



**KTH Computer Science  
and Communication**

# **Learning to Assess Grasp Stability from Vision, Touch and Proprioception**

**YASEMİN BEKİROĞLU**

Doctoral Thesis  
Stockholm, Sweden 2012

TRITA-CSC-A 2012:12

ISSN-1653-5723

ISRN-KTH/CSC/A-12/12-SE

ISBN 978-91-7501-522-4

KTH School of Computer Science and Communication

SE-100 44 Stockholm

SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi onsdagen den 14 november 2012 klockan 10.00 i Sal F3, Kungl Tekniska högskolan, Lindstedtsvägen 26, Stockholm.

© Yasemin Bekiroğlu, November 2012

Tryck: Universitetsservice US AB

## Abstract

Grasping and manipulation of objects is an integral part of a robot's physical interaction with the environment. In order to cope with real-world situations, sensor based grasping of objects and grasp stability estimation is an important skill. This thesis addresses the problem of predicting the stability of a grasp from the perceptions available to a robot once fingers close around the object before attempting to lift it. A re-grasping step can be triggered if an unstable grasp is identified. The percepts considered consist of object features (visual), gripper configurations (proprioceptive) and tactile imprints (haptic) when fingers contact the object. This thesis studies tactile based stability estimation by applying machine learning methods such as Hidden Markov Models. An approach to integrate visual and tactile feedback is also introduced to further improve the predictions of grasp stability, using Kernel Logistic Regression models.

Like humans, robots are expected to grasp and manipulate objects in a goal-oriented manner. In other words, objects should be grasped so to afford subsequent actions: if I am to hammer a nail, the hammer should be grasped so to afford hammering. Most of the work on grasping commonly addresses only the problem of finding a stable grasp without considering the task/action a robot is supposed to fulfill with an object. This thesis also studies grasp stability assessment in a task-oriented way based on a generative approach using probabilistic graphical models, Bayesian Networks. We integrate high-level task information introduced by a teacher in a supervised setting with low-level stability requirements acquired through a robot's exploration. The graphical model is used to encode probabilistic relationships between tasks and sensory data (visual, tactile and proprioceptive). The generative modeling approach enables inference of appropriate grasping configurations, as well as prediction of grasp stability. Overall, results indicate that the idea of exploiting learning approaches for grasp stability assessment is applicable in realistic scenarios.

## Acknowledgement

This thesis was carried out at the Computer Vision and Active Perception (CVAP) lab at the Royal Institute of Technology (KTH) in Stockholm, Sweden, from 2008-2012, and was supported by the EU project CogX (FP7-ICT-215181): Cognitive Systems that Self-Understand and Self-Extend.

There are many people who greatly influenced this work. First of all I would like to express my deepest gratitude to my supervisor Danica Kragic for giving me the opportunity to pursue my Phd under her guidance, support and encouragement, and for always making the time for a discussion.

Special thanks to my collaborators Niklas Bergström, Jimmy Alison Jørgensen, Ville Kyrki, Janne Laaksonen, Kai Hübner, Renaud Detry, Dan Song, and Lu Wang for their contributions. Thanks to Anaïs Peyrucq and Lu for their help with data collection and to Xavi Gratal, Niklas, Javier Romero, Jeannette Bohg, Renaud and Virgile Högman for their help with hardware and software problems. I would also like to thank David House, Renaud, Dan, Florian Pokorny, Carl Henrik Ek and Niklas for providing valuable review comments and suggestions during the writing of this thesis.

Thanks to everyone at CVAP (past and present): Alessandro Pieropan, Alper Aydemir, Andrzej Pronobis, Babak Rasolzadeh, Carl Henrik, Cheng Zhang, Christian Smith, Dan, Elin Anna Topp, Florian, Francisco Vina, Friné Portal, Gert Kootstra, Hedvig Kjellström, Heydar Maboudi, Hossein Azizpour, Ioannis Karayiannidis, Jan-Olof Eklundh, Jeanna Ayoubi, Jeannette, Javier, Johan Ekekrantz, Johannes Stork, John Folkesson, Josephine Sullivan, Kai, Kaiyu Hang, Kristoffer Sjöo, Lazaros Nalpantidis, Magnus Burénus, Maria Ralph, Marianna Madry, Marin Saric, Martin Hjelm, Matthew Johnson-Roberson, Miroslav Kobetski, Mårten Björkman, Niklas, Omid Aghazadeh, Oscar Danielsson, Oskar Linde, Patric Jensfelt, Renaud, Stefan Carlsson, Vahid Kazemi, Virgile, Xavi and Yuquan Wang, for creating a friendly and inspiring atmosphere.

It has been a pleasure to spend time at CVAP with all my colleagues. In particular, I want to thank Niklas, Maria, Alper, Renaud, Dan, and Kai for our many long research discussions. Thanks to Marianna for being a great office roommate. Thanks to Dan for not leaving me alone during conference trips, lab visits and lunches. Furthermore I want to thank our administrators Friné and Jeanna for always being kind and helpful.

Last, but not least, I want to thank my parents and brothers for their endless support, love and care. I am also thankful to Maria who has been a big sister to me from the first moment we met, and to Dan, Kai, Deniz, Selda, Hatice, Züleyha, Gülay and Diala for their great friendship and for being a constant source of encouragement.

# Contents

|          |                                                                       |           |
|----------|-----------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                   | <b>1</b>  |
| 1.1      | Challenges in Grasping . . . . .                                      | 2         |
| 1.2      | Contributions . . . . .                                               | 5         |
| 1.3      | Outline . . . . .                                                     | 6         |
| 1.4      | Publications . . . . .                                                | 7         |
| <b>2</b> | <b>Background and Related Work</b>                                    | <b>9</b>  |
| 2.1      | Grasping Approaches . . . . .                                         | 9         |
| 2.2      | Uses of Tactile Sensing . . . . .                                     | 13        |
| 2.3      | Our Approach . . . . .                                                | 14        |
| <b>3</b> | <b>Learning Grasp Stability through Tactile Sensing</b>               | <b>19</b> |
| 3.1      | Classification Approach . . . . .                                     | 21        |
| 3.2      | Data Acquisition . . . . .                                            | 25        |
| 3.3      | Experimental Evaluation . . . . .                                     | 29        |
| 3.4      | Integrating Grasp Planning with Online Stability Assessment . . . . . | 39        |
| 3.5      | Summary and Discussion . . . . .                                      | 47        |
| <b>4</b> | <b>Learning Grasp Stability through Visual and Tactile Sensing</b>    | <b>49</b> |
| 4.1      | Background and Motivation . . . . .                                   | 49        |
| 4.2      | Modelling Framework . . . . .                                         | 50        |
| 4.3      | Experimental Evaluation . . . . .                                     | 52        |
| 4.4      | Summary and Discussion . . . . .                                      | 66        |
| <b>5</b> | <b>Task-oriented Grasp Stability Assessment</b>                       | <b>69</b> |
| 5.1      | Bayesian Network . . . . .                                            | 70        |
| 5.2      | Modeling Sensor Data and Data Acquisition . . . . .                   | 71        |
| 5.3      | Model Selection . . . . .                                             | 73        |
| 5.4      | Model Evaluation . . . . .                                            | 77        |
| 5.5      | Summary and Discussion . . . . .                                      | 85        |
| <b>6</b> | <b>Conclusions</b>                                                    | <b>87</b> |
| 6.1      | Learning to Assess Grasp Stability . . . . .                          | 88        |

|                           |           |
|---------------------------|-----------|
| 6.2 Future Work . . . . . | 90        |
| <b>Bibliography</b>       | <b>91</b> |

# Chapter 1

## Introduction

Humans interact with the environment using rich sensory information. For interaction with an object, we integrate vision, proprioception and tactile sensing almost without thinking. For robots, we envision no less: one of the key requirements for a robot is to physically interact with the environment to execute useful tasks such as cleaning, sorting etc. A man-made environment is challenging for a robot due to the constant change in type and placement of objects it needs to interact with. Apart from the ability to detect and recognize objects, a robot should have some general object grasping and manipulation capabilities allowing for physical interaction. How advanced these capabilities are depends on the available sensory information as well as the mechanical and dynamic properties of the robot.

A successful grasp is often described as a relationship between an object and a robot hand/gripper that allows for some subsequent manipulation of the object such as, for example, placing an object somewhere else or using it for accomplishing a certain action such as hammering. Another example depicted in Figure 1.1 may be pouring. Apart from grasping, the manipulation process includes several other steps. The first step is to perform some form of reconstruction of the environment, commonly done by visual exploration. The robot starts to search for a mug which may require a process of segmentation in order to generate object hypotheses. Once the mug is detected, the robot needs to estimate its pose (position and orientation) in order to be able to plan the approach sequence. If the object is not within reach, it needs to plan how to get to it and move while avoiding the obstacles. Then, the robot needs to plan where to put fingers, i.e., solve the grasp planning problem. Since there are infinitely many grasps it can apply on the mug, it may be looking for one that also affords the tasks it is supposed to execute, namely pouring. In order to be able to do this, it needs to integrate the available visual information with some stored knowledge of grasps that afford pouring. The next step is grasp execution: placing the fingers on the mug by moving the hand and fingers so to achieve a contact with enough friction to hold the mug and resist sliding. The robot also needs to close the fingers around the mug with the right amount of force without breaking the mug. In this last step, sensory information (e.g., visual and/or force feedback) can be used to assure that the goal

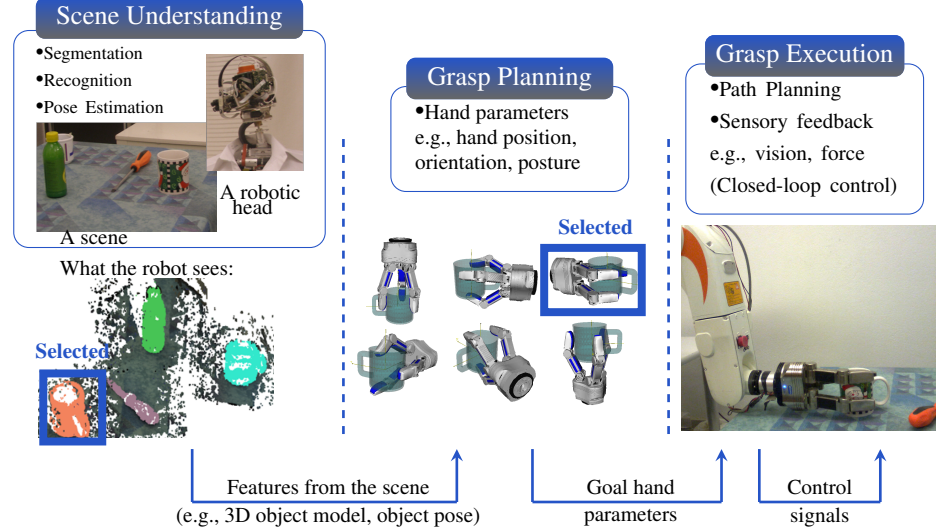


Figure 1.1: Grasping process with example systems in each step. Scene understanding is commonly based on noisy and incomplete visual data. Given the perceived object features from the scene and the knowledge of the task, a hand configuration is chosen among an infinite number of candidate hypotheses. Execution of the selected grasp is performed using sensory feedback.

is achieved. Although all the individual steps have been studied in depth during the last few decades [18, 21, 29, 77, 86, 92, 94, 96, 100, 103], a robust and general approach to grasping for wide variety of tasks and objects encountered in dynamic and unstructured environments and novel situations, which is close to human grasping skills does not exist yet. Current systems have severe limitations in terms of dealing with novelty, uncertainty and unforeseen situations.

## 1.1 Challenges in Grasping

There are multiple reasons that make the process of autonomous grasping challenging: Object properties required for grasp planning such as shape and pose are commonly not known a priori and sensory information used to extract this information from the environment, e.g., vision, is prone to error. Processes prior to grasping such as scene segmentation, object recognition and pose estimation systems are not perfectly accurate due to issues such as occlusions and noisy measurements as also seen in the example in Figure 1.1. To estimate the shape and pose of an object, visual sensing has been widely used [96, 34, 56, 91, 87, 23]. However, the accuracy of vision in terms of object pose estimation is limited even for known objects. Therefore small errors in object pose are common and these may cause failures in grasping. These failures are commonly difficult to prevent

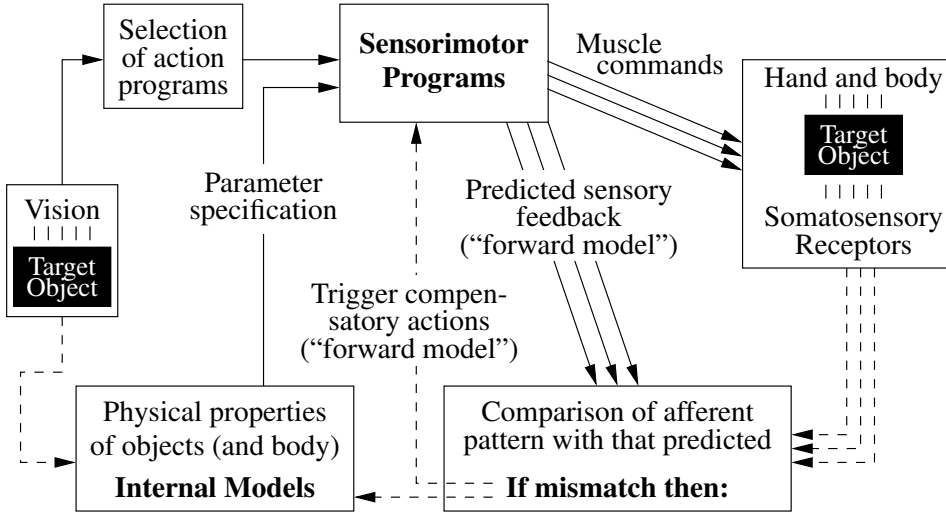


Figure 1.2: Reproduction of “Predictive ‘feed-forward’ sensory control of manipulation”, from R. Johansson [61], Figure 5, p. 56. In Johansson’s work [61], the sensorimotor planning and execution of manipulative tasks is formalized with a closed-loop system which integrates dynamic touch and visual perception with sensorimotor memories. A manipulative action is planned from visual input, which triggers an appropriate learned action program. As motor commands are being issued, sensor signals are continuously compared to values predicted by an internal forward model, which permits the detection of unexpected events. In turn, unexpected events trigger recovery procedures and the refinement of the forward model.

at the grasp execution stage if the hand is not equipped with sensors. In terms of object shape approximation, algorithms are far from approximating objects at an accuracy that makes the necessary parameters comparable to their real values. Therefore, noisy and incomplete visual data is used for grasp planning [21, 86, 94]. Planning strategy may rely on the agent’s grasping knowledge, e.g., in the form of memories of previously-executed grasps, or in the form of built-in strategies. To select a suitable grasp is a major challenge, since the number of potentially applicable grasps for a given object is huge and there are many factors that influence the decision on selection, such as the physical properties of the object, embodiment of the robot or the task. The agent must *adapt* its expertise to find a plan that best fits the current world configuration. Because of the uncertainty inherent to all these processes, designing grasp plans that are guaranteed to work in an open-loop system is difficult. Grasp execution can thus greatly benefit from a closed-loop controller which considers sensory feedback while issuing motor commands.

Humans make extensive use of input from several sensor modalities when executing grasps [60, 58]. Vision is one of the modalities which contribute substantially to grasp control and stability [114, 76, 48]. Touch is another one, as supported by numer-

ous studies which show the influence of tactile feedback on different grasp sub-processes [60, 58, 62, 61, 68]. For example, Johansson and Westling [62] have shown that anesthetizing a subject's fingers – thereby impairing his sense of touch while leaving his motor capabilities intact – directly leads to a loss in the subject's proficiency in grasping and lifting objects. These observations are reflected in the work of Johansson et al. [60, 62, 61] who presented an empirical formalization of the human grasping behavior as a closed-loop system involving visual and touch feedback and a memory-based controller, see Figure 1.2. A key part of this work emphasized humans' ability to predict the repercussion of manipulative actions onto sensory channels by means of a (learned) forward model, thereby allowing us to react to unexpected situation and maintain grasp stability.

In robotics, vision-driven grasping and manipulation have been extensively studied [117, 67]. Vision has typically been used to plan grasping actions, and to update action parameters as objects move to compensate for manipulator positioning inaccuracies and sensor noise. However, most vision based approaches have been used only for objects known to the robot prior to task execution, since they commonly need a desired pose with respect to the object to be defined beforehand, which is not easy for unknown objects. Touch-based grasp controllers have also been studied, with emphasis on designing programs for controlling finger forces to avoid slippage and to prevent crushing objects [20, 51, 50, 92]. Although tactile and force sensors can be used to reduce the uncertainty upon contact, a major issue still remains that grasps need to be evaluated for the target task from the data the robot can extract on-line in realistic applications. A grasp may fail even when all fingers have adequate contact forces for lifting and the hand pose is not much different from the planned one during the subsequent manipulation required for the task, e.g., due to unexpected reasons such as uneven weight distribution, slippery material or occluded object parts. Besides incomplete information about the environment and the objects, there is also a lack of generalizable quality measures for grasp stability assessment under uncertainty.

Grasps are executed in order to complete a specific task. Each task imposes its own constraints that limit the range of applicable grasps. These constraints affect the geometry of the grasp (a cup is grasped differently if we want to drink from it or simply put it away) and also the robustness we expect from the grasp (if we plan to shake a bottle very hard, we grasp it more carefully than if we plan to throw the bottle away). Therefore during the grasp selection process the task should be considered. A lot of current work in robotics is inspired by human goal-directed behavior [73]. In humans, goal-directedness is obtained through multiple development stages, both through the sensorimotor *exploration* (trial and error) and through the *observation* of others interacting with the world (imitation learning) [90]. The former is addressing the problem of learning through self-experience in order to associate the sensorimotor signals to the direct motor effects. The latter involves human supervision, which is especially beneficial for efficient learning of complex tasks. Robotic approaches often focus on just one of these two aspects. Linking between the two is often through manual encoding [115] or applied to simple tasks [108, 82, 90, 78]. The main challenges originate from the differences in commonly adopted representations [69]. The gap between the representations is especially visible when dealing with robot grasping tasks. For example, if a robot is given a high-level task command, e.g., *pour me a cup of coffee*, it needs to make decision on which object to use, how the hand should be placed around the

object, and how much gripping force should be applied so that the subsequent manipulation is stable. Several sensory streams (vision, proprioception and tactile) are relevant for manipulation. The problem domain and hence the state space becomes high-dimensional, involving both continuous and discrete variables with complex relations. Traditional dynamic systems approaches in robotics e.g., [57] focus mainly on optimal planning and control of hand trajectories, hence the state space only includes kinematic parameters of the arm and hand. The relations between many grasping-relevant variables mentioned above can not be addressed simultaneously. Some recent work in the area [103] linked the grasp plan to the manipulation tasks through probabilistic graphical models that can deal with high dimensional complex problem domains. The work emphasized the geometric constraint of a task for planning grasps based on simulated vision inputs. Tasks, however, also require various manipulations: *pouring* requires rotating a bottle that contains liquid, and a *hand-over* task requires only parallel transportation. The stability demands therefore differ due to different manipulations required by tasks and need to be taken into account in the planning process.

## 1.2 Contributions

In this thesis we study the exploitation of multisensory feedback (tactile, visual and proprioceptive) for modelling grasps and assessing grasp stability. The main contribution is a new approach that incorporates knowledge of uncertainty in the observations when predicting the stability of a grasp. This strategy also allows for detection of failures at the grasp execution stage. If the stability estimate is too low, the agent may decide to move the manipulator to a better configuration before lifting the object. Three aspects in predicting grasp stability has been studied:

- Grasp stability prediction based on tactile sensing has been studied and a learning system that infers grasp stability from tactile feedback is presented.
- Visual feedback has been included in the system and means of *learning* some of the models and sensorimotor programs which contribute to the system depicted in Figure 1.2 are also discussed. By observing the sensor signals issued during the execution of grasps given by a human, our agent learns what it feels like to grasp an object from a specific side, and learns which grasping configurations lead to a stable grasp. When planning a grasp, the agent is able to compute an initial estimate of the stability of the planned grasp. As the grasp is being executed and the manipulator's fingers are brought around the object, the pose (3D position and 3D orientation) of the object is continuously tracked. When fingers come in contact with the object, afferent tactile signals are compared to the signals predicted by the learned feed-forward model for the current object-gripper configuration, yielding an updated estimate of the stability of the grasp. In a learning scenario, the agent can then proceed with an attempt to transport and shake the object to gather an empirical confirmation of its stability assessment, possibly updating the feed-forward model.

- Semantic task information is also integrated with stability assessment. A method combining exploration and supervision is implemented, where exploration enables the robot to learn about its own sensorimotor ability (how to grasp an object to stably lift and manipulate it), while human tutoring helps the robot to associate its sensorimotor ability to high-level goals.

### 1.3 Outline

This thesis is structured in the following way:

#### **Chapter 2 - Background and Related Work**

In this chapter the contributions are discussed in detail in relation to the state-of-the-art work in the area and the approach is introduced along with the used robot platform.

#### **Chapter 3 - Learning Grasp Stability through Tactile Sensing**

In this chapter methods for assessing grasp stability based on haptic data and machine learning methods, including AdaBoost, Support Vector Machines and hidden Markov models are presented. Experiments with AdaBoost and Support Vector Machines were designed in collaboration with Janne Laaksonen and run by him. Some of the data used in the experiments were generated in the simulator RobWorkSim by Jimmy A. Jørgensen. This chapter also presents how a grasp planner can be integrated with a probabilistic technique for grasp stability assessment in order to improve the hypotheses about suitable grasps on different types of objects. The grasp hypotheses were generated by Kai Huebner using the grasp planner. This work has been published in [13], [11], [12], [14].

#### **Chapter 4 - Learning Grasp Stability through Visual and Tactile Sensing**

In this chapter, we integrate vision-based pose information into the model to further disambiguate grasp configurations. The whole system was designed in collaboration with Renaud Detry. This work has been published in [9], [10], [8].

#### **Chapter 5 - Task-oriented Grasp Stability Assessment**

In this chapter, a probabilistic framework for grasp modeling and stability assessment is presented. The framework facilitates assessment of grasp success in a goal-oriented way, taking into account both geometric constraints for task affordances and stability requirements specific for a task. The bayesian network framework for grasp modeling introduced by Dan Song has been extended to implement task based stability estimation and the classification experiments were designed and run in collaboration with her. Real data collection was done in collaboration with Lu Wang. This work has been presented in [15] and [16].

## Chapter 6 - Conclusions

In this final chapter, the experiments and results presented in this thesis are summarized and conclusions are drawn.

## 1.4 Publications

Most of the results presented in this thesis have appeared previously in the following publications:

### Journals

[13] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki and D. Kragic. Assessing grasp stability based on learning and haptic data. *IEEE Transactions on Robotics*, 27(3):616-629, 2011.

### Conferences

[9] Y. Bekiroglu, R. Detry and D. Kragic. Learning Tactile Characterizations of Object-And Pose-specific Grasps. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, USA, 2011.

[11] Y. Bekiroglu, K. Hübner and D. Kragic. Integrating Grasp Planning with Online Stability Assessment using Tactile Sensing. *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[12] Y. Bekiroglu, V. Kyrki and D. Kragic. Learning grasp stability with tactile data and HMMs. *IEEE International Symposium on Robot and Human Interactive Communication*, Viareggio, Italy, 2010.

### Workshops

[10] Y. Bekiroglu, R. Detry, D. Kragic. Grasp Stability from Vision and Touch. *IEEE IROS 2012 Workshop: Advances in Tactile Sensing and Touch-based Human-Robot Interaction*, Vilamoura, Portugal, 2012.

[15] Y. Bekiroglu, D. Song, L. Wang, D. Kragic. Learning Task- and Touch-based Grasping. *IEEE IROS 2012 Workshop: Beyond Robot Grasping - Modern Approaches for Dynamic Manipulation*, Vilamoura, Portugal, 2012.

[8] Y. Bekiroglu, R. Detry and D. Kragic. Joint Observation of Object Pose and Tactile Imprints for Online Grasp Stability Assessment. *IEEE ICRA 2011 workshop: Manipulation Under Uncertainty*, Shanghai, China, 2011.

[14] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki and D. Kragic. Learning grasp

stability based on haptic data. RSS 2010 workshop: Representations for object grasping and manipulation in single and dual arm tasks, Zaragoza, Spain, 2010.

Recent results are currently under review for publication in the following contribution:

[16] Y. Bekiroglu, D. Song, L. Wang, D. Kragic. A Probabilistic Framework for Task-Oriented Grasp Stability Assessment. IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 2013.

## Chapter 2

# Background and Related Work

In this chapter we first give a brief overview of robotic grasping approaches and how sensory information has been used during grasping and manipulation. We then introduce our approach and our robot platform.

### 2.1 Grasping Approaches

Planning and executing a grasp that is robust and stable is an important topic in grasp research [95, 101]. Grasp stability analysis is a tool often used in grasp planning. Grasp planning methods mostly rely on grasp quality measures derived from stability analysis. Most of the work on grasp stability assessment relies on analytical methods and focuses on rigid objects, albeit some work has considered the analysis of grasps on deformable objects [113]. The quality measures of stability [38] are mostly based on the concept of *force-closure*. A force-closure grasp means that any disturbing external forces can be balanced by the forces applied at the contacts. However, these analytical methods require perfect knowledge about the contacts between the hand and the object to estimate the stability of a grasp, which is usually an unrealistic demand on real setups. On the other hand, they enabled simulation of grasps and evaluation of grasp planners on a measurable basis. Thus, most state-of-the-art simulation environments for grasping, e.g., GraspIt! [74], RobWorkSim [63] or OpenGrasp [1], employ analytical measures to compute grasp stability in simulated scenarios. The possibility of performing extensive and efficient experiments in simulation alleviates the maintenance of expensive equipment. It also provides information about the necessary parameters for stability analysis, like contact surfaces, center of mass, or friction coefficients. In unstructured real-world environments, however, these parameters are uncertain, which presents a great challenge to the aforementioned approaches.

Selecting a good grasp for a given object is difficult, since the space of possible grasps is too large to search exhaustively. In order to prune the search space and generate a number of grasp candidates that are likely to be of good quality, various methods have been proposed. One approach that drastically reduces the search space of possible grasps is to approximate 3D object models with a number of primitives such as spheres, cones,

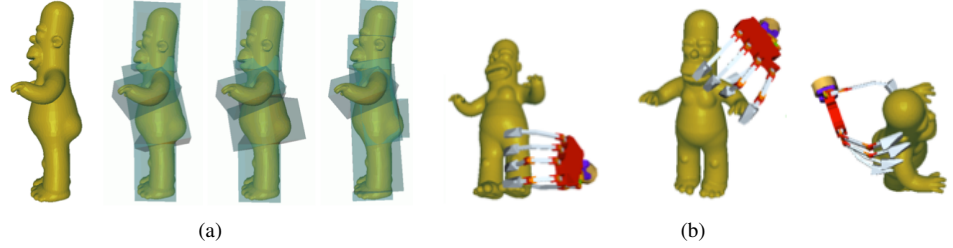


Figure 2.1: Grasp planning in [55]: (a) An object decomposition into boxes. The grasps are generated on the resulting boxes by approaching the boxes along the surface normals and closing the fingers. (b) Some of the resulting grasps.

cylinders, boxes [75, 55], or superquadrics [45] and plan grasps on the approximated shape rather than the actual object. Planning a grasp means determining grasp parameters such as hand approach direction, a target point on the object surface, roll angle of the hand about the approach direction and hand preshape. Then the next step is to simulate the planned grasps on the actual 3D model to acquire contacts and therefore evaluate grasps to choose the most successful grasp(s). These methods define heuristic rules to generate candidate grasps on the resulting object parts. Example results from the system introduced in [55] can be seen in Figure 2.1

Another way to reduce the space of possible grasps is to randomly generate grasp candidates as a set of contacts on the surface of the object [25] then use a heuristic filter to reduce the number of non force-closure candidates. As a final step in all these approaches, among the resulting candidates the most stable set of grasp hypotheses are selected according to an analytical quality metric. Using pre-grasp postures is another solution that limits the large number of possible hand configurations. Ciocarlie et al. [32] presented that stable grasp search can be performed in a low-dimensional subspace of the actual hand configuration space, which is determined by a number of basis vectors called eigengrasps. The method was tested using a grasping simulator and although the eigengrasp dimensionality reduction did not make assumptions about the object, cost functions used in optimization of grasp parameters required a 3D model. Generating grasp candidates based on observing humans grasping an object has also been studied in the literature. Romero et al. [93] presented a human-to-robot grasp mapping system where a human interacting with an object was observed visually based on a single image. The system recognized the human hand posture, including both the grasp type and hand orientation by comparison to a large database of grasp images and mapped the hand posture to robotic hands following a set of rules to grasp the same object in the same pose on a real setup. An example grasp execution from this system can be seen in Figure 2.2.

Exploration-based approaches where the robot learns to grasp by trial and error were also introduced. Detry et al. [35] followed such a strategy and through experience represented object-relative gripper poses composed of 3D position and orientation with object-

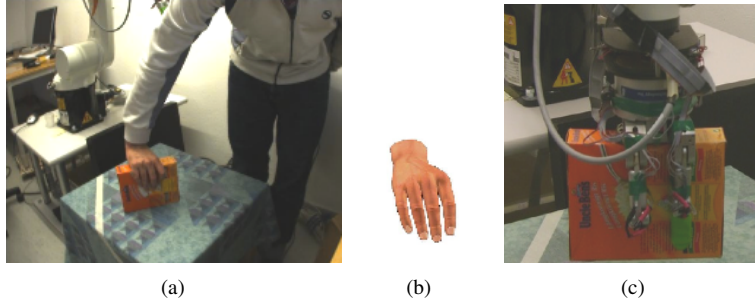


Figure 2.2: An example grasp execution from [93] where the robot processes (a) the image, retrieves (b) the nearest neighbor to the observed grasp from the database and (c) applies the mapped grasp.

specific continuous probability density functions referred to as grasp densities. These grasp densities modeled likelihood of success of a grasp. Grasp densities were learned and refined based on the outcomes of a set of grasps executed on a real setup. Grasps were marked as successful if objects could be lifted. Initial grasp densities were bootstrapped from visual cues that correspond to a 3D object-edge reconstructions, yielding bias towards edge grasps. The resulting densities were then used to sample grasp hypotheses.

The approaches presented so far generates grasp hypotheses for known objects. Various methods have also been proposed to grasp novel objects. Dunes et al. [37] approximated the object shape by a quadric surface whose parameters were estimated based on multiple views of the scene. With a quadric that best fitted the object, basic features that could be used to parameterize grasps such as the object major axis, its centroid position and its rough size were obtained. [24] et al. used a wrist-mounted camera and line laser to model the entire surface. The robot arm was moved to scan the laser across the object while the camera acquires images from multiple viewpoints. Object silhouettes were extracted from the images and used to form a 3D solid model of the object and the model was refined with the scanning results. Their grasp planning algorithm was designed for a parallel-jaw gripper and analyzed the resulting model to output the required gripper position and orientation by searching for a pair of nearly flat surface patches where the gripper jaws should squeeze the object.

