Project no.:	**027657**
Project full title:	**Perception, Action & Cognition through learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**
Deliverable no.:	**D2.2.1**
Title of the deliverable:	**Grasping Primitives and Haptic Exploration of Objects**

Contractual Date of Delivery to the CEC:	31. January 2010
Actual Date of Delivery to the CEC:	12. February 2010
Organisation name of lead contractor for this deliverable:	KTH
Author(s):	Tamim Asfour, Alexander Bierbaum, Kai Hübner, Danica Kragic and Norbert Krüger
Participant(s):	KTH, UniKarl, SDU
Work package contributing to the deliverable:	WP2 (partially related to WP1, 4 and 8)
Nature:	D
Version:	Draft
Total number of pages:	7
Start date of project:	1st Feb. 2006 **Duration:** 48 month

Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)
Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

WP2 is concerned with grasp posture modeling strategies, reactive grasping, and corrective movements for a robot hand, thus providing the basics for manipulation. Consequently, the final goal is to equip the hardware system with sensing and motor components that can be used for learning Object-Action-Complexes (OACs). As part of WP2.2, this deliverable reports particular work on grasping primitives and haptic exploration of objects. The included publications discuss different aspects of this area, including grasping primitives from 3D edge information, grasping primitives from box-based shape approximations, and haptic exploration.

Keyword list: Grasping Primitives, Haptic Exploration, 3D Object Representations.

Table of Contents

1. Executive Summary

The core focus of WP2 is on grasp posture modeling strategies, reactive grasping, and corrective movements for a robot hand, thus providing the basics for manipulation. Manipulation requires several basic capabilities of the robot to perceive and act, where of we approach

- acquisition of object information to form object representations (WP2 and WP4),
- generation of grasp hypotheses / reactive grasping (WP2.2, this deliverable),
- correction or verification of object information and object representations (WP2), and
- integration and demonstration of the developed strategies (WP1 and WP8).

As part of WP2.2, this deliverable reports particular work on grasping primitives and haptic exploration of objects. To do this, object representations in terms of 3D shape are necessarily required and immanent. While 3D shape representations also play a major role in WP4 "Object-Action Complexes", the relevance of WP2 is described by investigating grasp generation, haptic feedback and corrective finger movements.

It is important to note the complementary nature of the two different developed grasp generation techniques which we will describe in Section 2 and 3. While in Section 2, we successfully refer to grasp generation based from pure 3D edge-based representations of objects for sparse 3D data, Section 3 describes a scientifically differently motivated approach of grasp generation based on dense 3D data. Our publications prove that in both fields, we contributed to the state-of-the art, while project-internally, we value advantages of both techniques in terms of a complementary combination. However, it has early been recognized that successful grasping, especially grasp generation on unknown objects, can not be sufficiently solved by such vision-based methods only. Thus, besides presenting the impact of learning aspects in Section 2, we also conclude our contributions in terms of haptic exploration, which we believe can clearly boost the generation of successful grasps, in Section 4.

To conclude our introduction, this deliverable comprises 5 published papers and 1 technical report emerged from the project during the last project period (months 36 to 48). These publications discuss different aspects of different areas. Mainly, these are

- grasping primitives from 3D edge information,
- grasping primitives from box-based shape approximations, and
- haptic exploration.

In the following, we report the specific contributions of each of these topics before briefly sketching the papers related and attached to this deliverable.

2. Grasping Primitives from 3D Edge Information

We have presented a system to autonomously create and evaluate grasping hypotheses based on 3D edge information. For each scene, a large number of grasps become computed by using co-planar 3D contours as in [4]. We then learn an evaluation function which associates a success likelihood to each grasp by using a neural network. The learning is based on labeled data in terms of executed grasps that have been evaluated autonomously (i.e., the visual features the grasp has been derived from and its haptically measured success).

The integration of learning, both offline and online, enabled us to increase the overall success ratio of grasps by predicting their outcome. Especially the integration of online learning allows to define more complex grasping strategies, addressing the exploration of the feature space. The extraction of 3D edge primitives and the system diagram of creating and evaluating grasp hypotheses automatically is shown in Fig. 1 and 2.

A more detailed formulation about the complete system has been included in this deliverable [A1].

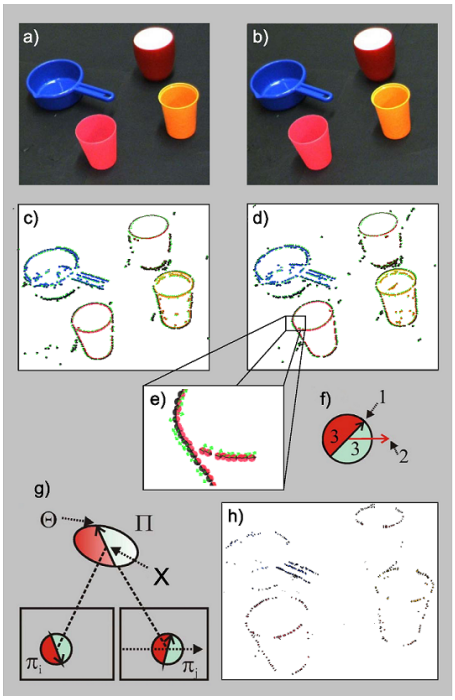


Figure 1: The extraction of primitives.

