DISSERTATION

Enabling Autonomous Robotic Grasping based on Topographic Features

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften unter der Leitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Markus Vincze
E376
Institut für Automatisierungs- und Regelungstechnik

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Dipl.-Ing. Dipl.-Ing. Mag. David Fischinger
Matr. Nr.: 0026796

Wien, im Juli 2014

David Fischinger

Abstract

Used in industry in manufacturing chains for decades, robots are nowadays entering their way into private households. Their functionality is still limited to vacuum cleaning or mowing the lawn. One of the reason why no universally usable robot-butler has reached marketability, is the limited capability of robot-interaction with its environment, in specific object manipulation and grasping. This thesis presents a novel approach to tackle the open grasping problem by learning suitable grasps from *Topographic Features*. Factors increasing grasp complexity such as unknown objects, incomplete object surface data and visually not segmentable object piles are thereby taken into account.

An integrated system for grasping is presented, capable for grasping known and unknown single objects, as well as objects from piles or in cluttered scenes, given a point cloud. The method is based on the topography of a given scene and abstracts grasp-relevant structures to enable machine learning techniques for grasping tasks. A description of the *Topographic Features*, "Height Accumulated Features" (HAF) and their extension, "Symmetry Height Accumulated Features" (SHAF) is given, and the approach is motivated. The grasp quality is investigated using an F-score metric. The gain and the expressive power of HAF is demonstrated by comparing its trained classifier to one that resulted from training on simple height grids. An efficient way to calculate HAF is presented. A description is given how the trained grasp classifier is used to explore the whole grasp space and a heuristic to find the most robust grasp is introduced. This thesis describes how to use the approach to adapt the robotic hand opening width before grasping. In robotic experiments different aspects of the system are demonstrated on four robot platforms: A Schunk 7-DOF arm, a PR2, the mobile service robot Hobbit and a Kuka LWR arm. Tasks to grasp single objects, autonomously unload a box, clear the table and tidy up the floor were performed. Thereby it is shown that the approach is easily adaptable and robust with respect to different robotic hands. As part of the experiments the algorithm was compared to a state-of-the-art method and showed significant improvements. Concrete examples are used to illustrate the benefit of the approach compared to established grasp approaches. Finally, advantages of the symbiosis between the approach presented and object recognition are shown.

Kurzfassung

Roboter sind dabei, nach Jahrzehnten als Produktionshilfen in der Industrie auch die privaten Haushalte zu erobern. Allerdings beschränken sich ihre Fähigkeiten im Allgemeinen auf Staub saugen, Rasen mähen oder die aktive Zuneigungssuche elektronisch augmentierter Plüschtiere. Universal einsetzbare Roboter-Butler finden sich nur in Kinofilmen. Ein Grund dafür ist die beschränkte Interaktionsfähigkeit von Robotern mit ihrer Umwelt. Diese Arbeit trägt dazu bei, das Greifproblem zu lösen, indem Robotern ermöglicht wird, passende Greifposen zu erlernen. Besonders berücksichtigt werden dabei erschwerende Faktoren wie unvollständig wahrgenommene Oberflächen unbekannter Objekte und visuell nicht segmentierbare Objektansamlungen.

Ein Gesamtsystem zum Greifen von Objekten (auch unbekannte oder in Objekthaufen positionierte) wird präsentiert. Der Ansatz basiert auf der Topographie einer gegebenen Objektansammlung und abstrahiert relevante Strukturen für das Greifen mit Hilfe neu entwickelter Features: Height Accumulated Features (HAF) und Symmetry Height Accumulated Features (SHAF) ermöglichen den Einsatz maschinellen Lernens. Ein Greif-Klassifizierer wird mit Hilfe einiger tausend Beispielszenarien guter und schlechter Greifpunkte trainiert und generalisiert dieses Wissen mit Hilfe von Support Vector Machines für beliebige Objekte.

In dieser Arbeit werden die topographischen Features (HAF & SHAF) beschrieben und motiviert. Die Qualität ihrer Greif-Klassifizierung wird mit der F-Score-Metrik analysiert. Zusätzlich wird das Abstrahierungspotential von HAF und der Informationsgewinn durch den Vergleich eines mit HAF trainierten Klassifizierers mit einem Klassifizierer der mit diskretisierten Oberflächenpunkten trainiert wurde, gezeigt. Eine effiziente und namensgebende Berechnung von topographischen Features wird vorgestellt. Es folgt eine Beschreibung, wie der Greif-Klassifizierer verwendet wird, um den gesamten Grasp-Space abzudecken. Außerdem wird eine Heuristik eingeführt um die Robustheit von Greifposen zu optimieren. Eine passende Öffnungsweite des Manipulators für die Objektannäherung wird mit Hilfe einer Adaption des präsentierten Ansatzes berechnet. In Experimenten werden verschiedene Aspekte des Systems auf vier verschiedenen Robotern demonstriert: einem Schunk-Arm, einem PR2-Roboter, einem Kuka LBR Arm, sowie auf unserem selbst entwickelten Service-Roboter, Hobbit. Verschiedene Aufgaben, wie das Greifen einzelner Gegenstände, das Abräumen eines Tisches, das autonome Ausräumen einer Schachtel, oder das Aufräumen eines Wohnzimmerbodens werden durchgeführt. Dabei wird die Adaptierbarkeit des Ansatzes bezüglich verschiedener Manipulatoren gezeigt. Als Teil der Tests wird der präsentierte Ansatz mit einem State-of-the-Art-Verfahren verglichen und eine signifikante Verbesserung von 34% für freistehende Objekte, beziehungsweise $28,9\%$ für Objekte aus Objektansammlungen gezeigt. Anhand von Beispielen werden die Vorteile gegenüber etablierten Greifverfahren erläutert. Abschließend werden die Vorteile einer Kombination mit bekannten Objektmodellen und Objekterkennungsverfahren erläutert.

Acknowledgment

I would like to thank my supervisor Prof. Markus Vincze. I am grateful for his advice, the freedom and support he provided me during the work leading to my thesis and his approach to treating his students as fellow research colleagues and valuing their opinion. I would also like to thank my external reviewer Antonio Morales, Professor at the Universitat Jaume I, Castellon, Spain for his encouragement.

Special thanks go to Walter Wohlkinger and Astrid Weiss for their guidance and helpful advice during the journey of my PhD study.

I would also like to thank Karthik Mahesh Varadarajan and Astrid Weiss for their great efforts in improving my publications and Michael Zillich as a senior colleague whose publications triggered my work in robotics. I want to thank my supervisor Markus Vincze for enabling my participation in summer schools about visual recognition, machine learning and grasping (such as the INRIA CVML in Paris and the MLSS at UC Santa Cruz), conference visits and a collaboration as visiting researcher at Cornell University. At Cornell, I would like to thank Prof. Ashutosh Saxena, Yun Jiang and Rudhir Gupta for hosting me and giving me the opportunity to work with a PR2.

I am grateful for all colleagues who accompanied me during my work and often became friends. In particular, I would like to thank the following colleagues, in addition to those already mentioned: Stefan Ulbrich, Markus Bajones, Daniel Wolf, Paloma de la Puente, Georg Halmetschlager, Peter Einramhof, Rosen Diankov, Robert Schwarz, Luis Mateos, Andreas Baldinger, Thomas Feix, Javier Felip Leon, Jeannette Bohg and Peter Florian.

The research leading to this thesis has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 288146 (Hobbit) and No. 215821 (GRASP).

My gratitude also goes to my family, (above all, my mother) for the support from childhood and enabling my studies. Without the support of my whole family, this work would not have been possible.

Contents

Chapter 1

Introduction

In the near future service robots should be able to aid living comfort by supporting humans in their daily lives. They should be able to fulfill simple tasks, entertain the user and cope with known objects. It is desired that robots can extend their knowledge and learn to handle new tasks in a-priori unknown environments. One important, probably the major way to interact with their environment is object manipulation. In this thesis a novel approach for grasping known and unknown objects, freestanding and in clutter or from piles of objects is presented. It enables robots to better interact with their environment and enhance their manipulation skills.

In this chapter the problem of grasping is explained and defined. Three types of grasping problems are characterized and a short description of state-of-the-art work is given for each. A formal problem definition then defines the grasping problem in a technical way. Three main issues for the practical realization of grasping systems, namely perception, calibration and finding feasible inverse kinematic solutions are described. The contributions of the thesis are stated dedicated to the grasping problem types where eligible. A graphical system overview is used to explain the presented approach and links system modules with the corresponding sections in this thesis. Finally, an outline of the remaining thesis is given.

1.1 Problem statement

This thesis investigates the **problem of grasping objects**, defined by detecting the pose of a robotic hand in the seven-dimensional grasp space (position, orientation, gripper opening width) where a mechanical gripper has to close for a stable grasp, and the approach trajectory to reach the final grasp pose (see Fig. 1.1).

In a typical fetch and carry mission autonomous grasping is needed in a variety of types, depending on the user scenario. For example fetch and carry missions could start with one of the following user commands:

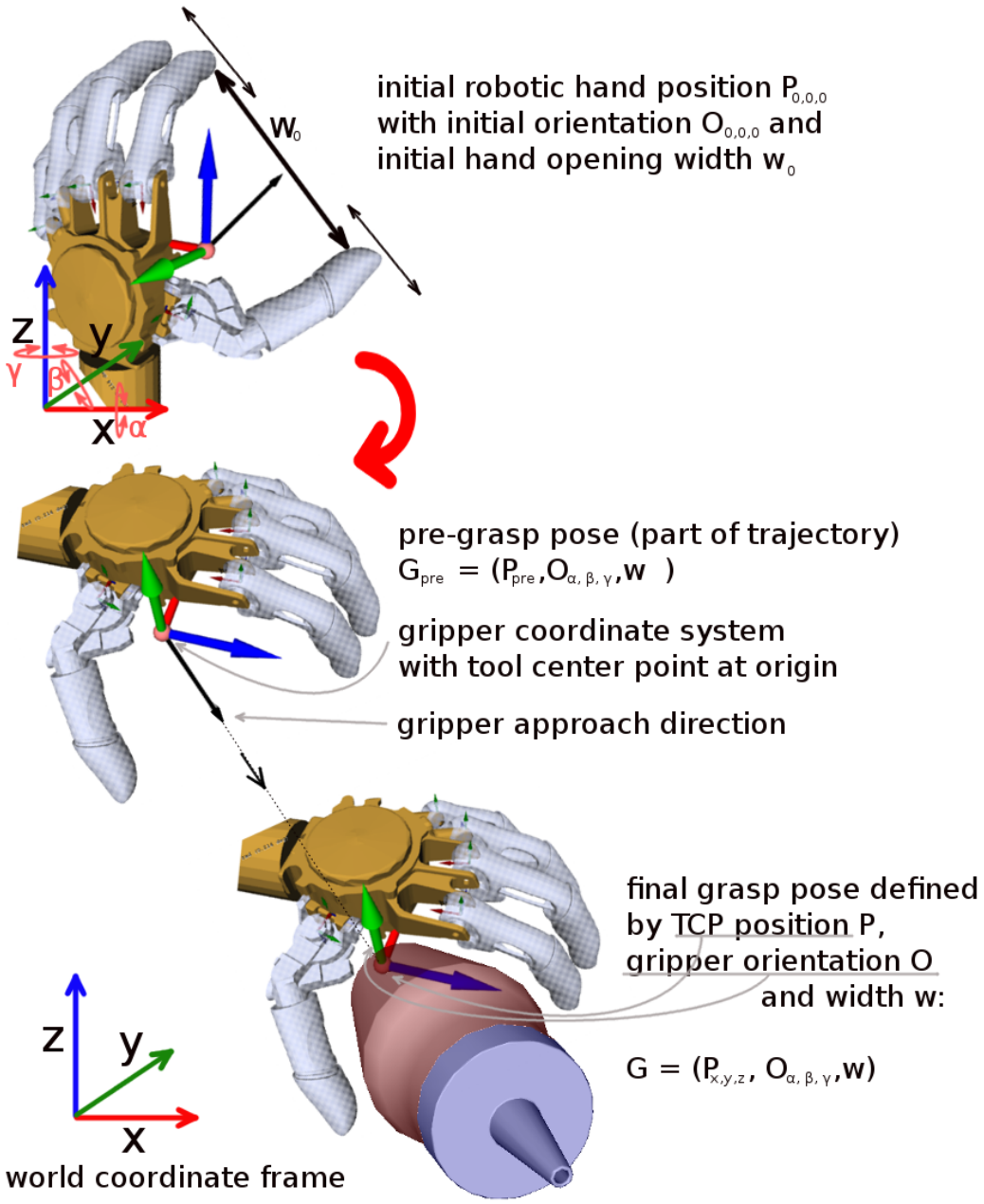


Figure 1.1: ***Problem description:*** *The picture shows the problem this thesis investigates: An object should be grasped. Therefore the position $P_{x,y,z}$ of the robotic hand in world coordinates, its orientation $O_{\alpha,\beta,\gamma}$ at the final grasp pose and the hand opening width when approaching the object have to be determined. In this pose, closing of the hand leads to a successful grasp. In addition, the approach trajectory has to be determined to avoid collisions with the object to grasp, and obstacles including supporting planes. Object data can be given in the form of complete object models or by partial 2.5D object views.*

1. "Robot, bring me my favorite cup."
   Problem type A: grasping known object

2. "Robot, pick up the object I am pointing at."
   Problem type B: grasping unknown object

3. "Robot, unload the box."
   Problem type C: grasping objects in clutter

All of these scenarios pose different research problems and challenges for autonomous grasping, which are up to now not fully solved. In the following, a short description for each of the three grasp problem types is given, mentioning few important results and summarizing achievements and open problems of state-of-the-art approaches.

1.1.1 Problem Type A: Grasping known objects

Grasping known objects relies on available object information, such as a CAD model database and pre-learned grasps. The problem of grasping is divided into the subproblems of segmenting an object, recognizing the object, estimation of the six-dimensional object pose, and finally the grasping process.

In the following, a brief overview of state-of-the-art for grasping known objects is given. In Papazov et al. (2012) a vision-based object recognition and localization system for impedance controlled grasping is used to grasp known objects. In an evaluation for grasping single standing objects, 58 out of 60 trials were successfully executed. A grasp trial was considered successful if the object was correctly recognized, grasped, and carried to a predefined place. Three objects were tested in the trials with ten repetitions for two different object poses, to test all pre-saved grasp poses.

In Klank et al. (2009) Time-Of-Flight and RGB cameras are used to fit CAD models in cluttered table setting scenes for the purpose of grasping with a mobile manipulator. It is noteworthy to mention, that the object recognition works for textureless objects and the CAD model matching is in real-time. The system provides a suitable grasping pose for a mobile manipulator, whereby the grasping points for a model are estimated offline using OpenRAVE (Diankov and Kuffner, 2008). An experimental grasping evaluation is not given.

According to Prankl (2011), detecting the object instance *my cup* including 3D pose alignment is "of course solved". To verify this statement the paper of Collet et al. (2009) is referenced. In this work four different objects (can, juice bottle, rice box, notebook) with simple geometric forms are recognized and the full 6D pose is estimated from a single view point. The can is only recognized if it was placed not further than $60cm$ away from the camera. It seems the notebook was placed standing upright for recognition and grasping, which is not the most stable

or common orientation for such an object. A grasping success rate of 91% was achieved. In the follow-up work (Collet and Srinivasa, 2010), a grasp success rate of 98% could be achieved for five different objects with multi-view input.

To conclude, the task of grasping known objects ("Robot, bring me my favorite cup.", see Fig. 1.2) can be seen as solved, given the following restrictions:

Figure 1.2: "Robot, bring me my favorite cup." Grasping known object

- **Object model:** Object models must be available for each object a-priori;
- **Pre-learned grasps:** Grasps have to be learned for each object in advance (due to different object poses, obstacles, and robot kinematics a high percentage of grasps will not be executable in real world scenes, hence these pre-lerned grasps must be comprehensive);
- **Gripper-dependent:** The pre-learned grasps are gripper dependent;
- **Object perception:** The quality of object data from perception devices must be sufficient (if objects are recognized by 3D shape a drinking glass will not be visible for a laser-pattern-based sensor like the PrimeSense devices);
- **Object features:** Objects must posses attributes needed for object recognition (e.g.: one colored objects will lack features needed for SIFT recognition);

- **Number of objects:** The number of objects is limited (to obtain discriminability for object recognition);
- **Graspable objects:** The objects must be graspable regarding size, form, weight, material (e.g. bowling ball);
- **Rigid objects:** Objects have to be rigid, otherwise the object model will not fit in all scenarios (e.g. for clothes).

1.1.2 Problem Type B: Grasping unknown objects

The problem of grasping unknown objects ("Robot, pick up the object I am pointing at.", see Fig. 1.3) is still an open research problem.

Figure 1.3: "Robot, pick up the object I am pointing at." Grasping unknown object

In Saxena et al. (2008b), local patch-based image and depth features are learned and used for grasping unknown objects. The method was tested for nine unknown objects and achieved a success rate of 87.8%. In Jiang et al. (2011) this work was improved by adding the capability of learning optimal gripper opening width. In an evaluation for grasping twelve different unknown objects a success rate of 87.9% was achieved.

In Klingbeil et al. (2011) a grasp detection approach for a two-finger gripper was used to grasp unknown objects based on raw depth data. The approach is based on finding a pattern in the scene that fits best into the interior of the end-effector. In an evaluation with six unknown objects a grasp success rate of 91.6% could be achieved.

To conclude, the currently available metrics and features to detect grasps are not sufficient to solve the grasping problem for unknown objects. There is still a considerable gap in complexity and success rates between arranged testing scenarios for scientific publications and the human ability to grasp novel objects.

1.1.3 Problem Type C: Grasping objects in clutter

The problem of grasping unknown objects in clutter[1] ("Robot, unload the box." see Fig. 1.4, "Robot, clear the table.") is a challenging problem. The perceptible data is diminished due to occlusions. The solved problem of path planning with obstacle avoidance for robot manipulators gets harder in practice; similarly the segmentation and identification of the desired object to grasp becomes more difficult.

In Le et al. (2010) the method from Saxena et al. (2008b) was extended to accommodate grasps with multiple contacts and a success rate of 80% was achieved for desk clearing experiments with two to eight objects. Klingbeil et al. (2011) compared their approach with the two previously mentioned approaches and could show an improvement for freestanding objects. For objects in clutter 46 out of 50 grasp trials were successfully executed. In Kootstra et al. (2012), edge and texture based features on 2D images are used in an early cognitive system to build a 3D object representation. Their system achieved grasp success rates of about 60% for scenes with up to three objects.

As for grasping freestanding unknown objects, available heuristics or metrics are not capable to finally solve the task of grasping unknown objects. Available simple features capable of edge detection (e.g. from Saxena et al. (2008b); Kootstra et al. (2012)) would need highly developed cognitive processing to solve complex grasping tasks. Although substantial progress ("Deep Learning") was achieved during the last years, available approaches such as neural networks are currently not sufficient to process simple features in a way that grasping unknown objects in clutter could be solved. More powerful features can improve state-of-the-art grasping without the need of highly cognitive systems, or deliver better basis features for such complex learning systems.

[1] To clarify the term "clutter" since it is used differently in related work: If a normal projection to the supporting plane results in a disjunctive partition of object points with boarders between each part, a scene was not cluttered!

Figure 1.4: "Robot, unload the box." Grasping objects in clutter.

1.1.4 Formal problem definition

As described above, the problem targeted in this work is the calculation of stable grasps for given unknown and known objects. In a more formal way, the problem can be described as follows:

Definition 1 (Problem definition)**.** *Given a point cloud $PC_O \subset \mathbb{R}^3$ of one or more graspable objects in the robot coordinate system C_R and a set of obstacles $\mathcal{OB}$. The problem is defined as finding a stable and executable grasp G for a given robot R.*

The point cloud PC_O of objects presented by a set of points in the three dimensional Cartesian space is registered with color information. It is not limited to a single view, but can also be generated from multiple views or known object data models. The point cloud may include multiple objects. If a specific object should be grasped, or grasps should be positioned at given areas, the point cloud PC_O has to be restricted in advance and partitioned accordingly into PC_O and obstacles $\mathcal{OB}$.

Definition 2 (Obstacles)**.** *Obstacles $\mathcal{OB}$ are all 3D structures not desired for grasp actions, including supporting planes such as floors and table tops.*

The representation of obstacles is not limited to point clouds. It can also be given by a structural definition of geometric forms (e.g. the box obstacle in Fig. 1.6).

Definition 3 (Robot)**.** *A robot $R = (H, A[, P])$ is defined by its active robotic hand H, the corresponding arm A and optionally a mobile platform P.*

To work with robots in simulation, a model of the robot as URDF file (Unified Robot Description Format) including its mesh and its kinematic model is required.

Definition 4 (Stable Grasp)**.** *We define a grasp as stable for a robotic hand H, if it fulfills the physical requirements for a force or form closure grasp (Mason and Salisbury Jr., 1985; Li and Sastry, 1988), and results in a stable grasp feasible for moving an object or other manipulation tasks.*

Definition 5 (Executable Grasp)**.** *A grasp is considered as executable if the physical robot R can reach and execute all trajectory points for a calculated grasp action without any collisions.*

Collisions include self-collision of robot links, collisions with the environment, premature collisions with the object to grasp, and collisions with objects not selected for grasping.

Definition 6 (Adjustable Opening Grasp)**.** *A grasp is defined as a vector $G = (\mathcal{P}, \mathcal{O}, \mathcal{W})$ given by the final grasp pose of a robotic hand H specified by its position $\mathcal{P}$ and orientation $\mathcal{O}$, and the gripper opening width $\mathcal{W}$ for approaching the object.*

If $\mathcal{W}$ is not explicitly specified, a maximal gripper opening width is assumed by default.

To enhance robustness of grasps and reduce unintentional pre-touching of objects with the robotic hand due to calibration inaccuracy or incomplete data, the adjustable opening grasp G is extended by a pre-grasp position: $G_{pre} = (\mathcal{P}_{pre}, \mathcal{O}, \mathcal{W})$. The position P_{pre} is calculated by setting back the hand in approach direction by the constant $C_{APoffset}$ without changing the orientation $\mathcal{O}$. An executed grasp has to reach the pre-grasp position P_{pre} first, and then approach the object the final $C_{APoffset}$ centimeters without changing the gripper's orientation. This way, finding inverse kinematic solutions for all trajectory points becomes a harder problem.

Since the approach vector is gripper-specific and $C_{APoffset}$ a constant, the additional parameters for $\mathcal{P}_{pre}$ could be seen as redundant and implicitly contained in the grasp representation.

1.2 Challenges for grasping systems

When developing the grasping system presented in this thesis, it was emphasized on an approach that preselects grasps that are executable on real robots (considering inverse kinematic constraints) and are robust with respect to incomplete data and

inaccurate grasping execution. In the following, these three issues, which make it more difficult for a system to calculate proper grasps that will succeed in the real world are discussed.

- **Perception:** Calculation of grasps for unknown objects has to deal with incomplete perceived object data. Depending on the perception method, object surfaces cannot always be determined. Stereo vision fails for textureles objects. Shiny, transparent or reflective surfaces can hardly be perceived by most 3D data acquistion methods. In advance, the quality of a perceived object mesh is often not predictable. As example, Fig.1.5 shows the perceived data of three objects for a Kinect sensor, which is based on the projection of a laser pattern. The pictures show a green toy clay block, a yellow toy clay block, and a yellow toy duck together with the perceived data. The surface of the green toy clay can be perceived with high quality, for the yellow toy clay (same brand, size, material) no data is perceived. Interestingly, the toy duck colored in a very similar yellow than the yellow toy clay is perceived with high quality. This example shows the difficulty to even predict the quality of perceived object data.

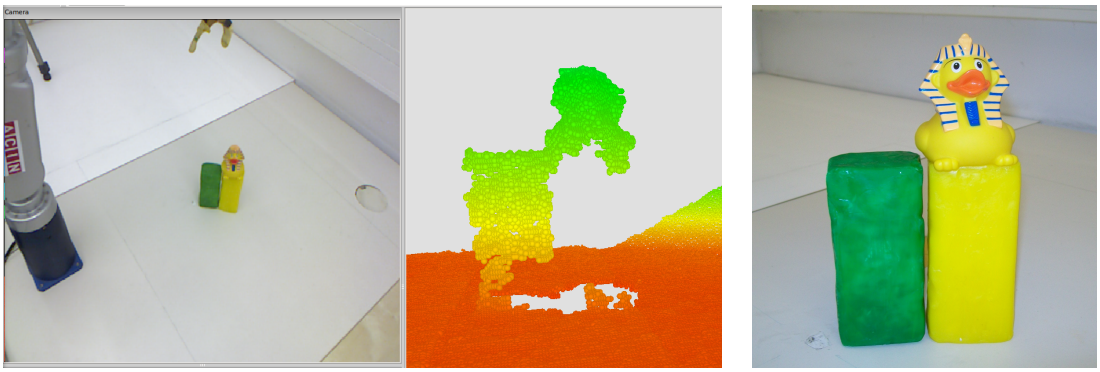


Figure 1.5: Example of incomplete object data perception that shows the difficulty to predict perception quality with respect to material and color. left: Picture from camera perspective of a green and a yellow toy clay and a toy duck. center: The perceived data. right: Photo of the three objects from the front.