Instead of approximating the object shape or reconstructing the 3D model, some approaches aim at finding a mapping from low-level visual features extracted from objects to grasp hypotheses, then select among them. Kraft et al. [65] used a stereo camera to obtain a model that includes local contour descriptors. They defined four elementary grasping actions and associated them to these features based on heuristics. Bergström et al. [17] fitted planes to the reconstructed contours of objects and used the normal of the plane as the grasp approach direction. An example grasp execution with this system can be seen in Figure 2.3. Another way is to grasp novel objects based on experience. Novel objects that

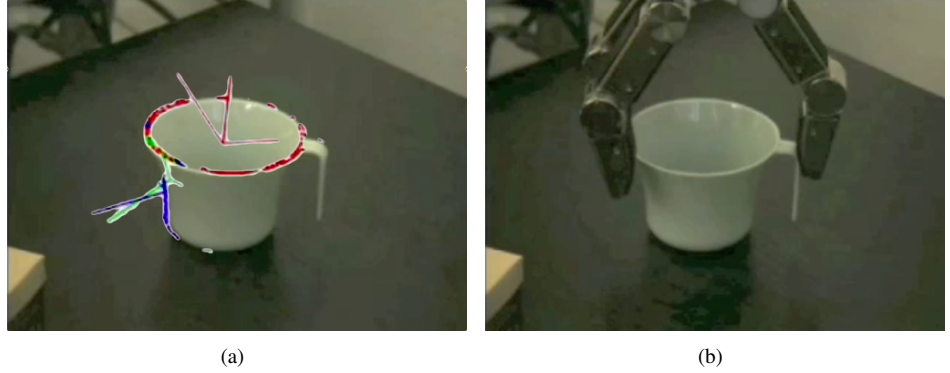


Figure 2.3: An example grasp generation and execution from [17]. (a) The grasp hypothesis depicted with the red lines on top of the object. The grasp hypothesis includes the approach direction and the orientation of the hand. When applying the grasp the fingers close until contact. The approach direction is the normal of the plane which is detected among the reconstructed object contours, while the orientation is determined by the minor axis of the points residing in that plane. (b) The robot executing the grasp.

are found to be similar to the known ones can be grasped in a similar way that is applied to the known ones. Saxena et al. [96] introduced a system that can infer grasping locations on an object from its images. The system was trained using images of objects labeled with grasping regions, such as the center region of the handle for a mug. They demonstrated that a wide variety of objects that were not in the training set could be grasped with their approach.

A good grasp should not only be stable, it also needs to be suitable for the task, i.e., *what do you want to do after you lift the object*. Very few work has put effort on planning grasps in a goal-directed manner. [115] manually encoded the expertise about task semantics provided by a human tutor. A recent work [103] used Bayesian networks to learn the grasping task constraints that depends on a set of geometric attributes from both objects and grasps (e.g., hand positions). However manipulation tasks do not just concern geometric constraints. A *pouring* task not only requires the bottle opening to be unblocked, but also needs the grasp to be stable enough to rotate the bottle. We need to link task information with stability in real world scenarios. A natural extension is to combine supervised task learning with experience-based stability learning. This allows stability to be assessed in a task-oriented manner. This is especially beneficial for energy-efficient control: when a task (e.g., *hand-over*) does not require strong grasping for difficult manipulations (e.g., waving for the *hammering* task), a relatively smaller gripping force can be applied. Combining task with stability was rarely studied. Some work [4, 72] defined task-related grasp quality measures which combined task knowledge with analytical stability measures used in traditional grasp stability studies. Such approaches therefore also suffer from partial and

uncertain knowledge of the world in real setups.

## 2.2 Uses of Tactile Sensing

Tactile sensing has been used for various purposes in prior studies. Learning approaches using tactile sensors were proposed to determine the properties of objects [84, 59, 30] and to recognize objects [30, 98, 97, 46]. Different properties of objects give valuable information that can be further used in grasp stability analysis. In [84], the pose of the object was determined using a particle filter technique based on the tactile information gained from the contacts between a gripper and the object. Similar work was presented by Hsiao et al. [52] where object localization was performed with knowledge of tactile contacts on specific objects. In [59], the surface type (edge, flat, cylindrical, sphere) of the tactile contact was determined using a neural network. In [30], tactile information extracted from the sensors on a two fingered gripper was used to determine the deformation properties of an object.

For recognition of manipulated objects based on tactile sensing, one way is to use multiple grasp or manipulation attempts and then learn the object through the haptic input from the manipulations or grasps. For that purpose, one shot data from the end of the grasps [97, 46] or temporal data collected throughout the grasp or manipulation execution [30, 98] can be used. In [97], a bag-of-words approach was presented which aimed to identify objects using touch sensors available on a two fingered gripper. The approach processed tactile images collected by grasping objects at different heights. In [46], a similar approach was taken for a humanoid hand. A more traditional approach to learning was employed with features extracted from tactile images in conjunction with hand joint configurations as input data for the object classifier. In [98] entropy was used to study the performance of various features in order to determine the most useful features in recognizing objects. In this case, a plate covered with tactile sensor was used as the manipulator. The object recognition using the recognized good features did not perform as well as in the other presented works.

Tactile sensing has also been used to supplement vision for object recognition [6, 27], exploration [104] or localization [26]. Combining visual and tactile sensing has been studied for manipulation tasks by employing visual sensors to obtain global information to reach the object and tactile sensors to provide local information to grasp and manipulate it. In other words, visual input is used for grasp generation/planning and tactile sensing is used for closed loop control once in contact with the object. Son et al. [102] performed experiments by using visual feedback to perform rough positioning of the hand and tactile feedback to detect contact, to compensate for the difference between the orientation of the object and the gripper and to control grasp force for delicate manipulation tasks. Kragic et al. [66] followed a similar strategy and used tactile feedback to compensate for the imperfect vision-based pose estimate by centering the object inside the hand after having a contact on the tactile sensors. The use of tactile sensors was proposed to maximize the contact surface for removing a book from a bookshelf in [79]. Application of force, visual and tactile feedback to open a sliding door was proposed by Prats et al. [88] where tac-

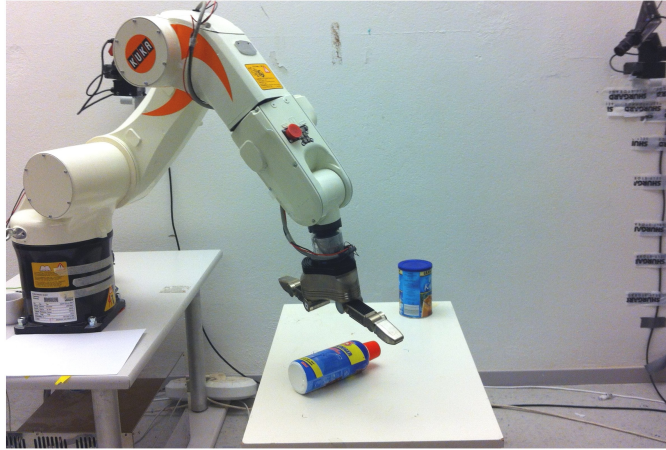


Figure 2.4: Experimental robot platform, composed of an industrial Kuka arm, a three-finger Schunk Dextrous hand equipped with tactile sensing arrays, and a monocular camera.

tile control ensured that an accurate alignment between the hand and the handle was kept. Allen et al. [7] used visual sensing and tactile sensing to calculate the position of contact along a finger in order to estimate applied finger forces for grasping tasks.

### 2.3 Our Approach

Differently from the aforementioned approaches in the previous section, this thesis proposes methods to perform prediction of grasp stability from real world sensory feedback. The aim is to make predictions on the stability of a grasp from the perceptions available to a robot, before attempting to lift the object. If its stability estimate is too low, the agent can for instance decide to back off and make another attempt, or possibly search locally for a more efficient grasp.

On a real world robot platform all measurements acquired from the environment are noisy and associated with a degree of uncertainty. In order for the system to be robust, the uncertainty in the observations needs to be taken into account. Probabilistic methods provide a framework for dealing with uncertainty in a principled manner and will to this end provide the foundation that our system is built upon. The aim is to model the embodiment specific and inherently complex relationship between grasp stability and the available sensory and proprioceptive information. We will first introduce our robot platform then how we approached the stability estimation problem.

### 2.3.1 Our Robot

Our robot platform seen in Figure 2.4 is composed of an industrial Kuka arm with 6 degrees of freedom that is mounted on a robust shelf, a three-finger Schunk Dextrous hand (SDH) with 7 degrees of freedom (Figure 2.5c) and a monocular camera that is used for pose tracking. Our robot platform implements an inverse-kinematics path planning given the target hand pose.

In our experiments, after preshaping the hand, grasping is run by simultaneously closing the fingers and applying constant closing torques on all joints. During grasping, the robot is also able to track the pose of an object despite object occlusions (Figure 2.6). We use a system which tracks the pose of a textured CAD model in a monocular video stream [80]. Tracking object textures greatly helps handling partial object occlusions and distractions induced by the hand. The robot can acquire tactile imprints via Weiss Robotics pressure sensitive tactile pads [3] mounted on the SDH’s fingers. Each finger of the SDH has 2 tactile sensor arrays (Figure 2.5b, c, d) that are composed of 6x13 and 6x14 texels with 3.4 mm spatial resolution and 250 kPa pressure measuring range.

### 2.3.2 Tactile Sensing for Stability Estimation

This thesis firstly studies how grasp stability can be assessed based on data extracted both *prior to* and *during* grasp execution and methods such as AdaBoost, Support Vector Machines and Hidden Markov Models. The data contain object information such as shape, grasp information such as approach vector, and online sensory and proprioceptive data including tactile measurements from fingertips and joint configuration of the hand.

The approach is a learning based framework that relies on having a training data-set which is assumed to sample the domain of possible scenarios well. This poses a challenge: acquiring such data is associated with a significant cost with respect to time and computation. In order to alleviate this problem a simulator is used, from which a large set of synthetic training data can be generated in a controlled environment with relative ease. The approach of using synthetic training data is justified by performing inference on real-world examples. Moreover, the generalizability of the grasp stability estimation is experimentally evaluated.

We also study how the uncertainty in grasp execution posterior to grasp planning can be dealt with using tactile sensing and machine learning techniques. The majority of the state-of-the-art grasp planners demonstrate impressive results in simulation. However, these results are mostly based on perfect scene/object knowledge allowing for analytical measures to be employed. It is questionable how well these measures can be used in realistic scenarios where the information about the object and robot hand may be incomplete and/or uncertain. Thus, sensory information is necessary for successful online grasp stability assessment. Therefore, integration of the online stability assessment based on tactile sensing with a simulation-based grasp planner is presented to improve its grasp selection process.

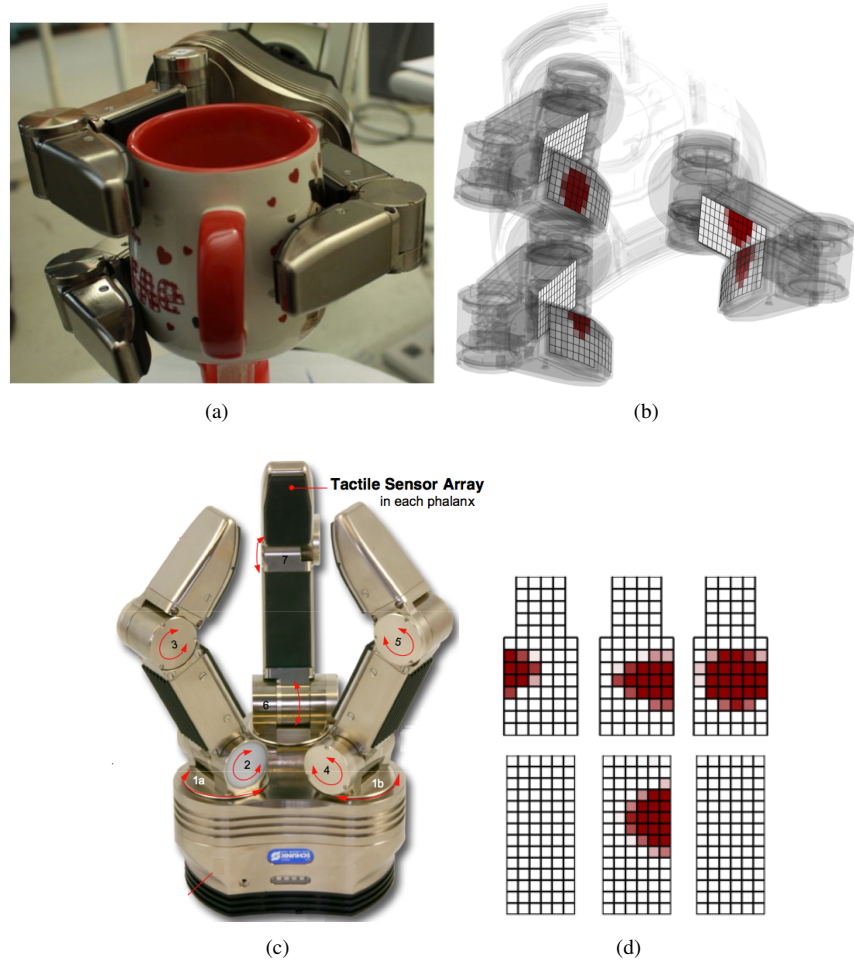


Figure 2.5: Tactile sensors on the SDH: (a) An example grasp. (b) The tactile readings obtained during the example grasp. Maximum pressure readings are represented with dark red. (c) The hand along with the seven marked joints, [2]. (d) The tactile sensor arrays separately from the hand.

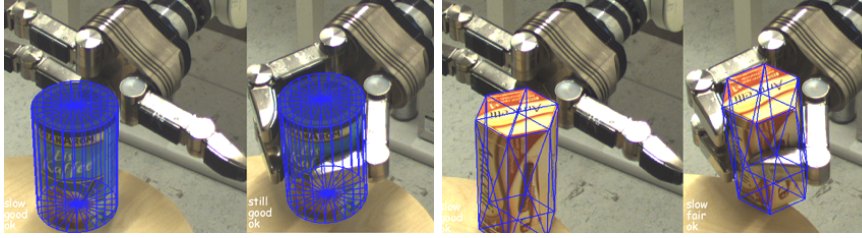


Figure 2.6: Example vision-based pose tracking results: Tracked object models are shown with blue wireframe. Tracking was successful during the grasping experiments starting from the initial hand positioning until the grasp was completed.

### 2.3.3 Combining Visual and Tactile Sensing

This thesis also studies the joint impact of visual and tactile sensing on grasp stability assessment. Vision and touch separately brings valuable information on grasp stability. However, in many situations one modality can substantially help disambiguating the readings obtained from the other one. For instance, it is conceivable that for some object, two grasps approaching from different directions would yield similar tactile readings, but one would allow for robustly moving the object while the other would let the object slip away. Such situations may occur, e.g., because one of the grasps benefits from an extra gripper-object contact point in an area that is not covered by tactile sensors, or because of a different relative configuration of the grasp with respect to the center of mass of the object. Considering both modalities jointly should intuitively lead to more robust assessments. Therefore the percepts considered has been expanded by adding the object-gripper configuration read before and until the robot’s manipulator is fully closed around an object.

To extract those visual features, the robot tracks the pose of an object and obtain the 6D object pose (3D position and 3D orientation) based on the vision system that works with the monocular camera during the application of a grasp. As an object will often move while the robot is closing its hand to grasp it, the agent needs to track the pose of the object during the grasp, which is made difficult by the partial object occlusions effected by the robot hand. This thesis suggests means of using these data, i.e., tactile feedback along with visual feedback, to learn a model of grasp stability and differentiate between successful and unsuccessful grasping configurations before further manipulating the object. In mathematical terms, the agent learns an empirical representation of pose- and/or touch-conditional grasp success probability.

A kernel-logistic-regression model of pose- and touch-conditional grasp success probability is presented and trained on grasp data collected by letting the robot explore and experience the effect on tactile and visual signals of grasps suggested by a teacher, and letting the robot verify which grasps can be used to rigidly control the object. By observing the pose/touch signals issued when executing grasps, the agent learns what it feels like to grasp an object from a specific side. Models defined on several subspaces of the input data

– e.g., using tactile perceptions or pose information only has been considered to analyze the relevance of the two modalities.

### 2.3.4 Task-oriented Estimations

Finally, assessment of grasp success is extended to be in a goal-oriented way, taking into account both geometric constraints for task affordances and stability requirements specific for a task. In this thesis we close the learn-plan-execute loop where the robot learns task knowledge from human teaching, and grounds this knowledge in low-level sensorimotor systems through exploration (manipulating the object) in a real environment.

To integrate the semantic goal of a task with a set of continuous sensory features, we study probabilistic graphical models. Probabilistic frameworks based on graphical models have proved to be powerful in various fields with high-dimensional complex problem domains [108, 36, 49, 78]. Graphical models encode the relations between variables through their probabilistic conditional distributions. Such distributions do not require the variables to have same underlying representations. Therefore, high-level symbolic variables such as task goals can be naturally linked to the low-level sensorimotor variables such as hand configuration. Furthermore, the model can be combined with the probabilistic decision making where grasp plan and control can be performed through inference even with noisy and partial observations [107]. To exploit these strengths, we use Bayesian Networks to model conditional relations between task and stability knowledge with a multitude of features from proprioception, tactile sensing and simulated vision. The generative modeling approach provides a flexible framework to guide detailed grasp planning and execution in a task-directed way.

## Chapter 3

# Learning Grasp Stability through Tactile Sensing

A vision based system can provide information about the specific objects in the scene and their pose [56, 91, 87] or potential grasping points on the object [22, 23]. Previous works have shown how this can be done for known [56], unknown [91, 87] and familiar objects [22, 23]. However, there were cases that resulted in unsuccessful grasps. One example using the system from [23] is shown in Figure 3.1. In the experiment, after placing the hand with respect to the object, the fingers were closed around the object. Based on the resulting grasping configuration, the object could not be lifted. In the experiment, although grasp planning was performed based on visual data, the grasp was executed without using any sensory feedback.

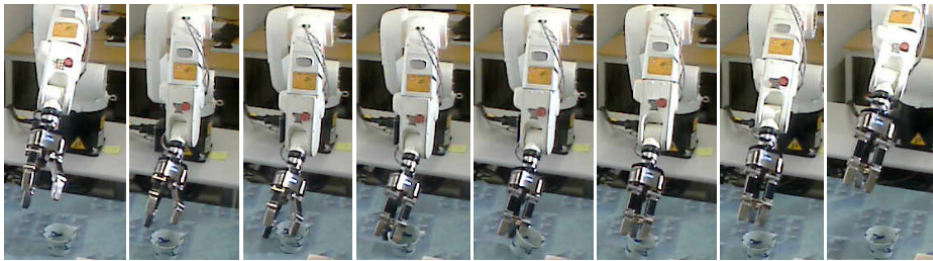


Figure 3.1: An example of a failed grasp when only visual input is used. Details about the system are reported in [23].

Such failures can be prevented by providing estimates for grasp stability based on sensory information during grasp execution. This approach enables the robot to decide not to lift the object when the stability estimate is low. In this chapter we will show how we achieve this goal.

Determining grasp stability is difficult when factors affecting the stability are uncertain or unknown. In this chapter we show that with a probabilistic approach it is possible to assess grasp stability using tactile measurements. Mapping from tactile sensor measurements to grasp stability is complex and not injective because of variability in object parameters, grasp and hand types, and the uncertainty inherent in the process. Thus, we consider grasp stability as a probability distribution

$$P(S|H_t, fcon_t, O, G), \quad (3.1)$$

where grasp stability, denoted by  $S$ , depends on different measured and/or known factors. The factors taken into account in our model are: i)  $H$ , force/pressure measurements from tactile sensors; ii)  $fcon$ , joint configuration of the hand; iii)  $O$ , object information, e.g., object identity or shape class; and iv)  $G$ , information relevant to the grasp, e.g. approach vector and/or hand preshape. Grasp stability,  $S$ , is a discrete variable with two possible states: a grasp is either stable or unstable, while the other variables can be discrete or continuous. Our goal is to assess the effect of factors in Eq. (3.1) to grasp stability by considering different subsets of the variables.

We study the problem using both instantaneous measurements of variables and time-series measurements. With instantaneous measurements, the stability is assessed only from the instant when the robot hand is static and has closed around the object. This approach is referred to as one-shot classification. In contrast, the time-series approach takes into account measurements generated during the whole grasping sequence. The variables  $H$  and  $fcon$  are thus represented from time  $t_0$  to  $t_n$  where  $t_0$  and  $t_n$  represent the start and the end of the grasping sequence respectively. In the case of one-shot classification, we use the measurements once the hand has reached a static configuration, an approach similar to [97]. Thus, we compare the distribution defined by Eq. (3.1) to one which discards the time series:

$$P(S|H_{t_n}, fcon_{t_n}, O, G). \quad (3.2)$$

We show that both approaches described by Eq. (3.1) and Eq. (3.2) are valid and that grasp stability can be assessed based on them. To study the contribution of object  $O$  and grasp knowledge  $G$ , we have set up a hierarchy as depicted in Figure 3.2. The hierarchy is divided into levels, each with increasing amount of sensory information being available. At the top level of the hierarchy only the information related to the hand itself,  $H$ , and  $fcon$  is used. Thus, we estimate

$$P(S|H, fcon) = \int \int P(S|H, fcon, O, G) P(G|O) P(O) dO dG. \quad (3.3)$$

Considering only sensor information, the overall distribution will be somewhat uninformative — there is significant uncertainty as the same sensor readings can be associated with both stable and unstable grasps for different objects, grasp approach vectors and hand preshapes. Subsequently, when more pieces of information are considered, the estimation of the distribution should be more specific resulting in better discrimination. At the second

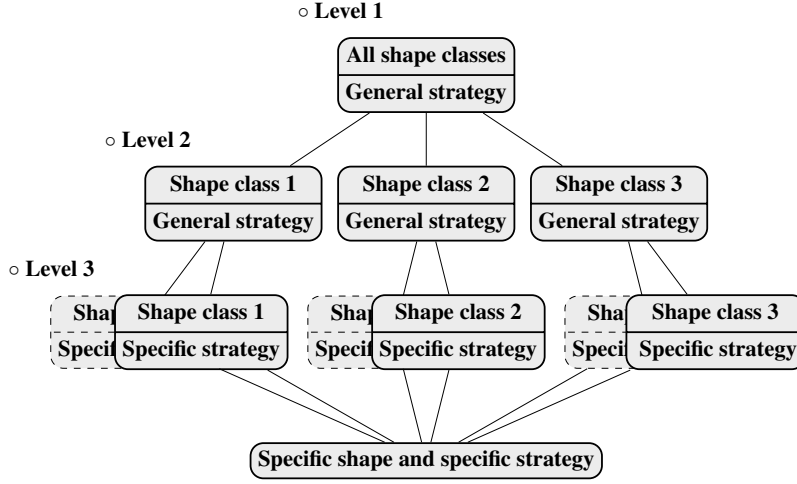


Figure 3.2: Hierarchical recognition of grasp stability taking into account different type of sensory knowledge.

level, we consider that object shape or object instance are known:

$$P(S|H, fcon, O) = \int P(S|H, fcon, O, G) P(G) dG . \quad (3.4)$$

Finally, at the third level we consider knowledge about the applied grasp, and estimate the stability through  $P(S|H, fcon, O, G)$ . Since knowledge of all the variables present in Eq. (3.1) is assumed, the uncertainty in the stability estimation is expected to decrease.

In the next section, we describe methods for estimating the density functions using a classification approach. Support Vector Machines and AdaBoost are used to model the instantaneous model, according to Eq. (3.2) while Hidden Markov models are used for the general time series case, according to Eq. (3.1). Although the probabilistic framework is presented as a method to estimate grasp stability using haptic data, it is also possible to use the proposed framework with other types of sensory information.

### 3.1 Classification Approach

First, we describe the data used to train the classifiers. Tactile measurements are recorded from the first contact with the object until a steady state is reached. The whole measurement sequence is denoted by  $(x_1^i, \dots, x_t^i, \dots, x_{T_i}^i)$ , where  $i$  is the index of the measurement. For one-shot classification, tactile measurements at the steady-state is used and denoted  $x_{T_i}^i$ . Training data is generated both in simulation and on real hardware and will be presented in Section 3.2. The notation used is as follows:

- $D = \{(o_i, S_i)\}_{i=1, \dots, N}$  denotes a data set with  $N$  observation sequences and the corresponding labels.
- $o_i = (x_1^i, \dots, x_t^i, \dots, x_{T_i}^i)$  is an observation sequence.
- $S_i \in \{\text{stable}, \text{unstable}\}$  is a label.
- $x_t^i = (M_t^{i,1}, \dots, M_t^{i,6}, fcon_t^{i,1}, \dots, fcon_t^{i,7})$ , is the observation at time instant  $t$  given the  $i$ -th sequence. Each observation contains features extracted from the six tactile sensor arrays and the seven joint angles of the robot hand.
- $M_t^{i,f}$  includes the moment features extracted from the tactile readings  $H_t^{i,f}$  on the sensor  $f$  at time instant  $t$  given the  $i$ -th sequence. Details about the extraction of these features are given later in this section.
- $fcon_t^{i,v}$  is a joint angle at time instant  $t$  given the  $i$ -th sequence.

The acquired data thus consist of tactile readings  $H$  and joint angles of the hand  $fcon$ . For the Schunk hand, we store  $3 \times (14 \times 6)$  readings on proximal and  $3 \times (13 \times 6)$  on distal sensors, and seven parameters representing the pose of the hand given the joint angles. Example images from the tactile sensors are shown in Figure 3.3. The tactile images in the figure represent a stable grasp of a cylinder.

Tactile data is relatively high dimensional and redundant. Thus, we borrow ideas from image processing and consider the two-dimensional tactile patches as images. Each tactile image is represented using image moments. The general parametrization of image moments for one tactile array  $f$  is given by

$$m_{p,q} = \sum_x \sum_y x^p y^q f_{(x,y)} \quad (3.5)$$

where  $p$  and  $q$  represent the order of the moment,  $x$  and  $y$  represent the horizontal and vertical position on the tactile patch and  $f(x, y)$  the measured contact. We compute moments up to order two,  $(p + q) \in \{0, 1, 2\}$ , which yields 6 numbers that model the total pressure and the distribution of the pressure in the horizontal and vertical direction. There are in total six features  $M_t^{i,f} = (m_{0,0}, m_{0,1}, m_{1,0}, m_{1,1}, m_{2,0}, m_{0,2})$  for each sensor based on Eq. (3.5), resulting in an observation  $x_t^i \in \mathbb{R}^{6 \times 6 + 7}$ . Normalizing the feature vector is a common step in machine learning methods. In our case, moment features and finger joint angles are normalized to zero-mean and unit standard deviation. Normalization parameters are calculated from the training data and then used to normalize the testing sequences.

### 3.1.1 One-shot Recognition

In this section, we examine the learning of grasp stability based on tactile measurements acquired at the end of a grasping sequence, that is, once the final grasp has been applied to the object. We claim that if successful separation between stable and unstable grasps can be learned from examples, one-shot classification can determine the stability of the grasp

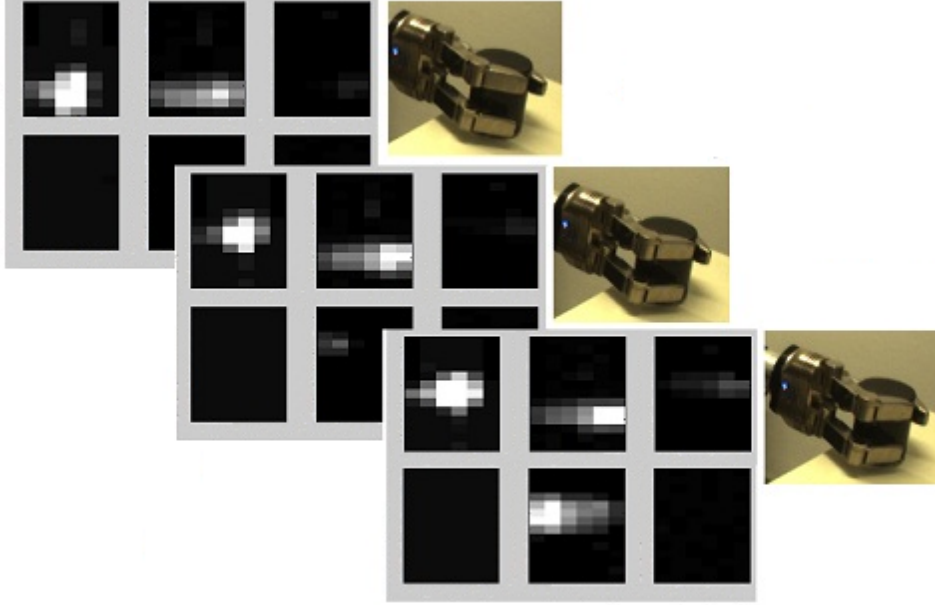


Figure 3.3: An example grasping sequence of a cylinder and the corresponding tactile measurements.

from any haptic observation  $x_t^i$  measured during a grasp. This information can then be used in grasp control to determine when the robot hand has reached a stable configuration.

Two types of non-linear classifiers, AdaBoost and Support Vector Machine (SVM), are used in the experiments to demonstrate the ability to learn the stability of the grasps. AdaBoost and SVM were the best performing classifiers in [70]. AdaBoost is a boosting classifier, developed by Freund and Schapire [42], that works with multiple so-called weak learners to form a committee that performs as the classifier. Here, we use AdaBoost implementation from [112].

Support vector machine classification [33, 111] is also suitable for the problem. SVM is a maximum margin classifier, i.e. the classifier fits the decision boundary so that maximum margin between the classes is achieved. This guarantees that the generalization ability between the classes is not lost during the training of the SVM classifier. We use the libSVM implementation presented in [28]. Another critical feature of the SVM for our use is the ability to use non-linear classifiers instead of the original linear hyper-plane classifier. Non-linearity is achieved using different kernels, in this study the radial basis function (RBF)

$$\mathcal{K}(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \quad \text{for } \gamma > 0, \quad (3.6)$$

is used as the kernel for SVM. Moreover, as an extension to the basic two-class SVM, probabilistic outputs for SVM are used to analyze the results given by the SVM. This idea was first presented in [85]. The SVM output  $y(\mathbf{x})$  is converted to a probability according to  $\sigma(\Gamma y(\mathbf{x}) + \Lambda)$  where  $\sigma(\cdot)$  is the logistic sigmoid function and parameters  $\Gamma$  and  $\Lambda$  are estimated using training data.

