%	82.61%	72.12%	73.30%
Accuracy Model	77.07%	83.85%	74.92%	74.32%

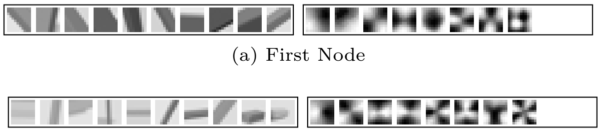


(a) First Node

(b) Second Node

(c) Third Node

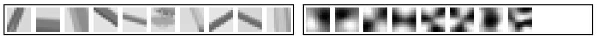
Figure 7.19: Cups: Ten samples and PCA components for each of the positive nodes of the decision tree.



(a) First Node

(b) Second Node

Figure 7.20: Elongated Objects: Ten samples and PCA components for each of the positive nodes of the decision tree.



(a) First Node

Figure 7.21: Pencils and Mugs: Ten samples and PCA components for the positive node of the decision tree.

Table 7.4: Accuracy of the models trained on different objects given the relative shape representation.

	Pencil	Cups	Elongated	Pencil & Mug
Fidelity	78.97%	79.66%	78.79%	80.82%
Accuracy Tree	71.38%	76.89%	73.40%	73.41%
Accuracy Model	82.55%	88.01%	85.56%	84.64%

section. The process includes

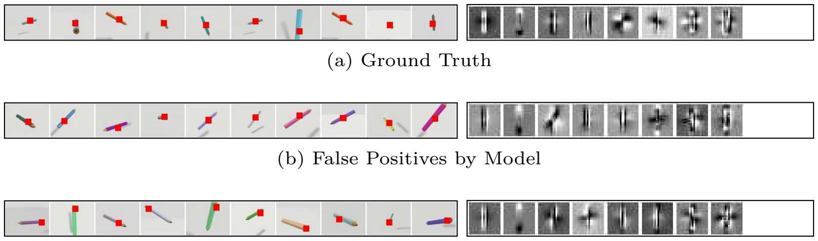
1. extracting the contour with the Canny edge detector,
2. filtering out spurious edge segments,
3. subsampling the contour,
4. normalizing the sampled contour with the median distance between contour points,
5. rotating the whole contour according to the average tangent directions of all the contour points falling into the patch that is currently considered by the classifier
6. and finally plotting the resulting contour on a 20 × 20 pixels patch with the grasping point in the centre.

The output of this procedure forms the input for PCA. The sample feature vectors for each node are depicted not as patches but as red squared labels located at the grasping point on the object.

Each of the induced trees in this section is of a slightly worse quality in terms of fidelity when compared with the trees obtained from the logistic regression method (see Table 7.3). We reason that this is due to the performance of the Trepan algorithm when approximating SVMs. Nevertheless, the purpose of this section is the visualisation of prototypical grasping point features rather than impeccable classification. This performance is therefore acceptable. The results for the induced trees are given in Table 7.4.

We start by analyzing the model trained on the set of pencils. The induced decision tree has one positive node. The samples from this node are depicted in Figure 7.22 along with the most relevant PCA components to which we will refer in the remainder of this section as *eigencontours*. These components do not encode the local appearance but clearly the symmetric relative shape of the grasping point.

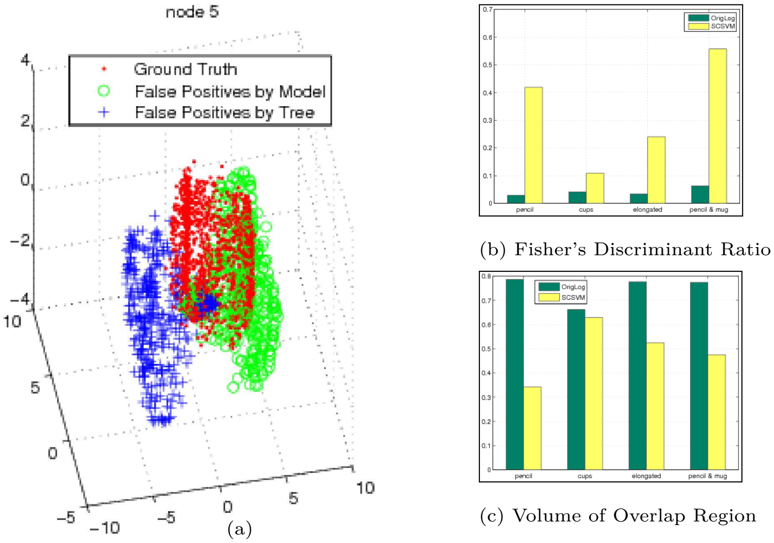
One interesting observation is that the feature vectors projected into the space spanned by the three best principal components of the ground truth samples are


(a) Ground Truth

(b) False Positives by Model

(c) False Positives by Tree

Figure 7.22: Pencil: Ten samples and PCA components of the positive node of the decision tree.



(a)

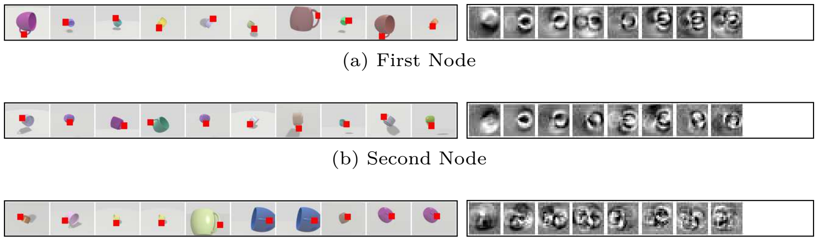
(b) Fisher's Discriminant Ratio

(c) Volume of Overlap Region

Figure 7.23: (a)Pencil: Feature vectors projected into the three-dimensional space spanned by the three eigenvectors of the sample set of true grasping points with the highest variance. (b) and (c) measure linear separability for models trained on different training sets and with different classification methods. Dark: OrigLog. Bright: SCSVM.

quite well separable, even with a linear decision boundary. There is almost no overlap between false positives produced by the tree and the ground truth features and little overlap between false positives produced by the models and the true graspable features. This result is shown in Figure 7.23.

We applied the same procedure to the models trained on the other sets of objects. The eigencontours for these are shown in 7.24, 7.25 and 7.26. For the sets consisting of different objects, each positive node in the decision tree is mainly associated with

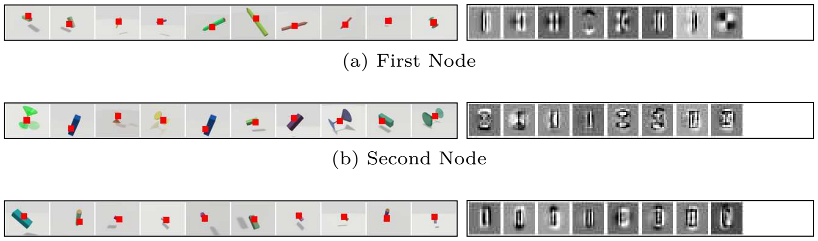


(a) First Node

(b) Second Node

(c) Third Node

Figure 7.24: Cups: Ten samples and PCA components for each of the positive nodes of the decision tree.



(a) First Node

(b) Second Node

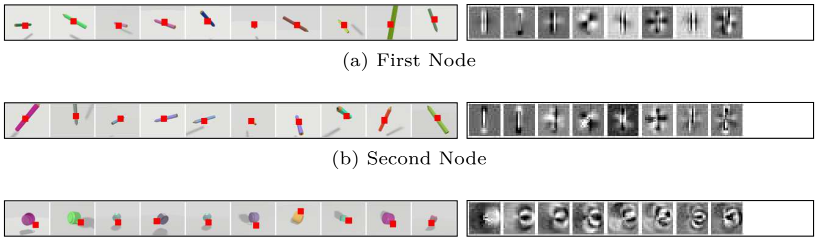
(c) Third Node

Figure 7.25: Elongated: Ten samples and PCA components for each of the positive nodes of the decision tree.

one of the objects and encodes where they are graspable.

Furthermore, we can observe a better separability compared to the models trained on local appearance. In order to quantify this observation, we analysed the distribution of the samples in the three-dimensional PCA space in terms of linear separability. As measures for that we employed Fisher's discriminant ratio and the volume of the overlap regions. Figure 7.23 (b) and (c) show a comparative plot of these two measures for all the models considered in this section.

Summary of Results The evaluation provided a valuable insight into different feature representations. We observed that our compact feature descriptor based on relative shape is more discriminative than the feature descriptor that combines the output of a filter bank. The dimensionality of our descriptor is almost four times smaller which also has implications for the time needed to train an SVM. The classification performance achieved with an SVM could even be improved by finding a decision boundary in the space spanned by the first three principal components of a set of ground truth prototypical features.



(a) First Node

(b) Second Node

(c) Third Node

Figure 7.26: Pencils and Mugs: Ten samples and PCA components of the first positive node of the decision tree.

7.4.2.3 Evaluation on Real Images

In the previous section, we showed that the performance of the relative-shape–based classifier is better compared to a method that applies local appearance. In these synthetic images no background clutter was present. However, in a real-world scenario we need to cope with clutter, occlusions, etc. One example is presented by Saxena et al. [200] in which a robot is emptying a dishwasher. In order to cope with the visual clutter occurring in such a scenario, the grasping-point model was trained on hand-labelled images of the dishwasher. Although the dishwasher was emptied successfully, for a new scenario the model has to be re-trained to cope with new backgrounds.

We argue that we need a way to cope with backgrounds based on more general assumptions. As described earlier in Section 7.4.1.1, our method relies on scene segmentation. In this section, we evaluate how the relative shape based representation is affected by different levels of segmentation. For that purpose, we collected images of differently textured and texture-less objects, e.g., boxes, cans, cups, elongated objects, or toys, composed in scenes of different levels of complexity. The example scenes range from single objects on a table to several objects occluding each other. These scenes were segmented with the three different techniques described in Section 7.4.1.1.

Ideally, we would like to achieve two things. First is the repeatability: the grasping points for the same object given different qualities of segmentation have to match. Second is the robustness: the grasping points should be minimally affected by the amount of clutter. Regarding the latter point, a quantitative evaluation can only be performed by applying the inferred grasps in practice. Thus, we demonstrate our system on real hardware in Section 8.1.2 and present here some representative examples of the grasping point inference methods when applied to different kinds of objects situated in scenes of varying complexity.

Examples for Grasping Point Detection In Figure 7.27, we show the results of the grasping point classification for a teapot. The left column shows the segmented input of which the first one is always the ground truth segment. The middle column shows the result of the grasping point classification when applying the local appearance based descriptor by [200] and the right one the results of the classification when using the relative shape-based descriptor. The red dots label the detected grasping points. They are the local maxima in the resulting probability distribution. Maximally the ten highest valued local maxima are selected.

The figure shows the grasping point classification when the pot is the only object in the scene and when it is partially occluded. Note that the segmentation in the case of local appearance-based features is only influencing which patches are considered for the selection of grasping points. In case of the relative shape-based descriptor, the segmentation also influences the classification by determining which edge points are included in the shape context representation. Nevertheless, what we can observe is that the detection of grasping points for the representation proposed in this approach is quite robust. For example in Figure 7.27b (last row), even though there is a second handle now in the segmented region, the rim of the teapot is still detected as graspable and the general resulting grasping point distribution looks similar to the cases in which the handle was not yet in the segment. This means that the object the vision system is currently in fixation on, the one that dominates the scene, produces the strongest responses of the grasping point model even in the presence of other graspable objects.

In Figure 7.28a, we applied the models trained on mugs and cups to images of a can and a cup. The descriptor based on local appearance responds very strongly to textured areas whereas the relative shape based descriptor does not get distracted by that since the whole object shape is included in the grasping point inference. Finally in Figure 7.28b, we show an example of an object that is not similar to any object that the grasping point models were trained on. In case of the local appearance based descriptor, the grasping point probability is almost uniform and very high-valued. In the case of shape context there are some peaks in the distribution. This suggests that the ability of these models to generalize over different shapes is higher than for local appearance-based models.

Repeatability of the Detection One of the goals of the method is the repeatability of grasping point detection. In order to evaluate this, we measured the difference of the detected grasping points in the differently segmented images. For real images, we do not have any ground truth labels available as in the case of synthetic data. Thus, we cannot evaluate the grasp quality as done in Section 7.4.2.1. Instead, we use the detected grasping points in a manually segmented image as a reference to quantify the repeatability of the grasping point detection.

We have a set $\mathcal{B} = \{\mathbf{b}_i \| i = 1 \ldots N\}$ of pictures and three different cues based on which they are segmented: zero disparity, a dominant plane and hue. If we want to measure the difference d_{b_i} between the set of grasping points $\mathcal{G}_{b_i} = \{\mathbf{g}_{(b_i,j)} \| j = 1 \ldots M\}$ and the set of reference points $\mathcal{G}_{b_i} = \{\mathbf{g}_{(b_i,r)} \| k = 1 \ldots R\}$ for a specific kind of segmentation of the image $\mathbf{b}_i$, then

$$\begin{aligned} d_{b_i} &= \frac{1}{K}\sum_{j=1}^{M} e^{\frac{-d_j^2}{2}} \text{ where} && (7.18)\\ d_j &= \min_{r=1}^{R} dist(\mathbf{g}_{(b_i,r)}, \mathbf{g}_{(b_i,j)}) && (7.19)\end{aligned}$$

where *dist* is the Euclidean distance and K the length of the image diagonal[3]. The mean and standard deviation of d_{b_i} for all images in the set $\mathcal{B}$ that are segmented with a specific cue is then our measure of deviation of the detected from the reference grasping points.