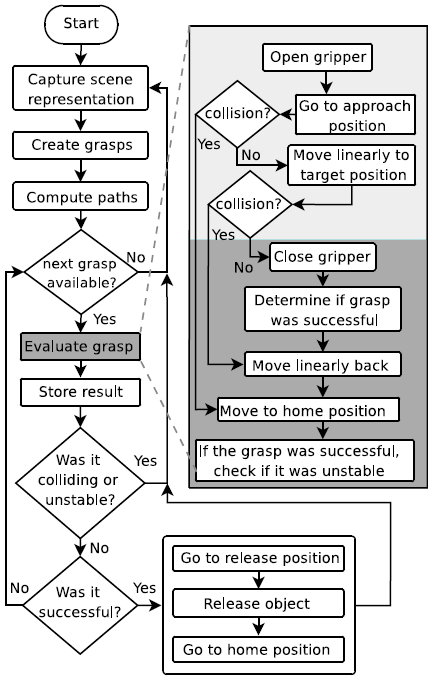


Figure 2: Flow diagram for creating and evaluating grasp hypotheses.

3. Grasping Primitives from Box-Based Shape Approximations

In the final phase, we presented the continuation of box approximation for the purpose of robot grasping. We specified the core algorithm and specific extensions of connecting box shape approximation and grasp hypotheses generation in earlier work [2, 3] and extended it in the attached recent publications [A2, A3]. In our approach, we prune the search space of possible approximations and grasp hypotheses by rating and decomposing very basic shapes, which intuitively corresponds to the "grasping-by-parts" strategy. In the attached technical report [A4], we focused in greater detail on all the parts of an entire framework taking advantage of the very simple shape representation of boxes. Starting from boxes and their facets which the algorithm extracts from 3D point data, we extended the idea of "grasping on boxes" towards an applicable grasping strategy. This strategy includes various heuristical selection criteria based on efficient geometrical calculations, as also learning from off-line simulation. The strength of our approach can be seen in its simplicity and its modularity. The simplicity is obvious by using boxes and rectangular facets in 3D space.

The proposed framework presented is one of few that approach 3D shape approximation from dense stereo data instead of 3D range data or 3D meshes for the purpose of grasping. The source of data for the presented algorithms is arbitrary, as long as it represents 3D point clouds. Nevertheless, the high complexity and manifold difficulties of a vision-based approach were pointed out. A conclusion drawn is clearly that experimental setups applying complete models from a database of known objects [A3] result in higher performance in terms of shape approximation as those applying online data gathered from single-view camera dense stereo [A4]. However, we believe that the proposed framework is flexible enough to be extended toward such issues. Merging the 3D data from stereo with 3D data from haptic contact points along regions of shape uncertainty of objects may therefore be a solution, and motivates fine correction based on haptic feedback [5] or haptic data acquisition and exploration.

Some examples for generated grasps from box-based shape approximation can be seen in Fig. 3.

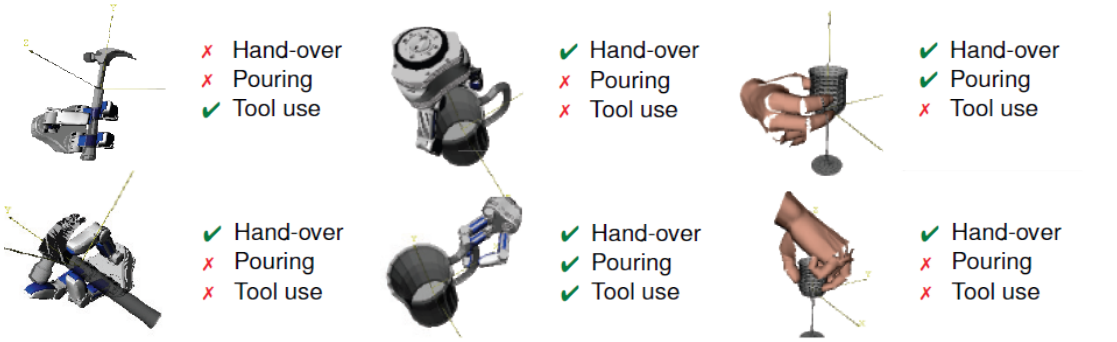


Figure 3: Exemplary grasps generated from box-based grasp generation system.

4. Haptic Exploration

Our approach on haptic exploration is based on dynamic potential fields for motion guidance of the fingers of a humanoid hand along the contours of an unknown object. During the exploration process oriented point sets from tactile contact information are acquired in terms of a 3D object model. Fig. 4 gives an overview on our tactile exploration module. We presented an initial version of this method earlier in [1]. Beyond this, we demonstrated concepts and preliminary results for applying the geometric object model to extract grasp affordances from the data [A5, A6]. The grasp affordances comprise grasping points of promising configurations which may be executed by a robot using parallel-grasps. For object recognition we have outlined our approach which relies on transforming the sparse and non-uniform pointset from tactile exploration to a model representation appropriate for 3D shape recognition methods known from computer vision.

We believe that the underlying 3D object representation of our concept is a major advantage as it provides a common basis for multimodal sensor fusion with a stereo vision system and other 3D sensors. As finger motion control during exploration is directly influenced from the current model state via the potential field, this approach becomes a promising starting point for developing visuo-haptic exploration strategies. We also believe that the proposed scheme is transferable to different manipulator and robot hand kinematics by adapting its parameters.