### 3.1.2 Temporal Recognition Using Hidden Markov Models

Time-series grasp stability assessment is performed based on Hidden Markov models [89] (HMMs). Here, we use the HMM implementation from [81]. We construct two HMMs: one representing stable and one unstable grasps. Classification of a new grasp sequence is performed by evaluating the likelihood of both models and choosing the one with higher likelihood. For the HMM, we use the notation  $\lambda = (\pi, A, B)$  where  $\pi$  denotes the initial probability distribution,  $A$  is the transition probability matrix

$$A = a_{ij} = P(\mathcal{S}_{t+1} = j | \mathcal{S}_t = i), i, j = 1 \dots N, \quad (3.7)$$

and  $B$  defines output (observation) probability distributions  $b_j(x) = f_{X_t|\mathcal{S}_t}(x|j)$  where  $X_t = x$  represents a feature-vector for any given state  $\mathcal{S}_t = j$ . In this work, we evaluate both ergodic (fully connected) and left-to-right HMMs.

The estimation of the HMM model parameters is based on the Baum-Welch procedure. The output probability distributions are modeled using Gaussian Mixture Models (GMMs):

$$f_X(x) = \sum_{k=1}^K w_k \frac{1}{2\pi^{L/2} \sqrt{|C_k|}} e^{-\frac{1}{2}(x-\mu_k)^T C_k^{-1}(x-\mu_k)}, \quad (3.8)$$

where  $\sum_{k=1}^K w_k = 1$ ,  $\mu_k$  is the mean vector and  $C_k$  is the covariance matrix for the  $k$ -th mixture component. The unknown parameters  $\theta = (w_k, \mu_k, C_k : k = 1 \dots K)$  are estimated from the training sequences  $o = (x_1, \dots, x_T)$ . Initial estimates of the observation densities in Eq. (3.8) affect the point of convergence of the reestimation formulas. Depending on the structure of the HMM (ergodic vs left-to-right), we use a different initialization method for the parameters of the observation densities. The two initialization procedures are given below:

- For an ergodic HMM, observations are clustered using  $k$ -means. Here,  $k$  is equal to the number of states in the HMM and each cluster is modeled with a GMM using standard expectation maximization. Initial parameters for the GMMs are found using  $k$ -means algorithm.
- For a left-to-right HMM, each observation sequence is divided temporally into equal length subsequences. Then, each GMM is estimated from the collection of corresponding subsequences. Thus, the GMMs represent the temporal evolution of the observations. Initial parameters are found as in the case of an ergodic HMM.

### 3.2 Data Acquisition

For a learning system to achieve good generalization capabilities, relatively large training data is typically required. Generating large datasets on real hardware is time consuming and in robotic grasping generating repeatable experiments is difficult due to the dynamics of the process. However, if suitable models are available, simulation can be used for generation of data for both training the learning system and performance evaluation. In our work, we generate both simulated and real training data as explained below.

#### 3.2.1 Simulator

The grasp simulator RobWorkSim, described in [63], is used to generate training data including tactile measurements. The simulator is used in combination with the Open Dynamics Engine (ODE) physics engine and provides support for simulating articulated hands, PD joint controllers, grasp quality measures, camera sensors, range sensors and tactile sensors. The primary motivation for using RobWorkSim over the more widely used GraspIt! [74], is the integrated support for tactile array sensors.

The tactile array sensor simulation in RobWorkSim is an experimental model that transforms the point contacts of the ODE to sensor measurements by describing the deformation of the sensor surface given a point force  $f$  applied perpendicular to it. The model was originally described in [64]. The model assumes that the deformation or response is linear with the magnitude of the point force, which is a fair assumption for small forces. Given the deformation function  $h(x, y)$  where  $x$  and  $y$  are specified relative to the center  $(a, b)$  of the contact, the total deformation of the surface of an array of rectangular texels with size  $(A, B)$  can be found by integrating over the surface of each texel by

$$g_{m,n}(a, b) = \int_{(A-\frac{1}{2})m-a}^{(A+\frac{1}{2})m-a} \int_{(B-\frac{1}{2})n-b}^{(B+\frac{1}{2})n-b} h(x, y) dx dy, \quad (3.9)$$

where  $(a, b)$  is the center point of the contact and  $(m, n)$  is the texel index. This surface integration is approximated using the rectangle method. Point force experiments on the real sensors suggested that the deformation decreased with the inverse of the square of the distance from the point force. We use an isotropic function to approximate the deformation of the sensor surface

$$h(x, y) = (f \cdot n_{texel}) \max(-\beta + \frac{\alpha}{1 + x^2 + y^2}, 0), \quad (3.10)$$

where  $(x, y)$  is specified relative to  $(a, b)$  and  $n_{texel}$  is the normal of the texel on which the point force  $f$  is applied. The parameters  $(\alpha, \beta)$  were found by fitting the model to experimental data extracted from real sensors.

Assessing grasp quality requires taking properties of the hand (orientation, joint configuration, friction, elasticity, grasping force) and object (shape, mass, friction, contact locations and area, contact force) into account. In the simulated environment these parameters are known. We use a widely known grasp quality measure based on the radius,  $\epsilon$ , of the largest enclosing ball in the grasp wrench space (GWS). We construct the

GWS as proposed in [39] by calculating the convex hull over the set of contact wrenches  $w_{i,j} = [f_{i,j}^T \lambda(d_i \times f_{i,j})^T]^T$ , where  $f_{i,j}$  belongs to a representative set of forces on the extrema of the friction cone of contact  $i$ .  $d_i$  is the vector from the torque origin to contact  $i$  and  $\lambda$  weighs the torque quality relative to the force quality.

It is not obvious how to determine  $\lambda$  due to the differences between forces and torques. We therefore calculate force space and torque space independently and use the radius of the largest enclosing ball in each of these to give a 2 dimensional quality value  $(\epsilon_f, \epsilon_\tau)$  for each grasp. A third quality measure  $\epsilon_{cmc}$  based on the distance between the centroid of the contact polygon  $C$  and the center of mass  $CM$  of the object [105] is used:  $\epsilon_{cmc} = \|CM - C\|$ . This measure captures the same properties as the torque measure, however it is more robust with regard to the point contact output of the simulator. Stable grasps are defined as those for which all three quality values are within a certain threshold. The thresholds have been determined experimentally.

### 3.2.2 Generating Training Data in Simulation

The database includes examples of stable and unstable grasps on different objects. We examine stability starting from the most general case in the hierarchy specified in Figure 3.2 and continue by including information about subsequent properties until reaching the most specific case. At the top level of the hierarchy, data is generated on objects with different shapes using approach vectors generated uniformly from a sphere, referred to as a spherical strategy. At the second level, the shape information is given, hence grasps are generated separately per object shape with the spherical strategy. At the third level, the approach vector is formed based on the object shape, namely side or top grasps are applied with more than one preshape. At the bottom level, the preshape is also chosen per object shape and approach vector. Figure 3.4 shows examples of objects that are included in the database.

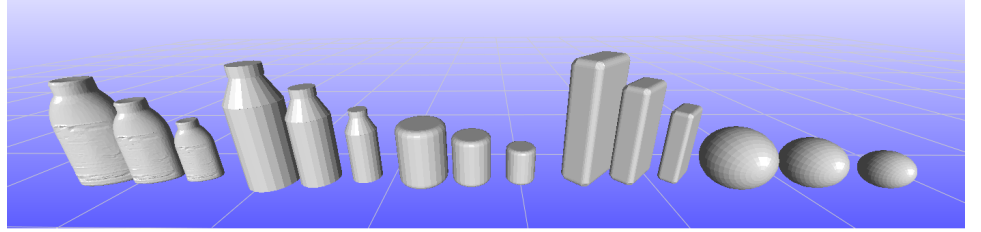


Figure 3.4: Objects in simulation were generated in three sizes (75%,100%,125%): Hamburger sauce, Bottle, Cylinder, Box, Sphere.

Each grasping sequence in the database is generated by placing the hand in a specific configuration with respect to the object and then closing the fingers. For the recognition that relates to levels 1 and 2 in the recognition hierarchy (see Figure 3.2), a simple spherical grasp strategy with a randomly chosen preshape is used. The spherical grasp strategy generates the approach direction for the hand by sampling the unit sphere around the center

of mass of the object. Each sample then consists of a vector pointing toward the center of mass of the object.

The strategy and the preshapes used for level 3 in the recognition hierarchy are shape specific. Therefore strategies were developed for each shape used in the experiments. The hand preshapes for level 3 were generated with finger joint values in the interval  $([-90; -70], [-10; 10])^\circ$ , where the 7th joint was one of  $90^\circ, 60^\circ, 0^\circ$  as shown in Figure 3.5.

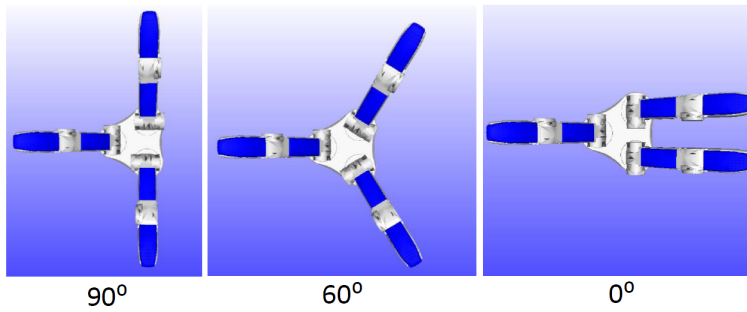


Figure 3.5: Hand configuration when the 7'th joint is at  $90^\circ$ ,  $60^\circ$  and  $0^\circ$

The following grasp strategies are applied for the shape primitives:

- Sphere - The approach directions are sampled randomly from the unit sphere with origin in the center of gravity of the object. Both the ball preshape ( $60^\circ$ ) and the parallel preshape ( $0^\circ$ ) were used.
- Cylinder - The object is approached either from the top or from the side. When approaching from the top, a ball grasp preshape is used and the approach direction is pointing towards the object center of mass. For side grasps, the approach is sampled with an angle of  $0 - 20^\circ$  with respect to the horizontal plane, pointing towards the center of mass of the object. The preshape in the side grasp uses an angle of  $0^\circ$  on joint 7, so that a parallel grasp is obtained.
- Box - The object is approached using a vector lying in the plane defined by the world z-axis and the longest axis of the box and pointing toward the center of gravity. A parallel preshape of the hand is used.

In addition, for the hamburger sauce and the bottle (see Figure 3.4), we used the same strategy as the cylinder. The tactile information and the joint configuration are recorded from simulation at regular time intervals.

In general, the performance of the simulation is largely dependent on the level of detail of the geometries in both hand and objects. In our setup generating a simulated grasp using a modern quad core computer took approximately 2 seconds.

### 3.2.3 Generating Training Data on a Robot

The real world experiments show the feasibility of assessing grasp stability on physical robot platforms. The experiments aim to serve as a proof-of-concept rather than assessing the exact performance rates in different use cases. The experimental evaluation on real data follows the methodology used in simulation such that similar objects and same grasp types are used. The objects are placed such that they are initially not well centered with respect to the hand to assess the ability of the methods to cope with the uncertainty in pose estimation. A few example grasps are shown in Figure 3.6. The real data includes side grasps on the objects in Figure 3.7 with the preshape shown in Figure 3.5 where the 7th joint is  $0^\circ$ .



Figure 3.6: A few examples from the execution of real experiments.



Figure 3.7: Objects used in real experiments: Box, cylinder, cone, orange bottle, pitcher, shampoo bottle, deformable cylinder, blue bottle, white bottle.

Tactile readings and corresponding joint configurations were recorded starting from the first contact until a static state is achieved. To generate stable/unstable label for a grasp, the object is lifted and rotated  $[-120^\circ, +120^\circ]$  around the approach direction. The grasps where the object was dropped or moved in the hand were labeled as unstable. 100 stable and 100 unstable grasps were generated for each object.

### 3.3 Experimental Evaluation

In this section, we study the effect of different types of information for the estimation of grasp stability. We first introduce the experimental setup, then present the results of one-shot and temporal recognition.

#### 3.3.1 Evaluation of Learning Capability

The experiments are divided according to the hierarchy presented in Figure 3.2. The goal is to evaluate the effect of the increasing knowledge on the classification results with both one-shot and temporal classification approaches.

##### Level 1: No constraints

On this level, no constraints are placed on the data used for training the classifiers. In other words, only tactile sensor measurements and the joint configuration are available and the other variables are unknown. The grasps are sampled from a sphere and the hand is oriented towards the object. The data is collected in simulation across multiple object shapes and scales.

##### Level 2: Constraints on object shape

The shape of the object is known, enabling the use of shape specific classifiers. The grasps are randomly sampled from a sphere and the hand is oriented towards the object. The data is collected in simulation.

##### Level 3: Constraints on approach vector, preshape and object shape

On level 3 of the hierarchy, constraints are placed on the approach vector, the grasp preshape and the object shape. The data are collected using a manually chosen approach vector, and the preshape is adjusted to the shape of the object. On this level, the shape is known so that shape specific classifiers can be used. Both simulated data and real data are available at this level.

#### 3.3.2 Experimental Setup

##### Data

The simulated data used in the experiments consists of five objects with three different grasp configurations applied to them. Three of the objects have primitive shape (box, cylinder, sphere), and two have natural shape (hamburger sauce, bottle). Each object is scaled to three different sizes, 0.75, 1.0, and 1.25 of the original size. For each object/size/grasp combination, 1000 unstable and 1000 stable grasps are randomly chosen from the database described in Sec. 3.2.2. Thus, each object/grasp dataset consists of 3000 stable and 3000 unstable grasps. When we refer to specific simulated object/grasp combination, terms *side*

or *top* are used for grasps generated as side and top grasps, while *sph.* is used for grasps generated uniformly from a sphere around the object (random approach vector). Altogether, there are then 30000 samples for the five objects. We also refer to the root node of the information hierarchy, which contains all samples of primitives shapes, a total of 18000 samples.

The real data collected includes nine objects with 100 unstable and 100 stable grasps for each object. Thus, there are 1800 samples in the real data set. The details of the real data collection are described in Sec. 3.2.3.

### One-shot Recognition

As mentioned in Section 3.1.1, we utilize the AdaBoost-algorithm in one-shot classification. Due to the formulation of the AdaBoost, a weak learner needs to be chosen. In the experiments, a decision tree with a branching factor of 1 was used as the weak learner, effectively reducing the tree to a series of linear discriminants. The branching factor was determined from series of tests that showed that using branching factor of 1 performed as good or better as larger branching factors on the data described in Sec. 3.2. 200 iterations of AdaBoost were run to find the final classifier in all experiments. For the SVM classifier,  $\gamma = 0.03$  and constant the  $C$  related to the penalty applied to incorrectly classified training samples [33] is set to  $C = 0.4$ .

All experiments are reported as 10-fold cross validation averages, except where otherwise noted. In each case, the data sets used for training and testing the classifiers are balanced, i.e. the data sets contain equal number of unstable and stable grasps. Image moments are used as the feature representation for the one-shot classifiers. The joint data in addition to the tactile data is also included in the features unless otherwise noted.

### Temporal Recognition

To study if the temporal information improves the recognition performance, two HMMs, one for stable grasps and another for unstable ones, were trained. The stopping criteria for HMM training was a convergence threshold of  $10^{-4}$  with a 10 iteration limit. In order to improve the reliability of the evaluation, both ergodic and left-to-right HMM were evaluated independently. The reason for these multiple experiments is that by evaluating multiple temporal models we aim to understand if the temporal ordering plays part in the modeling. The covariance of the mixture model component distributions was forced to be diagonal.

In the training of the temporal model, the structure of the HMM needs to be chosen in the form of structural parameters, which describe the number of HMM states and the number of mixture model components for each state. These were chosen experimentally such that the HMM was trained using different parameter settings and the setting producing at least lowest equal error rate result (equal number of false positives and negatives) or better performance than that was chosen. The number of states was varied between 2 and 6 while the number of mixture components was between 2 and 5.

Experiments were performed both on simulated and real data. For simulated data randomly chosen 80% of the samples were used for training and the rest 20% for testing. For the real data 10-fold cross validation was used to evaluate the performance and best parameter setting over all folds was chosen.

Image moments were used as features, similar to one-shot learning. However, to reduce the number of parameters in HMM and speed up the training process, principal component analysis (PCA) was applied to the moment and joint measurements separately to reduce the dimensionality of the dataset. The number of principal components was chosen such that at least 95% of the total variance is retained.

### 3.3.3 One-shot Recognition Results

In this section, we present a collection of experiments based on the information hierarchy in Figure 3.2 using the AdaBoost classifier. Support vector machine classifier is used with image moments to examine the separability of the grasp stability at each level by means of log-likelihood histograms. We also study the effect of the joint configuration data on the classification by including or excluding it from the feature vector for the classifier when using real data. Training time for the classifiers is less than five minutes, for the reported amount of samples. Adaboost training time increases linearly with the amount of samples while SVM training time increases quadratically. Classification of a single sample takes less than 10 ms with both of the presented classifiers. SVM classification time increases linearly with the amount of samples used for training.

#### Real Data

The experiments begin by showing results using real data. Sampling grasps with a real hand is a slow process and thus the sample size is limited. To study the effect of the amount of samples used for training, we ran a series of tests with variable sample sizes. In each case, the same object was used both for training and testing. The results of these tests are shown in Table 3.1, which shows the classification rates for training data sets of different sizes. The test shows that for a specific grasp on the cylindrical object, 100 samples are already enough to reach classification performance levels achieved with higher amount of samples, the differences in classification performance above 100 samples are not statistically significant. However, this is the case only when the stable and unstable grasps are distinctive, i.e. we achieve a high rate of correctly classified grasps. In the case of the white bottle data set, where the classification rate is lower, the results show that more than 200 samples could be useful in increasing the classification performance.

Classification results as percentages for single object classifiers (known object case) are presented in columns 2 and 3 of the Table 3.2. Classification rates are shown both with joint configuration data and without it, and the classification rates were computed for image moment feature representations. The main focus in this experiment is to study prediction of the grasp stability on known objects that the system has previously learnt. The average classification rate for known objects is 82.5% including joint data and 81.4% excluding it from the measurements. Thus, the inclusion of joint data seems to benefit the recognition

Table 3.1: AdaBoost classification rates (in percent) on data sets with variable amount of samples.

| Samples       | 50     | 100    | 150    | 200    |
|---------------|--------|--------|--------|--------|
| Def. cylinder | 74.6 % | 85.0 % | 84.8 % | 89.0 % |
| W. Bottle     | 64.6 % | 68.0 % | 68.5 % | 75.5 % |

but only to a minor effect. Moreover, the result indicates that at least with known objects the proposed approach seems to have adequate recognition rate for practical usefulness.

Table 3.2: AdaBoost classification rates (in percent) on known and unknown objects with and without joint data.

|               | Known obj. |        | Unknown obj. |        |
|---------------|------------|--------|--------------|--------|
|               | w/j        | wo/j   | w/j          | wo/j   |
| Cylinder      | 88.9 %     | 90.3 % | 80.4 %       | 81.9 % |
| Def. cylinder | 91.0 %     | 89.0 % | 76.0 %       | 76.5 % |
| Cone          | 79.5 %     | 81.0 % | 73.0 %       | 68.0 % |
| O. Bottle     | 77.0 %     | 78.5 % | 72.5 %       | 72.0 % |
| Shampoo       | 82.5 %     | 76.0 % | 70.0 %       | 71.5 % |
| Pitcher       | 84.5 %     | 78.0 % | 71.0 %       | 66.0 % |
| W. Bottle     | 76.0 %     | 73.5 % | 75.0 %       | 76.0 % |
| B. Bottle     | 74.0 %     | 75.0 % | 68.5 %       | 69.0 % |
| Box           | 89.0 %     | 91.0 % | 78.0 %       | 73.0 % |

We also study how well the trained system can cope with unknown objects, i.e. objects that have not been used to train the system. The results are shown as percentages of correct classification in columns 4 and 5 of the Table 3.2, adjacent to the results with known objects. The results are for a system that has been trained on all the objects except the object for which the classification rate is shown. The average recognition rate is 73.8% with joint data and 72.7% without it. The results show that while the classification rate is lower than with known objects it is still possible to make predictions of the grasp stability on unknown objects to some extent. However, this holds true only when similar grasps are applied on unknown objects as were applied to the objects that the system were trained on. In comparison, including grasps from all objects, including the one being tested, for a single classifier yields a result of 78.6 % correct classification across all the objects in the real object set. This indicates that the variety of objects used in training plays an important role in order to attain good performance, and that the knowledge of object identity is useful but does not seem necessary if the training data includes same or similar objects.

Two objects of a primitive shape are included in the real data, a box and a cylinder. Table 3.3 shows classification percentages when the classifier is trained only on one of the primitive objects. The classifier is then asked to classify the grasp stability of grasps made on real-world objects with different shapes. Cross validation was not needed in this case,

because the training and test sets are naturally separate. The average classification rate for the cylinder model is 68.0 % and for the box model 66.4 %. These results do not anymore seem adequate for a real system, which again suggests that the variety in the training data is essential.

Table 3.3: Classifier performance (in percent) when training with a primitive object.

| Trained object | Cylinder | Box    |
|----------------|----------|--------|
| Def. cylinder  | 76.0 %   | 73.5 % |
| Cone           | 66.0 %   | 69.5 % |
| O. Bottle      | 64.5 %   | 61.0 % |
| Shampoo        | 66.5 %   | 64.0 % |
| Pitcher        | 71.0 %   | 62.0 % |
| W. Bottle      | 73.5 %   | 69.5 % |
| B. Bottle      | 58.5 %   | 65.0 % |

### Simulated Data

In contrast to the real data, in simulation we are able to sample a large number of grasps from different objects and using different grasp strategies. The following classification results were achieved using the simulated data sets described in Section 3.2. In Table 3.4, classification percentages are reported for each node in the information hierarchy. The root node (Level 1) was randomly subsampled to 12000 samples due to computational constraints and has classification rate of 75.3%. The average classification for Level 2 (known object, unknown approach vector) is 76.5% and for Level 3 (known object, known grasp) 77.5%. A trend that increasing knowledge increases classification rate appears, similar to the experiments with real data. However, the trend is significantly weaker compared to the real data. Somewhat surprisingly, the real data classification rates are notably higher when more information is available and the trend is stronger, compared to simulation.

Table 3.4: AdaBoost classification rates (in percent) according to the information hierarchy on simulated data.

| Level   | Node                | Classification rate |
|---------|---------------------|---------------------|
| Level 1 | Root                | 75.3 %              |
| Level 2 | Prim. cylinder sph. | 73.5 %              |
|         | Prim. box sph.      | 79.2 %              |
|         | Prim. sphere sph.   | 77.0 %              |
| Level 3 | Prim. cylinder side | 80.7 %              |
|         | Prim. cylinder top  | 67.6 %              |
|         | Prim. box side      | 83.5 %              |
|         | Prim. sphere side   | 78.5 %              |

While the primitive shapes used in Table 3.4 are simple shapes, we can use these primitive shapes to train the classifier and then use the classifier to classify grasps sampled from

more natural, complex objects. The results are shown as percentages of correct classifications in Table 3.5. Each row corresponds to a tested natural object (hamburger sauce, bottle), while each column corresponds to a combination of a training object and grasp strategy. Comparison results when training the classifier with the natural object and corresponding grasping strategy are shown italic font. The figures in the table show that having data from the correct object has a notable positive effect on the classification rates. This is again a positive argument for the beneficial effect of a variety of training data.

Table 3.5: AdaBoost training with a primitive shape and classifying grasps sampled from a natural object with simulated data.

|        | Prim.<br>cyl.<br>sph. | Prim.<br>cyl.<br>side | Prim.<br>cyl.<br>top | Prim.<br>box<br>sph. | Prim.<br>box<br>side | Prim.<br>sphere<br>sph. | Prim.<br>sphere<br>side | All<br>classes<br>sph. |
|--------|-----------------------|-----------------------|----------------------|----------------------|----------------------|-------------------------|-------------------------|------------------------|
| Hamb.  | 71.5<br><i>78.7</i>   | 74.0<br><i>83.5</i>   | 62.9<br><i>72.4</i>  | 76.8<br><i>78.7</i>  | 73.6<br><i>82.0</i>  | 61.4<br><i>78.7</i>     | 62.7<br><i>83.5</i>     | 73.4<br><i>78.7</i>    |
| Bottle | 68.6<br><i>74.7</i>   | 77.4<br><i>82.0</i>   | 56.2<br><i>65.2</i>  | 72.6<br><i>74.7</i>  | 76.9<br><i>82.0</i>  | 59.4<br><i>74.7</i>     | 66.9<br><i>82.0</i>     | 69.7<br><i>74.7</i>    |

Using the SVM and its ability to output estimates of the prediction certainty, gives us a possibility to examine the performance of the classifier on different data sets in more detail compared to AdaBoost, which supports only the hard decision boundary. This comparison can be seen in Figure 3.8. In the figure, log-likelihood ratios,  $\log \frac{1-P(S)}{P(S)}$ , calculated from the probabilities for stable and unstable samples are shown in histogram form, red for unstable and light blue for stable. The classification errors are shown in filled color, with the filled area indicating the error probability. Figure 3.8a-c are from simulated data and Figure 3.8d is from the real cylinder. It is evident from the figure that increasing information makes the distributions for stable and unstable grasps more separate, which was also indicated by the earlier results. Moreover, the figure also supports the finding that classifying the real data seems to be easier than the simulated data. Finally, the figure supports the use of probabilistic approaches for grasp classification, as the ability to measure the uncertainty in classification is important as it can, for example, allow tuning the classification system to give fewer false positives.

### 3.3.4 Recognition Based on Temporal Model Results

In this section, we present HMM classification results obtained from the previously defined experiments. With given parameters, the training time for the HMM is less than thirty minutes, for the reported amount of samples. The training time increases linearly with the amount of samples. Classification of a single sample takes less than 50 ms.

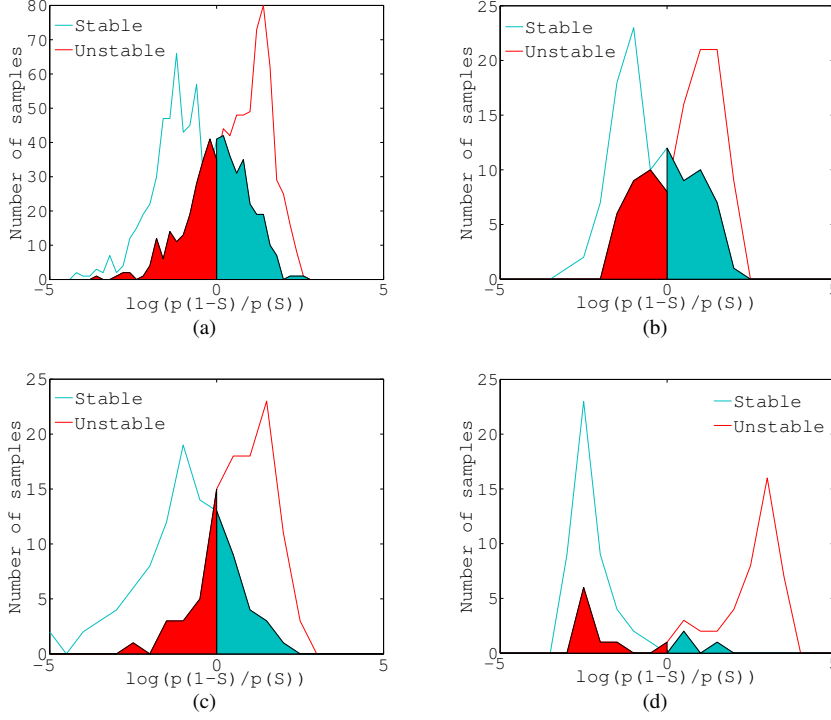


Figure 3.8: Likelihood ratios for comparison of separability: (a) Root node, all objects, random grasp vector; (b) Cylinder, random grasp vector; (c) Cylinder side grasp; (d) Real cylinder side grasps.

### Real Data

Similar to one-shot classification, we begin by investigating the general performance and the required number of samples for achieving good generalization properties. Table 3.6 shows HMM classification percentages corresponding to Table 3.1. The results demonstrate that the performance of HMM classifier does not change much for distinctive grasps such as the ones from the deformable cylinder. Compared to the one-shot model, the temporal model seems to have better generalization capability in that the classification rate does not decrease significantly with smaller data sets.

Classification percentages for single object classifiers are presented in Table 3.7 both with joint configuration data (w/j) and without it (wo/j), to study the prediction capabilities on objects the system has previously learnt with the two HMM types (left-to-right: LR, ergodic: ERG). The average classification rate for known objects (with joint data) is 82.4% with LR and 81.7% with ERG which are on a par with the one-shot learning (Table 3.2).

Table 3.6: HMM classification rates (in percent) on data sets with variable amount of samples.

| Samples       | 50     | 100    | 150    | 200    |
|---------------|--------|--------|--------|--------|
| Def. cylinder | 86.7 % | 85.0 % | 85.4 % | 87.0 % |
| W. Bottle     | 78.3 % | 82.0 % | 74.8 % | 75.0 % |

Thus, with single object classifiers the inclusion of temporal information did not increase classification performance.

Table 3.7 also includes the results that study how well the trained system can cope with unknown objects, corresponding to Table 3.2 for the one-shot learning. The rates not included (marked with a dash) were below the level of chance. The results are similar in the way that the classification rates drop with unknown objects, average rate with joint data being 77.5% for LR and 77.0% for ERG. However, the rate for unknown objects is in most cases high enough such that while the classification rate is lower than with known objects, it is still possible to make useful predictions of the grasp stability on unknown objects. LR seems to outperform ERG slightly in both cases but the difference is not significant. The reason for the difference is likely to be the simpler structure forced by the LR model, which in turn is likely to prevent overfitting. In comparison, using all data from all objects for a single classifier yields a result of 78.3% for LR model and 76.5% for ERG. It is remarkable that the difference between these and the results without the test object in the training data is less than 1%. Thus, with real data it seems that the generalizability of grasp stability across objects is surprisingly good.