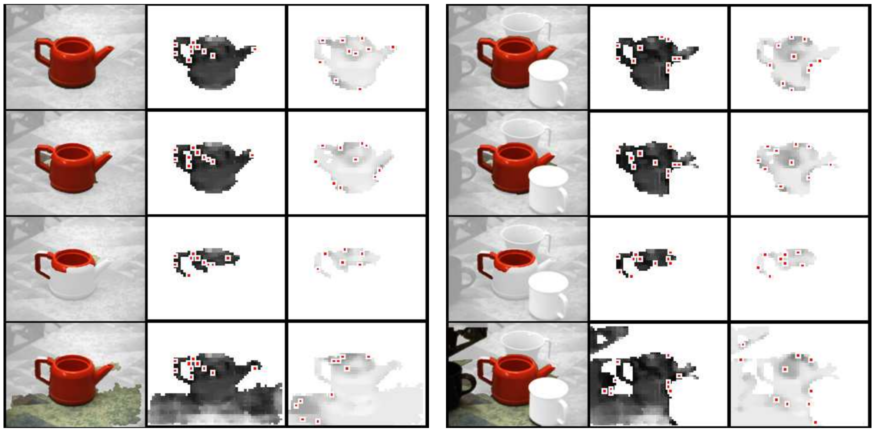
In Figure 7.29 we show this measure for a representative selection of objects and models. As already mentioned, ideally we would like to see no difference between detected grasping points when facing different qualities of segmentation. In practice, we can observe a flat slope. As expected for both methods, the grasping points detected in the image segmented with zero-disparity cues are the ones that are deviating most from the reference points. Although, the selection of points that are included in our representation is directly influenced by the segmentation, the difference between detected and reference grasping points is not always bigger than for the appearance based method. In fact, sometimes it performs even better. This holds for examples of the models trained on mugs and cups for which both methods show a similar accuracy on synthetic data (Figure 7.29 (a) and (b)). If the models are applied to novel objects, as can be observed in Figure 7.29 (c), our descriptors shows a better repeatability. This suggests again a better capability of the models to generalize across different relative shapes. In general, we can say that both methods are comparable in terms of repeatability.

Summary of Results In this section, we evaluated the performance of our approach on real images. Due to the encoding of global shape, the method is robust against occlusions and strong texture. Although our representation is strongly dependent on the segmentation, we observe that the repeatability of grasping points is comparable to the local appearance based method even when facing imperfect segmentation. The analysis included images of varying qualities of segmentation as well as occlusion and clutter.

7.5 Discussion

In this chapter, we reviewed approaches towards generating grasp hypotheses given different kinds of prior object knowledge. Our proposed taxonomy is inspired by research in neuropsychology and classifies them dependent on whether they assume

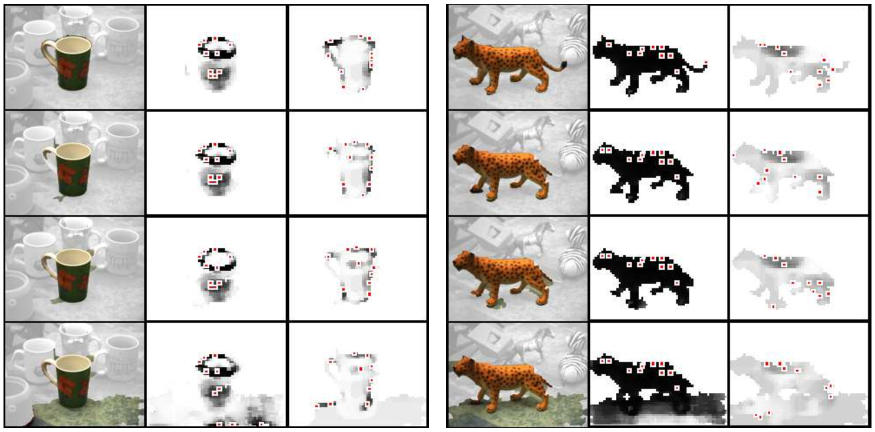
[3] In our case $K = 80$ since we are evaluating 10×10 pixel patches in images of size 640×480 pixels



(a) Single Teapot

(b) Partly Occluded Teapot

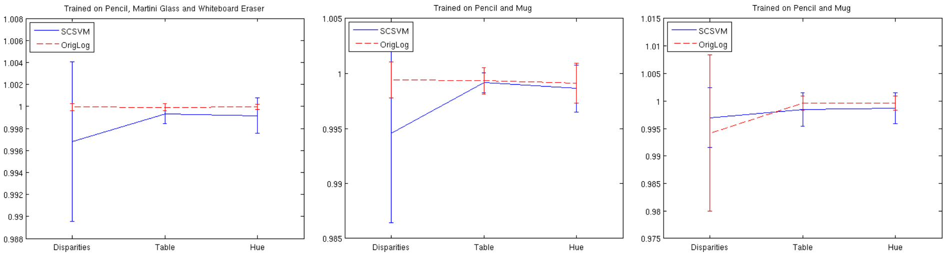
Figure 7.27: Grasping point model trained on mugs and pencils applied to a textureless teapot. The darker a pixel, the higher is the probability that it is a grasping point.



(a) Grasping point model trained on mugs and cups applied to a textured cup.

(b) Grasping point model trained on pencils, martini glasses and whiteboard eraser applied to a novel object.

Figure 7.28: The darker a pixel, the higher is the probability that it is a grasping point.



(a) Set of Cans (b) Set of Textureless Tea Pots (c) Set of Novel Objects

Figure 7.29: Comparing the stability of grasp point detection of SCSVM and OrigLog for different sets of objects and different grasping point models when facing imperfect segmentation.

known, familiar or unknown objects. For each of these cases, we have proposed our own methods. As input, most of them rely on a set of segmented object hypotheses.

For the case of known objects, we showed three different methods that fit object models to the sensory data. They are presented in the order of decreasing amount of assumptions. Thereby, they can cope with an increasing scene complexity. Furthermore, we found that using completed object models leads to increased robustness for joint object recognition and pose estimation.

For unknown objects, we used the results from Chapter 6 in which the 3D model of an unknown object is estimated from a partial point cloud. Grasp hypotheses are generated based on the completed object model. For familiar objects, we proposed an approach that combines 2D and 3D shape cues of an object. A machine learning approach is used to transfer grasping experience from objects encountered earlier. In the next chapter, we will show how the inferred grasp hypotheses can be executed by a real robot.

The majority of the grasp synthesis systems that we reviewed in this chapter including our own approaches can be easily positioned within the proposed taxonomy. In turn this means that given an object hypothesis, a prior assumption is made that either there is a corresponding object model in the database and just needs to be found (known objects), that a similar object has been grasped before and the grasp experience can be transferred (familiar objects) or that the object is unknown and dissimilar to any of the objects previously encountered. Only a few systems consider the problem of deciding in a flexible way what kind of method should be applied here. From the systems that we reviewed in this thesis, these are developed by Ciocarlie et al. [52] and Marton et al. [149] where a grasp planner for unknown objects is used for the object hypotheses that do not match previously encountered objects.

8

Grasp Execution

Many of the approaches that we reviewed in Section 7.1 for generating grasp hypotheses are not demonstrated on a robot. For those who are, most of them employ an open-loop controller. An open-loop controller computes its actions taking only the current state into account without observing the outcome of these actions and whether they reached the desired goal.

However, realistic applications require going beyond open-loop execution to deal with different types of errors occurring in an integrated robotic system. One kind of errors are systematic and repeatable, introduced mainly by inaccurate kinematic models. These can be minimized offline through precise calibration. The second kind are random errors introduced by a limited repeatability of the motors or sensor noise. These have to be compensated online through for example closed-loop controllers. In a grasping system, this can be achieved through for example visual servoing or reactive grasping approaches using force-torque or tactile sensors. In these cases, the output of actions is observed during execution and actions are adapted to achieve the desired goal.

In this chapter, we will first demonstrate open-loop grasping given grasp hypotheses from either familiar or unknown objects. This will serve as an example for motivating techniques for closed-loop execution of grasps. Different methods will be reviewed. This will be followed by a demonstration of different visual servoing techniques applied in our grasping pipeline for known and unknown objects.

8.1 Open Loop Grasp Execution

As an open-loop grasp controller, we consider a system that

- uses any of the grasp synthesis systems reviewed in the previous chapter to compute a set of grasp hypotheses for an object,
- computes a trajectory to reach the desired grasp pose and
- executes this trajectory without any adaptation to, e.g., changes in the environment.

We employ such a controller to execute grasps for unknown and familiar objects.

Algorithm 4: Pseudo Code for a Table Top Scenario

```
Data: Embodiment, Table Plane
Result: Cleaned Table Top
begin
    $\mathcal{S}$ = GetObjectHypotheses()
    // Grasp Cycle per Object Hypothesis
    for $i = 0; i < |\mathcal{S}|; i++$ do
        $s_i^*$ = PredictObjectShape($s_i$)
        $s_i^*$ = ConvexDecomposition($s_i$)
        $\mathcal{G}$ = GetGraspCandidates($s_i^*$)
        for $j = 0; j < |\mathcal{G}|; j++$ do
            success = PlanGrasp($g_j$)
            if success then
                // Grasp Candidate gives Stable Grasp
                $t_{(i,j)}$ = PlanArmTrajectory($g_j$)
                if $t_{(i,j)} \neq 0$ then
                    // Collision-Free Trajectory Exists
                    $\mathcal{T}$ = InsertTrajectory($\mathcal{T}, t_{(i,j)}, g_j$)
                end
            end
        end
        $k = 0$
        repeat
            success = ExecuteGrasp($T, k$)
            $k++$
        until success
    end
end
```

8.1.1 Executing Grasps on Unknown Objects

In this section, we consider the scenario of grasping unknown objects from a table top. We apply the method of predicting unknown object shape from partial point clouds as proposed in Chapter 6.

One grasp cycle for this scenario is outlined in Algorithm 4. As a step prior to the shape analysis of an object and grasping it, the robot needs to segregate potential objects from the background. The function `GetObjectHypotheses` to obtain a segmented point cloud is implemented as explained in Section 2.2.1. The function `PredictObjectShape` refers to the shape estimation as described in Chapter 6. With the complete object shape as an input, `GetGraspCandidates` implements the techniques outlined in Section 7.1.2. These grasp candidates are then simulated on the predicted object shape and a collision-free path for the arm planned.

The simulation platform chosen to perform the experiments is OpenGRASP [7], a simulation toolkit for grasping and dexterous manipulation. It is based on OpenRAVE [66], an open architecture targeting a simple integration of simulation, visualization, planning, scripting and control of robot systems. It enables the user to easily extend its functionality by developing their own custom plugins.

The simulator is used to perform the grasp before the real robot makes an attempt. It allows for testing different alternatives, choosing the one with the highest probability of success. This will not only take considerably less time than performing it with the real hardware but also prevents damaging the robot by avoiding collisions with the environment.

For collision detection necessary for motion planning, the simulator needs complete object models. When known objects are used, an accurate model of the objects can be created off-line using different technologies, like laser scans. In the case of unknown objects, an approximate model has to be created on-line. As mentioned before, in this section we propose to use the approach described in Chapter 6 to estimate a complete 3D mesh model. The obtained triangular mesh has thousands of vertices which makes the collision detection process computationally expensive. To increase efficiency, we pre-process the mesh by `ConvexDecomposition`, a library that was originally created by Ratcliff [183] and that is implemented in OpenRAVE. It approximates a triangular mesh as a collection of convex components. This process takes only a few seconds and drastically speeds up the grasp and motion planning.

8.1.1.1 Generation of Grasp Candidates

For selecting an appropriate grasp from the set of grasp hypotheses generated through `GetGraspCandidates`, each of them is simulated moving the end-effector until it collides with the object; then the fingers close around it and finally the contacts are used to test on force closure. In Algorithm 4, this process is referred to as `PlanGrasp`.

Each grasp hypothesis has the following parameters: the approach vector, the hand pre-shape, the approach distance and the end-effector roll. The number of different values that these parameters can take has to be chosen considering the time constraints imposed by the on-line execution of `PlanGrasp`. As a hand pre-shape, we defined a pinch grasp for each hand. The approach distance is varied between 0 to 20 cm. The roll is chosen dependent on the two grasping points (such that the fingers are aligned with them) or on the orientation of the circle on which the selected approach vector is defined. Compared to the sampling of grasp candidates based on normal direction, the amount of time taken to generate and save the set of grasp candidates is reduced significantly to less than a minute.

8.1.1.2 Grasp Execution Using Motion Planners

The next step `PlanArmTrajectory` in Algorithm 4, consists of selecting a stable grasp from the set of grasp candidates that can be executed with the current robot configuration without colliding with obstacles. For each stable grasp, it first moves the robot to the appropriate grasp pre-shape, then uses RRT and Jacobian-based gradient descent methods [225] to move the hand close to the target object, closes the fingers, grabs the object, moves it to the destination while avoiding obstacles and releases it.

