Technische Universität Berlin
School IV - Electrical Engineering and Computer Science
Department of Computer Engineering and Microelectronics
Robotics and Biology Laboratory

Diplomarbeit

Active Visual Primitives for Perception and Grasping

---

presented by

Georg Bartels
Matr.-Nr.: 306094
georg.bartels@gmail.com
Technische Informatik

Date of submission: **December 17, 2012**

Examiners: **Prof. Dr. Oliver Brock, Prof. Dr. Verena Hafner**
Advisors: **Prof. Dr. Oliver Brock, Clemens Eppner**

Eidesstattliche Erklärung

Die selbständige und eigenhändige Ausfertigung versichert an Eides statt

Berlin, den

Unterschrift

Abstract

Robotic grasping of unknown objects is a central research topic in robotics, which so far has been neither completely understood nor solved. It is an essential task that is often part of more complex assignments, and therefore grasping is a key ability robots have to be equipped with before becoming truly autonomous.

Inspired by the observation that the inherent compliance of the human hand causes it to automatically suit to the shape of an object very well, it has been argued that the problem of finding an adequate grasp can be reduced to finding an appropriate pregrasp [21]. Since these pregrasps are described in a low-dimensional space, the requirements for the perceptual component of a grasping system are also reduced. This is corroborated by findings from neuroscientific research indicating that human pregrasp posture control involves synergies that greatly reduce the problem's dimensionality [15].

This thesis describes the design of active visual primitives which reliably perceive common geometrical properties of objects corresponding to successful pregrasp configurations of the robotic hand. In contrast to existing approaches, the presented primitives are designed as active vision controllers which determine their exploration trajectory based on current, past and expected observations. Deployment of the active vision paradigm allows to make fewer assumptions about the shapes of the objects, and hence enables the primitives to generalize over various object shapes.

Extensive experimental evaluation in real world scenarios shows that the primitives reliably detect the desired information across an ample variety of objects and yield convincing grasping results. Additionally, a strong correlation between perceived object shapes and successful preshapes of the robotic hand is found for big objects, while for medium-sized objects the results do not confirm such a connection.

This work shows that solving grasping of unknown objects in the space of pregrasps is a very promising approach. The outcome of this research also indicates that there is a non-trivial relationship between the size of an object and the perceived shape and successful pregrasps, which deserves further investigation.

Keywords: robotics, grasping, perceptual primitives, active vision, compliance, object estimation.

Zusammenfassung

Das Greifen von unbekannten Objekten ist ein zentrales Problem der Robotik, das bis heute weder komplett verstanden noch gelöst wurde. Oft ist es Teilschritt weit komplexerer Aufgaben. Daher ist Greifen eine Schlüsselfähigkeit, über die Roboter verfügen müssen, bevor sie als autonom bezeichnet werden können.

Die Beobachtung, dass die Nachgiebigkeit der menschlichen Hand automatisch zu sehr großen Kontaktflächen zwischen Hand und Objekt führt, motivierte andere Forscher, das Finden eines guten Griffs auf das Finden eines adäquaten Vorgriffs zurückzuführen [21]. Da diese Vorgriffe in einem subdimensionalen Unterraum definiert sind, verringern sich auch die Anforderungen an das Wahrnehmungsmodul eines Greifsystems. Erkenntnisse neurowissenschaftlicher Forschung, die zeigen, dass die menschliche Regelung der Handform vor einem Griff Synergien nutzt, die die Dimensionalität des Problems stark verringern, stützen diese These [15].

Diese Arbeit beschreibt das Design von aktiven Wahrnehmungsprimitiven, die zuverlässig gemeinsame geometrische Eigenschaften von Objekten detektieren, die Eigenschaften von erfolgreichen Vorgriffen entsprechen. Im Gegensatz zu bestehenden Ansätzen wurden die Primitiven als aktive visuelle Regler entworfen, die ihre Erkundungsbahnen auf Basis aktueller, vergangener und erwarteter Beobachtungen bestimmen. Das Paradigma der aktiven Wahrnehmung erlaubt es, weniger Annahmen über die Objektformen zu machen, weshalb die Primitive besser über verschiedenste Objektformen generalisieren können.

Ausführliche Experimente in und außerhalb von Simulation zeigen, dass die Primitive zuverlässig die gewünschten Informationen in unterschiedlichsten Objekten detektieren, und liefern überzeugende Greifergebnisse. Außerdem wird für große Objekte eine starke Korrelation zwischen der wahrgenommen Form eines Objekts und den erfolgreichen Vorformen der Roboterhand gefunden, während sich eine solche Verbindung für mittelgroße Objekte nicht schlussfolgern lässt.

Diese Arbeit zeigt, dass es vielversprechend ist, das Greifen unbekannter Objekte im Raum der Vorgriffe zu lösen. Die Resultate dieser Forschung zeigen auch deutlich, dass es eine nicht-triviale Verbindung zwischen der Größe eines Objekts, seiner wahrgenommenen Form und erfolgreichen Handvorformen gibt, die in weiterführender Forschung untersucht werden sollte.

Table of Contents

Preface **I**
Eidesstattliche Erklärung I
Abstract II
Table of Contents IV
List of Figures VI
Notation Remarks VIII

1 Introduction **1**

2 Related Work and Background **5**
2.1 Grasping 5
2.1.1 Background 5
2.1.2 Related Work 7
2.2 Active Vision 14
2.2.1 Background 15
2.2.2 Related Work 16

3 Methods **19**
3.1 Analysis of the Problem 19
3.2 Robotic Platform and Software Interface 21
3.3 Image Processing and Image Feature Extraction 23
3.4 Visual Servoing for Centering and Aligning of a Blob 25
3.5 Distance Primitive 31
3.6 Axis Primitive 39
3.7 Shape Primitives 47
3.7.1 Sphere Primitive 49
3.7.2 Cylinder Primitive 51
3.7.3 Box Primitive 52

4 Experimental Results and Discussion **56**
4.1 Estimation of Geometrical Primitives in Simulation 56
4.2 Generalizing Properties of the Visual System 60
4.3 Real-World Grasping Performance 65

5 Conclusion **80**
5.1 Summary 80
5.2 Future Work 80

References **82**
Images and Internet 82
Bibliography 82

List of Figures

1.1 Possible future applications of autonomous robots. 1
1.2 Human hand tightly grasping an object. 2
1.3 Illustration of the Mitten-Thought-Experiment. 3

2.1 Example of a form closure grasp. 6
2.2 An example force closure grasp. 6
2.3 Cone of possible forces due to contact friction. 7
2.4 Object approximation with Shape Primitives 9
2.5 Approach directions of Shape Primitives. 9

3.1 Distinct preshapes of the BarrettHand 20
3.2 WAM arm with BarrettHand and camera 22
3.3 HSV color space 23
3.4 Pinhole camera model 26
3.5 Projective geometry model 27
3.6 Reference frame and homogeneous transformations 29
3.7 Results of center and align controllers 31
3.8 Basic exploration trajectory of the camera A 32
3.9 Basic exploration trajectory of the camera B 32
3.10 Exploration scenarios of Center Primitive 33
3.11 Spherical coordinates 35
3.12 Influence of different gains of the Distance Primitive 37
3.13 Confidence scoring of the Distance Primitive. 38
3.14 Sample confidence plot of Distance Primitive. 39
3.15 Orientation gradient during exploration. 40
3.16 Corrective motion to find main axis of object. 41
3.17 Influence of gain parameter on axis estimation 44
3.18 Confidence scoring of the Axis Primitive 46
3.19 Sample confidence plot of the Axis Primitive 47
3.20 Discriminitive blob properties for different objects 48
3.21 Scoring function for range of blob sizes 50
3.22 Definitions for size estimation of sphere 50
3.23 Definitions for size estimation of cylinder 52
3.24 Definitions for size estimation of box 53

4.1 Perfect objects – errors in spatial estimation 58
4.2 Perfect objects – errors in size estimation 59
4.3 Perfect objects – confidence scores 60
4.4 Shape perception of a cup. 61
4.5 A cup and its estimated bounding cylinder. 61
4.6 Shape perception of an angular cup. 62
4.7 An angular cup and its estimated bounding cylinder. 62
4.8 Shape perception of a beer bottle. 63
4.9 Beer bottle and its estimated bounding cylinder. 63
4.10 Shape perception of two stacked bowls. 64
4.11 Two stacked Bowls and their estimated bounding sphere. 64
4.12 Shape perception of a dog. 65
4.13 A dog and its estimated bounding box. 65
4.14 Objects that were used in the grasping experiments. 67
4.15 Results of grasping experiment with an apple 69
4.16 Results of grasping experiment with a bell pepper 70
4.17 Results of grasping experiment with a banana 71
4.18 Results of grasping experiment with a toy bridge 72
4.19 Results of grasping experiment with a spectacle case 73
4.20 Results of grasping experiment with a sponge 74
4.21 Results of grasping experiment with a soccer ball 75
4.22 Results of grasping experiment with a whiskey case. 76
4.23 Results of grasping experiment with a board game box. 77
4.24 Relation between object shape, chosen pregrasps, and grasp success. . 78

Notation Remarks

I will use the following notations in my mathematical formulas:

- Vectors are denoted by a lower case bold: $\mathbf{v}$
- Matrices are denoted by an upper case bold: $\mathbf{M}$
- Scalars are denoted with mixed case italics: S, r
- Homogenous transforms that describe frame B with respect to frame A are denoted as: ${}^{A}_{B}\mathbf{T}$

1 Introduction

Grasping is an everyday task that humans take for granted. But the intuition that everyday occurrence indicates simplicity is unfortunately misleading. Grasping is pretty hard; at least for robots. Up to today, a lot of research efforts have gone into providing robots with reliable grasping capabilities that enable them to autonomously perform meaningful actions in unstructured environments, i.e. surroundings that have not been especially co-designed to fit the robot's needs. Possible applications lie in taking care of elderly or disabled people, replacing human labor in hazardous working environments, or aiding with unpleasant household tasks. Until today, these visions have remained just that, because robots still lack reliable basic manipulation skills such as grasping.

Figure 1.1: Unstructured environments in which autonomous robots could help or replace humans, from left to right: Aiding in care taking, e.g. lifting of people (image taken from [1]), taking over boring household work (image taken from [2]), or replacing humans in hazardous working environments (image taken from [3]).

But grasping is not only hard for robots. Humans are outstanding learning machines, especially when it comes to learning tasks that include sensorimotor coupling. Still, it takes years until we have perfected our abilities and reach the kind of faultless and universal grasping capabilities that seem so natural to us. In fact, the different developmental stages during childhood appear to be carefully tailored to facilitate this learning process. Small children, for example, lack the ability to oppose their thumbs which in effect reduces the complexity of hand control. It is only in later years in childhood development that this feature emerges to enable more precise and delicate grasping and manipulation; an obvious hint that acquiring the high performance level of human grasping is not trivial at all.

The idea of grasping as an everyday task does, however, lead to a second point: As a task that is repeated excessively, it is crucial for performing more complex feats. Consequently, reliable grasping is a key ability that needs to be developed

rather early on the road towards universally deployable robots. Without perfect grasping, nobody would dare to deploy a robot to remove his precious champagne glasses from a dish-washer, nor to help his grandma out of bed, not to even mention sending robots to remove atomic material from an out-of-control power plant.

Any attempt at developing an autonomous grasping system should start with a contemplation of the tool that is used for grasping: The hand. Consider how a human hand firmly holds on to a grasped object. How the fingers wrap around it, and how the skin deforms to perfectly match the shape of the object. This compliance of the hand to the object leads to huge contact surfaces, which in turn lead to a tight and secure grip – one of the main goals of a lot of grasping approaches. Thus, the ingenious design of the compliant human hand with its many degrees of freedom (DOF) allows almost arbitrarily good matching of hand and object.

Figure 1.2: Example of a human grasp: The hand with all its built-in compliance tightly wraps around the details of the object to establish big contact surfaces, and thus guarantees stable hold of the object.

Unfortunately, a lot of DOF also hint at computationally demanding control strategies. The human hand, for example, has 20 controllable DOF [4], not counting the 6 DOFs of the wrist pose. If one assumes 6 distinct configurations for each joint, that equals to 6^{20} or roughly 10^{15} hand configurations; a huge space to control and search for a correct grasping configuration. There has got to be a way to reduce this search space to a feasible size.

In fact, neuroscientific research into human grasping has shown that humans do

not control all DOF of the hand in a decoupled way. Principal component analysis (PCA) of human hand preshapes before grasping of various objects revealed that the first two PCA components could account for more than 80% of the difference in preshapes across objects and subjects [15]. This indicates that selection of the preshape of the hand is done in a space of much lower dimensions, and that the object through its shape forces the compliant hand into its final configuration during the closing of the hand. Thus, it is not necessary to explicitly calculate or plan the final configuration of the hand to grasp a given object.

The "Mitten-Thought-Experiment" may help to further underline the importance of the hand's compliance for avoiding the need to analytically compute a detailed hand configuration. Consider a subject that is blind-folded and wears a thick mitten glove. Thus sensory-deprived and equipped with a super-compliant hand, the participant is to follow the instructions of the experimenter. The experimenter is now telling the subject how to form his or her hand, then positions the object relative to the hand, and tells the subject to close the hand. It is conceivable that such a setting yields outstanding grasping performance, even though neither the experimenter nor the subject have a chance to pre-calculate the detailed final position of the hand. It is all taken care of by the compliance of the mitten during hand closing.

Figure 1.3: The Mitten-Thought-Experiment: A sensory-deprived subject who is wearing a thick mitten can successfully grasp objects following instructions from an experimenter; due to the huge compliance of the mitten hand.

Thus, we have reduced the problem of finding the right grasp in the huge space of hand configurations to that of finding the right pregrasp (the combination of preshape and 6 DOF pose of the wrist) in a much smaller sub-space. This is the main idea behind the grasping approach that I present in this work: Provided a compliant grasping mechanism, successful grasping can be achieved by only selecting the

correct preshape of the hand (from a small set of shapes) and the correct rough pose of the hand in relation to the object.

The information that enables this selection has to come from a visual perception system. However, since I am only interested in task-specific and, as I have motivated, pretty rough shape and spatial information, it is possible to facilitate the vision task. There is no need to perceive the delicate and fragile nature of the small and big hand of a pocket watch to successfully grasp the pocket watch. On the other hand, there is one key requirement the vision component has to fulfill: Just as the compliant hand is able to generalize with a couple of preshapes over a huge variety of objects that are vastly different in their details, the visual system has to display the same kind of generalization property.

In this work, I propose to use visual primitives that follow the active vision paradigm to meet exactly this need. Visual primitives are components of the visual system that are designed to reliably detect a particular and very general 3-dimensional property in a broad set of objects, such as the centroid, or the main axis of an object, or membership in a specific class of shapes. Each of theses extracted properties is assumed to directly correspond to properties of the pregrasp of the hand that is best suited to grasp the object, thus grounding the purpose of perception in the needs of grasping.

The design of the visual system as a primitive-based system leads to several design advantages and requirements. Since each of the primitives only needs to extract its particular information, all of them can be designed to be as simple as necessary to be best suited to their job, thus reducing the complexity of the perceptual task to a minimum. On the other hand, in order to guarantee the necessary generalization capabilities, the primitives need to use image features that work with as few assumptions about the physical shape of the objects as possible.

The rest of this document is structured as follows: In Chapter 2, I review related work on grasping to show parallels and differences between my approach and existing work on robotic grasping. Additionally, a small introduction of the active vision paradigm is given to justify its usage in the proposed system. Chapter 3 describes the design and implementation of the visual primitives in detail. The experimental evaluation of the primitives in simulation and real world is presented in chapter 4. Chapter 5 concludes my thesis with a short summary of the findings and propositions for directions of future research.

2 Related Work and Background

As outlined in the introduction, the aim of this project is to develop active visual primitives that reliably extract grasping-specific geometrical information from a wide set of objects. Therefore, it is necessary to consider background and related work from two different fields: Classical grasping and active vision approaches. In this chapter I will give an overview of the basic concepts and relevant work that has been done before, show how it relates to my project, and outline how the proposed algorithm offers a new perspective on the problems of grasping and vision for grasping.

2.1 Grasping

2.1.1 Background

In this section I will describe the basic concepts and definitions in classical robotic grasping analysis that are necessary to understand a lot of the grasp planning algorithms that will be presented in the related work section.

Form and Force Closure

In robotic grasping the most fundamental ability of a hand is to restrain the movement of a grasped object, i.e. hold it. There are two widely used concepts of grasp restrain that have proven to be intuitively appealing for restraint description: *Form closure* and *force closure*. To get an intuition of the meaning of form closure consider an object grasped in one hand, with all joints of the fingers locked in place. This grasp is said to be form closure if it is impossible to move the object even a small amount, i.e. the object is encaged in every direction by a contact point. Figure 2.1 depicts a human form closure grasp. A grasped object is said to be force closed, if the hand can exert contact forces and moments on the object so that arbitrary non-contact forces and moments can be canceled out. Figure 2.2 depicts a human force closure grasp. For a good introductory text on robotic grasping refer to [16].

Contact Point Models

A lot of the early work done in robotic grasping was concerned with the mathematical analysis of the stability of grasps, i.e deciding whether a grasp is form or force closure.

Figure 2.1: A form closure grasp: The object is completely surrounded by contact points and cannot conceivably move at all.

Figure 2.2: An object held with a force closure grasp: Through contact points the hand can counter any external forces and moments, and stably hold the object.

In order to do this, one needs to first analyze the contact points and their properties. Generally, one assumes an object to be grasped by the hand at N idealized point contacts. Each of these contact points is then assumed to be either a frictionless point contact (i.e. the finger can only transmit force along the normal of the contact point), a point contact with friction (i.e. in addition to forces along the contact normal, tangential force can also be exerted by the finger), or a soft contact (i.e. here even a moment around the contact normal can be transmitted). These different contact models have been invented to model the different frictional properties of the contact surfaces. For a good review paper on grasp analysis and contact properties refer to [17].

Grasp Analysis with the Help of the Grasp Wrench Space

Once the contact points of a grasp have been assigned and the corresponding surface properties have been defined, the forces and moments at each of them are modeled as a so-called *wrench vector* (a 6×1 vector). For the forces along the contact normal of the point contact without friction, and the normal moments of the soft contact, this is straight-forward. The forces that can be exerted due to contact friction all lie within in a cone, where the normal and tangential forces limit its extends. The Coulomb friction model states that the following relation between tangential and normal forces holds: $||f_t|| \leq \mu f_n$, where μ is a material coefficient that describes the friction at the contact point [18]. In order to make the grasp analysis computationally feasible, the cone is often approximated with M vectors; Figure 2.3 depicts this idea.

After the contact wrenches have been assembled for all the N contact points, one

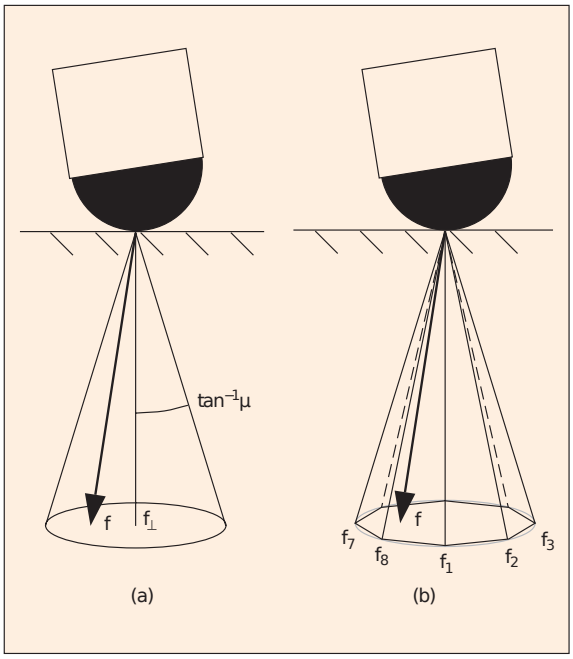


Figure 2.3: a) Any possible force **f** must lie within the cone defined by the friction coefficient μ. b) For grasp analysis the cone is approximated by M vectors, and **f** can be described as a linear combination of all of them. Image taken from [18].

can consider the grasp wrench space (GWS) – the space of wrenches that describes all the possible wrenches that can be exerted on the object. The GWS is one of the main tools to argue about the properties of a grasp. To decide force closure, for example, one calculates the convex hull of the GWS, and if the origin of the wrench space lies completely within it, the object is force closed [18]. Furthermore, Ferrari and Canny introduced two popular GWS-based measures that go beyond such a binary decision and assess the quality of a grasp: The first one can be used to assess the maximum contact force that needs to be applied at any contact, while the second criteria evaluates the overall sum of all contact forces [19]. In the GWS the former quality measure corresponds to the minimum distance of the convex hull of the GWS from the origin, while the latter is equivalent to the volume of the convex hull of the GWS [18].

2.1.2 Related Work

The following section will give an overview of the related work on grasping that has been presented by other researchers. It shall additionally outline how my proposed approach differs from theirs. Generally, one can categorize the existing grasping

algorithms based on the amount of information they need to perform. Firstly, there are approaches that need to have a detailed model of the object to work, thus they are only applicable for grasping of known objects. Algorithms for this class of problems typically perform a sampling-based search of good grasps. Then there are the approaches that try to exploit specific similarities, e.g. in appearance, between different objects to find good grasps, i.e. they work for familiar objects, too. These approaches often build around techniques from machine learning and rely on previously learned models. Finally, there are projects that try to tackle grasping of unknown objects, such as I do in this thesis. Here the common approach is to perceive general geometric properties that indicate reliable grasping possibilities.

Grasping of Known Objects

The first class of algorithms that I want to consider are the ones that try to tackle grasping of known objects. Typically, they perform a sampling-based search for good grasps that is guided by the quality measures that have been introduced by Canny et. Ferrari [19], i.e. they try to find grasps that minimize minimize the maximum contact force or the summed finger force. Construction and visualization of grasps with the help of these measures is often done with a grasping simulator, such as *GraspIt!* [18]. Note, however, that so far, I have only discussed work that focused on contact-level grasp analysis. So, no matter how a grasp planning algorithm uses the above work, it has to cope with its high input requirements: A high-fidelity model of the object and the hand are presupposed. There is also the issue of deriving the contact points from hand and object pose, because so far the analysis has been done at contact-level. *GraspIt!* tries to tackle some of these problems by providing a simulation environment that comes with several models of robots, hands, and objects. It allows the user to define the pose of hand and object, automatically extracts grasping points on the basis of collision detection, and implements several grasping measures to evaluate the quality of the simulated grasp.

The question remains of how to perceive all the necessary information, such as the poses of the hand and the object, in a real-world setting for unknown objects. Additionally, there is a need to specify the contact properties of both hand and object. Where shall this information come from for real-world unknown objects?

A first crack at combining *GraspIt!* and perception is presented in [20]. The authors present a system that extracts the pose of an object with a vision system in real-time, uses this pose to plan a grasp in *GraspIt!*, and visually monitors the grasp execution. The approach, however, has some strict limitations: The authors only

considered a single object for which the exact CAD model was known, user input was needed for initialization of the pose estimation (a set of corresponding points between image and CAD model had to be specified), and the actual grasp planning was also done manually: The user of the system used *GraspIt!* to try out several candidate grasps in simulation and to automatically calculate the grasp measures which can then be used by the user to choose a good grasp. So, the system basically represents a computer-aided manual grasp planning approach for known objects.

Figure 2.4: A coffee mug approximated by two shape primitives: A cylinder and a box. Image taken from [21].

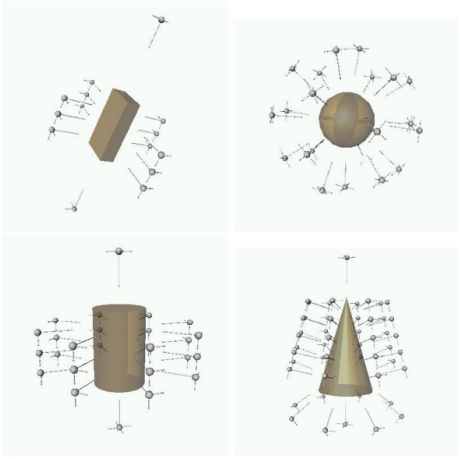


Figure 2.5: Specified approach directions for the four shape primitives: Sphere, box, cylinder, and cone. Image also taken from [21].