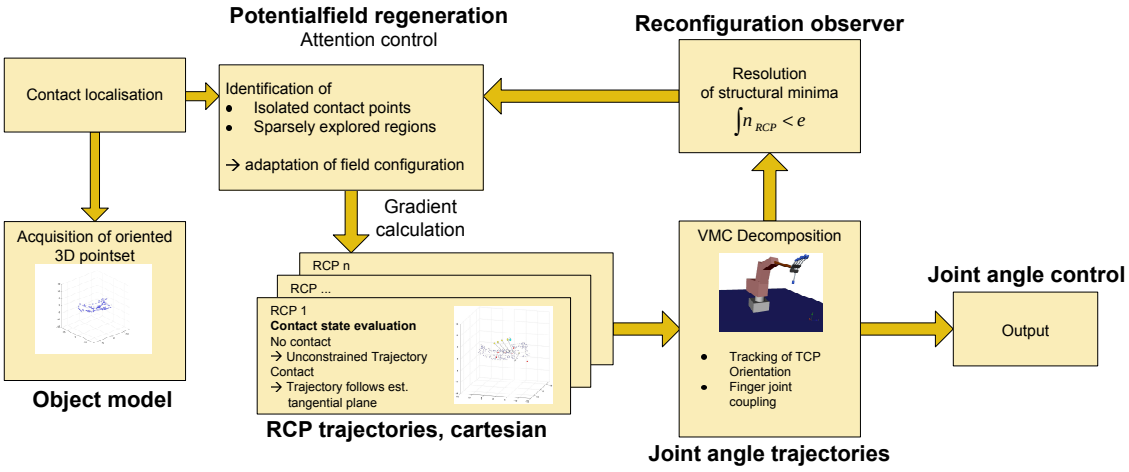


Figure 4: Tactile exploration scheme based on dynamic potential field.

Attached Papers

[A1] **Learning to Grasp Unknown Objects Based on 3D Edge Information**
L. Bodenhagen, D. Kraft, M. Popovic, E. Baseski, P. Eggenberger Hotz and N. Krüger
Presented and published on International Symposium on Computational Intelligence in Robotics and Automation 2009.

Abstract: In this work we refine an initial grasping behavior based on 3D edge information by learning. Based on a set of autonomously generated evaluated grasps and relations between the semi-global 3D edges, a prediction function is learned that computes a likelihood for the success of a grasp using either an offline or an online learning scheme. Both methods are implemented using a hybrid artificial neural network containing standard nodes with a sigmoid activation function and nodes with a radial basis function. We show that a significant performance improvement can be achieved.

[A2] **Learning of 2D Grasping Strategies from Box-Based 3D Object Approximations**
S. Geidenstam, K. Huebner, D. Banksell and D. Kragic
Presented and published on Robotics, Science, and Systems Conference 2009.

Abstract: In this paper, we bridge and extend the approaches of 3D shape approximation and 2D grasping strategies. We begin by applying a shape decomposition to an object, i.e. its extracted 3D point data, using a flexible hierarchy of minimum volume bounding boxes. From this representation, we use the projections of points onto each of the valid faces as a basis for finding planar grasps. These grasp hypotheses are evaluated using a set of 2D and 3D heuristic quality measures. Finally on this set of quality measures, we use a neural network to learn good grasps and therelevance of each quality measure for a good grasp. We test and evaluate the algorithm in the GraspIt! Simulator.

[A3] **Grasping Known Objects with Humanoid Robots: A Box-Based Approach**
K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic and R. Dillmann
Presented and published on 14th International Conference on Advanced Robotics 2009.

Abstract: Autonomous grasping of household objects is one of the major skills that an intelligent service robot necessarily has to provide in order to interact with the environment. In this paper, we propose a grasping strategy for known objects, comprising an off-line, box-based grasp generation technique on 3D shape representations. The complete system is able to robustly detect an object and estimate its pose, flexibly generate grasp hypotheses from the assigned model and perform such hypotheses using visual servoing. We will present experiments implemented on the humanoid platform ARMAR-III.

[A4] **Grasping by Parts: Robot Grasp Generation from 3D Box Primitives**
K. Huebner and D. Kragic
Unpublished internal Technical Report, Royal Insitute of Technology (KTH), Stockholm.

Abstract: One of the core challenges in the eld of robotics is to equip robots with the ability to intelligently interact with the world. To achieve this, a robot necessarily needs to perceive and interpret the environment in a proper way and understand the situations it is engaged in. The robot thus has to be able to gather and interpret the sensory information in new, unforeseen situations being provided some minimal knowledge in advance. For service robot applications, one of the key requirements is to be able to detect, recognize and manipulate objects, autonomously or in collaboration with humans and other robots. These capabilities should also include the generation of stable grasps to safely handle even objects unknown to the robot. We believe that the key to this ability is not to select a good grasp depending on the identication of an object (e.g. as a cup), but on its shape (e.g. as a composition of shape primitives). In this paper, we envelop our previous work on shape approximation by box primitives for the goal of simple and ecient grasping, and extend it with a deeper investigation of methods and robot experiments.

[A5] **Grasp Affordances from Multi-Fingered Tactile Exploration using Dynamic Potential Field**
A. Bierbaum, M. Rambow, T. Asfour and R. Dillmann
Accepted to IEEE/RAS International Conference on Humanoid Robots, Humanoids 2009.

Abstract: In this paper, we address the problem of tactile exploration and subsequent extraction of grasp hypotheses for unknown objects with a multi-fingered anthropomorphic robot hand. We present extensions on our tactile exploration strategy for unknown objects based on a dynamic potential field approach resulting in selective exploration in regions of interest. In the subsequent feature extraction, faces found in the object model are considered to generate grasp affordances. Candidate grasps are validated in a four stage filtering pipeline to eliminate impossible grasps. To evaluate our approach, experiments were carried out in a detailed physics simulation using models of the five-finger hand and the test objects.

[A6] **Dynamic Potential Fields for Dexterous Tactile Exploration**
A. Bierbaum, T. Asfour and R. Dillmann
Presented and published on 3rd International Workshop on Human Centered Robotic Systems 2009.