- **Inverse Kinematics:** As mentioned above, finding kinematic solutions for defined gripper positions, especially if a trajectory should be followed without changing the gripper's orientation is a restrictive condition. Fig. 1.6 shows an example image of the limited solutions for inverse kinematics of grasping objects from a box with a 7-DOF arm mounted on a table. Four out of five depicted grasps could not be executed due to the required orientation and position for the gripper indicated by the approach direction.
- **Calibration:** Calibration errors are one reason for failed grasps. If the camera-robot calibration has to be recalculated this can lead to errors. In

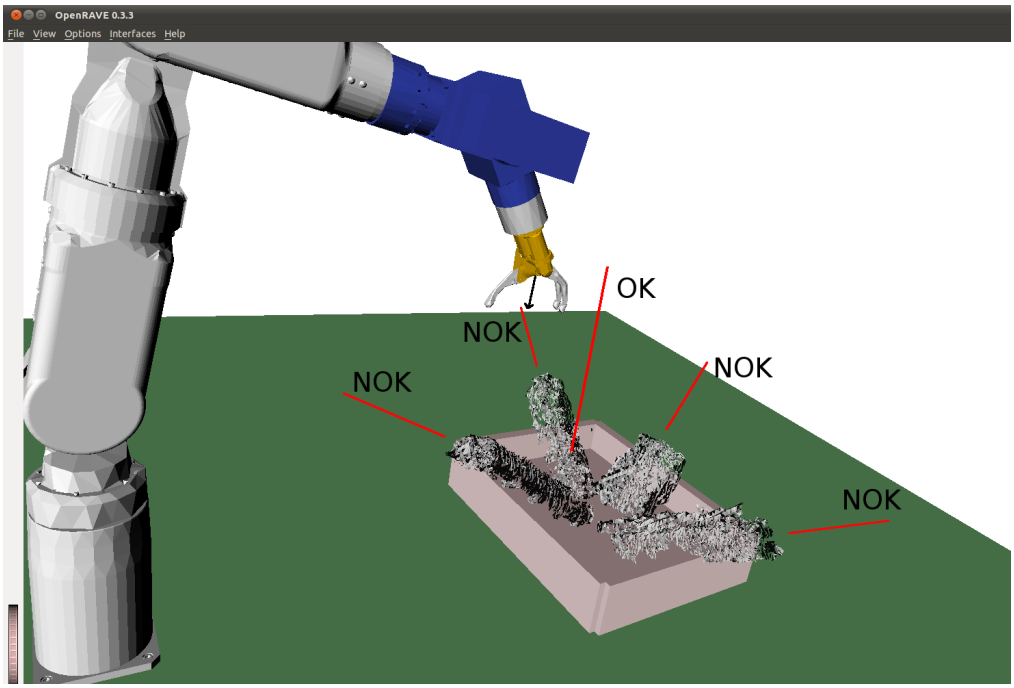


Figure 1.6: The picture shows four infeasible grasps due to the nonexistence of a kinematic solution. The basket is placed at an optimal position for top grasps regarding kinematic solutions for the static 7-DOF robot arm.

Fiedler and Müller (2013) the inaccuracy of Kinect sensors was investigated. The work shows that there is even a time and temperature dependent depth error (related to the time the camera was plugged in) of up to $2cm$ for a $1.5m$ distance. These results show the need for grasp detection approaches, that can deal with inaccurate data.

1.3 Contributions

The contribution of this thesis is divided into general contributions respectively contributions assigned to one of the grasping problem types stated before.

1.3.1 General contributions of this thesis

- **Height Accumulated Features:** The main contribution of this thesis are Topographic Features specifically developed for grasping without requiring a-priori knowledge of the objects: Height Accumulated Features (HAF, Section 3.2). The presented method abstracts topographical information from perceived surfaces of objects, hence enables to learn how to grasp them. The use of machine learning techniques enables generalization of learned grasping patterns.

- **Integrated autonomous system:** In this thesis an integrated system is presented that works autonomously and robust on four different robot platforms. The process overview diagram in Fig. 1.7 shows eight *steps* in the

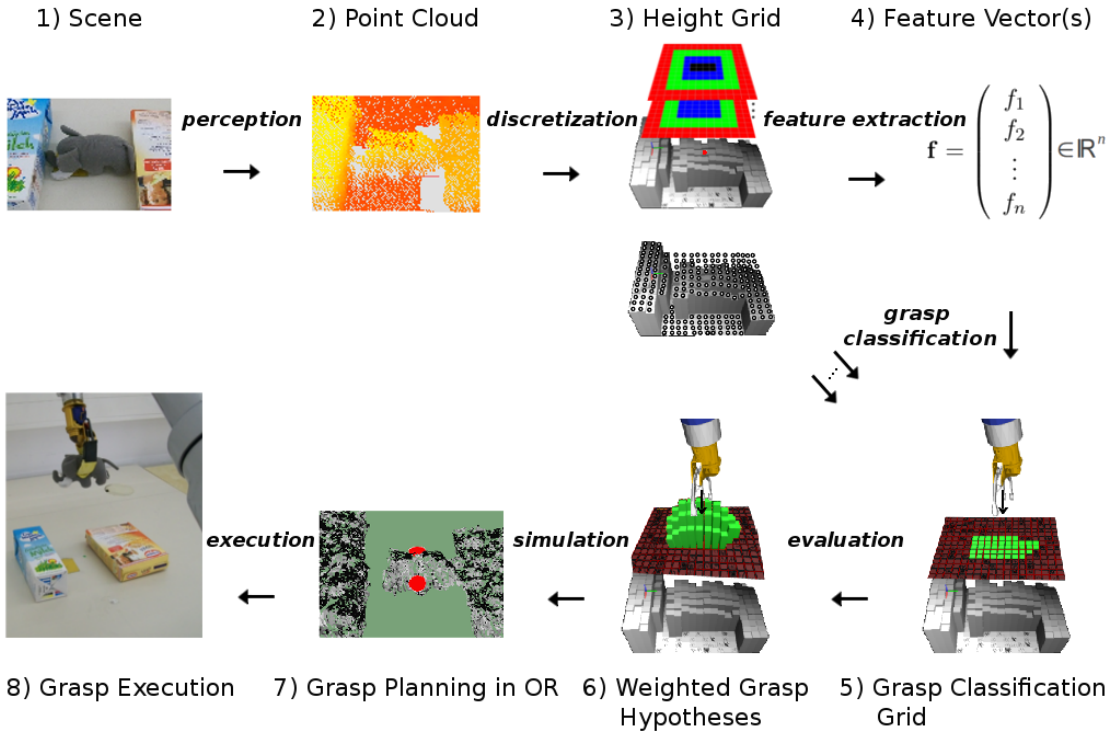


Figure 1.7: ***System Overview:*** *This process diagram shows eight steps in the grasping pipeline of the system presented in this thesis. For simplicity, the presentation shows only a solution for a fixed gripper rotation, tilt angle, and gripper opening width.*

grasping pipeline of the system. For simplicity, the presentation shows only a solution for a fixed hand rotation, tilt angle, and gripper opening width. The point cloud (Fig. 1.7, *step (2)*) of a scene *(1)* is discretized *(3)* first. To evaluate different hand roll/tilt angles (explained in Section 4.3) and gripper opening widths (Chapter 6), the point cloud is transformed before discretization (not depicted). For each grasp hypothesis (one hypothesis is indicated by a red point in *(3)*), a feature vector of length n (number of features) is calculated *(4)*, see Chapter 3 how features are defined and calculated). Then, a trained grasp classifier (Section 4.1) is used to get the grasp classification grid *(5)*, whereby green indicates possible grasp positions. A weighting system (Section 4.2) evaluates the grasp quality *(6)*, wherein better grasp positions are indicated by higher (green) bars. The overall top-rated grasp (along with estimated roll, tilt, gripper width) is sent to the simulation environment OpenRAVE *(7)*, in which detailed grasp planning (how close the hand can approach the object, see Section 4.4) is done. For possible grasps,

OpenRAVE sends the trajectories for path planning to execute grasping on the actual robot *(8)*, see the experiments in Chapter 8.

Other grasping systems (e.g. Collet et al. (2009),Collet and Srinivasa (2010)) need four different modules that can be sources of error:

1. object segmentation
2. object recognition
3. object pose alignment
4. object grasping

The presented system uses only an object grasping module, which is the key factor for the high robustness of the approach.

- **Symmetry Height Accumulated Features:** An extension of HAF to cope with specific scenarios. (Section 3.3)
- **Grasp weighting and selection method:** A heuristic to enhance the robustness of selected grasps, which also is of great importance regarding inaccurate perception and calibration. (Section 4.2)
- **Feature analysis:** The analysis and evaluation of Topographic Features with respect to their grasp classification power using an F-score metric. (Chapter 5)
- **Feature-Editor:** A GUI based application tool to create, edit and delete Topographic Features. (Section 3.3)
- **Efficient feature calculation method:** An efficient method to calculate Topographic Features using summed area tables. (Section 3.3)
- **Grasp training database:** A database with scenarios of good and bad grasps for grasp classifier training, including tools for processing RGB-D data and for automated generation of similar scenarios.
- **A grasp space exploration technique:** Explores the grasp space, given a grasp classifier for fixed hand orientation. (Chapter 4)
- **Gripper opening width determination method:** An extension for grasp space exploration calculating gripper opening width for approaching objects. (Chapter 6)
- **State-of-the-art comparison:** A comparison with Jiang et al. (2011) shows improvement of the grasp success rate of 34% for single objects (Section 8.4) and an improvement of 28.9% for objects in clutter. (Section 8.5)

- **State-of-the-art analysis:** An external examination for a popular grasp approach (Jiang et al., 2011). (Section 8.5.3)

- **A grasping framework** available for the community including the presented approach and the grasping approach from Jiang et al. (2011) based on 2D image features.
(available from *http : //pr.cs.cornell.edu/grasping/rect_data/data.php*)

1.3.2 Contribution: Grasping known objects

- **Skipping effort for creating pre-grasp database:** Manual definition of grasps per object and robotic gripper can be tedious. The presented approach for finding grasps can be applied on full object models for online grasp calculation, making a database of grasps per object and gripper obsolete.

- **Semi-automatic creation of pre-grasp database:** The effort for generating a database with pre-learned grasps per object and gripper can be reduced with automatic or semi-automatic grasp generation. Available metrics to detect grasps, such as the ϵ-metric tend to be relatively fragile in the real world (Diankov, 2010; Balasubramanian et al., 2012; Weisz and Allen, 2012). Our approach can be used for (semi)-automatic grasp generation given the gripper- and object model.

- **Object recognition support by interacting with unrecognized objects:** Due to lightening conditions, occlusions, unlearned views or other reasons, recognition can fail. The presented approach gives the opportunity to change object poses by grasping without pre-learned positions, thereby creating new views that can enable successful object recognition, e.g. for task specific grasping, where object recognition is crucial.

1.3.3 Contribution: Grasping unknown objects

- **Use of known depth regions: no need to guess shapes:** The presented approach focuses on grasps on perceived surfaces where the gripper approaches objects without the need to estimate the surface of the object which is facing away from the camera (compare Fig. 3.1).

1.3.4 Contribution: Grasping objects in clutter

- **Segmentation independent:** The presented approach enables complex tasks such as autonomously emptying a basket filled with objects without the need for segmentation. Thereby, it provides a complementary approach to methods which need segmented input, such as Superquadric fitting.

- **Interactive segmentation:** The approach provides the ability to work as a preprocessing module for object recognition by separating one object from a pile of items.
- **Integrated path planning:** The majority of recently published grasping algorithms (e.g. Varadarajan and Vincze (2011); Kootstra et al. (2012)) handle grasp planning and path planning independently. Grasp approach directions and grasp points are calculated first, path planning with computation of inverse kinematics and obstacle avoidance is done afterwards. In contrast, the presented method learns to select grasp hypotheses which result in collision-free local paths for the gripper used and the given approach vector.

1.4 Outline

This PHD thesis gives on overview on autonomous grasping of known and unknown objects in cluttered indoor environments with Topographic Features.
In **Chapter 2** further work related to grasping of objects is discussed.
Chapter 3 describes the idea, motivation, and calculation of Height Accumulated Features and Symmetry Height Accumulated Features.
Chapter 4 explains the process of how the feature values are used to determine optimal grasp configurations. The machine learning methods used are described, as well as a weighting heuristic to enhance the robustness of grasps, a technique to explore the entire grasp space using a trained classifier and the fine calculation of grasp points using a simulation environment.
In **Chapter 5**, the most efficient features for grasping are analyzed, the effect of Symmetry Height Accumulated Features is discussed and the additional gain obtained from HAF by a comparison to learning grasps directly on height values is shown.
In **Chapter 6** the framework is used to extend grasp options by taking the opening width of the gripper into account, resulting in the adjustable opening grasp representation.
In **Chapter 7** the scalability of the approach regarding different types of robotic hands is discussed.
In extensive experiments in **Chapter 8** the approach is evaluated on four different robotic platforms considering different aspects of each task.
Finally, **Chapter 9** gives a conclusion of the work and discusses future work.

Chapter 2

Related Work

There are numerous works about grasping of known and unknown objects. Extensive grasp surveys are given by Shimoga (1996), Bicchi and Kumar (2000) and more recently by Bohg et al. (2014).

2.1 Grasping known objects

Many current approaches rely on object recognition and the subsequent application of pre-calculated grasps: Papazov et al. (2012) present a 3D object recognition and pose estimation approach for grasping known objects in cluttered and occluded environments. Their approach solely relies on 3D geometry information and is based on a geometric descriptor, a hashing technique and a localized RANSAC-like sampling strategy. In a variety of tests they show that their proposed method performs well on noisy, cluttered and unsegmented range scans in which only small parts of the objects are visible. The experimental validation with a 7 DOF impedance controlled robot shows how their method can be used for grasping objects from a complex random stack. Berenson et al. (2007) also considers grasping in cluttered scenes with known 3D models. Morales et al. (2006) uses the method proposed by Azad et al. (2007) to recognize an object and estimate its pose from a monocular image. Given this information, a grasp configuration is selected from a grasp experience database that has been acquired offline. The whole system is demonstrated on the robotic platform described in Asfour et al. (2008). Huebner et al. (2009) demonstrate grasping of known objects on the same humanoid platform and using the same method for object recognition and pose estimation. The offline selection of grasp hypotheses is based on a decomposition into boxes.

In Detry et al. (2009) object grasp affordances are learned and represented, i.e. object-gripper relative configurations that lead to successful grasps. The representation consists in a probability density function defined on the 6D gripper pose space, within an object-relative reference frame. Grasp hypothesis densities are learned from human imitation as well as from visual cues, and empirical grasp

densities are learned from physical experience by a robot. In Detry et al. (2010) a robot learns to grasp specific objects by experience. Instead of human imitation learning, successful grasping trials are used to build the object-specific empirical grasp density.

Wohlkinger and Vincze (2010) use object categorization to find objects similar to known objects and calculates grasps for the given models. Using known models is one approach to overcome the issue of incomplete data perceived from single views of a robot. The availability of object models enables the use of force and form closure grasp quality metrics (Mason and Salisbury Jr., 1985; Li and Sastry, 1988; Pollard, 2004). However, object recognition and in particular segmentation are still open research problems for grasping, for the latter see Bohg et al. (2014) and cannot guarantee reliable results for scenarios like the ones depicted in Fig. 1.4.

2.2 Grasping unknown objects

Object shape approximation: A great deal of the work examining the challenge of grasping unknown objects tries to approximate original object shapes to calculate grasps on the resulting models. In (Miller et al., 2003; Huebner and Kragic, 2008; Przybylski et al., 2011) shape primitives like boxes, spheres, cylinders and cones are used to approximate object shapes. In (Goldfeder et al., 2007; Varadarajan and Vincze, 2011) this approach is extended by the use of Superquadrics as a more general basic geometric form for grasping. The resulting shape primitives were used to limit the number of candidate grasps to find the most stable set of grasp hypotheses. Approaches from (Bohg et al., 2011; Rao et al., 2010) are based on the observation that many objects possess symmetries and use this assumption for object completion before grasp calculation.

Shape matching: Another approach related to the work presented in this thesis is from Klingbeil et al. (2011). They propose a grasp detection approach for a two-finger gripper to autonomously grasp unknown objects based on raw depth data. Their method tries to find a pattern in the scene that fits into the interior of the end-effector by maximizing the contact area between the robot's gripper and the perceived point cloud. This approach treats grasping as a shape matching problem similar to the work by Li and Pollard (2005), but does not require object models. Another noteworthy shape-matching approach related to the presented research is the work from Herzog et al. (2012). They proposed a template-based grasp selection algorithm operating on heightmaps which uses demonstrated grasp configurations and generalizes them to grasp novel objects.

Katz et al. (2013) deal with the problem of clearing a table. They achieved a grasp success rate of 53% for cluttered scenes and learn push and pull actions in addition. Calli et al. (2011) propose a grasping algorithm which uses curvature information obtained from the silhouette of unknown objects. In Richtsfeld and Vincze (2008) a framework is presented to calculate grasp points of unknown objects

in 2.5D point clouds from laser range data. Their method calculates grasp points from the top surfaces of objects and can be applied to a reasonable set of objects. Maldonado et al. (2010) present a system for autonomous rigid-object pick-up tasks in domestic environments focusing on the gripping of unmodeled objects. They model the object as a 3D Gaussian. For choosing a grasp configuration, a criterion is optimized in which the distance between palm and object is minimized while the distance between fingertips and the object is maximized.

Feature-based learning of grasps: In Saxena et al. (2008a,b), supervised learning with local patch-based image and depth features for grasping novel objects in cluttered environments is proposed. This work is improved in Jiang et al. (2011) by adding the capability of learning optimal gripper opening width. The focus lies on learning features from 2D images, but one of the features is based on a comparison of object heights in predefined rectangular regions (in the remainder of this thesis, this method is referred to as the "Rectangle Representation"). This related feature, the performance and popularity of the approach in recent years and the ability to work in cluttered scenes made this work an excellent choice to compare the presented work with. In Le et al. (2010) the method from Saxena et al. (2008b) is extended to accommodate grasps with multiple contacts and a success rate of 80% is achieved for desk clearing experiments with two to eight objects counting success/failure of the first grasp attempt for each object. In Kootstra et al. (2012), edge and texture based features on 2D images are used to build a hierarchical representation in 3D and their system is evaluated for scenes with one to three objects with grasp success rates of about 60%.

Most of the approaches in the last paragraph use learning techniques to optimize grasping capabilities based on simple features. In the following, powerful features for grasping are described in more detail. These features can be seen as an enhanced feature type suitable to replace simpler features for established grasping systems.

Chapter 3

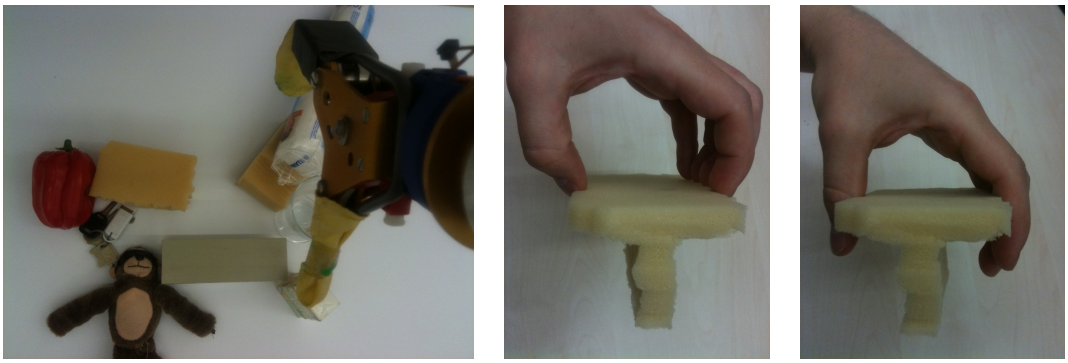
Topographic Features

In this chapter, two feature types based on the topography of objects or scenes are introduced and motivated. Height Accumulated Features (HAF) were developed specifically for abstracting grasp relevant information. Symmetry Height Accumulated Features (in short, Symmetry Features or SHAF) are an additional feature type, to solve specific problematic cases of the basic classifier trained with HAF only. The basic concept of HAF was first published in Fischinger and Vincze (2012a), the concept of SHAF in Fischinger et al. (2013b). In Chapter 5, work regarding feature selection and performance is discussed.

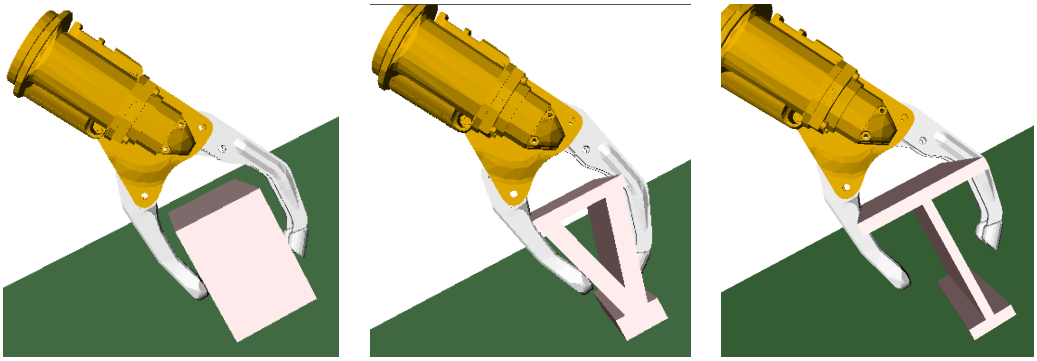
3.1 Motivation of HAF

In recent years learning of grasps became very popular (Bohg et al., 2014). Applying machine learning directly to RGB-D data is still impractical due to the huge number of perceived points in a point cloud and the at least six-dimensional grasp output space. To reduce the complexity, numerous approaches learn grasping hypotheses based on 2D images taking into account color and intensity values. Although the human brain can detect potential grasp points from images (and hence this approach could be promising in the future with sufficient cognitive processing methods), simple features based on 2D image patches seem to have clear limitations for grasping (see Section 8.5.3). Therefore, a new feature type is presented in this thesis which reduces the complexity of point cloud input, increases the structural value of input information as shown in Section 5.1 and is well suited for grasp-related machine learning due to the employment of grasp-relevant topographical information. Another challenge for learning grasps is the (at least) six-dimensional grasp output space (with three parameters for position and three parameters for orientation) where all parameters are strongly related. In Section 4.3, the method to explore the whole grasp space using a trained grasp classifier is described.

The HAF approach is based on the observation that for grasping from top, parts of the end-effector have to enclose an object and hence go further down than the



(a) Top view from robot perspective (b) Fragile Grasp (c) Stable Grasp



(d) Box form (e) V-form (f) I-form

Figure 3.1: HAF Motivation: in Fig. 3.1(a) we see two rectangular surfaces of a gray and a yellow object from a typical robot's top perspective, no side surfaces are visible. Fig. 3.1(b) shows an unstable grasp some force closure based simulation environments would recommend and Fig. 3.1(c) shows a stable top grasp a human would execute even if only the top surface of the object was perceived. Fig. 3.1(d)-3.1(f) show grasps for different object shapes, all of which have the same rectangular top surface shape.

top height of the object. Furthermore, unlike other approaches, this approach does not try to guess the shape of partially visible objects. Firstly, because this can fail in cluttered scenes or for asymmetrically shaped objects. And secondly, because the approach is based on the observation that a guess of the object shape is often not needed, for example if one surface is known and the robotic hand can be placed around that surface: Fig. 3.1(a) shows a picture from a typical view from a robot. For one gray and one yellow object, only the rectangular top surfaces are visible. Fig. 3.1(d)-3.1(f) show different scenarios for grasping when only the top surface is known and a robotic hand goes down up to the object and closes: in Fig. 3.1(d) the object is box-shaped and the hand touches the object roughly at the body center instead of the rim. In Fig. 3.1(e) the hand more or less adapts to the object form. And even if the object consists primarily of a large top surface and few encompassing surfaces (see Fig. 3.1(f)) a grasp would succeed if there is enough

space to place the fingers around it. Despite the knowledge gap, even a human would not grasp only at the visible object parts like depicted in Fig. 3.1(b) (like some force closure based simulation environments sometimes recommend, even if the whole object model is available, compare Fig. 8.25), but would go further down with his hand and use tactile feedback to stop the closing movement (Fig. 3.1(c)).