An attempt at including an automatic grasp planning algorithm into the *GraspIt!* framework is presented in [21]. The authors propose to generate a set of candidate grasps, score them using the quality metric from Ferrari and Canny [19] for the largest worst-case disturbance wrench that the grasp can resist, and select the candidate grasp with the highest score – all in simulation. In order to reduce the computation of the candidate grasp calculation, objects to be grasped are approximated with a number of geometrical shape primitives. Figure 2.4 depicts how such an approximation looks like for a coffee mug. For each of the primitives pregrasps were manually specified. Each pregrasp consisted of an approach direction of the hand and a corresponding hand preshape. Figure 2.5 shows the shape primitives and their respective approach directions. Candidate grasps were generated out of the pregrasps by analytically closing the fingers in simulation until collisions with the object occurred. Unfortunately, the authors did not include any real-world grasping experiments to evaluate their proposed algorithm. One reason might be that

no module for perceiving the shape primitives of real objects was included in the presented system; the authors named such an algorithm as one direction of future work. Another difference to the approach for grasping that I have outlined in the introduction is that the idea of the compliant hand as an inherent helper for grasping is not present in all stages of the algorithm: The authors propose an algorithm that generates grasps out of pregrasps with the help of compliance, but evaluates grasps and not pregrasps. The decision how to grasp an object is still done on contact-level, and thus computationally demanding. A very strong similarity, however, is the idea that objects can be modeled with the help of shape primitives, as the small details of an object are not necessarily important for pregrasp selection.

A different road towards automatic grasp planning is proposed in [22]. Inspired by insights from neuroscience that show that humans control only a subspace of the hand-configuration space [15], the authors introduce *eigengrasps* for robotic grasping. These eigengrasps span the space of all possible hand poses. They will serve to reduce the dimensionality of the searching problem during grasp planning, and additionally provide a generic control interface for different hand kinematics to allow planning for various hand designs. Based on the eigengrasps-presentation of pregrasp candidates the authors propose a simulated-annealing-based algorithm that searches for a grasp that maximizes the largest worst-case disturbance wrench that can be resisted, which is based on the same metric as described in the preceding paragraph. Deriving grasps out of pregrasps is done as in [21]: The fingers of the hand are closed in simulation until collisions are analytically detected. Evaluation is done in simulation only, and perception of the objects is not considered. Results show that planning in this dimensionality-reduced pregrasp space also yields force closure grasps in the majority of cases and enables a unified approach to grasp planning for different hand models. Interestingly, the proposed algorithm performs grasp planning in the space of pregrasps, only to then retreat to the computationally demanding tasks of grasps analysis on contact-level. The authors propose a modified grasp quality metric that is very similar to the standard metrics presented by Canny and Ferrari to evaluate their pregrasps. This new metric scales contact wrenches of certain *desired* contact points that do not touch the object with the calculated distance between desired contact point and object. As already outlined in the introductory chapter, my work is based on the same study, draws the same conclusion (i.e. it is enough to find the right pregrasp, which is situated in a space of lower dimensionality), but in contrast to [22] remains with the idea of just considering the space of pregrasps and ignoring contact-level analysis.

In a subsequent study the same authors used their eigengrasp planner to build a grasping database that aims at providing baselines grasps for a huge variety of objects, 3D models for grasping benchmarks, and a labeled dataset that might be used for machine learning approaches to robotic grasping with their *Columbia Grasp Database* [23]. Together with a set of 3D models of graspable objects, the database provides precalculated grasps that have been derived with the eigengrasp planner. Additionally, the authors presented an algorithm that finds the corresponding database object to the given CAD model of an object that shall be grasped. The precalculated grasps of that database object are then tried on the new object in simulation, and the grasp with the highest score is selected for execution. Experimental results in simulation showed that objects that were already in the database were correctly retrieved, and that the precomputed grasps could also generalize to objects that were similar to at least one object in the database. It is noteworthy that the authors report that calculating the roughly $240,000$ grasps took about one month of computation time, which highlights the high computational demand of the eigengrasp planning.

In order to expand the functionality of the Columbia Grasp Database [24] presents an algorithm that retrieves a suitable database object for a given object by just using a set range of sensor images of that given object. This is a valuable extension of the system because grasping of known objects, i.e. objects that are included in the database, will be facilitated because the retrieval no longer needs the entire CAD model of the object for comparison with the database.

In [25] Ciocarlie et al. present WillowGarage's software architecture for reliable grasping with the PR2 robot. One of the main contributions of the paper is to reveal the challenges that arise when one tries to integrate multiple state-of-the-art technologies that are necessary to achieve good grasping performance in everyday unstructured environments. In detail the authors used the following modules: Scene segmentation and object recognition, environmental modeling for collision avoidance, grasp planning for known and unknown objects, collision-free motion planning, and incorporation of techniques to use tactile sensing to correct errors that occur during grasp execution.

From the perspective of my project, the most relevant part of their system is the grasp planning module. After objects have been segmented from 3D point cloud data, they are divided into two groups: The ones that the algorithm from [24] has – with a certain confidence – matched to a known object in the object database of the system, and the ones that the algorithm could not find. For the former the pre-

computed grasp with the highest score is selected; according to the fashion in [24]. Due to the computational demands of grasp planning (the authors reported that computation of the possible grasps in the database took 4 hours for each object), a different heuristic approach was chosen for the latter, i.e. unknown, objects.
That heuristic is described in a second publication by Hsiao et al. [26]: Reasoning on the bounding box of the extracted 3D point cloud of an object, the algorithm generates a set of candidate grasps that are perpendicular to the main axis of the object and approach the object either from the top or the side. Finally, the grasp that maximizes the contact surface of the gripper and the objects' bounding box is chosen for execution. This approach has been inspired by findings from [27] that show that "humans tend to select grasps with wrist orientations that are orthogonal to the object's principal axis and its perpendiculars"[26], and is similar to the grasp execution that I will employ in my experiments. It is noteworthy, however, that in the cited work a pregrasp did not contain any preshape of the hand, because the simple structure of the gripper of the PR2 does not allow for such a differentiation. Thus, in contrast to my work, there was no further reasoning about the geometric shape of neither the known nor unknown objects was done. Experimental evaluation of the entire system on everyday objects yielded good results for both known and unknown objects, and outstanding performance when tactile information was used to correct errors during grasp execution.

With exception of the heuristic grasping module for unknown objects from Willow-Garage all of the above projects fall into the category of contact-level grasp planning. These approaches need detailed CAD models of the object to decide force closure and select a good grasp. Thus, these they are only applicable to grasping of known objects.

Grasping of Familiar Objects

A second class of approaches tries to grasp *familiar* objects. These projects follow the idea that previously seen objects and the current object share some easy to perceive key characteristics that indicate a good grasp. One such research project was presented in [28, 29]. The authors described a system that learns grasping points in usual camera images of objects in a supervised learning setting. At the heart of the project was the following assumption:

> "There are certain visual features that indicate good grasps, and that remain consistent across many different objects. [...] We propose a learn-

> ing algorithm that learns to use visual features to identify good grasping points across a large range of objects."

For training synthetic images with labeled stable grasping points were used to learn the prediction of the location of those points in an image. In order to translate these 2D points into 3D at least 2 different images, i.e. taken from different positions, of the object to grasp were needed. Triangulation of these 2D points yielded the 3D grasping point that was then grasped with a parallel plate gripper. Evaluation of the prediction of 2D grasping points was done on the synthetic images and showed good classification results (94.2% success rate). In order to investigate how well the learned model can generalize to real-world and especially to unknown objects and show the usefulness of the learned points for grasping, further experiments with real objects were done. Those objects were either similar (scaled in size, or contained small color or shape variations) to the synthetic ones from the training set, or completely new. Results showed that both the similar and novel objects could be reliably grasped (90% grasping success for similar and 87.5% for novel objects). The authors neither considered how the compliance of the hand could help during grasp execution, nor how the shape of an object may provide clues for grasp selection. Admittedly, such considerations would not have made a big impact, because a parallel plate gripper was employed that comes with limited compliance and supports only one kind of preshape.

A similar but different approach at grasping familiar objects is presented in [30]: The authors propose to learn grasping points in a supervised learning setting based on an image feature that represents the relative shape of a point on an object. This feature should encode the *shape context* of that point, i.e. how it relates to the overall shape of the object. In order to obtain 3D grasping points, 2D grasping point classification is done in both images of a stereo vision pair, then matched to find corresponding pairs, and finally triangulated. Learning performance based on this feature is compared for several classifiers to select the optimal setting. Training was done on synthetic labeled data, and evaluated in simulation and real-world. The results show high classification performance (generally well above 80%) for both known and unknown objects, both in simulation and real world. Since the authors tried to find stable grasping points for a precision pinch grasp only, they did not consider the question of selecting a preshape of the hand for grasping. Consequently, the aiding role of compliance of the hand during grasping is not discussed, either.

Grasping of Unknown Objects

Finally, there are approaches that investigate the question of grasping unknown objects, just as I do in my work. Huebner et. al. present an approach that is also working on the assumption that geometric shape of an object is a very important cue for choosing the right pregrasp [31]. They try to approximate objects with box primitives. A point cloud perception of an object from a pair of stereo cameras is approximated by a set of boxes using a fit-and-split algorithm until a desired quality of fit is achieved [32]. On the basis of the derived box primitives candidate graps are generated with a set of heuristics that take factors such as task dependency, box visibility, and box occlusion into account. An off-line trained artificial neural network is then used to select the most stable grasp. Evaluation of the work is done on only three different objects, and only in simulation. In contrast to my work, the authors do not employ an active vision scheme, thus there is no exploration of the object. Therefore, only a partial modeling of the object is done. Furthermore, only one kind of shape primitive is considered, which does not allow an investigation of the usefulness of different shape primitives for particular objects. On the other hand, since they model one object with various box primitives, their approach also easily scales to composite objects.

Dune et. al. also want to perceive the shape of an object to choose a good pregrasp [33]. They, however, do not model shapes with the help of primitives, but with a generic quadric-based approximation. An active vision scheme for a single camera that is mounted at a robotic manipulator is also deployed to explore an object from several observations. Next best views are chosen such that they reduce ambiguities and uncertainties in the perception. The final pregrasps is completely defined by properties of the perceived quadric: The end-effector position is given by the centroid of the quadric, the wrist orientation by the main axis of the quadric, and the size of the quadric determines the opening of the gripper. The authors did not further differentiate between different object shapes or preshapes of the hand. Unfortunately, experimental evaluations of the grasping approach are not reported.

2.2 Active Vision

When tackling the task of grasping unknown objects, one has to take care of obtaining the necessary information. As shown above, visual perception is *the* candidate in the related work to fulfill this requirement. As outlined in the introduction, I have

also chosen to follow the same path. But vision is not vision. There are a variety of different approaches, one of them being the active vision paradigm. In this section I will briefly describe this paradigm and the motivation that lead to its discovery. I will argue why it suits the needs of my given perceptual problem best. Finally, I will briefly point out some related work on active machine vision and active-vision-based robotic grasping that is relevant to my project.

2.2.1 Background

In his highly unfluential and famous book "Vision" David Marr has helped to build what would become the basis of machine vision research [34] for decades. It is worthwhile considering what he regarded as the purpose of vision:

> "Vision is a process that produces from images of the external world a description that is useful to the viewer and not cluttered with irrelevant information. [...] We have already seen that a process may be thought of as a mapping from one representation to another, and in the case of human vision, the initial representation is in no doubt-it consists of arrays of image intensity values as detected by the photo-receptors in the retina [34]."

According to him, the input of vision is a single or a series of intensity arrays, i.e. images, that have been generated by the eyes (or in the case of machines by cameras), which is used to produce a useful representation of the world. Just what the word "useful" means, highly depends on the task at hand:

> "The usefulness of a representation depends upon how well suited it is to the purpose for which it is used. A pigeon uses vision to help it navigate, fly, and seek out food. Many types of jumping spiders use vision to tell the difference between a potential meal and a potential mate [34]."

So, again, vision is not vision. Just what is the role that vision has to fulfill? In a later section, Marr goes on to say:

> "The second important thing, I thought, was that Elizabeth Warrington had put her finger on what was somehow the quintessential fact of human vision-that it tells about shape and space and spatial arrangement. Here lay a way to formulate its purpose-building a description of the shapes and positions of things from images [34]."

And that fits the needs of a robot that wants to grasp unknown objects pretty well: Vision delivers a description of the objects and their location in the external world using images.

This view of the vision problem guided years of computer vision research, until Aloimonos et al. proposed a paradigm shift in [35]. They theoretically investigated the question whether an active observer, i.e. an observer that moves with a known or unknown motion, has any advantage in comparison to an observer that does not. So basically, they added another input for the vision module -the motion of the camera- and a possible further output -the desired motion of the camera. The motivation for this scientific question came from the fact that years of research using the Marr-formalization of vision have proven that understand and emulating vision is a hard problem. Some basic problems turned out to be ill-defined, i.e. no unique solution existed. As a result, further assumptions about the physical nature of the external scene had to be introduced to make a given problem well-posed. This, however, was not an option for me in my project. The goal of this project was to develop a visual system that perceives object shapes while making as few assumptions about the objects as possible.

A second problem of the non-active vision scheme was that problems that were well-defined, often yielded unstable results. This means that small changes or errors in the input resulted in huge changes of the output - a highly undesired feature given noisy sensor data. Aloimonos et al. showed that deploying an active vision scheme let basic vision problems to become well-posed and stable, while at the same time it reduced the amount of constraints needed. Since I wanted to incorporate as few assumptions about the objects as possible, I chose to use an active vision framework for my visual system, hence *active* visual primitives.

2.2.2 Related Work

Aloimonis et al. already pointed out a further interesting research question, concerning whether or not there are camera motions that yield better perceptual results than others. Chaumette et. al. in [36] addressed exactly this problem in a structure-from-known-motion setting. Their approach to active vision is very similar to the one I deployed in this project: They use a single mobile camera that is mounted to the end-effector of a robot to actively explore 3D geometrical primitives and estimate their parameters. The authors also use visual servoing to keep the gaze of the cam-

era focused on the object, and include the current parameter estimations in the servoing. Theoretical analysis revealed that for the primitives under consideration (namely point, line, sphere, cylinder, and circle) optimal camera trajectories exist. Real-world experiments on the robot revealed improved estimation results and convergence properties of these trajectories in comparison to fixed trajectories that do not depend on the perceived object. There are, however, some differences to my project. First of all, the authors just considered perceiving perfect instantiations of the primitives, and are not looking for perception of general shape properties. As a result, they can use explicit surface descriptions of the objects in their visual servoing control laws. Thus removing generality in order to achieve more accurate results. Finally, the authors did not tailor their perception to any task, such as grasping, and therefore did not consider any further practical demands of this task.

Since the active vision paradigm has been introduced it has found ample applications in various fields of computer science as well as robotics. An extensive review of the research done on active vision in robotics can be found in [37]. One particularly interesting project that deploys active vision in a robotic grasping setting is presented in [38]. The authors designed a grasping algorithm for an eye-in-hand system that uses the curvature information of the silhouette of an object to find suitable grasping points. The robot actively explores the object on the basis of the current and past observations to find the maximum curvature and grasp the object around it. The main advantages of the proposed approach is that no prior knowledge about the objects that shall be grasped are needed, and that the curvature feature is able to generalize over a huge set of objects. Simplifications that may arise due to the compliance of the hand were not considered in the approach. Since a parallel-jaw gripper was deployed, discrimination of different preshapes of the hand and different shapes of the objects was not necessary. Evaluation in simulation and real-world experiments showed successful grasping results. Extensive experimental runs, however, were not reported.

In this chapter, I have given an overview of different kinds of grasping approaches. I have outlined that algorithms for grasping known objects require detailed object models that are difficult to obtain for unknown objects. Grasping of familiar objects requires a previously learned model to detect similar graspable features through classification. In order to grasp unknown objects, perception of the rough geometrical shape of such objects is a main research direction in the literature. I propose to fol-

low the same route, and will develop active visual primitives that extract common geometrical and spatial properties across a large set of objects. This information enables a compliant hand to achieve high grasping performance. In order to reduce the amount of assumptions about the objects under consideration I will employ the active vision scheme, thus increasing the capabilities of the primitives to generalize across different objects.

3 Methods

As described in the two preceding chapters, the aim of my thesis is to develop active perceptual primitives that reliably detect common geometrical properties across different objects which enable autonomous grasping of unknown objects. This chapter presents the detailed design and implementation of the primitives, and outlines their strengths and limitations.

3.1 Analysis of the Problem

At the beginning of the design of any system, it is necessary to determine the requirements that the system has to meet. In order to do that, one needs to first analyze and understand what the system shall tackle. Since I set out to develop a visual system that shall perceive geometrical object properties that correspond to properties of successful pregrasps of the robotic hand, the obvious question is: "What are the properties I am looking for?"

The answer to this question, as I have already mentioned in the introduction, depends on the hand that shall be deployed. Thus, it is necessary to analyze the capabilities of the hand which will be used for grasping. The robotic manipulator I used is equipped with a multi-fingered BarrettHand. The BarrettHand has three multi-joint fingers that are each controllable over one motor, and an additional DOF that allows lateral movement of two of the three fingers around the palm (The further details of the hardware and software setting of the project are discussed in section 3.2). Thus, the hand supports a huge variety of preshapes. As a result, in addition to the 6 DOF pose of the wrist one needs to further determine the 4 DOF preshape of the hand to completely specify a pregrasp.

I want to consider the pose of the wrist, first. An intuitively appealing approach that has proven to be useful in several different projects is to equal the desired end-effector point of the hand with the centroid of the object [33, 31, 27]. This simple heuristic may, of course, lead to problems in cases of extremely small or big objects. Making the approach less prone to errors of this kind can be considered in future work.

In order to decide the orientation of the wrist I will use a second heuristic that is inspired by human grasping approaches: Balasubramanian et al. performed a study in which humans were asked to move a robot arm and its hand to a desired pregrasp,

so that the robot could grasp the object just by closing the hand [27]. The authors analyzed the chosen pregrasps with regards to the presented objects, and found that humans tend to position the wrist so that the hand grasps an object around its main axis. This strategy led to very successful grasping across different kinds of objects [27]. Thus, in order to follow this strategy, the visual system has to be able to detect the main axis of objects.

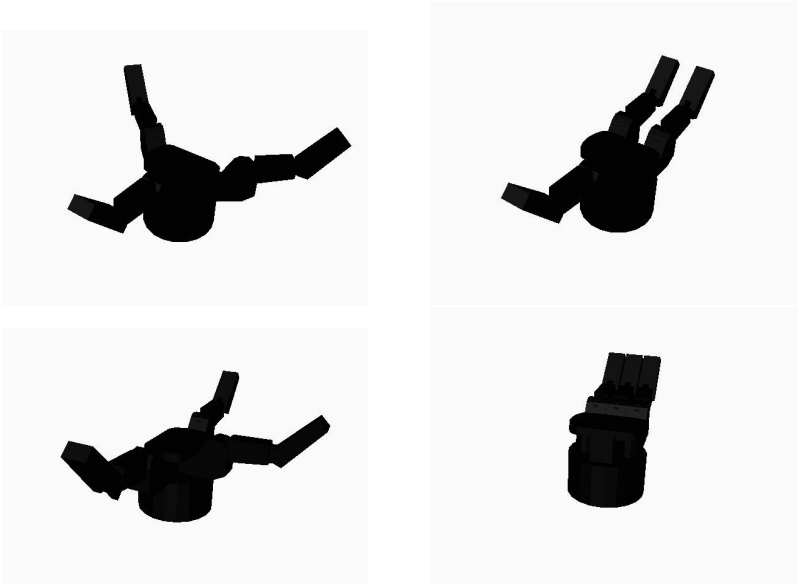


Figure 3.1: Possible meaningful preshapes of the BarrettHand as identified by Miller et al. in [21] (image taken from [21]). The upper two are the spherical and cylindrical preshape. Below are the precision-tip and hook preshape, which both appear to be rather special purpose preshapes.

With the information for deciding the wrist information already described, I want to consider the question of which information is needed to choose the preshape of the hand. Findings in neuropsychological research indicate that the shape of the object influences both the preshape of the hand and the approach strategy that is used [39, 40]. So, the grasping system ideally has access to clues about the general shape of an object when choosing the configuration of the hand before closing. The nature of those clues depends again on the preshapes that the hand can form. Miller et al. identified four distinct useful grasping preshapes that the BarrettHand can perform. They are on display in figure 3.1. The lower two, namely hook and precision-tip preshape, are rather peculiar forms that seem useful in special cases, such as very small and delicate objects (precision-tip) or huge objects or handles (hook). Thus, I chose to focus on the upper two preshapes: The spherical and cylindrical grasping preshape.

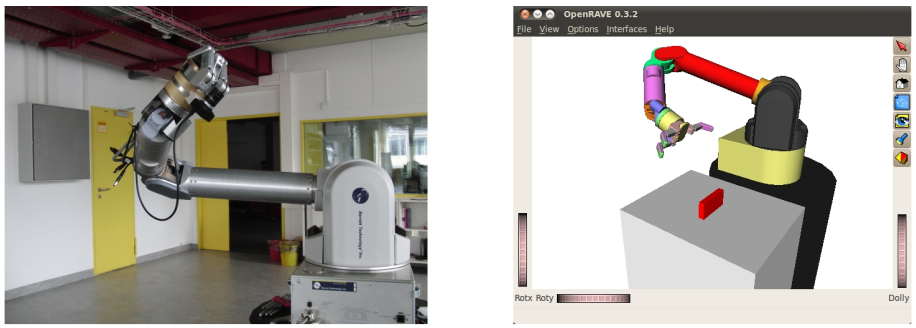
Based on these two preshapes I defined three distinct pregrasps. Firstly, there is the spherical pregrasp, which tries to grasp sphere-like objects with the spherical preshape. When using this pregrasp, the position of the end-effector will coincide with the centroid of the spherical object. Since both the spherical preshape and spherical objects do not have a main axis, the orientation of the wrist is not further specified. Secondly, there is the cylindrical grasp that employs the cylindrical preshape to grasp objects that are like a cylinder. Centroid of the object and end-effector point shall again coincide, and the main axis of the object shall be perpendicular to the palm of the hand. This still allows for various orientations of the hand, as long as it approaches the cylinder from its longitudinal. Finally, there is the box pregrasp that also uses the cylindrical preshape to grasp objects that are box-like. End-effector point and centroid of object shall again coincide, the object shall be grasped around its main axis, too. A further constraint is added, though, as the palm should be parallel to the face that connects the two opposing faces that each shall be contacted by fingers. Thus, the orientation of the wrist is further constrained to only four rough approach directions.

As a result, I have now described the perceptual demands that the primitives shall meet. I need to know the centroid of an object, its main axis, how sphere-like, how cylinder-like, and how box-like it is. For each of these properties I will present a separate active perceptual primitive that is designed to reliably perceive this information. Note that all of these results do not need to be perfect percepts of the real-world properties. They shall just afford successful grasping with a compliant mechanism, and need to be evaluated under exactly these premises.

I will now go on to describe the software and hardware setting that I used to implement my system and run experiments. Subsequently, the design and implementation of visual servoing controllers that are needed as auxiliary modules of the primitives will be presented before I describe the design of the primitives themselves.

3.2 Robotic Platform and Software Interface

For this project I used a WAM 7-DOF Arm equipped with a BarrettHand as the main robot platform for development and evaluation of the primitive-based visual system. Both, the hand and the arm are displayed in figure 3.2. The robot and the hand are both controlled via a joint-space-control interface, which is based on the provided drivers from Barrett and uses the ROS operating system[6].



a) b)

Figure 3.2: a) The eye-in-hand setup: Barrett Hand equipped with a camera around its wrist. b) The robot for experiments: WAM 7-DOF Arm with attached Barrett Hand, shown in the OpenRAVE simulator.

The usage of the joint-space-control interface motivates the deployment of an inverse kinematics algorithm to relate desired operational space poses to corresponding joint angles. For this functionality, as well as collision detection, the OpenRAVE software package[7] is deployed. OpenRAVE also provides a convenient simulation environment for software development and initial prototype testing. For communication between the visual system and the OpenRAVE simulator ROS is used.

In order to obtain the actual image I use a PointGrey Chameleon Cam that is attached around the wrist of the hand, thus, choosing an eye-in-hand system design. Figure 3.2 also depicts the setting of the camera on the wrist.