Abstract: Haptic exploration of unknown objects is of great importance for acquiring multimodal object representations, which enable a humanoid robot to autonomously execute grasping and manipulation tasks. In this paper we present our ongoing work on tactile object exploration with an anthropomorphic five-finger robot hand. In particular we present a method for guiding the hand along the surface of an unknown object to acquire a 3D object representation from tactile contact data. The proposed method is based on the dynamic potential fields which have originally been suggested in the context of mobile robot navigation. In addition we give first results on how to extract grasp affordances of unknown objects and how to perform object recognition based on the acquired 3D point sets.

References

[1] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. A Potential Field Approach to Dexterous Tactile Exploration. In *International Conference on Humanoid Robots*, 2008.

[2] 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, 2008.

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

[4] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger. A Strategy for Grasping Unknown Objects based on Co-Planarity and Colour Information. *Robotics and Autonomous Systems*, 2010.

[5] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, and J. Wikander. Demonstration-based Learning and Control for Automatic Grasping. *Intelligent Service Robotics*, 2:23–30, 2009.

Learning to Grasp Unknown Objects Based on 3D Edge Information

Leon Bodenhagen, Dirk Kraft, Mila Popović,
Emre Başeski, Peter Eggenberger Hotz and Norbert Krüger
Mærsk Mc-Kinney Møller Institute
University of Southern Denmark
Odense, Denmark
{lebo, kraft, mila, emre, eggen, norbert}@mmmi.sdu.dk

***Abstract*— In this work we refine an initial grasping behavior based on 3D edge information by learning. Based on a set of autonomously generated evaluated grasps and relations between the semi-global 3D edges, a prediction function is learned that computes a likelihood for the success of a grasp using either an offline or an online learning scheme. Both methods are implemented using a hybrid artificial neural network containing standard nodes with a sigmoid activation function and nodes with a radial basis function. We show that a significant performance improvement can be achieved.**

I. Introduction

Being able to grasp unknown objects is becoming a more and more important goal within emerging application areas e.g., service robotics which do not rely on strongly structured environments as they are available in industrial robotics. Furthermore, for many of these applications a system that is able to learn how to do this—and potentially adapt to changes later on—is preferable.

It has been shown that already rather high success rates of grasping can be achieved for unknown objects by defining grasps based on co-planar pairs of 3D contours ([1], [2]). In this work, we show that the success rate of this approach can be further increased by the introduction of learning. The resulting system is a grasping behavior that decides which grasp from a number of grasping options to perform, based on a success prediction.

For each scene, a large number of grasps become computed (see Fig. 4(b)) by using co-planar 3D contours as in [1]. We then learn an evaluation function which associates a success likelihood to each grasp by using a neural network. The learning is based on labeled data in terms of executed grasps that have been evaluated autonomously (i.e., the visual features the grasp has been derived from and its haptically measured success). We then apply a supervised learning scheme on that autonomously generated data. This is possible by making use of a highly robust robot-vision system with force-torque sensors, motion planning and collision detection. The system is also able to deal with critical and unforeseen situations such as collisions and non-successful grasps. After learning, the system is able to compute success likelihoods for potential grasps and hence a behavior which selects the most promising grasp that can be performed. The grasp selection can be learned offline and online and success rate can be increased from 42.0% to 51.1% in the offline learning case and from 42.0% to 47.9% in the online case.

Grasping in general, both with and without knowledge about the object to grasp, is an area where a lot of effort has been spent as grasping allows a robot to take control over an object, manipulate it, and in general to interact actively with the environment. Classically it has been assumed that an object model is available either as given (e.g. a CAD-model) or learned. A grasp can then be investigated and planned in detail. Computing optimal grasps—where optimal can mean different things, e.g., maximize different torques and forces that the gripper can apply to the object—based on surface models has been extensively treated in the literature (see [3] for an overview). In contrast Borst et al. showed in [4] that while finding the optimal grasp is hard, finding a good grasp is not difficult.

If no object model is available, grasps need to be planned based purely on sensor-information and heuristic knowledge. Apart from e.g. laser range or ultrasound, vision sensors are often used as they can provide significant amount of information about the environment and the object to grasp. The complexity of dealing with this information leads to very different approaches about how to compute a grasp, of which some are presented in the following. While using laser scanners, a common approach is to segment the laser data into point clouds and create practical grasping points (see e.g., [5]). A possible approach based on vision sensors is to perform a simple line scanning of the environment (see e.g. [6]). Another approach is to build up a dense 3D model of the environment by using a laser line in addition to the stereo camera, which simplifies the stereo matching problem (e.g. [7]). When such a dense description of the environment is available, the planning of a grasp becomes similar to the situation where the object is known. As in Saxena et al. [8], sparse stereo can also be used by triangulating different 2D visual features. The position of the grasp can then be defined by the reconstructed 3D points and the orientation of the gripper is determined by the intention not to collide with other objects and with the limitation of the robot having only five degrees of freedom.

Learning how to choose grasps and generalizing this information has been studied both in vision and robotics community (see e.g., [9], [10]) where previously performed grasping trials are used to predict the quality of future grasps. Similar to our approach, [11], [12] learn a function that allows to predict the outcome of a grasp based on

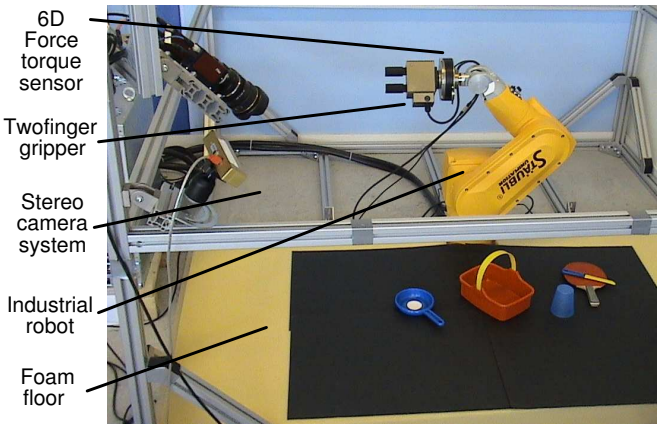


Fig. 1. Hardware setup.

two dimensional visual features, proprioceptive and features derived from these. On the other hand, the selection of the features induces a high amount of prior knowledge. This approach which focused on scenes containing a single planar object with a solid color was extended by Speth et al. in [13] to non-planar objects.