Table 3.7: HMM classification rates (in percent) on known and unknown objects.

|          | LR, Kn. |      | ERG, Kn. |      | LR, Unkn. |      | ERG, Unkn. |      |
|----------|---------|------|----------|------|-----------|------|------------|------|
|          | w/j     | wo/j | w/j      | wo/j | w/j       | wo/j | w/j        | wo/j |
| Cyl.     | 90.0    | 86.5 | 92.5     | 82.0 | 83.0      | 77.5 | 81.0       | 75.0 |
| Def. cyl | 87.0    | 83.5 | 85.0     | 83.0 | 76.0      | 75.5 | 76.0       | -    |
| Cone     | 83.0    | 80.0 | 81.0     | 85.0 | 77.0      | 73.5 | 76.0       | 69.5 |
| O. Bott. | 74.0    | 76.5 | 75.0     | 73.5 | 77.5      | 77.5 | 74.5       | 77.5 |
| Shamp.   | 81.0    | 77.5 | 78.5     | 77.5 | 81.0      | 75.5 | 79.0       | 75.0 |
| Pitcher  | 83.0    | 81.5 | 84.0     | 73.5 | 72.5      | 77.5 | 72.5       | 65.0 |
| W. Bott. | 75.0    | 69.0 | 74.0     | 59.5 | 77.0      | 65.0 | 77.5       | -    |
| B. Bott. | 78.5    | 71.0 | 75.0     | 66.0 | 75.5      | 69.0 | 75.0       | -    |
| Box      | 90.5    | 67.0 | 90.5     | 68.0 | 78.5      | 79.0 | 81.5       | -    |

Table 3.8 shows classification results when the classifier is trained only on one of the primitive objects, corresponding to one-shot learning results in Table 3.3. The average rate

for cylinder primitive is 64.6% for LR and 62.3% for ERG, which are below the results of one-shot recognition. For box primitive, the recognition rate for pitcher was below level of chance and is thus not shown. On average, the rates for box primitive are nevertheless higher than for the cylinder primitive and also higher compared to the one-shot learning. The cause of failure for the single object could not be identified. Altogether, the results are in agreement with those from one-shot learning in that the variety of training data seems important to attain good and stable performance.

Table 3.8: HMM classification rates (in percent) when training with a primitive object.

| Trained object | Cylinder |      | Box  |      |
|----------------|----------|------|------|------|
|                | LR       | ERG  | LR   | ERG  |
| Def. cylinder  | 67.0     | 69.5 | 74.0 | 74.5 |
| Cone           | 66.0     | 66.0 | 70.0 | 76.5 |
| O. Bottle      | 63.0     | 60.0 | 72.0 | 74.5 |
| Shampoo        | 61.5     | 57.5 | 75.5 | 77.5 |
| Pitcher        | 79.5     | 78.5 | -    | -    |
| W. Bottle      | 58.5     | 50.0 | 76.5 | 76.5 |
| B. Bottle      | 57.0     | 55.0 | 73.5 | 74.5 |

### Simulated Data

Using the simulated data, Table 3.9 reports the results for each node in the information hierarchy, corresponding to Table 3.4 for the one-shot learning. For LR model, the average classification for Level 1 (root node, unknown object, unknown approach vector) is 64.9%, 69.9% for Level 2 (known object, unknown approach vector), and for Level 3 (known object, known grasp) 67.5%. The results for ERG are similar. There are two observations to be made. First, these are consistently lower than those with one-shot learning, which is the opposite behavior compared to the real data experiments, indicating that the simulated and real data do not match exactly. Second, the trend that increasing knowledge increases performance is broken for Level 3, although the difference is not significant. A possible explanation for this is that the stability of top and side grasps is on average more difficult to model with the HMM compared to modeling the stability of a grasp with random approach vector, because it is possible that some of the grasps with random approach vector might be especially easy to recognize correctly.

The classification performance when training with primitive shapes but testing with real-world objects is shown in Table 3.10, corresponding to Table 3.5 for the one-shot classification. The classification rates with the correct object are shown in *italic* for comparison. The results indicate that on average the classification is significantly improved by having the correct object model instead of a general primitive model, again indicating the importance of variety in training data. Moreover, the results are again inferior to one-shot recognition, strengthening the finding that the temporal information is not essential

Table 3.9: HMM classification rates (in percent) according to the information hierarchy on simulated data.

| Level   | Node                | LR   | ERG  |
|---------|---------------------|------|------|
| Level 1 | Root                | 64.9 | 64.6 |
| Level 2 | Prim. cylinder sph. | 70.2 | 70.2 |
|         | Prim. box sph.      | 62.1 | 59.0 |
|         | Prim. sphere sph.   | 77.4 | 76.9 |
| Level 3 | Prim. cylinder side | 69.3 | 64.3 |
|         | Prim. cylinder top  | 69.5 | 69.3 |
|         | Prim. box side      | 68.6 | 69.0 |
|         | Prim. sphere side   | 62.8 | 63.2 |

for recognition with the available simulated data. To conclude, the real-world cases seem to contain dynamic phenomena which can be modeled better using a temporal model.

Table 3.10: HMM Training with a primitive shape and classifying grasps sampled from a natural object with simulated data.

|                   |    | Hamb. |      | Bottle |      |
|-------------------|----|-------|------|--------|------|
| Prim. cyl. sph.   | LR | 61.2  | 60.1 | 58.4   | 57.8 |
|                   | ER | 60.8  | 57.2 | 58.3   | 55.6 |
| Prim. cyl. side   | LR | 63.3  | 67.5 | 67.6   | 65.8 |
|                   | ER | 60.3  | 68.0 | 64.3   | 66.8 |
| Prim. cyl. top    | LR | 57.8  | 68.1 | 63.1   | 68.8 |
|                   | ER | 57.3  | 64.8 | 65.4   | 69.1 |
| Prim. box sph.    | LR | 59.2  | 60.1 | 58.4   | 57.8 |
|                   | ER | 57.3  | 57.2 | 54.2   | 55.6 |
| Prim. box side    | LR | 63.1  | 67.5 | 57.2   | 65.8 |
|                   | ER | 61.1  | 68.0 | -      | 66.8 |
| Prim. sphere sph. | LR | 51.6  | 60.1 | 52.4   | 57.8 |
|                   | ER | 52.9  | 57.2 | 52.8   | 55.6 |
| Prim. sphere side | LR | 65.2  | 67.5 | 62.7   | 65.8 |
|                   | ER | 63.2  | 68.0 | 59.6   | 66.8 |
| All classes sph.  | LR | 59.3  | 60.1 | 57.4   | 57.8 |
|                   | ER | 59.6  | 57.2 | 58.5   | 55.6 |

### 3.4 Integrating Grasp Planning with Online Stability Assessment

In this section we show how the uncertainty in grasp execution posterior to grasp planning can be dealt with using our method. We integrate the stability assessment method with a grasp planner. We concentrate on grasp stability implementation and evaluation on a real system rather than in simulation. For this purpose, we integrate our method with a simulation-based grasp planner, in order to analyze the relation between analytical stability in simulation and real grasps under consideration of a range of uncertainties. We use a part-based grasp planner in the experiments. Grasps on parts are useful in terms of task-oriented grasping, and they represent a good trade-off between learning from random grasps and learning from manual side- and top-grasps only. Specifically, using a grasp planner that prunes the space of possible grasps, and thereby also tactile patterns, simplifies the classification as well as acquisition of training data on a real platform. On the other hand, part-based grasping is more challenging than grasps centered on the center of mass. Especially for part-based grasps, the center of mass is an important factor for stability. For the experimental evaluation, we selected a publicly available box-based grasp planner, BADGr [54].

The rest of this section is organized as follows: We first describe the data acquisition process where we generate grasps in simulation, then execute them on the real platform. We discuss grasp analysis results obtained during grasping experiments. We then train HMM based classifiers with the sensory data and the stability labels obtained from the real grasps. We present classification performance and show examples of stability prediction.

#### 3.4.1 Grasp Planning in Simulation and Execution on the Real Platform

We illustrate the complete processing pipeline of our data acquisition in Figure 3.9. We first generate grasp hypotheses using the planner BADGr. BADGr allows us to approximate given objects with primitive box shapes based on an efficient minimum volume bounding box implementation. After object decomposition into boxes, BADGr plans grasps on decomposed parts. The grasp generation on the object parts is done in the following way: The hand is placed at a constant distance from the box face’s center aligned with the face normal. The hand approaches the box along the normal until contact. Then the hand is placed back a small distance and closes the fingers. Quality measures (e.g., a worst-case epsilon measure for force-closure grasps [39]) are used to evaluate the grasp based on the acquired contacts. An example box decomposition of one of the objects used in the experiments and a resulting grasp hypothesis generated by the planner can be seen in Figure 3.10.

Except for the work presented in [56], this grasp planner has not been evaluated on a real robot. In our case, we use known objects in a similar way by first generating object-centered grasps offline. In our setup, the pose of each object will be fixed. This allows us to evaluate and compare their analytical stability in GraspIt [74] with our evaluation of stability in the real world. The grasps are parameterized by the relative pose of the hand with respect to the object. The planner is configured for a specific preshape which uses an angle of 0 on joint 7 so that a parallel grasp is obtained as in Figure 3.5. The planned grasp hypotheses are applied on the real robot by placing each of the five objects (Figure 3.9a) in

a known position. For each hypothesis, the hand is moved to the corresponding grasp pose and the fingers are closed. Tactile measurements and corresponding joint configurations are recorded starting from the first contact with the object until a steady-state is reached.

### 3.4.2 Results for Analytic Grasp Analysis in Simulation

In Table 3.11, we present the results generated for the five test objects, and divide into those detected as stable or unstable through analytic analysis in the simulator. For the asymmetric objects, a range over all 4 orientations is shown. For our later training, we aim at approximately 20 samples per object and stability class. We therefore increase the number of grasps by adding 3 offset distances that influence the palm position along the approach vector. Given a side-view of a box around a part, (see the light green box in

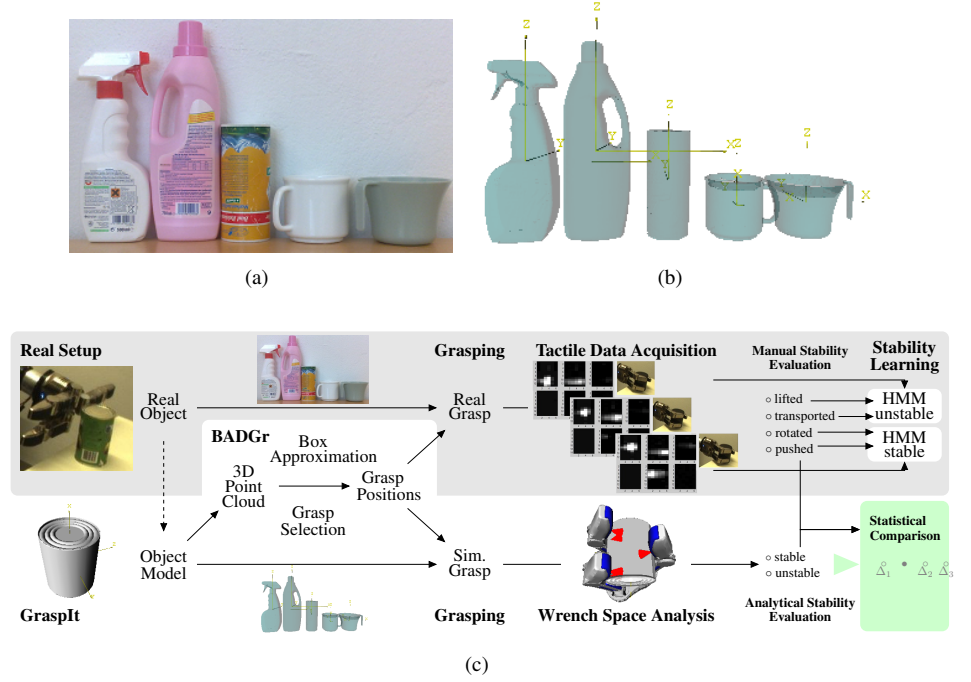


Figure 3.9: Data Acquisition: (a) Real objects from left to right: the spray bottle, the pink bottle, the salt bottle, the white cup, the green cup, and (b) their corresponding models used to generate our dataset. (c) The data acquisition pipeline: We generate grasp positions for 5 known objects using the BADGr framework. We compare acquired stability levels of both the real (top) and the simulated (bottom) execution of each grasp. For the real setup, we acquire sensor data to learn real stability and instability from two HMMs. The light green box on the right bottom represents a side view of a box for a part of an object. The offset distances mentioned in Section 3.4.2 and Table 3.11 are described by  $\Delta_i$ .

Figure 3.9), and an approach vector (see the arrow from its left), these  $\Delta_i$  decentralize the original grasp center position on the object.

### 3.4.3 Results for Real Grasp Analysis

While applying hypotheses on the real platform, we use an extended evaluation strategy based on the following stability degree formulations for our learning mechanism. We label a grasp as *reached* if it was at least possible to move the hand into the planned pose. If *lifted*, the object was also successfully lifted up vertically. *Transported* means a stable transport to a fixed location without dropping. The two last stages describe if the object

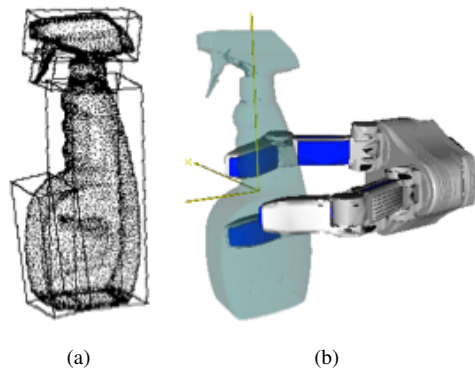


Figure 3.10: Grasp planning by BADGr: (a) The box decomposition of one of the object models used in the experiments. (b) A grasp hypothesis planned on a resulting box.

Table 3.11: Statistics on stability in simulation and real experiments.

| Object         | labels | In Simulation  |                       | Real Evaluation |          |             |          |           | Dataset   |                     |
|----------------|--------|----------------|-----------------------|-----------------|----------|-------------|----------|-----------|-----------|---------------------|
|                |        | planned grasps | +3 $\Delta$ distances | reached         | lifted   | transported | rotated  | pushed    | stable    | unstable + repeated |
| Cup (white)    | s      | 6-9            | 27-36                 | 2               | -        | -           | <b>8</b> | <b>10</b> | <b>20</b> | <i>19+1</i>         |
|                | u      | 3-11           | 12-48                 | <i>17</i>       | -        | -           | <b>2</b> | -         |           |                     |
| Cup (green)    | s      | 9-10           | 21-26                 | <i>1</i>        | -        | -           | <b>1</b> | <b>19</b> | <b>22</b> | <i>7+15</i>         |
|                | u      | 3-6            | 24-38                 | <i>5</i>        | <i>1</i> | -           | <b>2</b> | -         |           |                     |
| Salt (round)   | s      | 11             | 40                    | <i>5</i>        | <i>1</i> | -           | <b>7</b> | <b>14</b> | <b>28</b> | <i>14+14</i>        |
|                | u      | 5              | 24                    | <i>5</i>        | <i>1</i> | 2           | <b>6</b> | <b>1</b>  |           |                     |
| Bottle (pink)  | s      | 5-8            | 17-19                 | -               | 3        | 7           | <b>7</b> | <b>9</b>  | <b>18</b> | <i>18</i>           |
|                | u      | 2-14           | 14-62                 | 5               | 3        | -           | <b>2</b> | -         |           |                     |
| Bottle (spray) | s      | 5-6            | 15-23                 | 4               | 2        | 3           | <b>7</b> | <b>14</b> | <b>30</b> | <i>26+4</i>         |
|                | u      | 11-16          | 45-65                 | 2               | 7        | 8           | <b>8</b> | <b>1</b>  |           |                     |

was successfully *rotated* in the transported location and finally if the grip was so firm that it could be manually *pushed*. We mark a grasp as stable when it at least reached *rotated* state.

The labels obtained by executing the simulated grasps on the real setup are given in Table 3.11. The column *planned grasps* and  $+3\Delta$  *distances* include the ranges for the number of hypotheses produced by the grasp planner depending on the orientations of the objects and the corresponding ranges extended by the offset distances. Table 3.11 shows the differences between simulation and real experiments in terms of stability evaluation. For example, for the *green cup*, 2 out of 8 unstable and 1 out of 21 stable grasps in simulation, were labeled as stable (being *rotated*) and unstable (being *reached*) respectively in the experiments on the robot. In order to have equal number of stable and unstable grasps, some grasps were repeated as seen in the last column of the table. As seen from Table 3.11, we had the following results: For small and light objects such as the white cup and the green cup, labels in simulation were in general similar to the labels obtained in the real experiments. For the salt bottle, which was relatively small but heavier than the cups, 22.2% of the stable grasp hypotheses were labeled as unstable in the real experiments. This rate was 38.4% for the pink bottle, which was the most deformable object and 30.0% for the spray bottle, which had a less regular shape. From the table, we note that grasps that are supposed to succeed may fail during real experiments due to several reasons which will be discussed shortly. Two example grasps are given in Figure 3.11 to demonstrate that grasps may be labeled differently from the planned ones after execution on the real setup. The two hypotheses that were stable in the simulation were executed on the real setup. The grasps during the real experiments did not yield the same contacts obtained in the simulation and failed.

We have also chosen five different grasps and executed them ten times to see the variance in the labels assigned by the real evaluation. These grasps can be seen in Figure 3.12a. Resulting variances can be seen in Figure 3.12b. Different outcomes were observed during these repeated grasping experiments. While applying the grasp hypothesis  $G_1$  which was labeled as stable in simulation, the object could not even be lifted in any of the repeated experiments and labeled as unstable after all the trials. While executing the hypothesis  $G_2$



Figure 3.11: Examples of stable grasp hypotheses that failed in the real experiments.

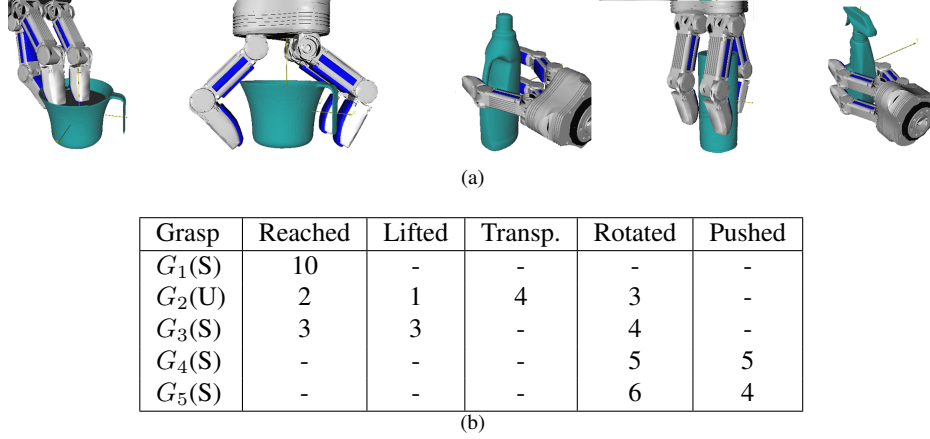


Figure 3.12: Repeated experiments with five different grasps: (a) Grasps  $G_1, G_2, G_3, G_4, G_5$  used in the repeated experiments. (b) Variance of levels of stability for the five grasps. Grasp names are given with the GraspIt label. S represents stable label and U represents unstable label assigned in the simulation.

which was labeled as unstable in simulation, different outcomes such as *reached*, *lifted*, *transported* and *rotated* were observed and therefore stable labels were obtained in some trials. The experiments with the hypothesis  $G_3$  which was stable in simulation, resulted in the states *reached*, *lifted* and *rotated*. Therefore, for  $G_3$  unstable labels were obtained in some trials. The experiments with the stable hypotheses  $G_4$  and  $G_5$  resulted in the states *rotated* and *pushed* and therefore received stable labels. In summary, these experiments show that even the same grasp hypotheses may be classified differently in real experiments, which therefore motivates the need for online assessment.

#### 3.4.4 Analytical vs. Real Grasps

The above analysis exemplifies typical uncertainties and variances emerging for several reasons: not only state-of-the-art simulation environments lack modeling real correspondences, but also repetition of the same grasps may result in variable outcomes. We sketch the major issues immanent in our comparison of GraspIt and our real system:

1. Perception: In our study, we use static poses for known objects. However, even the human inaccuracy in placing an object with the same pose is a factor that can make repetition difficult. In automated machine vision scenarios, e.g., pose estimation of known, or surface estimation of unknown objects, we can assume such accuracy to be even worse.

2. Actuator Control: Similarly to a slight variance in the object’s pose, inaccuracies in the joint positioning of both robot arm and hand, or in dynamic grasp control may have strong effects on the success of a grasp.
3. Contact & Sensor Models: For the sake of efficiency, contact models, e.g., point contacts or soft contacts, may be inaccurately or only partially provided in a simulated environment. In a similar way, simulated models of the tactile sensors have to be reliable to match the real world.
4. Physics & Dynamics: Another key to bring simulation closer to the real world are robust physics and dynamics engines. As in the real world, an object should be dynamically affected by any forces, e.g., exerted by the fingers or gravity.
5. Object Properties: Linked to the dynamics of the scene, knowledge about various object properties is fundamental. On a basic level, this is often approached by mapping discrete material properties, e.g., plastic, metal or glass, to friction cone angles that define the grasp wrench space. Other properties, as deformability, center of mass, or wearout of objects and hands, are more difficult to model.

These considerations emphasize the strengths of a method for online stability assessment in the real world, using learning techniques and close loop methods. Our approach to this problem by using HMMs will be evaluated in the next section.

### 3.4.5 Evaluation and Demonstration

As final experimental results in this chapter, we show that our robot can estimate grasp stability during the execution of the planned grasps. Our robot learns the relations between the sensory data (tactile features and the joint configuration of the hand) and the stability outcomes, which are obtained from executing grasps generated by the box-based grasp planner BADGr. We provide a quantitative evaluation of the learning method based on the collected data. We then demonstrate the feasibility of the approach with two grasp examples. The demonstration is included to better show how the proposed methodology can be integrated in a real robotic system.

To examine the recognition performance, time-series grasp stability assessment is performed using HMMs as described before. We train two HMMs: one for stable grasps and another for unstable grasps. In general, ergodic and left-to-right HMMs had comparable results, hence the experimental results were given with left-to-right structure. Based on the settings mentioned before, the experimental result with 10-fold cross validation run with the best structural parameters (the number of HMM states and the number of mixture model components for each state) is presented. Cross-validation experiments were run on the dataset with 118 stable and 118 unstable grasps described in Table 3.11. A grasp is estimated as stable if the probability of the grasp being stable exceeds the probability of the grasp being unstable, that is,  $P(S = \text{stable}) > P(S = \text{unstable})$ . The probabilities are estimated using the well-known HMM “forward algorithm” to compute the probability of the observed sequence of measurements, assuming equal prior probabilities for stable

and unstable. The classification rate based on 10-fold cross validation is 87.1% with the true positive rate 88.7% and the true negative rate 85.5%. These classification results show that our classifiers can be used to predict the grasp stability during the execution of planned grasps. These rates also support the finding that the approach is sufficient for the variety of objects and grasps used in the experiments.

Before the system can be operated, a training (calibration) process, required for each individual robotic hand, needs to be completed. The calibration process is summarized in Algorithm 1. The grasp generation that was explained in this section follows the steps in the algorithm. While the calibration algorithm is not tied to a particular classification methodology, in the demonstration the HMM classifiers presented previously were used. We first train two HMM classifiers with the grasp data described above using fixed values for the structural parameters. As in the data generation process, we place objects in the workspace of the robot in a known position and orientation with respect to the robot. The operation mode of the demonstration system is described in Algorithm 2. During the operation, we apply two grasp hypotheses generated by the planner. These hypotheses were not used in the training process.

---

**Algorithm 1** Calibration mode.

---

- 1: Choose a suitable grasping strategy for object  $O$ .
  - 2: **for**  $i = 1$  to  $n$  **do**
  - 3:   Preshape the hand
  - 4:   Grasp object  $O$  according to the chosen grasping strategy.
  - 5:   Record tactile and joint configuration data during the grasp.
  - 6:   Manipulate the object  $O$  along a predetermined path.
  - 7:   Record object motion relative to the hand  $\Delta T$ .
  - 8:   **if**  $\Delta T > 0$  **then**
  - 9:     Grasp  $i$  is unstable.
  - 10:   **else**
  - 11:     Grasp  $i$  is stable.
  - 12:   **end if**
  - 13: **end for**
  - 14: Using recorded data from each grasp  $i$ , train a classifier  $C$ .
- 

Figure 3.13 shows snapshot images from the operation of the system. The robot is attempting to grasp a bottle by first placing the hand in a preshape position given by the planner, as shown in Figure 3.13a. Then, the fingers are closed. After a steady state is reached, the stability of the grasp is estimated. The closed grasp is shown in Figure 3.13b with the corresponding tactile measurements in Figure 3.13c. Based on the obtained sensory data, the grasp is predicted to be unstable, with the log-likelihood ratio  $\log \frac{P(\text{unstable})}{P(\text{stable})}$  of the two models being  $191.1270 > 0$ , indicating unstable grasp. Now, in order to demonstrate that the failure was correctly predicted, instead of regrasping, the robot is nevertheless commanded to lift the object. The object drops as shown in Figure 3.13d, demonstrating the ability to correctly recognize an unsuccessful grasp. Next, to demonstrate that the stable

**Algorithm 2** Operation mode.

- 
- 1: Generate a grasp using our grasp planner.
  - 2: Preshape the hand.
  - 3: Grasp the object by closing the fingers.
  - 4: Evaluate classifier using the acquired sensor data.
  - 5: **if**  $P(S = \textit{stable}) > P(S = \textit{unstable})$  **then**
  - 6:   Lift the object.
  - 7: **else**
  - 8:   Go to 1.
  - 9: **end if**
- 

grasps are also successfully recognized, another grasp generated by the same grasp planner is shown in Figure 3.13e. The closed grasp and the corresponding tactile measurements are shown in Figs. 3.13f and 3.13g. Based on the measurements, the grasp is predicted to be stable, with the difference across log-likelihoods of the two models being  $-537.7687 < 0$ , indicating a stable grasp. Lifting and rotating the object around demonstrates this in Figure 3.13h, which concludes the demonstration.

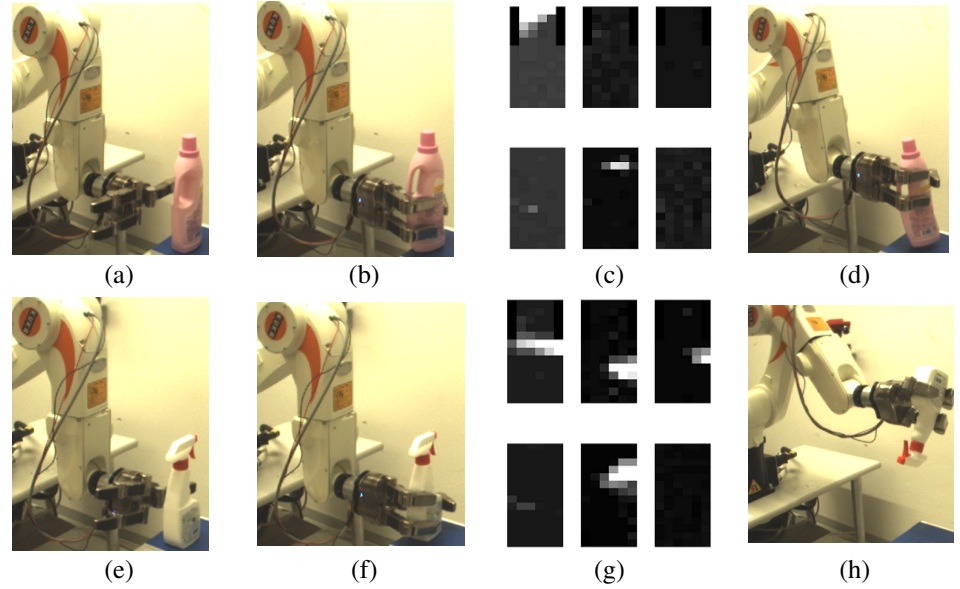


Figure 3.13: Operation of the system. First row shows an unsuccessful grasp, second row shows a successful grasp: (a,e) Hand in a preshape position, (b,f) Closed grasp, (c,g) Tactile measurements, (d) The object dropped while lifting, (h) Lifting and rotating the object successfully.

### 3.5 Summary and Discussion

In this chapter we demonstrated how grasp stability can be assessed based on sensory data using machine learning techniques while grasping an object before the object is further manipulated. The learning framework takes into account object shape, approach vector, tactile data and joint configuration of the hand. The methods were evaluated both on simulated and real data. Both one-shot and temporal learning techniques were implemented and evaluated.

The temporal information was found to increase generalization capabilities in that a smaller number of training examples was needed and that the generalization performance to new objects was slightly increased. These come with the cost of increased computational complexity. One focus of the experiments was to study prediction capabilities of the proposed methods for known objects. How the system can cope with unknown objects, i.e., objects that have not been used in the training step, was also studied.

The results showed that while the classification rate is lower than with known objects it is still possible to make useful predictions of the grasp stability on unknown objects. In summary, the experimental results show that tactile measurements allow assessment of grasp stability. The aim was not a perfect discrimination between successful and unsuccessful grasps but rather a measure of certainty of grasp stability. This also means that a system may be built to reject some stable grasps while having fewer unstable grasps classified as stable ones.

We have also shown how a grasp planner can be integrated with a probabilistic technique for grasp stability assessment in order to improve the hypotheses about suitable grasps on different types of objects. An important contribution of the presented work is an implementation and evaluation of the approach on a real robot system. The relation between analytical stability in simulation and real grasps was analyzed under consideration of a range of uncertainties. In addition, we have applied an evaluation strategy based on different stability levels for the probabilistic learning technique. Experimental evaluation showed the feasibility and strength of the integrated approach. Results proved that we can estimate grasp stability with high accuracy when executing planned grasps. We also showed that some of the planned grasp hypotheses that were stable in simulation might fail during real execution, which motivates the need for online stability assessment.

We have based our estimations on tactile sensing so far. However, using tactile sensing alone classifiers may not always distinguish stable and unstable grasps. For example, similar readings that correspond to different stability outcomes might be received from different parts of the objects. In such cases, including visual information, such as how the object is approached, can improve the predictions on grasp stability. In the next chapter, we study combining visual and tactile sensing to assess grasp stability.



## Chapter 4

# Learning Grasp Stability through Visual and Tactile Sensing

In the previous chapter it was shown that tactile feedback helps assessing grasp stability and that a robot can learn to discriminate between tactile imprints of stable and unstable grasps. However, for stability assessment, tactile feedback alone can also be ambiguous. For example, similar tactile readings might correspond to different stability outcomes. In this chapter vision-based pose information is integrated into the model to further disambiguate grasp configurations. The aim is to infer grasp stability from the tactile imprints and the object-gripper configuration available *before* lifting an object.

### 4.1 Background and Motivation

We aim at designing a stability predictor that is independent of the pose of an object. For this reason, we do not predict stability from the manipulator and object poses directly. Instead, we base our predictions on the *relative* object-manipulator pose. Object-relative manipulator configurations allow our system to encode notions such as “grasping a bottle from the side is better than grasping it from the top.” However, stability will often not only depend on the relative object-gripper configuration, but also on the absolute orientation of the object. When an elongated object lies on a flat surface, it is generally better to grasp it close to its center of mass. Yet, if the object is standing, grasping it near its tip is acceptable. As a result, we also base our predictions on the angle between the gripper’s approach vector and a direction aligned with gravity.