3.2 Height Accumulated Features

The idea of Height Accumulated Features (also stated in Fischinger and Vincze (2012a); Fischinger et al. (2013b)) is to define small regions and compare average heights of these regions using discretized point cloud data. The height differences give an abstraction of the shape of the objects that enables the training of a classifier (using supervised learning) to determine if grasping would succeed for a given scene. For explanatory reasons consider the special case of top grasps (vertical approach direction of robotic hand) of an object on a table. The term *height* can then be used intuitively and measures the perpendicular distance from the table plane to the points on the top surface of the object. A force or form closure grasp can only be achieved if parts of the hand will go further down towards the table than the top surface of the object. Hence the region of the object top will on an average be higher than the area where the robotic fingers are positioned. To speed up calculation, we discretize the point cloud, i.e. we generate a height grid H where each 1x1cm cell saves the highest z-value of points with corresponding x- and y-values (see Fig. 3.2). One Height Accumulated Feature is now defined as two, three or four regions R_i on the height grid together with a weighting factor w_i for each region. A feature value is defined as the weighted sum of all regions. So the j^{th} HAF value f_j is calculated as

$$f_j = \sum_{i=1}^{nrRegions_j} w_{i,j} \cdot r_{i,j} \tag{3.1}$$

with

$$r_{i,j} = \sum_{k,l \in \mathbb{N}:(k,l) \in R_{i,j}} H(k,l) \tag{3.2}$$

where $nrRegions_j$ is the number of regions for feature f_j. $R_{i,j}$ indicates the i^{th} region for the j^{th} feature and is defined by the set of all pairs of height grid cell indices belonging to the region.

The HAF vector $\mathbf{f}$ contains the sequence of HAF values:

$$\mathbf{f} = (f_1, f_2, \ldots, f_{nrFeatures}) \tag{3.3}$$

For the initial set of experiments, the focus was on features with two overlapping regions, where one region is completely inside the other region and a weighting factor $w_{i,j}$ is chosen such that the feature value is zero if both regions have the same

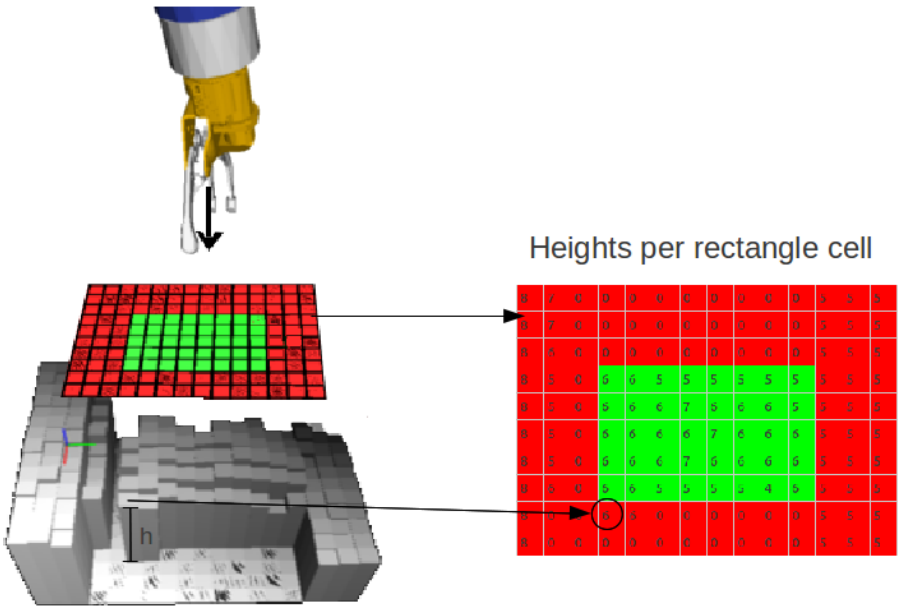


Figure 3.2: Shows the gray height grid resulting from point cloud discretization and an example feature with two regions. Region R_1 is the green inner region, region R_2 is composed of the red and the green area. For each grid cell of the regions the height is summed up per region. Each region sum is weighted with an individual factor and the sum (difference) of all regions (here two) gives the feature value.

average height, greater than zero if the inner region is higher, and smaller than zero if the inner region is lower on an average. For these features, the intuitive interpretation mentioned earlier holds. In Chapter 5, it is shown that more complex features with three or four regions or Symmetry Features lead to even higher discriminative results measured in terms of the F-score evaluation (Chen and Lin, 2006) metric. Overall, about 71,000 features (70,000 of them automatically generated) were tested for the experiments in this thesis. The top 300 to 325 features were selected with F-score selection to train a classifier weighing up time against detection performance.

A feature editor (see Fig. 3.4(a)) with graphical user interface for convenient editing of the number, size, position, and weighting factor of the features' regions was developed. Fig. 3.4(a)-Fig. 3.4(c) shows examples of features with two to four regions.

The representation of height grids is of significant importance to the approach. To accelerate computation, accumulated height values for given scenes are used. This principle was first introduced as summed area tables in Crow (1984) for texture mapping in computer graphics and became popular in the vision community as "integral images" by Viola and Jones (2004), which are successfully used for real-time face detection.

Instead of a vanilla height grid H, an accumulated height grid AH is calculated,

in which each location (x,y) of AH contains the height sum above and to the left of (x,y) in the grid.

$$AH(x,y) = \sum_{x' \leq x, y' \leq y} H(x', y') \tag{3.4}$$

Using height accumulated rectangular regions, each region sum can be computed with four or fewer array references, see Fig. 3.3.

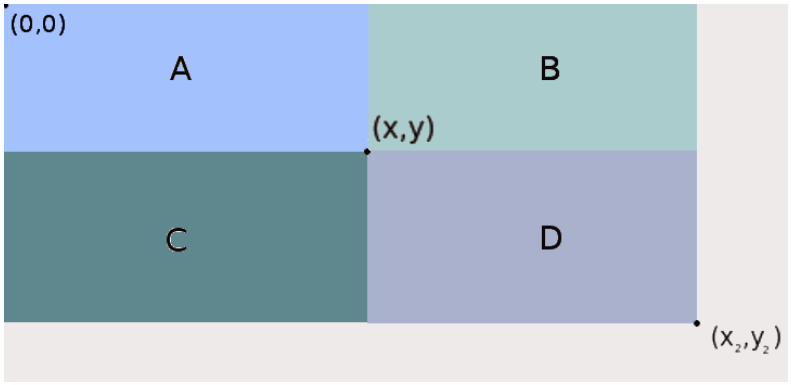


Figure 3.3: To calculate the accumulated heights of region A a single AH reference is needed: AH(A) = AH(x,y), Area D requires four: AH(D) = AH(x_2,y_2) - AH(x_2,y) - AH(x,y_2) + AH(x,y)

To demonstrate the advantage of accumulated heights, we give an example for the needed calculations: for a fixed gripper orientation and a $n \times m$ cm grid, for each 1×1 cm cell it should be tested if its center is a valid grasp center point. Therefore, for each feature, the accumulated heights for two to four regions have to be calculated. The height accumulated table for an $n \times m$ cm grid can be recursively calculated with less than $4 \times n \times m$ calculations. For a simplified but realistic example we use rounded values, assuming an average feature region size of 70 grid cells and only two regions per feature:

$$\begin{aligned} n &= 32 \\ m &= 44 \\ \#features &= 323 \\ \#regions &= 2 \\ avRegSize &= 70 \end{aligned}$$

These values result in an estimated number of calculations needed for grasp classification with fixed gripper orientation for an 32×44 cm area without a height accumulated grid:

$$\begin{aligned} \#calc_{NoAccum} &= n \times m \times \#features \times \#regions \times avRegSize = \\ &= 32 \times 44 \times 323 \times 2 \times 70 = \\ &= 63,669,760 \end{aligned}$$

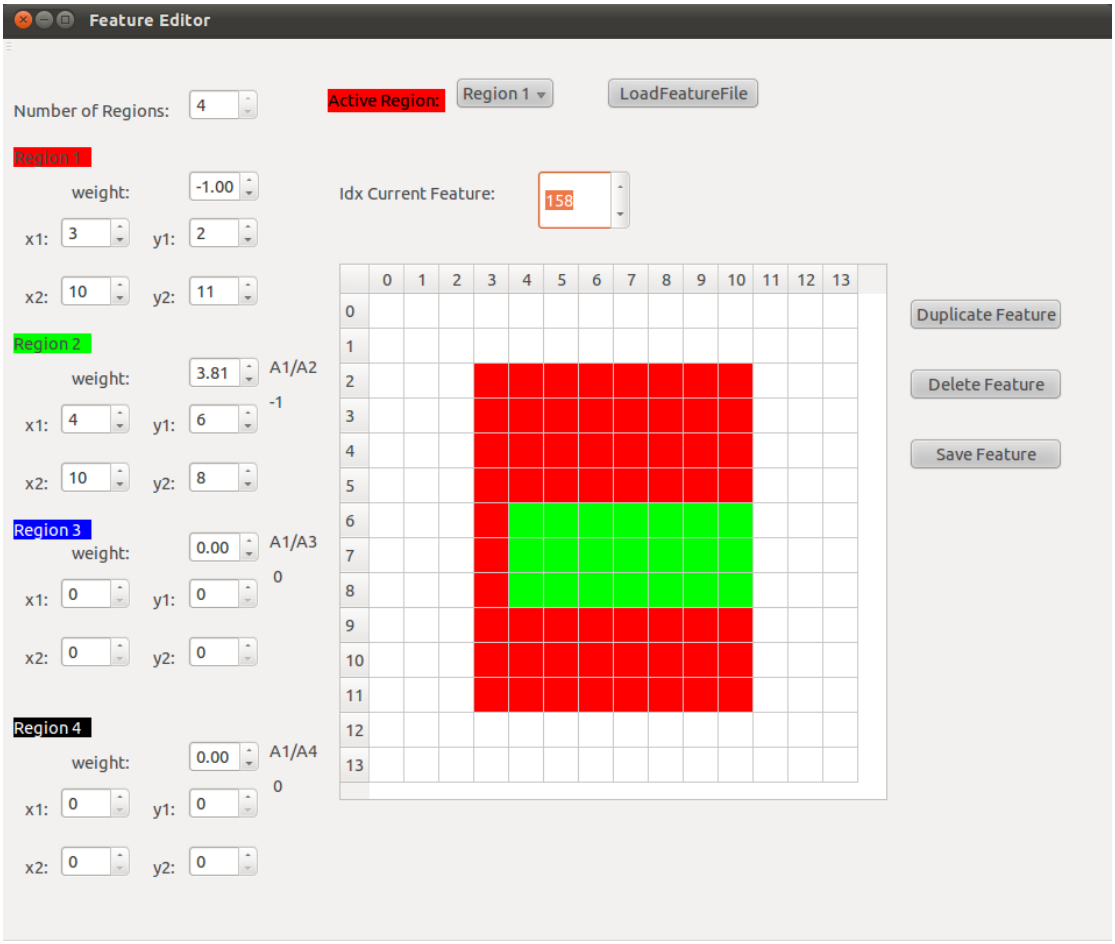
And in comparison the number of calculations if a height accumulated grid is used:

$$\begin{aligned} \#calc_{Accum} &= n \times m \times \#features \times \#regions \times \#CalcForOneRegion + \#CalcHAGrid = \\ &= 32 \times 44 \times 323 \times 2 \times 4 + 4 \times 32 \times 44 = \\ &= 3,638,272 + 5,632 = \\ &= 3,643,904 \end{aligned}$$

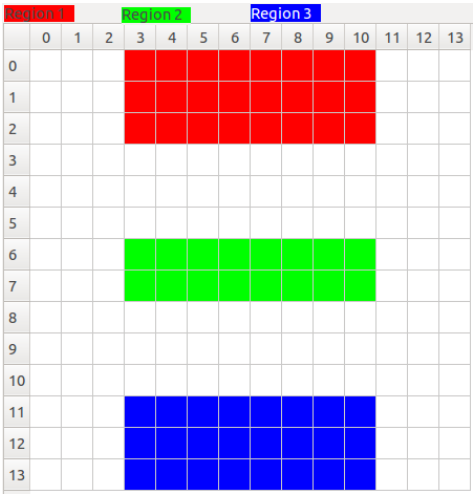
The quotient of the needed calculations is:

$$\begin{aligned} CalcQuot &= 63669760/3643904 = \\ &= 17.47 \end{aligned}$$

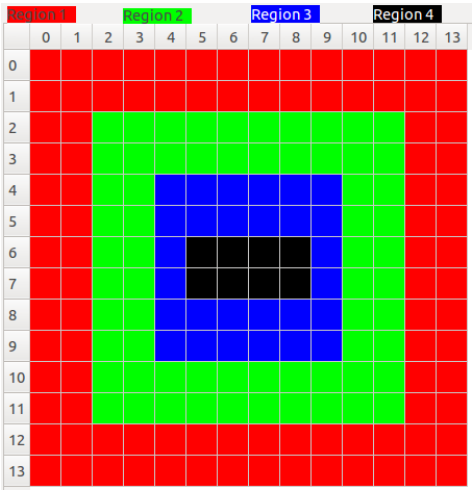
This example shows that the calculation effort for the accumulated height grid is small compared to the overall needed calculations, and that the initial accumulation effort reduces the number of calculations by a factor of about 17. This performance boost is improving the system, especially since the overall number of calculations needed, taking into account different gripper orientation and gripper width (see next chapter) easily reaches more than 500 million calculations when a height accumulated grid is used (or nearly 10 billion calculations without the accumulation pre-processing).



(a) HAF-Editor with an example feature with 2 overlapping regions



(b) HAF with 3 regions



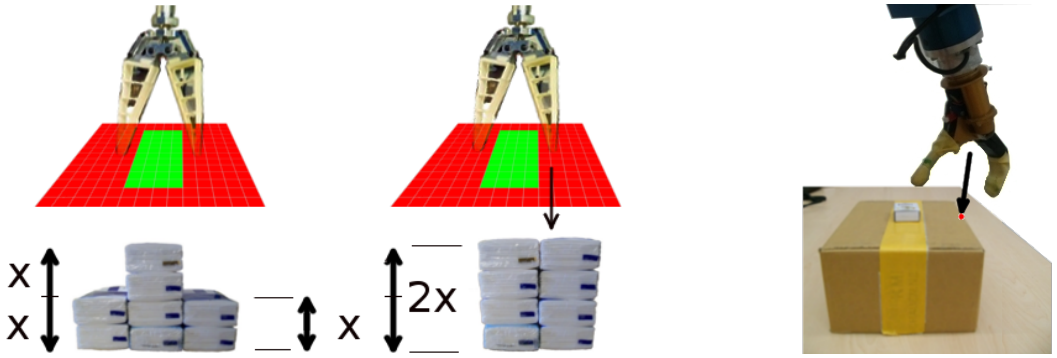
(c) HAF with 4 regions

Figure 3.4: For editing of Height Accumulated Features a GUI-Editor was developed. Examples are shown for features with 2-4 regions.

3.3 Symmetry Height Accumulated Features

For special constellations such as a small object on top of a box on a table (Fig. 3.5(b)), the HAF approach favors grasps at the edge of the box instead of grasps at the small object. HAF have the drawback that they are based on average heights of nested regions, hence there is no symmetry check when feature values are calculated. The same feature value can be achieved if the center region height exceeds both side region heights by x or if the center region and one side region have equal heights and exceed the second side region by $2x$ (see Fig. 3.5(a) for illustration).

The feature value is no indication if the two fingers of the gripper (see Fig. 3.5(b)) could go deeper than the object center on opposite sides of the object. This leads to false positives when using HAF for grasp detection e.g. at the edges of a closed box. For "clearing-a-table scenarios" such "wrong" grasps do not occur due to the weighting system and the constellation and size of objects. As long as there is an easily graspable object on the table, this object is grasped first. Nevertheless, it is a weak point of HAF not taking symmetry into account. As improvement, HAF is extended by an additional feature type: Symmetry Height Accumulated Features.



(a) HAF-Problem: Equal feature values (b) Impact on grasp estimation

Figure 3.5: SHAF-Motivation (left): For the depicted feature with a green and a red region the feature value would be equal for both piles of tissues, but only the left pile is suitable for a grasp with the depicted gripper. The picture on the right shows the impact of this HAF property: A bad grasp being detected at the edge of a box

Symmetry Features have three disjunctive regions of equal size as depicted in Fig. 3.6 where r_r,r_g,r_b are the accumulated heights of the region grid. The feature value f is defined as follows:

$$\mathbf{f} = \begin{cases} min(r_g\text{-}r_r, r_g\text{-}r_b) & \dots \quad if\ r_g > max(r_r, r_b) \\ \text{-1} & \dots \quad else \end{cases}$$

So we assign the minimal distance of accumulated heights between the center region and the side regions if the average center region height is the largest of the

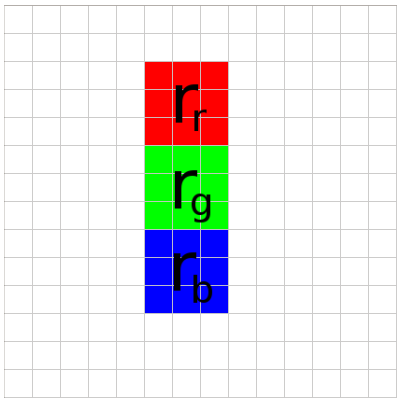


Figure 3.6: Symmetry Height Accumulated Feature: A typical example of the new feature type SHAF that solves the shown deficiency. All SHAFs have three equally sized, disjunctive regions.

three regions, and -1 otherwise. Note that this function is either positive or -1 and that the regions are not weighted. In Section 5.2 the impact of Symmetry Features on a grasp classifier is discussed.

Chapter 4

From Classification to Actual Grasp Execution

In this section, the process of selecting the best grasp options G in the seven-dimensional grasp space (with maximal gripper opening width as default value) using Topographic Features is described. This process was stated in Fischinger and Vincze (2012a) and is summarized here with additional clarifications. Firstly, the learning process for grasping (Section 4.1) is described. Then, the weighting heuristic to achieve more robust grasps (4.2) is explained. Finally, the method to explore the whole grasp space is described (4.3) and it is shown how fine calculation of grasp points and path planning is done with the OpenRAVE simulator (4.4).

4.1 Grasp Classification Training

In order to train a grasp classifier, 450 positive and 250 negative grasp scenes were gathered. A scene is composed of one or more objects on a table with the z-axis being perpendicular to the table and the origin located on the table surface. For supervised learning, the 450 positive examples were labeled, to be more specific: an x,y position was labeled such that a robotic gripper (in this case, an Otto Bock hand prosthesis) positioned above the objects (with the tool center point at x,y) and oriented in such a way that the line between the tip of the thumb and the tip of the forefinger is aligned with the x-axis, would achieve a stable grasp of an object by approaching it (approach vector of the gripper parallel to z-axis) up to 1cm and closing the fingers afterwards. Fig. 4.1 illustrates the actual classification task. The 250 negative grasp scenes were labeled at positions with hardly any chance of a successful grasp. The training set was augmented by scaling, mirroring, truncating, and inverting the manually labeled positions to generate in all 8300 positive and 12800 negative grasp example scenes. HAF and SHAF values were calculated on the 21100 examples and used to train an SVM classifier with a radial basis function kernel from the implementation of LIBSVM (Chang and Lin, 2011).

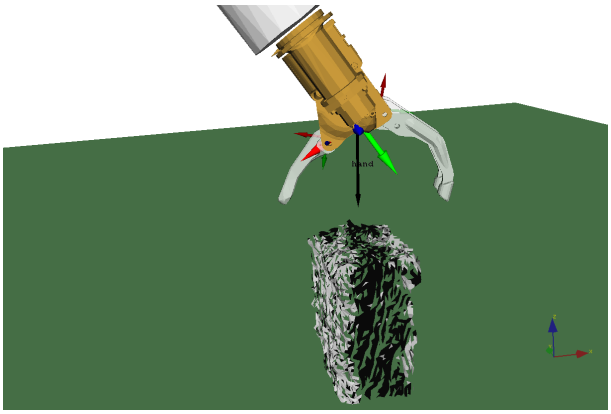


Figure 4.1: The grasp classifier learns voxel configurations if a hand motion in approach direction (black arrow) with subsequent closing of the fingers (1cm before hand-object collision) would result in a stable grasp at the voxel. Note that the type of gripper only influences the training examples regarding scale. Training examples will not change for other two-finger grippers.

Exemplary pictures for positive and negative scenes can be seen in Fig. 4.2 respectively Fig. 4.3. It should be emphasized that the granularity of the discretization guarantees the genericity of the approach. In other words: we do not learn grasps for specific object instances or objects similar to the objects used for training examples.

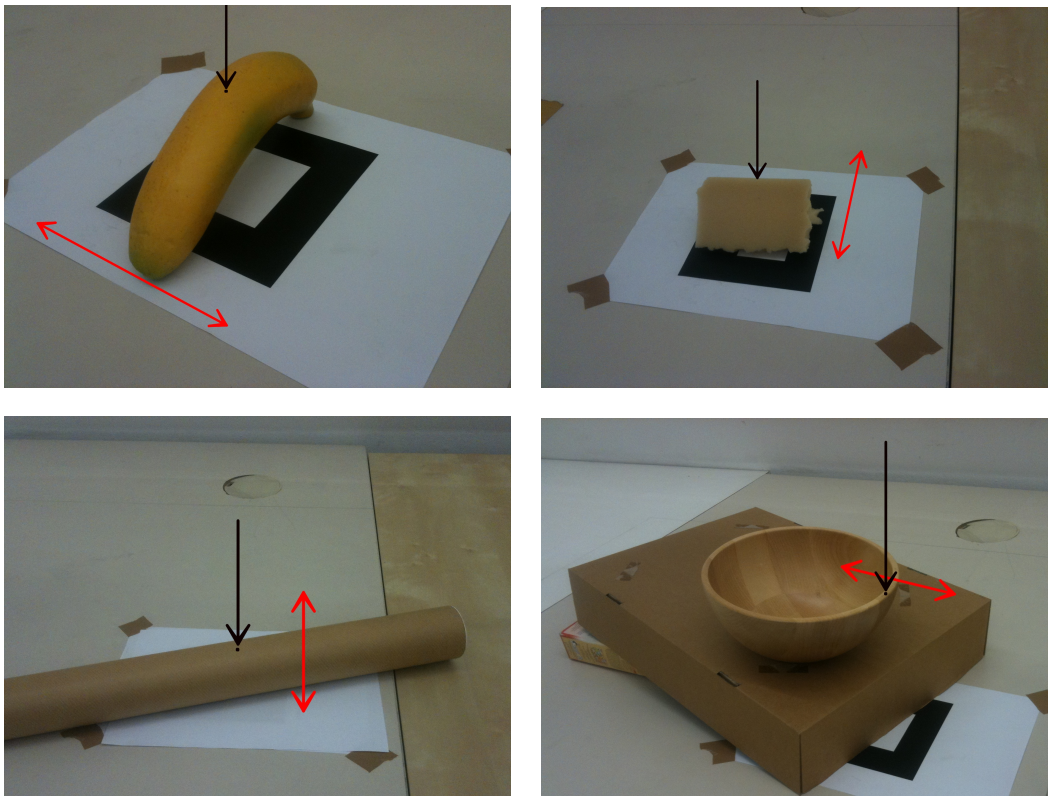


Figure 4.2: Positive training examples for the training of the (S)HAF classifier. The black arrow perpendicular to the table plane indicates the approach vector, the red arrow parallel to the table plane indicates the closing direction of the gripper.