The most significant difference of our approach to the approaches mentioned above is that we do learning of grasping of unknown objects based on 3D contour relations instead of 3D features. Moreover, our system is able to act in complex environments with multiple objects since it does not require any specific prior object knowledge. A particular strength is also that we have a system, in which the training data is generated autonomously by haptic evaluation during exploration. This is in particular important in the context of online learning as done in this paper.

II. System

A. Robotic Setup

The hardware setup (see also Fig. 1) used for this work consists of a six-degree-of-freedom industrial robot arm (Stäubli RX60) with a force/torque (FT) sensor (Schunk FTACL 50-80) and a two-finger-parallel gripper (Schunk PG 70) attached. The FT sensor is used to detect collisions. Together with the foam floor, this permits graceful reactions to collision situations which might occur because of limited knowledge about the objects in the scene. In addition, a calibrated stereo camera system (Point Grey BumbleBee2) is mounted in a fixed position in the scene. The system also makes use of a path-planning module which allows it to verify the feasibility of grasps with respect to workspace constraints and 3D structure discovered by the vision system.

B. Early Cognitive Vision System and Grasp Hypothesis Definition

In this work, we make use of the visual representation delivered by an early cognitive vision system [14], [15], [16]. A calibrated stereo camera setup is used to create sparse 2D and 3D features, so-called *multi-modal primitives*,

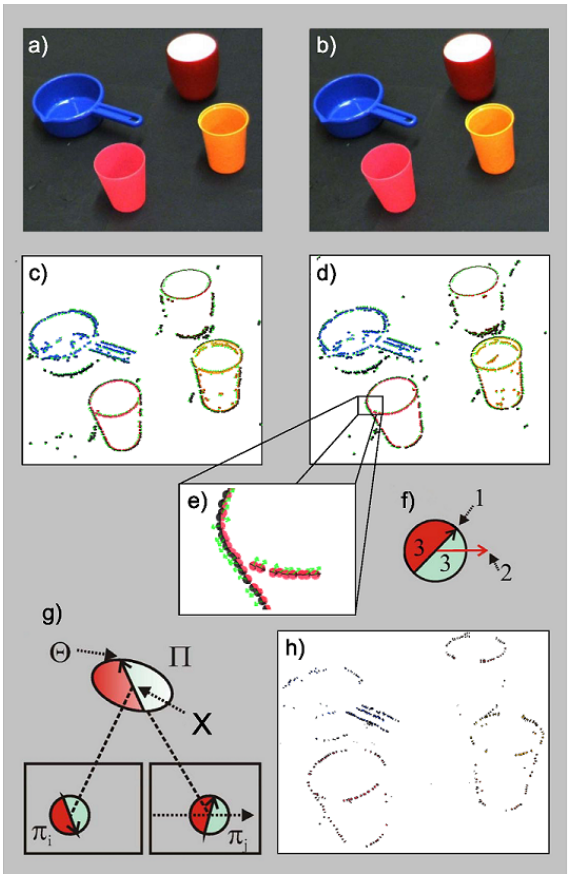


Fig. 2. The extraction of primitives. (a,b) show the original left and right image. (c,d) show the corresponding 2D primitives. (e,f) show a close up of the 2D primitives and their modalities: orientation (1), phase (2) and color (3). (g) illustrates the reconstruction of a 3D primitive using two 2D primitives. (h) shows the resulting 3D primitives.

along image contours. 2D features represent a small image patch in terms of position, orientation, phase, color and optical flow, denoted by $\pi = (\mathbf{x}, \theta, \phi, (\mathbf{c_l}, \mathbf{c_m}, \mathbf{c_r}), \mathbf{f})$ (see Fig. 2(f)). These are matched across two stereo views, and pairs of corresponding 2D features permit the reconstruction of a 3D equivalent encoded by the vector $\Pi = (\mathbf{X}, \Theta, \Phi, (\mathbf{C_l}, \mathbf{C_m}, \mathbf{C_r}))$.

The procedure to create the visual representation is illustrated in Fig. 2 on an example stereo image pair. Note that the resultant representation not only contains appearance-based (e.g., color and phase) but also geometrical information (i.e., 2D and 3D position and orientation).

2D and 3D primitives are organized into perceptual groups in 2D and 3D (called 2D and 3D *contours* in the following) based on good continuation in terms of geometry and appearance. To ease the mathematical usage of contours a parametric description is fitted to the primitives of a contour. Hereby positions can be determined continuously at the contour. In the following we will use the notation $C(u)$ for the 3D position on the contour at u, where u is in the interval $[0, 1]$. This means that $C(0.5)$ denotes the center of a contour.

The sparse and symbolic nature of the visual features allows for defining perceptual relations on them that express

relevant spatial relations in 2D and 3D (e.g., co-planarity, co-colority). This relational space is then used to trigger grasping and to learn the success likelihood of grasping given a certain constellation of contours. The perceptual relations used in this work are briefly described in Section IV.

Given two co-planar and co-color 3D contours, four different grasping actions have been defined in [1] for a two-finger gripper, as illustrated in Fig. 3. Note that, a grasp for a two-finger gripper is defined through a 3D location and two directions. While we initially based this definition on local features and their relations, [1] already uses contour relations to find more stable feature tuples.