As already mentioned, the hand that is deployed is the BarrettHand that comes with 3 multi-joint fingers. Each of the fingers has an "inner link" that is actuated by its own motor, and a second "outer" link that is coupled to its inner one, thus, yielding 6 DOFs, with 3 control elements. One of the three fingers is fixed, while the other two can rotate around the palm. This motion capability is called "the spread" and adds another 2 DOF that are controlled via one interface value, i.e. each movable finger always has the same circular distance to the fixed finger. For closing the fingers, the BarrettHand provides a "clutch" mechanism that allows the outer link of each finger to keep on moving after the inner link has already hit an object and can no longer move. This mechanism facilitates compliant hand control and enables better grips to be established by simple closing of the hand.

3.3 Image Processing and Image Feature Extraction

The first step of visual perception is concerned with image processing and feature extraction from the image. This is a very important and crucial step, as it basically limits which sort of information can be extracted from a single image, and thus constitutes the basis of estimation. Additionally, if the feature extraction step is not robust to image noise, unexpected object shapes, or changing scene lighting the entire strategy will exhibit the same deficiency.

I decided to use the blob of the object under consideration as image feature. The blob extraction method that I employ does not rely on any assumptions about the shape of the item. So, by using this feature I have directly gained robustness against a variety of changes of the size and shape of the object. The only problem that this poses is segmentation of the item from the rest of the image. In order to facilitate this step, I assume that there is just one object in front of the robot and that it is very colorful. This is a pretty strong assumptions considering real-world settings, but I did not want to focus on this segmentation step too much. Additionally, powerful algorithms are available that can help overcome this restriction [41].

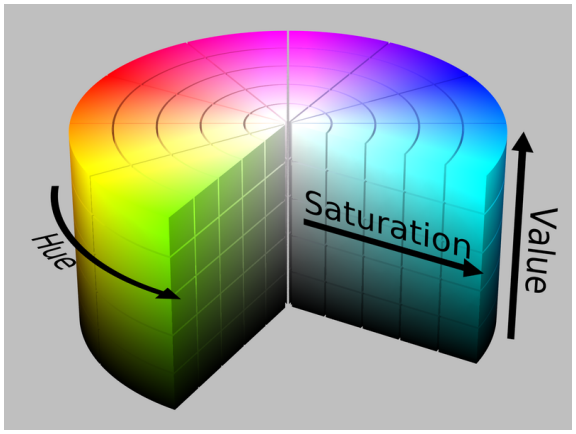


Figure 3.3: Topology of the HSV color space. High values for the saturation channel clearly hint at a colorful object[8].

For image processing I use the OpenCV software library that provides implementations for all the basic processing steps that I need[9]. First, the input image is converted to the hue-saturation-value (HSV) representation. This is a cylindrical coordinate representation of the RGB-color model that is very well suited for the task at hand, because high values in the saturation channel hint at a very colorful object. Figure 3.3 depicts the topology of the HSV-color space. Binary thresholding of the saturation channel yields a binary image which contains connected compo-

nents that correspond to colorful image regions. Labeling and blob extraction for these components is then done using the cvBlob library[10].

From these blobs a variety of simple properties can be extracted for later usage as features. I chose to obtain the area, centroid, and orientation of the blob. Image moments are a compact and convenient description of these properties that are often used in computer vision. The raw image moment for a 2D continuous function is defined as:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy. \tag{3.1}$$

For a 2D image the above integral becomes a sum, and for binary images M_{00} directly gives the area of the blob. The centroid is also easy to obtain. It is given as $\mathbf{x} = [\bar{x}, \bar{y}]^T = [\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}]^T$. Central image moments are a slightly different means of describing the pixels in an image. They can be used to obtain the orientation of a blob in the image. The continuous 2D formula is given by:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y) dx dy. \tag{3.2}$$

Using these moments one can build a covariance matrix for the distribution of pixels. The eigenvectors of this matrix correspond to the major and minor axes of the blob. The angle between the major axis of the blob and the image then equals to the orientation of the blob in the image. The angle can be calculated as

$$\theta = \frac{1}{2} arctan(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}) \quad \text{with} \quad \mu'_{20} = \frac{\mu_{20}}{\mu_{00}}, \quad \mu'_{02} = \frac{\mu_{02}}{\mu_{00}}, \quad \text{and} \quad \mu'_{11} = \frac{\mu_{11}}{\mu_{00}}. \tag{3.3}$$

Information, definitions, and formulas about image moments have been taken from [11], [12], and [42].

Now that I have decided on a set of visual features, I can already discuss their implications on the entire system. I have already pointed out that they include no assumptions beyond the fact that the objects shall be colorful, and that they show a connected blob from various angles. Thus, most real-life items are covered, and robustness against different object shapes has been incorporated.

However, I also need to analyze the features with regards to what they do not give me - direct 3D information. In fact, as Aloimonos et al. have argued in [35], without

further assumptions about the surface of the object under observation, extracting 3D information is an ill-defined problem. This fact introduces the need to move the camera. Through movement of the camera new measurement signals with different –possibly 3D– information content are generated. The main message here is that through moving the camera, one can extract 3D information with greater ease, i.e. simpler image processing, and more robustly, i.e. results do not stem from nonlinear equations that change rapidly even with small image noise [35]. But because these motions represent a new source of error, I have obtained robustness to different object shape, while still being able to extract 3D information, for the cost of having to deal with movement uncertainties.

Additionally, in order to keep the visual system rather simple I also had to assume that there is only one object in the scene and that the scene is not "distracting", so to speak, the blob detection with a flashy background or table covering. Although this assumption is not met by unstructured everyday environments clearly do not meet, the overall goal has been fulfilled, i.e. to choose image features that enable extracting of general object properties across a huge variety of objects.

3.4 Visual Servoing for Centering and Aligning of a Blob

A way to deal with uncertainties such as motion uncertainty is to incorporate further feedback through an additional sensor. Visual servoing is concerned with control of the motion of an end-effector using image data. Since I already have a camera in the experimental setting, I can use it to deploy the stabilizing effects of visual servoing in my design. In the following section, I will describe the basics of visual servoing and how I use them to provide my system with simple centering and aligning controllers.

In order to do practical computer vision one needs to have a model that describes the physical imaging processing that takes place in the camera. A simple and often-used model for this is the pinhole model. The real-world pinhole camera is one of the simplest devices that can be used to produce an image. It consists of a small box, the so-called camera obscura, with a small hole on one side and a light-sensitive material on the opposite inside of the box. The light enters through the pinhole and is then projected on the photo-reactive material. Figure 3.4 depicts the basic geometrical setting of a pinhole camera. Equation 3.4 relates size of the original object, image size, and distance between camera and object through one factor: the

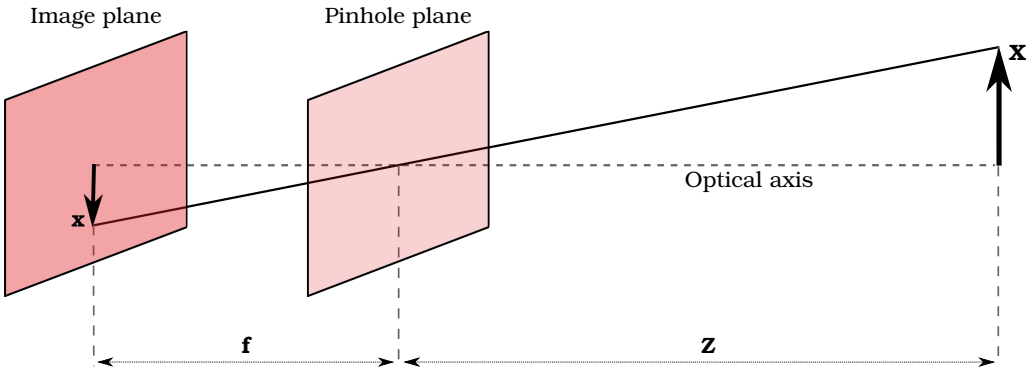


Figure 3.4: The simplified model of the pinhole camera: Only those rays that pass through the small pinhole get projected onto the image plane. This depiction has been inspired by a similar figure in [42].

focal length of the camera.

$$-x = f\frac{X}{Z} \tag{3.4}$$

It is possible to obtain an equivalent but easier-to-use model by swapping the position of the pinhole with that of the image plane. As can be seen in equation 3.5, the equation describing the projection almost stays the same. This new model for projection in the 3-dimensional case is on display in figure 3.5. I have taken this information about the pinhole model and projective geometry from the corresponding introductory chapter about camera models from [42].

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \tag{3.5}$$

Now that I have a model for imaging, I can use this input to control the motion of the arm. There are different settings in which visual servoing techniques can be applied successfully - each with its own advantages and disadvantages. Distinctions are usually made for different positioning of the camera with respect to the end-effector, the space in which the desired features for servoing are specified, and how tightly the visual feedback is coupled with the actual servo loop.

First of all, one can discriminate between an eye-in-hand and an eye-to-hand setting. In the former the camera is attached in or close to the end-effector of the robot and is moved with it. In the latter the camera is looking at the manipulator which

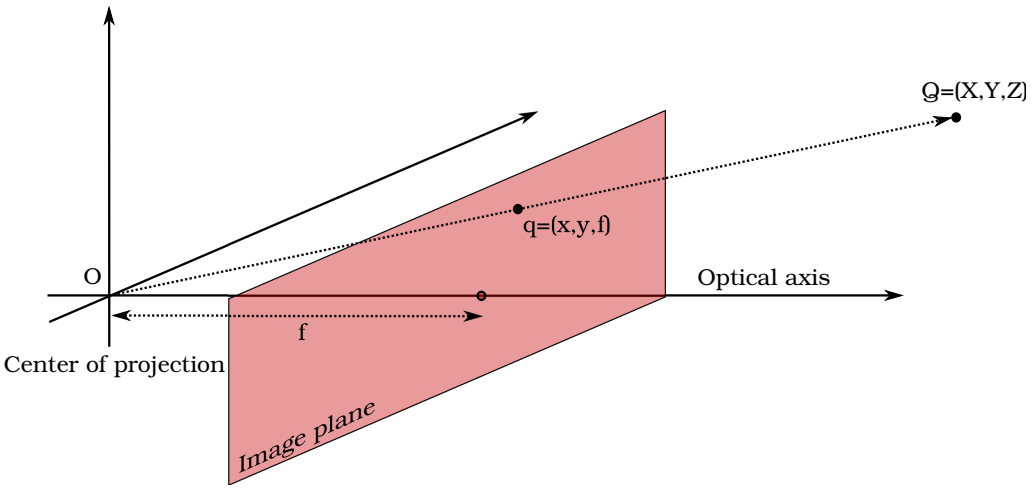


Figure 3.5: Projective geometry with the slightly changed pinhole model: The 3D-point $Q = (X, Y, Z)$ is projected onto the image plane by a ray that passes through the center of projection that is now behind the image plane. The resulting image point is $q = (x, y, f)$. This figure has been inspired by a similar one in [42], too.

it controls. Since I want to explore the scene with a moving camera, the eye-in-hand setting is the more appropriate one.

The next distinction can be made between image-based and position-based visual servoing. For position-based visual servoing one estimates the 6D pose of an object based on the current image, and then uses this estimation for servoing. Obviously, for such an estimation several model assumptions have to be made which introduce uncertainty. Since I do not want to introduce further sources of perturbation at this early stage of processing position-based visual servoing is not an option. I therefore use the image-based approach, in which the desired state of the system is specified in form of desired image features. A delta based on the current observation can then be computed in image space and translated into a desired motion of the camera for servoing.

Finally, one has to decide whether to incorporate the visual feedback directly into the high-frequency servo-loop. This approach basically replaces the inner control loop of the servoing with the visual servoing loop. Thus, the visual servoing is now responsible for stabilizing the end-effector. For that the dynamics of the robot have to be taken into account by the visual servoing, which is not trivial [43]. Additionally, the image processing also has to be fast enough to actually realize this high-frequency feedback. For all these reasons I chose to incorporated a dynamic look-and-move schema. In this setup the high-frequency servo loop stays untouched, and the low-frequency visual servoing loop is stacked on top of it, updating the desired pose

of the end-effector at a lower rate. This approach has the advantage to be easy to implement, but it allows positioning errors to accumulate longer and to become bigger before the visual servoing can intervene. Thus, one has to make sure that motions that occur during one cycle of the visual servo loop are small enough for it to be countered effectively.

Now that the overall setup is clear, I want to describe the visual servoing technique that I use more formally. Since I want to use image-based visual servoing, one has to first define the desired image features $\mathbf{s}_{des}$ that the controller has to maintain. Based on these one can define an error in image space:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}_{des} \tag{3.6}$$

This error vector in image space has to be related to the motion of the camera. This can be done using the *Image Jacobian* (sometimes also *interaction matrix*) $\mathbf{J}_I$:

$$\dot{\mathbf{s}} = \mathbf{J}_I \mathbf{v}_{cam}. \tag{3.7}$$

Assuming that the desired image features stay fixed, $\dot{\mathbf{e}} = \dot{\mathbf{s}}$ holds. And if the controller is to make sure that the error decreases exponentially, i.e. $\dot{\mathbf{e}} = -\lambda \mathbf{e}$, the following expression for the desired motion of the camera can be obtained:

$$\mathbf{v}_{cam} = -\lambda \mathbf{J}_I^{-1} \mathbf{e}. \tag{3.8}$$

$$\prescript{world}{EE\_des}{\mathbf{T}} = \prescript{world}{cam}{\mathbf{T}} \; \prescript{cam}{cam\_des}{\mathbf{T}} \; \prescript{EE}{cam}{\mathbf{T}}^{-1} \tag{3.9}$$

Using $\mathbf{v}_{cam}$ one can construct the homogeneous transform that represents the new desired pose of the camera $\prescript{}{cam\_des}{}\prescript{cam}{}{\mathbf{T}}$, with respect to the camera frame. Using the current pose of the camera in world frame $\prescript{world}{cam}{\mathbf{T}}$, the transform between camera and end-effector frame $\prescript{EE}{cam}{\mathbf{T}}$, and the knowledge that $\prescript{EE}{cam}{\mathbf{T}} = \prescript{EE\_des}{cam\_des}{\mathbf{T}}$ one can calculate the new desired pose of the end-effector with respect to the world frame; see equation 3.9. Figure 3.6 depicts the assigned reference frames and homogeneous transformations between them.

Of course, $\mathbf{J}_I^{-1}$ can only be computed if $\mathbf{J}_I$ is square. Otherwise, stacking of image

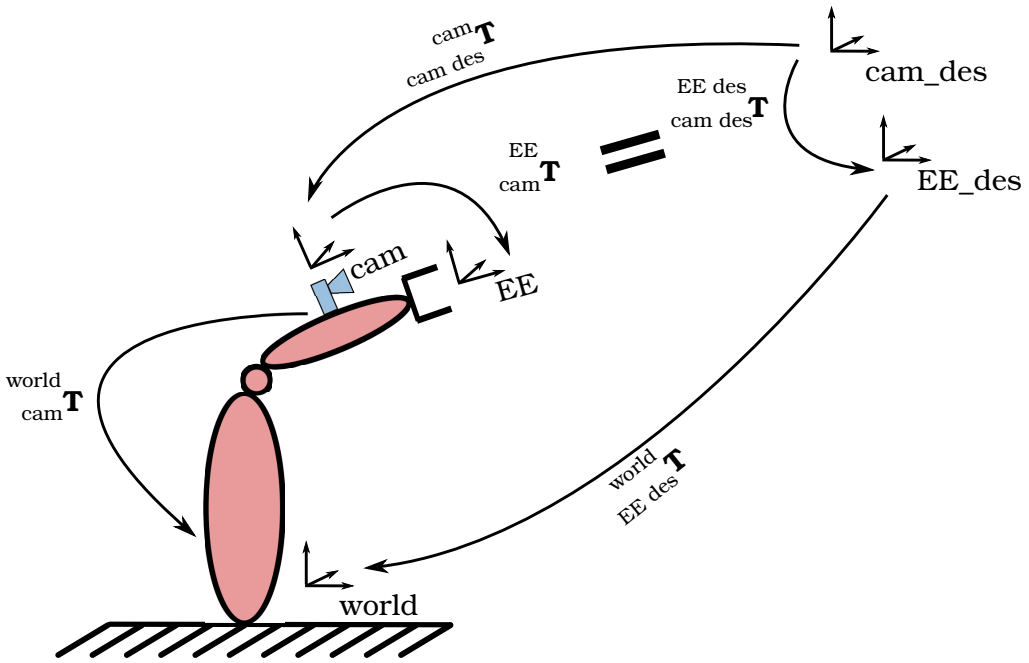


Figure 3.6: Definition of the major reference frames of the robotic and camera setting, and the resulting homogeneous transformations.

features to make the Image Jacobian square, or usage of the pseudo-inverse $\mathbf{J}_I^+$ as an approximation can help circumventing this problem. At this point of the discussion it is interesting to introduce the structure of the $\mathbf{J}_I$ for a point feature:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{f}{z} & 0 & -\frac{u}{z} & -\frac{uv}{f} & \frac{f^2+u^2}{f} & -v \\ 0 & \frac{f}{z} & \frac{-v}{z} & \frac{f^2+v^2}{f} & \frac{uv}{f} & u \end{bmatrix}}_{=\mathbf{J}_I} \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3.10}$$

I can now use the above knowledge to design simple visual servo controllers that my active vision system can use to perform perception. A first such controller is a *center controller* that centers the centroid of the detected biggest blob in the center of the image. There are three straight forward ways of doing this with the above Image Jacobian:

1. Use the linear displacements T_x and T_y of the camera to center the blob. Calculation of the resulting Image Jacobian will require knowledge of the distance

between camera and object, the image features, and the focal length of the camera.

2. Use the angular velocities ω_x and ω_y of the camera to center. The corresponding Image Jacobian will just require the focal length of the camera and the image features.

3. Perform both in parallel. This will obviously have the same requirements as option 1.

With my given mono-camera setting it is quite a strong requirement to have depth information for this very basic operation of centering. It should be my goal to design this controller in a way that it solely relies on no model representation of the world at all, and just uses the given sensor information. Therefore, option number 2 is the obvious choice. The resulting Image Jacobian is square, and thus invertible if it is not singular:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{uv}{f} & \frac{f^2+u^2}{f} \\ \frac{f^2+v^2}{f} & \frac{uv}{f} \end{bmatrix}}_{=\mathbf{J}_{I\_center}} \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} \tag{3.11}$$

For aligning the blob within the image with an *alignment controller*, the case is even more straightforward: The difference between the extracted image feature θ and the specified desired orientation of the blob θ_{des} yield the error in orientation e_θ which can be directly applied as a desired rotation about the z-axis of the camera frame to determine ω_z. As a result, the alignment operator also does not rely on any information that goes beyond its current sensor information, and does not make any further assumption about the object than what feature extraction necessitates. Figure 3.7 depicts the centroid and angle feature for centering and alignment of one blob. Both exhibit a nice convergent behavior without any overshooting or steady-state-error.

All of the above theory and formulas about visual servoing basics have been taken from Chaumette's tutorial [44] and from Oliver Brock's lecture slides of the introductory robotics course "Robotics" which was taught in the winter term of 2010 [13].

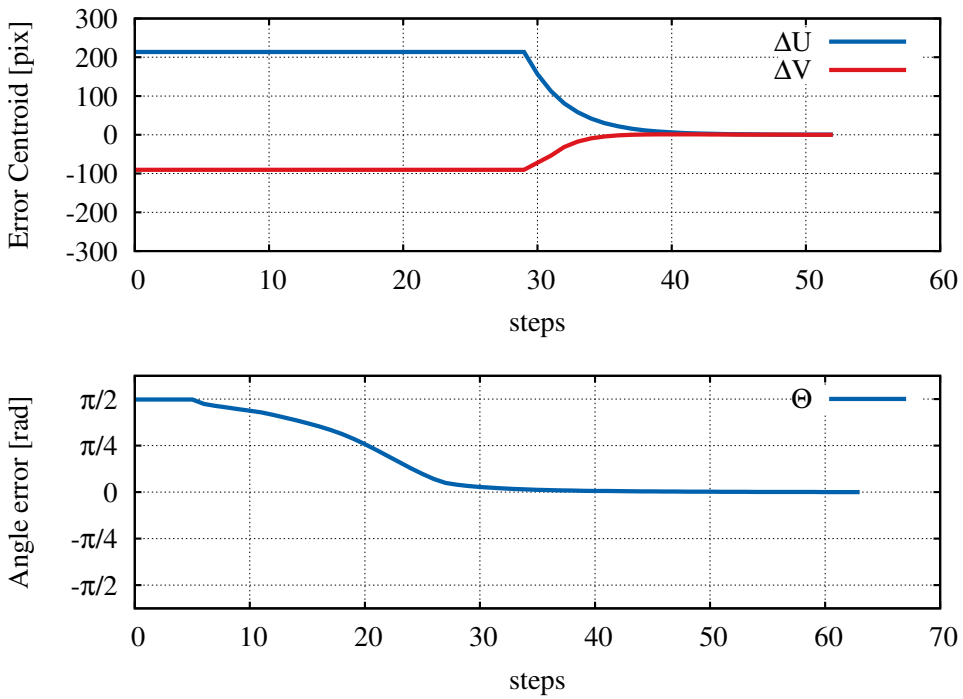


Figure 3.7: Center and alignment controllers at work: The upper plot shows the delta between centroid of blob and center of image while centering. The lower blob depicts the angle of the blob while aligning it with the image.

3.5 Distance Primitive

In this section I will use the just described center controller to build a perceptual primitive that is able to estimate the distance of the camera to an object under observation.

The basic idea for the distance primitive is the following: If one moves on a circular trajectory around an object, with the object in the middle of the trajectory, the image of the object will stay in the center of the image. During such a motion, the radius of the trajectory will give a rough estimate of the distance between object and camera. Additionally, the center of motion can serve as a rough approximation of the centroid of the object. Figures 3.8 and 3.9 depict such a motion for two sample settings.

To achieve such a motion robustly and quickly, I use an easily detectable feedback signal that arise during circular movement around an object: Does the image stay in the center of the picture? If no, the radius of the current motion is wrong and has to be adapted.

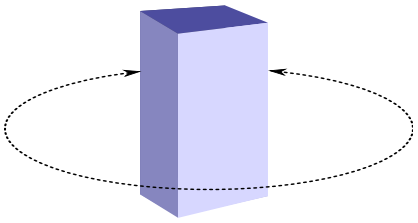


Figure 3.8: Desired circular motion around an object.

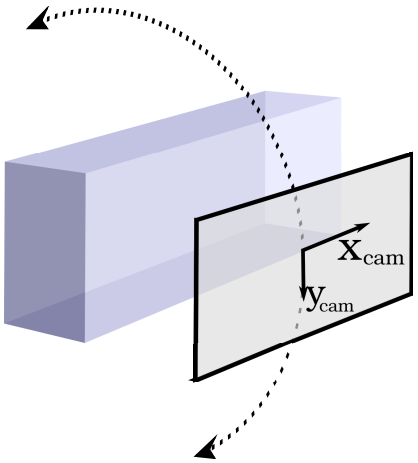


Figure 3.9: Definition of the right-handed image coordinate system during exploration.

Figure 3.10 visualizes the idea behind the distance primitive: Assume an initially centered blob. If one now performs a small part of the circular exploration trajectory towards the upper end of the image, i.e. towards $-y_{cam}$ (compare Figure 3.9), using any initial non-zero radius, only two error cases are conceivable. In the first one the assumed radius was too small, thus one has to increase the assumed radius. In the second case the used radius estimation was too big and needs to be decreased. After one has improved the radius estimation, the center controller re-centers the blob and another step of exploration can be done. After several exploration and correction steps the radius estimation will be correct and the blob will not leave the center of the image during further exploration.

Algorithm 3.1 shows the pseudo code description of the distance primitive. There are a couple of small tweaks in the algorithm that I have so far not described: The variable $countErrorfree$ serves as a memory for how many exploration steps a correction was not necessary. Thus, it provides a measure of how good the estimation is, and can be used to influences the magnitude of an update of the estimation. In case of an uncentered blob, $countErrorfree$ will not be reset to 0. This helps to reduce the magnitude of Δr (compare algorithm 3.2) in later stages of the exploration, i.e. yields a stabler convergence behavior. A second counter-variable *countSteps* is used to count the exploration steps of the primitive and stop estimation after a fixed amount of exploration steps (called $threshExplore$).