Through visual and proprioceptive feedback, our robot is able to acquire object and gripper poses in real time. Gripper poses are simply obtained from the kinematics of the robot. Obtaining object pose is more challenging. An object will often move while the robot is closing its hand to grasp it. Therefore, the robot needs to compute the pose of the object *after* having closed the hand around it. This computation is made difficult by the partial object occlusions effected by the hand. We address this issue by tracking the movement of the object for the complete duration of the grasp, as described in Section 2.3.1. Tracking

is more robust than performing pose estimation at the end of the movement, since it uses all previous frames to make an estimate. Our aim however is not to get perfectly accurate pose information, but rather a rough idea of how the object is approached.

We predict grasp stability with object-specific classifiers trained to discriminate between percepts that lead to stable or unstable grasps for a specific object. Our agent learns an empirical representation of pose- and touch-conditional grasp stability probability. This model is learned from a set of examples denoted by

$$Z = \{(x_i, S_i)\}_{i=1, \dots, N} \quad (4.1)$$

where each pair  $(x_i, S_i)$  is composed of perceptual readings  $x_i \in \mathbb{R}^d$  (pose and touch) and a binary stability label  $S_i \in \{\text{stable}, \text{unstable}\}$ . Perceptual data are read during the execution of a grasping plan, shortly after the agent closed the manipulator's fingers around the object, but before any attempt to lift or transport the object.

The probability of pose- and touch-conditional grasp stability is modeled with kernel logistic regression (KLR). Logistic regression is a widely used technique to model class probabilities. It directly learns the class probabilities and therefore it can naturally yield confidence of class prediction. The probabilistic nature of this modeling is preferable for us, since we would like to provide confidence about the results. We will study KLR models based on different modalities and we would like to see which model will be more confident in predicting stability.

In the next section, we give an intuitive explanation of KLR [116] applied to our problem.

## 4.2 Modelling Framework

KLR models the stability probability of a grasp characterized by a perceptual vector  $x$  with the help of a weighted sum of the similarities between  $x$  and each vector in the training dataset  $Z$ . The weights associated to stable grasps will generally be positive, while those associated to unstable grasps will be negative. If  $x$  resembles percepts of  $Z$  that lead to stable grasps, its probability of stability will thus be high. In order to restrict values to the  $[0, 1]$  interval, KLR models probabilities by plugging the weighted sum described above into the logistic function  $f(z) = \frac{1}{1+e^{-z}}$ , which smoothly grows from 0 to 1 as its argument varies from minus infinity to infinity. Weights are usually chosen to maximize the probability of the training set.

Formally, we model the probability of pose- and touch-conditional grasp stability as

$$P(S = \text{stable}|x; v) = \frac{1}{1 + \exp \left\{ - \sum_{i=1}^n v_i \mathcal{K}(x, x_i) \right\}} \quad (4.2)$$

where  $P(S = \text{stable}|x)$  is the probability of success of a grasp characterized by the tactile and pose vector  $x$ ,  $\mathcal{K}$  is a kernel function that models the similarity between two perceptual readings and  $v$  is a weight vector chosen to maximize the regularized stability probability

of the data

$$-\sum_{i=1}^n \log P(S_i|x_i; v) + c \text{trace}(vKv^T) \quad (4.3)$$

where  $K$  is the kernel Gram matrix, with  $K_{ij} = \mathcal{K}(x_i, x_j)$ , and  $c$  is a constant. This problem can be solved, e.g., with Newton's method [116]. The constant  $c$  was chosen by cross-validation during the experiments which is explained in the next section.

#### 4.2.1 Kernel Function

As explained above, we consider perceptual signals in the form of tactile readings, relative object-gripper configurations, and an angle that represents the tilt of the hand's approach vector relative to gravity.

A vector  $x$  representing perceptual observations can be written as  $x = (t, g, a)$  where  $t$  is the tactile data, the object-relative gripper pose is denoted by  $g$ , and  $a$  is the angle between the approach vector and the vertical.

The tactile data  $t$  is represented based on image moments as  $(c_x, c_y, \frac{\mu_{2,0}}{m_{0,0}}, \frac{\mu_{0,2}}{m_{0,0}}, \frac{\mu_{1,1}}{m_{0,0}})$  where  $(c_x, c_y) = (\frac{m_{0,1}}{m_{0,0}}, \frac{m_{1,0}}{m_{0,0}})$  and  $\mu_{p,q} = \sum_x \sum_y (x - c_x)^p (y - c_y)^q f_{(x,y)}$ . These moment-based features were normalized to zero mean and unit variance.

The kernel  $\mathcal{K}$  is defined as

$$\mathcal{K}(x_1, x_2) = \mathcal{K}_t(t_1, t_2) \mathcal{K}_g(g_1, g_2) \mathcal{K}_a(a_1, a_2). \quad (4.4)$$

When classifying on tactile imprints or visual pose exclusively, the kernel is redefined as  $\mathcal{K}(x_1, x_2) = \mathcal{K}_t(t_1, t_2)$  or  $\mathcal{K}(x_1, x_2) = \mathcal{K}_g(g_1, g_2) \mathcal{K}_a(a_1, a_2)$  respectively. The kernel function  $\mathcal{K}_t$  simply corresponds to a multivariate isotropic Gaussian function

$$\mathcal{K}_t(t_1, t_2) = \mathcal{N}(t_1; t_2, \sigma_t^2 \mathbf{I}), \quad (4.5)$$

where  $\sigma_t$  is a bandwidth parameter. During the experiments an optimal bandwidth was computed by cross-validation, which is described in the next section.

An object-relative gripper pose is composed of a 3D position and 3D orientation. We define the gripper pose kernel  $\mathcal{K}_g$  as the product of a position and an orientation kernel. Let us denote the decomposition of a pose  $g$  into position and orientation by  $p$  and  $o$  respectively. We define  $\mathcal{K}_g$  with

$$\mathcal{K}_g(g_1, g_2) = \mathcal{N}(p_1; p_2, \sigma_p^2 \mathbf{I}) \frac{e^{\sigma_o o_1^T o_2} + e^{-\sigma_o o_1^T o_2}}{2} \quad (4.6)$$

where  $\mathcal{N}$  is a trivariate isotropic Gaussian kernel, the fraction corresponds to a pair of antipodal von-Mises Fisher distributions (Gaussian-like distribution on the rotation group [41, 106]). The bandwidths  $\sigma_p$  and  $\sigma_o$  were fixed by inspection during the experiments.

The kernel function  $\mathcal{K}_a$  corresponds to a Gaussian function

$$\mathcal{K}_a(a_1, a_2) = \mathcal{N}(a_1; a_2, \sigma_a^2 \mathbf{I}), \quad (4.7)$$

where  $\sigma_a$  is a bandwidth parameter. During the experiments an optimal bandwidth was computed by cross-validation, which is described in the next section.



Figure 4.1: The four objects used in the experiments: The box, the oval bottle, the cylindrical bottle and the spray bottle, respectively from left to right.

### 4.3 Experimental Evaluation

Our robot learns an empirical representation of pose- and touch-conditional grasp stability probability. To achieve this learning goal, we let the robot explore chosen objects around given grasp configurations to learn the mapping between the percepts extracted during grasp executions and the success outcomes of the grasps. In this section, we present how this exploration was performed, the resulting perceptual grasping data, how we evaluated our learning method and discuss the classification results based on tactile feedback alone, visual feedback alone and both tactile and visual feedback together to study the contributions of these percepts for our stability estimation purpose.

#### 4.3.1 Exploration of Objects around Given Grasps

In our grasping experiments, we used the four home-environment objects shown in Figure 4.1: a box, and three bottles of different shapes which we call as the spray bottle, the oval bottle and the cylindrical bottle. We chose these objects because of several reasons. These objects were large enough for the Schunk hand to grasp with three fingers. The objects were similar in weight and had different geometrical features and deformation properties. For example, the spray bottle had a less regular shape compared to the other objects, the cylindrical bottle was the most deformable object and the box was the least deformable among all the objects. The pose tracking was equally successful for each object during grasping and manipulation experiments.

In order to learn the correlation between the tactile and visual measurements and the success/failure in grasping, we let our robot gather perceptual data based on exploration of the objects by grasping and lifting. However the exploration cannot be done exhaustively. To make the experiments feasible, we focused on certain regions of objects and generated a set of grasps to execute on these regions. Grasps for a region were generated by randomly sampling a distribution centered around an initial grasp  $g_i$ . Therefore a random grasp  $\hat{g}_r$  was obtained from  $P(g_r|g_i) \propto \mathcal{K}_g(g_i, g_r)$ . Resulting grasps were distributed a few

centimeters/degrees away from  $g_i$ . In this way, by executing multiple grasps in a region the robot could also learn the effect of small differences in hand positioning. In a real-world scenario it is important for the robot to learn the relations between the perceptions and the stability outcome in a region of an object rather than in a single place, because it is not reasonable to expect that the robot will always be able to grasp an object exactly at the same place.

Initial grasps were defined around the middle parts of the objects as shown in Figure 4.2a, 4.3a, 4.4a and 4.5a. These initial grasps were parametrized by the pose of the hand with respect to the object. Each grasping experiment during the exploration of the objects in the neighborhood of the initial grasps was then run in the following way: An object was placed at an arbitrary position reachable by the robot. The standing/lying configuration of the objects also varied. The robot estimated the pose of the object and executed a random grasp  $\hat{g}_r$  around  $g_i$ . Once the hand had stopped closing the fingers, the robot recorded the tactile imprints and the pose of the object. As an example, the tracked object pose at the end of the grasp was depicted with the blue wireframe in Figure 4.2b, 4.3b, 4.4b and 4.5b for each object. The final pose of the hand relative to the object as depicted in Figure 4.2d, 4.3d, 4.4d and 4.5d, the angle between the approach vector and the vertical, and the corresponding tactile features were extracted. Example tactile readings obtained at the end of grasp executions are shown in Figure 4.2e, 4.3e, 4.4e and 4.5e. At the end of the grasp executions, the robot obtained the object-relative hand poses by comparing the vision-based object pose estimates to the known hand poses. The robot then attempted to lift the object by 5 cm. If the object slipped or rotated in the hand while being lifted, the grasp was marked as unstable. If lifting could be achieved robustly without any slippage or rotation, the grasp was marked as stable. An unstable grasp is shown in Figure 4.6 where the object rotated during lifting. The robot was also able to track the object pose during lifting. Example pose tracking results after lifting can be seen in Figure 4.6, 4.2c, 4.3c, 4.4c and 4.5c. Therefore, the robot could extract success labels by comparing the resulting translation/rotation, which was measured based on the differences in the tracked poses, to the applied 5 cm translation along the vertical during lifting.

In total 584 grasps were executed, i.e., 134 for the box, 170 for the oval bottle, 138 for the cylindrical bottle and 142 for the spray bottle, during the exploration process explained above. Half of these grasps were stable and the other half were unstable. All the executed grasps for each object can be seen in Figure 4.7 to 4.10. In these figures all the object-relative hand grasping configurations were plotted by using a simplified hand model, which can be seen together with the whole hand model in Figure 4.7b. In order to reveal more details about the distributions of these grasping configurations we also plotted the heatmaps of the 2D y-z projections of the hand positions relative to the object in Figure 4.7, 4.8, 4.9 and 4.10. These heatmaps were obtained based on estimates of the probability density function of hand positions. Both from the heatmaps and the grasping configurations with the simplified hand models, it can be seen that regions covered by the hand positions of stable and unstable grasps overlap. As also supported by the heatmaps we can list the following general observations for the objects: When the box and the detergent bottle were standing, increase in the distance between the object and the hand yielded unstable grasps. This can be seen along z-axis in Figure 4.7d, f and Figure 4.8c, e. The increase in that

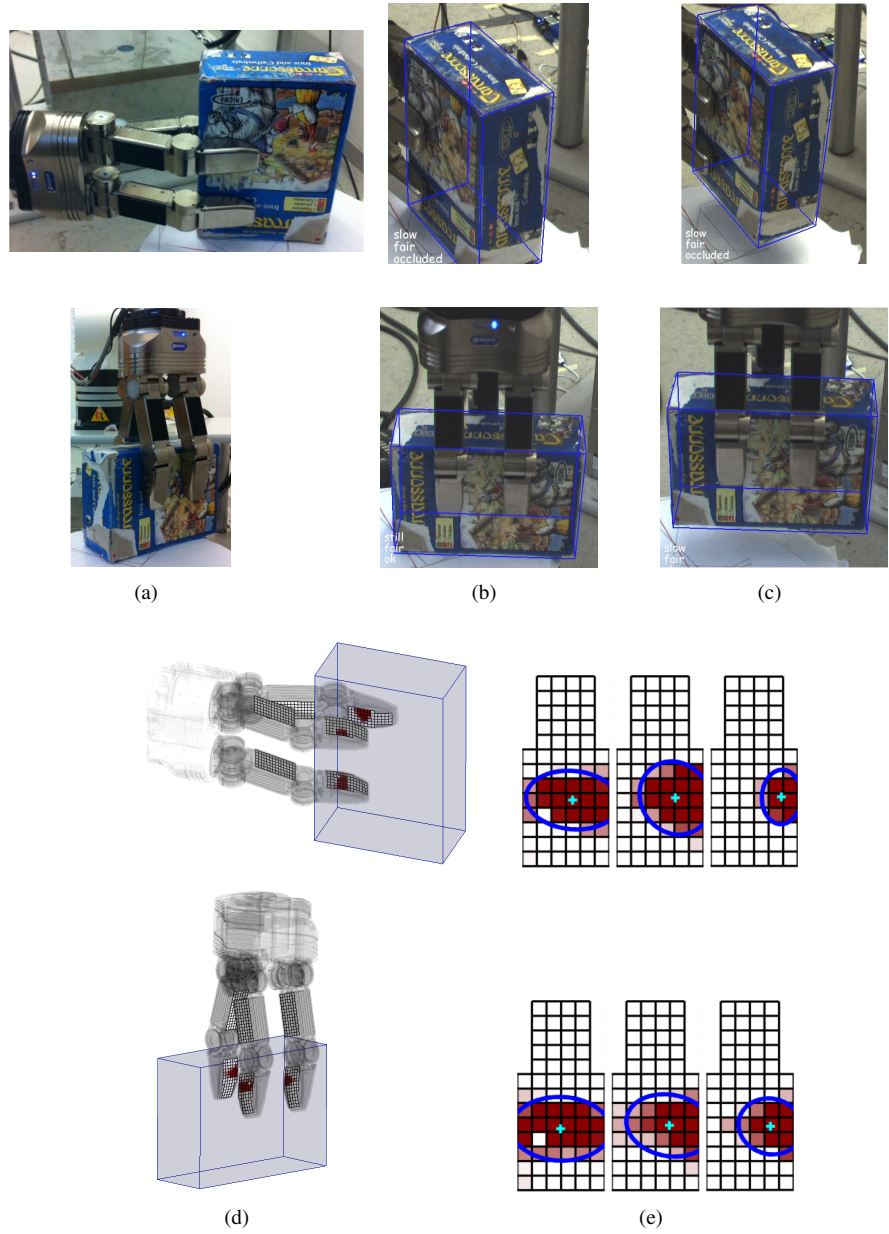


Figure 4.2: Illustration of the initial grasp and the information extracted during grasping experiments: (a) The initial grasp in lying and standing object configurations. (b) The tracked object pose in those grasps is depicted with a blue wireframe before lifting. (c) The tracked pose after lifting the object by 5 cm. (d) The reconstructed object-relative hand configuration obtained from the object pose and the hand pose in the robot's frame. (e) The tactile readings on the distal sensor arrays together with the illustrations of the chosen representations that correspond to the pressure distributions and centroids.

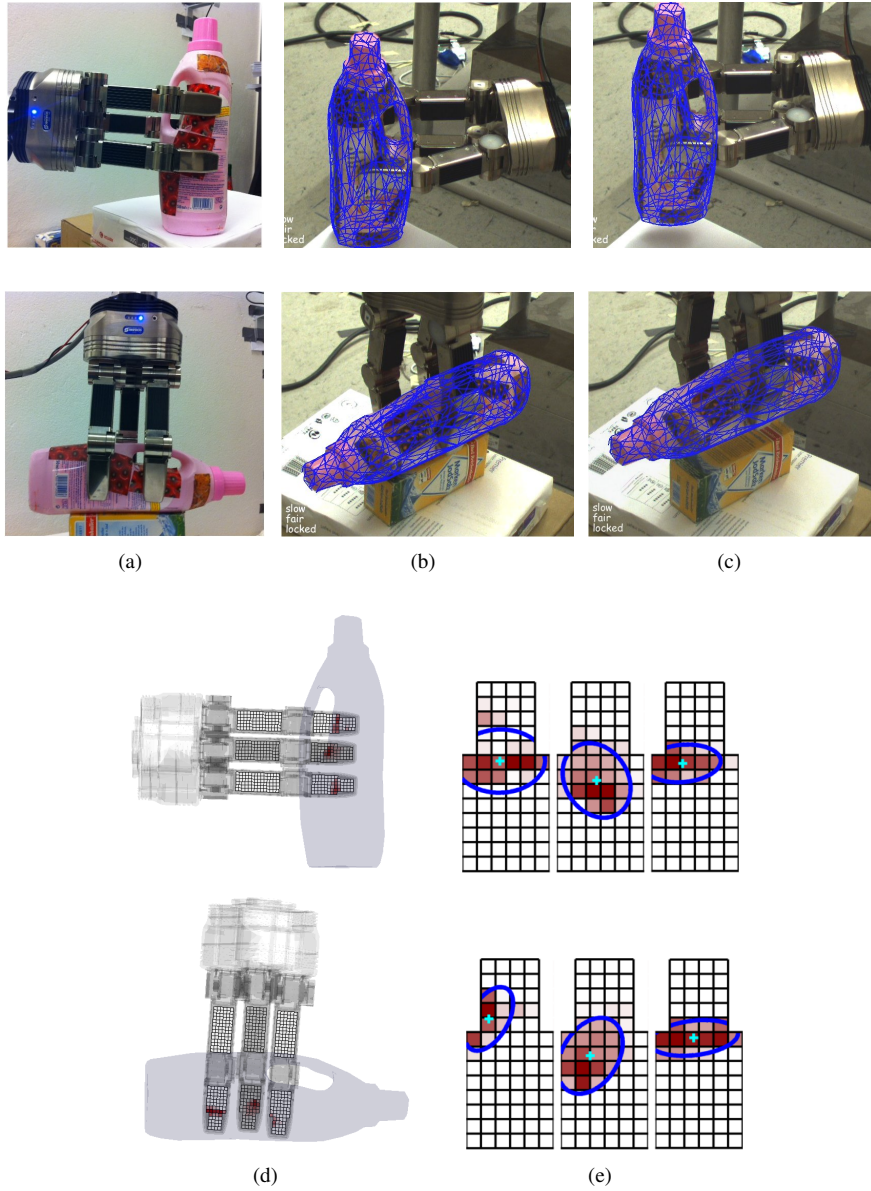


Figure 4.3: Illustration of the initial grasp and extracted information during grasping experiments on the oval bottle. See Figure 4.2 for details.

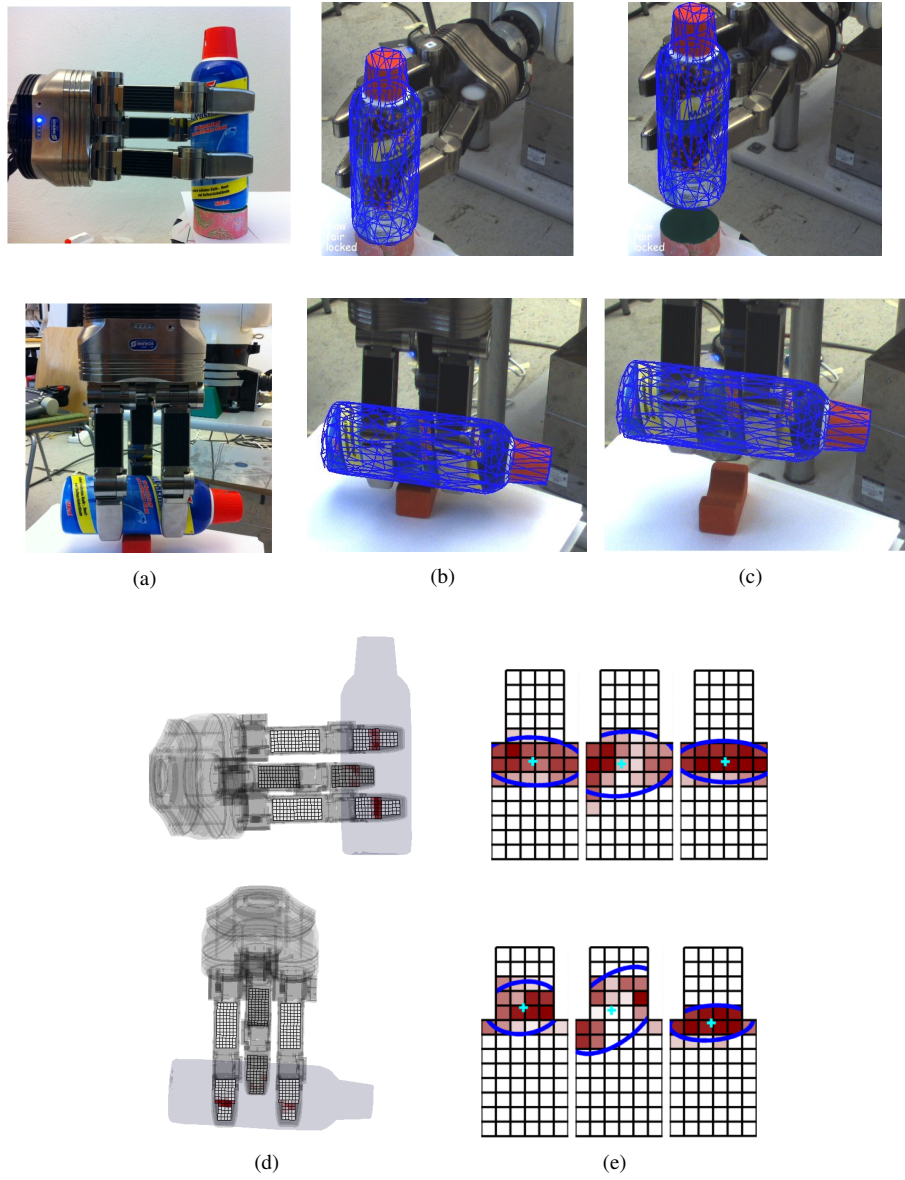


Figure 4.4: Illustration of the initial grasp and extracted information during grasping experiments on the cylindrical bottle. See Figure 4.2 for details.

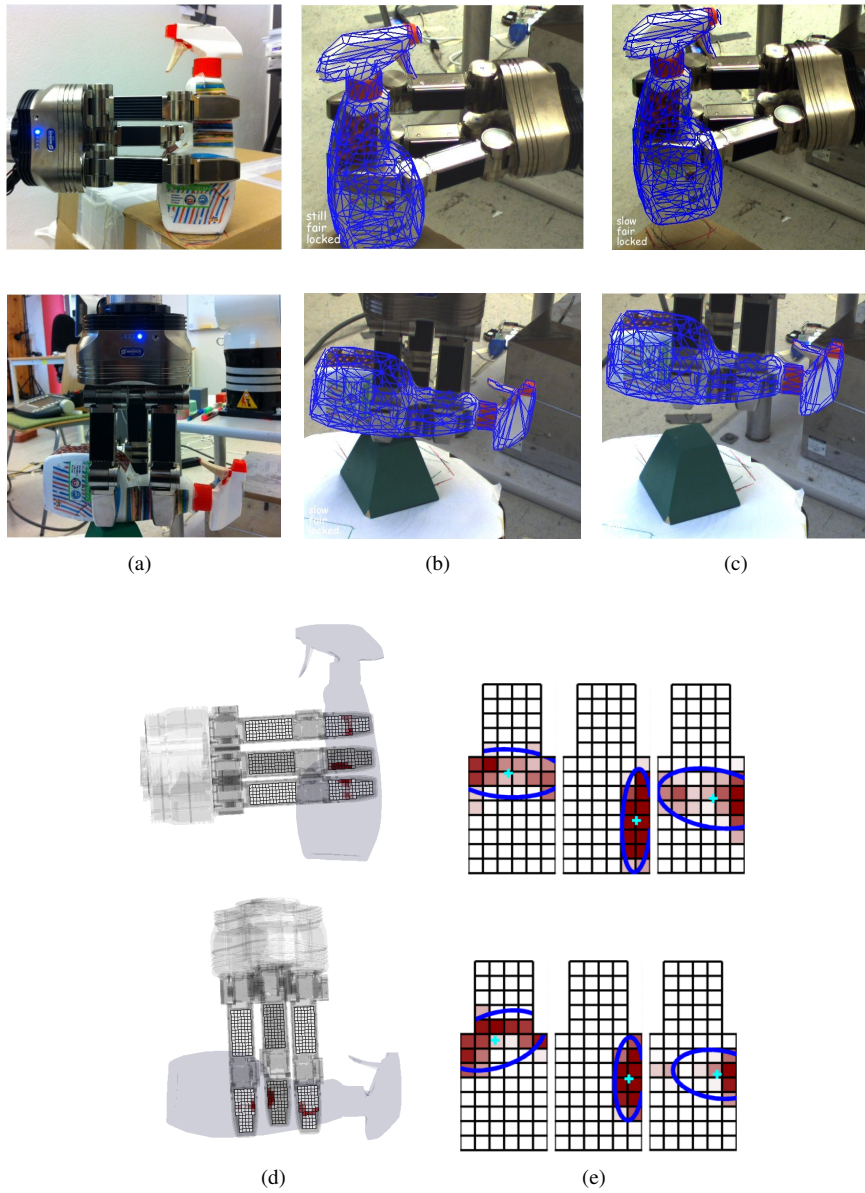


Figure 4.5: Illustration of the initial grasp and extracted information during grasping experiments on the spray bottle. See Figure 4.2 for details.

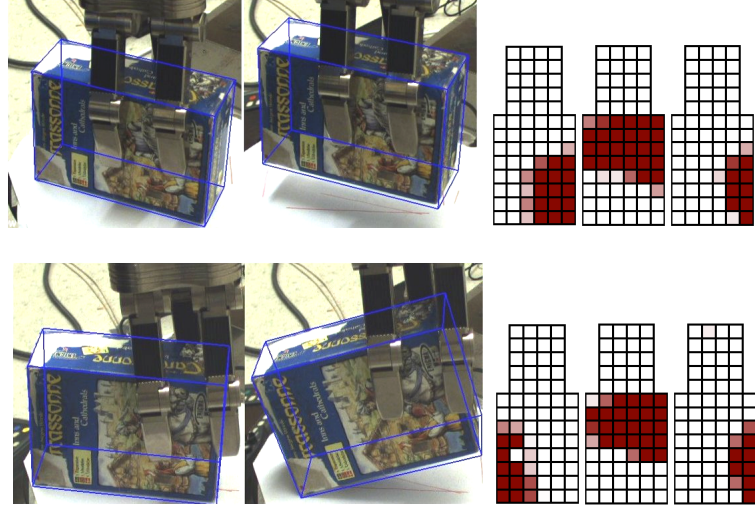


Figure 4.6: Examples of grasps with the pose tracking results and the associated tactile readings obtained from the box. The tracked object poses were marked with blue wire-frame and they were used to obtain success labels after the object was lifted.

distance resulted in grasping by the tips of the fingers and therefore led to unstable grasps. When the objects were lying, the grasps applied at the extremities of the objects tended to be unstable, which can be seen along y-axis in Figure 4.7c, e and Figure 4.8b, d. For the oval bottle the distance between the hand and the object influenced the outcome in standing and lying configurations, which can be seen along y-axis in Figure 4.9b to e. For the spray bottle there was not any distinctive difference between stable and unstable grasps in terms of hand positions when the object was lying as seen in Figure 4.10b, d. However, in standing configuration, around the bottom end of the object small increase in the distance between the hand and the object produced unstable grasps, which can be seen along y-axis in Figure 4.10c, e.

### 4.3.2 Evaluation Method

We evaluated our learning method by computing classification rates based on the rule: A grasp characterized by  $x$  was predicted to be stable if  $P(S = \text{stable}|x) > \frac{1}{2}$ . Classification rates were computed with a repeated double cross validation procedure [40]. The inner loop of the cross validation was run to determine the best parameters for the classifier. The outer loop was run to obtain an unbiased estimate of the predictive performance with the parameters coming from the inner loop. In the outer loop, the data set was randomly split into a train set and a test set. The best model parameters for the train set were calculated by performing an inner cross-validation with this train set. These parameters from the inner

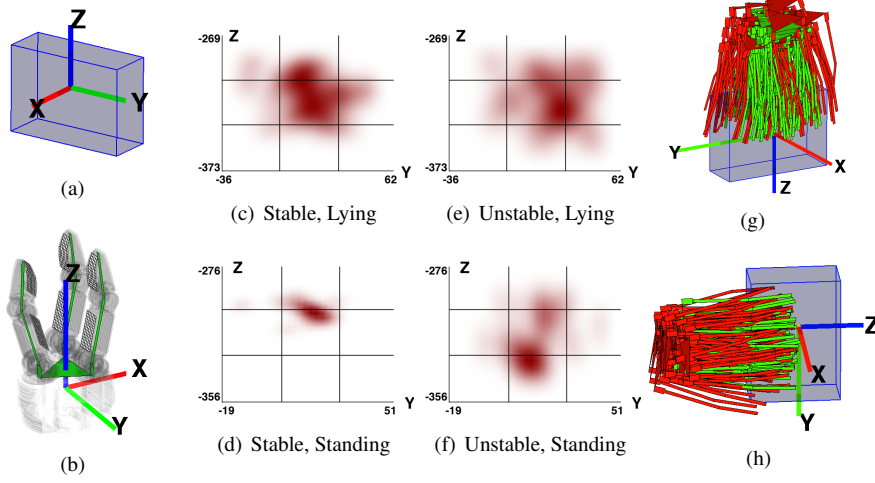


Figure 4.7: The distribution of all the executed grasps when the box is lying and standing: (a) The box model together with its reference frame. (b) The Schunk hand together with its simplified model in green. (c-f) The heatmaps of 2D projections of the object-relative hand positions with respect to the object frame are seen separately for stable and unstable grasps in lying and standing object configurations. (g), (h) All the executed grasps represented with the simplified hand model when the object is lying and standing. Green hand models represent the stable hand configurations with respect to the object and the red ones represent the unstable grasps.

loop were used to calculate the error rates in the outer loop. This procedure was repeated in a loop with different random splits providing more test sets so that the prediction performance and its variability could be better estimated. In this way, model evaluation was done independent of the model selection and the prediction based on the outer loop test samples gave an unbiased estimate of the error rate.