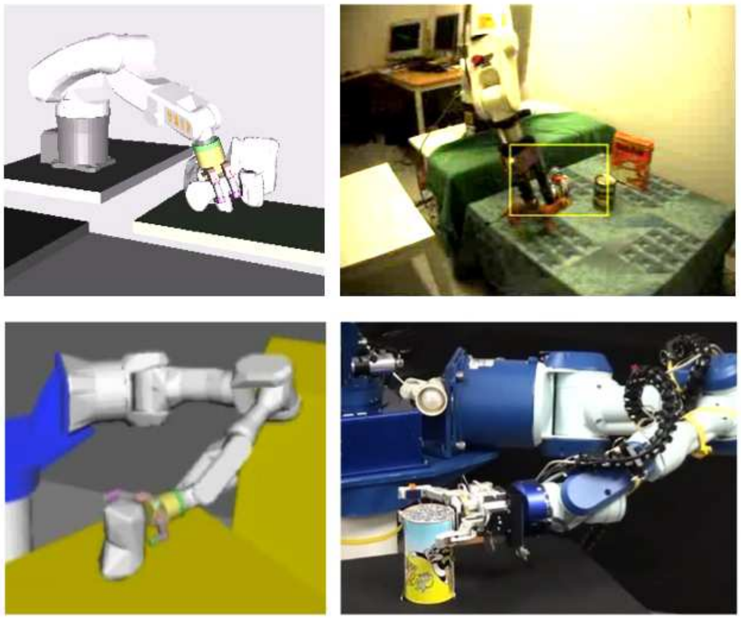


Figure 8.1: Example of the grasp performed by the simulated robot and the real one, using Top) KTH platform and Bottom) UJI platform.

If the robot successfully grabs the object and moves it to the destination, the stable grasp is returned for execution with the real robot. Otherwise, the next stable grasp from the set is tried. Figure 8.1 shows snapshots of the grasp execution using the simulator and our robot. This work has been done in collaboration with the Universitat Jaume I (UJI). How this grasp cycle is implemented on the very different UJI platform is detailed in [4]. An example for the execution is shown in Figure 8.1.

8.1.2 Executing Grasps on Familiar Objects

In Section 7.4, we proposed a method for generating grasp hypotheses on familiar objects in which 2D and 3D shape cues of an object hypothesis were combined. A machine learning approach was used to transfer grasping experience from objects encountered previously. 2D grasping points detected in such a way were used to search for full grasp poses given a minimal 3D representation. Since no mesh model was built of the object, the simulator was not used for grasp and motion planning. Instead, we inferred one grasp hypothesis and directly executed it on the robot. In Figure 8.2, video snapshots from the robot grasping three different objects are given along with the segmented input image, inferred grasping point distribution and detected dominant plane [1]. For this demonstration, we rejected grasping points for which the approach vector would result in a collision with the table.

In general, we can observe that the generated grasp hypotheses are reasonable selections from the huge amount of potentially applicable grasps. Failed grasps are due to the fact that there is no closed-loop control implemented either in terms of visual servoing or hand movements.Grasps also fail due to the slippage or collision.

8.2 Closed-Loop Grasp Execution

Different to an open-loop grasp controller, a closed-loop controller does not execute the desired trajectory without taking feedback into account. Instead, the outcome of the execution is monitored constantly and adapted to ensure that the desired goal is reached. Different closed-loop systems have been employed for executing grasping. Visual Servoing techniques are mainly used for controlling the reaching trajectory. Reactive-grasp controllers employ force-torque or haptic sensors to ensure that the grasp is successful.

After outlining the error sources in our robotic system in more detail, we will review existing approaches for closed-control in grasping. This is followed by examples on how we use different visual servoing techniques to grasp known and unknown objects as described in our previous work [10].

8.2.1 Error Sources in Our Robotic System

The grasp cycle outlined in Algorithm 4 relies on the assumption that all the parameters of the system are perfectly known. This includes e.g. internal and external camera parameters as well as the pose of the head, arm and hand in a globally consistent coordinate frame.

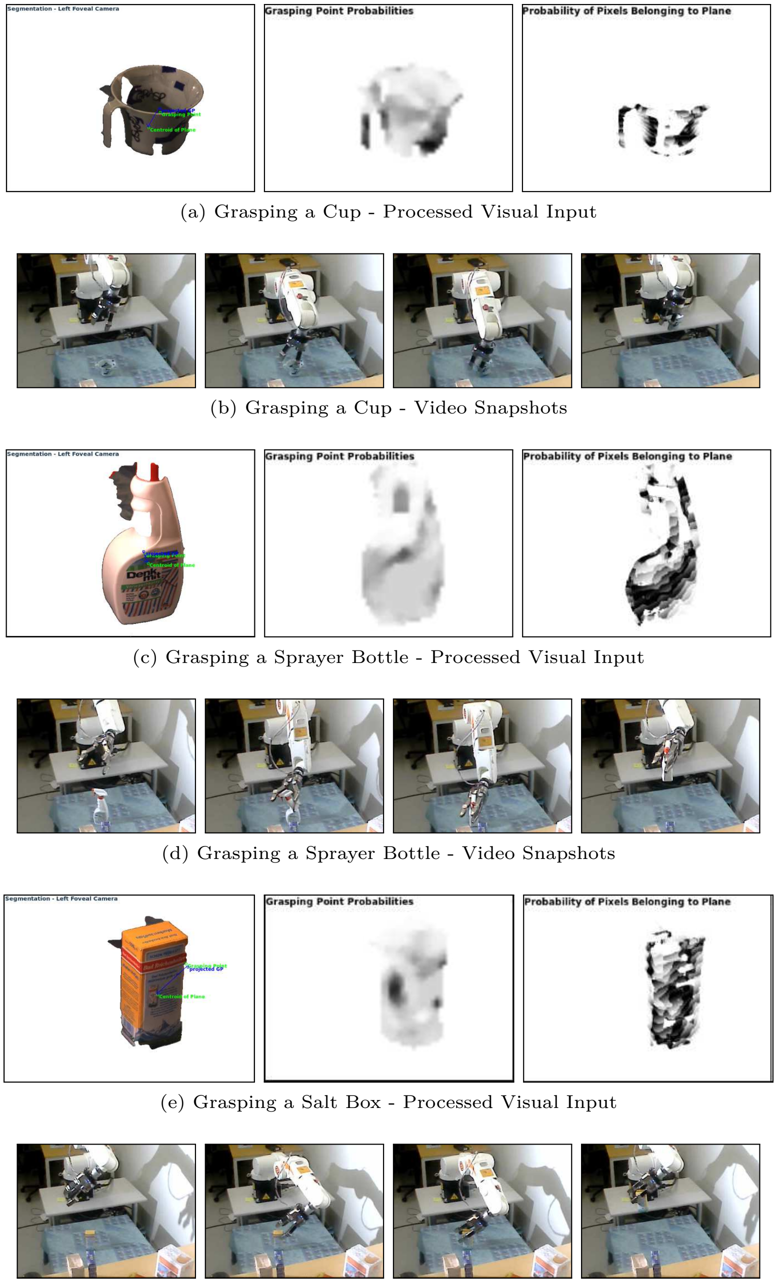
In reality, however, two different types of errors are inherent to the system. One contains the systematic errors that are repeatable and arise from inaccurate calibration or inaccurate kinematic models. The other group comprises random errors originating from noise in the motor encoders or in the camera. These errors propagate and deteriorate the relative alignment between hand and object.

The most reliable component in our system is the Kuka arm that has a repeatability of less than 0.03 mm [126].

Our stereo calibration method uses the arm as world reference frame and is described in more detail in [10]. We achieve an average re-projection error of 0.1 pixels. Since peripheral and foveal cameras of the active head are calibrated simultaneously, we also get the transformation between them.

Additionally to the systematic error in the camera parameters, other effects such as camera noise and specularities lead to random errors in the stereo matching.

[1] A number of videos can be downloaded from `www.csc.kth.se\~bohg`.


Segmentation - Left Foveal Camera
Grasping Point
Centroid of Plane
Grasping Point Probabilities
Probability of Pixels Belonging to Plane
Segmentation - Left Foveal Camera
Grasping Point
Centroid of Plane
Grasping Point Probabilities
Probability of Pixels Belonging to Plane
Segmentation - Left Foveal Camera
Grasping Point
Projected GP
Centroid of Plane
Grasping Point Probabilities
Probability of Pixels Belonging to Plane

(a) Grasping a Cup - Processed Visual Input

(b) Grasping a Cup - Video Snapshots

(c) Grasping a Sprayer Bottle - Processed Visual Input

(d) Grasping a Sprayer Bottle - Video Snapshots

(e) Grasping a Salt Box - Processed Visual Input

(f) Grasping a Salt Box - Video Snapshot

Figure 8.2: Generating grasps for different objects: Left: Grasping Point, Projected Grasping Point and Plane Centroid. Middle: Grasping Point Probabilities. Right: Probabilities of Pixels Belonging to the Dominant Plane.

Since we are using the arm to calibrate the stereo system, the error in the transformation between the arm and cameras is proportional to the error of the stereo calibration.

We are using the robot head to actively explore the environment through gaze shifts and fixation on objects. This involves dynamically changing the epipolar geometry between the left and right camera system. Also the camera position relative to the head origin is changed. The internal parameters remain static. Only the last two joints of the head, the left and right eye yaw, are used for fixation and thereby affect the stereo calibration. The remaining joints are actuated for performing gaze shifts.

In order to accurately detect objects with respect to the camera, the relation between the two camera systems as well as between the cameras and the head origin needs to be determined after each movement. Ideally, these transformations should be obtainable just from the known kinematic chain of the robot and the readings from the encoders. In reality, these readings are erroneous due to noise and inaccuracies in the kinematic model. To reduce the systematic errors, we are extending our stereo calibration method to not just calibrate for one head pose, but for several. This is described in more detail in [10][13].

Regarding random error, the last five joints in the kinematic chain of the active head achieve a repeatability in the range of ±0.025 degrees [17]. The neck pitch and neck roll joints have a lower repeatability in the range of ±0.13 and ±0.075 degrees respectively.

8.2.2 Related Work

The systematic and random errors in the robotic system lead to an erroneous alignment of the end effector with an object. A system that executes a grasp in an open loop fashion without taking any sensory feedback into account is likely to fail. In the following, we will review existing approaches towards calibrating the robot and towards closed-loop control.

8.2.2.1 Calibration Methods

In [222], the authors presented a method for calibrating the active stereo head. The correct depth estimation of the system was demonstrated by letting it grasp an object held in front of its eyes. No dense stereo reconstruction has been shown in this work.

Similarly, Welke et al. [229] present a procedure for calibrating the Karlsruhe robotic head. Our calibration procedure [10] is similar to the one described in those papers, with a few differences. We extend it to the calibration of all joints, thus obtaining the whole kinematic chain. Also, the basic calibration method is modified to use an active pattern instead of a fixed checkerboard.

Pradeep et al. [173] present a system in which a robot moves a checkerboard pattern in front of its cameras to several different poses. Similar to our approach, it can use its arms as kinematic chain sensors and automate the otherwise tedious calibration process. However, the poses of the robot are hand crafted such that the checkerboard is guaranteed to be visible to the sensor that is to be calibrated. In our approach, we use visual servoing to automatically generate a calibration pattern that is uniformly distributed in camera space.

8.2.2.2 Visual Servoing for Grasp Execution

Some grasping systems make use of visual feedback to correct the wrong alignment before contact between the manipulator and the object is established. Examples are proposed by Huebner et al. [102] and Ude et al. [223], who are using a similar robotic platform to ours including an active head. In [102], the Armar III humanoid robot is enabled to grasp and manipulate known objects in a kitchen environment. Similar to our system [13], a number of perceptual modules are at play to fulfill this task. Attention is used for scene search. Objects are recognized and their pose estimated with the approach originally proposed by Azad et al. [19]. Once the object identity is known, a suitable grasp configuration can be selected from an offline constructed database. Visual servoing based on a spherical marker attached to the wrist is applied to bring the robotic hand to the desired grasp position as described in Vahrenkamp et al. [224]. Different from our approach, absolute 3D data is estimated by fixing the 3 DoF for the eyes to a position for which a stereo calibration exists. The remaining degrees of freedom controlling the neck of the head are used to keep the target and current hand position in view. In our approach, we reconstruct the 3D scene by keeping the eyes of the robot in constant fixation on the current object of interest. This ensures that the left and right visual field overlap as much as possible, thereby maximizing e.g. the amount of 3D data that can be reconstructed. However, the calibration process becomes much more complex.

In the work by Ude et al. [223], fixation plays an integral part of the vision system. Their goal is however somewhat different from ours. Given that an object has already been placed in the hand of the robot, it moves it in front of its eyes through closed-loop vision-based control. By doing this, it gathers several views from the currently unknown object for extracting a view-based representation that is suitable for recognizing it later on. Different to our work, no absolute 3D information is extracted for the purpose of object representation. Furthermore, the problem of aligning the robotic hand with the object is circumvented.

8.2.2.3 Reactive Grasping Approaches

The main problem of open-loop grasping is that uncertainties introduced by the perception and motor systems are not taken into account during execution. Several approaches deal with this by either adapting the grasps online, by re-grasping or by taking uncertainties directly into account during learning appropriate grasps.

In the following, we will review existing approaches divided regarding the assumed prior knowledge on the object.

Known Objects When the task of the robot is to grasp a known object, the main source of uncertainty lies in pose estimation. Bekiroglu et al. [25] use a combination of visual pose tracking and tactile features to assess grasp quality. The method is solved through supervised learning based on labeled training data to decide before lifting whether a grasp is going to be successful or not. Hsiao et al. [100] approach the problem of grasping under pose uncertainty through a decision theoretic approach. Based on the current belief state, the system decides between grasping and information-gathering actions using tactile input. Once the object pose is known with sufficient certainty, grasping is executed. Tegin et al. [216] develop a reactive grasp controller in simulation in a framework of human-to-robot mapping of grasps. The approach is shown to be robust against pose errors introduced by visual pose estimation. Romero et al. [186] present a similar system based on visual hand pose estimation instead of using a magnetic tracker. Not only uncertainties in object pose but also inaccuracies in the mapping are taken into account. Although the object is not explicitly modelled, it is assumed that the grasp to be imitated by the robot is performed on the same object in the same pose.