Finally, there is the *explore* functionality which uses the current radius estimation to cause a small exploration motion along the assumed circular trajectory. To con-

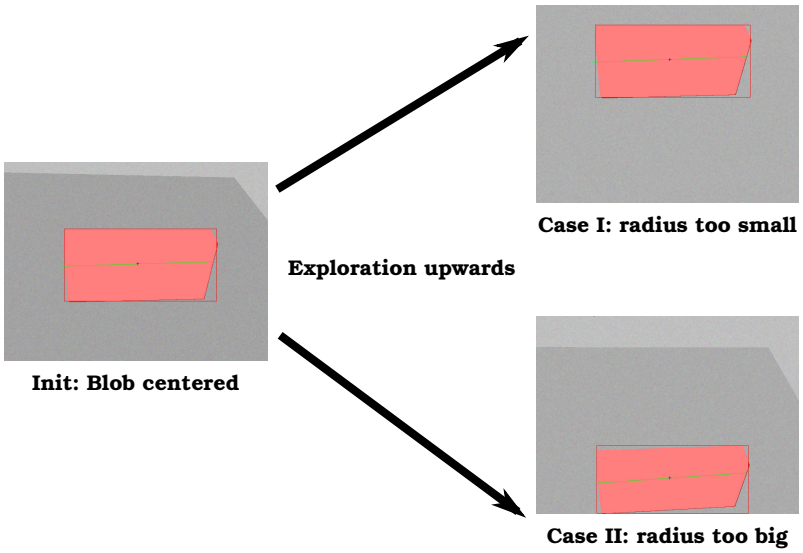


Figure 3.10: Detectable error cases during exploration: Too small radius estimations yield moving of the blob to the top of the image. Too big estimations will cause the blob to move down.

veniently describe this desired motion, I used an intermediate spherical coordinate system, that was centered at the desired center of motion. Transforming this desired relative camera motion into desired movements of the end-effector is analogous to 3.9; one just needs to add another transformation from the spherical coordinate system to the local camera frame. Figure 3.11 depicts the chosen definition of spherical coordinates, and the correspondences to Cartesian coordinates is given by:

$$\begin{bmatrix} x_{intermediate} \\ y_{intermediate} \\ z_{intermediate} \end{bmatrix} = \begin{bmatrix} r \cdot cos\Theta \cdot cos\phi \\ r \cdot cos\Theta \cdot sin\phi \\ r \cdot sin\Theta \end{bmatrix} \tag{3.12}$$

With the help of these formulas such a fraction of a circular motion can be easily described as, e.g $[r, \Theta, \phi] = [r_{estimate}, \Theta_{step}, 0]$.

To conclude the presentation of the distance primitive, the actual update rule is depicted in 3.2. The most interesting part for the discussion is the calculation of the update Δr. As already pointed out, the higher the absolute value of $countErrorfree$, i.e. the longer the exploration periods without decentering, the smaller the error in estimation seems to have been. The actual error of the blob center in image space linearly influences Δr. This is reminiscent of a typical proportional controller. The final factor that can be used to influence the convergence

Algorithm 3.1 Basic distance estimation

Require: $blob \neq NULL \wedge image \neq NULL \wedge threshCenter > 0 \wedge threshExplore > 0$
  $countErrorfree \leftarrow 0$
  $countSteps \leftarrow 0$
  $radius \leftarrow 0.01$
  $direction \leftarrow up$
  while $countSteps < threshExplore$ **do**
    $error \leftarrow blob.center - image.center$
    if $|error| \geq threshCenter$ **then**
      $radius \leftarrow updateRad(radius, countErrorfree, error, direction)$
      $countErrorfree \leftarrow countErrorfree/2$
      while $|error| \geq threshCenter$ **do**
        $center(blob)$
      end while
    else
      $direction \leftarrow explore(radius, direction)$
      $countErrorfree \leftarrow countErrorfree + 1$
      $countSteps \leftarrow countSteps + 1$
    end if
  end while

properties of the estimation process is *gain*. Just like the gain of a P-controller it can lead to overshooting behavior or slow convergence (see figure 3.12).

Now that the algorithmic idea behind the distance primitive has been introduced, it is worthwhile to contemplate how this primitive extracts depth estimates even though the employed visual feature (blob center) does not directly yield this sort of information. The clue is that the change of the blob center while performing a certain motion provides the information necessary for the depth estimation. Thus, the proposed algorithm follows the idea of structure from motion, that is: it extracts information from observing image features over time.

One key aspect – that is not found in every typical structure from motion algorithm – is that the primitive exploits the fact that the camera motion can be determined by the needs of the perception. This fact can be used to perform a motion that facilitates perception, i.e. the circular trajectory. Additionally, there is a tight coupling between motion and perception: The result of perception, i.e. the distance estimation, is fed into the desired trajectory, which in turn influences subsequent perception. The only example of related work that works on equal premises is the project by Chaumette et al. [36]. They, however, obtain depth information by assuming perfect instances of geometric shapes such as circles, cylinders, or spheres

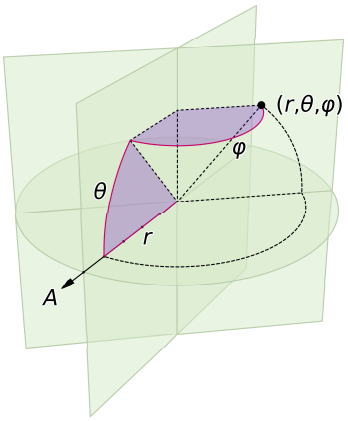


Figure 3.11: Spherical Coordinates to describe the circular exploration trajectory (taken from [14]).

–an assumption that my algorithm does not make.

With regards to robustness, there are several aspects of the primitive that lead to reliable and stable results. First of all, neither the basic algorithm nor the update rule make any assumption about the kind of object that is under observation. In fact, just the center property of the blob is used. This makes the algorithm applicable for a wide range of objects, and thus robust to object changes. As a result, the distance primitive is able to estimate the distance to an arbitrary object through an actively-controlled motion.

A second interesting property is that the algorithm bases its distance estimation just on the decision whether or not the blob got uncentered. This computation is fairly simple and does not involve any knowledge about the object, either. This ease of computation is due to the fact that the primitive itself ensures that not any trajectory around the object is executed, but the one which reveals most information. To further elaborate this point, consider a fairly different approach: One could try to calculate the error in estimation with high accuracy in every time step with an inverse model. This model would take the pixel error and the exact desired camera trajectory, and assume a detailed object model, to eventually yield the correct pose of the object. Such an approach would be computationally quite demanding and require a multitude of assumptions to hold: Accurate object models are given, image noise is very low, and motion uncertainties are not present. In contrast, the distance primitive can cope without either of these assumptions. Additionally, the decision that the estimation needs to be updated can be made pretty robust by choosing a bigger *threshCenter* to cope with bigger measurement and motion perturbations.

Furthermore, the estimation is not based on a single measurement. Instead a

Algorithm 3.2 Update radius estimation

Require: $error \neq NULL$
  if $countErrorfree > 0$ **then**
    $\Delta r \leftarrow |gain * error.y/countErrorfree|$
    if $direction == up$ **then**
      if $error.y > 0$ **then**
        $radius \leftarrow radius - \Delta r$
      else
        $radius \leftarrow radius + \Delta r$
      end if
    else
      if $error.y > 0$ **then**
        $radius \leftarrow radius + \Delta r$
      else
        $radius \leftarrow radius - \Delta r$
      end if
    end if
  end if
  return $radius$

successful estimation is a result of an integration of a series of observations. Thus, positioning and image processing errors that are normally distributed have a chance of canceling each other out. Or in other words: Since this per design is not a one-shot policy, single errors are less likely to completely thwart perception.

Finally, through incorporation of constant visual feedback, i.e. recentering the blob after it got uncentered, the perceptual process is made robust to small positioning errors. Such errors can happen because the current estimation is faulty, or because the robot reaches a desired configuration only up to a certain accuracy. Either way, the constant recentering makes sure that the object of interest remains in the center of the image.

There are, however, some limitations of the proposed distance primitive. First of all, in case a desired motion is not executed correctly and as a result the blob will get uncentered, the estimation will be altered; even if the estimation was correct in the first place! That is because the estimation step does not include the current pose of the camera, thus assuming correct motion execution. It would be desirable to also correctly use the information obtained through a faulty motion into the estimation. Additionally, huge motion errors or exploration steps that take the object out of the camera image cannot be corrected by the visual servoing component. That is because I incorporate a dynamic look-and-move schema which basically waits for a motion to stop until new visual feedback is used to correct the pose of the camera.

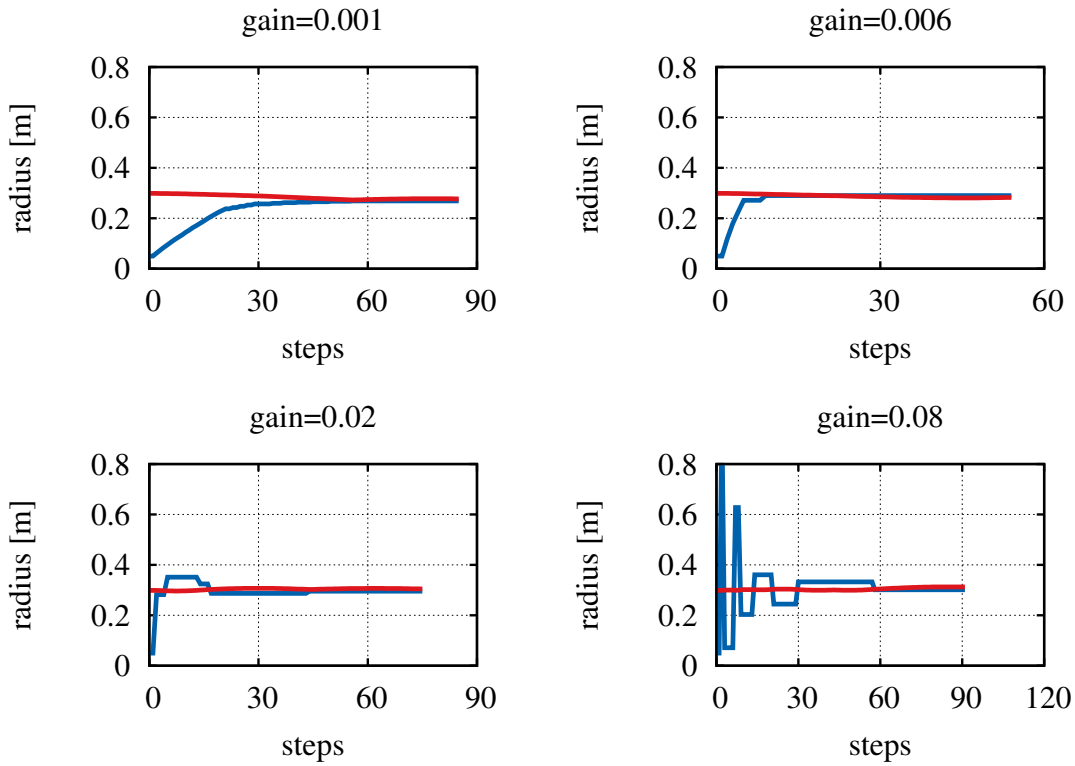


Figure 3.12: Simulation results of distance estimation with different gain values. The blue line shows the estimation at every time step, while the red one depicts the calculated ground truth at every point of the experiments. The upper left plot shows slow convergence, whereas the lower right illustrates extreme overshooting. For a value of $gain = 0.006$ the estimation behavior resembles that of critical damping.

Usage of visual feedback in the high-frequency control loop of the system or a higher frequency of the outer loop could remove or reduce this problem. Finally, the actual exploration trajectory that the camera follows depends completely on the object, its pose, and the initial pose of the camera. If a combination of those three yields an exploration trajectory that is not feasible, no alternative trajectory will be tried. Addition of further heuristics plus motion planning to cope with such problematic situations would be useful extensions of the system in future work.

In order to be able to easily use the distance primitive as a sub-module in a more elaborated vision system, it is desirable to also provide a measure of confidence on the convergence of estimation. For this purpose I have designed a heuristic function $conf(t)$ that provides for every time step t a value from the range $[0; 1]$, where 1 denotes the highest degree of certainty about convergence and 0 the exact opposite.

The idea behind $conf(t)$ is to log the last n distance estimations, and assess how great the variation of estimation was over the last n elements. This is done by the function $f(\Delta est)$. However, using such a buffer-based functionality, one has to think about using cases in which the buffer is not full yet. In such cases the variation might simply be small because not enough values have been seen. Therefore I also introduced a punishment factor $p(i)$ that is used to reduce the overall confidence if the buffer is not full. The overall confidence function is given as:

$$conf(t) = p(buffer.size) \cdot f(|buffer.max - buffer.min|), \tag{3.13}$$

where its parts have been chosen to be

$$p(i) = \begin{cases} i/n, & \text{if } i < n \\ 1.0, & \text{else} \end{cases} \qquad \text{and} \qquad f(\Delta est) = e^{-6\Delta est}. \tag{3.14}$$

Note that both functions yield values in the range of $[0; 1]$. Thus, their product also lies within the same range of values. Figure 3.13 depicts the curves of the two parts of the confidence function, while a sample plot of $conf(t)$ for an experimental run is on display in figure 3.14.

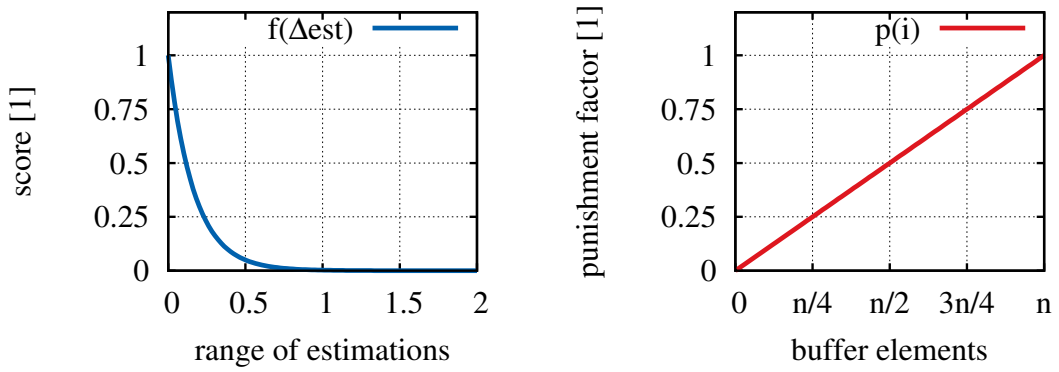


Figure 3.13: Left plot: Scoring function that evaluates how stable the estimation has been over all buffer entries. For small ranges the score quickly approaches a value of 1. Thus, it heavily penalizes ranges that are not small. Right plot: Punishment factor $p(i)$ that reduces the confidence value in situations with an not-full buffer. As soon as the buffer is full, i.e. $i = n$, there will be no punishment.

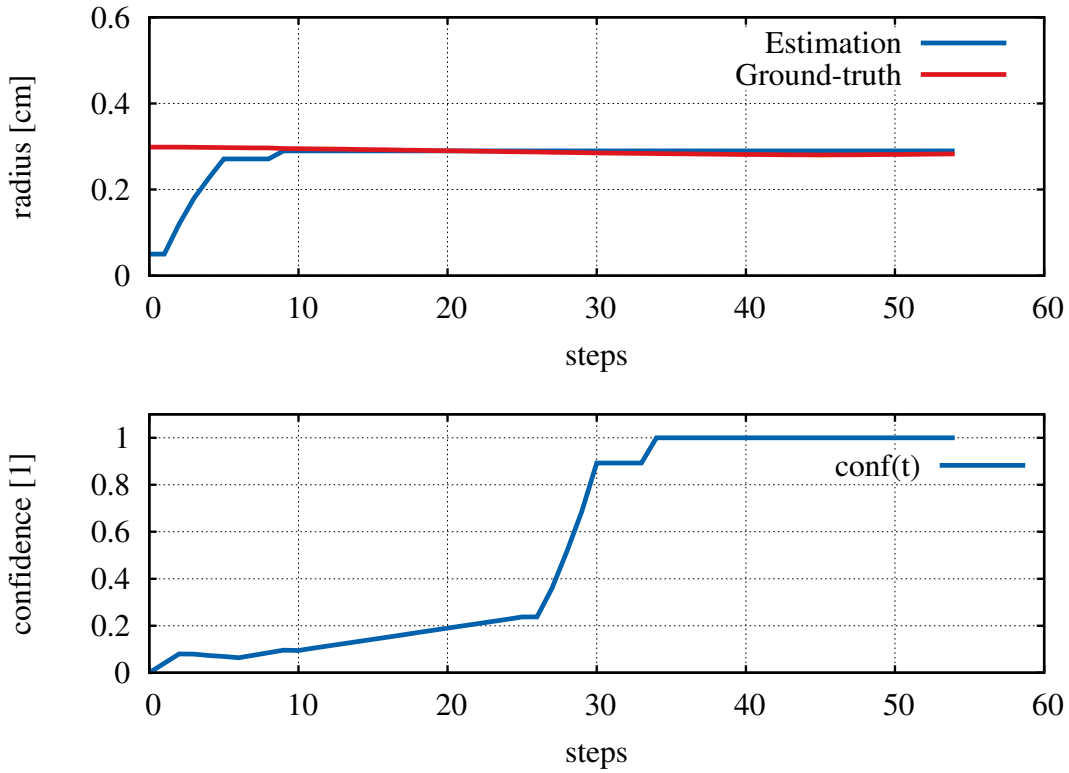


Figure 3.14: Upper plot: Shows distance estimation and calculated ground truth for an experimental run from simulation. The plot below shows the corresponding development of the confidence value over time. Not surprisingly, the confidence value mimics the estimation plot and stays close to 1.0 after enough unchanged estimation results have been accumulated.

3.6 Axis Primitive

The next primitive that I want to discuss is the Axis Primitive. While the Distance Primitive helped to roughly estimate the centroid of the object that is under consideration, the Axis Primitive estimates the main axis of a given object.

The ideas underlying Distance and Axis Primitive are very similar: If one moves a camera in a circular motion around the main axis of an object and the axis of rotation and the axis of the object are coinciding, then the main axis of the blob of this object will stay aligned in the camera image. So, again a circular motion will be performed by the camera, and again an unchanged property of the blob during that motion is the feedback signal that indicates successful estimation.

There is, however, one key difference between Distance and Axis Primitive, and that is the representation of the estimation. For the Distance Primitive there was an estimation variable – the radius – that was used for estimation. In the case of

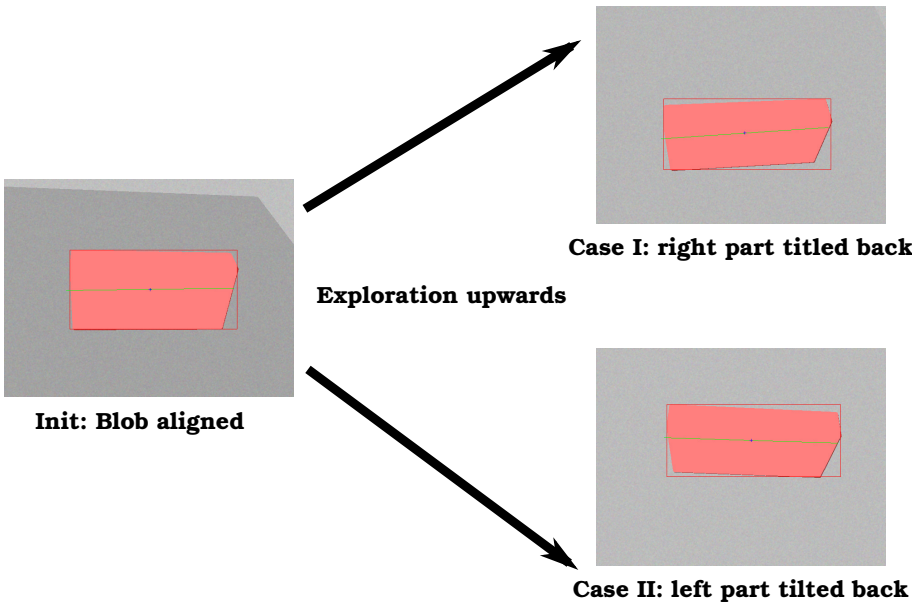


Figure 3.15: Feedback signal used by the Axis Primitive: Main axis of the blob during exploration. If one moves upwards in a circular motion, there are only two ways in which the blob can get unaligned. Either of them indicates that the estimation has to be changed, and also shows how this has to be done. For the depicted case, one has to move to the side where the object is tilted, i.e. for a blob with the right side pointing up one moves to the right to improve the estimation, otherwise to the left.

the Axis Primitive, the estimation is directly represented in the current orientation of the camera. Thus, there is no need for any state variable to hold the current estimation. An advantage of this design is its simplicity, as there is no need to further *re*present the orientation of the camera. A disadvantage is for example that possible motion errors during exploration can make the estimation worse and cause a loss of information.

The design of the Distance Primitive used an uncentered blob as a feedback signal. The Axis Primitive, however, is detecting unaligned blobs as a trigger that indicates that the estimation has to be changed. Consider an initially aligned blob, i.e. the angle between the main axis of the blob and x-axis of the image is 0. If one now performs a fraction of a circular exploration towards the top of the image, the blob can either stay aligned or get unaligned. In the former case the estimation seems to be correct, in the latter case the estimation has to be updated. In this case there are only two possible errors scenarios: Either the right or the left side of the blobs is pointing upwards in the image. Figure 3.15 visualizes both cases.

With the help of this error information, one can perform corrective motions to improve the axis estimation. As can be seen in Figure 3.15, the camera has to

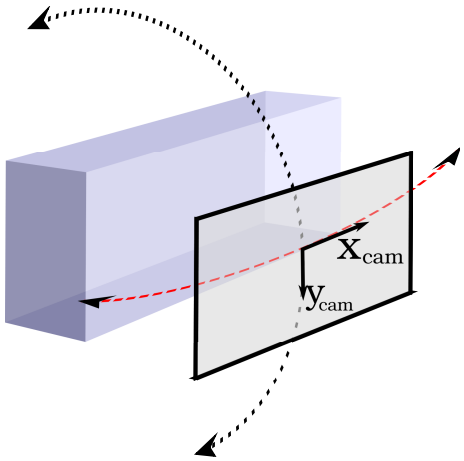


Figure 3.16: The exploration movement (black) and the corrective motion (red) of the camera are both circular motions that try to keep the object in the center of the image.

be moved to the side where the object is tilted backwards. Thus, there is a need for a small motion that is perpendicular to the exploration. One possibility to do this is to perform another circular motion – only this time the motion is side-wise. An advantage of this is that such a motion would try to keep the object in the center of the image. A disadvantage is that it needs an estimation of the distance to the object. I will, however, propose to perform such a motion and postpone the discussion of the disadvantages of this choice to a later part of this section. Figure 3.16 depicts the directions of possible circular corrective motions with regards to the exploration around the object.

In order to describe this motion, one can again use spherical coordinates as defined in 3.12. This time, however, the corrective motion is parametrized as $[r, \Theta, \phi] = [r_{estimate}, 0, \Delta\phi]$. After the corrective motion has been performed, the alignment controller can realign the blob in the image with changing the position of the camera, thus resetting the initial case of an aligned blob. One is again looking at the aligned projection of the main axis of the blob, only this time angle between projection and actually estimated axis is smaller. Repeated exploration and correction steps will yield a constantly better estimation.

The pseudo-code specification of the axis estimation algorithm is depicted in 3.3. It corresponds to the verbal description of the workings of the Axis Primitive. There are, however, some subtle details that I have not mentioned yet: The Axis Primitive is again counting the number of error-free steps to decide how big the corrective motion shall be. The intuition behind this is again that the more error-free steps one has done, the better is the estimation. The corresponding counter variable

Algorithm 3.3 Basic axis estimation

Require: $blob \neq NULL \wedge threshExplore > 0 \wedge threshAngle > 0$
  $countSteps \leftarrow 0$
  $countErrorfree \leftarrow 0$
  $direction \leftarrow up$
  while $countSteps < threshExplore$ **do**
    $[tiltAngle, ratioEV] \leftarrow getImageMoments(blob)$
    if $(|tiltAngle| > threshAngle) \wedge \sim (ratioEV \approx 1.0)$ **then**
      $correctiveMotion(angle, countErrorfree, radius, direction)$
      $counterErrorfree \leftarrow counterErrorfree/2$
      while $|tiltAngle| > threshAngle$ **do**
        $align(blob)$
      end while
    else
      $counterErrorfree \leftarrow counterErrorfree + 1$
      $counterSteps \leftarrow counterSteps + 1$
      $direction \leftarrow explore(direction)$
    end if
  end while

$counterErrorfree$ is again not reset to 0 in case of an unaligned blob. This is done to keep a certain "memory" of previous exploration steps.