We set the repetition number to 100, since higher values for this parameter did not provide improved results with a lower variance. The number of folds in the inner and the outer loops was set to 5 to use 20% of the dataset as a test set in each loop. The inner cross-validation loop was run for several values of the bandwidth parameters  $\sigma_t$  and  $\sigma_a$  defined in Eq. (4.5) and (4.7), and several values of the regularization constant  $c$  defined in Eq. (4.3). The bandwidth parameters  $\sigma_p$  and  $\sigma_o$  defined in Eq. (4.6) were fixed by inspection. We trained 3 types of classifiers: classifiers based on (1) tactile feedback alone, (2) visual feedback alone, and (3) both tactile feedback and visual feedback together. We compared their classification results to examine the relevance of tactile and visual feedback for stability estimation. Each classifier is trained and evaluated with the data collected for a single object due to the fact that pose parameters cannot be shared across objects.

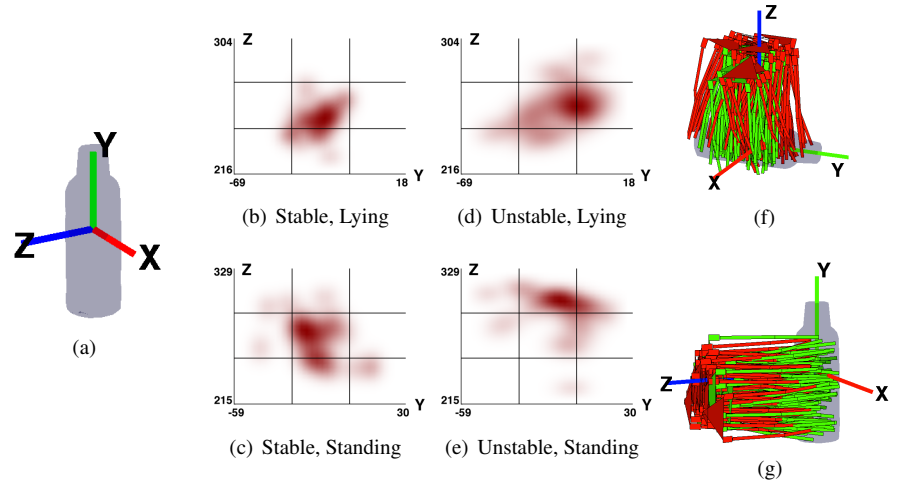


Figure 4.8: The distribution of all the grasps executed on the cylindrical bottle when it is lying (Figure (b), (d) and (f)) and standing (Figure (c), (e) and (g)). See Figure 4.7 for details.

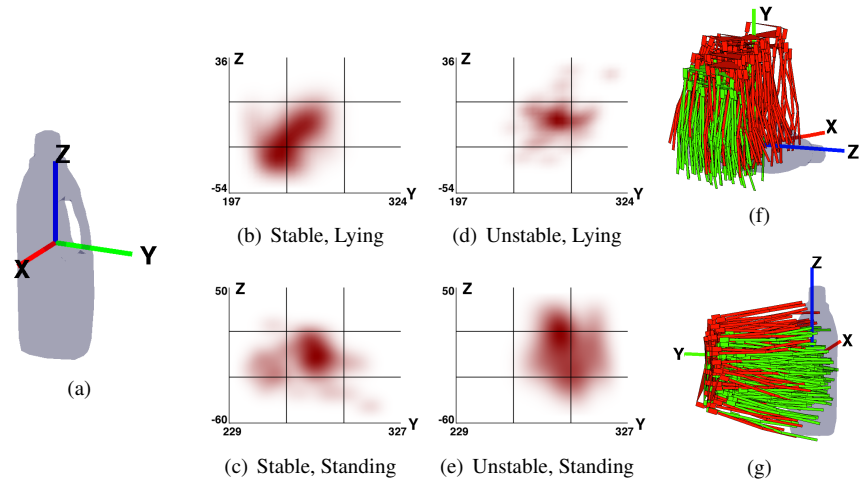


Figure 4.9: The distribution of all the grasps executed on the oval bottle when it is lying (Figure (b), (d) and (f)) and standing (Figure (c), (e) and (g)). See Figure 4.7 for details.

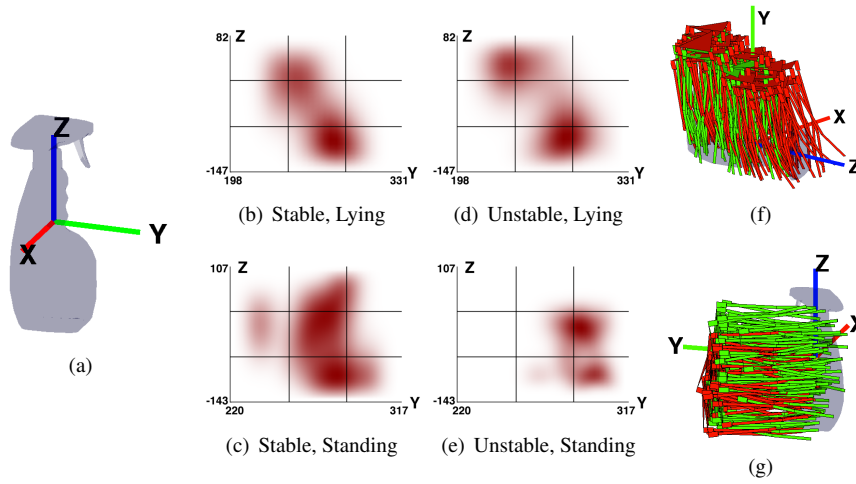


Figure 4.10: The distribution of all the grasps executed on the spray bottle when it is lying (Figure (b), (d) and (f)) and standing (Figure (c), (e) and (g)). See Figure 4.7 for details.

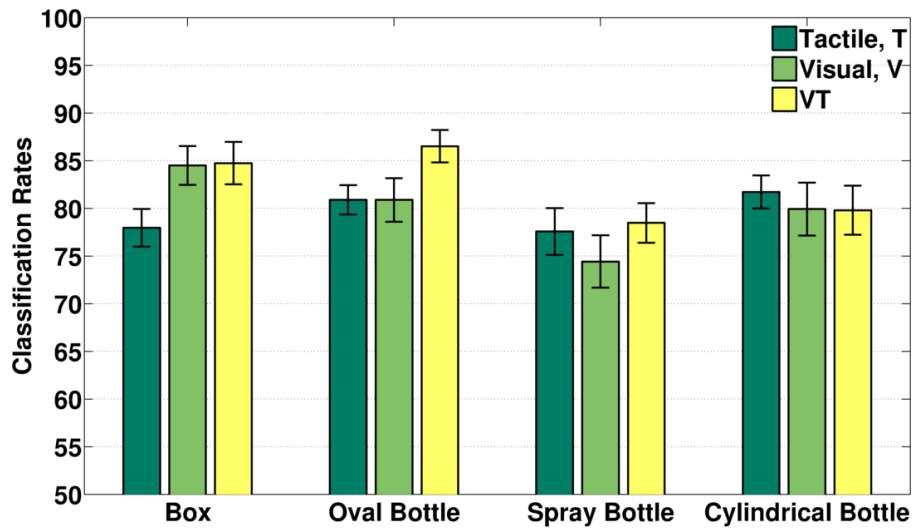


Figure 4.11: Classification rates from the cross validation experiments of three variants of the stability classification model for the objects.

### 4.3.3 Classification Results

In this section we present the stability classification results obtained on the chosen objects. Our learning method was evaluated by applying repeated double cross validations. The classification rates and the average Receiver Operating Characteristic (ROC) curves for the outer loops were presented in Figure 4.11 and Figure 4.12.

Each object produced different types of results in terms of classification rates. For the oval bottle, tactile and visual data together provided higher classification rate than both visual data alone and tactile data alone. For the box, visual and tactile data together yielded similar classification rate to visual data alone but better rate than tactile data alone. For the spray bottle visual and tactile data together provided similar classification rate to tactile data alone but better than visual data alone. For the cylindrical bottle, all three classifiers performed similarly. In summary, visual and tactile data together led to at least as good classification rates as visual or tactile data alone. This result motivates using the two modalities together, since in advance it is not known which modality would be more useful. Furthermore, as seen from the ROC curves, visual and tactile data together provided low false positive rates compared to using only visual or tactile data for all the objects. Having as few false positives as possible is a desired property for our classifiers, since false positive means classifying an unstable grasp as a stable grasp and confirming to lift the object. But the object might fall and break, since the grasp was in fact unstable.

So far, we demonstrated the accuracy of the classifiers. In addition, to compare the classifiers in terms of their confidence in the results, we have also plotted distributions of probability scores  $P(S = \text{stable}|x)$  for stable and unstable test samples for each classifier in Figure 4.13. The overlapping filled regions correspond to classification errors. As supported by these distributions, for the box and the cylindrical bottle, stable and unstable cases were separated with higher confidence by including visual data rather than only using tactile data. For the oval bottle, while the tactile data alone provided higher confidence than the other two cases, visual and tactile data together provided better confidence than the visual data alone. There was not any significant difference in terms of confidence between the three classifiers for the spray bottle. The figure also motivates the use of a probabilistic approach for our classification problem, since being able to measure certainty in classification is useful to prevent failures. As seen from the above results in terms of classification rates or the classifier confidence, the relevance of tactile and visual data for stability prediction varied between objects. Compared to tactile data, visual data provided more distinctive confidence for the objects that were with more regular shapes in the explored region, such as the box and the cylindrical bottle.

As seen from the above results, tactile and/or visual data can be more useful for different objects and it is not straightforward to determine if tactile and/or visual data can provide more confident or correct predictions for any given object beforehand. The main result is that using both visual and tactile information to predict grasp success produces accuracy/confidence that is similar to or higher than using either of the two modalities alone for all the objects. This finding suggests that it is beneficial to exploit both modalities so that they support each other to be able to deal with different cases.

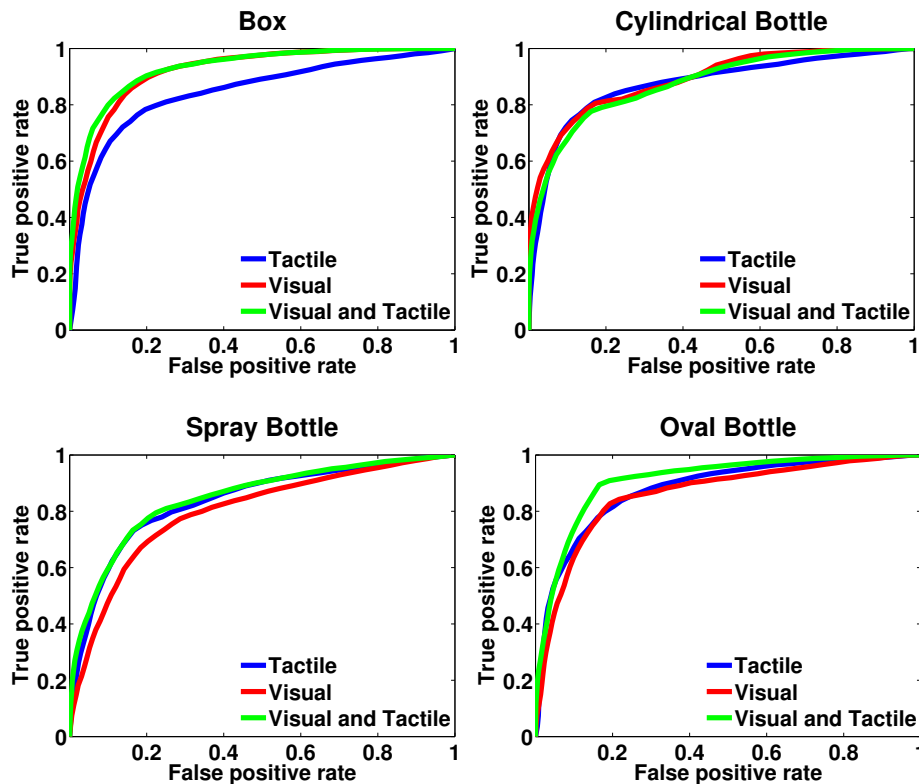


Figure 4.12: Average ROC curves for the three variants of the classification model: The plots show true positive rates against false positive rates at various discrimination threshold values.

#### 4.3.4 Experiments with Noisy Pose Estimates

In our setup the objects cover a large fraction of the camera’s field of view. If the camera were to cover a larger field, pose estimation would be less accurate, and pose-based classification would be less reliable. In addition to that, for smaller objects, fingers will occlude a larger relative area, and pose parameters will become more noisy. In order to study the feasibility of our method with less reliable visual measurements, we evaluated our models with noisy pose estimates.

One way to produce noisy visual data is to vary distance between the camera and the object. This can be done by either moving the object or the camera. In the former case, the object may not always be reachable for the robot. The latter requires placing the camera in different locations and renewing the calibration between the camera and the robot for

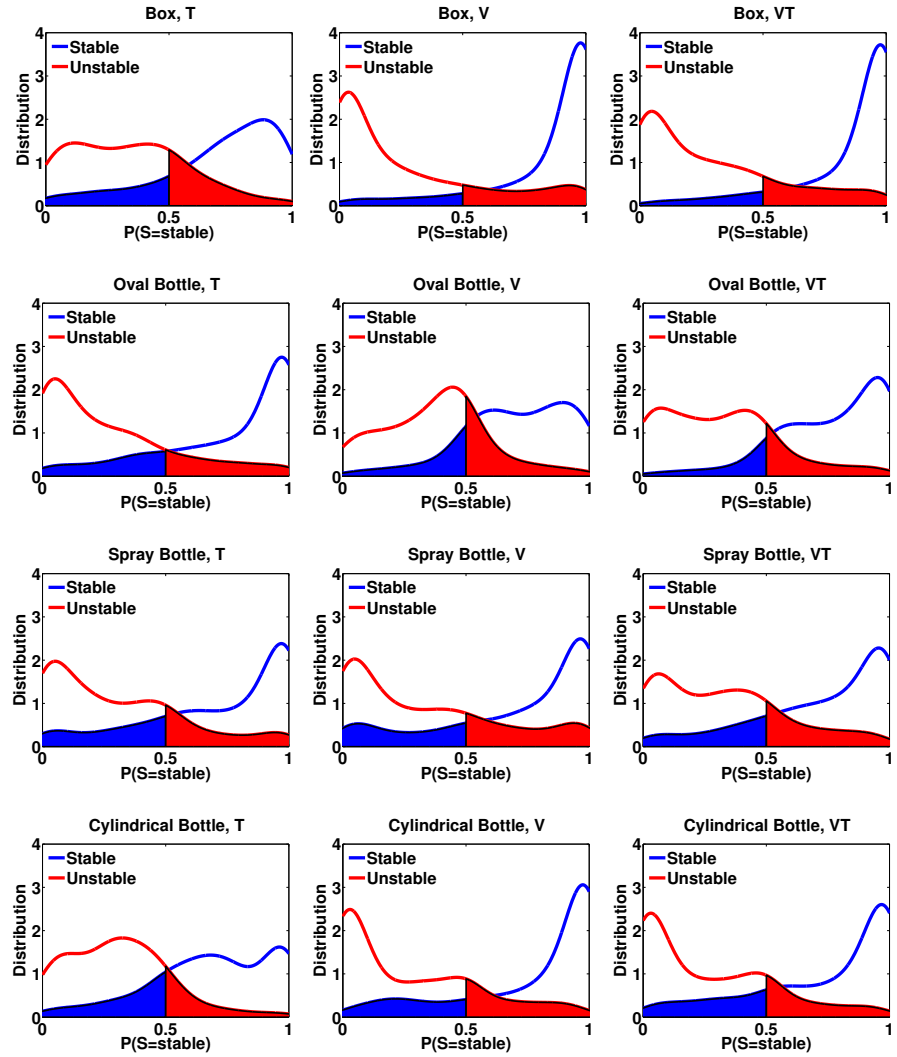


Figure 4.13: Distributions of the probability scores of the classifiers for test samples.

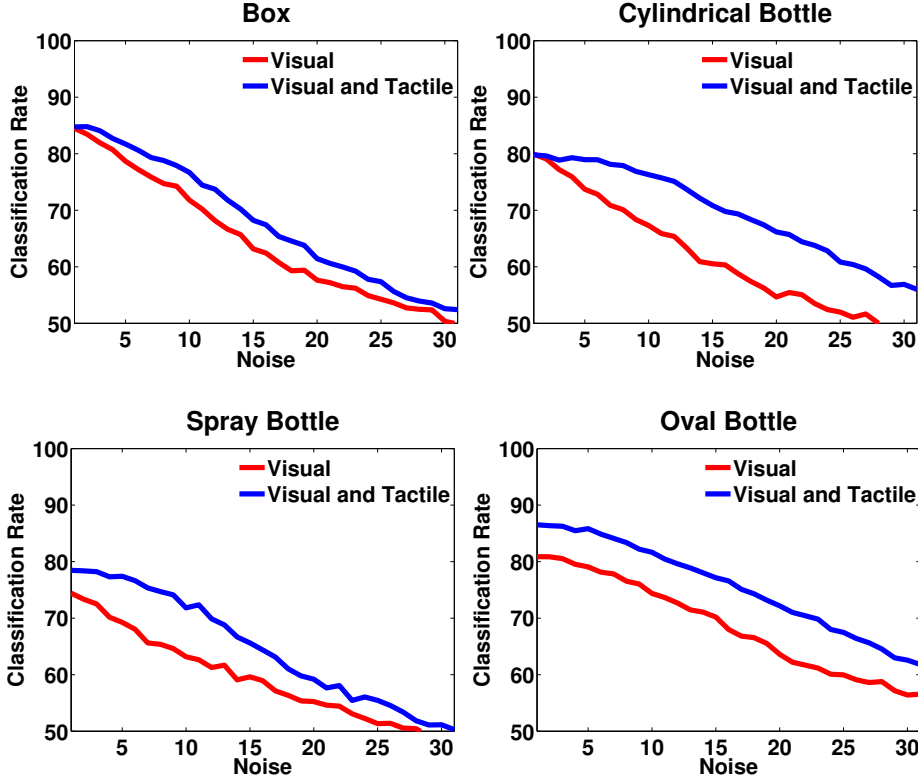


Figure 4.14: Classification rates of the three variants of the model with noisy pose estimates: Noisy pose estimates were simulated by perturbing executed grasps with deviations up to 30 mm and 30°.

each location. Both are time consuming. Instead, we generated noisy hand poses from the previously described object-relative hand poses. The object-relative hand poses were derived from the tracked object pose, therefore the noise in the object poses would be available in the resulting object-relative hand poses. To simulate a noisy pose estimate, we perturbed an executed grasp  $g_e$  that is described by the hand pose in the object frame and generated a grasp  $\hat{g}_n$  from  $P(g_n|g_e) \propto \mathcal{K}_g(g_e, g_n)$  with the bandwidths  $\sigma_p$  and  $\sigma_o$  defined in Eq. (4.6) that allow deviations up to 30 mm and 30°. The resulting perturbed grasp  $\hat{g}_n$  corresponded to the object-relative hand pose derived from the noise pose estimate.

The classification rates with only vision-based classifiers and vision-and-tactile based classifiers against the noise are given in Figure 4.14. As the deviations increase, the classification rates decrease. For large deviations around 30 mm and 30°, visual data become useless as confirmed by the classification rates around 50%. For all objects, adding tactile

data helped to achieve a better discrimination as vision-and-tactile-based classifier yielded better classification rates than the vision-based classifier. For the box, when the tactile data was included, the improvement in the classification rates was smaller than the other objects. As the noise in the visual measurements increases, in general the difference between the classification rates of vision-and-tactile-based classifiers and the classification rates of vision-based classifiers increases as observed for the box, cylindrical bottle and the spray bottle. In summary, when the object pose estimates are not accurate the tactile information can still support the system and provide improved results compared to using only visual feedback.

#### 4.4 Summary and Discussion

Visual information can help to improve predictions for stability, for example when grasps yield similar tactile readings but have different stability outcomes. Hence, in this chapter we also considered visual feedback and studied the viability of concurrent object pose tracking and tactile sensing for assessing grasp stability. We used one-shot measurements obtained from the end of the grasp executions. We presented a KLR model of pose- and touch-conditional grasp success probability. We preferred KLR, since it can directly learn the class probabilities and provide us with confidence of predictions. Based on probabilistic outputs, the robot can have a degree of certainty about its predictions and decide not to lift the object when it is uncertain. Additionally, we could also compare classifiers trained with different modalities in terms of their certainty. The system used object-gripper configurations and acquired tactile imprints of the grasps to extract features for our stability predictor.

The robot was able to explore given objects by grasping and lifting them in order to observe stability outcomes and learn grasp stability classifiers based on the extracted visual and tactile features during grasps. We focused on specific regions on objects to make experiments feasible. The robot executed multiple grasps in a region. These grasps were generated by sampling distributions centered around predefined grasps for each object. We note that as pose parameters cannot be shared across objects, each classifier was specific to one object – a classifier was learned and evaluated with the data collected for a single object. Different models, i.e., vision-based, tactile-based and vision-and-tactile-based, were evaluated to analyze the relevance of tactile and visual data.

Experimental results demonstrated that considering both visual and tactile input for stability assessment is beneficial. While the relevance of tactile and pose data varied between objects, models trained on both pose and tactile parameters performed similarly or better than the models trained exclusively on visual or tactile signals. This result shows that, despite the modeling difficulties associated to an increased perceptual dimensionality, our learning algorithm successfully identifies the discriminative characteristics within the joint visual and tactile channel.

In our setup, pose tracking was stable. However, in general, it is unreasonable to assume that objects will be perfectly tracked during grasps due to reasons such as hand occlusion or large distance between the camera and the object. Therefore in order to eval-

uate our method in the presence of inaccurate pose estimates, we simulated noisy poses and we showed that in such cases, the tactile data can still be complementary and improve prediction results.

The approaches introduced so far did not take task requirements into account. Each task has specific constraints on both the geometry and the robustness of the grasp. Different tasks require different manipulations and therefore different stability levels. In the next chapter, we study how semantic task information can be taken into account while assessing grasp stability.



## Chapter 5

# Task-oriented Grasp Stability Assessment

This chapter proposes to use a generative approach, the Bayesian network (BN) [83], to model the grasp space that is composed of a set of sensory features  $X$  relevant for grasping tasks  $T$ .  $X$  originates from three groups of features,  $\{O, A, H\}$ , where  $O$  denotes an object feature set (from simulated visual sensing),  $A$  denotes an action feature set that represents gripper configurations (from proprioception) and  $H$  denotes a haptic (or tactile) feature set. Detailed feature descriptions can be found in Section 5.2.1.

A Bayesian network is a generative model where not only the class probabilities  $P(T|X)$  can be inferred, but also the class conditional distributions can be predicted  $P(X|T)$ . The former means we can use a BN to predict success of a grasp to achieve a task given observed object and action features by inferring the posterior distribution  $P(T|O, A)$ . The latter means that we can also find, given an assigned task, the distribution of the object  $P(O|T)$  and/or grasp features  $P(A|T, O)$ . This provides the basis for the robot to select objects that afford a given task, e.g., *something to drink from*, and plan an optimal grasp strategy using the object to fulfill the task requirements.

In addition, Bayesian networks allow us to infer the domain knowledge through data. The network structure depicts an influence diagram illustrating the conditional relations between different variables. Also the class conditional on feature variables provides an intuitive evaluation of task and stability-related requirements.

Another strength of the Bayesian network is its ability to infer the grasp success with partial observation. In a task-based grasp adaptation scenario (see Figure 5.9), this is especially important because we can predict the grasp success given observations on only object features and grasp parameters planned in a simulation environment. A grasp replan therefore can be initiated without having to execute an unstable grasp using real robot platforms. Though this can also be done using discriminative approaches, each observation condition requires training of a separate model. The next section provides an overview of the Bayesian networks.

### 5.1 Bayesian Network

A Bayesian network [83] is a probabilistic graphical model that encodes the joint distribution of a set of random variables  $V = \{V_1, V_2, \dots, V_m\}$ . Each node in the network represents one variable, and the directed arcs represent conditional in-dependencies. Given a structure of the network  $\phi$  and a set of local conditional probability distributions (CPDs) of each variable  $V_i$ , the joint distribution of all the variables can be decomposed as

$$P(V; \theta, \phi) = \prod_{i=1}^m P(V_i | \text{pa}_i; \theta_i, \phi), \quad (5.1)$$

where  $\text{pa}_i$  denotes the parents of the node  $V_i$ , and the parameter vector  $\theta = (\theta_1, \dots, \theta_m)$  specifies the CPDs. Learning a BN includes discovering from a dataset: 1) how one variable depends on others ( $\theta$ ), and 2) what the conditional in-dependencies between different variables are ( $\phi$ ). The former is an instance of the parameter learning and the latter of the structure learning. Various algorithms and techniques have been developed to learn a BN in different model and data conditions [47].

In this chapter, we use the Bayesian network to model the joint distribution of a set of task and stability-relevant variables (see Table 5.1), i.e.,  $V = \{T, X\}$  where  $X \subseteq \{O, A, H\}$ . To correctly describe a grasping task, both conceptual high-level information such as object category and continuous low-level sensorimotor variables such as hand pose are required. The variables in this work are therefore both discrete (e.g., *task*, *obcl*), and continuous (most *O*, *A*, *H* features). The continuous features such as hand grasp configuration can be high-dimensional with complex probabilistic distributions.

Learning BN structures from both continuous and discrete data is an open problem, particularly when continuous data is high-dimensional and sampled from complex distributions. Most algorithms for structure learning only work with discrete variables. Therefore, a common approach is to convert the mixed modeling scenario into a completely discrete one by discretizing the continuous variables [43]. In this thesis we use a two-step discretization scheme. For a high-dimensional continuous variable  $X$ , the data in the original observation space is first projected to a low-dimensional space, and then a parametric mixture model (multi-variate Gaussian mixture) is learned to model the data density in this space,

$$P(X = x) \propto \sum_{k=1}^M \lambda_k \mathcal{N}(x; \mu_k, \Sigma_k). \quad (5.2)$$

where  $\mu_k$  and  $\Sigma_k$  are the mean and the covariance of each Gaussian component, and  $\lambda_k$  is the mixing proportion. The parameters of the mixture model are learned using the standard EM approach. The number of the clusters for each variable is found through cross-validation where the task classification performance with the BN is maximized.

We use a greedy search algorithm to find the network structure (the directed acyclic graph, or DAG) in a neighborhood of graphs that maximizes the network score (Bayesian information criterion [99]). The search is local and in the space of DAGs, so the effectiveness of the algorithm relies on the initial DAG. As suggested by Leray and Francois

[71], we use another simpler algorithm, the maximum weight spanning tree [31], to find an oriented tree structure as the initial DAG.

### 5.1.1 Inference in Bayesian Networks

A trained network defines the factorization of the joint distribution of the observations,  $P(V) = P(T, O, A, H)$ , in terms of a graph of conditional dependencies. We can compute the posterior distribution of one or group of variables given the observation of others. A common way for doing this is to apply the junction tree algorithm [53]: an algorithm of local message passing to compute the distribution of the variables of interest. The output of the network is a multinomial distribution over each of the discrete states of the network,

$$P(U_i = u_{ik} | \Psi = \psi). \quad (5.3)$$

stating as “the probability that the discrete variable  $U_i$  that corresponds to variable  $V_i$ , is at state  $u_{ik}$  when other discretized variables  $\Psi$  are observed to be at states  $\psi$ , where  $\Psi = U \setminus U_i$ ”.

## 5.2 Modeling Sensor Data and Data Acquisition

In this section firstly a detailed description of the sensory data representation is presented. Then the data acquisition process which uses both a grasp simulation environment and a real robot platform is described.

### 5.2.1 Feature Description

Table 5.1 lists all the features used for the representation of the sensor data. The object features  $O$  include the object class identity *obcl*, the three dimensional *size*, and the convexity *cvox*. The action features  $A$  describe the hand pose (position and orientation) in the object-centered coordinate system and the final hand configuration *fcon*. We decompose the grasp position into a unit sphere *npos* and the radius *rad* for visualization purpose in the inference results.

In terms of haptic features  $H$ , we calculate a set of tactile features measured by the 6 tactile sensors on the Schunk hand. Figure 5.1 shows illustrations of tactile features. *iG* carries information about the distribution of the pressure in the vertical and the horizontal directions and also the pressure centroids (*iC*) locally for each sensor array. *pG* is the 3D version of *iG* with respect to the wrist frame, it is calculated using *fcon* and it represents the pressure distribution and the pressure centroid (*pC*) considering all the sensors. Another tactile feature is the average normal vector *aNV* that is calculated by  $\sum_{i=1}^{486} \tau_i r_i$  where  $r_i$  is the normal vector of the texel  $i$  and  $\tau_i$  is the normalized tactile reading in the texel  $i$  ( $\sum_{i=1}^{486} \tau_i = 1$ ).

We emphasize that the representation of a grasp may be redundant, e.g., *iG* contains information of e.g., *iC*. Such an over-representation of the feature variables allows us to select the most representative variables and enables an efficient learning and inference. It

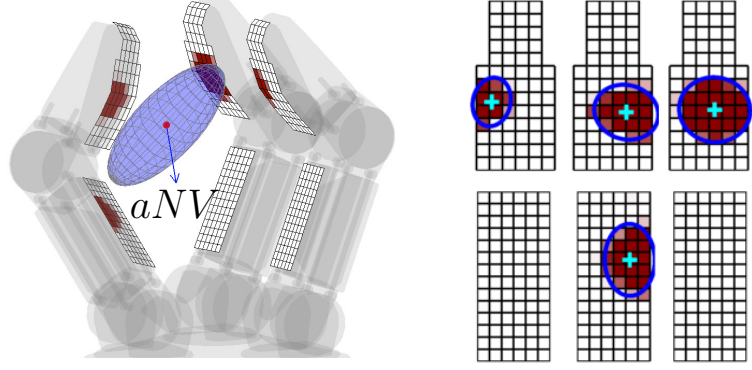


Figure 5.1: Illustration of the tactile features: The left side shows the pressure distribution considering all the sensors  $pG$ , which is represented by the 3D ellipsoid that models the spread of the tactile readings by its size and orientation. The pressure centroid considering all the sensors ( $pC$ ) is marked with a red dot. The blue arrow represents the scaled average normal vector  $aNV$ . The right side shows the dark blue ellipses representing the pressure distributions locally  $iG$  and also each pressure centroid  $iC$  marked with a cross.

also allows us to use Bayesian networks to identify the importance of, and the dependencies between these variables in various scenarios of robot grasping tasks.

### 5.2.2 Data Acquisition

The goal of the data acquisition is to obtain a set of data that instantiate the variables in  $\{O, A, H, T\}$ . Seven home-environment objects including three bottles and four mugs were chosen for the data generation. In GraspIt! [74], a Schunk hand model was used for planning grasps on the corresponding object models and extracting features. The seven object models that capture the similar sizes and shapes of the real objects and the real objects can be seen in Figure 5.2.