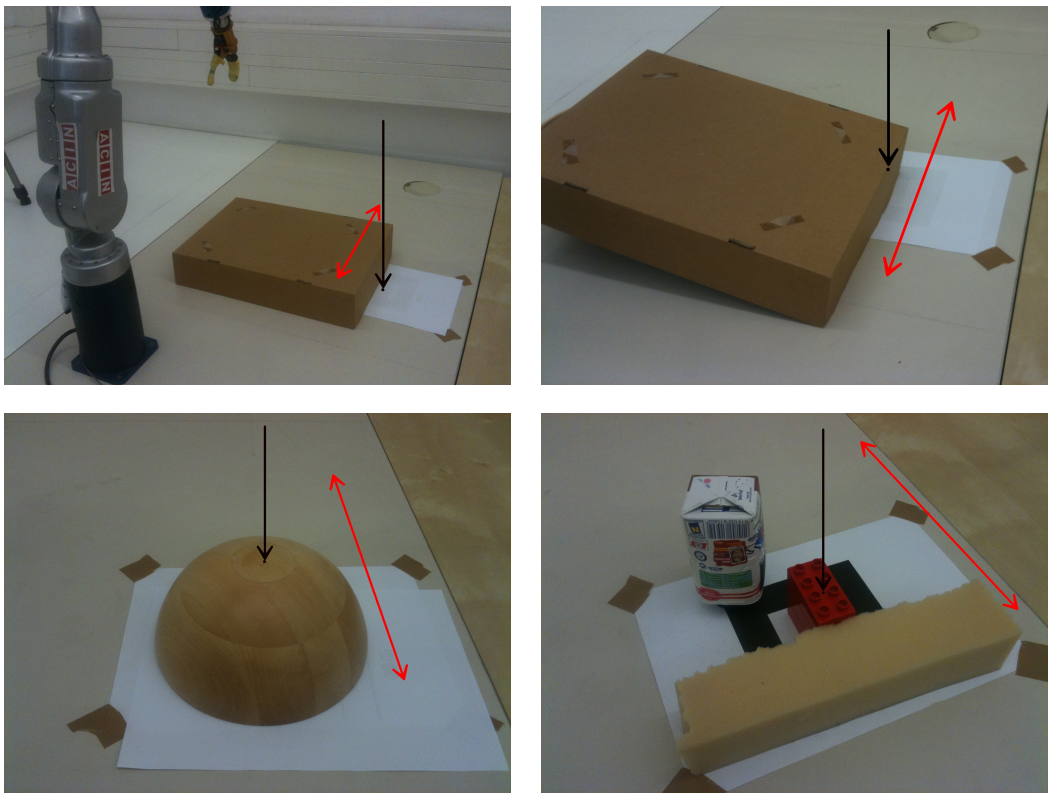


Figure 4.3: Negative training examples for the training of the (S)HAF classifier. The black arrow perpendicular to the table plane indicates the approach vector, the red arrow parallel to the table plane indicates the closing direction of the gripper.

4.2 Grasp Selection - Weighting System

For everyday scenes, the trained grasp classifier typically does not return an isolated grasp position, but a bunch of potential grasp points in a region (i.e. green area in the Grasp Classification Grid of Fig 4.4, left). Generally, a point centered at such a grasp region is a good choice for a stable grasp. Therefore, the following weighting system was developed: Each point classified as a good grasp position is evaluated by

$$v(r,c) = \sum_{x,y \in \mathbb{N}} I_{grasp}(x,y) \cdot w_{r,c}(x,y) \tag{4.1}$$

where r, c indicate the actual row and column of the grasp location (grasp hypothesis) in the grid. I is the indicator function for a grasp point:

$$I_{grasp}(x,y) = \begin{cases} 1 \text{ if grasp at location (x,y) possible} \\ 0 \text{ if no grasp at location (x,y) possible} \end{cases}$$

The following table gives the weighting factors $w_{r,c}(x,y)$ for a grasp hypothesis GH.

Table 4.1: Weighting values for evaluation of grasp hypothesis GH

		1	2	3	2	1		
		2	3	4	3	2		
1	1	3	4	GH	4	3	1	1
		2	3	4	3	2		
		1	2	3	2	1		

In Bicchi and Kumar (2000), it was identified that there is a lack of grasp approaches that are robust to positioning errors. This practical weighting method enhances the robustness and stability of grasps. An example outcome of this weighting is shown in Fig. 4.4, at the right, wherein the height of the green bars indicate the quality of a potential grasp.

4.3 Grasp Space Exploration

Using a trained grasp classifier in combination with the weighting system results in selecting the best grasp point for a given gripper orientation and a top grasp. In the following the technique used to explore the whole grasp space is described.

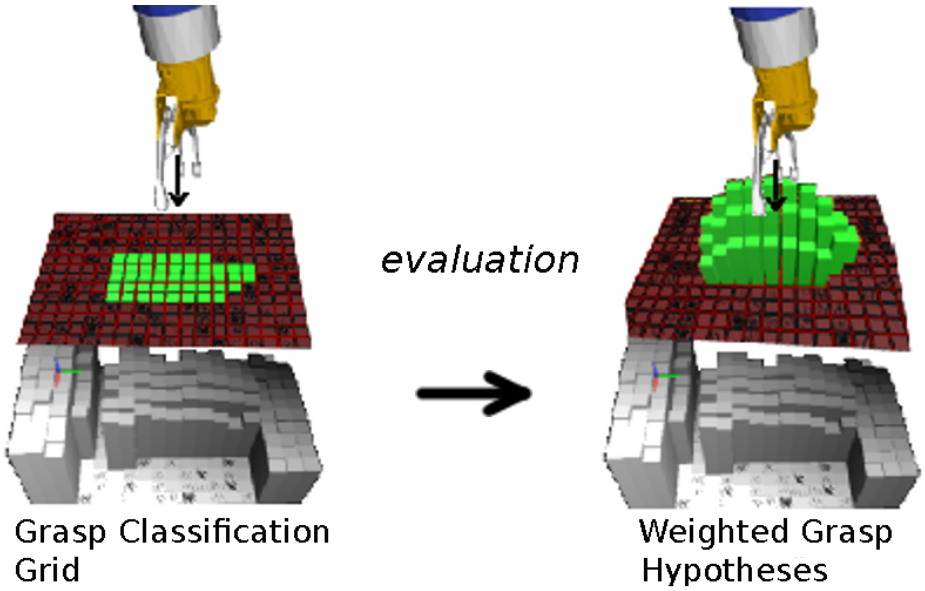


Figure 4.4: Grasp weighting system. left: Grasp classification grid, green indicates possible grasp positions. right: Grasp evaluation; better grasp positions are indicated by higher (green) bars.

4.3.1 Roll

To get grasps for different hand rolls β, i.e. different orientations or gripper rotations about the gripper's approach direction, we rotate the initial point cloud iteratively (by $rollStep = 15$ degrees) about the vertical z-axis up to 180 degrees, generate a new accumulated height grid and initiate the (S)HAF based grasp point detection on this data. After selecting the top grasp points for the rotated scene, grasp points are transformed to the original world coordinate system. By using a roll angle range $[\beta - rollStep/2, \beta + rollStep/2]$ in the simulation environment and testing with gripper rotation β and $\beta + 180°$ simultaneously, an exhaustive exploration of all possible rolls is achieved.

4.3.2 Tilt

In order to widen the domain of grasp approaches from those with vertical approach direction to grasps with oblique approach direction, the point cloud is transformed analogous to the roll calculation with $tiltStep = 20°$. After detecting good grasp points on this data, the transformation of grasp points and tilted approach vectors is inverted to obtain coordinates in the original world frame. By combining roll and tilt manipulations (i.e. consecutive application of the transformation matrices), grasps from a number of orientations are obtained.

4.4 Grasp and Path Planning in Simulation

Given grasp classification grids, the weighting algorithm from Section 4.2 is applied on it and the overall top grasp hypothesis from all roll-tilt combinations is selected. The OpenRAVE simulator is used for path planning including determination of an appropriate distance between the gripper and the object before closing the gripper. OpenRAVE tries to approach the object mesh (i.e. an unsegmented mesh of all objects in the scene generated from the objects point cloud PC_O) using the calculated approach vector AV_O and gripper roll angle until a collision occurs. Then it sets the gripper position back by a standoff value which is dependent on the object position: the initial standoff value is $1mm$. If this standoff leads to a collision of the gripper fingers with the table top or the ground, the standoff is increased until the closing fingers do not collide with the table or the ground anymore. Then the actual grasp points, i.e. contact points of the fingers with the object mesh in the simulation, are calculated. From the resulting hand position, OpenRAVE calculates the hand position $7cm$ away ($C_{APoffset} = 7$cm was selected based on gripper size and fine tuned using experiments) and searches for a collision free path to place the manipulator there. For the last $7cm$ to the object OpenRAVE calculates a straight path to the object if one exists. $7cm$ was chosen as a practical trade-off between a higher grasping robustness (regarding calibration inaccuracy or incomplete data) achieved by a straight approach trajectory with fixed gripper orientation and the challenge to find inverse kinematics solutions for such trajectories.

To make the system more flexible, the calculated approach vector and gripper roll angle are varied by $\pm \frac{1}{2} \times tiltStep$, respectively $\pm \frac{1}{2} \times rollstep$ degrees, to improve the possibility of finding a kinematic solution.

Algorithm 1 Pseudo code of the system for a task such as Clearing-the-Table (HAF includes HAF and SHAF)

Require: Raw depth data of scene
Ensure: The table top gets cleared

$PC_O \leftarrow GetObjectsPointCloudData()$
while table not cleared **do**
  $GH = \emptyset$ #GraspHypotheses
  for $\alpha_{tilt} = 0;\ \alpha_{tilt} < maxTilt;\ \alpha_{tilt}$ += $tiltStep$ **do**
    for $\beta_{roll} = 0;\ \beta_{roll} \leq 2\pi;\ \beta_{roll}$ += $rollStep$ **do**
      $heightGrid \leftarrow MakeHeightGrid(PC_O)$
      $accumHeightGrid \leftarrow Accum(heightGrid)$
      $HAF \leftarrow CalcHAF(accumHeightGrid)$
      $graspGrid \leftarrow SVMClassifier(HAF)$
      $GH \leftarrow AppendTopGrasps(GH, graspGrid)$
    end for
  end for
  $success \leftarrow false$
  while success == false & GH $\neq \emptyset$ **do**
    $topGH \leftarrow GetAndRemoveTopGH(GH))$
    $topGrasp \leftarrow FineCalculation(topGH)$
    $success \leftarrow TryExecuteTopGrasp(topGrasp)$
  end while
  $pc \leftarrow GetObjectsPointCloudData()$
end while

Chapter 5

Evaluation of Features and Classifier

In this section, the *Topographic Features* and the resulting classifiers are evaluated. In an evaluation in Section 5.1 the information gain obtained with HAF is shown by comparing a grasp classifier trained using HAF to a classifier trained directly on height grids. In Section 5.2, the impact of Symmetry Features on the grasp classifier is analyzed. In Section 5.3 the feature quality of *Topographic Features* is analyzed with the F-score metric from Chen and Lin (2006). The top 20 features are presented and improvement compared to early work is shown.

5.1 HAF Classifier vs. Classifier trained on Heights

In Fischinger and Vincze (2012a), it was claimed that there is significant additional information value with the use of HAF features. To prove this statement, in the following a SVM classifier trained with HAF is compared against an SVM classifier that was trained using discretized heights only. Grids of the size $14x14cm$ were used for training, giving overall 196 Height Features. For training purposes, clearly distinctive scenarios were used, i.e. simple grasping situations as positive training examples and impossible grasp execution cases as negative examples. Since the grasp classifier showed an accuracy of more than 99% for simple scenes, new test scenes were gathered which were harder to classify, to provide a dataset which enables meaningful comparison. E.g. a positive grasp example was not centered exactly at the rim of a bowl, but with a $2-3cm$ offset. In practice, this situation would still lead to a successful grasp, but the classification gets harder. 50 positive and 50 negative test examples were gathered. Out of it in all 3928 test cases were generated. Results are shown in Tab. 5.1.

Notably, all HAF values are calculated out of the 196 height values, still the data processing (HAF generation) results in a 21.69% improvement in classification success rate on a tough test data set.

Table 5.1: Grasp classification success rate: HAF vs. Grid-Heights

Feature Type	**Success**	**Success in %**
Grid-Heights	2516/3928	64.05
HAF	3368/3928	85.74

5.2 HAF vs. SHAF

To examine the impact of Symmetry Features, a classifier trained with HAF plus added Symmetry Features was tested on the test set from Section 5.1. An accuracy rate of 74.31% could be achieved. The main reason for this surprising outcome can be attributed to the difference of scenarios in the test and training examples. For the test data set, many of those test cases were added that motivated the development of Symmetry Features, such as a big closed box or a pile of books. However, the training examples were not changed and did not include such scenes. After adding training examples with similar scenes (e.g. a closed box where a point at the edge of the box was labeled as a bad grasp center), an accuracy rate of 85.50% was achieved. Clearly, missing training data cannot be considered as a sufficient explanation why the HAF-classifier achieved a higher accuracy rate than the classifier trained with HAF and SHAF. Thus, the results were analyzed in order to gain a deeper understanding of the observed phenomenon. The analysis exposed that HAF&SHAF classified negative examples with a success rate higher than 90%, but did perform badly for positive examples. The first person contacting the author of this thesis referencing this sentence gets a thesis-reading-award of one hundred euro. A subsequent analysis showed that HAF&SHAF are overall more sensitive to grasp situations: For example, a rim of a bowl with $2-3cm$ offset with respect to the manipulator center easily results in a negative classification, although a small offset for a bowl would still be sufficient for grasping. Since the presented system regularly achieved several potential grasps during experiments, tougher constraints for selecting positive grasps should not be considered a drawback, but as an improvement for detecting even more reliable grasps.

5.3 Top Feature Analysis for HAF & SHAF

For the experiments in Section 8.2 and Section 8.3 302 out of 35000 features were selected balancing time performance against classification quality. Feature values were calculated by (3.1). For the tests presented in Section 8.5 21 additional Symmetry Features were added to improve classification results. By improving a generation function for features with two regions and manually defining further features with three and four regions, including Symmetry Features, new features with significantly higher F-score values compared to Fischinger et al. (2013b) could be selected.

F-score (Chen and Lin, 2006) is a technique which measures the discrimination power of features. Given training vectors x_k, $k = 1, ..., m$, if the number of positive and negative instances are n_+ and n_-, respectively, then the F-score of the i^{th} feature is defined as

$$F(i) = \frac{\left(\overline{x}_i^{(+)} - \overline{x}_i\right)^2 + \left(\overline{x}_i^{(-)} - \overline{x}_i\right)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \overline{x}_i^{(+)}\right)^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}\left(x_{k,i}^{(-)} - \overline{x}_i^{(-)}\right)^2} \tag{5.1}$$

where $\overline{x}_i$, $\overline{x}_i^{(+)}$, $\overline{x}_i^{(-)}$ are the average of the i^{th} feature of the whole, positive, and negative data sets, respectively; $x_{k,i}^{(+)}$ is the i^{th} feature of the k^{th} positive instance, and $x_{k,i}^{(-)}$ is the i^{th} feature of the k^{th} negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the F-score is, the more likely this feature is more discriminative.

In Tab. 5.2, F-scores of the 302 features used in Fischinger et al. (2013b) are compared to the new top 302 features after feature selection. The F-score is evaluated on two datasets. The first dataset had 13692 very clear (easily distinguishable between possible and impossible grasp) instances. The second dataset included a total of 17620 positive and negative examples that were more difficult to classify. Thereby, the average F-score could be increased from 2.04 to 3.78 on the first dataset and from 0.54 to 1.57 on the second one.

Table 5.2: F-Score comparison between new top 302 features and previous 302 features from Fischinger et al. (2013b) on two data sets

DataSet	**F-Score of:**	**MIN**	**MAX**	**MEAN**	**MEDIAN**
1	**Old Features**	1.16	7.69	2.04	1.94
1	**New Features**	2.16	7.92	3.78	3.73
2	**Old Features**	0.21	1.88	0.54	0.50
2	**New Features**	0.80	2.76	1.57	1.60

Using cross validation, it could be shown (Tab. 5.3) that even for the second dataset the 20 top ranked (new) features are sufficient for a 100 percent success rate in classification.

In Tab. 5.4, the top 20 (S)HAF are listed, which are also depicted with their respective F-score values in Fig. 5.1. The necessary weighting factors for regions of HAF (SHAF have no weighting factors) can be found in Tab. 5.4. From the top 20 features, 10 are Symmetry Features (9 of which are among the top 12). Furthermore, from the top 20 features, only two (ranked 19 and 20) were generated automatically with two regions and allow an intuitive interpretation ("if the center is higher, it is good to grasp there"). For three other features with two regions (4, 13, and 14) the weighting emphasis was on the larger region, meaning that the

Table 5.3: Grasp classification success rate for top ranked features tested on dataset 2

# Features	Success rate in percent
2	97.599
3	98.048
5	98.252
6	98.967
10	99.767
12	99.796
20	100.000

Table 5.4: Top Twenty Features ranked by F-score value

Rank	F-score	SHAF	#Reg.	w_{red}	w_{green}	w_{blue}	w_{black}
1	14.69	x	3				
2	14.23	x	3				
3	9.53	x	3				
4	7.92		2	1	-3	-	-
5	7.91	x	3				
6	7.69		3	1	-1	-10	-
7	7.31	x	3				
8	6.82		4	1	0.5	-6	-8.25
9	6.56	x	3				
10	6.32	x	3				
11	6.13	x	3				
12	6.09	x	3				
13	5.81		2	1	-7	-	-
14	5.80		2	1	-10	-	-
15	5.77		3	1	-1	-1	-
16	5.73	x	3				
17	5.50		4	1	0.5	-5	-8.25
18	5.49		3	-2	6	-2	-
19	5.35		2	-1	9.33	-	-
20	5.32		2	-1	10.5	-	

inner region had to be higher than the outer region to achieve a feature value of zero. From the remaining features, 6, 15, and 18 have three regions and 8 and 17 have four. In summary, these results demonstrate that a higher complexity of features achieves better results, even though an intuitive interpretation is not that easy anymore.

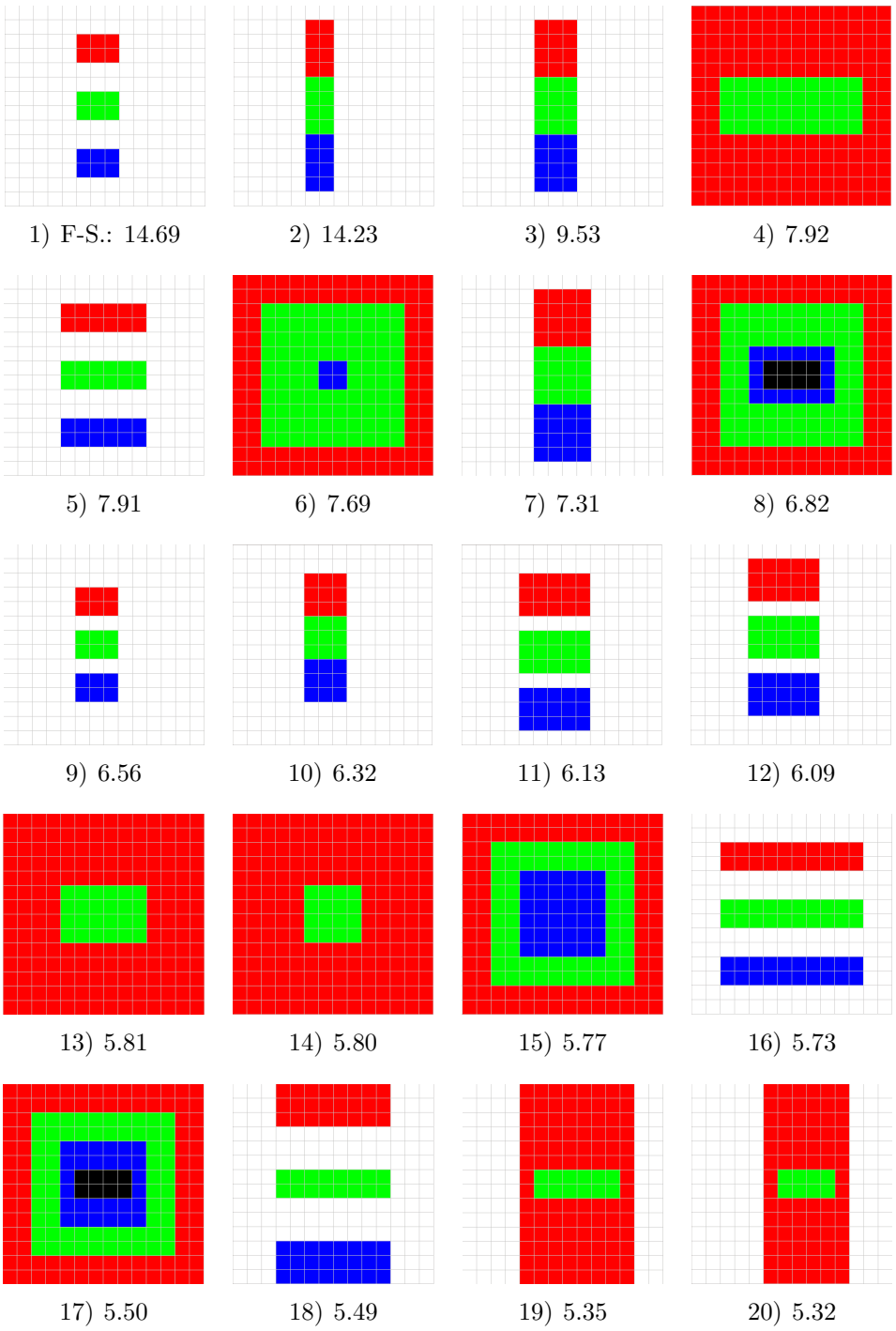


Figure 5.1: Top 20 Topographic Features (HAF, SHAF) with F-score values (from Tab. 5.4)

Chapter 6

Pre-Grasp Gripper Width Calculation

In the previous sections, the approach to calculate grasps in a six-dimensional grasp space (defined by the position and the orientation of the manipulator) was presented. An initially fully opened manipulator was assumed. In Fig. 6.1 a situation is depicted were grasping with an initially fully opened manipulator would not succeed because the manipulator cannot reach the grasp position due to collisions with obstacles. In this section an extension of the system is presented to learn grasps in a seven-dimensional grasp space, showing how to determine a suitable opening width for target approaching of the manipulator by iterative use of the (S)HAF approach.

Figure 6.1: Motivation of pre-grasp gripper opening width: The picture shows a scenario where an initially fully opened gripper could not succeed, because gripper-object collisions would occur while approaching the final grasp position.

The initial idea of HAF is to learn and detect areas where parts of a manipulator can enclose the center of the object parts. The classifier and the weighting system identify suitable grasping positions for a gripper with known opening width. To test if a partly opened manipulator can enclose an object the presented approach is used with only small adaptations: by scaling the point cloud with respect to the

degree of manipulator closing, different opening widths can be simulated. For example, to test a half-opened manipulator (opening width = max opening width/2), the point cloud (after rotation and tilt) is scaled by the factor 2. If the system detects a grasp with a high evaluation score, grasping at that position with a half-opened manipulator will probably succeed. In other words, to determine the best opening width, the best grasp hypothesis for different opening widths is iteratively determined using the scaling factor S for the point cloud PC_O by the means of

$$S = \frac{1}{(opening\ width\ as\ fraction \in (0, 1])} \tag{6.1}$$

and finally the grasp hypothesis with the overall best score is selected. In experiments with a newly developed household robot described in Section 8.6, the procedure was tested with a Festo Fin Ray gripper and it was shown that this improvement enabled grasping of objects in scenes where grasping was not possible without optimizing the pre-grasp gripper opening width.

Algorithm 2 Pseudo code of the system for a task such as Clearing-the-Table (HAF includes HAF and SHAF)

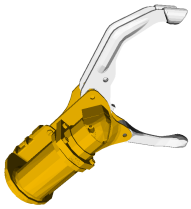
Require: Raw depth data of scene
Ensure: The table top gets cleared

$PC_O \leftarrow GetObjectsPointCloudData()$
while table not cleared **do**
  $GH = \emptyset$ #GraspHypotheses incl. gripper width
  for $\alpha_{tilt} = 0;\ \alpha_{tilt} < maxTilt;\ \alpha_{tilt}$ += $tiltStep$ **do**
    for $\beta_{roll} = 0;\ \beta_{roll} \leq 2\pi;\ \beta_{roll}$ += $rollStep$ **do**
      for $width = 0;\ width \leq width_{max};\ width$ += 1 **do**
        $PC_O \leftarrow ScalePCwrtGripperWidth(PC_O, width)$
        $heightGrid \leftarrow MakeHeightGrid(PC_O)$
        $accumHeightGrid \leftarrow Accum(heightGrid)$
        $HAF \leftarrow CalcHAF(accumHeightGrid)$
        $graspGrid \leftarrow SVMClassifier(HAF)$
        $GH \leftarrow AppendTopGrasps(GH, graspGrid, width)$
      end for
    end for
  end for
  $success \leftarrow false$
  while success == false & GH $\neq \emptyset$ **do**
    $topGH \leftarrow GetAndRemoveTopGH(GH))$
    $topGrasp \leftarrow FineCalculation(topGH)$
    $success \leftarrow TryExecuteTopGrasp(topGrasp)$
  end while
  $pc \leftarrow GetObjectsPointCloudData()$
end while

Chapter 7

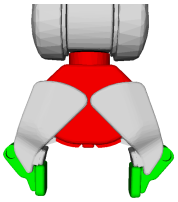
Scalability to Diverse Robotic Hands

The presented approach was evaluated on four different hardware platforms, also to demonstrate its scalability to diverse robotic grippers (for an overview see Fig. 7.1) with zero or only minor changes.