In addition to the angle between main axis of blob, and x-axis of image ($tiltAngle$, which is calculated as described in section 3.3), the algorithm also calculates the ratio of the two eigenvalues of the blob (again as described in section 3.3). If this ratio is close to 1 then no obvious main axis of the blob exists. Consider for example the image of a cylinder that is seen from above. It is a circle. In this case both eigenvalues have almost equal values and because of the image noise the eigenvectors and eigenvalues vary for every frame. Trying to align with such a main blob will lead to endless chaotic alignment, until in some random case the extracted main blob axis is by chance aligned with the image. In order to cope with such a situation, the Axis Primitive ignores these cases and treats them as if the blob was aligned. Thus, the primitive will keep on exploring. In case of the cylinder perception further exploration steps will soon take the camera to a viewpoint from which it sees more of the side of the cylinder. Then a stable main axis is visible, realignment can be done, and the axis will eventually be estimated correctly.

A pseudo-code description of the corrective motion is given in algorithm 3.4. Again a gain parameter that influences speed of convergence is included in the design. Its behavior also mimics that of a p-gain in a p-control term: It can cause quicker or slower convergence behavior, but also overshooting. It is important, however, to

Algorithm 3.4 Corrective motion for axis estimation

```
if countErrorfree > 0 then
  Δφ ← |gain * angle/countErrorfree|
  if direction == up then
    if angle > 0 then
      correctRight(radius, Δφ)
    else
      correctLeft(radius, Δφ)
    end if
  else
    if angle > 0 then
      correctLeft(radius, Δφ)
    else
      correctRight(radius, Δφ)
    end if
  end if
end if
```

note that for axis estimation the main factor that leads to a quick convergence of the estimation is the constant aligning of the blob in between exploration steps, not the corrective motion. The plots that are depicted in Figure 3.17 visualize these points. They show the error of axis estimation in degrees from three different simulation experiments that each used three different gains for the corrective motion. During the first 20 steps of the experiment the align controller performs the initial alignment of the blob in the image. This already accounts for a big part of the existing orientation error. Up to this point, all three experiments yield the same plot. Then the exploration starts and as soon as the blob becomes unaligned for the first time, the different behaviours for the different gains become apparent. The blue line with $gain = 0$, i.e. no corrective motion is performed, just realigns, thus showing the smallest correction of the error. The green line with a high $gain$ of 50 performs a big corrective motion that yields an overshoot, i.e. the correction is too much and the estimation gets worse. This means that after the corrective motion the blob is tilting to the other side in the camera image. The plot with the medium $gain = 30$ does not show overshoots. Both runs that use the corrective motion do show quicker convergence than the run without it. However, just using constant realigning still successfully improves the axis estimation.

An important feature of the design of the Axis Primitive is that it allows the distance estimation to run in parallel because it also performs a circular motion around an object. This is exactly the desired exploration of the Distance Primitive,

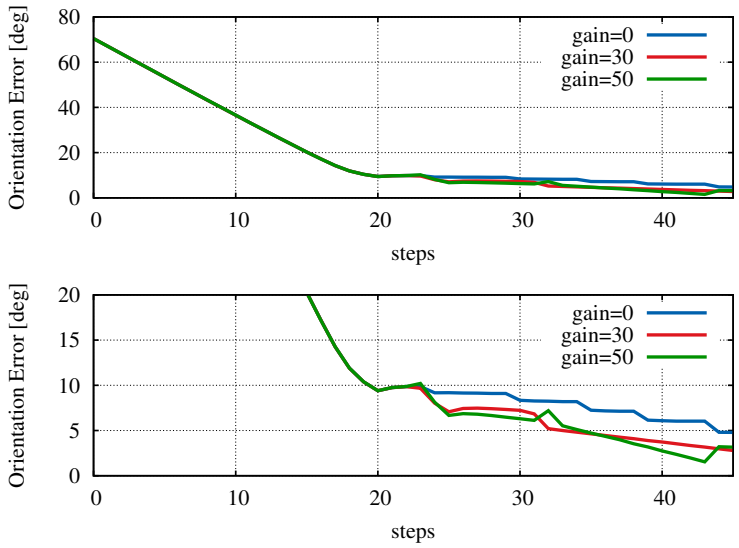


Figure 3.17: Influence of different gain parameters for corrective motion during axis estimation: The upper plot shows axis estimation with three different gain parameters. Initially, alignment controller aligns blob within image for the first time (steps until 20), which already gets estimation close to desired orientation. Then, exploration with corrective motion and repeated re-alignment refines estimation. The lower plot shows a zoom-in on the upper plot: Estimation without corrective motion (blue) has slowest conversion, while corrective motions with high gains (green) cause overshoots at 23, 32, and 44.

only this time the Axis Primitive adds a further requirement that the motion shall be perpendicular to the main axis of the object, but this does not hinder the distance estimation algorithm. Furthermore, in case the blob gets uncentered or unaligned during the exploration this is first corrected, then exploration –as desired by the Axis Primitive– is continued.

As I already pointed out, the corrective motion needs an estimation of the distance to work perfectly. In case the distance estimation is wrong and a big corrective motion is desired, the blob can get uncentered. But it can only get uncentered in a direction that is perpendicular to the exploration direction. Thus, this error is not interfering with the distance estimation, and a simple recentering motion can be done to correct this. The only real problem that can occur is the combination of an extremely big error in distance estimation, and a big desired corrective motion. Then, the object can get out of the image, thus effectively terminating the estimation. The solution that I have incorporated into my design for this case is to clamp the corrective motion parameter $\Delta\phi$ to limit the extend of this worst-case scenario.

To conclude the presentation of the axis estimation algorithm, I will shortly discuss the advantages and disadvantages of the design. As the designs of the Distance and

the Axis Primitive have a lot in common, they also share the same strength and weakness. Estimation of the main axis of an object is done without any assumptions about the shape of the object with a very simple algorithm that bases its update decisions on easy to detect property changes. On the other hand, motion errors that cause faulty perception can lead to wrong estimations. So far, there is no module that compensates for objects that get out of the camera image. Finally, exploration trajectories that are desired because of the initial pose of the object and not feasible result in aborted estimations.

In order to assess the convergence of the axis estimation, I also designed a confidence measure for the Axis Primitive. It can be interpreted as "the confidence that the current x-axis of the camera frame corresponds to the main axis of the object". It shall yield a value within the range of $[0; 1]$, where a small value can mean two things: The estimation has either not yet converged, or there is no main axis for this object. In case of a high confidence measure, there are no two interpretations – the main axis has been found.

The idea is again to remember the last N estimations and score their maximum range with a heuristic function. Thus, an estimation is considered to have converged if the estimation has not or hardly changed during the last N exploration steps. Since a buffer is employed situations with a non-full buffer have to be taken into account. Therefore, I propose to use the same punishment term $p(i)$ that was used in the Distance Primitive. Additionally, one has to incorporate the case of an object that does not have a main axis in the scoring function. To achieve that, I add another scoring function which evaluates the ratio of the eigenvalues of a blob to assess the roundness of the blob. If the blob is too round, i.e. eigenvalue ratio close to 1, then the camera is obviously not perpendicular to an axis and the confidence value should be zero.

The entire scoring function is given in equation 3.15. $f_1(EV_1/EV_2)$ evaluates the roundness of the current blob, while $f_2(max\Delta)$ scores the maximum range between the last N estimations.

$$conf(t) = p(buffer.size) \cdot f_1(EV_1/EV_2) \cdot f_2(\max_i \parallel buffer.last - buffer(i) \parallel) \tag{3.15}$$

Note that $buffer$ holds the last N x-axes of the camera frame. Thus, $\max_i \parallel$

$buffer.last - buffer(i) \parallel$ is looking for the biggest euclidean distance between the current estimation and any of the last N buffered estimations. $f_1(\frac{EV_1}{EV_2})$ and $f_2(max\Delta)$ are given as:

$$f_1(\frac{EV_1}{EV_2}) = tanh(2 \cdot |\frac{EV_1}{EV_2} - 1|) \quad \text{and} \quad f_2(max\Delta) = e^{-|3.0 \cdot max\Delta|}. \tag{3.16}$$

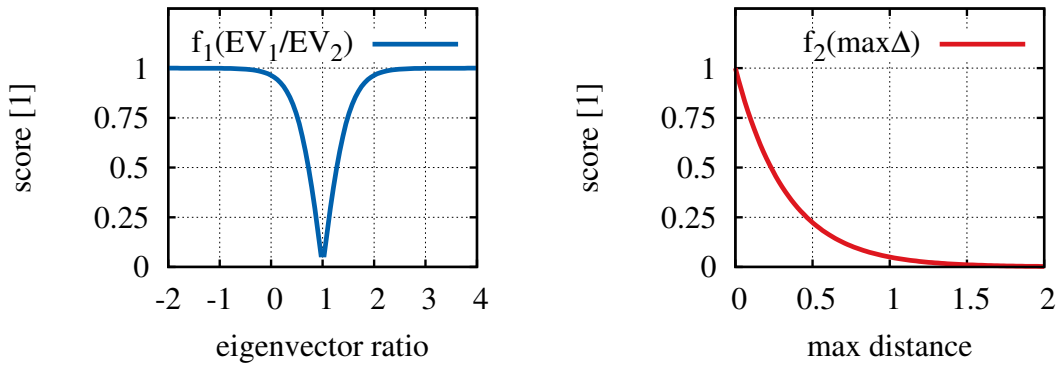


Figure 3.18: Plots of scoring function for evaluation of eigenvalue ratio (left) and blob roundness (right).

Figure 3.18 depicts plots of both $f_1(\frac{EV_1}{EV_2})$ and $f_2(max\Delta)$. A plot of $p(i)$ can be seen in Figure 3.13. Note how eigenvalue ratios close to 1 are severely punished, while all other cases lead to a score of 1, i.e no punishment. Scoring of the max range of estimations yields high values for small differences, and smoothly decreases the higher the maximum range gets.

Finally, a sample plot of the confidence measure of the Axis Primitive for one estimation run is on display in Figure 3.19. The object under consideration was a long cylinder with a prominent main axis. As a result the eigenvalue ratio was also very far away from 1. The buffer size that was used during this experimental run and all other experiments for the Axis Primitive was 25. One can see how the confidence raises during steps around step 40, which is a mirror of the smaller range of estimations after step 16. After sample 52 the estimation does not change anymore. Consequently, the confidence estimation reaches a value of 1 at step 77.

Concluding, I want to note that this is a hand-tailored heuristic scoring function which may in some special cases show unwanted or counterintuitive behavior, such as the weird-looking turns between steps between 43 and 56. But in general, this confidence function has served its purpose well during my experiments.

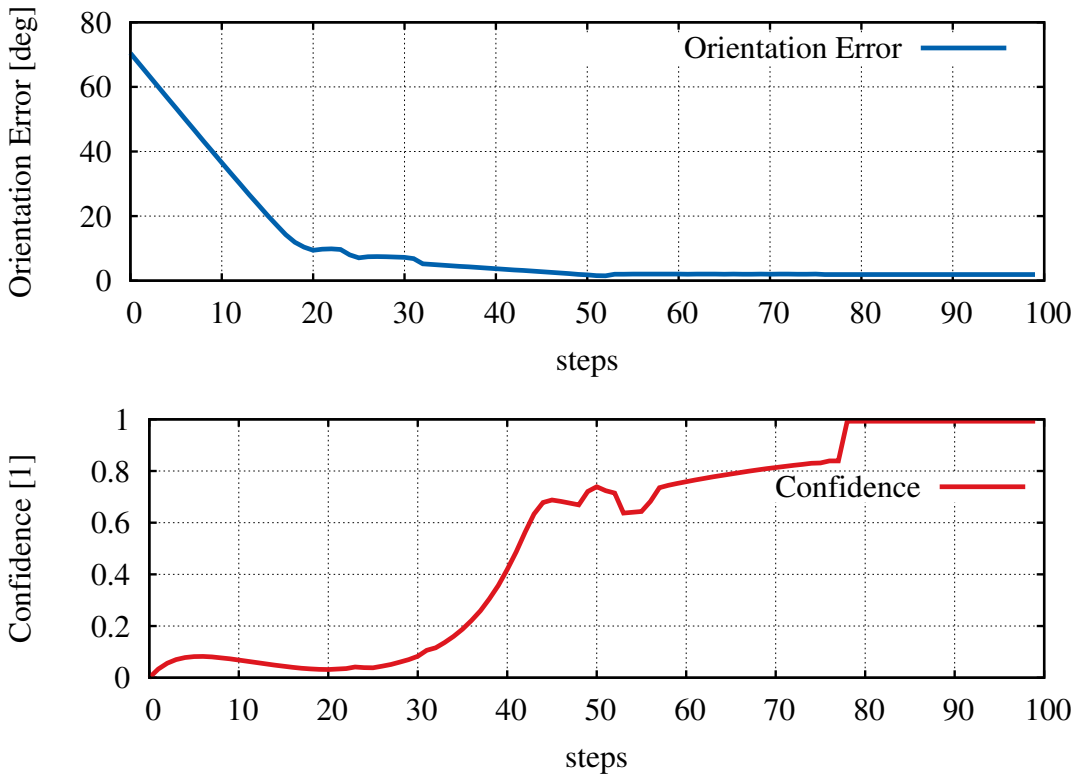


Figure 3.19: Sample plot of an axis estimation of a cylinder. Buffer size $N = 25$ was used. Initially confidence is low. Only 25 steps after the differences between estimations have become smaller, the confidence raises around step 45. 25 Steps after the estimation has converged, the confidence value reaches 1.

3.7 Shape Primitives

The last three primitives all belong to the same category of shape primitives. That means they are responsible for calculating confidence measures which indicate to which shape category the object under observation belongs. Additionally, they also calculate a size estimation for the object. The final pregrasp selection is based on their estimation results. This section will present the design of the confidence functions and the formulas that are used for size and shape estimation of the different shape primitives.

All of the three shape primitives use the Axis and Distance Primitive to provide estimations of the main axis and centroid of the object. This additional information is needed for size calculation and object class estimation. Additionally, the Axis Primitive is used by all three shape primitives to perform the exploration of the

object. As a result, all three shape primitives have the same desired motion for a given object and can perform their estimations in parallel. This is a nice feature as only one single exploration is needed to choose the appropriate pregrasp.

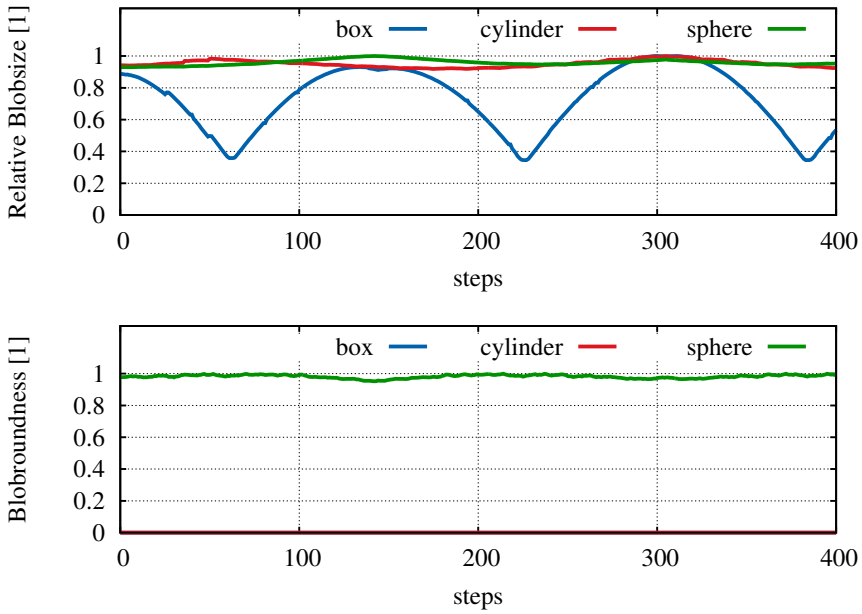


Figure 3.20: Relative blob size and blob roundness are two properties that can be used to discriminate between the three different primitives. The upper plot shows the relative blob size during the exploration of all three objects. Clearly, the box is easy to tell apart, because it shows easily distinguishable maxima and minima. Sphere and cylinder, on the other hand, constantly show a value close to 1. The lower plot shows the blob roundness over time. The sphere is always round, i.e. a constant value of 1, while cylinder and box always yield a non-round blob, which is indicated by a roundness value of 0.

The two blob properties that are used to estimate the class and size of an object are the size and roundness of the blob during the exploration. These two properties are enough to discriminate between the three shape hypotheses. Figure 3.20 depicts the course of the two blob features during the exploration of a sphere, a cylinder, and a box. What becomes obvious from the plots is that the box can be easily identified on the basis of the size feature, because it shows clear maxima and minima. Note that I chose to use the relative blob size, i.e. the current blob size over the overall maximum blob size that has been seen, to make this blob feature comparable for objects of different sizes and exploration trajectories of different radii. Looking at the blob roundness property, on the other hand, one can easily identify the sphere as the only object with a high and constant roundness.

Combining those two blob properties makes it possible to design simple heuristic

functions that tell all three shapes apart: The sphere is an object with constant and high roundness, while showing no huge delta in relative blob size. The cylinder, however, also has a low delta in relative blob size, but constantly shows a non-round blob. Finally, the box also never shows a round blob, while its relative size changes tremendously during exploration.

3.7.1 Sphere Primitive

The calculation of the confidence of the Sphere Primitive is again done by reasoning over the last N blob size and blob roundness measurements. Therefore, the term $p(i)$ that punishes situations with a non-full buffer with a value between 0 and 1 is again used. Since the quality of the distance estimation influences the quality of the sphere estimation (see size calculation below), the average confidence over the last N steps of the Distance Primitive is also multiplied to the equation. Additionally, the average blob roundness, as scored with function $f_1(\frac{EV_1}{EV_2})$ from 3.15 is included to argue about the roundness of the object. Thus, objects with low roundness will automatically lower the confidence of the sphere hypothesis. Finally, a new scoring function $f_3(\Delta_{rel})$ that evaluates the difference between maximum and minimum relative blob size is introduced. For huge deltas it yields a value close to 0, while low deltas, i.e. constant blob size during exploration, result in a value close to 1. The entire confidence function is given as:

$$conf_{sph} = p(i) \cdot avg(conf_{dist}) \cdot avg(\underbrace{f_1(\frac{EV_1}{EV_2})}_{\text{blob roundness}}) \cdot \underbrace{f_3(\frac{blob_{max} - blob_{min}}{blob_{max}})}_{\text{score small blob size range}}. \quad (3.17)$$

The new scoring function that assesses whether or not the relative blob size has stayed constant during the exploration is defined as

$$f_3(\Delta_{rel}) = -0.5 \cdot tanh(15 \cdot (\Delta_{rel} - 0.2)) + 0.5, \quad (3.18)$$

while Figure 3.21 depicts the plot of $f_3(\Delta_{rel})$ for the possible range of relative blob sizes from 0 to 1. One can see that low ranges which indicate constant relative blob size during the last N steps of exploration yield a scoring value close to 1, while big differences between maximum and minimum relative blob size result in scoring values closer to 0. As a result the entire scoring will also get a low value, indicating

a low confidence that the given object is a sphere.

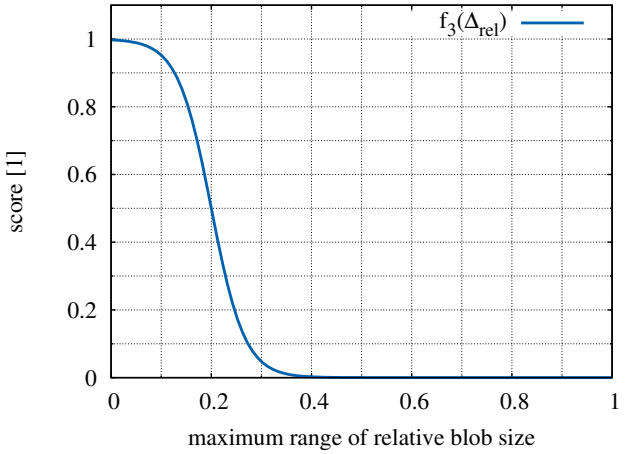


Figure 3.21: Scoring function that assesses the difference between maximum and minimum relative blob size. Big ranges lead to low scores, thus indicating objects with changing blob sizes. Low ranges result in high scores, e.g. in the case of a sphere.

Another important information that the Sphere Primitive delivers is the size estimation of the object. For a sphere it is enough to specify the estimated centroid and radius to completely describe the estimation. The centroid is estimated by the Distance Primitive, and the result just passed on by the Sphere Primitive. For calculation of the estimated radius rad_{est} please consider Figure 3.22 which depicts the geometrical setting during the exploration of a sphere. It is obvious that the radius of the trajectory is a good approximation of the distance between the camera and the visible object silhouette that gives rise to the circle image projection.

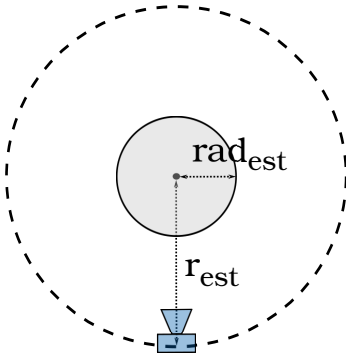


Figure 3.22: Exploration of a sphere seen from above: The radius of the trajectory is an approximation of the distance of the camera to the central cut through the sphere, which gets projected as a circle onto the image.

Using the estimated distance r_{est} from the Distance Primitive, the formula for

calculating the area of a circle, and the focal length of the camera f, the estimated radius of the sphere can be calculated to be

$$rad_{est} = \frac{\sqrt{\frac{blob.size}{\pi}} \cdot r_{est}}{f}. \tag{3.19}$$

3.7.2 Cylinder Primitive

The confidence of the Cylinder Primitive is just a little different from that of the Sphere Primitive. Again a buffer with the blob properties of the last N exploration steps is used to calculate the confidence score. Thus, the punishment term $p(i)$ is used, too. Both rely on the estimations of the Distance Primitive. Therefore the average confidence of the Distance Primitive is again part of the equation. The first difference is that the Cylinder Primitive also takes the orientation of the object under consideration into account. As a result the average confidence of the Axis Primitive is included into the overall scoring function. Thus, objects that do not have a clear main axis are not considered cylinders. The second difference is that the Cylinder Primitive is looking for non-round blobs, whereas the Sphere Confidence was looking for round blobs. Finally, a constant blob size from various view points indicates a cylinder, thus $f_3(\Delta_{rel})$ is again included to assess how big the range between the biggest and smallest blob was. The overall confidence function is defined as:

$$\begin{aligned} conf_{cyl} =& p(i) \cdot avg(conf_{dist}) \cdot avg(conf_{axis}) \cdot \dots \\ & \dots \cdot avg(\underbrace{1 - f_1(\frac{EV_1}{EV_2})}_{\text{blob non-roundness}}) \cdot \underbrace{f_3(\frac{blob_{max} - blob_{min}}{blob_{max}})}_{\text{score small blob size range}} \end{aligned} \tag{3.20}$$

In order to provide a spatial and geometrical estimation of a cylinder one needs to specify the centroid, the orientation, the radius, and the height of the object. The centroid will be estimated by the Distance Primitive, and the orientation is calculated by the Axis Primitive. Thus, the other two parameters are estimated by the Cylinder Primitive. Figure 3.23 shows a top-down view of the exploration of a cylinder. The radius of the trajectory is again a good approximation for distance between camera and object if one wants to calculate the radius of the cylinder. In order to calculate the height of the cylinder, however, one needs use the distance z_H, which is the shortest distance between cylinder and camera. This is necessary

because the length rad_{est} is not negligible in comparison to rad_{est}. Thus, r_{est} and r_H cannot be assumed to be roughly the same.

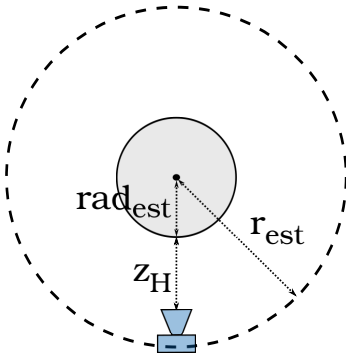


Figure 3.23: Exploration of a cylinder seen from above: As in the case of the sphere, the radius of the trajectory is a good enough approximation for the distance between camera and object to calculate the radius of the object. In order to calculate the height, however, it is better to use z_H as a distance approximation because one cannot assume that $r_{est} \approx z_H$.