All the steps in the data generation process is depicted in Figure 5.3. To extract the features in Table 5.1, we first generated grasp hypotheses using the grasp-planner BADGr [54]. Each grasp hypothesis was first visualized in GraspIt! by a human tutor who associated it with a task label from the simulation ( $T_{sim}$ ). Then the hypothesis that was good for at least one task was used on the robot platform to perform a set of grasps and manipulations on the similar real object during exploration. If a grasp that is considered to be good for a task, e.g., *pouring* (by label  $T_{sim}$ ) results in an unstable  $90^\circ$  rotation (object drops/slips) which is defined to be the required manipulation for *pouring* task, then it will be considered to be bad for *pouring* in the final task label  $T$ .

Three tasks were defined for the experiments: *hand-over*, *pouring*, and *dishwashing*. Each task was associated with certain manipulations and geometric constraints: For *hand-over*, the object should be stably transported (Tr) horizontally and there should be a suffi-

Table 5.1: Feature set with dimensionality  $D$  (low/high) and the number of discrete states  $M$  (optimized for each of the three tasks [hand-over, pouring, dishwashing] and shown for the selected features only).  $T$ ,  $O$ ,  $A_{1,2,3}$  are from the simulation,  $A_4$  and  $H$  are from the real robot.

|       | Name   | $D$  | $M$          | Description                 |
|-------|--------|------|--------------|-----------------------------|
| $T$   | $task$ |      | 2            | Binary task identifier      |
| $O_1$ | $obcl$ |      | 2            | Object class                |
| $O_2$ | $size$ | 3    |              | Object dimensions           |
| $O_3$ | $cvox$ | 1    | [5, 5, -]    | Convexity value [0, 1]      |
| $A_1$ | $dir$  | 4    |              | Quaternion hand orientation |
| $A_2$ | $npos$ | 3    | 16           | Unit grasping position      |
| $A_3$ | $rad$  | 1    | [15, 14, 14] | Radius of $npos$            |
| $A_4$ | $fcon$ | 2/7  | [7, 7, 7]    | Final hand configuration    |
| $H_1$ | $iG$   | 5/30 |              | Local pressure distribution |
| $H_2$ | $iC$   | 3/12 | [11, -, -]   | Local pressure centroid     |
| $H_3$ | $pG$   | 3/9  |              | Pressure distribution       |
| $H_4$ | $pC$   | 3    |              | Pressure centroid           |
| $H_5$ | $aNV$  | 2/3  | [-, 5, 5]    | Average normal vector       |

cient free volume of the object that is not covered by the hand to deliver it. For *pouring* the object should be rotated  $90^\circ$  (R90) and the top of the object must be unblocked. For *dishwashing*, the object should be stably rotated  $180^\circ$  (R180) to place it upside-down and the fingers should not block the top of the object. Figure 5.4 shows example grasps with different stability conditions according to the defined manipulation requirements.

During data generation, our goal is to execute the planned hypotheses around the object. To avoid that some grasps are not reachable, we place the object in a known location in front of the robot, and manually rotate the object along the vertical axis by a  $45^\circ$  increment to place the hypotheses in the robot’s working space.

Because of the uncertainty introduced in both the motor system and the manual placement, the real hand pose will not precisely represent the values generated in simulation. This uncertainty was simulated by adding zero-mean and Gaussian-distributed noise to the data. Figure 5.5 shows example grasps generated on bottles and mugs in both clean (left) and noisy (right) versions. The resulting grasping position had noise with standard deviation about 0.4 to 1.1 (cm) in the three dimensions. For each task a grasp dataset with equal number of positive and negative samples was obtained. The number of positive samples was 1026 for *hand-over*, 1143 for *pouring* and 831 for *dishwashing*. These three datasets were used to train task-specific BNs.

### 5.3 Model Selection

Model selection is a process including three steps: 1) dimension reduction, 2) variable selection, and 3) optimizing data discretization. These three steps were applied on the



Figure 5.2: The seven objects and the similar object models used in the simulation.

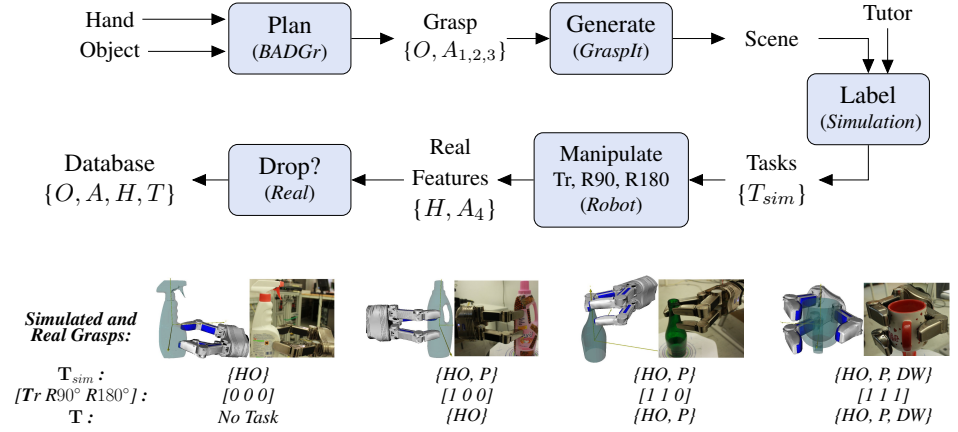


Figure 5.3: The Data Generation Process: The top row is a diagram of the process. The bottom row shows 4 example grasps and how they were labeled with different tasks. The three tasks are hand-over (HO), pouring (P) and dishwashing (DW), each of which has to satisfy one of the stable manipulations: transport (Tr), 90° rotation (R90), and 180° rotation (R180), respectively.

three datasets separately for each task and task-specific BNs with binary task variables were built.

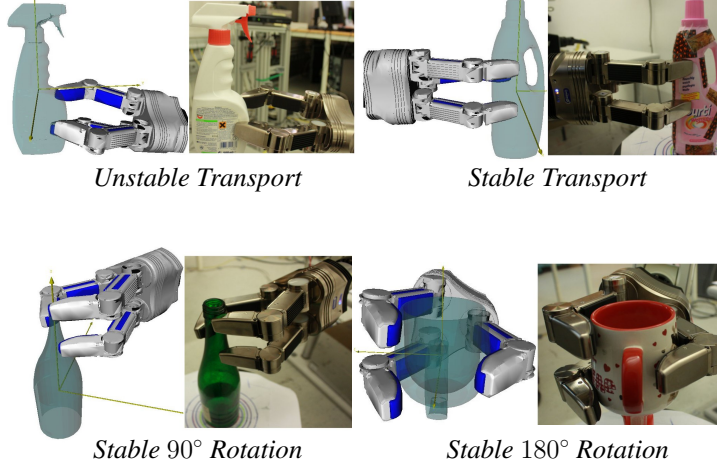


Figure 5.4: Four example grasps with different stability conditions: Simulated grasps were executed on the real platform and for each grasp three manipulations were applied: Horizontal transport, 90° rotation and 180° rotation. Each real grasp was marked with the stability outcome of the manipulation.

### 5.3.1 Dimensionality Reduction

There are many techniques for dimension reduction ([110]). Ideally a cross-validation process should be used to select optimal technique and their parameters. However, we have many steps for model selection, a full-scale model selection will be expensive. Considering the main focus is not to evaluate dimension reduction techniques, we decide to select a single method. Kernel PCA [109], was chosen as the dimensionality reduction technique because of its capability to model non-linear manifolds which is a character of our problem domain. Table 5.1 shows the resulting dimensionality together with the original dimensionality on a set of variables.

### 5.3.2 Variable Selection

We use the HITON algorithm [5] to perform the optimal variable selection for the three tasks. HITON works by first inducing the Markov Blanket of the target variable to be classified. For our problem, the target is the binary task variable  $T$ , and its Markov Blanket is denoted by  $MB(T)$ .  $MB(T)$  is defined as the set of variables that makes  $T$  independent of all other variables. Then support vector machine was used to further remove the unnecessary variables in the  $MB(T)$  in a greedy hill-climbing fashion. The performance metric was the task classification rate. Exhaustive search through all subsets of features returned in  $MB(T)$  is prohibitive, so we adopted a set of heuristics to form a smaller search space: 1) the subset must include *obcl* and *npos* because we are interested in inferring the conditionals involving these variables, 2) there must be at most two features in each of the  $O$ ,

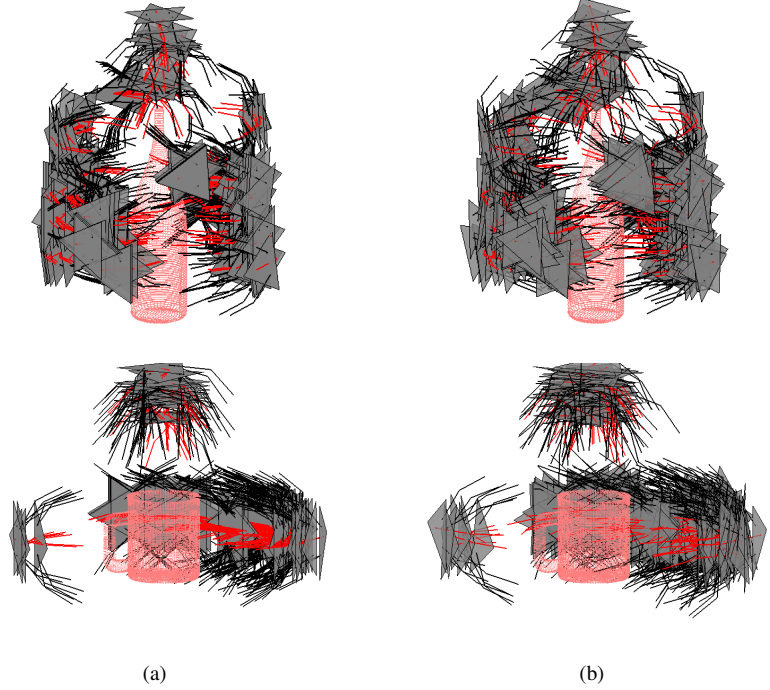


Figure 5.5: Grasp examples from the datasets: The Schunk hand’s position and orientation is represented by the simplified triangular hand model. (a) Example grasp hypotheses planned on the two classes of objects (mugs and bottles) in simulation. The number of grasp hypotheses were increased by changing the hand preshape, the distance between the hand and the object, and also rotating the hand along the approach vector. (b) The grasps whose poses added with noise to simulate real executed grasps.

$A$  and  $H$  feature sets. We adopted a stopping point at a 95% threshold of classification accuracy. The subset of features with the highest score discovered up to this point was selected as the satisfactory set of features. Figure 5.6 shows which variables were selected for each of the three tasks.

### 5.3.3 Optimizing Data Discretization

This is a step for only Bayesian networks. The structure learning requires discrete data. However, the data discretization leads to loss of information. When the resolution is low (i.e., a few discrete states), the variance in the original continuous domain that is discriminative may be smoothed out. On the other hand, for the variables that are not discriminative, a high resolution will jeopardize the classification performance due to the curse of

dimensionality. We therefore want to find an optimal granularity on a small set of variables ( $\{cvelx, rad, fcon, iC, aNV\}$ ). The optimal granularity maximizes the task classification performance with the BNs. Table 5.1 shows the resulting number of discrete states  $M$  for each of the three tasks.

## 5.4 Model Evaluation

In this section, we first show the task classification results. Then the inference results on selected continuous variables are presented. The goal is to illustrate the ability of the proposed model to encode the different aspects of the problem domain and provide insights into the dependencies between the variables. Finally we use a task-oriented, stability-based grasp adaptation scenario to demonstrate how to apply the model.

For classification performance, the generative approach was compared with a discriminative approach, Kernel Logistic Regression (KLR). Given a class variable (the task  $T$ ) and the input feature set ( $X \subseteq \{O, A, H\}$ ) KLR models the probability of the class variable  $P(T|X)$  through the weighted sum of the similarities (kernels  $\mathcal{K}$ ) between a testing point and each training point as described in the previous section. During training to obtain the weight parameters, the bandwidth parameters of Gaussian kernels and the regularization constant chosen by cross-validation were used.

The comparison of the two modelling approaches was done under two observation conditions: the partial observation when only simulated object and action variables ( $O$  and  $A_{1,2,3}$ ) were observed, and the full observation when haptic information was also available after grasp execution in the real environment ( $O, A, H$ ). Under these conditions, 50 trials of cross-validation with 20% hold-out splits were performed. In each trial, for each task three models were trained: 1) KLR with all the selected variables, 2) KLR with only simulated variables, and 3) BN with all the selected variables. We do not need to train BN with only simulated variables because the task probability can be inferred in BNs with partial observations. When training KLR models, the continuous low-dimensional representations were used. And when training BNs, the optimal discrete data was used. In each trial, both structure and parameters of the BNs were learned. Since each trial used different set of training data, the resulting structure could be different.

For each task, the inference results on two variables are shown:  $npos$  and one of the selected H features for the task. We chose  $npos$  because it represents from which direction the hand is placed relative to the object, and it is therefore an intuitive variable to demonstrate task constraints. We chose one tactile-related feature to show that the BN can be used to produce an expectation over sensor data given task constraints. For each variable, a set of points  $\mathbf{x}$  were evenly sampled in the low-dimensional space for easy visualization. For each sampled point, a conditional likelihood was obtained given the three tasks and the object class  $P(\mathbf{x}|task, obcl)$  to generate the *likelihood maps* seen in Figure 5.8.

### 5.4.1 Network Structures

Figure 5.6 shows the Bayesian network structures (DAGs) with the highest task classification performance for the three tasks. The represented nodes in each network are the

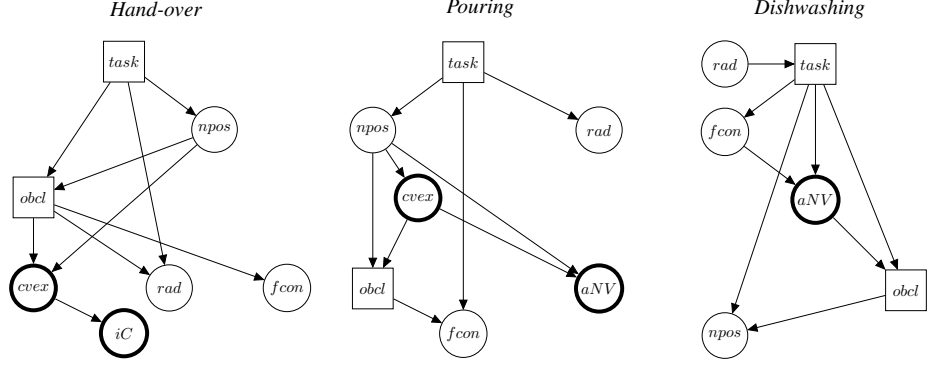


Figure 5.6: The Structure of the BNs with the Highest Classification Performances: DAGs of the three BNs each of which models a binary classification of one task. Square nodes represent discrete variables and circled nodes continuous ones. The differences in variable selection among the three tasks are highlighted by thick border of the nodes.

selected variables. The differences in the selected variables between the different tasks are highlighted by the thick-bordered nodes.

Considering haptic features  $H$ , *hand-over* task selected *iC*, whereas *pouring* and *dishwashing* tasks both selected *aNV*. *iC* is a feature characterizing the local pressure centroid of each tactile sensor pad on the fingers, whereas *aNV* summarizes the whole pressure distribution considering all the sensors and also the finger configurations. In other words, *aNV* encompasses stronger information that may be relevant to stability especially when the task demands stronger grasping such as *pouring* or *dishwashing*.

As to the network structure, all the three tasks had direct conditional relations with *npos* and *rad*. This is natural since the position of the hand relative to the object is an important factor influencing both the affordance of a task (from which direction to approach the object *npos*), and its stability requirements (how far away the hand is from the object center of mass *rad*). For *dishwashing*  $T$  was directly connected to *aNV*, whereas for *pouring*  $T$  influenced *aNV* through *npos*. This may be due to that *dishwashing* requires a manipulation with  $180^\circ$  rotation, which, compared to  $90^\circ$  rotation for *pouring*, is more demanding in terms of grasp stability. So the task success for *dishwashing* depended on *aNV* even if the *npos* was also observed.

## 5.4.2 Classification

The area (AUC) under the ROC curve was used as the performance metric. The ROCs were derived by thresholding the classifier outputs, the probability of task success  $P(T = \text{true}|X)$ . Figure 5.7 shows the ROC curves for task classification results averaged over 50 trials. Table 5.2 shows the mean and the standard deviation of the AUCs.

In general, the BNs with both full and partial observations had good classification performances for all the three tasks. Under full observations, KLR models performed better than BNs. However, we note that when the real sensor data ( $H$  and  $A_4$ ) were not

Table 5.2: The mean and the standard deviation of AUCs for the three tasks.

| Task        | KLR <sub>full</sub> | KLR <sub>partial</sub> | BN <sub>full</sub> | BN <sub>partial</sub> |
|-------------|---------------------|------------------------|--------------------|-----------------------|
| Hand-over   | 0.97 (0.01)         | 0.90 (0.01)            | 0.90 (0.04)        | 0.86 (0.01)           |
|             | $\Delta 0.07 > *$   |                        |                    |                       |
| Pouring     | 0.98(0.01)          | 0.90 (0.01)            | 0.88 (0.02)        | 0.86 (0.02)           |
|             | $\Delta 0.07 > *$   |                        |                    |                       |
| Dishwashing | 0.98 (0.01)         | 0.87(0.02)             | 0.92 (0.01)        | 0.86 (0.02)           |
|             | $\Delta 0.07 > *$   |                        |                    |                       |

observed, KLR models yielded larger performance drops compared to BNs, which was confirmed by a two-sample t-test on the AUC scores over the 50 trials of the experiment. The hypothesis was: “*The classification performance with full observation is 0.07 higher than the performance with partial observation*”, briefed as “full  $\Delta 0.07 >$  partial”. The results showed that at the significance level 0.05, the hypothesis was accepted for the KLR models, but rejected for the BNs.

Another result is that, when real sensory features ( $H$  and  $A_4$ ) were not observed, the performance drop for *dishwashing* task in the BN was higher than for the other two tasks. This is related to the differences in the task requirements of grasp stability which explained the structural differences depicted in Figure 5.6. For example, when  $aNV$  was not observed in *dishwashing*, more useful information was lost than in *pouring*. This is an interesting result, providing an important insight of the effect of different variables/sensory data in grasping tasks.

### 5.4.3 Inference

Figure 5.8 shows the likelihood maps conditioned on the tasks and the object categories. The brighter color indicates higher probability of a successful grasp. On the left side, we can see the results on  $P(np_{os}|task, obcl)$ , where the hand positions in the object frame were projected on the unit sphere. For the *pouring* task, the robot should not grasp the mugs or the bottles from the top, which is reflected by the dark color on the  $np_{os}$  sphere. However, top grasps are allowed for the *hand-over* task. Among the two object classes, only the mugs afford the *dishwashing* task, which is indicated by the fact that the likelihood maps are almost completely black.

On the right side we can see the results of two tactile features projected on the low dimensional space, 3D  $P(iC|task, obcl)$  for the *hand-over* task and 2D  $P(aNV|task, obcl)$  for the other two tasks. We observe clear differences in these “haptic images” both between the two different object classes, and also between the different tasks. This reflects different “haptic expectations” given task and object conditions. For the pouring task, we observe that the mugs have a clear cut between “bad” and “good” regions in the  $aNV$  map, whereas the bottles have more gradual change in the likelihood map. The reason may be that the bottles are much taller than the mugs therefore there are more grasps along the longitudinal direction on the bottles that have gradual changes in grasp quality.

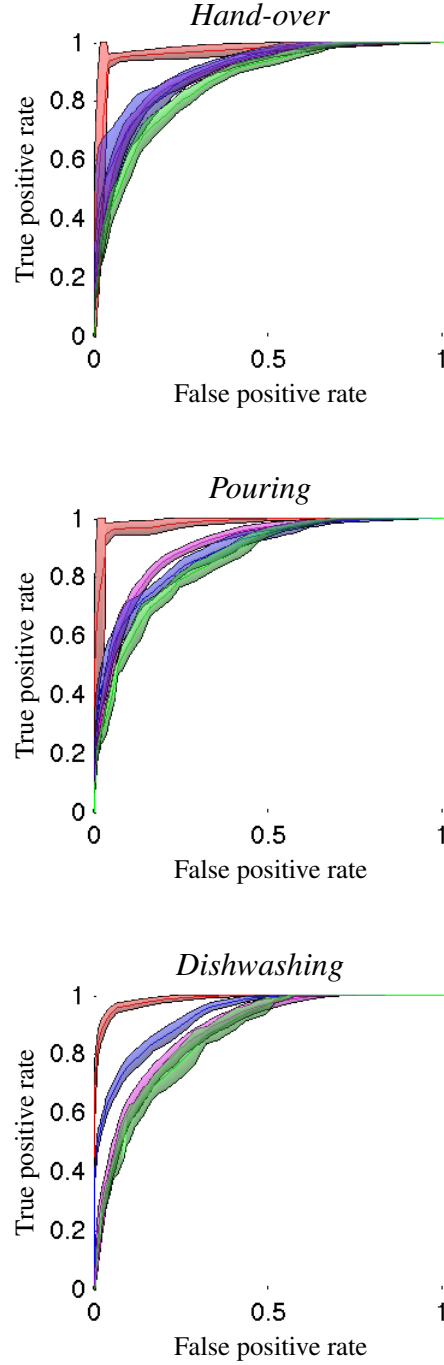


Figure 5.7: Classification: The average ROC curves for the three tasks. Red is KLR with full observation ( $O, A, H$ ). Pink is KLR with partial observation ( $O, A_{1,2,3}$ ). Blue is BN with full observation ( $O, A, H$ ). Green is BN with partial observation ( $O, A_{1,2,3}$ ). The transparent regions represent the one standard deviation of the true positive rate.

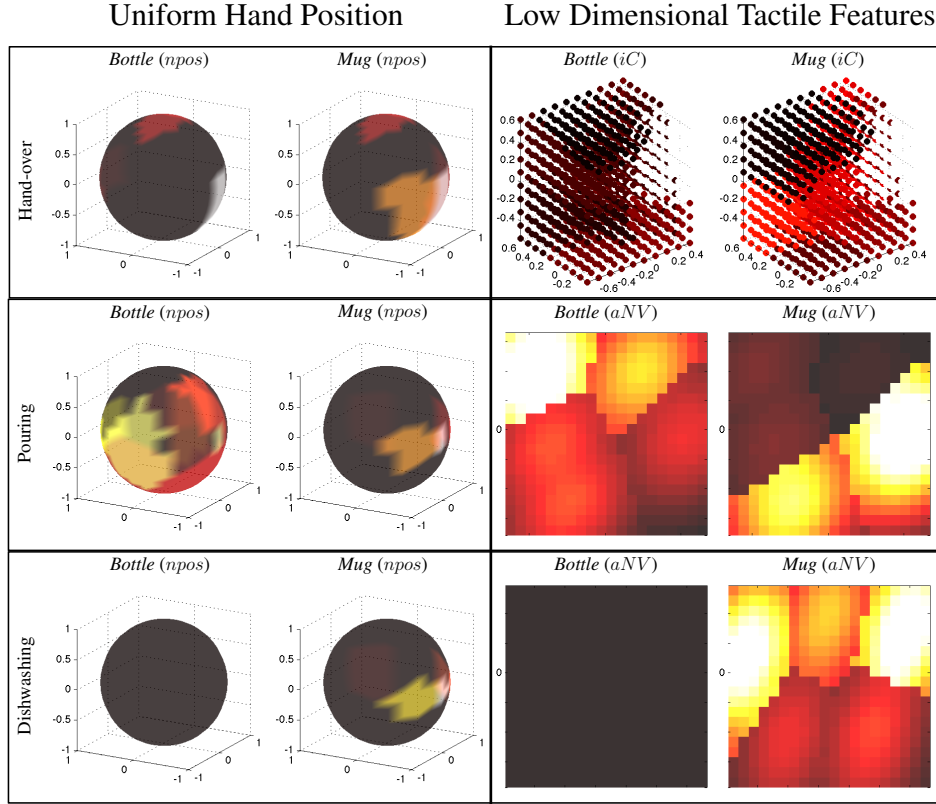


Figure 5.8: The likelihood maps of the continuous variables conditioned on the task and the object class. Left side in inference results shows  $P(npos|task, obcl)$  for all the three tasks. On the right side,  $P(iC|task, obcl)$  is obtained for *hand-over* and  $P(aNV|task, obcl)$  is obtained for the other two tasks.

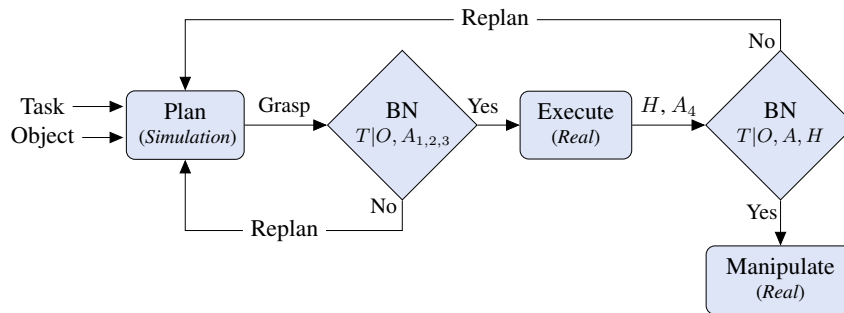
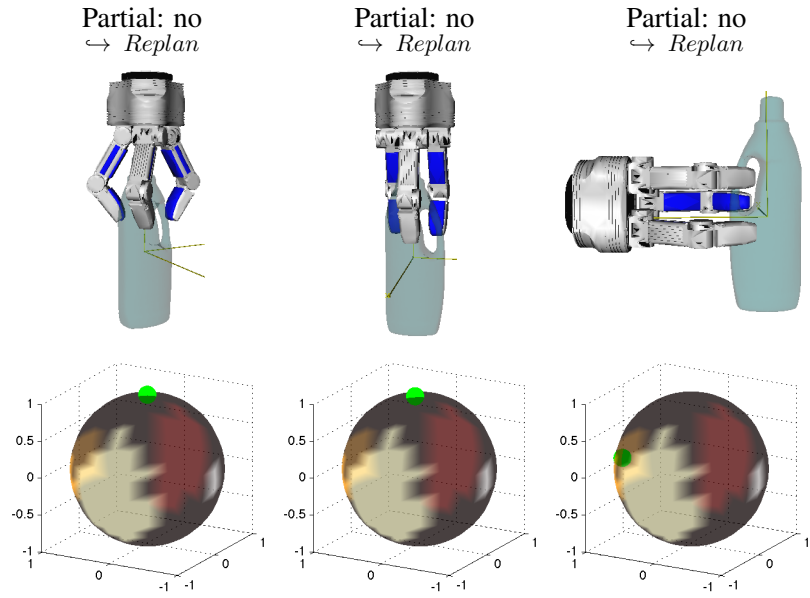
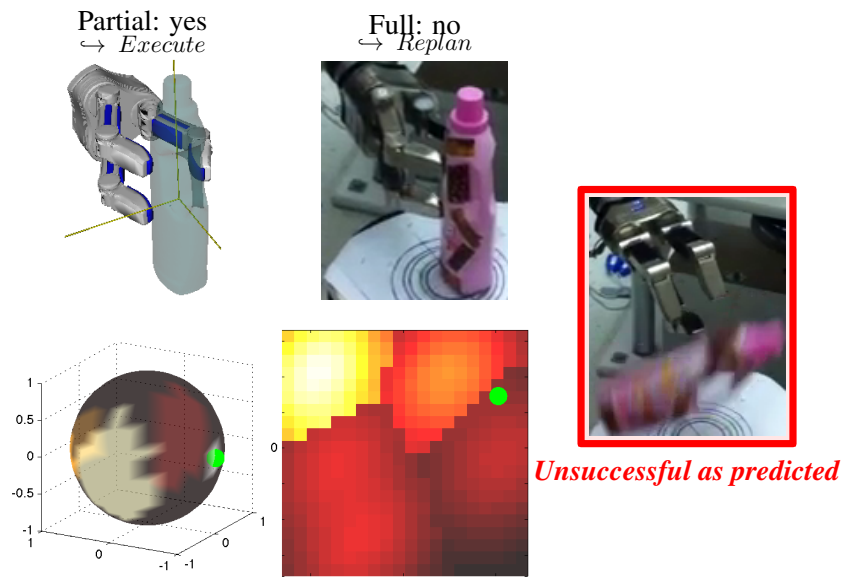


Figure 5.9: Task-based grasp adaptation.



(a)



(b)

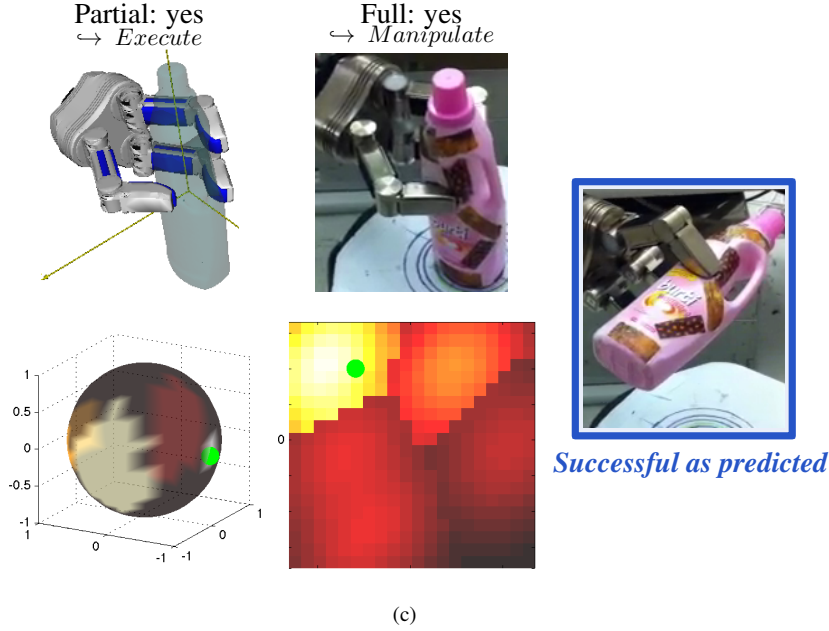


Figure 5.10: Application: The two-loop grasp adaptation when the task is *pouring* with the oval bottle following the flowchart in Figure 5.9. (a) Rejected grasps based on the partial observation. (b) Another rejected grasp based on the full observation. (c) An accepted grasp based on the full observation.