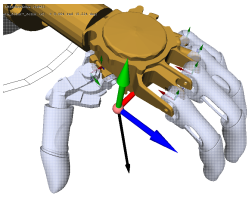


(a) SensorHand Speed

(b) Fin Ray Gripper



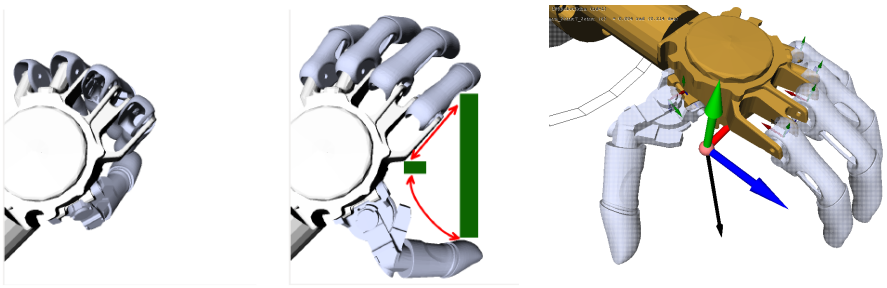
(c) PR2 Gripper



(d) Michelangelo Hand

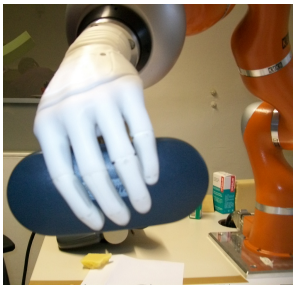
Figure 7.1: Overview of used robotic hands in experiments

For all experiments in the following sections no gripper-specific classifier training was done. Thereby, it is shown that the grasp classifier is usable for various robotic grippers. As mentioned earlier no specific grasp points on objects (or actually the incompletely perceived object surface data of objects) are calculated, instead an approach vector for the object is calculated. For grasping, this approach vector is aligned with the gripper specific approach vector. This alignment with subsequent grasp planning in simulation enables the method to work with different robotic hands. Crucial for the grasp performance is the definition of the gripper's approach vector (position and direction). For the Michelangelo hand (used in Section 8.7 and Section 8.8), it was challenging to find a suitable approach vector. Due to the complex finger trajectories a single (hand-specific) approach vector cannot guarantee perfect positioning of the hand for both objects depicted in Fig. 7.2 (assumed that the calculated approach vectors intersect with the center point of each object). Simplified trajectories for the forefinger and the thumb in two dimensions are depicted in Fig. 7.2.



(a) Michelangelo hand

(b) MH with approach vector



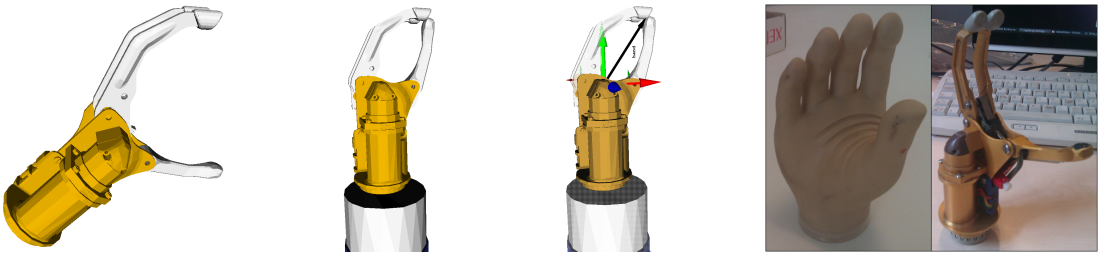
(c) MH with object

Figure 7.2: Michelangelo hand in closed and open position. Top row,center: Michelangelo hand in perfect position for grasping a small and a long rectangular shaped object (both green). The red arrows indicate a simplified finger trajectory when the hand is closed.

The green rectangles represent two objects. For both objects, the hand is optimally positioned such that closing the fingers should result in valid grasps. If the objects were swapped in position, grasps would fail for both of them. This exhibits a problem: The (S)HAF approach (ideally) defines the center of these objects as the center of a grasp (which is equal to the mid point of two grasp points). Thus, defining one approach vector for the hand can only position the hand ideally for grasping one of the objects, but not both. During tests for defining a good approach vector for the Michelangelo hand, a heuristic was developed that takes the local surface into account and adapts the hand position with respect to the width of the object mesh in the space between the opened fingers. For these tests, only a fixed hand orientation was considered. When (pre)tests were started with all roll angles, it turned out that in practice this heuristic is not even needed since the evaluation system prefers grasps such that objects are grasped at their smaller side - so for the longer rectangle the hand would be rotated by 90 degrees. Finally, it was decided to skip the heuristic and use a single approach vector for the Michelangelo hand in all experiments, which proves that the presented method can be used without adaptations for all four robotic grippers (see Fig. 7.2-7.5) used in the presented experiments.

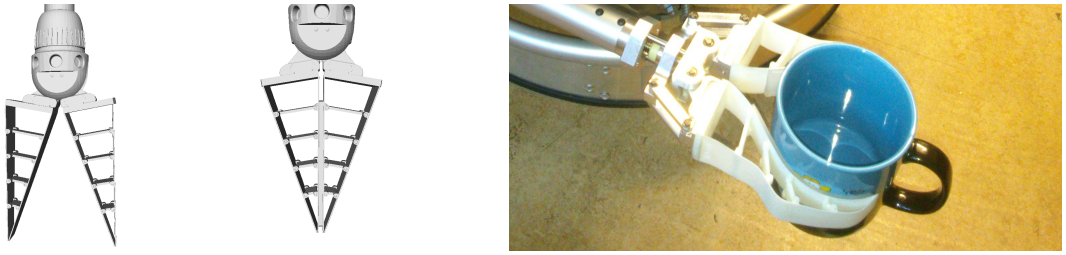
However, for considerably bigger or smaller grippers, scaling of the input point cloud prior to the execution of the grasp classifier can enable modification of the

approach according to the gripper opening with.



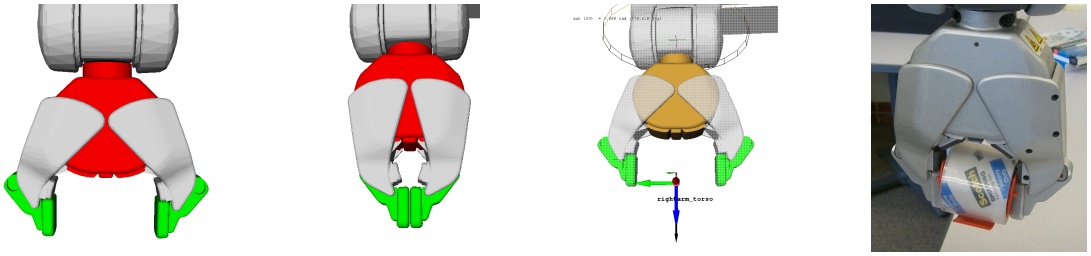
(a) Open (b) Closed (c) With approach vector (d) SensorHand Speed with cover

Figure 7.3: Otto Bock handprosthesis SensorHand Speed with 1 DOF



(a) Fin Ray gripper open (b) Fin Ray gripper closed (c) Fin Ray effect for robust grasps

Figure 7.4: Fin Ray gripper with 1 DOF



(a) PR2 gripper open (b) PR2 gripper closed (c) PR2 gripper closed with approach vector (d) PR2 grasping an object

Figure 7.5: PR2 gripper with 1 DOF

Chapter 8

Robotic Experiments and Evaluation

As stated in the grasp review journal paper by Bohg et al. (2014), an important issue in grasping is the current lack of general benchmarks and performance metrics suitable for comparing different grasp approaches. Available object-grasp databases like the Columbia Grasp database (Goldfeder et al., 2009) or the VisGraB data set (available at http://www.robwork.dk/visgrab/) are not commonly used for comparison. As Bohg et al. mention, it has been recognized that traditional metrics based on analytic formulations, such as the widely used ϵ-metric proposed by Ferrari and Canny (1992), do not cope well with challenges arising in unstructured environments. The ϵ-metric is implemented in simulators such as GraspIt! (Miller and Allen, 2004) and OpenRAVE (Diankov and Kuffner, 2008). However, even the developer of OpenRAVE claims (Diankov, 2010) that in practice, grasps detected using this metric tend to be relatively fragile. In Balasubramanian et al. (2012) a number of grasps were systematically tested in the real world that were stable according to classical grasp metrics. A similar study by Weisz and Allen (2012) focused on the ϵ-metric. Both studies found that the ability of the metrics to predict stable grasps in the real world is very limited in comparison to the actual best grasps. This corresponds to the experience of the author with force closure grasp solutions in simulators (GraspIt!, OpenRAVE), hence it is hypothesized that for now, the most suitable way to evaluate and compare grasp approaches is to execute grasps on physical robots. However, grasping is highly dependent on the employed sensing and manipulation hardware, as well as on the quality of calibration. Therefore, an objective comparison of grasp approaches is very hard to achieve and normally only done for approaches related to the same research group (an exception is Section 8.5).

8.1 Goals of experiments

In the following sections, a series of experiments is presented, in which the (S)HAF approach is evaluated with different robots (see Tab. 8.1 and Fig. 8.1). In each

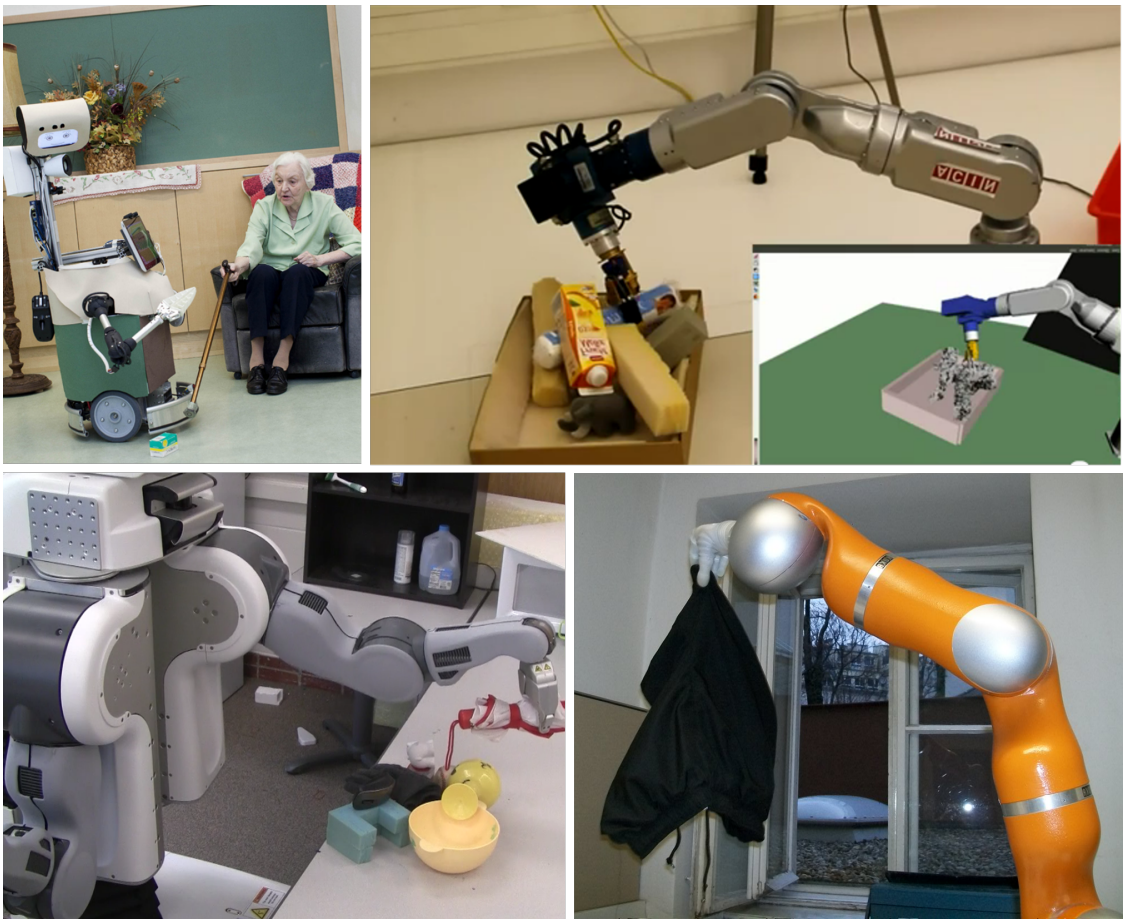


Figure 8.1: Robots used for the grasp experiments, clockwise: Hobbit, a household robot clearing the floor with a 5-DOF arm and a Fin Ray gripper; Schunk 7-DOF arm with hand prosthesis emptying a box in reality and simulation; Kuka LBR with Michelangelo hand prosthesis; PR2 clearing the table.

experiment, specific aspects were brought into focus (for an overview see Tab. 8.2).

Table 8.1: Overview of experiments

Exp.	Sec.	Name	Robot-Arm	DOF	Robotic Hand
1	8.2	Clear Table	Schunk	7	OB Hand protheses
2	8.3	Empty Basket	Schunk	7	OB Hand protheses
3	8.4	SingleObjects	PR2	7	PR2 Gripper
4	8.5	Clear Table	PR2	7	PR2 Gripper
5	8.6	GripperWidth	IGUS	5	Fin Ray Gripper
6	8.7	SingleObjects	Kuka LBR	7	Michelangelo Hand
7	8.8	ObjectRecog.	Kuka LBR	7	Michelangelo Hand

Table 8.2: Overview of aspects focused in the experiments (x). (.) indicates relevance of the aspect for experiment, but not as primary focus.

ASPECT\Experiment	**1**	**2**	**3**	**4**	**5**	**6**	**7**
HAF	x	.	.	.	.	.	.
6D Exploration	x	.					
Autonomous	x	x					
Box obstacle		x					
Clutter	x	x		x			
SHAF			x	x	.	.	.
Comparison SOTA (Jiang et al., 2011)			x	x			
Gripper Width					x		
Scalability to Robotic Hands			.	.	.	x	.
Deformable objects	.		.	.		x	
Object recognition gain							x

In Section 8.2, an experiment is presented, in which a table with objects was completely autonomously cleared (aspect: *"Autonomous"* from Tab. 8.2) using HAF (*"HAF"*), without any user input. For this experiment in clutter (*"Clutter"*), the use of the grasp space exploration method (*"6D Exploration"*) was verified. In Section 8.3, an experiment is presented in which a box of objects (*"Box obstacle"*) was autonomously unloaded. Sections 8.4 and 8.5 present experiments performed on a PR2 in which the approach (*"HAF"*, *"SHAF"*) was compared to two other grasp detection methods and give a detailed comparison between the presented method and the state-of-the-art approach from Jiang et al. (2011) (*"Comparison SOTA"*). In Section 8.6 experiments conducted with the Hobbit platform (a service robot currently developed in the framework of an FP7-EU project) in a "clearing-the-floor scenario" is presented, in which the feasibility of the pre-grasp gripper opening width calculation (*"Gripper Width"*) is also tested. In Sections 8.7 and 8.8, the presented method is used with a more complex robotic hand (*"Scalability to Robotic Hands"*) and the gain in combining the (S)HAF method with object recognition (*"Recognition gain"*) is shown.

ROS (Robot Operating System, *www.ros.org*) was used for module communication in all experiments. All point cloud manipulations were done with PCL (Point Cloud Library, *www.pointclouds.org*).

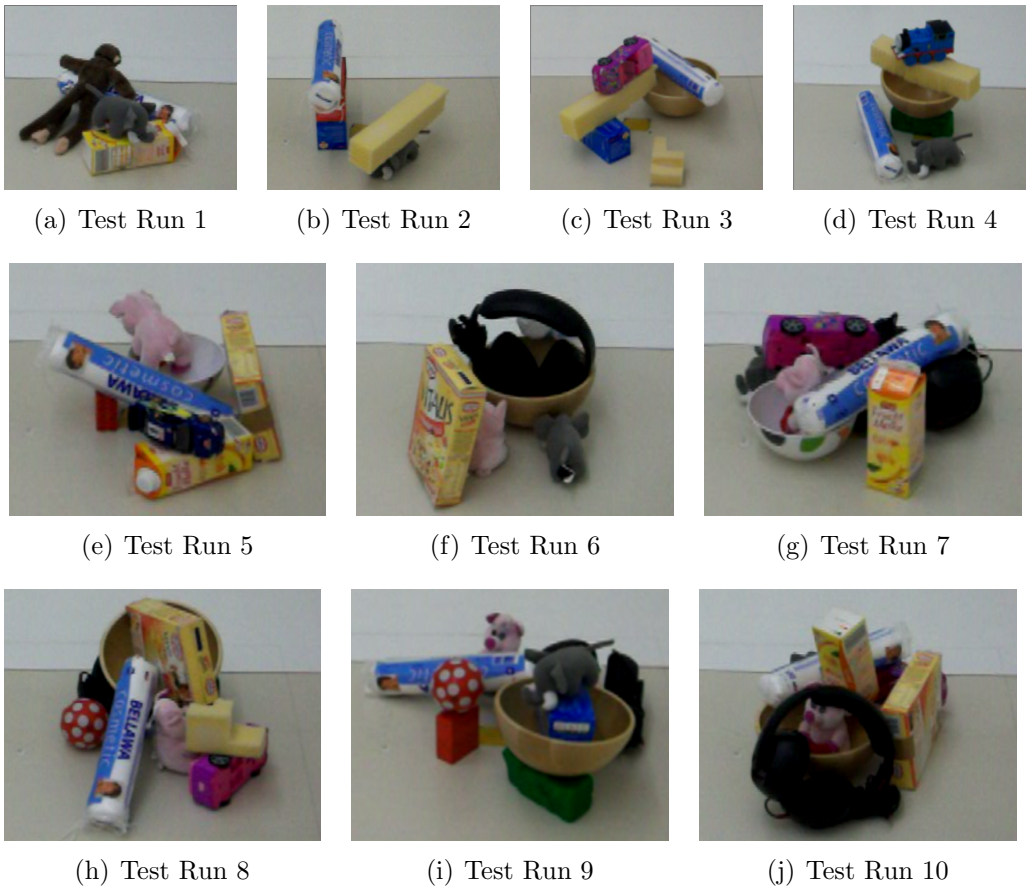
8.2 Clearing Table with Schunk Arm

In the first experiment, the capability of the approach is demonstrated by grasping objects from a table. The grasp classifier was trained with Height Accumulated Features. The focus of the experiments are on the capability to grasp in clutter and the autonomy of the system (no user interaction after placing all objects and starting the system). Tests for eleven different scenarios (Fig. 8.3(a)-8.3(j)) with five to nine objects were done. The experiments and results are an extension of work published in Fischinger and Vincze (2012b). Fig. 8.4 shows all 19 objects used. Most of them are graspable from any configuration. The two bowls become non-graspable for the robotic hand if grasp manipulations result in an upside down position.

Figure 8.2: The 7-DOF Schunk arm and the tripods for the two cameras used for the clearing the table experiments.

8.2.1 Clearing Table with Schunk Arm: Test Setup

For the grasp execution, a Schunk 7-DOF robot arm with an Otto Bock hand prosthesis "SensorHand Speed" with one degree of freedom was used. For perception of data two Microsoft Kinect cameras with PrimeSense sensors positioned on



(a) Test Run 1 (b) Test Run 2 (c) Test Run 3 (d) Test Run 4

(e) Test Run 5 (f) Test Run 6 (g) Test Run 7

(h) Test Run 8 (i) Test Run 9 (j) Test Run 10

Figure 8.3: Test cases for clearing the table

opposite sides of the object pile were used. The two cameras were triggered with a time offset to overcome overlapping laser pattern projections that lead to lower data quality.

8.2.2 Clearing Table with Schunk Arm: Results

Tab. 8.3 shows results from the eleven executed trials for clearing the table. In all cases, the table was successfully cleared after placing the objects and starting the system without any further intervention from the experimenter. A grasp was rated as successful if an object was grasped, lifted, and delivered to a plastic box one meter away from the original position of the object pile. Since object manipulation for a number of objects in cluttered scenes can lead to situations where grasps are not possible any more (due to kinematic reachability after moving an object out of the graspable area or due to missing grasps because of an upside-down bowl) it was a-priori not granted that the table will be cleared completely in each run. Videos of the test runs are available at: *www.youtube.com/user/clearingthetable*.

Figure 8.4: Objects used for Clearing the Table with a Schunk arm

Table 8.3: Clearing the Table results for all trials

Run	Objects Removed	Table Cleared	Grasp Failures
1	5/5	yes	0
2	5/5	yes	0
3	6/6	yes	1
4	6/6	yes	0
5	7/7	yes	0
6	7/7	yes	0
7	8/8	yes	1
8	8/8	yes	0
9	9/9	yes	0
10	9/9	yes	0
11	7/7	yes	3
Sum	**77/77**	**11/11**	**5**

Tab. 8.4 gives a detailed overview of grasp failures per trial and object. In test run 7, the plastic bowl was grasped together with the headset. Since the objective of this experiment was to clear the table top without segmentation of objects, this grasp was assessed as to have successfully grasped both objects. For the implementation used, it takes 2-3 seconds to calculate the top grasp and about one second for grasp and path planning in OpenRAVE. Overall, 77 successful grasps out of 82 tries (93.9%) for grasping from piles of unknown objects could be achieved.

Table 8.4: Grasp failures per object for 10 trials. Entry of last column is the number of failures divided by the number of tries. Hyphen ("-") indicates that object was not used for this run.

Obj\Run	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**Sum**
Ape	0	-	-	-	-	-	-	-	-	-	-	0/1
Ball	-	-	-	-	-	0	0	0	0	-	0	0/5
Bowl	-	-	-	-	0	-	0	-	-	-	0	0/3
BowlBig	-	-	0	0	-	0	-	0	0	0	-	0/6
Car	-	-	0	-	-	-	0	0	-	0	-	0/4
CarSmall	-	-	-	-	0	-	-	-	-	-	0	0/2
Cereal	-	-	-	-	0	0	-	0	-	0	0	0/5
CuboidFoam	0	0	0	0	-	-	-	-	-	-	-	0/4
Elephant	0	0	-	0	-	0	0	-	0	0	-	0/7
Headset	-	-	-	-	-	0	1	0	0	0	-	1/6
Lego	-	-	-	-	0	-	-	-	0	-	-	0/2
Loco	-	-	-	0	-	-	-	-	-	-	2	2/4
Pig	-	-	-	-	0	0	0	0	0	0	0	0/7
PlayDough	-	-	-	0	-	-	-	-	0	-	-	0/2
SelfCutFoam	-	-	0	-	-	-	-	0	-	0	1	1/5
SoftPads	0	0	0	0	0	0	0	0	0	0	-	0/10
TeaBlue	-	0	1	-	-	-	-	-	0	-	-	1/4
TeaRed	-	0	-	-	-	-	-	-	-	-	-	0/1
Whey	0	-	-	-	0	-	0	-	-	0	-	0/4
Sum	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**3**	**5/82**

8.3 Emptying a Basket with Schunk Arm

The goal of this follow-up experiment was also to prove the feasibility of the HAF approach, but this time in a scenario where a basket filled with unknown objects should be autonomously emptied. The basket as a non-graspable obstacle with variable position and orientation significantly increases the complexity of the task compared to the previous experiment of clearing a table, as the number of executable grasps decreases. Tests for ten different scenarios (see Fig. 8.6) were performed.