Using the pinhole camera model one can calculate the estimated radius of the cylinder to be

$$rad_{est} = \frac{\Delta y \cdot dist}{2f}. \tag{3.21}$$

Again using the pinhole camera model, and as described above z_H as a distance approximation, the height of the cylinder can be obtained with the following formula:

$$h_{est} = \frac{\Delta x \cdot z_H}{f} \quad \text{with} \quad z_H = r_{est} - rad_{est}. \tag{3.22}$$

3.7.3 Box Primitive

The last of the shape primitives is the Box Primitive. Its confidence function is very similar to that of the Cylinder Primitive. Both remember the blob properties from the last N exploration steps. Thus, $p(i)$ as a punishment term for nun-full buffers is included. Average confidence score of Distance and Axis Primitive are part of the equation, as well the average non-roundness of the blob. The only difference is that in case of a box one expects to find a big difference between the maximum and minimum relative blob size. Therefore the confidence function of the Box Primitive has a term that scores high for non-constant blob sizes. This term is given by

$1 - f_1(\frac{EV_1}{EV_2})$. The overall confidence function is defined as:

$$\begin{aligned} conf_{box} =& p(i) \cdot avg(conf_{dist}) \cdot avg(conf_{axis}) \cdot \ldots \\ & \ldots \cdot avg(\underbrace{1 - f_1(\frac{EV_1}{EV_2})}_{\text{blob non-roundness}}) \cdot \underbrace{(1 - f_3(\frac{blob_{max} - blob_{min}}{blob_{max}}))}_{\text{score non-constant blob size range}}. \end{aligned} \tag{3.23}$$

The complete description of the size and position of a box has to include the following parts: Its centroid, its main axis, and the three size parameters width, depth, and height. Estimation of the centroid and the main axis are again provided by the Distance Primitive and the Axis Primitive, respectively. The geometrical setting of the size calculation is depicted in Figure 3.24, that again depicts the scene in a top-down view that is perpendicular to the exploration trajectory. One can see the definition of both width and depth. The height parameter is the length of the box along its main axis, and thus not in depicted in the figure. During the exploration there are two extreme values, each corresponding to a perpendicular view on one of the faces of the box. The dimensions of the blob in each of those cases can be used to calculate the dimensions of the box. Please note that in each of the cases the actual distance between the camera and the face of the box that is projected onto the image is not r_{est}. And since the size of the box is also not negligible in comparison to the small exploration trajectories that the WAM arm allows, one needs to use two approximations z_1 and z_2 as depicted in Figure 3.24.

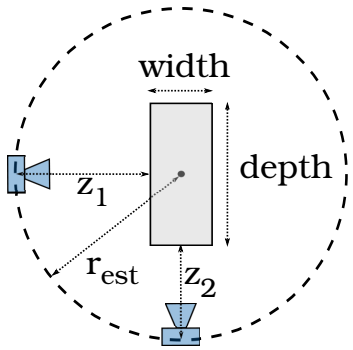


Figure 3.24: Circular exploration of box seen from above: The radius of the trajectory is only a very poor estimation for the distance between the visible maximum and minimum faces of the object and the camera. In order to improve the estimations one can use the depicted distances z_1 and z_2, which are better approximations.

Using the pinhole camera model, one can directly derive the following formula that

describe the maximum and minimum blob size

$$\Delta y_{min} = \frac{blob_{min}}{\Delta x_{min}} = f\frac{width}{z_2} \quad \text{with} \quad z_2 = r_{est} - \frac{depth}{2}, \tag{3.24}$$

and

$$\Delta y_{max} = \frac{blob_{max}}{\Delta x_{max}} = f\frac{depth}{z_1} \quad \text{with} \quad z_1 = r_{est} - \frac{width}{2}. \tag{3.25}$$

Note that Δy is the extent of the blob that is perpendicular to the exploration trajectory. Figure 3.9 visualizes the definition of the image coordinates in relation to the exploration motion. With the help of the above equations, one can directly derive closed-form terms for the depth and width estimation of the box:

$$depth = \frac{2r \cdot \Delta y_{max} \cdot (2f - \Delta y_{min})}{4f^2 - \Delta y_{max} \cdot \Delta y_{min}} \tag{3.26}$$

$$width = \frac{2r \cdot \Delta y_{min} \cdot (2f - \Delta y_{max})}{4f^2 - \Delta y_{max} \cdot \Delta y_{min}} \tag{3.27}$$

Finally, the height of the box can be calculated with help of the pinhole camera model:

$$height = \frac{\Delta x_{min} \cdot z_2}{f} \quad \text{or} \quad height = \frac{\Delta x_{max} \cdot z_1}{f}. \tag{3.28}$$

This concludes my description of the design of all of the five active visual primitives. In this chapter I have presented their design in great detail, and showed how they actively explore an object to model it as either a sphere, a box, or a cylinder. Exploration is done actively to keep the amount of assumptions that are needed to a minimum. Additionally, each next step of the exploration is based on current and past observations to always perform the motion that most increases convergence speed. All of this is done without any assumptions or previous knowledge about the concrete geometrical shape of the objects. In the next chapter I will describe the experimental evaluation, show results, and discuss the performance of the primitives with regards to estimation accuracy, their capabilities to successfully generalize over

object shapes that do not perfectly fit either of the three shape hypotheses, and usefulness of the derived information for successful grasping of unknown objects.

4 Experimental Results and Discussion

In this section, I will present and discuss the simulation and real-world experiments that I conducted to evaluate the performance of the primitives. The experiments were designed to investigate the following question: How good is the estimation of spatial and size properties? Is the estimation of correspondence to a shape class helpful to discriminate correctly between geometrical shapes? Furthermore, are the primitives still able to perceive the desired information even if presented objects are not perfect instances of any of the three shape hypotheses? Additionally, I investigated whether the estimated information enables successful robotic grasping. The last and most important question is whether the overall perceived shape of an object is a good indicator that the corresponding pregrasps will be superior to the others.

In order to investigate these questions I performed three sets of experiments. Firstly, the primitives were used to perceive a perfect box, a perfect sphere, and a perfect cylinder in simulation. This was done to evaluate the performance of the primitives without the influence of any perturbations –neither from the robot nor from the shape of the objects. Secondly, evaluation of the ability of the primitives to generalize across various shapes was done by presenting several "imperfect" objects, i.e. objects that are neither a cylinder nor a sphere nor a box, to the primitives in simulation. Finally, real-world perception and grasping experiments were done to establish whether the primitives still show the same percetual properties in the real world, and whether the estimated information affords successful robotic grasping. Furthermore, this final experimental setting was also used to investigate a possible connection between perceived object shapes and successful preshapes.

4.1 Estimation of Geometrical Primitives in Simulation

Experimental Setting

Firstly, I evaluated the performance of the primitives on perfect instances of the three shape primitives. In simulation a sphere, a box and a cylinder were individually presented on a white table next to the base of the robot. The surrounding scene was defined to also be white. Thus, no other colorful object besides the object under consideration was present. Each object was explored 5 times with each of the three shape primitives performing their estimation in parallel. The exploration was

stopped after 70 exploration steps, and mean values and standard deviations of the estimation results for each of the three objects were calculated.

I chose to use simulation for this evaluation, because a simulator provides complete control over the experimental setup, such as the stimuli that shall be presented, as well as the disturbances that are allowed. Additionally, in simulation one has access to the exact ground truth of the experiments. Thus, exact evaluation of the estimation results is possible. Since a lot of real-world perturbations, such as e.g. inaccurate motion control or bad lighting conditions, were not present in the simulated experiments, the obtained results present the best-case performance to be expected from the primitives.

Results

Figure 4.1 depicts the results of the centroid and main axis estimations for all three shapes. In order to assess the centroid estimations the mean Euclidean errors between the estimated and the actual centroids were calculated. Whereas for the axis estimations the angles between the estimated and the actual axes of the objects were used to evaluate the quality of the estimations. In all three case the centroid estimation yielded a mean Euclidean error of about 1 cm, and a very small standard deviation of less than 2 mm. The axis estimation –which is not necessary for a sphere– was done with a mean error of around 4 degrees, while the standard deviation of the errors again had a very small value – the biggest standard deviation was roughly 0.6 deg for the axis estimation of the box. Thus, both the centroid and main axis estimation performed at a very high level.

The results of the size parameter estimations are on display in Figure 4.2. For the sphere only the radius had to be estimated, while the cylinder was described by its radius and height, and the box was modeled with its three size parameters width, depth, and height. For all six parameters mean errors and error standard deviations were calculated and used for visualization. All parameter estimations were done with very high accuracy (note that the error is depicted in mm in Figure 4.2). The size of the presented box was $(width \times depth \times height) = (2cm \times 6cm \times 12cm)$, while the sphere had a radius of 5 cm, and the cylinder had the dimensions $(radius \times height) = (3cm \times 15cm)$. The standard deviation of the errors of all estimations was negligibly small. Thus, one can conclude that the size estimation for all three objects reliably yielded very accurate results.

Finally, the results of the shape estimations that were done by the three shape primitives are depicted in Figure 4.3. Each of the three shape primitives yielded a

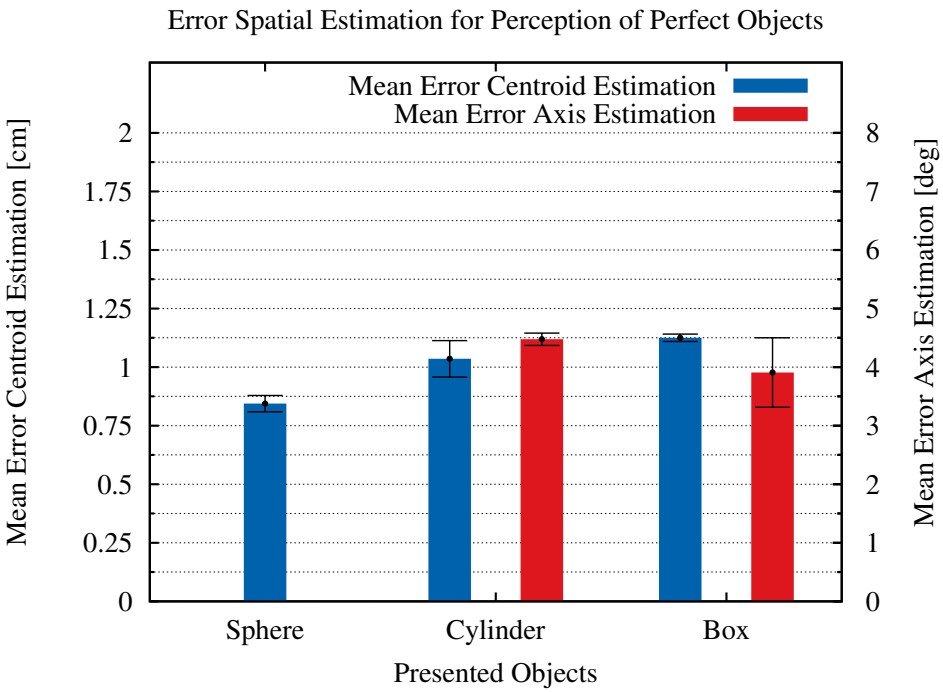


Figure 4.1: Mean errors of the centroid estimation for all three objects, and mean estimation errors of the main axis estimation for a perfect cylinder and a perfect box. Each object was presented separately, and explored 5 times. The colored bars indicate the mean errors, while the small black bars visualize the standard deviations of the errors. Results of all estimations had a small mean error, while the standard deviations of the errors were very small.

confidence score for each of the overall 15 experimental runs. Thus, for each single object there were five shape estimation results for each primitive. Mean confidences and the corresponding standard deviations are on display in Figure 4.3. The exploration of the box (depicted on the very left) yielded a very high mean score for the Box Primitive and a mean score of almost 0 for the other two hypotheses. Thus, the box could be easily identified on the basis of the three scores. The perception of the sphere yielded similarly good results: Only the Sphere Primitive returned a high mean confidence value, while the other primitives reported extremely low mean confidences. The results for the cylinder (on the very right) were also very good, though not as perfect: The Cylinder Primitive correctly returned a very high mean confidence score, while the Box Primitive did also have a small but non-negligible mean confidence value. Please note that the confidence scores of the primitives are not probabilities, thus they do not have to add up to 1 for one single perception. The standard deviations of all confidence scores for all objects were very small. Therefore the performance of the shape perception for perfect instances of the three different

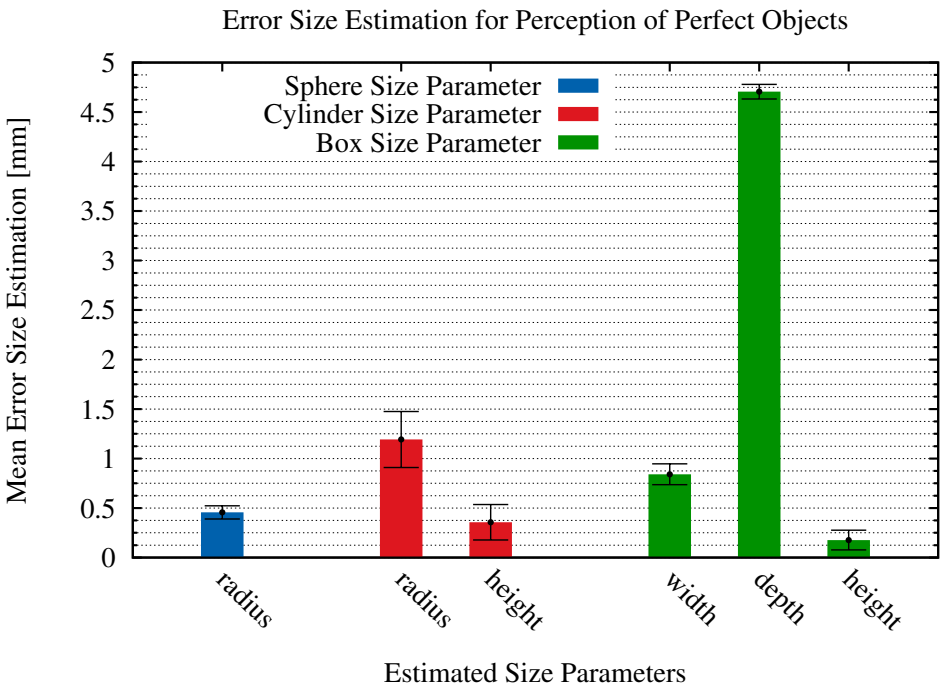


Figure 4.2: Results of size estimation for the three perfect objects in simulation. Mean errors and error standard deviations for all six estimated size parameters are on display. The colored bars show the mean errors, while the black error bars indicate the calculated standard deviations. Please note that the unit of this plot is mm. Estimation was done with high accuracy and very high reliability, as mean errors and error standard deviations were very small.

hypotheses can be characterized as very high and reliable.

Discussion

Summing up, one can say that spatial, size, and shape confidence estimations for the case of perfect instances of the three shape hypotheses worked very well. Mean errors and error standard deviations in size and spatial estimation were very small, and the mean confidences of the shape primitives clearly identified the objects under consideration. However, since the experiments were done in simulation and thus free of perturbations and since only objects that fitted perfectly onto one of the perceptual categories of the visual system were presented, these results represent the best-case scenario.

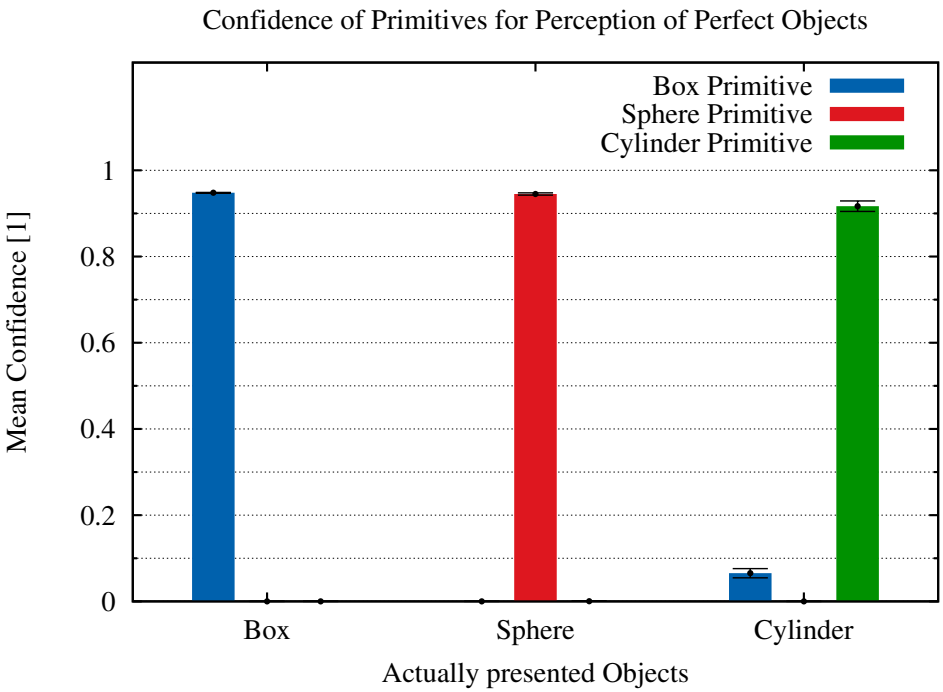


Figure 4.3: Results of the shape perception experiments: For each of the three presented objects each shape primitive performed five confidence estimations. Mean values and standard deviations of the confidence scores were calculated for each object-primitive-combination. The colored bars show the mean values, while the black error bars indicate the standard deviations of the confidence scores. With the exception of the small and wrong confidence value of the Box Primitive after the exploration of the a cylinder, the mean results were almost perfect. Standard deviations were very small and indicate reliable performance.

4.2 Generalizing Properties of the Visual System

Experimental Setting

In order to remove the limitation of perfect objects from the experimental setup, I chose to investigate the question whether the primitives were still able to perceive the rough shape of an object if it was neither a perfect sphere nor a perfect cylinder nor a perfect box. Therefore I presented five different objects in simulation that belonged to neither of the three perceptual shape categories. Each object was again explored five times to make sure that single experimental results were no chance events. Mean perception results were analyzed to answer the question that I raised above. Simulation was again the surrounding of choice because it allowed me to remove all other sources of perturbation besides non-perfect objects from the experiments. The objects were again colorful, while the table that they were placed on and the

background were completely white.

Results

The first object that was presented was the cup that is on display in Figure 4.5. The shape of the cup could be described as a truncated cone. Thus, the shape category that it is most similar to is that of the cylinder. The mean results of the shape perception of the three shape primitives come to the same conclusion: The mean confidence score of the Cylinder Primitive is very high, while that of the other two primitives is very small. Additionally, the standard deviations of the confidence estimations were very small; again indicating reliable performance. In order to assess the quality of the spatial and size estimation, I added a semi-transparent version of the final cylinder approximation of the cup to the simulated scene. A view of this setup can be scene in Figure 4.5. Obviously, centroid, orientation, and size of the estimation are good approximations of the overall shape of the cup.

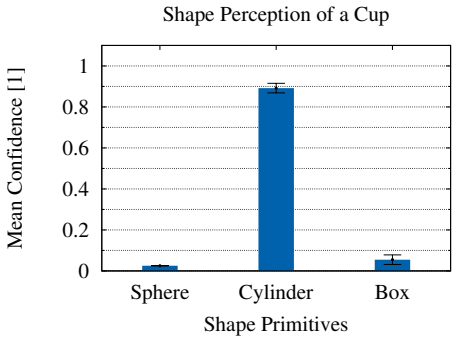


Figure 4.4: Results of shape perception of a cup for five explorations: The Cylinder Primitive displays a high mean confidence, while the other two primitives have very small mean values. Standard deviations of the scores –depicted through the small black error bares– are very small. Thus, the cup has reliably been perceived as a very cylinder-like object.

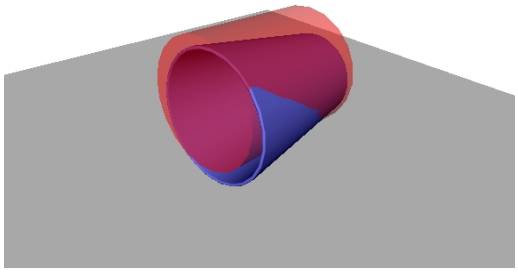


Figure 4.5: Presented cup and its calculated final estimation: The cylinder fits the overall shape of the cup pretty well. Orientation and position are also estimated pretty well.

The second object under consideration was an angular cup. This object is very interesting because it represents the first object that is in a way a mixture of two of the perceptual shape categories of the system: A cylinder and a box. Mean confidence scores of the 5 experimental runs that were conducted support this view. The Cylinder Primitive returned a high mean confidence score, but it was not as

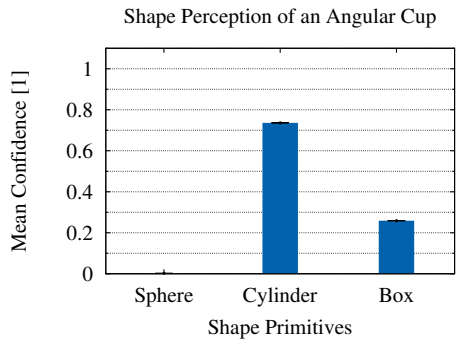


Figure 4.6: Perceptual results of five explorations of an angular cup: Mean confidence scores indicate an object that is very cylinder-like, but also has features of a box. These observations fit the description of an angular cup very well. Very small standard deviations of the confidence score show reliable results across all five experimental runs.

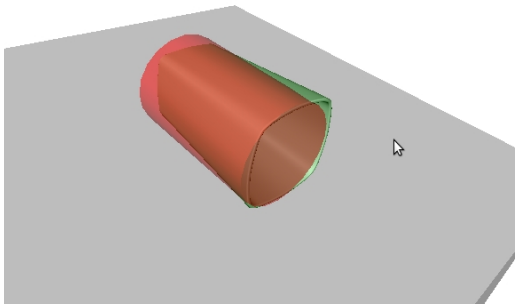


Figure 4.7: A view of the angular cup and its semi-transparent final cylinder estimation in simulation: Only a small error in pose and orientation is visible as most of the object lie within the approximation.

high as that for the perception of the cup was. At the same time, the Box Confidence had a small, but definitely non-negligible mean confidence score of 0.26. These are very important as they clearly show that the primitives are capable of perceiving general geometric shape properties such as *box-likeness* or *cylinder-likeness* in one single object. Standard deviations of the confidence scores were extremely small, so one can expect reliable performance of the primitives for this object. Figure 4.7 depicts the angular cup and its final semi-transparent cylinder approximation. Clearly, results of the centroid, the main axis, and the size parameters estimations were good, too.

The third object that was explored was a beer bottle. A bottle could again be described best as a cylinder because it has a distinct main axis and is rotation symmetric. Thus, when moving perpendicularly around this axis one sees a constant blob size. Note that the silhouette of a beer bottle that one sees during such a motion is very different from that of a cylinder. However, one would clearly want to still perceive the bottle as a cylinder. The mean confidence results from the three shape primitives that are presented in Figure 4.8 also support this interpretation: The mean confidence value of the cylinder primitive is very high, while the mean scores of the other two primitives are very small. This is because the primitives use the blob size during the exploration, and not the silhouette of an object, to score the shape confidence. Low confidence standard deviations that are indicated by the

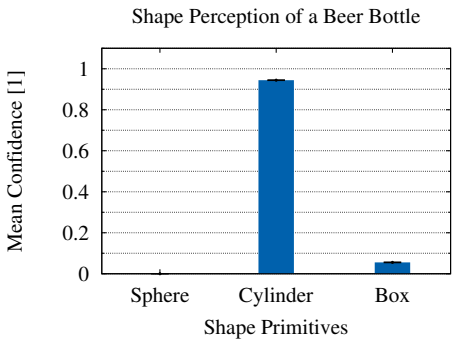


Figure 4.8: Results of shape perception for a beer bottle. The mean confidence from five experimental runs of the Cylinder Primitive is very high, while the other shape primitives have a very small mean score. Thus, the object is perceived as a very cylinder-like object. The small black errors bars indicate confidence standard deviations that were very small across all primitives.