[124, 167] model the whole reaching and grasping movements. Pastor et al. [167] adjust movement primitives online to take pose uncertainties into account. Expected sensor readings during grasping an object are recorded from an ideal demonstration. During autonomous execution, the movement primitive is adapted online in case of divergences between the expected and actual sensor readings. It is shown that this approach leads to an increased grasp success rate for different objects. As an open question remains how the system can select between a potential set of object-specific expected sensors readings when executing a grasping action.

Kroemer et al. [124] apply a reinforcement learning approach in wich a low-level reactive controller is applied to provide the high-level learner with rewards dependent on how much the fingers moved during the execution of the grasp. Primitives modeling the whole reaching movement are also adapted using a sparse visual representation of the environment to avoid collisions. This approach has been demonstrated for different objects in which the learnt policy converges to high rewarding grasps.

Unknown Objects The challenge in the case of grasping unknown objects is different from the previous case. Instead of an unknown relative pose between hand and object, here we have incomplete and noisy information about the object geometry. The following approaches assume that there is a system that provides them with an initial grasp hypothesis. This can be any of those presented in Section 7.1.2. However, these hypotheses are usually inferred on either incomplete object models or on completed but not necessarily accurate object models.

Felip and Morales [78] developed so called grasp primitives as indivisible actions defined by initial hand pre-shape, a sensor based controller and an end condition. The goal of these primitives is to achieve an alignment between object and hand that ensures a stable grasp. This is achieved by a sequence of re-grasping actions. Mainly, force-torque and finger joint readings are used. Hsiao et al. [99] propose a different reactive controller that uses contact information from tactile sensors. An adaption of the hand position dependent on the readings follows until contact with the palm is detected and a compliant grasp is executed. Maldonado et al. [142] propose a similar approach using contact sensing in the fingertips during the approach of an object. Sensed obstacles are included in the representation of the environment and subsequently used to adapt the grasp hypothesis. All these approaches adapt the initial grasp hypothesis based on force-torque or tactile sensor readings. One of their disadvantages is that during contact the previous model of the scene might change. Previously inferred grasp hypothesis can then be inappropriate or even impossible. A new model of the world has to be acquired which is an expensive operation.

A different reactive grasping approach is taken by Hsiao et al. [98] in which optical proximity sensors are used. Contact with the object is avoided while enough information can be sensed to adapt the initial grasp strategy to something that has a much higher likelihood to succeed.

8.2.3 Visual Servoing for Grasping Known Objects

In this section, we will demonstrate a simple position based visual servoing approach to perform grasping of known objects from a table top. We assume that the object to be grasped has been segmented and reconstructed using the vision system described in Section 2.2.1. Furthermore, we assume that the object has been recognized and its pose estimated using the approach described in Section 7.2.2.1. Due to the usage of a fixating system, we also know that the object is in the center of both, the left and the right foveal camera.

From this information, we can infer all the parameters for a grasp. As approach vector we choose the negative gravity vector, i.e. in this scenario we will only perform simple top grasps. To determine the wrist orientation of the hand, we use the information from pose estimation. In case of a roughly rectangular shape, the minor dimension is aligned with the vector between thumb and finger of the Schunk hand. For a circular shape, the wrist is left oriented in its default pose. As a fixed grasping point, we choose the intersection between the principal axes of the left and right foveal camera in the arm base coordinate frame which we know to be lying on the object.

8.2.3.1 Position-Based Visual Servoing for Alignment

In this scenario, we will use position-based visual servoing to align the hand with the object prior to the actual grasp. The goal is to align the end effector in position $\mathbf{x}_m^{\mathsf{A}}$

with a point $\hat{\mathbf{x}}_g^{\mathsf{A}}$ above the grasping point, both defined in the arm base coordinate frame A. Please note that the goal position is labeled with a hat to indicate that it is only an estimate of the true position. This is due to remaining errors in the calibration of the cameras and the kinematic chain of the active head.

The position of the end effector is known with an accuracy of 0.3mm. However, we want to minimise the distance between current end effector position and goal position with respect to the camera frame. Therefore, we are using a fiducial marker, an LED rigidly attached to the Schunk hand, to determine the position of the end effector relative to the camera coordinate frame. The LED can be seen in Figure 8.3.

As defined in [104], such a positioning task can be represented by a function $e : \mathcal{T} \to \mathbb{R}^N$ that maps a pose in the robot's task space $\mathcal{T}$ to an N-dimensional error vector. This function is referred to as *kinematic error function*. In our simple scenario $\mathcal{T} = \mathbb{R}^3$ and therefore $N = 3$ since we will only change the position of the hand but keep the orientation fixed. In this way, we ensure the visibility of the LED. We consider the positioning task as fulfilled when $e(\mathbf{x}_m^{\mathsf{A}}) = 0$.

8.2.3.2 Computing the Goal Point

Let us define $\hat{\mathbf{x}}_{(g,l)}$ and $\hat{\mathbf{x}}_{(g,r)}$ as the projection of the goal point $\hat{\mathbf{x}}_g^{\mathsf{A}}$ on the left and right image. The parameters of the calibrated stereo camera and robot kinematics are available through a calibration procedure described in [10]. We refer to the complete transformation from image space I to arm space A as $\hat{\mathbf{T}}_{\mathsf{I}}^{\mathsf{A}}$. Note that we label this transform with a hat to indicate that it is only an estimate of the true transform. Using this, the center of both the left and right image can be transformed into the arm coordinate system yielding

$$\hat{\mathbf{x}}_c^{\mathsf{A}} = \hat{\mathbf{T}}_I^{\mathsf{A}}(\hat{\mathbf{x}}_{(c,l)}, \hat{\mathbf{x}}_{(c,r)}). \tag{8.1}$$

Since we are considering grasping of known objects, we shift this point according to the height of the object and the known length of the fingers. Back-projecting the result gives us the desired goal positions $\hat{\mathbf{x}}_{(g,l)}$ and $\hat{\mathbf{x}}_{(g,r)}$ in the stereo image:

$$(\hat{\mathbf{x}}_{(g,l)}, \hat{\mathbf{x}}_{(g,r)}) = \hat{\mathbf{T}}_{\mathsf{A}}^{\mathsf{I}}(\hat{\mathbf{x}}_c^{\mathsf{A}} + (0,0,h)^T). \tag{8.2}$$

8.2.3.3 Control Law

We want to compute the desired translational velocity $\mathbf{u}$ to move the end effector such that the LED at position $\hat{\mathbf{x}}_m^{\mathsf{A}}$ is aligned with $\hat{\mathbf{x}}_g^{\mathsf{A}}$. This means in turn that the projection of these two points have to be aligned in the images.

The position of the LED $\hat{\mathbf{x}}_{(m,l)}$ and $\hat{\mathbf{x}}_{(m,r)}$ projected to the left and right image is detected with subpixel precision. Given this, we can compute $\hat{\mathbf{x}}_m^{\mathsf{A}}$ using the same transformation $\hat{\mathbf{T}}_{\mathsf{l}}^{\mathsf{A}}$. Then the translational velocity is computed as

$$\begin{aligned}
\mathbf{u} &= -k\ e_{pp}(\hat{\mathbf{x}}_m^{\mathsf{A}}; \hat{\mathbf{T}}_{\mathsf{l}}^{\mathsf{A}}(\hat{\mathbf{x}}_{(g,l)}, \hat{\mathbf{x}}_{(g,r)}), \hat{\mathbf{T}}_{\mathsf{l}}^{\mathsf{A}}(\hat{\mathbf{x}}_{(m,l)}, \hat{\mathbf{x}}_{(m,r)})) \quad &(8.3)\\
&= -k\ e_{pp}(\hat{\mathbf{x}}_m^{\mathsf{A}}; \hat{\mathbf{x}}_m^{\mathsf{A}}, \hat{\mathbf{x}}_g^{\mathsf{A}}) \quad &(8.4)\\
&= -k\ (\hat{\mathbf{x}}_m^{\mathsf{A}} - \hat{\mathbf{x}}_g^{\mathsf{A}}) \quad &(8.5)
\end{aligned}$$

where the argument left of the semicolon is the parameter to be controlled (the end effector position), and the arguments right after the semicolon parameterize the positioning task. k is the gain parameter.

Once the robot hand is hovering over the object, we align the wrist correctly and perform the actual grasp, lifting and transporting the object in an open-loop manner. Example snapshots from a grasping action executed on the toy tiger is shown in Figure 8.3.

8.2.4 Virtual Visual Servoing for Grasping Unknown Objects

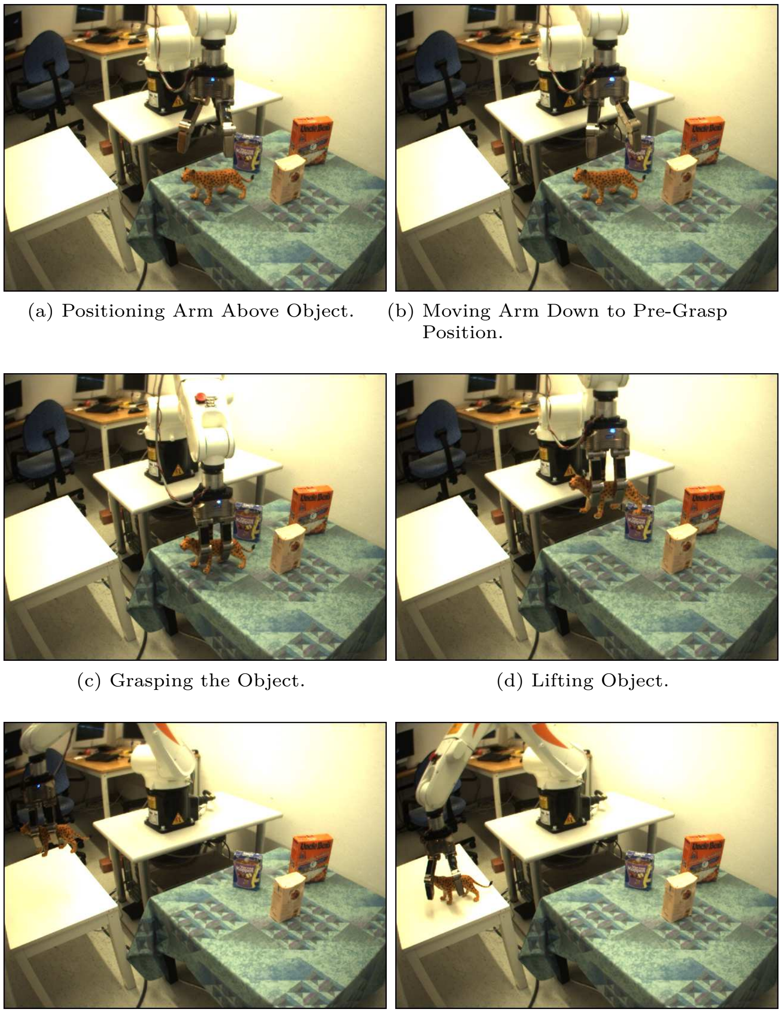
For the accurate control of the robotic manipulator using visual servoing, it is necessary to know its position and orientation (*pose*) with respect to the camera. In the system exemplified in the previous section, the end effector of the robots is tracked using an LED. These kind of fiducial markers are common solutions to the tracking problem. In Section 8.2.2.2, we mentioned other approaches using colored spheres or Augmented Reality tags. A disadvantage of this marker-based approach is that the mobility of the robot arm is constrained to keep the marker always in view. Furthermore, the position of the marker with respect to the end effector has to be known exactly. For these reasons, we propose in [10] to track the whole manipulator instead of only a marker. Thereby, we alleviate the problem of constrained arm movement. Additionally, collisions with the object or other obstacles can be avoided in a more precise way.

8.2.4.1 Image-Based Visual Servoing for Aligning the Virtual with the Real Arm

Our approach to estimation and track the robot is based on virtual visual servoing (VVS) where a rendered model of a virtual robot is aligned with the current image of their real counterparts. Figure 8.4 shows a comparison of the robot localisation in the image using only calibration with the case when this localisation has been corrected using VVS. In [10], this is achieved by extracting features from the rendered image and compare them with the features extracted from the current camera image. Based on the extracted features, we define an error vector

$$\boldsymbol{s}(t) = [d_1, d_2, \ldots, d_N]^T \tag{8.6}$$

between the desired values for the features and the current ones. Let $\dot{\boldsymbol{s}}$ be the rate of change of these distances given a change in pose of the end effector. This change



(a) Positioning Arm Above Object.

(b) Moving Arm Down to Pre-Grasp Position.

(c) Grasping the Object.

(d) Lifting Object.

(e) Moving Object to Goal Position.

(f) Release Object at Goal Position.

Figure 8.3: Example Grasp for the Recognized Tiger.