In this work, in contrast to [1] entire contours are used to define the grasps, rather than their center primitives. This is necessary to preserve consistency between the grasping hypotheses and the learning problem as the latter utilizes relations defined on both contours used to define a grasp. Furthermore this leads to a more robust definition as semi-global rather than local entities are used [17]. The 3D position of a grasping hypothesis is determined using the center positions of the contours. The 3D orientation is determined using a common plane fitted to both contours, and the tangents of the contours at the position to be grasped. Fig. 4 shows an example of different grasping actions.

III. Behavior for Autonomous Exploration

For the generation and evaluation of a grasping hypothesis, an autonomous grasping agent has been implemented (an overall flow diagram is shown in Fig. 5). The evaluation is done in two steps. First the gripper is closed (or opened if it is an EGA2 grasp). If the gripper touches the object, the jaws will apply a force to the object. As the force the gripper applies is limited, the movement will stop. If the movement stops but the gripper is not fully closed (resp. opened) the grasp is assumed to be successful as the object must have prevented the gripper from closing/opening fully.

When the robot returns to the home position and the grasp has initially been detected to be successful, it is checked again whether it is successful or not. If the grasp is found to be unsuccessful, it is labeled as being unstable, as it is assumed that the object has been lost during the movement, hence the grasp was not sufficiently stable.

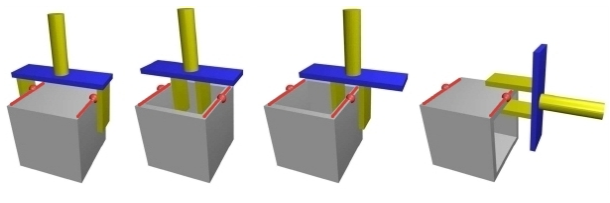


Fig. 3. Elementary grasping actions (EGAs). Red lines indicate 3D contours, red dots their center. EGA3 and EGA4 can each be defined with two different positions — one for each contour.

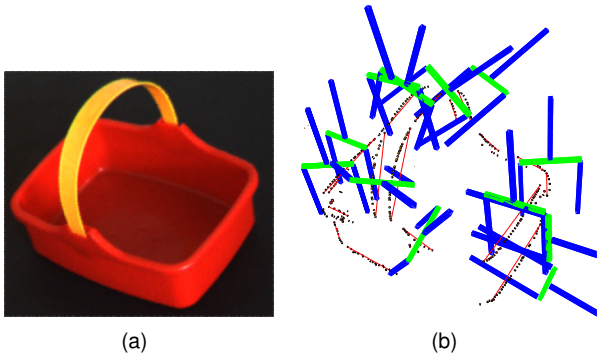


Fig. 4. Generation of multiple grasping hypotheses. (a) A sample object. (b) Grasping hypotheses generated for the sample object. Note that not all of them will lead to successful grasps.

IV. Features Used for Learning

To learn the success likelihood of grasping for a certain visual feature constellation, the visual and geometrical events that trigger the grasp are used as a feature vector β for the learning algorithm (see Section V for the details of the learning algorithm).

$$\beta = \{F_{par}, F_{cop}, F_{col}, F_{dist}, F_{cocol}\} \quad (1)$$

Since the grasping hypotheses are defined based on second order 3D contour relations, the feature vector contains relations between the contours that have been used for the definition of the grasp. In this section, we briefly discuss these visual and geometrical relations which are based on the visual representation presented in Section II-B.

Co-colority: Since contours are composed of image patches that have a left, right and middle color, for every contour a mean color is calculated for all sides. Note that the CIELAB[1] standard is used to encode colors. The co-colority between two contours is calculated as the color difference (CIE 1994, [18]) between the sides that are facing each other. The co-colority between contours C_A and C_B is defined as:

$$F_{cocol}(C_A, C_B) = dE\,(c_1, c_2) \quad (2)$$

where c_1 and c_2 are the colors of the contours and dE is the CIE 1994 color difference

To compute which sides of the contours are facing each other, 2D-projections of the contours are used. For each contour a line is defined (as illustrated in Fig. 6(a)) by its center-point $C(0.5)$ and the overall direction of the contour **u** given by direction of the greatest positional variance which is determined using principal component analysis. Further it is checked whether the center of the other contour is on the left or the right of this line and hereby which of the color values to use.

Co-planarity: The co-planarity between two contours is determined by combining their primitives to a set of pairs

[1]Color space defined by the International Commission on Illumination with the goal to approximate human color perception.

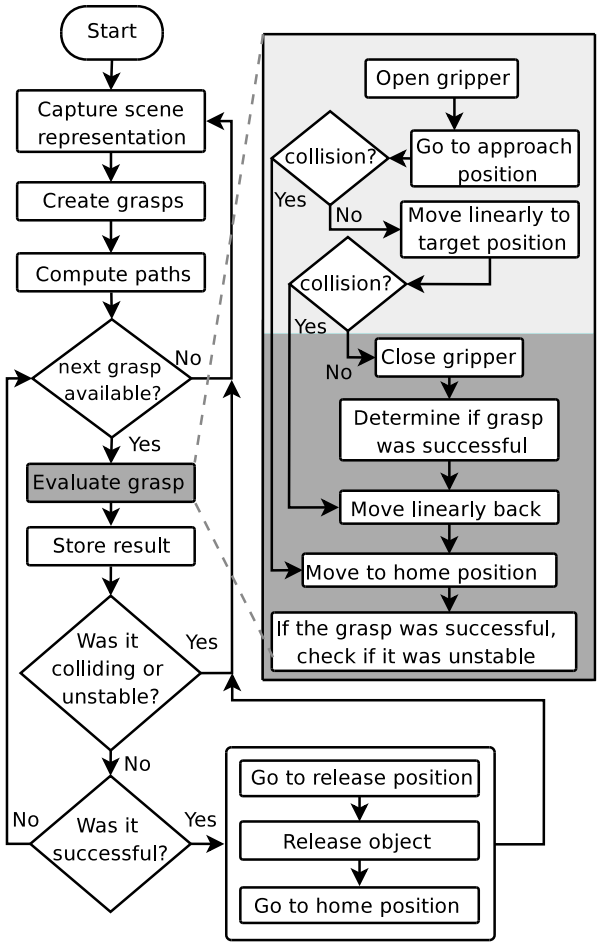


Fig. 5. Flow diagram for the process of creating and evaluating grasping hypotheses automatically. The bright part (top right) indicates the utilization of active collision detection.