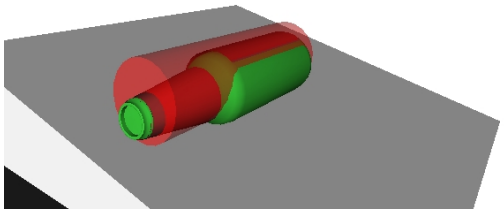


Figure 4.9: Depiction of the beer bottle and its cylinder estimation (semi-transparent object). The cylinder hypothesis is a good approximation of the object, but a small error in a position estimation is clearly visible.

small black error bars point at reliable performance. The image of the bottle and its cylinder estimation in Figure 4.9 show that the cylinder hypothesis is a good approximation of the object, even though a small positioning error is apparent.

In order to test the perceptual system on an object that I would not expect to be cylinder-like I stacked two angular bowls on top of each other. The resulting object and its dominant estimation (a sphere) are on display in Figure 4.11. The mean confidence estimation and standard deviations of the estimations are depicted in Figure 4.10. The mean confidence of the Sphere Primitive is above 0.8, while the other two hypotheses have a mean score below 0.05. Thus the visual system perceives the two bowls clearly as sphere-like. Additionally, the very small confidence standard deviation indicates reliable performance and further supports this slightly surprising result. Although one of the two bowls on its own does not look like half of a sphere, views from all parts of the exploration trajectory around the two bowls yielded blobs with very similar blob size and an eigenvalue ratio that was very close to 1. As a result, the Sphere Primitive reported a high mean score for the sphere. Figure 4.11 shows the good fit of the sphere estimation around the two bowls, thus further supporting the perception of this object as very sphere-like.

Finally, I chose to investigate the perception of an object that does not even slightly fit into any of the three shape categories that the primitives are trying to find, thus,

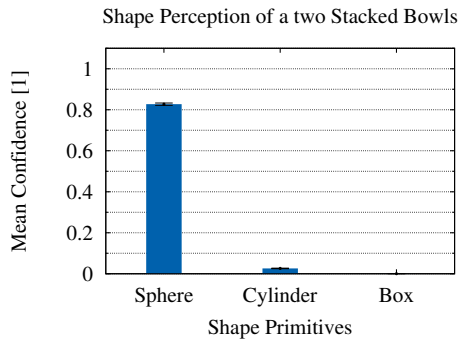


Figure 4.10: Mean confidence scores of the shape primitives for the perception of two bowls that are stacked on top of each other. The Sphere Primitives showed a high mean value of more than 0.8, while the other two primitives had mean values below 0.05. As a result the combination of the two bowls, can be considered very sphere-like. Low confidence standard deviation across the five experiments indicate reliable performance.

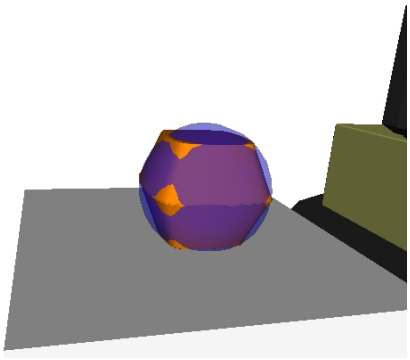


Figure 4.11: Image of the two bowls on top of each other and their final approximation with the sphere estimation (blue semi-transparent object). The sphere seems to fit the object pretty well, indicating also a good estimation of the centroid and size of the object.

using this as the ultimate test for the capabilities of the primitives to generalize over different object shapes. As a model I chose the mesh-model of a dog. An image of the model with the final best approximation is on display in Figure 4.13. One can clearly see that the dog is neither a sphere, nor a cylinder, nor a box. The mean shape estimations of the primitives, however, perceive it with high confidence (value above 0.8, while the others are practically 0) as a box. The results are shown in Figure 4.12. Negligibly small standard deviations hint again at reliable performance. At first, this perception as a box appears to be somewhat counterintuitive. But contemplation of the perceptual process of the primitives with the model of the dog in mind helps explaining these results. The blob of the dog obviously has a clear main axis from various views during the exploration, which all project onto the same 3D-axis. Additionally, the size of the blob is much smaller when seeing the dog from the top, in comparison to the seeing it from the side. Thus, there is a huge delta in blob size during exploration. So, there is a high confidence that this a box. Figure 4.13 also shows the final box estimation of the dog. The box obviously is not a bounding box around the entire dog, which is because in equations 3.24 and 3.25 I did not use maximum extents of the blob but the maximum and minimum blob size to calculate the size of the box. As a result, the head and the legs of the dog do not

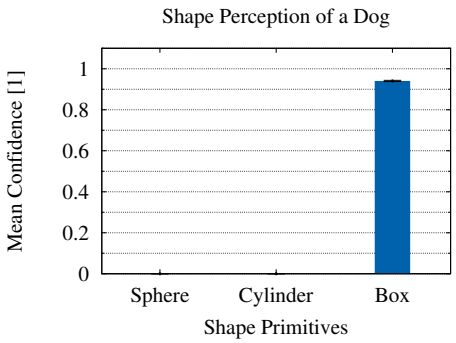


Figure 4.12: Mean shape perception of the dog model. A high mean value of the Box Primitive and mean values of practically 0 indicate that this is a box. Very small standard deviations again indicate reliable performance.

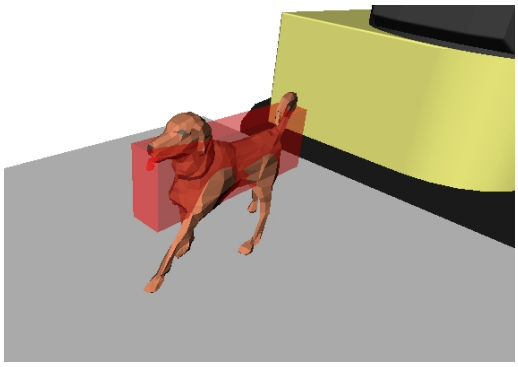


Figure 4.13: View of the dog model and the calculated box estimation. Since the head and the legs do not contribute a lot to the blob size of the dog, the box mainly approximates the body of the animal.

contribute a lot to the size estimation. Thus, the box approximates the body of the dog well, and also make sense in terms of grasping it.

Discussion

Summarizing, one can say that the shape primitives yielded the intuitively correct results for perception of objects that were only slightly different than one of the hypotheses, but also for objects that were very different. Across all experiments the confidence standard deviation of all primitives was extremely low. So, for overall 125 confidence estimations the primitives delivered results with high reliability. Images of the shape estimations with the highest score and of the objects under consideration showed intuitively good approximations with small positioning and size errors. Thus, one can conclude that the primitives are capable of perceiving general geometrical properties over very different kinds of objects in simulation.

4.3 Real-World Grasping Performance

Experimental Setting

The final experimental setup investigated whether the primitives are also able to show the same capabilities for perception of real-world objects, and more importantly whether the perceived information is sufficient to perform successful grasping experiments. Additionally, I examined to which extent the perceived shape of an

object indicates properties of successful pregrasps.

During the experiments nine different real-world objects were one by one placed on a desk with a white surface. Objects were chosen to be either very colorful or wrapped in yellow paper (whiskey case and board game box). Each of the shape primitives explored each of the objects *separately* ten times. Exploration was stopped after 70 exploration steps, and then the shape primitives determined the pregrasp properties based on the estimated information and executed the pregrasps. Afterwards the fingers of the hand were closed in torque mode, i.e. closing stopped for each finger once a certain torque threshold was exceeded in that very finger, thus imitating a blind close-mode that stops fingers after collisions occurred. After completion of the closing operation, the wrist was lifted 10 cm. A grasp was deemed successful if the closed hand could lift the object, i.e. no more contact between object and table, and hold the object in this pose for ten seconds. This setup resulted in a total of 270 grasping experiments –a comparably large experimental sample size. Since each of the primitives separately performed exploration, estimation, and pregrasp selection a comparison of the success of the different strategies was possible.

The following simple heuristics were used to determine the pregrasps of the different shape primitives:

- *Sphere Primitive*: After the estimation was stopped, the exploration (without further update of the estimation) was continued until a local minimum of the angle between the z-axis of the camera frame and z-axis of the global frame, i.e. gravitation vector, was found. The orientation of the wrist at this point was chosen as the pregrasp orientation, thus ensuring an orientation that was pointing downwards as much as possible. The end-effector position was chosen to coincide with the estimated centroid of the sphere. The preshape of the hand was chosen to be spherical as depicted in Figure 3.1.

- *Cylinder Primitive*: After the estimation, the exploration was also continued without further estimation until a maximally downwards-pointing orientation was found. The estimated centroid of the cylinder was chosen as the desired end-effector position. The preshape of the hand was set to be cylindrical, as shown in Figure 3.1.

- *Box Primitive*: After the estimation, the exploration continued without updating the estimation until the global minimum of the blob size was revisited,

thus assuring grasping a box around its smallest face. In order to avoid endless searching for a global minimum that does no longer exist, e.g. because the hand had gotten closer to the object due to positioning errors, view points that got within plus 10% of the global minimum were also accepted. The desired end-effector was chosen to be the estimated center of the cylinder, and the hand preshape set as cylindrical.

The nine objects that were used for the grasping are on display in Figure 4.14. They cover various shapes and sizes of everyday household objects that one would expect a robot to be able to pick up to be called autonomous. There were three kinds of fruits or vegetables: an apple, a banana, and a bell pepper; three medium-sized man-made objects: a spectacle case, a toy bridge, and a sponge; and three big objects: a soccer ball, a whiskey container, and a board game box.

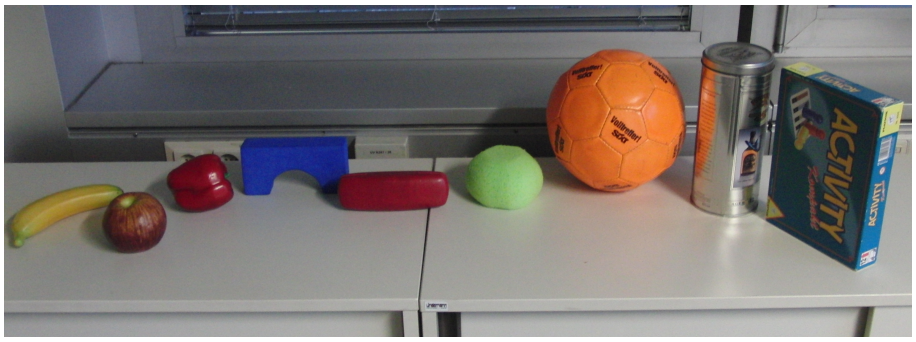


Figure 4.14: Objects that were used in real-world my grasping experiments. There was a group of three natural objects: An apple, a banana, and a bell pepper. Then there were three medium-sized man-made objects: a toy bridge, a spectacle case, and a sponge. Finally, there were three big man-made objects: A soccer ball, a whiskey case, and board game.

Results

The first results that I want to present are those of the apple grasping. They are on display in Figure 4.15. The grasping success of the ten experimental runs with each of the three shape primitives and their corresponding pregrasp hypotheses are depicted by the red bars. The blue bars indicate the mean shape confidence of the three primitives, while the small black error bars again show the standard deviations of the confidence scores. All of the results of the other following experiments will be presented in the same way, thus, I will not repeat this formal description of the plots for later experimental results.

The shape estimation of the apple shows results that are not as perfect as could be expected after the promising results from simulation: The sphere primitive, which

one would clearly expect to be the most confident, indeed has the highest mean confidence value. But it is a very small confidence value, and the difference in comparison to the other two hypotheses is rather small. Additionally, the standard deviation is rather big for the Sphere Primitive. Thus, one could argue that the Sphere Primitive could only marginally be detected as the most confident primitive in a parallel estimation setting.

During the experiments I identified two main reasons for this phenomenon. Firstly, during some of the experimental runs the lighting conditions changed rather dramatically, causing the background window and the blob of the apple to "melt" together in the image, thus yielding several views with an extremely high blob size. This caused a high delta in relative blob size, thus effectively making the sphere hypothesis very unlikely. Additionally, no main axis was found. So, none of the primitives displayed a high confidence score. Secondly, the apple was not perfectly round. That meant that the blob size did show a certain delta during the exploration, which already resulted in a punishment through scoring function $f_3(\Delta_{rel})$. Thus, two points for improvements of the visual system have already been identified: Further tweaking of $f_3(\Delta_{rel})$, and improvement of the blob detection. I have already pointed out the latter during the description of the image processing setup in section 3.3.

The grasping results, however, show very good performances. The only pregrasp that yielded unsuccessful grasps was the cylinder grasp. In those two error cases one of the fingers hit the object earlier than the others, thus pushing the object out of the closing area of the fingers. Surprisingly, the box grasp yielded good grasps, too. This is surprising because the pregrasp orientation is determined as the view with the minimal blob size. For the case of a sphere-like or cylinder-like object this can be any view point. Thus, the pregrasp orientation will be equal to the orientation the wrist had when the exploration stopped, possibly leading to approach directions that result in collisions of the fingers with the table. In the case of the Box Sphere exploring the apple, however, the exploration only stopped in situations in which no such collisions occurred.

The second object under consideration was the bell pepper. The results of perceptual estimation and grasping are on display in Figure 4.16. In the case of the bell pepper a most-confident shape primitive was obvious: It was the Cylinder Primitive. It showed a mean confidence of almost 0.7, while the other two primitives had mean values well below 0.1. The standard deviation of the perception, however, was again pretty high. During the experimental sessions of the bell pepper changing

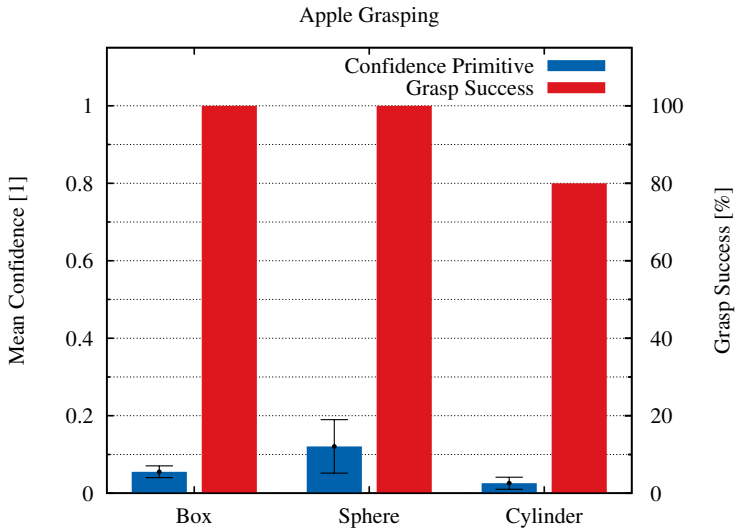


Figure 4.15: Grasping results and shape perception of an apple in real-world. The red bars indicate grasping success with each of the three pregrasps. The blue bars show the mean shape confidence of each of the shape primitives after exploration. The small black error bars show the confidence standard deviations. Each shape primitive was run 10 times, yielding 30 apple grasping experiments. Please note that the layout of the following plots will be identical to this one. Shape perception only showed a small difference between the hypothesis with the highest mean score and the rest, while the standard deviation of the confidences was rather big. The grasping performance for all the pregrasp hypotheses was very high.

lighting conditions did again lead to unnaturally big blobs from some view points, thus causing some trials with a higher confidence of the Box Primitive.

The grasping experiments showed very good performance of the Sphere and Cylinder Primitive. Both only had one error due to pushing of the object with one finger. The box pregrasps, however, led to several pregrasps that resulted in collisions of one or two fingers with the top of the table. Thus, one could conclude that choosing the estimated center of the bell pepper as the desired end-effector position and a downwards orientation of the wrist were sufficient for successful grasping of the bell pepper.

Afterwards, I investigated the performance of the perceptual primitives and the pregrasps on the banana. The results are depicted in Figure 4.17. The banana was clearly seen as a box-like object, while the other shape primitives had extremely small values. The confidence standard deviations were also very low, indicating reliable performance during all 30 experimental runs. The estimation as a box-like object is also not too surprising, as the banana has both, a clear 3D main axis and a

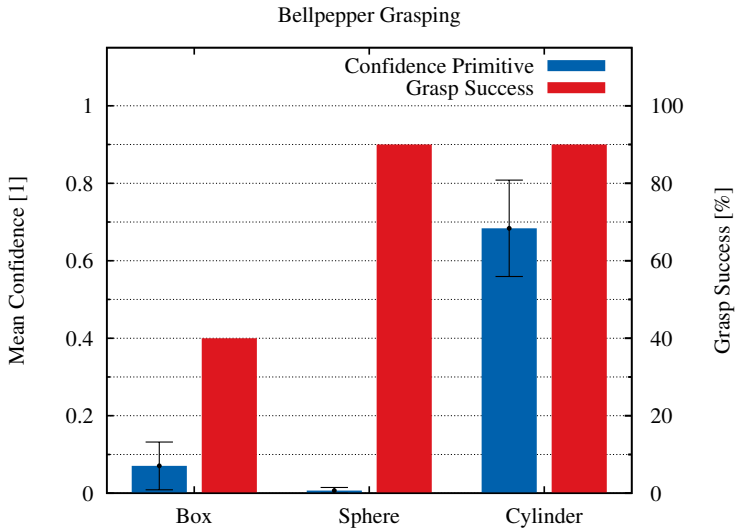


Figure 4.16: Perceptual and grasping results for the bell pepper. The mean shape confidences clearly indicate a cylinder-like object. The standard deviations, however, are pretty high – especially for the Cylinder Primitive. Grasping success for both the sphere and cylinder pregrasps were very high, while the box hypothesis led to several collisions of fingers with the top of the table.

high delta between the maximum and minimum relative blob size. Grasping performances for the pregrasps that used the cylindrical preshape was very high, while the cylindrical preshape of the hand appeared to be too small to grasp the object. As a result several attempts with the spherical pregrasps failed, while others resulted in awkward-looking grasps. Please note that the banana was positioned in various poses with its minimum side always directed such that it could be reached by the box pregrasps.

The next object under consideration was the toy bridge. Figure 4.18 depicts the resulting performance. Perception of the toy bridge yielded what one could be intuitively expecting –a clear percept of a box-like object. In fact, the toy bridge is just a box with a piece missing. Thus, the resulting high mean value of the Box Primitive and the low mean values of the other two shape primitives mimic the results from the previous section that investigated the general perceptual properties of the primitives in simulation. The standard deviations were also very low.

With regards to the grasping performance, one has to note that all three pregrasps yielded perfect results. This was because the minimal side of the box was reachable, and because the toy bridge was small enough for the BarrettHand to reach around. Thus, also the spherical pregrasp successfully encased the object.

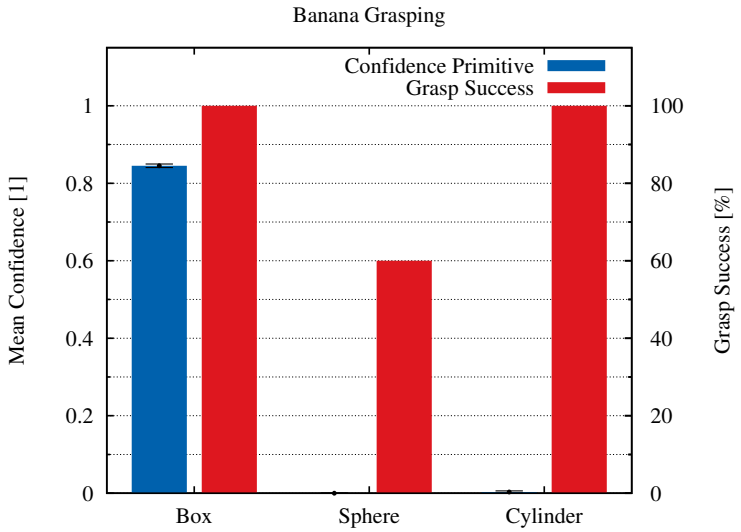


Figure 4.17: Experimental results for the banana. Perception identified a box-like object with very low standard deviation. Both pregrasps using a cylindrical pregrasps yielded perfect performance, while the spherical preshape often proved to be too small to completely wrap the fingers around the object.

Subsequently, I conducted experiments with the spectacle case. The results are displayed in Figure 4.19. In fact, the perceptual results and the grasping performance are almost identical to that of the toy bridge: The mean confidences clearly show a box-like object, which is not surprising as the spectacle case is basically a box with round edges and surfaces. The standard deviation of the box confidence, however, is bigger than that of the perception of the toy bridge, though still acceptable. Furthermore, the grasping performance was very good. The only error occurred using the box pregrasp, when the hand closed before the pregrasp position was reached – a bug that only surfaced twice during all my experiments, and that I could therefore not trace. The spherical preshape was again able to completely wrap around the medium-sized objects.

In Figure 4.20 the experimental results of the perception and grasping of the sponge are displayed. These are very interesting for two reasons. First of all, the perception of this object indicates that it has box-like, sphere-like, and cylinder-like properties in medium proportions. Additionally, the standard deviations of all three confidence values are quite high. Especially the Cylinder Primitive showed a very high difference in confidence estimations, while it was also the shape primitive with the highest

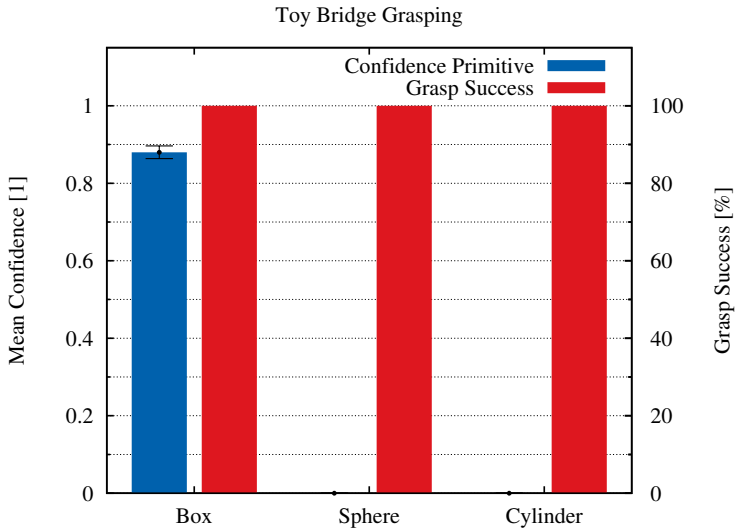


Figure 4.18: Results of perceiving and grasping the toy bridge: Perception shows a clear box-like object with very small standard deviations. Grasping performances were perfect for all pregrasps, as all hand preshapes were able to completely encase the relatively small object.

mean score. Thus, one can conclude that this is an object that does have properties of all three hypotheses, and thus small alterations in the exploration lead to rather big differences in perception – a sign of nonlinearity. The second interesting aspect of this experiment is that the sponge itself is a very compliant object. Thus, it facilitates grasping for the hand to a big extent, because it shapes *itself* to perfectly fit the fingers very well. The grasping results reflect this fact, too. None of the pregrasps failed to successfully grasps the sponge.

So far, almost all of the experimental results showed high grasping success, no matter the actual and perceived shape of the objects. Even in cases in which the preshape did not appear to be a good candidate, successful grasping was still possible. Since the fingers of the hand could almost always completely wrap around the object, it managed to encase the object, thus creating a lot of contact points. The only exception to this observation was the performance of the spherical preshape on the banana. Here the hand was not able to wrap around the object in all chosen pregrasp poses. Inspired by this finding, I continued to grasp bigger objects, that would challenge the capabilities of the hand more. The hypothesis behind this was that for big objects a correspondence between perceived object shape, and thus a good pregrasp, and grasping success would surface.

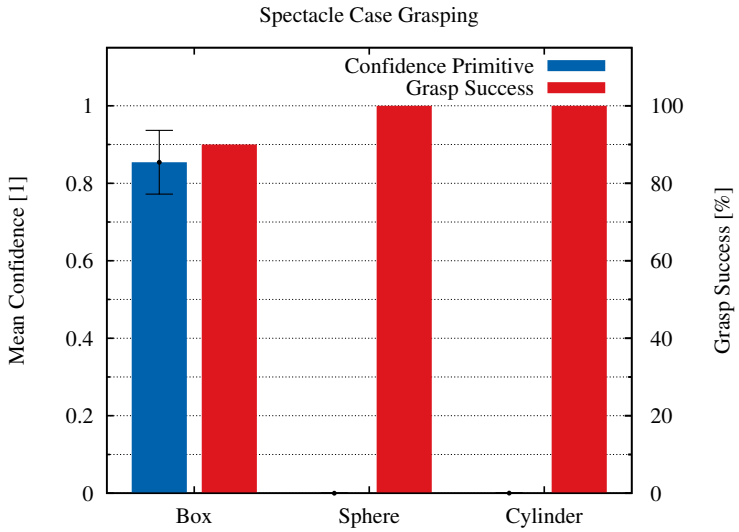


Figure 4.19: Grasping and perception of the spectacle case: Mean confidences indicate a box-like object. Standard deviation of the box confidence was medium. Grasping performance was again very high across all three pregrasp hypotheses, while the only error of the box pregrasp was caused by a bug of the hand closing routine.