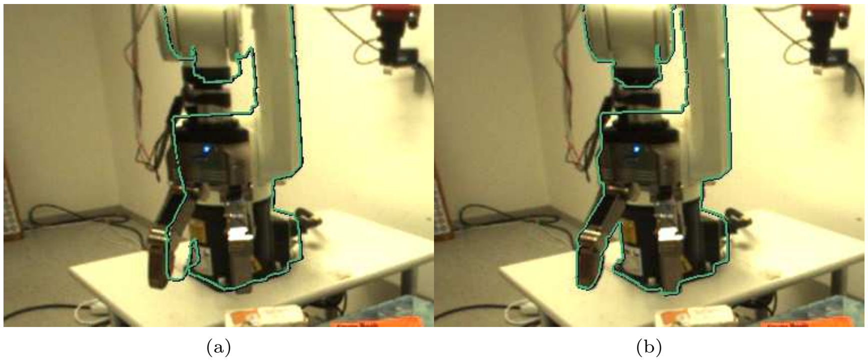

(a) (b)

Figure 8.4: Comparison of robot localization with (right) and without (left) the Virtual Visual Servoing correction.

can be described by a translational velocity $\mathbf{t} = [t_x, t_y, t_z]^T$ and a rotational velocity $\omega = [\omega_x, \omega_y, \omega_z]^T$. Together, they form a velocity screw:

$$\dot{\boldsymbol{r}}(t) = \left[t_x, t_y, t_z, \omega_x, \omega_y, \omega_z\right]^T \tag{8.7}$$

We can then define the image Jacobian as $\mathbf{J}$ so that:

$$\dot{\boldsymbol{s}} = \boldsymbol{J}\dot{\boldsymbol{r}} \tag{8.8}$$

where

$$\boldsymbol{J} = \left[\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{r}}\right] = \begin{bmatrix} \frac{\partial d_1}{\partial t_x} & \frac{\partial d_1}{\partial t_y} & \frac{\partial d_1}{\partial t_z} & \frac{\partial d_1}{\partial \omega_x} & \frac{\partial d_1}{\partial \omega_y} & \frac{\partial d_1}{\partial \omega_z} \\ \frac{\partial d_2}{\partial t_x} & \frac{\partial d_2}{\partial t_y} & \frac{\partial d_2}{\partial t_z} & \frac{\partial d_2}{\partial \omega_x} & \frac{\partial d_2}{\partial \omega_y} & \frac{\partial d_2}{\partial \omega_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial d_N}{\partial t_x} & \frac{\partial d_N}{\partial t_y} & \frac{\partial d_N}{\partial t_z} & \frac{\partial d_N}{\partial \omega_x} & \frac{\partial d_N}{\partial \omega_y} & \frac{\partial d_N}{\partial \omega_z} \end{bmatrix} \tag{8.9}$$

which relates the motion of the (virtual) manipulator to the variation in the features.

Our goal however is to move the robot about $\dot{\boldsymbol{r}}$ such that $\boldsymbol{s}$ becomes equal to $\mathbf{0}$. Therefore, we need to compute $\dot{\boldsymbol{r}}$ given $\dot{\boldsymbol{s}}$. In our case, the Jacobian is non-square and therefore not invertible. Therefore, we compute

$$\dot{\boldsymbol{r}} = \boldsymbol{J}^{+}\dot{\boldsymbol{s}} \tag{8.10}$$

where $\boldsymbol{J}^{+}$ is the pseudoinverse of $\boldsymbol{J}$.

Let us denote the pose of the virtual robot as $\mathbf{T}^{\mathsf{A}}$, then the kinematic error function e_{im} can be defined as

$$e_{im}(\mathbf{T}^{\mathsf{A}}; \boldsymbol{s}, \mathbf{0}) = \mathbf{0} - \boldsymbol{s} \tag{8.11}$$

which leads to the simple proportional control law

$$\mathbf{u} = -\mathbf{K}\boldsymbol{J}^{+}\boldsymbol{s} \tag{8.12}$$

where $\mathbf{K}$ is the gain parameter.

For details on how the Jacobian and the features are computed, we refer to [10].

8.2.4.2 Incorporating Pose Corrections for Grasping Unknown Objects

Given the corrected pose of the robot arm relative to the camera, we have a corrected transformation $\mathbf{T}_{\mathsf{l}}^{\mathsf{A}}$ converting pixel coordinates in the left and right image to coordinates in the arm base frame. Therefore, given goal positions $\mathbf{x}_{(g,l)}$ and $\mathbf{x}_{(g,r)}$ of the robot's end effector in the left and right image, we can convert them to goal positions $\mathbf{x}_{g}^{\mathbf{A}}$. This position is computed based on an already corrected pose estimate. We therefore send the end effector to this position in open loop to correctly align the hand with the object.

Similarly to the previous section, the goal position of the robot arm is above the object to be grasped. This point can be inferred using any of the methods proposed in Section 7. However, in [10] we demonstrated grasping without assuming any knowledge about the object. Given a segmented point cloud, we project it to the table and use its centre of mass as an approach vector for a top grasp. The wrist is aligned with the minor eigenvector of the projection.

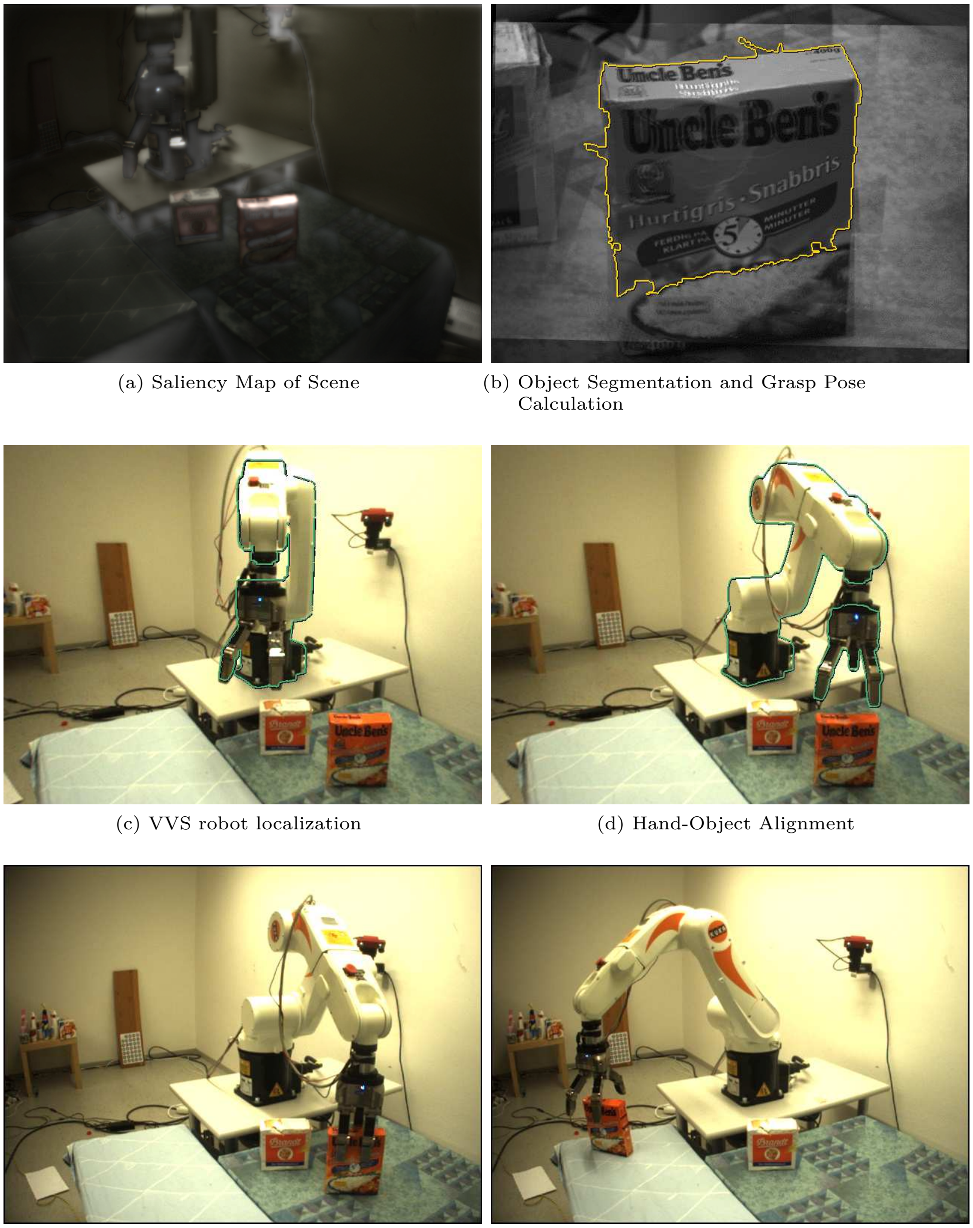
Example snapshots of one execution can be found in Figure 8.5.

8.3 Discussion

In this section, we have first reviewed different closed-loop control schemes for grasp execution. We distinguished between vision and tactile-based techniques. Visual Servoing is usually applied to bring the end effector into an appropriate pre-grasp position. Reactive grasping approaches usually start at this point and execute grasping using force-torque or haptic sensors to ensure grasp success.

We then presented open-loop grasping of unknown or familiar objects and demonstrated their limitations in coping with erroneous calibration or imprecise kinematic models. Two applications for visual servoing were shown. The first one was a simplified position-based visual servoing scheme based on tracking a fiducial marker. Since this approach limits the mobility of the end effector, we also presented virtual visual servoing based grasping of unknown objects. This method is advantageous when the goal is to precisely align the fingertips of the robotic hand with the object.

We have demonstrated this approach for simple top grasps only. As a next step, more complex grasps in potentially cluttered environments need to be considered. Furthermore, a reactive grasping approach would be advantageous to cope with the remaining errors in the relative pose estimate between hand and object.



(a) Saliency Map of Scene

(b) Object Segmentation and Grasp Pose Calculation

(c) VVS robot localization

(d) Hand-Object Alignment

(e) Close hand.

(f) Release Object at Goal Position.

Figure 8.5: Example Grasp Using Virtual Visual Servoing

9

Conclusions

A long-term goal in robotics research is to build an intelligent machine that can be our companion in everyday life. It is thought to help us with tasks that are for example difficult, dangerous or bothersome. However for many of these tasks that we perform effortlessly, current robotic systems still lack the robustness and flexibility to carry them out equally well.

In this thesis, we considered a household scenario in which one of the key capabilities of a robot is grasping and manipulation of objects. Specifically, we focussed on the problem of scene understanding for object grasping. We followed the approach of introducing prediction into the perceive-act loop. We showed how this helps to efficiently explore a scene but also how it can be exploited to improve manipulation.

We studied different scene representations as a container for knowledge about the geometric structure of the environment. They are of varying complexity and utility, reaching from a classical occupancy grid map, over a height map to a 3D point cloud. While the occupancy grid map allows to estimate free and occupied spaces in the scene, a height map provides additional geometric information essential to manipulation. A point cloud can retain scene geometry most accurately while not providing the efficiency of the other two approaches.

All these representations where studied considering fusion of multi-modal sensory data. Visual sensing through monocular and stereo cameras is the most important modality used in this thesis. Haptic sensors provided information about areas in the scene that could not be visually observed. Furthermore, we used object recognition and pose estimation as a virtual sensor. Finally, we showed how information from a human operator enhances object segmentation.

In this thesis, prior knowledge about *objectness* or common scene structure was exploited for predicting unexplored, unobserved or uncertain parts of the scene. In Chapter 4, we used Gaussian Processes to predict scene geometry based on parts that have already been observed. The prior knowledge is constituted of a kernel function that reflects common object shape. In Chapter 5, we made use of the observation that objects are relatively homogenous in their attribute space. This prior helped the robot to autonomously evaluate segmentation quality. In Chapter 6, we proposed an approach for estimating the backside of unknown objects. It is based on the prior knowledge that man-made objects are commonly symmetric.

Additionally to estimating the unobserved parts of the scene, we showed how we can quantify the uncertainty about the prediction. This becomes useful for choosing the most beneficial exploratory actions. In Chapter 4, these actions were performed with the robot hand equipped with haptic sensor matrices. In Chapter 5, a human was questioned to confirm whether an uncertain object hypothesis has to be corrected or not.

Apart from selecting exploratory actions to refine a scene model, we have also shown how predictions can improve grasp synthesis. Dependent on whether the objects are known, familiar or unknown, different methodologies for grasp inference apply. In Chapter 7, we reviewed related approaches according to this taxonomy.For each case, we proposed a novel method extending the state of the art. For known objects, we showed that using a completed point cloud is rendering pose estimation more robust. For unknown objects, we showed how completed point cloud help to infer more robust grasps. And finally for familiar objects, we showed that given an accurate object segmentation, good grasp points can be inferred based on the object's global shape. We demonstrated these approaches in real-world scenes of table-top scenarios thereby proving their feasibility.

In Section 1.1, we showed an example scenario in which a robot is asked to clean a very messy office desk. We listed several scientific challenges that have to be solved before a real robot will be able to fulfill such a task satisfactorily. In this thesis, we made contributions to the subtasks of scene understanding and robotic grasping. Given different scene representations, sensors and prior knowledge, we proposed several prediction mechanisms that provide the robot with information beyond what is has directly observed.

However, many scientific challenges still remain before a robot can autonomously clean a desk as envisioned in the introduction of this thesis. In the following, we will outline the parts that we believe have to be solved next.

Independence of Table-Top Detection Almost all of the proposed methods for scene understanding rely on knowing the dominant plane on which the objects where arranged. This allowed us to use efficient scene representations and a reduction of the search space for object shape prediction. However, extracting the table-top might not be trivial due to clutter. Hence, the proposed representations have to be made independent of this knowledge while at the same time remain efficiently computable. If the dominant plane can however be detected, the resulting possibility to reduce the search space should be made use of.