P (as illustrated in Fig. 6(b)) and by computing the mean local co-planarity between all pairs.

$$F_{cop}(C_A, C_B) = \frac{1}{|P|} \sum_{s=1}^{|P|} cop\left(P\left(s\right)\right) \tag{3}$$

where $P(s)$ denotes the s'th pair, $|P|$ denotes the number of the pairs and $cop(P(s))$ denotes the local co-planarity between Π_i and Π_j. The local co-planarity between Π_i and Π_j is defined as the angle between n_i and n_j where n_i is the vector defined as $n_i = \Theta_i \times v$, Θ_i is the orientation of Π_i and $v = X_i - X_j$ (vector between the 3D locations of Π_i and Π_j),

Distance: The Euclidean distance between two contours is defined as the distance between their centers:

$$F_{dist}(C_A, C_B) = \|C_A(0.5) - C_B(0.5)\| \tag{4}$$

Parallelism: The parallelism between two contours C_A and C_B is measured as the projection of the normalized overall direction vector of C_A ($\mathbf{u}_A$) and C_B ($\mathbf{u}_B$).

$$F_{par}(C_A, C_B) = |\mathbf{u}_A \cdot \mathbf{u}_B| \tag{5}$$

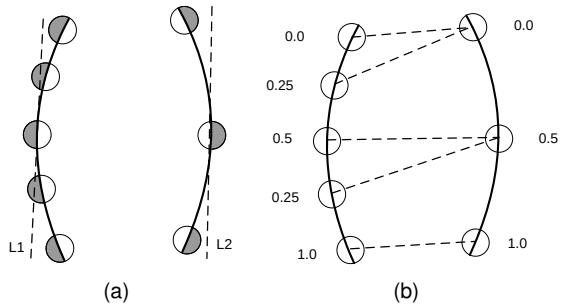


Fig. 6. **(a)** Whether the left or the right color of a contour is used for the co-colority computation is determined by the contour's position relative to the line which is defined by the center and the first principal component vector of the other contour. **(b)** Dashed lines indicate the pairing of primitives, used for relations that do not address the entire contours.

Collinearity: Collinearity is an important relation as collinear contours still can be co-planar and parallel. But for collinear contours it is no longer possible to define grasps properly, as the plane fitted to collinear contours is not valid and subsequently the orientation determined by the plane can be arbitrary. In order to determine if a pair of contours is collinear, the normalized overall direction vector of the contour combined with the positions of the contours are used:

$$\mathbf{v} = \frac{C_A(0.5) - C_B(0.5)}{\|C_A(0.5) - C_B(0.5)\|} \tag{6}$$

$$F_{col}(C_A, C_A) = \frac{|\mathbf{v} \cdot \mathbf{u}_A| + |\mathbf{v} \cdot \mathbf{u}_B|}{2} \tag{7}$$

V. Offline- and Online-Learning

Based on triplets, consisting of a feature-vector, a grasping hypothesis and the evaluation of the grasp, learning can be applied. The aim is to predict the success likelihood of an unevaluated hypothesis. Learning is realized using a neural network. The following sections describe the architecture of the network and how both offline and online learning are implemented.

A. Basic Structure of Neural Net

The network has been chosen to be a hybrid of a radial basis function network (RBFN) and a standard neural network. The benefit of RBFNs is that their Gaussian activation $h(x)$ is defined by a center position $\mathbf{c}$ in the feature space and a width in each dimension of the feature-space:

$$h(\mathbf{x}) = \exp(-1 \cdot (\mathbf{x} - \mathbf{c})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{c})) \tag{8}$$

where $\mathbf{S}$ is a diagonal matrix defining the N widths of the Gaussian function:

$$\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \cdots, \sigma_N\} \tag{9}$$

$$\mathbf{S} = diag(\boldsymbol{\sigma}) \tag{10}$$

Therefore a single RBFs affects the result only locally which allows subsequently to investigate where in the feature space something has been learned. Further individual RBF-nodes can be removed or altered without side effects, which is not possible in a standard neural network as the impact of the individual neuron is difficult or even impossible to

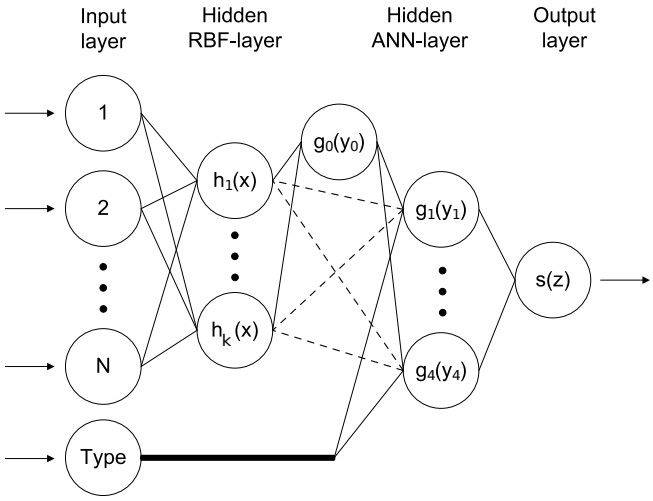


Fig. 7. Architecture of neural network. The input is given by an N-dimensional feature vector and the type of the corresponding hypothesis. The output reflects the likelihood for the grasping hypothesis to be successful. The 0'th neuron in the hidden ANN-layer provides a bias for the other four neurons. The dashed lines connecting them to the RBF-neurons represent the weights that are obtained during learning.