The first of the big objects under consideration was the soccer ball. Perception and grasping results are on display in Figure 4.21. The mean confidences of the shape primitives clearly indicate a sphere-like object, with a high standard deviation, however. Both pregrasps that use a cylindrical preshape failed to grasp the ball in all runs. The sphere pregrasp, however, succeeded in 8 of the 10 experimental runs. In both error cases the slippery ball fell out of the hand after closing because the fingers did not encase the object, i.e. completely reach behind it. This is a clear sign that the size of the ball did challenge the capabilities of the hand and the quality of the pregrasp more than the smaller apple, which also had a sphere-like shape.

The second big object that I investigated was the whiskey case. In order to ensure a colorful appearance, the object was wrapped in yellow paper for the experiments. The mean confidence scores of the shape primitives indicatee a cylinder-like object, while the box hypothesis also showed a significant mean value. This conflicts a little with the actual shape of the whiskey case which was cylindric. The reason for this false detection of box-likeness was that during some steps of the exploration the big object partly left the camera image –an error case which I had not foreseen in my design. This led to differences in blob-size across several view points and caused the punishment of the cylinder hypothesis and the increased confidence in the box

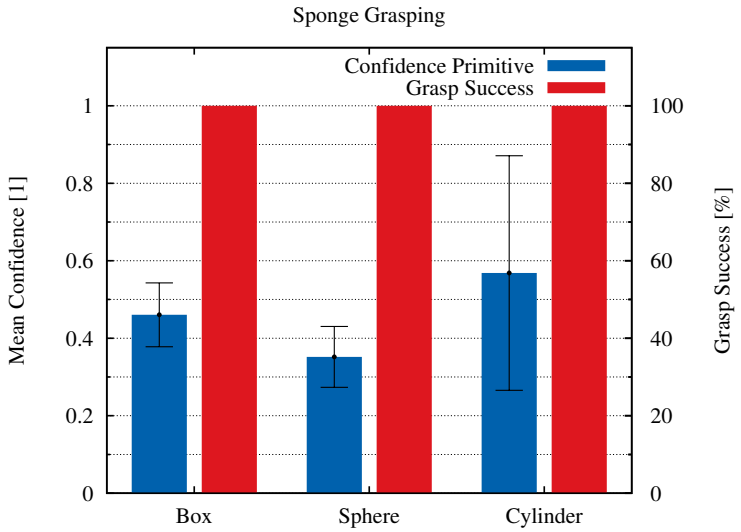


Figure 4.20: Perception of the sponge yielded results that indicate properties of all three shape hypotheses, as the mean confidence for all three shapes is medium. Additionally, the high standard deviations of the confidences further points to an unclear percept. Despite the unclear perception, grasping was perfect for all three pregrasps, as the high compliance of the object greatly facilitated the grasping task.

hypothesis, respectively. The grasping performance of the cylinder pregrasp was excellent, while the box pregrasp yielded three trials with collisions between table top and object. The performance of the spherical preshape was much poorer, with only two successful trials.

At last, the board game box was considered. In order to ensure successful object segmentation based on color it was also wrapped in yellow paper. The results of the perception show a very high mean confidence of the Box Primitive, while the other two primitives yielded mean confidence of almost zero. Thus, the board game box was perceived as a box-like object. The small standard deviations of all three confidence scores show that estimation was done with high reliability. The performance of the pregrasps for grasping shows that the spherical preshape was hardly successful. The cylinder pregrasp was only slightly better, failing in half of the trials. The best performance was achieved using the box pregrasp which resulted in successful grasps for all ten experimental trials. Figure 4.23 displays the perceptual and grasping results for the board game box.

Summing up the grasping results of the big objects, it can be said that they do support the hypothesis that actual and perceived object shape indicate more

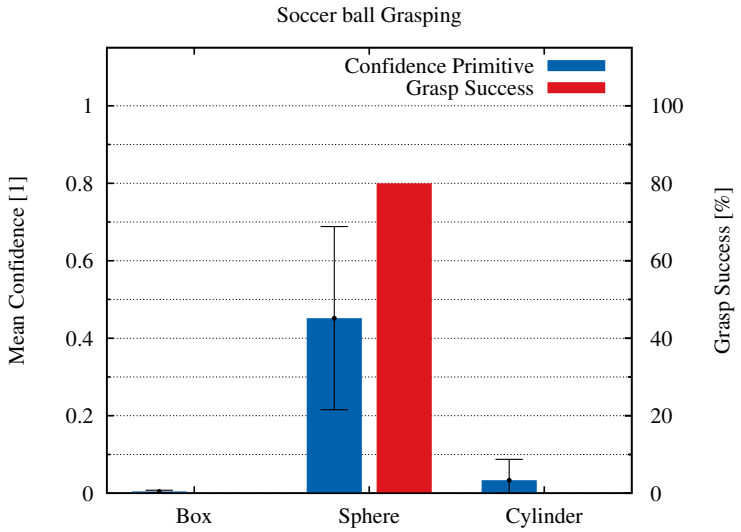


Figure 4.21: The soccer ball was perceived as a sphere-like object, with only the Sphere Primitive showing a significant mean confidence value. The grasping results show no success for any of the runs that used the cylindrical preshape, and good performance of the spherical preshape.

successful preshapes, and thus pregrasps of the hand. The results of the soccer ball experiment underline this very well, as the cylindrical preshape completely failed to yield successful grasps. But also the experiments with the whiskey case and the board game box display an obvious correlation between perceived shape and successful pregrasps.

Discussion

With regards to the interpretation of the entire results of all nine experimental sets, one can conclude that the shape primitives are capable of correctly perceiving the overall shape of an object also in real-world experiments. The perceptions of the apple and the whiskey case are to name as the only minor exceptions, because for both cases the highest mean confidence was indeed correct, but there were also high mean confidences for alternative shape hypotheses that were not correct. For both experiments I did, however, point out the flaws in the perceptual system, e.g. bad design of scoring function $f_3(\Delta_{rel})$ and no error handling in case parts of an object get out of the camera image, that led to these wrong perceptions. As for the case of the sponge, the shape perception was unclear, which is indeed the correct answer, as the the shape of the sponge clearly does not fall completely into to either of the three categories. Additionally, one has to note that for several of the experiments

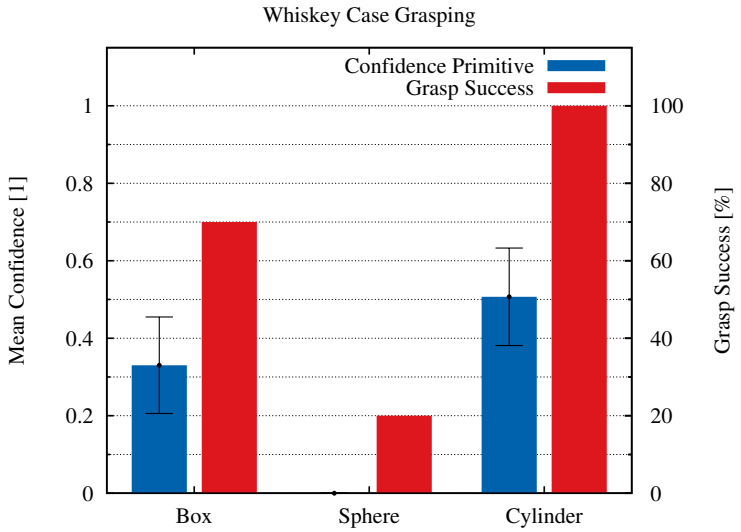


Figure 4.22: Perception of the whiskey case hints at a cylinder-like object, even though the box hypothesis also has a considerable mean confidence. This faulty perception was caused by several views which had parts of the big object not in the image, thus leading to a changing blob size. The grasping experiments showed that the cylindrical preshape outperformed the spherical one.

significant standard deviations of the confidence scores were observed, which was not the case in the simulated experiments, indicating higher uncertainty in the perception. Still, in a parallel estimation experiment, only the perception of the apple would have been subject to a changing highest-confident hypothesis.

Regarding the grasping success of the experiments, the performance of the system was very good. Selecting the pregrasp of the shape primitive with the highest confidence in all nine cases led to 86 successful grasps in 90 trials, or a success rate of $95,6\%$. This clearly shows that also the estimations of the centroids and the main axes of all objects were accurate enough to enable successful grasping of unknown objects. The four error trials were results of either a bug in the system, or the fact that one of the fingers touched the object too early and pushed it away. In order to cope with such situations one could incorporate simple contact-reactive heuristic, such as the one presented in [26]. These results clearly support the hypothesis behind the Mitten-Thought-Experiment that for successful robotic grasping of unknown objects as I investigated in this project, i.e. making contact, picking up, and holding in mid-air, only a compliant grasping mechanism and a strategy to perceive the rough spatial position, the shape, and the size of an object are necessary.

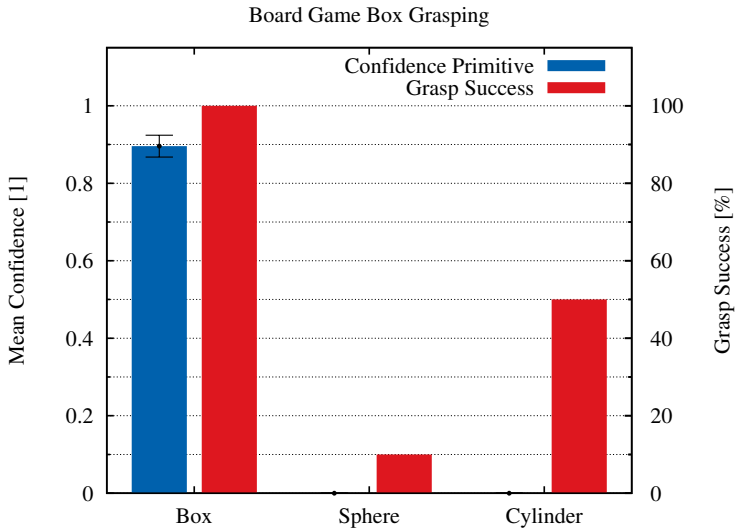


Figure 4.23: Perception and grasping results of the board game box: Perception clearly indicated a box-like with a very small standard deviation. Grasping success of the box pregrasps was superior to that of both the cylinder and the sphere pregrasps.

As for the final question of whether there was a clear correlation between perceived object shape and successful pregrasps, the answer is two-fold. For the first six medium-sized objects there were only small differences in grasp performances between pregrasps that the perception indicated to be the right one and the other two. Here the most-confident hypothesis yielded successful grasps in 58 of 60 attempts (success rate of 96.7%), while the alternatives resulted in 107 successful grasps in 120 trials (success rate 89.2%). Thus, it seems as if the hand was able to generalize over the shape of the objects with arbitrary preshapes as it could completely wrap its fingers around the objects. As a result, wrist orientation and end-effector position of the pregrasp became the single deciding factors for grasping success. For the big objects, the observation was quite different. The pregrasps with the highest shape confidence yielded 28 successes in 30 attempts (success rate of $95,6\%$), while the not so confident hypotheses resulted in only 45 successful grasps for 90 trials (success rate 50%). Apparently, for big objects, i.e. when the fingers are no longer able to encase the object with arbitrary preshapes, a good fit of object shape and hand preshape become a necessity for successful grasping.

Figure 4.24 tries to visualize this correlation between object shape, object size, and successful preshape for grasping in a qualitative way. The objects under consideration are depicted in a polar coordinate system, with the radial axis representing the shape of the object and the distance from the origin showing the overall size

of the object. The colored areas indicate successful grasping performance of the corresponding pregrasp, where red stands for the Sphere Primitive, green for the Box Primitive, and blue for the Cylinder Primitive. Gray areas represent object that cannot be successfully grasped with either of the pregrasps. Obviously, very small objects cannot be grasped because they simply slip through the fingers, while extremely big objects exceed the capabilities of the hand. Then there is an area of medium size, for which several or all pregrasps yield good performance. Finally, for bigger objects hand preshape becomes more and more important as only one of the pregrasps is still able to successfully grasp the object. Please note, that the amount and specific nature of the preshapes that are meaningful for classification of objects are grounded in the capabilities of the hand. For a simpler hand model, for example, it could be useful to divide objects in only two shape categories, while sophisticated hand designs probably also call for more primitive shape categories.

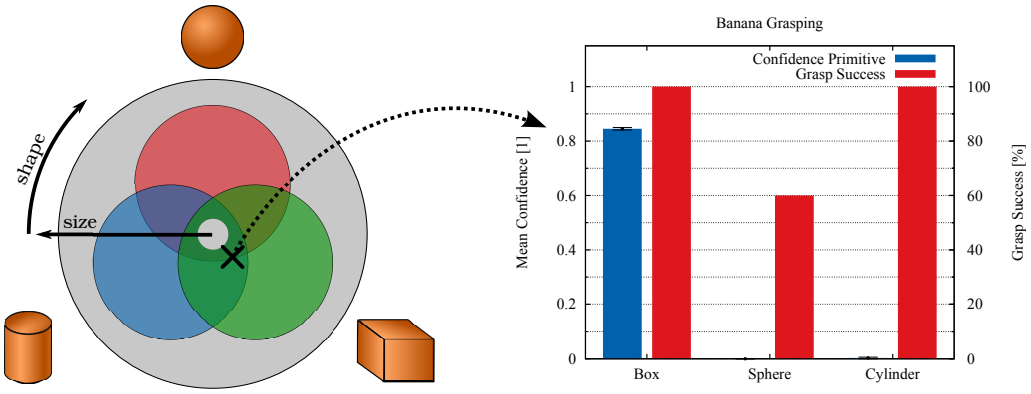


Figure 4.24: Qualitative visualization of the correlation between object shape and size and successful pregrasps: Object shapes are depicted in a polar coordinate system, with the radial component representing object shape and the distance from the origin representing object size. Colored regions indicate successful pregrasps: Red stands for the Sphere Primitive, green for the Box Primitive, and blue for the Cylinder Primitive. Gray regions show objects that cannot be grasped with either of the hypotheses. Obviously, too small and too big objects cannot be grasped at all. Medium-sized objects can be grasped with several successful hypotheses, as the hand is able to completely wrap around the object. For bigger objects this is no longer possible and a correlation between selected preshape and overall object shape becomes apparent. The qualitative location of the results of the banana grasping is depicted to illustrate the interpretation of the image.

In this chapter I performed extensive simulation and real-world experiments to evaluate the performance of the perceptual primitives. Results showed that the primitives were able to reliably estimate the spatial, size and shape properties of very different objects. Successful grasping experiments using the estimated information

confirmed the hypothesis underlying the Mitten-Thought-Experiment: Successful robotic grasping of unknown objects is feasible with a compliant grasping mechanism and a strategy to perceive the rough spatial position, the shape, and the size of an object. Additionally, a correlation between perceived object shape and successful pregrasps was obvious for big objects, while for medium-sized objects no significant performance differences across pregrasps became apparent.

5 Conclusion

5.1 Summary

In this thesis, I considered the problem of robotic grasping of unknown objects. I chose to reduce the problem of finding a good grasp of an object to finding a good pregrasp. During the closing of the robotic hand its compliance would then automatically cause it to fit the shape of the object well, thus leading to stable grasps. This approach was inspired by findings from neuroscientific research which indicate that during grasping humans control the configuration of their hand only in a low-dimensional subspace [15].

In order to determine the correct pregrasp of an object I proposed to use visual primitives. Theses primitives explore an object to estimate its spatial, size, and shape properties. The main assumption behind this is that the perceived properties of an object correspond to the properties of the pregrasp that enables successful grasping of this object. The primitives were designed as active vision modules that can control the exploration motion of the camera. This design choice allowed for fewer assumptions about the object's shape, and thus better capabilities to generalize across different kinds of objects.

Experimental evaluation, which was performed both in simulation and real world, confirmed that the primitives have the ability to reliably estimate the desired information across a huge variety of objects. The pregrasps that were chosen with this information yielded very good grasping performance. Additionally, a strong correlation between the perceived shape of an object and the preshape of a successful pregrasp was found for big objects, while for medium-sized objects such a relationship could not be established.

5.2 Future Work

Several aspects deserve consideration for future work. First of all, one could expand the design of the primitives to also incorporate information from motions that do not correspond to the desired exploration motion. This would greatly increase the robustness of the system against motion errors, and allow for various exploration trajectories if the optimal trajectory is not feasible.

Consideration of further hand preshapes, e.g. the hook and precision tip preshapes

of the BarrettHand, is an interesting avenue for future projects. One could investigate which new perceptual primitives are necessary to fulfill the informational needs of the resulting new pregrasps.

Improving the overall robustness of the perceptual system to enable it to better cope with error cases of real-world scenarios would also be worth spending time on. Error handling for objects that partially or completely leave the picture, or improved blob detection are just two examples that come to mind. Incorporation of further partially redundant image features could increase the convergence speed and improve robustness of the image processing step.

Additionally, one could evaluate the perceptual system with different robotic hands: How does the design of the primitives change to control the new pregrasps? What happens if a hand with even more (or less) compliance is deployed?

Furthermore, it would also be very interesting to extend the system to use continuous hand preshapes. Usage of for example the eigengrasps from Ciocarlie et al. [22] would allow to generate hand shapes that mimic percepts with mixed shape confidences much better than the currently used discrete preshapes. Investigation of the resulting grasping performances could provide more insight into the tight coupling of perception and hand control which is necessary to provide robots with human-like grasping capabilities.

Images and Internet

[1] http://topnews.net.nz/content/23147-elderly-care-sparks-investigation.

[2] http://www.ehow.com/info_12116081_particles-sticking-glasses-dishwasher.html.

[3] http://www.dailymail.co.uk/news/article-1369435/Japan-nuclear-crisis-3-Fukushima-Fifty-suffer-radiation-poisoning.html.

[4] http://spectrum.ieee.org/automaton/robotics/humanoids/dlr-super-robust-robot-hand.

[5] http://www.merriam-webster.com/dictionary/robust.

[6] http://www.ros.org.

[7] http://openrave.programmingvision.com/en/main/index.html.

[8] http://en.wikipedia.org/wiki/File:HSV_color_solid_cylinder_alpha_lowgamma.png.

[9] http://opencv.willowgarage.com/wiki/.

[10] http://code.google.com/p/cvblob/.

[11] http://en.wikipedia.org/wiki/Image_moment.

[12] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT2/node3.html.

[13] http://www.robotics.tu-berlin.de/menue/teaching/.

[14] http://en.wikipedia.org/wiki/File:Coord_system_SE_0.svg.

Bibliography

[15] A. T. Miller, S. Knoop, H. I. Christensen and P. K. Allen. *Automatic Grasp Planning Using Spape Primitives.* IEEE International Conference on Robotics and Automation (ICRA), 1824–1829, 2003.

[16] M. Santello, M. Flanders and J. F. Soechting. *Postural Hand Synergies for Tool Use.* The Journal of Neuroscience, volume 18 (23):10105–10115, 1998.

[17] B. Siciliano and O. Khatib, editors. *Handbook of Robotics.* Springer, 2008.

[18] A. Bicci and V. Kumar. *Robotic Grasping and Contact: A Review.* IEEE International Conference on Robotics and Automation (ICRA), 348–353, 2000.

[19] A. T. Miller and P. K. Allen. *Graspit! A versatile simulator for robotic grasping.* IEEE Robotics & Automation Magazine, volume 11:110–122, 2004.

[20] C. Ferrari and J. Canny. *Planning Optimal Grasps.* IEEE International Conference on Robotics and Automation (ICRA), 1992.

[21] D. Kragic, A. T. Miller and P. K. Allen. *Real-time Tracking Meets Online Grasp Planning.* IEEE International Conference on Robotics and Automation (ICRA), 2460–2465, 2001.

[22] M. Ciocarlie, C. Goldfeder and P. K. Allen. *Dimensionality reduction for hand-independent dexterous robotic grasping.* IEEE International Conference on Intelligent Robots and Systems (IROS), 3270–3275, 2007.

[23] C. Goldfeder, M. Ciocarlie, H. Dang and P. K. Allen. *The Columbia Grasp Database.* IEEE International Conference on Robotics and Automation (ICRA), 1710–1716, 2009.

[24] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang and P. K. Allen. *Data-Driven Grasping with Partial Sensor Data.* IEEE International Conference on Intelligent Robots and Systems (IROS), 1278–1283, 2009.

[25] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu and I. A. Sucan. *Towards Reliable Grasping and Manipulation in Household Environments.* New Delhi, India, 2010.

[26] K. Hsiao, S. Chitta, M. Ciocarlie and E. G. Jones. *Contact-Reactive Grasping of Objects with Partial Shape Information.* IEEE International Conference on Intelligent Robots and Systems (IROS), 1228–1235, 2010.

[27] R. Balasubramanian, L. Xu, P. Brook, J. Smith and Y. Matsuoka. *Human-Guided Grasp Measures Improve Grasp Robustness on a Physical Robot.* IEEE International Conference on Robotics and Automation (ICRA), 2010.

[28] A. Saxena, J. Driemeyer, C. Osondu and A. Y. Ng. *Learning to Grasp Novel Objects Using Vision.* Experimental Robotics, volume 39:33–42, 2008.

[29] A. Saxena, J. Driemeyer and A. Y. Ng. *Robotic Grasping of Novel Objects using Vision.* The International Journal of Robotics Research, volume 27:157–173, 2008.

[30] J. Bohg and D. Kragic. *Learning Grasping Points with Shape Context.* Robotics and Autonomous Systems, volume 58:362–377, 2010.

[31] K. Huebner and D. Kragic. *Selection of Robot Pre-Grasps using Box-Based Shape Approximation.* IEEE International Conference on Intelligent Robots and Systems (IROS), 1765–1770, 2008.

[32] K. Huebner, S. Ruthotto and D. Kragic. *Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping.* IEEE International Conference on Robotics and Automation (ICRA), 1628–1633, 2008.

[33] C. Dune, E. Marchand, C. Collowet and C. Leroux. *Active rough shape estimation of unknown objects.* IEEE International Conference on Robotics and Automation (ICRA), 3622–3627, 2008.

[34] D. Marr. *Vision.* The MIT Press, 2010 (originally published 1982).

[35] Y. Aloimonos and I. Weiss. *Active Vision.* International Journal of Computer Vision, volume 1 (4):333–356, 1988.

[36] F. Chaumette, S. Boukir, P. Bouthemy and D. Juvin. *Structure From Controlled Motion.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 18 (5):492–504, 1996.

[37] S. Chen, Y. Li and N. M. Kwok. *Active vision in robotic systems: A survey of recent developments.* The International Journal of Robotics Research, volume 30 (11):1343–1377, 2011.

[38] B. Calli, M. Wisse and P. Jonker. *Grasping of Unknown Objects via Curvature Maximization using Active Vision.* IEEE International Conference on Intelligent Robots and Systems (IROS), 995–1001, 2011.

[39] M. Gentilucci. *Object motor representation and reaching-grasping control.* Neuropsychologia, volume 40 (8):1139–1153, 2002.

[40] R. H. Cuijpers, J. B. J. Smeets and E. Brenner. *On the Relation Between Object Shape and Grasping Kinematics.* Journal of Neurophysiology, volume 91 (6):2598–2606, 2004.

[41] M. Ajay, Y. Aloimonos and C. L. Fah. *Active segmentation with fixation.* 2009 IEEE 12th International Conference on Computer Vision, 468–475, 2009.

[42] G. Bradski and A. Kaehler. *Learning OpenCV - Computer Vision with the OpenCV Library.* O'Reilly Media, Inc., 1 edition, 2008.

[43] G. J. Garcia, C. A. Jara, J. Pomares and F. Torres. *Direct visual servo control of a robot to track trajectories in supervision tasks.* 2010 11th International Conference on Control Automation Robotics & Vision, 1434–1439, 2010.

[44] F. Chaumette and S. Hutchinson. *Visual Servo Control, Part I: Basic Approaches.* IEEE Robotics and Automation Magazine, volume 13 (4):82–90, 2006.