Grasp-Oriented Active Perception Considering the tremendous amount of memory that would be required to store an accurate and detailed 3D point cloud of a large scale environment, we need to find a more efficient way to represent it. This representation has to retain the amount of information that makes it possible for the robot to roughly plan its course of actions. We believe, that the fine detail information necessary for example for collision avoidance, opening doors or grasp planning, should be obtained rapidly and only on demand in the context of specific behaviors and tasks.

Learning Priors In all chapters, we have used prior assumptions about scene structure and objectness. This reaches from blob-like kernel functions over homogeneity in specific object attributes to symmetric shape of objects. We showed that these priors are useful in predicting the structure of unknown space. While we believe that it is adequate to make assumptions about some low-level priors, the questions remains how additional priors can be learnt autonomously and adapted iteratively over a much longer time span (potentially life-long).

Transferring Affordances As has been shown in Chapter 7, there is quite a large body of work on generating grasp hypotheses given an object hypothesis. One kind of these methods aims at transferring grasp experience from previously encountered objects to new ones. Many of these methods rely on relatively low-level features like distribution of edges, color or texture. Only very few make use of higher-level information like object identity or category. However, we believe that only given such information allows to link the generated grasp hypotheses to tasks.

Adaptation of Grasp Hypotheses The prediction mechanism that we proposed in this thesis have mainly been used to trigger exploratory actions. This in turn led to an improved scene model with correct object hypotheses. However only the result of the method proposed in Chapter 6 was used to improve grasp hypothesis selection. The robustness of grasping could be improved if the remaining uncertainty in the scene model is explicitly included in the grasp synthesis process.

Multi-Modal Closed-Loop Control In Chapter 8, we showed how Visual Servoing and Virtual Visual Servoing can help to improve grasp execution in a closed-loop fashion. However, these techniques were so far only used for aligning the robot hand with the object. The actual grasping, i.e. closing the fingers around the object and lifting it, were performed open loop only. To enable the robot to grasp objects robustly and adapt its behavior when facing unexpected sensor reading, we need to adopt a reactive grasping approach. This should not only involve haptic sensor data from tactile or force-torque sensors. It should also include visual tracking of the object that is manipulated. Furthermore, the uncertainty about the geometry of the object or the scene should influence execution of the grasp as well. It might for example influence grasp parameters such as pre-shape or velocity of the end effector.

A

Covariance Functions

Throughout this thesis, we use different learning techniques to enable a robot to extrapolate from prior experience. For example in Chapter 4, the robot has to predict the geometry of a scene from partial visual observations. These observations can be formally seen as a set of inputs $\mathcal{X}$ for which we already know the correct target values $\mathcal{Y}$. For new inputs x' that we do not yet know the appropriate target value of, we use the concept of *similarity*. New inputs that are close to already known ones should have similar target values.

In the framework of Gaussian Processes, the similarity is defined through covariance functions. In the following, we present the covariance functions used in Chapter 4 in a bit more detail. Figure A.1 visualises these different functions. For a more detailed review, we refer to [181].

Squared Exponential Covariance Function Most of the covariance functions presented here are *stationary*. They are functions of the difference $r = x - x'$ between input points. These functions are therefore invariant to translations. The squared exponential covariance function (SE) is defined as

$$k_{SE}(r) = \exp(-\frac{r^2}{2l^2}). \tag{A.1}$$

Dependent on the hyperparameter l, the length scale, this function places a more or less strong smoothing on the input signal. It is visualized for different length scales in Figure A.1a.

Mátern Class of Covariance Functions The strong smoothing of the SE kernel has been argued to be unrealistic for most physical processes [181]. The Mátern class is more flexible and allows varying degrees of smoothness. It is dependent on the two parameters ν and l. In this thesis, we only looked at the case where ν is a half-integer and the covariance function can be significantly simplified. With $\nu = 1/2$, the covariance function is equal to the exponential function

$$k_{Mat1}(r) = \exp(-r/l). \tag{A.2}$$

We also tested the cases where $\nu = 3/2$ and $\nu = 5/2$ that are claimed by Bishop [31] to be the most interesting for machine learning applications:

$$k_{Mat3}(r) = (1 + \frac{\sqrt{3}r}{l}) \exp(-\frac{\sqrt{3}r}{l}) \tag{A.3}$$

$$k_{Mat5}(r) = (1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}) \exp(-\frac{\sqrt{5}r}{l}). \tag{A.4}$$

These cases of the Mátern class are visualized in Figures A.1b, A.1c and A.1d. With $\nu \to \infty$, the Mátern covariance function becomes the squared exponential function.

Rational Quadratic Covariance Function This covariance function is dependent on the two hyperparameter l and α. It is defined as

$$k_{RQ}(r) = (1 + \frac{r^2}{2\alpha l^2})^{-\alpha}. \tag{A.5}$$

Fixing the length scale $l = 1$, the RQ kernel is visualized for different α in Figure A.1e. It can be seen as a scale mixture of SE kernels with different characteristic length scales. With $\alpha \to \infty$ the RQ function becomes an SE function with length scale l.

SE with Automatic Relevance Detection Although the previous RQ kernel is flexible through a mixture of SE kernels of different length scales, it still values each input value in the set $\mathcal{X}$ as equally important. If the dimension of the input space $D > 1$, then the SE covariance function can also be written as

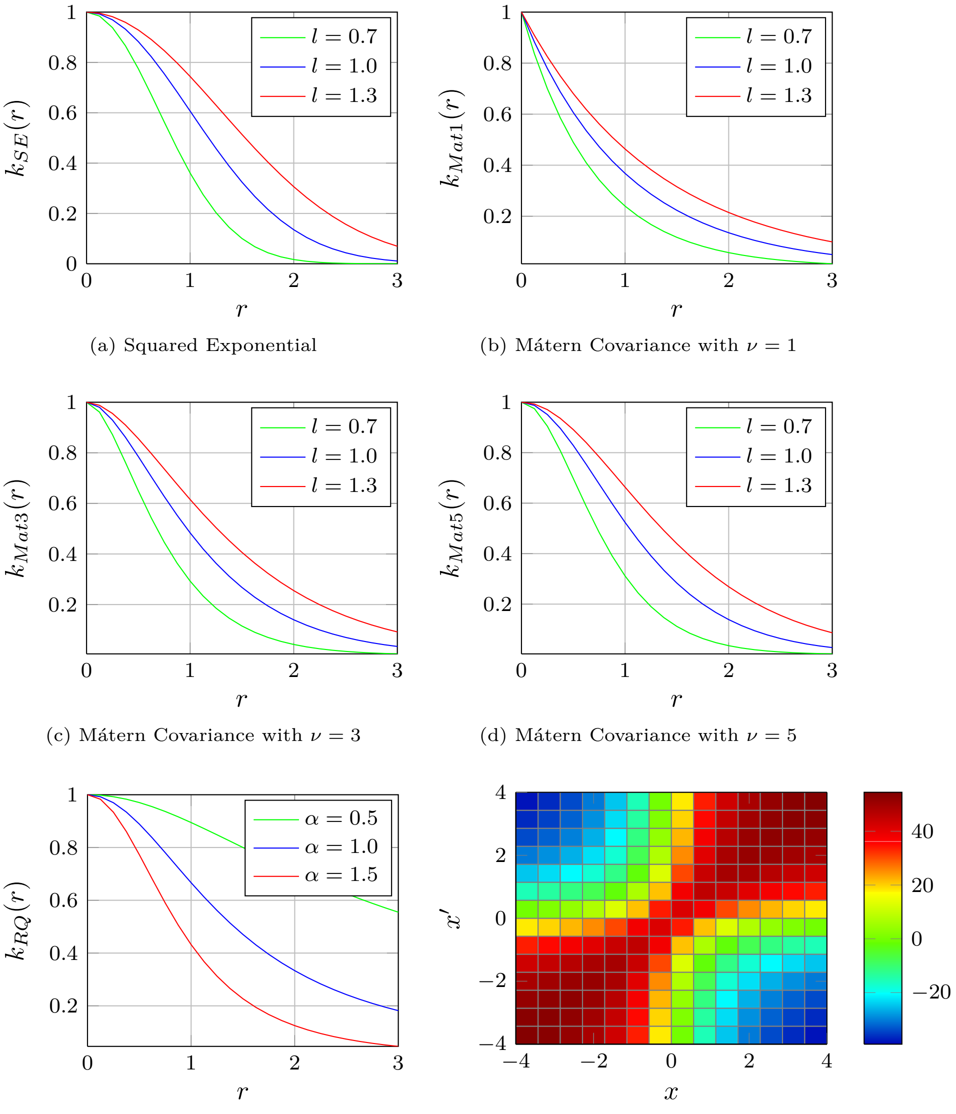
$$k_{SEARD}(\mathbf{r}) = \sigma_f^2 \exp(-\frac{1}{2}\mathbf{r}^T M \mathbf{r}) \tag{A.6}$$

where $M = diag(\mathbf{l})^{-2}$. The vector $\mathbf{l}$ is of dimension D. Each component equals the characteristic length scale of each dimension of the input space. Therefore, if one of the dimensions has a very large length scale, it will become irrelevant for inference.

Neural Network Covariance Function This covariance function is *non-stationary*, i.e., it is not invariant to translations. It is derived from a neural network architecture with N units that implement a specific hidden transfer function $h(\mathbf{x}; \mathbf{u})$ with $\mathbf{u}$ as input-to-hidden weights. It takes in an input $\mathbf{x}$ and combines the outputs of the units linearly with a bias to evaluate the function $f(x)$. Setting the transfer function $h(\mathbf{x}; \mathbf{u}) = \text{erf}(u_0 + \Sigma_{j=1}^{D} u_j x_j)$, the covariance function is defined as follows

$$k_{NN}(x, x') = \frac{2}{\pi} \arcsin(\frac{2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1 + 2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}})(1 + 2\tilde{\mathbf{x}}'^T \Sigma \tilde{\mathbf{x}}')}}) \tag{A.7}$$

where $\tilde{\mathbf{x}} = (1, x_1, \cdots, x_D)$ is the augmented input vector. It is visualized in Figure A.1f for $\Sigma = \text{diag}(\sigma_0^2, \sigma^2)$ with $\sigma_0^2 = \sigma^2 = 10$.



(a) Squared Exponential

(b) Mátern Covariance with $\nu = 1$

(c) Mátern Covariance with $\nu = 3$

(d) Mátern Covariance with $\nu = 5$

(e) Rational Quadratic

(f) Neural Network

Figure A.1: Visualisation of Covariance Functions

Bibliography

[15] K. S. Aarun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 9:698–700, 1987.

[16] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 169–175, 2006.

[17] T. Asfour, K. Welke, P. Azad, A. Ude, and R. Dillmann. The Karlsruhe Humanoid Head. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 447–453, Daejeon, Korea, December 2008.

[18] A. Aydemir, M. Göbelbecker, A. Pronobis, K. Sjöö, and P. Jensfelt. Plan-based object search and exploration using semantic spatial knowledge in the real world. In *European Conf. on Mobile Robots (ECMR)*, Örebro, Sweden, Sept. 2011.

[19] P. Azad, T. Asfour, and R. Dillmann. Stereo-based 6d object localization for grasping with humanoid robot systems. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 919–924, 2007.

[20] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 78(8):966–1005, 1988.

[21] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2294–2301. IEEE, 2010.

[22] D. H. Ballard. Animate vision. *Artif. Intell.*, 48:57–86, February 1991.

[23] L. W. Barsalou. Grounded cognition. *Annual Review of Psychology*, 59(1): 617–645, 2008.

[24] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth. Robotic roommates making pancakes. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, Bled, Slovenia, October, 26–28 2011.

[25] Y. Bekiroglu, R. Detry, and D. Kragic. Joint observation of object pose and tactile imprints for online grasp stability assessment. In *Manipulation Under Uncertainty (Workshop at IEEE ICRA 2011)*, 2011.

[26] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522, 2002.

[27] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner. Grasp planning in complex scenes. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 42 – 48, December 2007.

[28] N. Bergström, C. H. Ek, M. Björkman, and D. Kragic. Scene understanding through autonomous interactive perception. In *Proceedings of the 8th international conference on Computer vision systems*, ICVS'11, pages 153–162, Berlin, Heidelberg, 2011. Springer-Verlag.

[29] N. Bergström, C. H. Ek, M. Björkman, and D. Kragic. Generating object hypotheses in natural scenes through human-robot interaction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 827–833, San Francisco,USA, September 2011.

[30] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 14:239–256, 1992.

[31] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007.

[32] M. Björkman and J.-O. Eklundh. Foveated Figure-Ground Segmentation and Its Role in Recognition. *British Machine Vision Conf.*, 2005.

[33] M. Björkman and J.-O. Eklundh. Vision in the real world: Finding, attending and recognizing objects. *Int. Jour. of Imaging Systems and Technology*, 16 (5):189–209, 2006.

[34] M. Björkman and D. Kragic. Combination of foveal and peripheral vision for object recognition and pose estimation. *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 5:5135 – 5140, April 2004.

[35] M. Björkman and D. Kragic. Active 3d scene segmentation and detection of unknown objects. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3114–3120, 2010.

[36] N. Blodow, R. B. Rusu, Z. C. Marton, and M. Beetz. Partial View Modeling and Validation in 3D Laser Scans for Grasping. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 459 – 464, Paris, France, December 7-10 2009.