#### 5.4.4 Model Application

We conclude this chapter by a task-oriented, stability-based grasp adaptation scenario. The goal is to demonstrate one way of applying the proposed probabilistic framework. Figure 5.9 depicts a two-step grasp adaptation process, where the first step predicts if a planned grasp hypothesis affords an assigned task (from the simulated  $O$  and  $A_{1,2,3}$  features) before it is executed on real robots, and the second step predicts if the grasp affords manipulation demanded by the task once the grasp has been executed. Here the sensory inputs  $H$  and  $A_4$  are available which allows more accurate prediction with full observation  $P(T|O, A, H)$  before the object is lifted. Such a *double-guarded* system is beneficial to efficiently plan and execute the robot grasping.

The system requirements include: 1) the object is familiar, i.e., it can be registered to a 3D model with a similar size and shape; 2) there is a grasp planner that can, given the familiar model and the robot hand, either pre-plan a set of grasps, or sequentially produce hypotheses; and 3) a feature extraction system exists to calculate  $O$ ,  $A$  and  $H$  feature set that is modeled by the task BN.

Given these conditions, Figure 5.10 demonstrates a grasp adaptation process for the task command *pour with the oval bottle*. The top row in each subfigure shows the grasp hypotheses sequentially produced by a planner. Before the first three hypotheses seen in Figure 5.10a were executed on the real robot platform  $P(T|O, A_{1,2,3})$  rejected them. This is reflected by the location of the data point (green dot) in the dark region of  $n_{pos}$  likelihood maps. The grasp replan was triggered until the fourth hypothesis seen in Figure 5.10b was found to be good for grasp execution. Once the fingers closed around the object, the haptic features were available so the task success could be predicted based on the full observation  $P(T|O, A, H)$ . It was however predicted to fail under the full observation  $P(T|O, A, H)$  ( $a_{NV}$  is in the dark region of the likelihood map). A replan was again triggered until a good grasp was found with the full observation. Figure 5.10c shows the grasp that was predicted to succeed under the full observation.

## 5.5 Summary and Discussion

In this chapter, a unified probabilistic framework using Bayesian networks to model grasps and assess grasp stability in a task-oriented manner was proposed. The framework combined human supervision and exploration during manipulation to encode task dependent stability requirements. The trained network could successfully predict the outcomes of a grasping action both in terms of its geometric requirements and in terms of the stability demands for the subsequent manipulations. A two-loop grasp adaptation was proposed to allow a goal-directed grasp selection in an efficient manner.

Differently from discriminative approaches which model class boundaries, generative models encode all the variations in the data (i.e. through the joint distribution of variables) which is a more demanding process. Hence, for classification discriminative approaches often outperform generative models. To evaluate how well our BN models perform compared to a discriminative approach, classification results of KLR models were also presented.

Bayesian networks yielded good classification performances with both partial observations (when only simulated object and action variables are observed) and full observations (when real features are also available after grasp execution in the real environment). Under full observation KLR models yielded better performance. However under partial observation their performance drop was larger compared to the BNs.

The disadvantages of the BN-based modeling was that it required a large dataset to have good accuracy and needed an efficient discretization step. However, the flexibility of the generative models outweighs the disadvantages. BN models were preferred for the grasp adaptation system, because they allow inference on any variable given full or partial observation of others. Discriminative approaches such as KLR can also be used with partial or full observations, but they require training separate models for different observation conditions.

The BNs can generalize to other similar objects because they incorporate object class as one object representation. They are also extensible to any tasks as long as training data for the new tasks are obtained.

Since the high-level task goals are seamlessly linked to low-level haptic sensory outputs, grasp plan and control become more efficient and goal-oriented. In addition, the generative model allows us not only to predict grasp success and task relevance but also convey domain knowledge. The task conditioned likelihood maps in Figure 5.8, and the learned relationships between variables as shown in network structures in Figure 5.6 demonstrated the use of generative model to interpret the problem domain. We can infer structural dependencies between different variables, and form conditional expectations on sensory features. In other words, we can reason on which sensory features are most relevant for a specific task and robots can perform on-line decision making on what-to-measure, thus optimizing the use of sensory data.



## Chapter 6

# Conclusions

Object grasping and manipulation skills are a necessity for a robot to physically interact with the environment. Most of the today's robot systems, however, demonstrate only limited object grasping and manipulation capabilities. There are main issues which make grasping difficult for robots:

- Unknown information required to plan grasps such as object shape and pose need to be extracted from the environment through sensors. However, sensory measurements are noisy and associated with a degree of uncertainty. Therefore, grasp planning is based on noisy data.
- Even if perfectly accurate information is obtained, planning a suitable grasp is still a challenge. There are a huge number of possibilities to choose from and the parameter space cannot be searched exhaustively. A good planning strategy should take important factors into account such as frictional properties, obstacles or the kinematics of the robot.
- The task that the robot needs to accomplish is another important factor that greatly influences the decision on grasp selection. Each task has its own requirements on the geometry and the robustness of the grasp. Therefore, objects are grasped differently according to the tasks. If a mug is to be placed somewhere else, grasping from the top without applying much force might be suitable. However if the task is to pour water with the mug, then a firm grasp that does not block the top is required. There is thus a need to model both low-level sensory data e.g., haptic feedback and high-level semantic task knowledge e.g., how to place the hand on an object for a given task.
- Object grasping and manipulation in real-world environments are, from a robotics viewpoint, uncertain processes. Despite efforts in improving autonomous grasp planners, either by learning or by building into agents sophisticated visuomotor programs, one cannot assume that a grasp will work exactly as planned. One obvious reason for this, amongst many other, is that the perceptual observations on which the

planner bases its reasoning are always noisy. It is thus unlikely that the robot's fingers will come in contact with the object at the exact intended points. The object will generally move while fingers are being closed, and the final object-gripper configuration, even if geometrically similar to the intended one, may present a prohibitively different force configuration. For this reason, executing grasping actions in an open-loop system is unlikely to prove viable in real-world environments. Real-world environments will often require a closed-loop system in which perceptual feedback is constantly monitored and triggers plan corrections.

Various solutions to these problems have been proposed, e.g., for grasping from noisy vision [21, 86, 94], planning grasps analytically [19, 100] or by learning from experience [77, 96], taking task constraints into account [29, 103], controlling finger forces during grasp execution to stably lift objects [92].

## 6.1 Learning to Assess Grasp Stability

Differently from those proposed approaches, this thesis firstly studied how perceptual feedback (e.g., visual, tactile and proprioceptive) available to a robot, before attempting to manipulate an object, can be utilized to predict grasp stability during grasp execution. This enables a robot to be aware of the outcome of its grasping action and allows it to trigger plan corrections. This thesis also proposed a method that incorporates visual information, task constraints, touch-based stability and grasp parameters together. The aim is to develop an embodied-cognitive system that can learn its low level sensorimotor ability for grasping by exploration and select, grasp and manipulate objects for an assigned task.

### Learning Grasp Stability through Tactile Sensing

This thesis presented how grasp stability can be assessed based on tactile sensing and machine learning methods, including AdaBoost, Support Vector Machines and Hidden Markov Models. In particular, the effect of different sensory streams to grasp stability was studied. This included object information such as shape, grasp information such as approach vector, tactile measurements from fingertips and joint configuration of the hand. In general, increasing knowledge improved classification performances.

Experiments on both simulated and real data were shown. The results indicated that the idea of exploiting the learning approach is applicable in realistic scenarios and knowledge about grasp stability can be inferred using information from tactile sensors while grasping an object before the object is further manipulated.

The results also demonstrated that the stability estimation generalizes well to new objects even with a moderate number of objects used in training. Another result was that the variety of objects used in the training was important to achieve good classification performances.

Training classifiers with the simulated data and using those classifiers for predictions on the real data would provide an efficient way to build stability classifiers, since large datasets can be generated in simulation faster than on the real platforms. Some preliminary

experiments were performed for that purpose, but the real data could not be classified with a good accuracy. A possible reason why simulated data could not be useful for predictions on the real data is that the simulated tactile sensor outputs did not match the real sensor outputs to a sufficient extent. Due to this poor performance of the classifiers trained with the simulated data, we focused on experiments on the real robot.

An integration of grasp planner with online stability assessment based on tactile sensing was also presented in order to improve the hypotheses about suitable grasps on different types of objects. We showed that real executions of stable grasp hypotheses produced by the planner may sometimes result in failures, which supports the need for online stability evaluation during grasp executions.

### **Learning Grasp Stability through Visual and Tactile Sensing**

We also studied classifiers based on both visual and tactile data to further discriminate between stable and unstable grasps. The training data was collected by exploring objects around grasps suggested by a human. The robot executed multiple grasps in a region of objects. The aim was to make the experiments feasible and also let the robot learn the relations between perceptions and the stability outcomes in a region of an object. This is important because on real platforms it cannot be guaranteed that the robot will always be able to grasp an object exactly at the same place.

Kernel logistic regression models were built based on the perceptual data which consist of the object-gripper configuration and a tactile description of the contacts between the object and the fully-closed gripper. The experimental evaluation demonstrated that joint tactile and visual perceptions carry valuable grasp-related information and reliable predictions are formed based on the joint perceptions as models trained on both visual and tactile parameters performed at least as well as the models trained exclusively on one perceptual input on all the objects used in the experiments. This result motivates the use of the two modalities together, since tactile and/or visual data can yield better results for different objects and which modality would be more useful for a given object is not known in advance.

### **Goal-directed Grasp Stability Assessment**

This thesis finally addressed the problem of encoding task relevant grasping based on multisensory data, i.e., visual (simulated), proprioceptive and tactile. To enable goal-directed grasp planning, grounding symbolic representation of goal state (e.g. to pour) to continuous representation of low-level sensory feedback (e.g. grasping pose) is the main challenge. We resolved this by adopting a probabilistic graphical model, a Bayesian network, which encodes relations between variables using conditional probabilistic distributions. Such distributions do not require the variables to comply to the same underlying representations. Therefore we used Bayesian Networks to integrate the discrete semantic information with the continuous representation of sensory data. Our robot learned goal-directed grasps based on a combination of human-supervision and exploration by physical interaction with the environment.

Exploration enables the robot to learn about its own sensorimotor ability (how to grasp an object to stably lift and manipulate it), while human tutoring helps the robot to associate its sensorimotor ability to high-level goals. During exploration, the robot collected visual, proprioceptive (joint sensor) and haptic (tactile sensing array) data by executing a set of grasps on a set of objects. During human supervision, each grasp was labeled with tasks it afforded by a human tutor. A trained BN using this data allows inference of conditional probability of task success given a full or partial observation of all sensory data. The BN also allows inference of the class conditional distributions which is the basis for goal-directed object selection and grasp planning. The results showed that the proposed model accurately estimates grasp success both at the stage of planning (before execution in real environments) and during grasp execution in a goal-oriented way.

## 6.2 Future Work

The methods introduced in this thesis can be extended in different directions. For example, one limitation with the Kernel logistic regression models is that each model that the agent learns is only usable with that particular object, since the models rely on the pose of an object. It is not realistic to imagine that an agent would learn a different model of every object that exists. To overcome this limitation, we plan to study learning models that characterize only a *part* of an object, and which would thus be applicable to novel objects that share the same part.

Another research direction is to include force-torque sensors and track grasp stability during manipulation based on visual, tactile and force-torque measurements. This would also involve examining different approaches for sensor fusion.

Based on the dataset that includes visual and tactile measurements together with the corresponding stability labels, it would also be interesting to explore approaches that can suggest a better hand pose that is likely to result in a stable grasp, when an unstable grasp has been identified.

The Bayesian network models relied on a single time instance for decision making and this is at the point of grasp completion. Although this may be sufficient for many tasks, temporal data is more informative for cases where on-line grasp adaptation and control is needed, e.g., if the shape of the object is complex or if mass distribution is changing rapidly. We can explore dynamic models such as Dynamic Bayesian Networks for this purpose.

Grasping is not a stationary process and it may need further on-line adaptation. Task requirements may vary given different contexts or environments. In addition, sensory measurements may also change over time, requiring model update. One of the areas of future research is development of learning algorithms that allow incremental data discretization and structure update of Bayesian networks.

Finally, the current system does not consider dynamic control problems and the reaching motion. Following the idea of [108], the current likelihood map of the grasping position that encodes task affordances can be combined with the reachability map in [44] to compose a task-oriented path plan, trajectory optimization, and grasp control system.

# Bibliography

- [1] OpenGRASP – A Simulation Toolkit for Grasping and Dexterous Manipulation. URL <http://opengrasp.sourceforge.net>.
- [2] The Schunk Dextrous Hand. URL <http://www.schunk.com>.
- [3] Weiss Robotics Tactile Sensor. URL <http://www.weiss-robotics.de/en.html>.
- [4] J. Aleotti and S. Caselli. 2010. Interactive teaching of task-oriented robot grasps. *Robotics and Autonomous Systems*, 58(5):539–550.
- [5] C. F. Aliferis, I. Tsamardinos, and A. Statnikov. 2003. HITON: a novel markov blanket algorithm for optimal variable selection. *AMIA Annu Symp Proc.*, pages 21–25.
- [6] P. K. Allen. 1987. *Robotic object recognition using vision and touch*. Kluwer Academic Publishers, Boston.
- [7] P. K. Allen, A. T. Miller, P. Y. Oh, and B. S. Leibowitz. 1999. Integration of vision, force and tactile sensing for grasping. *International Journal of Intelligent Machines*, 4:129–149.
- [8] Y. Bekiroglu, R. Detry, and D. Kragic. 2011. Joint observation of object pose and tactile imprints for online grasp stability assessment. In *IEEE ICRA 2011 workshop: Manipulation Under Uncertainty*.
- [9] Y. Bekiroglu, R. Detry, and D. Kragic. 2011. Learning tactile characterizations of object- and pose-specific grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [10] Y. Bekiroglu, R. Detry, and D. Kragic. 2012. Grasp stability from vision and touch. In *IEEE IROS 2012 Workshop: Advances in Tactile Sensing and Touch-based Human-Robot Interaction*.
- [11] Y. Bekiroglu, K. Huebner, and D. Kragic. 2011. Integrating grasp planning with online stability assessment using tactile sensing. In *IEEE International Conference on Robotics and Automation*.

- [12] Y. Bekiroglu, V. Kyrki, and D. Kragic. 2010. Learning grasp stability with tactile data and hmms. In *IEEE International Symposium on Robot and Human Interactive Communication*.
- [13] Y. Bekiroglu, J. Laaksonen, J. Jørgensen, V. Kyrki, and D. Kragic. 2011. Assessing grasp stability based on learning and haptic data. *IEEE Transactions on Robotics*, 27(3):616–629.
- [14] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki, and D. Kragic. 2011. Learning grasp stability based on haptic data. In *RSS 2010 workshop: Representations for object grasping and manipulation in single and dual arm tasks*.
- [15] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic. 2012. Goal-directed grasp stability assessment based on learning. In *IEEE IROS 2012 Workshop: Beyond Robot Grasping - Modern Approaches for Dynamic Manipulation*.
- [16] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic. 2013. A probabilistic framework for task-oriented grasp stability assessment. In *(review for) IEEE International Conference on Robotics and Automation*.
- [17] Niklas Bergström, Jeannette Bohg, and Danica Kragic. 2009. Integration of visual cues for robotic grasping. In *Computer Vision Systems*, volume 5815 of *Lecture Notes in Computer Science*, pages 245–254. Springer Berlin / Heidelberg.
- [18] A. Bicchi and V. Kumar. 2000. Robotic grasping and contact: a review. In *IEEE Int. Conf. on Robotics and Automation*.
- [19] A. Bicchi and V. Kumar. 2000. Robotic grasping and contact: A review. In *IEEE Int. Conf. on Robotics and Automation*, pages 348–353.
- [20] A. Bicchi, J.K. Salisbury, and P. Dario. 1989. Augmentation of grasp robustness using intrinsic tactile sensing. In *IEEE International Conference on Robotics and Automation*.
- [21] J. Bohg, C. B. Holst, K. Huebner, M. Ralph, B. Rasolzadeh, D. Song, and D. Kragic. 2009. Towards grasp-oriented visual perception for humanoid robots. *Int. Journal of Humanoid Robotics*, 6(3):387–434.
- [22] J. Bohg and D. Kragic. 2009. Grasping familiar objects using shape context. In *14th International Conference on Advanced Robotics*, Munich, Germany, June 2009.
- [23] J. Bohg and D. Kragic. 2010. Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362–377.
- [24] G. M. Bone, A. Lambert, and M. Edwards. 2008. Automated Modelling and Robotic Grasping of Unknown Three-Dimensional Objects. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 292–298, Pasadena, CA, USA, May 2008.

- [25] C. Borst, M. Fischer, and G. Hirzinger. 2003. Grasping the Dice by Dicing the Grasp. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3692–3697.
- [26] M. Boshra and H. Zhang. 1994. Use of Visual and Tactile Data for Generation of 3-D Pose Hypotheses. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 73–80.
- [27] M. Boshra and H. Zhang. 1995. A constraint-satisfaction approach for 3d vision/touch-based object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 368–373.
- [28] C. C. Chang and C. J. Lin. 2001. *LIBSVM: a Library for Support Vector Machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [29] L.Y. Chang, S.S. Srinivasa, and N.S. Pollard. 2010. Planning pre-grasp manipulation for transport tasks. In *IEEE Int. Conf. on Robotics and Automation*.
- [30] S. Chitta, M. Piccoli, and J. Sturm. 2010. Tactile object class and internal state recognition for mobile manipulation. In *IEEE International Conference on Robotics and Automation*, pages 2342–2348, Anchorage, AK, USA, May 2010.
- [31] C. Chow and C. Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467.
- [32] M. Ciocarlie, C. Goldfeder, and P. Allen. 2007. Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem. In *In Robotics: Science and Systems Manipulation Workshop*.
- [33] C. Cortes and V. Vapnik. 1995. Support vector networks. *Machine Learning*, 20: 273–297.
- [34] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Krueger, O. Kroemer, J. Peters, and J. Piater. 2010. Learning continuous grasp affordances by sensorimotor exploration. In O. Sigaud and J. Peters, editors, *From Motor Learning To Interaction Learning in Robots*, first edition, pages 451–465. Springer-Verlag, Berlin, Germany.
- [35] R. Detry, D. Kraft, A. G. Buch, N. Krüger, and J. Piater. 2010. Refining grasp affordance models by experience. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2287–2293.
- [36] B. Douillard, D. Fox, F. Ramos, and H. D. Whyte. 2010. Classification and semantic mapping of urban environments. *The Int. Journal of Robotics Research*, 30(1):5–32.
- [37] C. Dunes, E. Marchand, C. Collwet, and C. Leroux. 2008. Active Rough Shape Estimation of Unknown Objects. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3622–3627.

- [38] C. Ferrari and J. Canny. 1992. Planning optimal grasps. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2290–2295.
- [39] C. Ferrari and J. Canny. 1992. Planning optimal grasps. In *IEEE Int. Conf. on Robotics and Automation*, pages 2290–2295.
- [40] P. Filzmoser, B. Liebmann, and K. Varmuza. 2009. Repeated double cross validation. *Journal of Chemometrics*, 23(4):160–171.
- [41] R. A. Fisher. 1953. Dispersion on a sphere. In *Proc. Roy. Soc. London Ser. A*.
- [42] Y. Freund and R. E. Shapire. 1996. Experiments with a New Boosting Algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann.
- [43] L. D. Fu and I. Tsamardinos. 2005. A comparison of bayesian network learning algorithms from continuous data. In *AMIA*.
- [44] M. Gienger, M. Toussaint, and C. Goerick. 2008. Task maps in humanoid robot manipulation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2758–2764.
- [45] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. 2007. Grasp Planning Via Decomposition Trees. In *IEEE International Conference on Robotics and Automation*, pages 4679–4684.
- [46] N. Gorges, S. E. Navarro, D. Göger, and H. Wörn. 2010. Haptic object recognition using passive joints and haptic key features. In *IEEE International Conference on Robotics and Automation*, pages 2349–2355, Anchorage, AK, USA, May 2010.
- [47] D. Heckerman. 1996. A tutorial on learning with bayesian networks. Technical report, Microsoft Research.
- [48] C. Hesse and V.H. Franz. 2010. Grasping remembered objects: Exponential decay of the visual memory. *Vision Research*.
- [49] T.M. Hospedales and S. Vijayakumar. 2008. Structure inference for bayesian multisensory scene understanding. *IEEE transactions on pattern analysis and machine intelligence*, 30(12):2140–2157.
- [50] R.D. Howe. 1993. Tactile sensing and control of robotic manipulation. *Advanced Robotics*, 8(3):245–261.
- [51] R.D. Howe, N. Popp, P. Akella, I. Kao, and M.R. Cutkosky. 1990. Grasping, manipulation, and control with tactile sensing. In *IEEE International Conference on Robotics and Automation*.
- [52] K. Hsiao, L. Kaelbling, and T. Lozano-Perez. 2010. Task-driven tactile exploration. In *Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

- [53] C. Huang and A. Darwiche. 1994. Inference in belief networks: A procedural guide. *Int. Journal of Approximate Reasoning*, 15:225–263.
- [54] K. Huebner. 2012. BADGr - A toolbox for Box-based Approximation, Decomposition and GRasping. *Robotics and Autonomous Systems*, 60(3):367–376.
- [55] K. Huebner, S. Ruthotto, and D. Kragic. 2008. Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping. In *IEEE International Conference on Robotics and Automation*, pages 1628–1633.
- [56] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann. 2009. Grasping known objects with humanoid robots: A box-based approach. In *International Conference on Advanced Robotics*, Munich, Germany.
- [57] A. J. Ijspeert, J. Nakanishi, and S. Schaal. 2002. Movement imitation with non-linear dynamical systems in humanoid robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 1398–1403.
- [58] P. Jenmalm, S. Dahlstedt, and R. S. Johansson. 2000. Visual and tactile information about object-curvature control fingertip forces and grasp kinematics in human dexterous manipulation. *Journal of Neurophysiology*, 84(6):2984.
- [59] A.R. Jiménez, A.S. Soembagijob, D. Reynaerts, H. Van Brussel, R. Ceresa, and J.L. Ponsa. 1997. Featureless classification of tactile contacts in a gripper using neural networks. *Sensors and Actuators A: Physical*, 62(1–3):488–491.
- [60] R. Johansson. 1996. Sensory control of dexterous manipulation in humans. *Hand and brain: The neurophysiology and psychology of hand movements*, pages 381–414.
- [61] R. Johansson. 1998. Sensory input and control of grip. In *Novartis Foundation Symposium*, pages 45–59.
- [62] R. S. Johansson and G. Westling. 1984. Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Experimental Brain Research*, 56(3):550–564.
- [63] J. A. Jørgensen, L. P. Ellekilde, and H. G. Petersen. 2010. RobWorkSim - an open simulator for sensor based grasping. In *41st International Symposium on Robotics (ISR 2010) and ROBOTIK 2010*, Munich, Germany, June 2010.
- [64] Jimmy Alison Jørgensen and Henrik Gordon Petersen. 2008. Usage of simulations to plan stable grasping of unknown objects with a 3-fingered schunk hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, September 2008. IEEE. URL <http://www.robot.uji.es/research/events/iros08/contributions/petersen.pdf>.

- [65] D. Kraft, N. Pugeault, E. Baseski, M. Popovic, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krueger. 2009. Birth of the object: Detection of objectness and extraction of object shape through object action complexes. *Int. Jour. of Humanoid Robotics*, pages 247–265.
- [66] D. Kragic, S. Crinier, D. Brunn, and H. I. Christensen. 2003. Vision and tactile sensing for real world tasks. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 1545 – 1550.
- [67] D. Kragic, Andrew T. Miller, and Peter K. Allen. 2001. Real-time tracking meets online grasp planning. In *IEEE International Conference on Robotics and Automation*, pages 2460–2465.
- [68] A. Kritikos and C. Brasch. 2008. Visual and tactile integration in action comprehension and execution. *Brain Research*, 1242:73–86.
- [69] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, A. Agostini, and R. Dillmann. 2011. Object-Action complexes: Grounded abstractions of sensory-motor processes. *Robotics and Autonomous Systems*, 59(10):740–757.
- [70] J. Laaksonen, V. Kyrki, and D. Kragic. 2010. Evaluation of feature representation and machine learning methods in grasp stability learning. In *10th IEEE-RAS International Conference on Humanoid Robots*, pages 112–117.
- [71] P. Leray and O. Francois. 2006. Bnt structure learning package: Documentation and experiments. Technical report, Université de Rouen.
- [72] Z. Li and S. Sastry. 2003. Task oriented optimal grasping by multifingered robot hands. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 389–394.
- [73] A. N. Meltzoff. 2002. *Elements of a Developmental Theory of Imitation*, pages 19–41. Cambridge University Press, Cambridge, MA, USA.
- [74] A. T. Miller and P. K. Allen. 2004. Graspit! A Versatile Simulator for Robotic Grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122.
- [75] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. 2003. Automatic Grasp Planning Using Shape Primitives. In *IEEE International Conference on Robotics and Automation*, pages 1824–1829.
- [76] A.D. Milner and M.A. Goodale. 2006. *The visual brain in action*. Oxford University Press, USA.
- [77] L. Montesano and M. Lopes. 2009. Learning grasping affordances from local visual descriptors. In *IEEE Int. Conf. on Development and Learning*.

- [78] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. 2008. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26.
- [79] A. Morales, M. Prats, P.J. Sanz, and A. P. Pobil. 2007. An Experiment in the Use of Manipulation Primitives and Tactile Perception for Reactive Grasping. In *Robotics: Science and Systems, Workshop on Robot Manipulation: Sensing and Adapting to the Real World*.
- [80] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze. 2010. BLORT—the blocks world robotic vision toolbox. *Best Practice in 3D Perception and Modeling for Mobile Manipulation (Workshop at ICRA 2010)*.
- [81] Kevin Murphy. 2005. *Hidden Markov Model (HMM) Toolbox for Matlab*. Available at <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>.
- [82] E. Oztop, D. Wolpert, and M. Kawato. 2005. Mental state inference using visual control parameters. *Cognitive Brain Research*, 22(2):129–151.
- [83] J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- [84] A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng. 2006. Bayesian Estimation for Autonomous Object Manipulation based on Tactile Sensors. In *IEEE Int. Conf. on Robotics and Automation*, pages 707–714.
- [85] J. C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.
- [86] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger. 2010. A strategy for grasping unknown objects based on co-planarity and colour information. *Robotics and Autonomous Systems*.
- [87] M. Popovic, D. Kraft, L. Bodenhagen, E. Baseski, N. Pugeault, D. Kragic, T. Asfour, and N. Kruger. 2010. A strategy for grasping unknown objects based on co-planarity and colour information. *Robotics and Autonomous Systems*, 58(5):551–565.
- [88] M. Prats, P.J. Sanz, and A. P del Pobil. 2009. Vision-tactile-force integration and robot physical interaction. In *IEEE International Conference on Robotics and Automation*, pages 3975–3980, Kobe, Japan.
- [89] L. R. Rabiner. 1990. *Readings in Speech Recognition*, chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Inc., San Francisco.
- [90] R. Rao, A. Shon, and A. Meltzoff. 2004. A bayesian model of imitation in infants and robots. In *Imitation and Social Learning in Robots, Humans, and Animals*, pages 217–247.

- [91] B. Rasolzadeh, M. Bjorkman, K. Huebner, and D. Kragic. 2010. An active vision system for detecting, fixating and manipulating objects in real world. *International Journal of Robotics Research*, 29(2-3):133–154.
- [92] J.M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K.J. Kuchenbecker. 2011. Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics*, 27(99):1–13.
- [93] J. Romero, H. Kjellström, and D. Kragic. 2009. Modeling and evaluation of human-to-robot mapping of grasps. In *Int. Conf. on Advanced Robotics (ICAR)*, pages 228–233. IEEE.
- [94] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. 2010. Fast 3d recognition and pose using the viewpoint feature histogram. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.
- [95] A. Sahbani, S. El-Khoury, and P. Bidaud. 2012. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326 – 336.
- [96] A. Saxena, J. Driemeyer, and A. Y. Ng. 2008. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173.
- [97] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard. 2009. Object Identification with Tactile Sensors using Bag-of-Features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 243–248.
- [98] M. Schöpfer, M. Pardowitz, and H. J. Ritter. 2009. Using entropy for dimension reduction of tactile data. In *14th International Conference on Advanced Robotics*, Munich, Germany, June 2009.
- [99] G. Schwarz. 1978. Estimating the dimension of a model. *Annals of Statistics*, 6(2): 461–464.
- [100] K. B. Shimoga. 1996. Robot grasp synthesis algorithms: A survey. *The Int. Journal of Robotics Research*, 15(3):230.
- [101] B. Siciliano and O. Khatib, editors. 2008. *Springer Handbook of Robotics*. Springer. ISBN 978-3-540-23957-4.
- [102] J. S. Son, R. D. Howe, J. Wang, and G. D. Hager. 1996. Preliminary results on grasping with vision and touch. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1068–1075.
- [103] D. Song, K. Huebner, V. Kyrki, and D. Kragic. 2010. Learning task constraints for robot grasping using graphical models. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.

- [104] S. A. Stansfield. 1988. A robotic perceptual system utilizing passive vision and active touch. *International Journal of Robotics Research*, 7:138–161.
- [105] R. Suárez, M. Roa, and J. Cornella. 2006. Grasp quality measures. Technical report, Technical University of Catalonia.
- [106] Erik B. Sudderth. 2006. *Graphical models for visual object recognition and tracking*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.
- [107] M. Toussaint, L. Charlin, and P. Poupart. 2008. Hierarchical POMDP controller optimization by likelihood maximization. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*.
- [108] M. Toussaint, N. Plath, T. Lang, and N. Jetchev. 2010. Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference. In *IEEE Int. Conf. on Robotics and Automation*.
- [109] L. J. P. van der Maaten. 2010. Matlab toolbox for dimensionality reduction (v0.7.1b).
- [110] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. 2008. Dimensionality reduction: A comparative review.
- [111] V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
- [112] A. Vezhnevets. 2006. *GML AdaBoost Matlab Toolbox*. Available at <http://graphics.cs.msu.ru/ru/science/research/machinelearning/adaboosttoolbox>.
- [113] H. Wakamatsu, S. Hirai, and K. Iwata. 1996. Static analysis of deformable object grasping based on bounded force closure. In *International Conference on Robotics and Automation*, pages 3324–3329, Minneapolis, USA, April 1996.
- [114] R. S. Woodworth. 1899. The accuracy of voluntary movement. *The Journal of Nervous and Mental Disease*, 26(12):743.
- [115] Z. Xue, J. Zoellner, and R. Dillmann. 2008. Automatic optimal grasp planning based on found contact points. In *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pages 1053–1058.
- [116] M. Yamada, M. Sugiyama, and T. Matsui. 2010. Semi-supervised speaker identification under covariate shift. *Signal Processing*, 90(8):2353–2361.
- [117] B.H. Yoshimi and P. Allen. 1996. Closed-loop visual grasping and manipulation. In *IEEE International Conference on Robotics and Automation*.