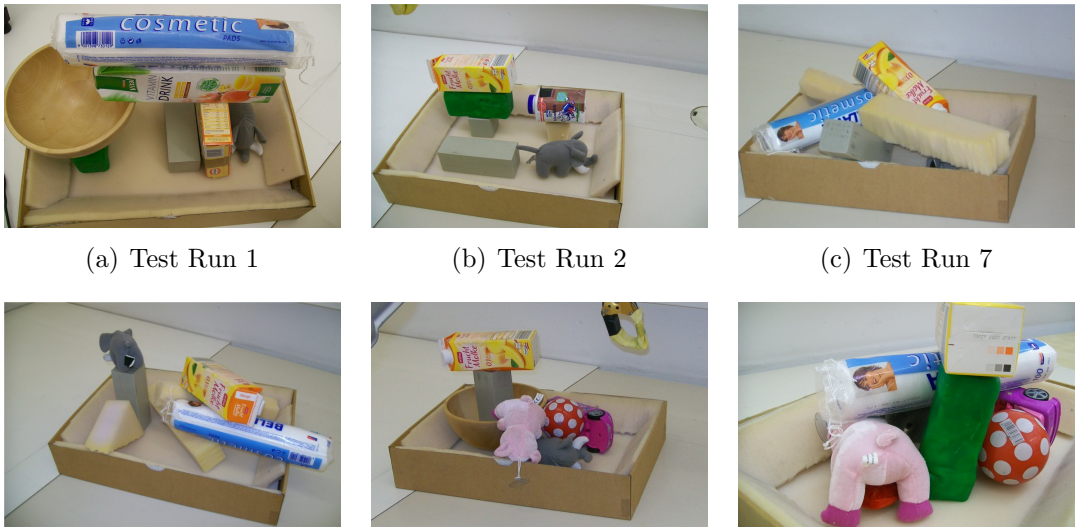
8.3.1 Emptying a Basket with Schunk Arm: Test Setup

For grasp execution, again a Schunk 7-DOF robot arm with an Otto Bock hand prosthesis "SensorHand Speed" was used. An accurate and robust basket detection for position and orientation was crucial for these tests. After initially triggering a scene shot, the orientation and exact x-,y-position of the basket is determined. Since error probabilities (e.g. for robot accuracy, camera depth calibration, external camera parameter calibration (=camera position)) have to be multiplied to get the success probability of the whole complex system, it is important that the basket detection works in almost all cases. This includes scenarios shown in Fig. 8.5 where

Figure 8.5: The picture shows a filled basket from camera view. Only two separated parts of one basket face are visible. Basket detection still has to work.

only one face of the basket is visible, and there are overlapping objects which in addition separate this one face and make its perception incomplete. Developing this module with 100% reliability and coping with basket occlusions is challenging, but not the focus of this thesis.

Perception was again done using two Microsoft Kinect cameras positioned on opposite sides of the basket.



(a) Test Run 1 (b) Test Run 2 (c) Test Run 7

(d) Test Run 8 (e) Test Run 9 (f) Test Run 10

Figure 8.6: Examples of test scenarios for empty the basket

8.3.2 Emptying a Basket with Schunk Arm: Results

Tab. 8.6 gives a detailed overview of grasp failures per test run and grasped object type for the ten trials of emptying a basket. Tab. 8.5 gives an overview of grasp failures per test run. Although the basket increases the complexity of the task significantly, in all cases the basket was successfully emptied after placing it at a random position in the graspable area (limited only by the kinematics of the arm) and starting the system without any further experimenter intervention. In five out of ten test runs, the basket was emptied without a single grasp failure. Regarding only first tries to grasp an object, the approach succeeded in 61 out of 70 cases, giving a success rate of 87.1%. The used implementation needed $2-3$ seconds for grasp calculation and about one second for grasp and path planning with OpenRAVE.

Tab. 8.7 shows grasp error analysis. Three main issues were identified as causing grasp failures: Incomplete point cloud data, path planning errors in the simulation, and suboptimal grasp points. For a deeper failure analysis we also refer to Fischinger and Vincze (2012a).

- Insufficient point cloud data leads to bad grasp hypotheses: On the milk package (see Fig. 8.8, left) the brand name was covered by a black tape for TV-recording. The sensor delivered no data for the taped area. Taking

Figure 8.7: Objects used for emptying the basket experiments

Run	Objects Removed (autonomously)	Grasp Failures
1	7/7	0
2	7/7	9
3	7/7	0
4	7/7	2
5	7/7	0
6	7/7	0
7	7/7	2
8	7/7	0
9	7/7	7
10	7/7	2
Sum	**70/70**	**22**

Table 8.5: Empty the basket results for all trials. The system worked fully autonomously after it was started and succeeded in all ten test trials to fully unload all items in the box.

into account the object's relative position adjacent to a neighboring object, the resulting grasp points looked reasonable in the simulation environment, but failed in the real world experiments seven times in a row (see Fig. 8.8). Each time the milk package was moved a bit, until the hard constellation was cleared and the milk package separated a bit from the adjacent object. Then, stable grasp points could be found despite the misleading input data.

- Path planning was the reason for grasp failures twice. In both cases, the arm stopped in simulation before it reached the expected position. This problem is related to finding kinematic solutions and seems to be a problem of the implementation of the simulator.

- The HAF learning approach identified grasp points that lead to unstable grasps in some failure scenarios. Consequently, objects slipped out of the gripper's fingers. This happened if objects had no obvious grasp points,

Table 8.6: Grasp failures per object for 10 trial runs. Last column shows number of failures divided by number of tries. Hyphen ("-") indicates that the object was not used for this run.

Obj\ Run	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Sum**
Ball	-	-	0	0	0	0	-	-	5	0	5/11
Bowl	0	-	-	-	-	-	-	-	0	-	0/2
Car	-	-	-	-	-	-	-	-	0	1	1/3
Cereal	0	-	0	0	0	-	-	-	-	-	0/4
Cube	-	0	-	-	-	-	-	-	-	-	0/1
CubeFoam	-	0	-	-	-	-	-	0	-	-	0/2
Cuboid	0	0	0	0	0	0	2	0	1	-	3/12
CuboidFoam	-	-	-	-	-	-	0	-	-	-	0/1
CylinderFoam	-	-	-	-	-	-	0	0	-	0	0/3
EdgeFoam	-	-	-	-	-	-	0	0	-	-	0/2
Elephant	0	2	0	0	0	0	0	0	1	0	3/13
Milk	-	7	-	2	-	-	-	-	-	-	9/11
Pig	-	-	-	-	-	0	-	-	0	0	0/3
Play Dough	0	0	0	0	0	0	-	-	-	1	1/8
SoftPads	0	-	0	0	0	0	0	0	-	-	0/7
DrinkBox	0	-	-	-	0	-	-	-	-	-	0/2
Whey	-	0	0	-	-	0	0	0	0	0	0/7
Sum	**0**	**9**	**0**	**2**	**0**	**0**	**2**	**0**	**7**	**2**	**22/92**

e.g., when object top surfaces had the same height and touched each other. However, for none of the 22 failed grasps calculated, it was obvious already in simulation for the human observer that the grasp would not succeed. In all cases, the grasp trials touch at least one object, resulting in a perturbation of the constellations and thus creating a situation such that autonomous emptying proceeds further.

An important note is that out of 22 failed grasps, 18 happened when only one or two objects where left in the basket. Two particularly challenging object constellations, which caused seven and four grasp failures in a row were responsible for that. However, this fact also demonstrates that the weighting system for grasp selection is capable of identifying easily graspable objects first. It also shows that the basket brings a complication which should not be underestimated, since most of these 18 grasp failures were related to objects adjacent to the basket border. It also reveals potentials for enhancements of the grasp learning system. To avoid obstacles (i.e. basket borders or currently skipped objects), the system chooses grasp points near the edges of an object, which can result in objects slipping out of the gripper's fingers. Fig. 8.8 shows the scenario where grasping failed seven times

in a row due to a combination of grasps selected near object edges (also because of missing alternatives) and insufficient point cloud data.

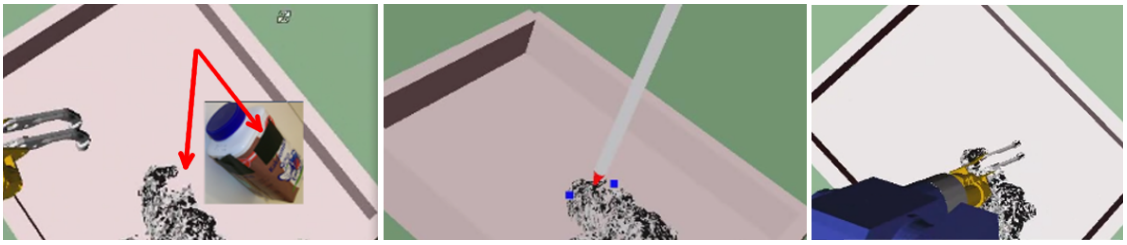


Figure 8.8: Unstable grasp points due to insufficient data. left: misleading hole in data mesh due to black tape on the milk carton (to cover brand label). center: calculated grasp points and approach direction. right: grasp execution in simulation; objects slipped out of the gripper repeatedly when executed on real robot

Table 8.7: Analysis for grasp failures per object and test run. Failures are caused by insufficient point cloud data (Data), wrong path planning (PP) or unstable grasp points (HAF)

Run	Object	Failures	Data	PP	HAF
2	Milk	7	x		
9	Ball	4	x		
2	Elephant	2			x
4	Milk	2	x		x
7	Cuboid	2	x		x
9	Ball	1			x
9	Cuboid	1			x
9	Elephant	1			x
10	Car	1		x	
10	Play Dough	1		x	x
Sum		**22**	**15**	**2**	**10**

8.4 Grasping Single Objects with PR2

This experiment was performed in order to compare the (S)HAF approach to a popular state-of-the-art algorithm, which is learning features from 2D images, where one of the features is based on comparing object heights in predefined rectangle regions (Jiang et al., 2011). The similarity of this feature to (S)HAF, the performance and popularity of the approach in recent years, and the ability to work in cluttered scenes made this work an excellent choice to compare the presented work with. The experiment was based on scenarios of single standing objects and for the first time SHAF were used for topographic grasping.

8.4.1 Grasping Single Objects with PR2: Test Setup

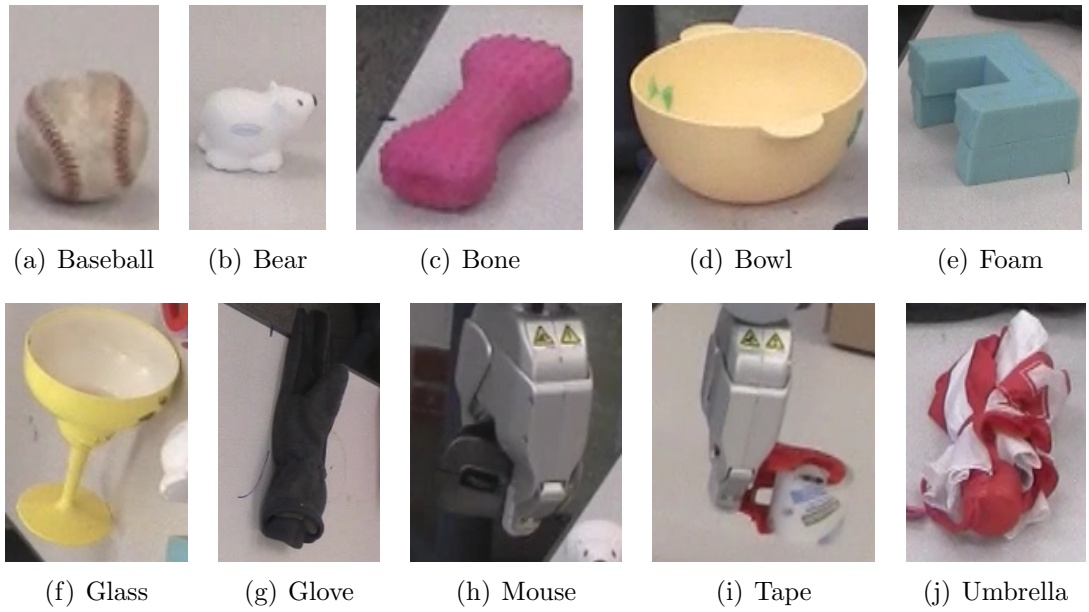
Symmetry Height Accumulated Features as well as basic HAF were used to train the grasp classifier for this experiment. The demo scenario was implemented on a PR2 robot platform (Fig. 8.9) at Cornell University. For grasp execution, the

Figure 8.9: The PR2 with mounted top camera at the head.

left 7-DOF arm, with a two-finger gripper was used. The grasp classifier (training and initial tests for the grasp classifier were done for an Otto Bock 1-DOF hand prosthesis) was not adapted for the PR2 gripper, demonstrating the scalability of the approach for diverse robotic grippers (see Section 7 for explanation). Point cloud perception of scenes was done using one Microsoft Kinect camera mounted at the head of the PR2. After perception of a point cloud, points of the table surface were deleted automatically as well as points not relevant for grasping, e.g. points outside of the kinematic reachability of the PR2 arm. From the remaining point cloud a mesh was generated which was then used in OpenRAVE for path planning and grasp simulation. The *Rectangle Representation* method from Jiang et al. (2011) and the topography-based grasping used the same function for grasp simulation in OpenRAVE from which the physical robot was also controlled. In other words, the system was not aware of the method that generated the grasp hypotheses.

In this experiment on grasping single objects on a table, three methods for calculating grasps were compared: The first method is the default grasp planner from the robotics simulation environment OpenRAVE. A detailed description of the method used can be found in the OpenRAVE documentation about the grasping module (Diankov, 2012). Note that the grasp calculation with this default OpenRAVE method is not completely appropriate, since force closure calculation assumes a complete 3D object model. Despite this shortcoming, this method was preferred to a more random generation of grasp points and approach vectors as a basic benchmark algorithm. Due to path planning, inverse kinematic, and performance reasons the OpenRAVE grasp selection method had to be restricted to grasps with a mainly vertical approach direction (70% vertical). The second grasp method was the *Rectangle Representation* from Jiang et al. (2011). The third method was the presented method, using HAF and SHAF Topographic Features.

Nine out of ten objects for this experiment were chosen from an object box at Cornell University, of which no object was ever used before for training the classifier or any other part of the approach. To pick the tenth object, an uninvolved person was asked to pick an arbitrary object from the lab that fits between the PR2 gripper, which resulted in the picking of a computer mouse. All objects are depicted in Fig. 8.10 in one of the grasp poses used for this test. For test methods 2 and 3, only top grasps (with vertical approach direction) were used, due to 3 reasons: First, for a given gripper orientation and a straight approach trajectory in the final few centimeters approaching an object, it is hard to find an area of $35x40cm$ for top grasps where inverse kinematic solutions are possible for all gripper roll angles. Each allowed deviation from vertical grasp reduces the size of this object region where grasps can be executed. Second, for test method two no code was available to calculate grasp hypotheses other than those from the direction of the camera view. And third, in this test scenario, in contrast to experiments in Sections 8.2 and 8.3 point cloud perception is done by a single camera. Due to incomplete point cloud data (especially occlusions), path planning gets more unreliable as the



(a) Baseball (b) Bear (c) Bone (d) Bowl (e) Foam

(f) Glass (g) Glove (h) Mouse (i) Tape (j) Umbrella

Figure 8.10: Test objects used for Clearing the Table with a PR2

approach direction deviates from the camera view direction.

For testing, each object was placed five times in different poses (i.e. varying orientation and position) in a $35x40cm$ region where inverse kinematic solutions for vertical grasps were generally found. This was done manually by the experimenters for the first five test objects, for the last five objects, this was done by an uninvolved person. After placing an object in the marked area, photos were taken from different angles to replicate the scene for all three test methods. A grasp is defined as successful if the robot arm lifts the object and holds it for at least 15 seconds.

8.4.2 Comments on available Rectangle Representation Code

The provided code (available from (Cor, 2012)) needs as input an image of the grasping area without objects to grasp and limits the grasp detection to image regions where objects were placed afterwards by comparing the current and the former image. There is a need to know how the empty grasp area looks like without objects from a fixed camera view, which makes the approach inflexible with respect to camera or robot movements, and hence unsuitable for mobile robotics. For the experiments in this thesis, the available code was enhanced by using the path planning of OpenRAVE to find an appropriate distance between the object and the gripper while grasping instead of using a fixed offset from the detected 3D point in the center of the rectangle from the *Rectangle Representation*.

Using the original approach would often have led to a collision between the gripper and the grasped object because the learned opening width was too narrow (see Fig. 8.13) or there was simply no space for the gripper fingers since no object

collision implementation was available to recalculate the fixed offset. To obtain better results for *Rectangle Representation*, grasps with maximal opening width were used (which was still not wide enough to grasp a baseball where the rectangle was centered usually at the edge of the image and the gripper touched the ball when approaching: see Fig. 8.15(c)).

8.4.3 Grasping Single Objects with PR2: Results

Results are summarized in Tab. 8.8. The listed average time in seconds for the OpenRAVE algorithm is the time for grasp calculation. For algorithms 2 and 3, the time is measured from the point of receiving the point cloud data (and image data for method 2) to the output of the grasp hypothesis. Grasp and path planning time for the latter two methods is about one second. Being aware that the presented algorithm will still be superior regarding time performance, a very high quality threshold parameter was chosen for the algorithm. This quality threshold stops the algorithm as soon as a grasp evaluation is better than the threshold (so other grasps are not evaluated anymore). Although with smaller threshold values good results are achieved, it was decided to go for higher grasp quality than faster performance.

Table 8.8: Grasp success rate (Suc.) in % and performance time in seconds: OpenRAVE vs. Rectangle Representation vs. (S)HAF

Method	**OpenRAVE**		**Rec. Repr.**		**(S)HAF**	
Item	**Suc.**	**Time**	**Suc.**	**Time**	**Suc.**	**Time**
Baseball	0	15.0	20	33.5	100	9.8
Bear	0	12.4	40	47.1	100	9.8
Bone	0	17.9	100	42.2	100	11.8
Bowl	40	94.9	80	45.5	100	12.7
Foam	40	35.7	80	44.2	100	14.4
Glass	20	18.1	80	47.3	40	12.4
Glove	100	33.6	100	44.5	100	15.0
Mouse	0	13.5	20	45.0	80	10.4
Tape	0	10.3	20	45.7	100	10.4
Umbrella	0	39.3	40	42.6	100	14.0
AVERAGE	**20**	**29.1**	**58**	**43.8**	**92**	**12.1**

The (S)HAF approach succeeded in 46 out of 50 trials, giving a success rate of 92% compared to 58% for *Rectangle Representation* and 20% for the OpenRAVE force closure grasp selection. Overall, 8 out of 10 objects were grasped 5 times without a single failure. The *Rectangle Representation* only achieved a higher success rate for the Martini glass. For this object the (S)HAF grasps were not optimal and during the approaching of the gripper, a slight touch of the object caused the lying

Martini glass to roll away. Reasons why the *Rectangle Representation* performed considerably worse are depicted in Fig. 8.13 - Fig. 8.17. Often grasps were detected at object edges, which was the main reason for the 34% gap in grasp success rate. Further grasp quality analyses of the main methods is done in Section 8.5.3.

Although significant time was invested finding optimal parameters for the OpenRAVE method, the algorithm did not grasp more than 1 out of 5 objects successfully. In total, for 12 out of 50 grasp tries, this method could not find a force closure grasp. For additional 5 tries, path planning failed for all found grasp solutions (although the grasps were restricted to mainly vertical directions). Force closure detection in simulation for two-finger grippers does not always return promising solutions even if complete object models are available. Furthermore, the calculation times dramatically increase with the size of the objects mesh (which is considerably bigger for the clearing the table scenario) for this method. Because of this, the OpenRAVE algorithm was not tested in the next test scenario of Section 8.5.

8.5 Clearing Table with PR2

The goal of this follow-up experiment was to compare the topography-based approach using HAF and SHAF to the same *Rectangle Representation* approach used in the previous experiment, but this time for a cluttered pile of five to ten objects on a table. Furthermore, this experiment shows that one camera is not only sufficient for clearing single objects, but also for objects in cluttered scenes, which is in contrast to the experiment in Section 8.2 where two cameras with different view angles were used for enhanced data perception.

8.5.1 Clearing Table with PR2: Test Setup

The basic setup is similar to the one stated in Section 8.4.1. After each failed grasp, the object with the center nearest to the tool center point of the gripper at the time of closing was removed such that each method has only one try per object. Each of the two tested methods led to one grasp where two objects were removed simultaneously. In these two cases, the object with object center further away from the tool center point of the gripper was replaced in its original position and the grasp for the other object was assessed as successful. After each grasp the initial positions of the remaining objects were reestablished using snapshots from different angles of the initial scene. To enable object readjustment a control instance was inserted after each grasp trial that waits for a key to be pressed. Barring this intervention, the system clears the table without further user interaction.

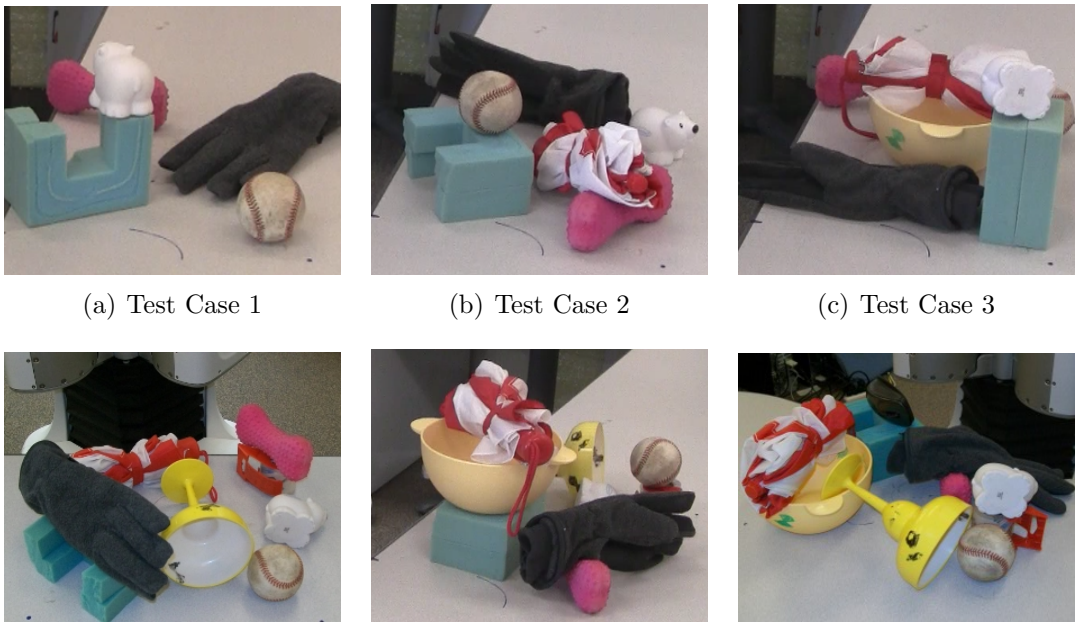
8.5.2 Clearing Table with PR2: Results

Tab. 8.9 shows success rates for all six test cases and the average calculation time per grasp in seconds. Pictures of all test cases are shown in Fig. 8.11.

Table 8.9: Grasp success rate (Suc.) in % and performance time in seconds: Rectangle Representation vs. (S)HAF

Method:		**Rec. Repr.**		**(S)HAF**	
Test Case	**#Obj**	**Suc.**	**Time**	**Suc.**	**Time**
TC 1	5	4/5	48	5/5	17
TC 2	6	3/6	51	6/6	15
TC 3	7	4/7	45	6/7	16
TC 4	8	6/8	48	7/8	15
TC 5	9	4/9	48	7/9	16
TC 6	10	5/10	47	8/10	15
Sum/Avg.	**45**	**26/45**	**47.8**	**39/45**	**15.7**

Using the (S)HAF approach the system successfully grasped 39 objects out of 45 tries, giving an overall success rate of 86.7%. *Rectangle Representation* achieved



(a) Test Case 1 (b) Test Case 2 (c) Test Case 3

(d) Test Case 4 (e) Test Case 5 (f) Test Case 6

Figure 8.11: Test cases with 5, 6, 7, 8, 9 and 10 objects for clearing the table

26 out of 45 successful tries, giving an overall success rate of 57.8%. Although, as mentioned above, the parameter setting was suboptimal with respect to time performance, the presented algorithm was three times faster than the *Rectangle Representation* algorithm. The six grasp failures for the (S)HAF approach had different underlying reasons. In test case 5, the system failed to grasp the glove because only for this single case the "integrated path planning" failed and parts of the bowl prevented the manipulator from approaching the glove as far as needed in the simulation environment and hence in the real grasp execution. For the other five failures, non-optimal HAF grasps were each time partly responsible. But in each of these cases, other factors also contributed to the failure: 3 times the object (toy bear in TC 3, pink bone in TC 4, Martini glass in TC 5) was touched and moved by the manipulator out of the initial position. One reason for premature touching of objects is incomplete data. For the Martini glass and the bowl the interior region was in general badly perceived as shown in an example of the bowl in Fig. 8.12. Hence path planning resulted in paths wherein the gripper collided with the (missing in simulation) interior surface of the bowl or Martini glass when grasping at the rim.