[37] G. M. Bone, A. Lambert, and M. Edwards. Automated Modelling and Robotic Grasping of Unknown Three-Dimensional Objects. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 292–298, Pasadena, CA, USA, May 2008.

[38] A. Borghi. *Grounding Cognition: The Role of Perception and Action in Memory, Language, and Thinking*, chapter Object Concepts and Action. Cambridge University Press, 2005.

[39] C. Borst, M. Fischer, and G. Hirzinger. Grasping the Dice by Dicing the Grasp. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3692–3697, 2003.

[40] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, pages 1124–1137, 2004.

[41] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 648–655, 1998.

[42] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, pages 1222–1239, 2001.

[43] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[44] T. P. Breckon and R. B. Fisher. Amodal volume completion: 3d visual completion. *Computer Vision and Image Understanding*, 99(3):499–526, 2005.

[45] P. Brook, M. Ciocarlie, and K. Hsiao. Collaborative grasp planning with multiple object representations. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, ICRA'11, pages 2851 – 2858, 2011.

[46] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1&2):3–15, June 1990.

[47] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47: 139–159, 1991.

[48] U. Castiello and M. Jeannerod. Measuring Time to Awareness. *Neuroreport*, 2(12):797–800, 1991.

[49] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10:273–304, 1995.

[50] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[51] H. I. Christensen, A. Sloman, G.-J. M. Kruijff, and J. Wyatt, editors. *Cognitive Systems.* Springer Verlag, 2009.

[52] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan. Towards reliable grasping and manipulation in household environments. In *Int. Symposium on Experimental Robotics (ISER)*, New Delhi, India, December 2010.

[53] M. Ciorcarlie, C. Goldfeder, and P. Allen. Dexterous Grasping via Eigengrasps: A Low-Dimensional Approach to a High-Complexity Problem. *Robotics: Science and Systems Manipulation Workshop*, 2007.

[54] D. Cole, A. Harrison, and P. Newman. Using naturally salient regions for SLAM with 3D laser data. In *Int. Conf. on Robotics and Automation, SLAM Workshop*, 2005.

[55] A. Collet Romea, D. Berenson, S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 48 – 55, May 2009.

[56] M. Craven and J. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems (NIPS-8)*, pages 24–30. MIT Press, 1995.

[57] S. H. Creem and D. R. Proffitt. Grasping Objects by Their Handles: A Necessary Interaction between Cognition and Action. *Jour. of Experimental Psychology: Human Perception and Performance*, 27(1):218–228, 2001.

[58] R. H. Cuijpers, J. B. J. Smeets, and E. Brenner. On the Relation Between Object Shape and Grasping Kinematics. *Jour. of Neurophysiology*, 91:2598–2606, 2004.

[59] N. Curtis and J. Xiao. Efficient and effective grasping of novel objects through learning and adapting a knowledge base. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2252 –2257, sept. 2008.

[60] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV Int. Workshop on Statistical Learning in Computer Vision*, 2004.

[61] E. Decker. *Paik, Video.* DuMont, Köln, 1988.

[62] R. Detry, E. Başeski, N. Krüger, M. Popović, Y. Touati, O. Kroemer, J. Peters, and J. Piater. Learning object-specific grasp affordance densities. In *IEEE Int. Conf. on Development and Learning*, pages 1–7, 2009.

[63] R. Detry, N. Pugeault, and J. Piater. A probabilistic framework for 3D visual object representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31(10):1790–1803, 2009.

[64] R. Detry, D. Kraft, A. G. Buch, N. KrÃ¼ger, and J. Piater. Refining grasp affordance models by experience. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2287–2293, 2010.

[65] R. Diankov. *Automated Construction of Robotic Manipulation Programs.* PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.

[66] R. Diankov and J. Kuffner. Openrave: A planning architecture for autonomous robotics. Technical Report CMU-RI-TR-08-34, Robotics Institute, Pittsburgh, PA, July 2008.

[67] C. Dunes, E. Marchand, C. Collowet, and C. Leroux. Active Rough Shape Estimation of Unknown Objects. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3622–3627, 2008.

[68] R. Eidenberger and J. Scharinger. Active perception and scene modeling by planning with probabilistic 6d object poses. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1036–1043, Taipei, Taiwan, October 2010.

[69] C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI 2007)*, volume LNCS 4892, pages 132–143, Brno, Czech Republic, Jun. 2007. Springer-Verlag.

[70] S. Ekvall and D. Kragic. Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4715–4720, 2007.

[71] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating active mobile robot object recognition and slam in natural environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5792–5797, 2006.

[72] S. El-Khoury and A. Sahbani. Handling Objects By Their Handles. In *IROS-2008 Workshop on Grasp and Task Learning by Imitation*, 2008.

[73] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22:46–57, 1989.

[74] J. A. Etzel, V. Gazzola, and C. Keysers. Testing simulation theory with cross-modal multivariate classification of fmri data. *PLoS ONE*, 3(11):1–6, 11 2008.

[75] Q. Fan, A. Efrat, V. Koltun, S. Krishnan, and S. Venkatasubramanian. S.: Hardware-assisted natural neighbor interpolation. In *Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2005.

[76] D. R. Faria, R. Martins, J. Lobo, and J. Dias. Extracting data from human manipulation of objects towards improving autonomous robotic grasping. *Robotics and Autonomous Systems*, Aug. 2011.

[77] J. Feldman, K. Pingle, T. Binford, G. Falk, A. Kay, R. Paul, R. Sproull, and J. Tenenbaum. The use of vision and manipulation to solve the "instant insanity" puzzle. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 359–364, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.

[78] J. Felip and A. Morales. Robust sensor-based grasp primitive for a three-finger robot hand. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1811–1816, Piscataway, NJ, USA, 2009. IEEE Press.

[79] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(1):36–51, 2008.

[80] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381—395, June 1981.

[81] S. Foix, G. Alenyà, J. Andrade-Cetto, and C. Torras. Object modeling using a tof camera under an uncertainty reduction approach. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1306–1312, Anchorage, Alaska, USA, May 2010.

[82] S. Furman. *rad Robots*. Kyle Cathie Limited, London, UK, 2009.

[83] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31: 77–98, May 2001.

[84] V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, Dec. 1998.

[85] D. Gallup, J.-M. Frahm, and M. Pollefeys. A Heightmap Model for Efficient 3D Reconstruction from Street-Level Video. In *IEEE Conf. of Computer Vision and Pattern Recognition*, 2010.

[86] H. M. Geduld. *Robots, Robots, Robots*, chapter Genesis II: The Evolution of Synthetic Man. New York Graphic Society, 1978.

[87] M. Gentilucci. Object Motor Representation and Reaching-Grasping Control. *Neuropsychologia*, 40(8):1139–1153, 2002.

[88] T. Gevers and A. Smeulders. Colour based object recognition. *Pattern Recognition*, 32:453–464, March 1999.

[89] J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1979.

[90] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp Planning Via Decomposition Trees. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4679–4684, 2007.

[91] M. Goodale. Separate Visual Pathways for Perception and Action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[92] M. A. Goodale, J. P. Meenan, H. H. Bülthoff, D. A. Nicolle, K. J. Murphy, and C. I. Racicot. Separate Neural Pathways for the Visual Analysis of Object Shape in Perception and Prehension. *Current Biology*, 4(7):604–610, 1994.

[93] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *IEEE Int. Conf. on Computer Vision (ICCV)*, volume 2, pages 1458–1465, 2005.

[94] J. Grezes and J. Decety. Does Visual Perception of Object Afford Action? Evidence from a Neuroimaging Study. *Neuropsychologia*, 40(2):212–222, 2002.

[95] T. Grundmann, M. Fiegert, and W. Burgard. Probabilistic rule set joint state update as approximation to the full joint state estimation applied to multi object scene analysis. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2047–2052. IEEE, 2010.

[96] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42:335–346, 1990.

[97] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *COMPUTER GRAPHICS-NEW YORK-ASSOCIATION FOR COMPUTING MACHINERY-*, 26:71–71, 1992.

[98] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng. Reactive grasping using optical proximity sensors. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2098–2105, 2009.

[99] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1228 – 1235, Taipei, Taiwan, October 2010.

[100] K. Hsiao, L. Kaelbling, and T. Lozano-Perez. Task-driven tactile exploration. In *Robotics: Science and Systems (RSS)*, Zaragoza, Spain, June 2010.

[101] K. Hübner and D. Kragic. Selection of Robot Pre-Grasps using Box-Based Shape Approximation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1765–1770, 2008.

[102] 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 *Int. Conf. on Advanced Robotics (ICAR)*, pages 1–6, 2009.

[103] J. R. Hurley and R. B. Cartell. The Procrustes Program: Producing Direct Rotation to test a hypothesized factor structure. *Behavioural Science*, 7: 258–262, 1962.

[104] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, Oct. 1996.

[105] iRobot. Roomba. `www.irobot.com`, Accessed Oct. 2011.

[106] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2:194–203, 2001.

[107] N. R. Jared Glover, Daniela Rus. Probabilistic models of object geometry for grasp planning. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008.

[108] I. Kamon, T. Flash, and S. Edelman. Learning to grasp using visual information. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2470–2476, 1994.

[109] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Gaussian Processes for Object Categorization. *Int. Jour. of Computer Vision*, 88:169–188, 2010.

[110] D. Katz, A. Orthey, and O. Brock. Interactive perception of articulated objects. In *Int. Symposium of Experimental Robotics (ISER)*, pages 01–15, 2010.

[111] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

[112] M. Kazhdan. Poisson Surface Reconstruction (Version 2). `http://www.cs.jhu.edu/~misha/Code/PoissonRecon/`, 2006.

[113] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics symposium on Geometry processing (SGP)*, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[114] J. Kenney, T. Buckley, and O. Brock. Interactive segmentation for manipulation in unstructured environments. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, ICRA'09, pages 1343–1348, Piscataway, NJ, USA, 2009. IEEE Press.

[115] H. Kjellström, J. Romero, and D. Kragic. Visual Object-Action Recognition: Inferring Object Affordances from Human Demonstration. *Computer Vision and Image Understanding*, pages 81–90, 2010.

[116] H. Kjellström, J. Romero, and D. Kragic. Visual object-action recognition: Inferring object affordances from human demonstration. *Comput. Vis. Image Underst.*, 115:81–90, January 2011.

[117] E. Klingbeil, B. Carpenter, O. Russakovsky, and A. Y. Ng. Autonomous operation of novel elevators for robot navigation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 751 – 758, Anchorage, AK, USA, may 2010.

[118] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib. Grasping with application to an autonomous checkout robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2837–2844, 2011.

[119] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 26(2):147–159, 2004.

[120] D. Kraft, N. Pugeault, E. Baseski, M. Popovic, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krueger. Birth of the object: Detection of objectness and extraction of object shape through object action complexes. *Int. Jour. of Humanoid Robotics*, pages 247–265, 2009.

[121] D. Kragic. *Visual Servoing for Manipulation : Robustness and Integration Issues.* PhD thesis, KTH, Numerical Analysis and Computer Science, NADA, 2001.

[122] D. Kragic, M. Björkman, H. I. Christensen, and J.-O. Eklundh. Vision for robotic object manipulation in domestic settings. *Robotics and Autonomous Systems*, 52(1):85–100, 2005.

[123] M. Krainin, B. Curless, and D. Fox. Autonomous generation of complete 3d object models using next best view manipulation planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5031–5037, May 2011.

[124] O. B. Kroemer, R. Detry, J. Piater, and J. Peters. Combining active learning and reactive control for robot grasping. *Robotics and Autonomous Systems*, 58:1105–1116, September 2010.

[125] J. Kuffner and S. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 995 –1001, Piscataway, NJ, USA, 2000. IEEE Press.

[126] KUKA. KR 5 sixx R850. `www.kuka-robotics.com`, Accessed Oct. 2011.

[127] F.-F. L. and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531, 2005.

[128] K. Lai, L. Bo, X. Ren, and D. Fox. Sparse distance learning for object recognition combining rgb and depth information. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4007–4013, Shanghai, China, May 2011.

[129] K. Lai, L. Bo, X. Ren, and D. Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1817 – 1824, 2011.

[130] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Computer Science Dept, Iowa State University, 1998.

[131] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

[132] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng. Learning to grasp objects with multiple contact points. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5062–5069, 2010.

[133] B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*, pages 508–524, 2006.

[134] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2036–2043, 2009.

[135] Y. Li and N. Pollard. A Shape Matching Algorithm for Synthesizing Humanlike Enveloping Grasps. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 442–449, Dec. 2005.

[136] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 303–308, New York, NY, USA, 2004. ACM.

[137] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 0, pages 1972–1979, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[138] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface reconstruction algorithm. In *SIGGRAPH*, pages 163–169, 1987.

[139] D. Lowe. Object recognition from local scale-invariant features. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1150–1157, September 1999.

[140] G. Luo, N. Bergström, M. Björkman, and D. Kragic. Representing actions with kernels. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2028 – 2035, San Francisco,USA, September 2011.

[141] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on Multiple-View geometric cues with application to robotic towel folding. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2308 – 2315, 2010.

[142] A. Maldonado, U. Klank, and M. Beetz. Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2586–2591, Oct. 2010.

[143] T. Malisiewicz and A. A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*, December 2009.

[144] D. Martens, B. Baesens, T. V. Gestel, and J. Vanthienen. Comprehensible Credit Scoring Models using Rule Extraction from Support Vector Machines. *European Jour. of Operational Research*, 183(3):1466–1476, December 2007.