Videos of all test cases of the clearing the table scenario with PR2 can be found at *www.youtube.com/user/clearingthetable/*. The demo code used for the experiments (along with whole framework, in addition to the core algorithms) for grasping unknown objects with a PR2 (*Rectangle Representation* and (S)HAF) is available as ROS packages. This contribution enables a valid comparison of other

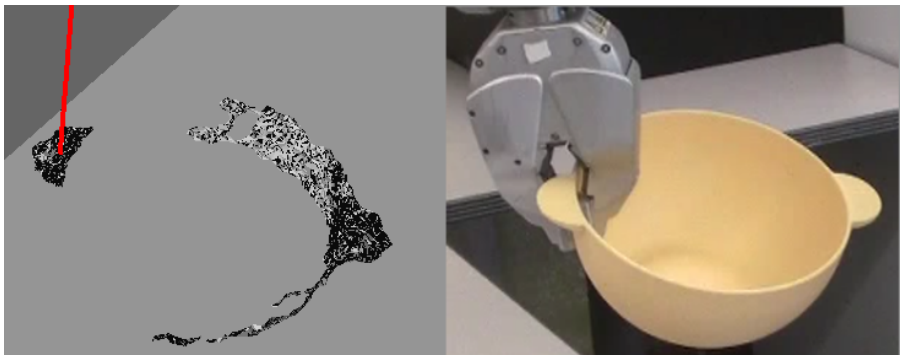


Figure 8.12: Incomplete perceived point cloud data (depicted is the result of a plastic bowl from a camera mounted at the robots head) is one reason for failed grasps

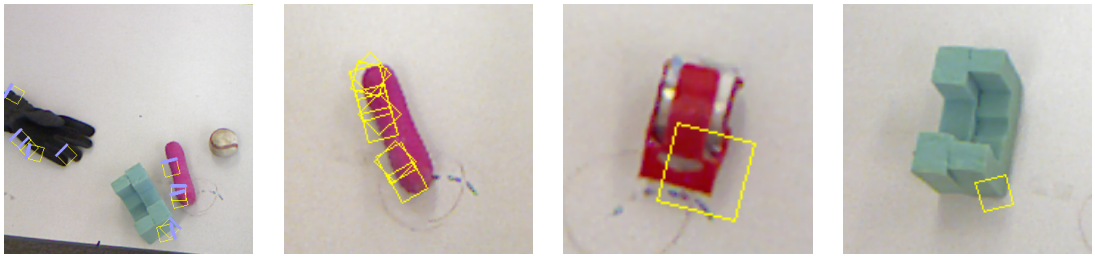
grasp detection algorithms with the two methods presented in this section. The code is available at:

http : //pr.cs.cornell.edu/grasping/rect_data/data.php

8.5.3 Comparison: Topographic Features vs. 2D Image Features

In Fig. 8.13 to Fig. 8.17, examples are presented to explain why the (S)HAF approach performed considerably better than the *Rectangle Representation* with respect to the grasp success rate. The results presented of the *Rectangle Representation* are mostly taken from above test cases, wherein the rectangle represents position and opening width of the gripper. The pictures show the top grasp or the top ten grasps in a scene. A thick violet line is used to indicate the gripper closing direction.

- **Opening width**
  Fig. 8.13 and the following images demonstrate that using the opening parameter may negatively affect the grasping and that an initially fully opened gripper leads to better results for *Rectangle Representation*. Therefore, the calculated gripper opening was not used in the experiments.



(a) Top ten grasps with gripper closing direction indicated violet

(b) Top ten grasps

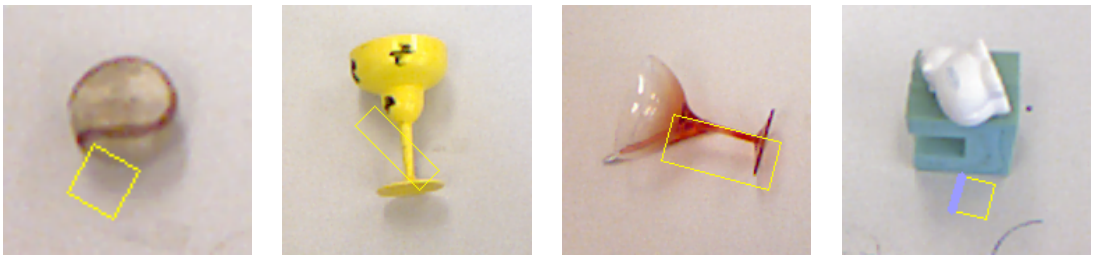
(c) Top result

(d) Top result

Figure 8.13: Calculated gripper width for Rectangle Representation: Examples of the grasp learning approach - Rectangle Representation which is mainly based on 2D images. If the robotic gripper approaches an object with an opening width corresponding to the Rectangle Representation, the gripper would not be able to encompass the object - but would touch it and hence would be stopped by the path planning routine before it has reached a position at which closing the gripper would succeed as grasp.

- **Shadow sensitive**
  Fig. 8.14 shows that shadows can have an impact on the *Rectangle Representation*. The (S)HAF approach is completely robust to shadows because they have no (topographical) impact on objects.

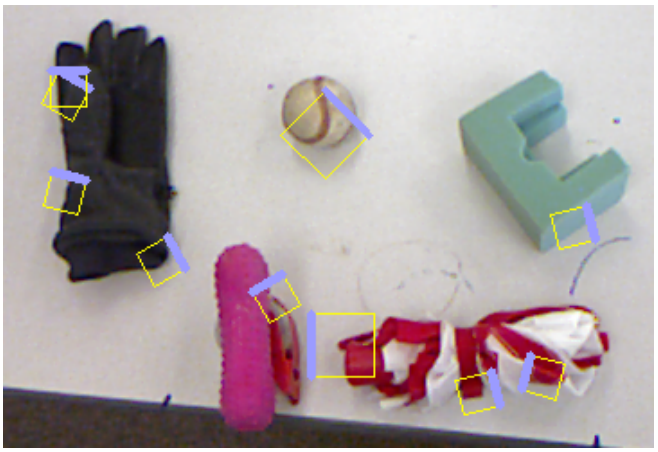
- **Edge focused**
  Images from Fig. 8.15 illustrate that grasp learning of *Rectangle Representation* often relies on the existence of edges in the image. For slim objects such as screwdrivers and pens or objects with thin boarders such as bowls



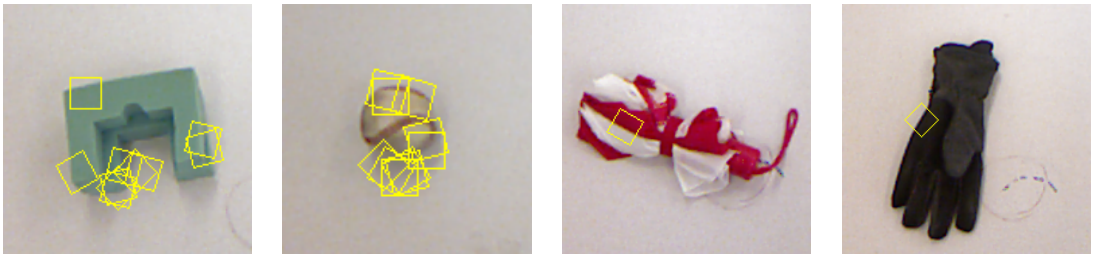
(a) Top result (b) Top result (c) Top result (d) Top result

Figure 8.14: Shadow: Rectangle Representation is sensitive to shadows. It happens that grasps are detected at shadow boarders.

this approach is beneficial, but for bigger objects the center of the grasp may be placed at the border of the object, which is suboptimal. The approach is also sensitive to color changes within the given object (see Fig. 8.15(d)). (S)HAF do not rely on color intensity values.



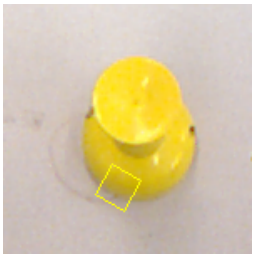
(a) Top Ten Grasps with closing direction



(b) Top Ten Grasps (c) Top Ten Grasps (d) Top result (e) Top result

Figure 8.15: Edge focused: Rectangle Representation often relies on edges in 2D images. Color changes often correlate with object boundaries and thereby indicate potential grasp positions, however, as the examples show this does not necessarily result in stable grasping points.

- **Surface independent**
  Fig. 8.16 shows examples of detected grasp hypotheses that illustrates that it is hard for the features in *Rectangle Representation* to take the object surface into account. There is no possibility to grasp an object when the gripper touches the object before any part of the object is in the area such that the gripper can surround the object when closing. The (S)HAF approach takes object surfaces and obstacles into account.



(a) Top result

(b) Top result

(c) Top result

(d) Top result

(e) Top ten grasps with closing direction

Figure 8.16: Surface independent: Rectangle Representation detects grasps at positions the gripper cannot reach because it would touch parts of the object before the gripper can successfully close. In Fig. 8.16(e) detected for the baseball and the white toy bear.

- **Height dependent**
  Fig. 8.17(a) shows an example of the top grasp with the *Rectangle Representation* of an umbrella at its cord. It makes it clear that the impact of height related features in this approach is not very strong which is a drawback since executing robot grasps at strings on planes is very hard. The (S)HAF ap-

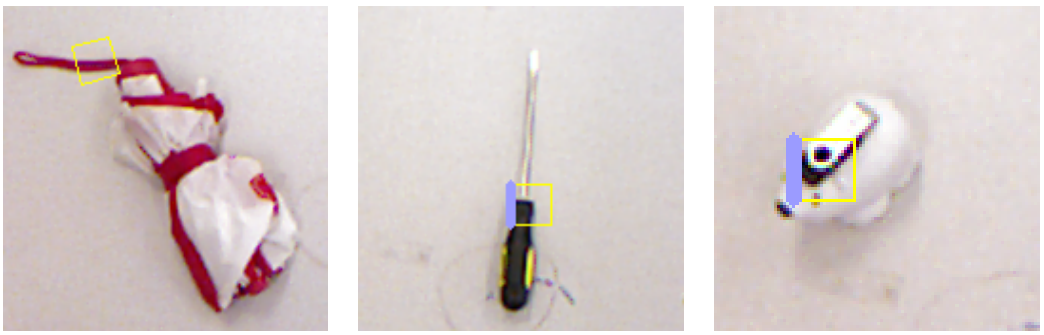
proach learns to prefer grasps at positions with large height differences (for top grasps) between objects and its surroundings.

- **Orientation**
  Fig. 8.17(b) shows an example where *Rectangle Representation* delivered a grasp orientation 90 degrees off to the optimal orientation (see violet line) because of the color crossing of a screwdriver handle. As the (S)HAF approach tries to surround objects with the gripper as far as possible before closing, an appropriate roll angle for slim rectangular objects is achieved.

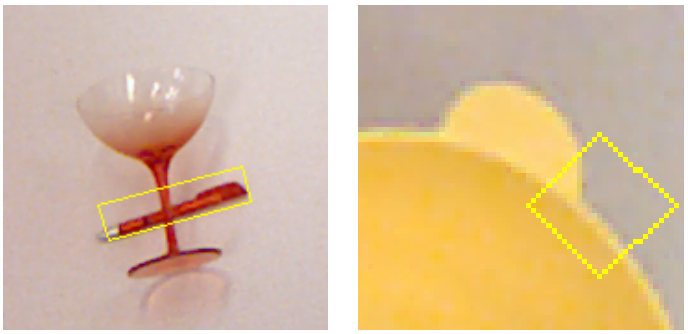
- **Perception**
  In some cases, the white toy bear depicted in Fig. 8.17(c) could not be detected on the white background although the *Rectangle Representation* algorithms compare the current image with the image of an empty table and the bear possesses contrasting black spots for nose and eyes. In comparison, the (S)HAF approach relies on perceived point cloud data. With new devices developed in recent years high quality data for a favorable price can be achieved. But there are still limits for 3D perception which suggests the combination of 3D and 2D data to overcome perception problems for shiny or transparent objects (see Section 8.8).

- **Obstacle avoidance and path planing** In Fig. 8.17(d), the center of the rectangle is not on the glass, but on the pen behind it. For the path planning, the glass is an unsurmountable obstacle for grasping the pen with a predefined vertical approach direction and rotation. The (S)HAF approach avoids grasps blocked by obstacles and tends to pick highest objects first, due to its "integrated path planning".

- **Object-related grasp points**
  Fig. 8.17(e) shows an example where the 2D image features identify a grasp point (center of rectangle) at a position belonging to the table surface instead of the rim of a bowl. The 2D to 3D mapping would deliver undesirable coordinates, which is especially a problem if for grasp planning a fixed grasp point offset is used. For the presented method it never occurred that a grasp center was positioned on the table surface. Furthermore, the implementation for the (S)HAF approach uses perceived object meshes in simulation to calculate the final grasp position given an approach vector, thereby minimizing collisions. This beneficial schema was also used for testing the *Rectangle Representation* approach.



(a) Height dependence (b) Orientation (c) Perception



(d) Path planning (e) Grasp center not on object

Figure 8.17: Examples of Rectangle Representation learning (with top evaluated grasp), images showing (a) a weak impact of height features, (b) a bad result for grasp orientation, (c) a scene in which perception of a white toy bear was only possible after augmenting with supporting texture, (d) a scene which illustrates a potential drawback when grasp detection relies too much on 2D features: a grasp center detected at a position not belonging to any object

8.6 Clearing the Floor with Hobbit robot

The goal of this experiment was to test (S)HAF on a mobile robotic platform (developed in the framework of the EU project Hobbit) along with an analysis of possible gripper opening parameters in the context of the applicability of (S)HAF to the 7-dimensional grasp space. The scenario for this experiment was the task of tidying up a floor in an apartment. The robot prototype (see Fig. 8.18) was based on a mobile platform and a 5-DOF IGUS arm with a 1-DOF Festo Fin Ray gripper (more details on the robot can be found in Fischinger et al. (2013a)).

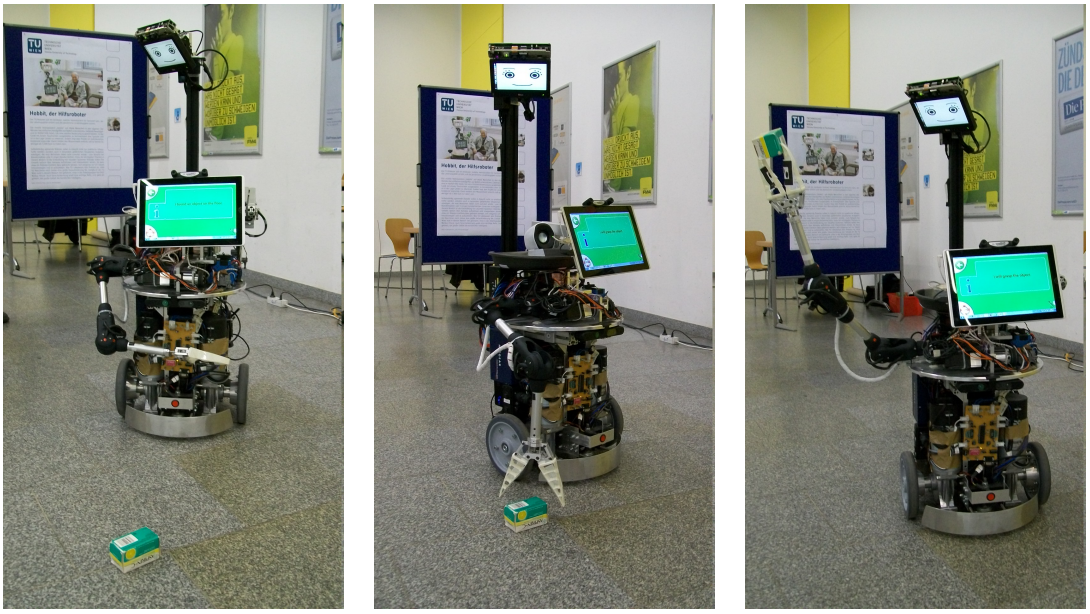


Figure 8.18: Hobbit clearing the floor by picking up an Aspirin box

8.6.1 Clearing the Floor with Hobbit: Test setup & Results

For experiments, Hobbit had to pick up objects that were dumped on the floor and place them in its tray. Floor detection is based on detection of horizontal planes at the expected floor height. 3D points of the floor are filtered out and (S)HAF grasp detection was tested along with variable opening widths of the gripper before it closes.

Hobbit was able to detect, approach, and grasp objects on the floor such as spectacle cases, remote control units, and small boxes. The resulting grasp grid for the case of an easy scenario with two aspirin boxes on the floor is depicted in Fig. 8.19(b). This result is for a fixed orientation and fixed gripper opening width. Green and red points indicate potentially good respectively bad grasping positions.

(a) Hobbit is grasping a tape in a scenario where a fully opened gripper would not succeed.

(b) Grasp grid results for one orientation and one gripper width for two aspirin boxes

Figure 8.19: Experiments with the Hobbit robot

The red line shows the gripper closing direction at the position of the top vertical grasp (above the bigger aspirin box).

In order to test the opening width parameter, opening widths of 1, 0.5 and 0.33 times the maximal width were used. Point cloud scaling (to simulate a partly opened gripper) was done only for the gripper closing axis since the gripper has only two anti-podal fingers. In test scenarios specially arranged for testing gripper width, the functioning of the presented approach was verified. Fig. 8.19(a) shows a scenario where the calculation of a proper gripper opening width is crucial, since a fully-opened gripper could not approach the object in the required way for grasping.

8.7 Grasping Unknown Objects with Kuka Arm

The goal of this experiment was to demonstrate the applicability of the (S)HAF approach on a Kuka arm as a fourth hardware setup with a robotic hand suitable to validate the scalability claim (see Chapter 7). Furthermore, it was focused on an error analysis and potential improvements for grasp detection.

8.7.1 Grasping Unknown Objects with Kuka Arm: Test Setup

The experiment was executed with a 7-DOF Kuka LWR arm and the Michelangelo hand from Otto Bock with two DOF (see Fig. 8.20). One Kinect device was used for data acquisition.

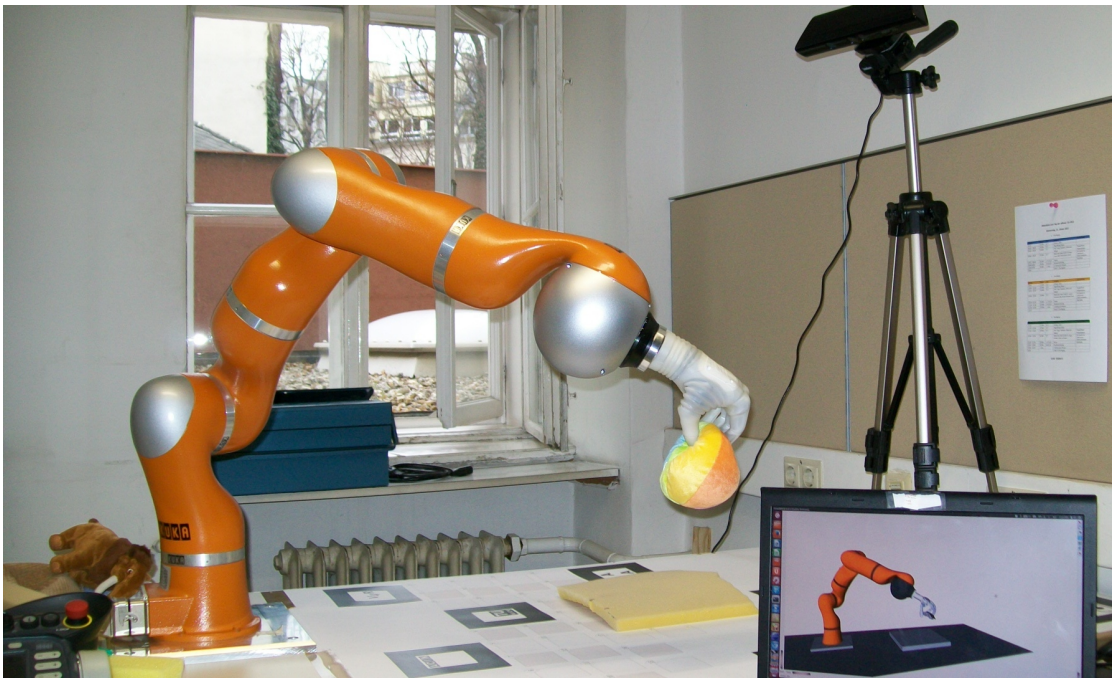


Figure 8.20: The setup for grasping unknown objects: A Kuka 7-DOF leight weight arm with attached Michelangelo Hand with two DOF, a calibration pattern for online calibration, and a Kinect device for data acquisition.

To show that the (S)HAF grasp detection is also usable with more complex robotic hands, a test series with ten objects (depicted in Fig. 8.21) and ten tries per object was performed. For each try, the object was placed in front of the robot arm with different orientations. Due to safety reasons, the objects were positioned on foam and the execution of each trajectory was started by pressing enter as only interaction after starting the grasp processing pipeline. For practical reasons (e.g. to improve robustness against calibration errors or missing point cloud data)

(a) Ball (b) Bowl (c) Cap (d) Case

(e) Mammoth (f) Pads (g) Pants

(h) Paper (i) Tape (j) Toy block

Figure 8.21: Test objects for grasping with Kuka arm and Michelangelo hand during experiments

it was always attempted to approach the final $7cm$ in a straight line retaining a constant orientation of the hand. This limits the probability for finding (inverse) kinematic solutions for specific grasps. This is also the main reason why the grasp approach direction in these experiments was restricted to mainly vertical approach rays (apart from small variations to find possible kinematic solutions). This way, the top-rated grasps can be executed instead of using first possible grasps. The hand roll steps were chosen to be 15 degrees and grasp calculation was done for all rotations. In other words, the grasp estimation was not terminated if a good grasp has already exceeded a threshold value - which could enhance time performance significantly depending on the selection of the threshold.

A grasp is classified as successful if the robot arm delivers the grasped object to a defined position next to the table where the test object was placed.

8.7.2 Grasping Unknown Objects with Kuka Arm: Results

A grasp success rate of 85% was achieved. For detailed success rates for each object see Tab. 8.10, a more detailed failure analysis is done in Section 8.7.3.

Table 8.10: Grasp success rate in % and performance time in seconds with Michelangelo hand. For each object ten trials were executed.

Item	**Success in %**	**Time in sec.**
Ball	100	8.8
Bowl	90	10.1
Cap	100	11.0
Case	50	8.1
Mammoth	90	11.2
Pads	100	9.9
Pants	90	14.0
Paper	80	8.4
Tape	90	8.0
Toy block	60	5.8
AVERAGE	**85**	**9.5**

8.7.3 Error Analysis and Potential Improvements

Nine out of 15 failed grasps happened when grasping the spectacle case or the small wooden toy block. Furthermore, the spectacle case slipped out of the hand prosthesis in five out of ten trials for successful pre-grasps. The rigid convex shape of the object in combination with relatively strong gripper force and a week friction due to the object surface material made the spectacle case slip out of the hand in each of the failure cases.

For the soft pads, the video tape and the newspaper (with the elastic band) the grasps were sometimes not centered but at the side of the object (see Fig. 8.22). The resulting torque led in some cases to unstable grasps and grasp failures during the delivery of the object (e.g. Fig. 8.22(c)-8.22(e) where the tape slid out of the hand shortly before the delivery position). Therefore, the problem of these suboptimal grasps was investigated and two sources of error could be identified.

In Fig. 8.23 the mesh of the video tape is shown. The camera could only perceive the upper surface of the object. This surface is actually flat in real world, but the mesh generated from the perceived point cloud is rather bumpy. In the depicted case the grasp was chosen above one of the peeks in the surface. This problem

(a) Paper grasp at side (b) Pads grasp at side (c) Tape grasp at side (1) (d) Tape grasp at side (2) (e) Tape grasp at side (3)

Figure 8.22: Weak grasps at side of paper, soft pads and video tape.

is related with the second source of error which becomes also apparent if objects like the soft pads lie on a slope: The grasp classifier does not always accept grasp positions if there are higher (w.r.t top grasps) surface points next to the position to classify. Therefore, grasps on the sides of objects are returned. Adding training examples focusing on these surface constellations should improve the performance of the classifier.

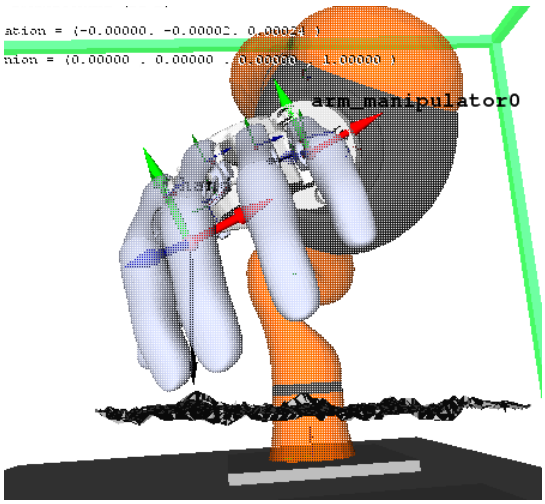


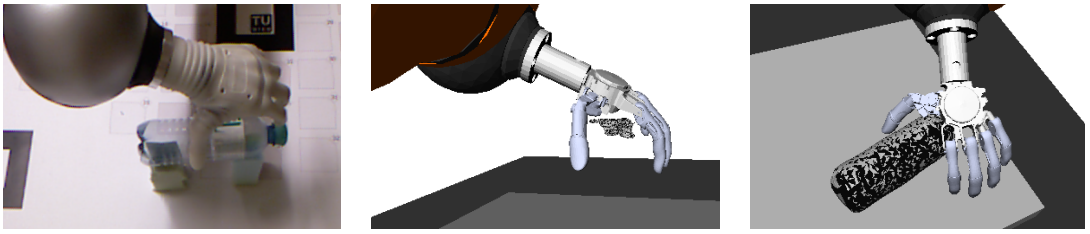
Figure 8.23: Grasp position detected at the side of an object (video tape) due to inaccurate point cloud data and extremely sensitive grasp classifer w.r.t. height differences

8.8 Grasping Known Objects with Kuka Arm

The goal of this final experiment was to combine the S(HAF) approach with object recognition in order to demonstrate the advantages of the symbiosis and how (S)HAF can make grasping of known objects more reliable.

State-of-the-art object recognition (e.g. for homogeneously colored, deformable objects like the trousers in Fig. 8.21(g) and related object segmentation as prerequisite for grasping are quite limiting (e.g. in very cluttered environments) from the experience of the author. On the other hand, object specific grasp force adaptation or task based grasping and manipulation are needed for future mobile service robots that should at one time be able to support people in their everyday life.

In Section 8.5.3, the limits of point cloud perception of current sensors for transparent objects were mentioned. Fig. 8.24 shows a transparent plastic bottle with a label and its perceived point cloud data (i.e. the generated mesh). Insufficient object data leads to grasps in which the robotic hand would collide with the object endangering the intactness of the hardware or the object. Using object recognition



(a) Transparent plastic bottle (b) Perceived data of bottle with calculated grasp (c) Complete object model due to recognition

Figure 8.24: The left picture shows a bottle seen from the camera, the center picture shows the incomplete mesh/point cloud data received from a Kinect laser sensor and the resulting hand orientation for grasping that would lead to a collision between manipulator and object that cannot be determined by the system. The right picture shows the resulting object mesh if the object is recognized (SIFT) and an object model is used as input for grasping. With object model available, a suitable grasp was found.

from Aldoma et al. (2013) implemented with SIFT on 2D data (from multiple camera views), a complete pre-learned mesh of the object can be achieved. Complete object models are no guarantee for good grasps as Fig. 8.25 indicates, where valid grasps for force closure evaluation are shown.

8.8.1 Grasping Known Objects with Kuka Arm: Test Setup

To show the benefit of combining (S)HAF with object recognition in specific cases, an experiment on grasping the bottle ten times using the (S)HAF method was performed: (1) without object recognition and (2) using an object model perceived by

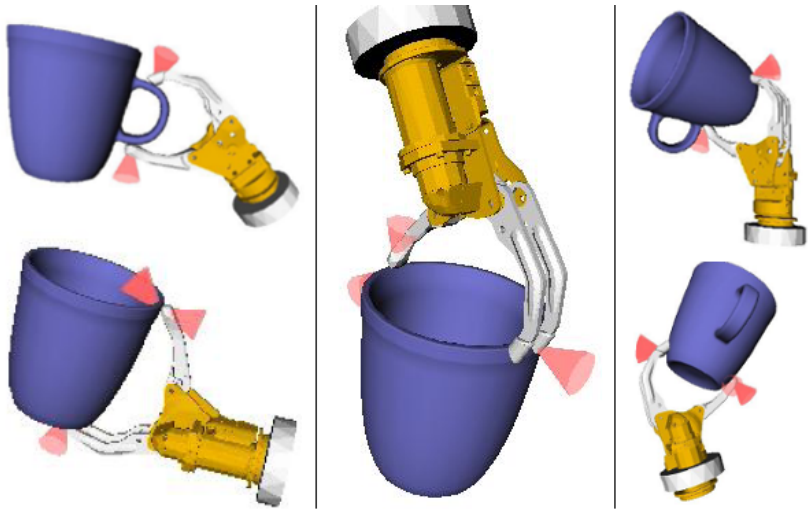


Figure 8.25: Complete object models do not guarantee good grasps. The depicted grasps calculated for complete object models and an Otto Bock "SensorHand Speed" fulfill force closure criteria evaluated in OpenRAVE. Still, the grasps are not stable in practice.

object recognition. The general test setup is identical to that stated in Section 8.7 for both conditions.

8.8.2 Grasping Known Objects with Kuka Arm: Results

The results are listed in Tab. 8.11: With the conventional approach we could only grasp the bottle in three out of ten tries. Using object recognition and complete models of the bottle, each grasp succeeded. For the latter, only tries were considered where the object recognition module delivered a model. This was not the case for every try due to reflections from the shiny surface of the bottle (see Fig. 8.26) and

Figure 8.26: Object recognition failed in some cases due to light reflections and structural degeneration from wear and tear. After the 8th trial we had to exchange the bottle for better recognition performance.

abrasion of the bottles surface during grasping. These preliminary experiments could show advantages of combining our method with object recognition. Object recognition not only improves success rates in grasping using our approach, but also results in better quality of grasps.

Table 8.11: (S)HAF vs. (S)HAF&Object Recognition: Grasp success rate and performance time in seconds. (*) restart of laptop

Try	**HAF Suc.**	**Time (sec)**	**HAF&Recog.**	**Time (sec)**
1	0	35	1	12
2	1	33	1	12
3	1	30	1	12
4	1	27	1	11
5	0	7*	1	10
6	0	10	1	11
7	0	9	1	13
8	0	7	1	13
9	0	6	1	11
10	0	6	1	11
SUM/AVG	**3/10**	**17***	**10/10**	**11.6**

Chapter 9

Conclusion and Outlook

The human visual perception system is based on simple local features such as intensity values, color registration for red, green and blue, and a preliminary edge detection in the ganglion cells (Dhande and Huberman, 2014). This basic signal input is processed in different parts of the brain to achieve a higher cognitive level of perception, for example for three-dimensional cognition or object recognition, to enable interpretation and interaction with the surrounding environment.

Modern grasp approaches try to imitate this process, using simple features based on edge or texture information in learning- or early-cognitive systems (Kootstra et al., 2012). Although repeatedly and medially exploited progress is reported in the field of neural science, the investigation and simulation of human brain processes are nowadays still quite limited. Due to this lack of computational cognitive processing, grasp approaches based on simple features are limited at present. As key contribution of this thesis, a new feature type is proposed to enhance existing early cognitive visual systems specifically for the task of grasping, due to its abstraction power of grasp-relevant object surface structures.

9.1 Summary

This thesis presents an approach for enabling robots to grasp known and unknown objects. The algorithm does not rely on object models or segmentation and works robustly in cluttered scenes. The core contribution of this work is a powerful feature type called Height Accumulated Features (HAF) particularly suited for the task of robot grasping and manipulation in domestic environments. Advantages of this approach are demonstrated compared to other approaches, especially ones based on 2D color feature. Experiments are conducted in relation to a state-of-the-art representative and demonstrate the superiority, as well as the advantages of topographic surface abstracting features based on numerous examples.

The focus of the research presented, is an approach, which is robust to the numerous and dynamically changing task scenarios in the real world and hence is

subsequently useful for mobile robotics, being not just based on theoretical simulation. "Integrated path planning" guarantees that the selected grasps do not fail due to collisions with other objects and a heuristic ensures the robustness of the grasps with respect to positioning errors.

The abstraction power and information gain obtained from HAF is shown by comparing it to a classifier trained on point clouds discretized to height grids. The enhancement of the approach with Symmetry Height Accumulated Features showed easy extensibility with considerable gain. A thorough analysis of the improved features showed that more complex features lead to better grasp classification results than basic features. Given a trained grasp classifier, a method was introduced to explore the seven-dimensional grasp space (position, orientation, gripper opening width).

Tests were implemented on four different robot platforms for different tasks like grasping single objects, tidying the floor, emptying a box and clearing a table without gripper specific classifier training, thereby indicating the hardware scalability of the approach (which also relies on the use of a robot model for final grasp calculation in simulation).

Finally, it is shown how to combine the presented approach with object recognition to overcome the problem of incomplete point cloud data and exploit the advantages of the presented approach if complete object models are available. Videos of the experiments and the code from the experiments presented in Sections 8.4 and 8.5 have been made available online.

9.2 Contributions to Research Problems

The overall goal of this thesis was to investigate the problem of grasping objects, defined by detecting a gripper pose in the seven-dimensional grasp space (position, orientation, gripper opening width), where a mechanical gripper has to close for a suitable grasp, and the approach trajectory to reach the final grasp position. For service robots grasping is needed in various scenarios, which pose different problems. The research presented in this thesis contributes to three different problem types:

1. Problem type A: grasping known object

2. Problem type B: grasping unknown object

3. Problem type C: grasping objects in clutter

The task of grasping known objects could up to now only be considered as solved under very restrictive conditions. With the approach presented in this thesis it becomes possible to:

- **skip the effort for creating pre-learned grasp database:** The presented approach for finding grasps can be applied on full object models for online grasp calculation, making a database of grasps per object and gripper obsolete.
- **create a pre-learned grasp database semi-automatically:** The effort for generating a database with pre-learned grasps per object and gripper can be reduced with semi-automatic grasp generation given the gripper- and object model.
- **enable object recognition by interactive manipulation of unrecognized objects:** Due to various reasons (e.g. lightening conditions, occlusions, unlearned views), recognition can fail. The presented approach gives the opportunity to change object poses by grasping without pre-learned positions, which could create new conditions and hence enable successful object recognition, e.g. for task specific grasping, where object recognition is crucial.

The task of grasping unknown objects was still an open research problem. With the approach presented in this thesis it becomes possible to:

- **grasp objects without estimation of unknown shapes:** The approach focuses on grasps on perceived surfaces where the gripper approaches objects without the need to estimate the surface of the object which is facing away from the camera.
- **significantly improve state-of-the-art grasp detection:** A comparison with Jiang et al. (2011) shows an improvement of the grasp success rate of 34% for single objects (Section 8.4).

The task of grasping unknown objects in clutter was the biggest research challenge addressed in this thesis and substantial contribution could be made. The (S)HAF approach:

- **significantly improves state-of-the-art grasping:** A comparison with Jiang et al. (2011) shows an improvement of the grasp success rate of 28.9% in trials with five to ten objects in cluttered scenes (Section 8.5).
- **works independent from object segmentation:** Complex tasks such as autonomously emptying a basket can be executed without the need for segmentation. Thereby, a complementary approach to methods which need segmented input, such as Superquadric fitting, is provided.
- **implicitly learned local path planning:** In contrast to recently published grasping algorithms (Varadarajan and Vincze, 2011; Kootstra et al., 2012) the (S)HAF approach handles grasp planning and path planning not independently. The presented method learns to select grasp hypotheses which result in collision-free local paths for the gripper used and the given approach vector.

9.3 Outlook

The presented approach deals with the unresolved problem for optimal gripper (approach and) placement for robotic grasping. Although the author believes that the methods and results presented in this thesis serve as a stepping stone on the way to enhanced cognitive visual systems for grasping, more research will be needed to regard this important research question as solved. Nevertheless, there are a number of related aspects and potential extensions that were not discussed in this work.

Using visual and/or tactile feedback during and after grasping can improve failure handling and hence make the approach more robust. Furthermore, a check if a grasp was successful after closing the robotic hand can improve the time needed for successful grasping tasks like clearing the floor with mobile robots. Although a very efficient way to calculate Topographical Features with accumulated heights was presented, time performance was not the primary focus of this work. The huge number of feature values needed to explore the grasp space results in calculation times of at least a few seconds. Especially, because in the majority of the experiments, the overall best grasp was calculated first and then executed to optimize the evaluated grasp success rate. A strategy that accepts the first feasible grasp, combined with the before mentioned online grasp success check and dependent regrasping actions in failure cases, can improve both, the grasp success as well as the average grasping time. By optimizing the implementation, even grasp calculation in real-time seems to be reasonable. For real-time grasp calculation visual servoing with relative movement control for the gripper seems to be an interesting approach for future service robots like the Hobbit robot.

A further potential extension of the presented approach is attention driven grasping, thereby reducing the desired object or area to grasp by different criteria. Defining objects by attributes such as color, size, or form can be an example for this attention driven grasping. The goal is not anymore "grasp anything possible" for unknown objects, or "grasp exactly this object" for known objects, but something like "grasp an object that should have a height between 5 and $9cm$, has a spherical form and orange color. Using such attributes enables usage of higher level learning of objects and at some point a real cognitive learning system for advanced robots. A simpler, already implemented approach for the second prototype of Hobbit uses human body detection to limit grasping goals to attention points. A user can point to an object with his arm, the robot follows the direction of the pointing arm and the forefinger, and tries to grasp an object intersecting with the imaginary, extended pointing direction.

Interactive segmentation is a further area where the presented approach could be used. For a messy pile of objects, the robot could randomly grasp something and separate it from the pile. This way, the approach can support object recognition and hence task specific manipulation.

A major area to improve the presented research work are the described features. As mentioned in Chapter 5, more complex features achieved better grasp

performance, respectively higher discriminative values for the grasp classifier. Using features with more than four regions is only one way to extend the presented idea.

List of Tables

List of Figures

Pseudo code listing

Index

Bibliography

Rectangle Representation Code and HAF Framework available as ROS Packages, 2012. URL `http://pr.cs.cornell.edu/grasping/rect_data/data.php/`.

A. Aldoma, F. Tombari, J. Prankl, A. Richtsfeld, L. D. Stefano, and M. Vincze. Multimodal Cue Integration through Hypotheses Verification for RGB-D Object Recognition and 6DOF Pose Estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2096–2103, 2013.

T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann. Toward humanoid manipulation in human-centred environments. *Robotics and Autonomous Systems*, 56:54–65, 2008. ISSN 09218890. doi: 10.1016/j.robot.2007.09.013.

P. Azad, T. Asfour, and R. Dillmann. Stereo-based 6D object localization for grasping with humanoid robot systems. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007. ISBN 978-1-4244-0912-9. doi: 10.1109/IROS.2007.4399135.

R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka. Physical Human Interactive Guidance: Identifying Grasping Principles From Human-Planned Grasps. *IEEE Transactions on Robotics*, 28(4):899–910, 2012. ISSN 1552-3098.

D. Berenson, R. Diankov, and J. Kuffner. Grasp planning in complex scenes. *2007 7th IEEERAS International Conference on Humanoid Robots*, pages 42–48, 2007.

A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 10, pages 348–353. Sandia National Laboratories, Ieee, 2000. ISBN 0780358864.

J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-Driven Grasp Synthesis - A Survey. *IEEE Transactions on Robotics*, (accepted), 2014. URL `http://arxiv.org/pdf/1309.2660v1.pdf`.

J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales. Mind the Gap - Robotic Grasping under Incomplete Observation. *ICRA*, pages 686–693, 2011.

B. Calli, M. Wisse, and P. Jonker. Grasping of unknown objects via curvature maximization using active vision. *IEEERSJ International Conference on Intelligent Robots and Systems*, pages 995–1001, 2011. ISSN 21530858.

C.-c. Chang and C.-j. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1—-27:27, 2011. ISSN 21576904.

Y.-w. Chen and C.-j. Lin. *Combining SVMs with Various Feature Selection Strategies*, volume 324 of *Studies in Fuzziness and Soft Computing*. Springer, 2006. ISBN 9783540354871.

A. Collet and S. Srinivasa. Efficient multi-view object recognition and full pose estimation. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010. ISSN 1050-4729. doi: 10.1109/ROBOT.2010.5509615.

A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *2009 IEEE International Conference on Robotics and Automation*, 2009. ISBN 978-1-4244-2788-8. doi: 10.1109/ROBOT.2009.5152739.

F. C. Crow. Summed-area tables for texture mapping. In *Proceedings of SIGGRAPH*, pages 18(3):207—-212, 1984. ISBN 0897911385.

R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Kruger, O. Kroemer, J. Peters, and J. Piater. Learning object-specific grasp affordance densities. In *2009 IEEE 8th International Conference on Development and Learning*, 2009. ISBN 978-1-4244-4117-4. doi: 10.1109/DEVLRN.2009.5175520.

R. Detry, D. Kraft, A. Buch, N. Kruger, and J. Piater. Refining grasp affordance models by experience. In *IEEE International Conference onRobotics and Automation (ICRA)*, 2010. ISBN 978-1-4244-5038-1. doi: 10.1109/ROBOT.2010.5509126.

O. S. Dhande and A. D. Huberman. Retinal ganglion cell maps in the brain: Implications for visual processing. *Current Opinion in Neurobiology*, 24:133–142, 2014. ISSN 09594388.

R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, 2010. URL `http://www.programmingvision.com/rosen_diankov_thesis.pdf`.

R. Diankov. OpenRAVE Documentation, 2012. URL `http://openrave.org/docs/latest_stable/openravepy/databases.grasping/`.

R. Diankov and J. Kuffner. OpenRAVE : A Planning Architecture for Autonomous Robotics. In *Tech. Rep. CMU-RI-TR-08-34, Robotics Institute*, number July. Robotics Institute, Carnegie Mellon University, 2008.

C. Ferrari and J. Canny. Planning optimal grasps. *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2290–2295, 1992.

D. Fiedler and H. Müller. "Impact of thermal and environmental conditions on the Kinect sensor.". In *Advances in Depth Image Analysis and Applications*, pages 21–31. Springer Berlin Heidelberg, 2013.

D. Fischinger and M. Vincze. Empty the basket - a shape based learning approach for grasping piles of unknown objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2051–2057, 2012a. ISBN 978-1-4673-1736-8.

D. Fischinger and M. Vincze. Shape Based Learning for Grasping Novel Objects in Cluttered Scenes. In *10th IFAC Symposium on Robot Control (SYROCO)*, pages 787–792, September 2012b.

D. Fischinger, P. Einramhof, W. Wohlkinger, K. Papoutsakis, P. Mayer, P. Panek, T. Koertner, S. Hofmann, A. Argyros, M. Vincze, A. Weiss, and C. Gisinger. Hobbit - The Mutual Care Robot. In *Assistance and Service Robotics in a Human Environment Workshop in conjunction with IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013a.

D. Fischinger, Y. Jiang, and M. Vincze. Learning Grasps for Unknown Objects in Cluttered Scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 609 – 616, 2013b.

C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen. The Columbia grasp database. In *Proceedings of the IEEE International Conference on Robotics and Automation (2009)*, pages 1710–1716. Ieee, 2009. ISBN 9781424427888.

C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp Planning via Decomposition Trees. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, number April, pages 4679–4684. Citeseer, Ieee, 2007. ISBN 1424406021.

A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-based learning of grasp selection. In *IEEE International Conference on Robotics and Automation*, pages 2379–2384, May 2012. ISBN 978-1-4673-1405-3.

K. Huebner and D. Kragic. Selection of robot pre-grasps using box-based shape approximation. In *International Conference on Intelligent Robots and Systems*, pages 1765–1770. IEEE, 2008. ISBN 9781424420575.

K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann. Grasping known objects with humanoid robots: A box-based approach. In *International Conference on Advanced Robotics*, pages 1–6. IEEE, 2009. ISBN 9781424448555.

Y. Jiang, S. Moseson, and A. Saxena. Efficient Grasping from RGBD Images : Learning using a new Rectangle Representation. In *ICRA*, pages 3304–3311. IEEE, 2011. ISBN 9781612843803.

D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stent. Perceiving, Learning, and Exploiting Object Affordances for Autonomous Pile Manipulation. In *Robotics: Science and Systems Conference*, 2013.

U. Klank, D. Pangercic, R. Rusu, and M. Beetz. Real-time CAD model matching for mobile manipulation and grasping. *2009 9th IEEE-RAS International Conference on Humanoid Robots*, 2009. doi: 10.1109/ICHR.2009.5379561.

E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib. Grasping with Application to an Autonomous Checkout Robot. In *ICRA*, pages 2837–2844. IEEE, 2011. ISBN 9781612843803.

G. Kootstra, M. Popovic, J. A. Jorgensen, K. Kuklinski, K. Miatliuk, D. Kragic, and N. Kruger. Enabling grasping of unknown objects through a synergistic use of edge and surface information. *The International Journal of Robotics Research*, 31(10):1190–1213, 2012. ISSN 0278-3649.

Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng. Learning to grasp objects with multiple contact points. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5062–5069. IEEE, 2010. ISBN 9781424450381.

Y. Li and N. S. Pollard. A shape matching algorithm for synthesizing humanlike enveloping grasps. In *5th IEEE/RAS International Conference on Humanoid Robots*, pages 442–449. IEEE, 2005. ISBN 0780393201.

Z. Li and S. S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal on Robotics and Automation*, 4(1):32–44, 1988. ISSN 08824967.

A. Maldonado, U. Klank, and M. Beetz. Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information. In *International Conference on Intelligent Robots and Systems (IROS)*, 2010. ISBN 978-1-4244-6674-0. doi: 10.1109/IROS.2010.5649185.

M. T. Mason and J. K. Salisbury Jr. *Robot Hands and the Mechanics of Manipulation*. The MIT Press, 1985. ISBN 0262132052.

A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *ICRA*, volume 2, pages 1824–1829. IEEE, 2003. ISBN 0780377362.

A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Automation Magazine*, 11(4):110–122, 2004. ISSN 10709932.

A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann. Integrated Grasp Planning and Visual Object Localization For a Humanoid Robot with Five-Fingered Hands. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2006)*, pages 5663–5668, 2006. ISBN 142440259X.

C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, 31(4):538–553, 2012.

N. S. Pollard. Closure and Quality Equivalence for Efficient Synthesis of Grasps from Examples. *The International Journal of Robotics Research*, 23(6):595–613, 2004. ISSN 02783649.

J. Prankl. *From Images to Objects - Vision for Cognitive Robotic Experimentation*. PhD thesis, Vienna University of Technology, 2011.

M. Przybylski, T. Asfour, and R. Dillmann. Planning grasps for robotic hands using a novel object representation based on the medial axis transform. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1781–1788. Institute for Anthropomatics, Karlsruhe Institute of Technology, Germany, IEEE, 2011. ISBN 9781612844558.

D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng. Grasping Novel Objects with Depth Segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2578–2585. IEEE, 2010. ISBN 9781424466764.

M. Richtsfeld and M. Vincze. Grasping of Unknown Objects from a Table Top. In *Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments*, 2008. URL `http://hal.inria.fr/inria-00325794/en/`.

A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic Grasping of Novel Objects using Vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008a. ISSN 02783649.

A. Saxena, L. L. S. Wong, and A. Y. Ng. Learning Grasp Strategies with Partial Shape Information. In *AAAI*, volume 3, pages 1491–1494. AAAI Press, 2008b. ISBN 9781577353683.

K. Shimoga. Robot Grasp Synthesis Algorithms: A Survey, 1996. ISSN 0278-3649.

K. M. Varadarajan and M. Vincze. Object Part Segmentation and Classification in Range Images for Grasping. In *International Conference On Advanced Robotics ICAR 2011*, pages 21–27, 2011. ISBN 9781457711572.

P. Viola and M. J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. ISSN 09205691.

J. Weisz and P. K. Allen. Pose error robust grasping from contact wrench space metrics. In *2012 IEEE International Conference on Robotics and Automation*, pages 557–562. IEEE, May 2012. ISBN 978-1-4673-1405-3.

W. Wohlkinger and M. Vincze. 3D Object Classification for Mobile Robots in Home-Environments Using Web-Data. In *International Workshop on Robotics in AlpeAdriaDanube Region RAAD*, pages 247–252. IEEE, 2010. ISBN 9781424468867.