[145] Z. C. Marton, L. Goron, R. B. Rusu, and M. Beetz. Reconstruction and Verification of 3D Object Models for Grasping. In *Int. Symposium on Robotics Research (ISRR)*, Lucerne, Switzerland, August 2009.

[146] Z. C. Marton, R. B. Rusu, D. Jain, U. Klank, and M. Beetz. Probabilistic categorization of kitchen objects in table settings with a composite sensor. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4777 – 4784, St. Louis, MO, USA, 10 2009.

[147] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz. General 3D Modelling of Novel Objects from a Single View. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3700 – 3705, Taipei, Taiwan, October 18-22 2010.

[148] Z. C. Marton, D. Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz. Hierarchical object geometric categorization and appearance classification for mobile manipulation. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 365 – 370, Nashville, TN, USA, December 2010.

[149] Z. C. Marton, D. Pangercic, N. Blodow, and M. Beetz. Combined 2D-3D Categorization and Classification for Multimodal Perception Systems. *Int. Jour. of Robotics Research (IJRR)*, 2011. Accepted.

[150] C. R. Maurer, G. B. Aboutanos, B. M. Dawant, R. J. Maciunas, and J. M. Fitzpatrick. Registration of 3-D Images Using Weighted Geometrical Features. *IEEE Trans. on Medical Imaging*, 15(6):836–846, December 1996.

[151] G. Metta and P. Fitzpatrick. Better Vision through Manipulation. *Adaptive Behavior*, 11(2):109–128, 2003.

[152] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4):110–122, 2004.

[153] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic Grasp Planning Using Shape Primitives. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1824–1829, 2003.

[154] N. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Annual Symposium on Computational Geometry*, page 328. ACM, 2003.

[155] A. Morales, E. Chinellato, A. Fagg, and A. del Pobil. Using Experience for Assessing Grasp Reliability. *Int. Jour. of Humanoid Robotics*, 1(4):671–691, 2004.

[156] A. Morales, P. Azad, T. Asfour, D. Kraft, S. Knoop, R. Dillmann, A. Kargov, C. Pylatiuk, and S. Schulz. An Anthropomorphic Grasping Approach for an Assistant Humanoid Robot. In *Int. Symposium on Robotics Research*, pages 149–152, 2006.

[157] H. Moravec and A. E. Elfes. High resolution maps from wide angle sonar. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 116 – 121, March 1985.

[158] G. Mori, S. Belongie, J. Malik, and S. Member. Efficient shape matching using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 27:1832–1837, 2005.

[159] G. L. Murphy. *The Big Book of Concepts (Bradford Books)*. The MIT Press, Mar. 2002.

[160] A. Newell and H. A. Simon. Computer science as empirical inquiry: symbols and search. *Commun. ACM*, 19:113–126, March 1976.

[161] V.-D. Nguyen. Constructing stable grasps. *Int. Jour. on Robotics Research*, 8(1):26–37, 1989.

[162] S. T. O'Callaghan, F. Ramos, and H. Durrant-Whyte. Contextual occupancy maps using gaussian processes. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3478 – 3485, Kobe, Japan, May 2009.

[163] K. Pahlavan and J.-O. Eklundh. A head-eye system – analysis and design. *CVGIP: Image Understanding*, 56(1):41 – 56, 1992.

[164] V. Papadourakis and A. Argyros. Multiple objects tracking in the presence of long-term occlusions. *Comput. Vis. Image Underst.*, 114:835–846, July 2010.

[165] C. Papazov and D. Burschka. Stochastic optimization for rigid point set registration. In *Int. Symposium on Advances in Visual Computing (ISVC)*, pages 1043–1054, Berlin, Heidelberg, 2009. Springer-Verlag.

[166] C. Papazov and D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *ACCV (1)*, volume 6492 of *Lecture Notes in Computer Science*, pages 135–148. Springer, 2010.

[167] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal. Online Movement Adaptation based on Previous Sensor Experiences. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 365 – 371, San Francisco,USA, September 2011.

[168] R. Pelossof, A. Miller, P. Allen, and T. Jebera. An svm learning approach to robotic grasping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3512–3518, 2004.

[169] M. Persson, T. Duckett, and A. J. Lilienthal. Virtual sensors for human concepts - building detection by an outdoor mobile robot. *Robotics and Autonomous Systems*, 55(5):383–390, May 31 2007.

[170] L. Petersson, D. Austin, and D. Kragic. High-level control of a mobile manipulator for door opening. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 3, pages 2333–2338, Oct. 2000.

[171] R. Pfeifer and J. C. Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press, Nov 2006.

[172] M. Popović, G. Kootstra, J. A. Jørgensen, D. Kragic, and N. Krüger. Grasping unknown objects using an early cognitive vision system for general scene understanding. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 987 – 994, San Francisco, USA, September 2011.

[173] V. Pradeep, K. Konolige, and E. Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *Int. Symposium on Experimental Robotics (ISER)*, New Delhi, India, 12/2010 2010.

[174] M. Prats. *Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback*. PhD thesis, Robotic Intelligence Laboratory, Jaume-I University, June 2009.

[175] R. Prim. Shortest connection networks and some generalisation. *Bell System Technical Journal*, 36:1389–1401, November 1957.

[176] W. Prinz. Perception and action planning. *European Jour. of Cognitve Psychology*, 9(2):129–154, 1997.

[177] A. Pronobis. *Semantic Mapping with Mobile Robots*. PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, June 2011.

[178] A. Pronobis and P. Jensfelt. Hierarchical multi-modal place categorization. In *European Conf. on Mobile Robots (ECMR'11)*, Örebro, Sweden, Sept. 2011.

[179] M. Przybylski and T. Asfour. Unions of balls for shape approximation in robot grasping. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1592–1599, Taipei, Taiwan, October 2010. IEEE.

[180] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng. Grasping novel objects with depth segmentation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2578–2585, Taipei, Taiwan, October 2010.

[181] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.

[182] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic. An Active Vision System for Detecting, Fixating and Manipulating Objects in Real World. *Int. J. of Robotics Research*, 29(2-3), 2010.

[183] J. Ratcliff. Convex decomposition library, 2009. URL `http://codesuppository.blogspot.com/2009/11/convex-decomposition-library-now.html`.

[184] M. Richtsfeld and M. Vincze. Grasping of Unknown Objects from a Table Top. In *ECCV Workshop on 'Vision in Action: Efficient strategies for cognitive agents in complex environments'*, Marseille, France, September 2008.

[185] G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, 27(1):169–192, 2004.

[186] J. Romero, H. Kjellström, and D. Kragic. Modeling and evaluation of human-to-robot mapping of grasps. In *ICAR : 2009 14TH INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS*, pages 228–233. IEEE, 2009.

[187] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439, 1976.

[188] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics*, 23(3):309–314, August 2004.

[189] M. Roy, S. Foufou, and F. Truchetet. Mesh Comparison Using Attribute Deviation Metric. *Int. Jour. of Image and Graphics*, 4:127–, 2004.

[190] S. D. Roy. Active recognition through next view planning: a survey. *Pattern Recognition*, 37(3):429–446, 2004.

[191] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, int. edition edition, 2003.

[192] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universität Müchen, Germany, October 2009.

[193] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1–4, Shanghai, China, May 2011.

[194] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1848–1853, Piscataway, NJ, USA, 2009. IEEE Press.

[195] R. B. Rusu, A. Holzbach, G. Bradski, and M. Beetz. Detecting and segmenting objects for mobile manipulation. In *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE Int. Conf. on Computer Vision (ICCV)*, Kyoto, Japan, September 27 2009.

[196] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz. Laser-based perception for door and handle identification. In *Int. Conf. on Advanced Robotics (ICAR)*, pages 1–7, Munich, Germany, 06 2009.

[197] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2155 – 2162, Taipei, Taiwan, October 2010.

[198] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, Aug. 2011.

[199] P. Sakov. Natural Neighbours interpolation library. `http://code.google.com/p/nn-c/`, 2010.

[200] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng. Robotic Grasping of Novel Objects. *Neural Information Processing Systems*, 19:1209–1216, 2007.

[201] A. Saxena, L. Wong, and A. Y. Ng. Learning Grasp Strategies with Partial Shape Information. In *AAAI Conf. on Artificial Intelligence*, pages 1491–1494, 2008.

[202] D. Schlangen and G. Skantze. A General, Abstract Model of Incremental Dialogue Processing. In *Conf. of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 710–718, Athens, Greece, 2009.

[203] SCHUNK. Sdh. `www.schunk.com`, Accessed Oct. 2011.

[204] K. Shimoga. Robot Grasp Synthesis Algorithms: A Survey. *Int. Jour. of Robotic Research*, 15(3):230–266, 1996.

[205] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conf. Computer Vision (ECCV)*, pages 1–15, 2006.

[206] B. Siciliano and O. Khatib. Introduction. In *Springer Handbook of Robotics*, pages 1–4. Springer-Verlag, Berlin, Heidelberg, 2008.

[207] H. A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, USA, 3 edition, 1996.

[208] G. Skantze. Jindigo: A Java-based Framework for Incremental Dialogue Systems, 2010. `www.jidingo.net`.

[209] D. Song, C. H. Ek, K. Hübner, and D. Kragic. Multivariate discretization for bayesian network structure learning in robot grasping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1944–1950, Shanghai,China, May 2011.

[210] J. Speth, A. Morales, and P. J. Sanz. Vision-Based Grasp Planning of 3D Objects by Extending 2D Contour Based Algorithms. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2240–2245, 2008.

[211] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele. Functional Object Class Detection Based on Learned Affordance Cues. In *Int. Conf. on Computer Vision Systems (ICVS)*, volume 5008 of *LNAI*, pages 435–444. Springer-Verlag, 2008.

[212] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich. Common metrics for human-robot interaction. In *HRI '06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40, New York, NY, USA, 2006. ACM.

[213] J. Stückler, R. Steffens, D. Holz, and S. Behnke. Real-Time 3D Perception and Efficient Grasp Planning for Everyday Manipulation Tasks. In *European Conf. on Mobile Robots (ECMR)*, Örebro, Sweden, September 2011.

[214] F. Stulp, E. Theodorou, M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal. Learning motion primitive goals for robust manipulation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 325 – 331, San Francisco,USA, September 2011.

[215] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conf.*, 2009.

[216] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, and J. Wikander. Demonstration based Learning and Control for Automatic Grasping. *Jour. of Intelligent Service Robotics*, 2(1):23–30, August 2008.

[217] J. P. Telotte. *Replications – A Robotic History of the Science Fiction Film*. University of Illinois Press, Urbana and Chicago, 1995.

[218] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz. Web-enabled Robots – Robots that use the Web as an Information Resource. *Robotics & Automation Magazine*, 18(2):58–68, 2011.

[219] S. Thrun and B. Wegbreit. Shape from symmetry. In *IEEE Int. Conf. on Computer Vision (ICCV)*, volume 2, pages 1824–1831, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[220] E. A. Topp. *Human-Robot Interaction and Mapping with a Service Robot: Human Augmented Mapping*. PhD thesis, Royal Institute of Technology, School of Computer Science and Communication, Stockholm,Sweden, September 2008.

[221] A. Turing. Computing machinery and intelligence. In N. Wardrip-Fruin and N. Montfort, editors, *The New Media Reader*, chapter 3, pages 50–66. The MIT Press, Cambridge and London, 2003.

[222] A. Ude and E. Oztop. Active 3-d vision on a humanoid head. In *Int. Conf. on Advanced Robotics (ICAR)*, pages 1–6, Munich, Germany, 2009.

[223] A. Ude, D. Omrcen, and G. Cheng. Making Object Learning and Recognition an Active Process. *Int. J. of Humanoid Robotics*, 5(2):267–286, 2008.

[224] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez, T. Asfour, and R. Dillmann. Visual Servoing for Humanoid Grasping and Manipulation Tasks. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 406–412, 2008.

[225] M. Vande Weghe, D. Ferguson, and S. Srinivasa. Randomized path planning for redundant manipulators without inverse kinematics. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, pages 477 –482, nov. 2007.

[226] G. Veruggio. Euron roboethics roadmap, January 2007.

[227] A. Vrečko, D. Skočaj, N. Hawes, and A. Leonardis. A computer vision integration model for a multi-modal cognitive system. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3140–3147, St. Louis, MO, USA, October 2009.

[228] M. J. Webster, J. Bachevalier, and L. G. Ungerleider. Connections of Inferior Temporal Areas TEO and TE with Parietal and Frontal Cortex in Macaque Monkeys. *Cerebral cortex*, 4(5):470–483, 1994.

[229] K. Welke, M. Przybylski, T. Asfour, and R. Dillmann. Kinematic calibration for saccadic eye movements. Technical report, Institute for Anthropomatics, Universität Karlsruhe, 2008.

[230] S. Wenhardt, B. Deutsch, J. Hornegger, H. Niemann, and J. Denzler. An information theoretic approach for next best view planning in 3-d reconstruction. In *Int. Conf. on Pattern Recognition*, volume 1, pages 103–106, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[231] W. Wohlkinger and M. Vincze. Shape-based depth image to 3d model matching and classification with inter-view similarity. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4865–4870, San Francisco, USA, Sept 2011.

[232] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: understanding multi-object traffic scenes. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 467–481, Berlin, Heidelberg, 2010. Springer-Verlag.

[233] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010. Software available at `http://octomap.sf.net/`.

[234] YafaRay. `http://www.yafaray.org/`, Accessed Oct. 2011.