determine. The drawback of RBFs, is that they are located continuously in the feature space which makes it difficult to integrate discrete inputs, as for instance the type of the grasp which cannot be neglected. Therefore an additional layer has been defined (see Fig. 7) containing four standard neurons — one for each type. The type of the grasp presented to the network will then activate exactly one of these neurons. An alternative to this solution would be to independently train four individual networks. This requires the entire training set to be split, which has been considered to be worse than the fact that the RBFs might not be chosen optimally for each grasping type.

An additional neuron provides a bias to the other four neurons. Without this bias, the response of the network would depend highly on the fact whether the current input causes a significant activation of one of the RBFs. If the outputs of the all RBFs are close to zero, the estimate of the overall network would be misleading. The weight for this additional neuron depends on the overall grasping strategy. If it is desired to select grasps with a high likelihood for being successful the bias can force the output of the neural close to zero when no RBF has a significant response. If on the hand it is desired to explore the feature space further, the bias enables the system to prefer grasping hypotheses with novel feature vectors.

Different learning methods are used to determine the RBFs and the weights connecting them to the standard neurons.

B. Offline Learning

Apart from defining the individual RBFs it is also necessary to determine how many of them are actually needed. As each RBF corresponds to a limited area in the space spanned by the features, it is natural to locate an RBF in the feature space where grasps have occurred. Considering a set of recorded grasps T, which form a training set for an offline learning algorithm, the RBFs should be located at regions where multiple grasps occurred. These positions can easily be determined by applying a clustering algorithm like k-means on T. The k-means-algorithm detects k clusters in T, where k needs to be defined beforehand. The k-means implementation from [19] was used in this work.

Several approaches exist to estimate k. As it cannot be predicted if the training set is nicely clustered, a more intuitive approach proposed by [20] is used.

The key idea is to create an artificial uniformly distributed data set and to compare the performance of the clustering using k clusters on the training set with respect to the artificial data set (see [20]). The evaluation is based on distance between the data points in a cluster. If a 'required' cluster is added, the distances in the real dataset are expected to decrease more than the ones in the artificial data set. When a superfluous cluster is added, the reduction of the distances will stagnate (compared to the artificial data set). Computing this distance for every appropriate value allows us to determine the optimal value for k.

The positions of the clusters are used as the centers for the RBFs. The widths are determined by the standard deviation of the data points associated to each cluster. The determined deviations are scaled by a constant α which is chosen manually within the range $[0; 1]$.

Now only the output weights need to be determined. For this purpose a method from Orr (see [21]) is adapted which is based on a global optimization of the weights with respect to the training set. First the training set T is divided into four subsets $T_1, \cdots, T_4$, each containing the evaluated grasps of one of the four types (e.g. EGA1). For each grasp the features vector $\mathbf{x}$ is used as input for the radial basis functions $h_i(\mathbf{x})$ with $i = 1, \cdots, k$. The responses are stored in a matrix:

$$\mathbf{H}_s = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_k(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_k(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_t) & h_2(\mathbf{x}_t) & \cdots & h_k(\mathbf{x}_t) \end{bmatrix} \tag{11}$$

where $t = |T_s|$ and $s = 1, \cdots, 4$.

As the training set has been split based on the type of the grasps, only one neuron in the second hidden layer needs to be considered for each subset T_s. The input y_s, which this neuron receives, is the sum of the responses of the RBFs, weighted by the k weights $\mathbf{w}_s$. For the set T_s y_s can be written as a vector:

$$\mathbf{y}_s = \mathbf{H}_s \mathbf{w}_s \tag{12}$$

As the outcomes of the grasps are known, $\mathbf{y}_s$ can be defined based on these and then be used to determine appropriate weights as described below. Usually $g(y)$ is chosen to be a sigmoid activation function $\phi(y)$, as it has the advantages of being smooth, bounded to the range $[0; 1]$ and easily differentiable:

$$\phi(y) = \frac{1}{1 + \exp(-y)} \tag{13}$$

For the determination of $\mathbf{w}_s$, it is advantageous to consider a linear activation function that only computes the weighted sum as this allows to apply linear least squares. For this purpose the mean success ratio of all grasps associated to a cluster is computed and the value of y which leads to an output of $\phi(y)$ equaling the success ratio is determined. Each entry of $\mathbf{y}_s$ is set to the prior computed value. The weights can then be found using the pseudo inverse:

$$\mathbf{w}_s = \left(\mathbf{H}_s^T \mathbf{H}_s\right)^{-1} \mathbf{H}_s^T \mathbf{y}_s \qquad s = 1, \cdots, 4 \tag{14}$$

The benefit of this method is that it provides a solution for the weights that allows for an optimal estimate of the likelihoods for being successful of the individual grasps, based on the linear activation function. It is on the other hand important to remember that the maximum value of k is limited by the size of the individual sets T_s.

C. Online Learning

The online learning approach uses the same architecture as the offline learning method presented in section V-B. In contrast to offline learning it does not access the entire data set at once, but uses the instances iteratively. Therefore no global optimization can be used, which in general leads to a performance that is worse than with offline learning.

The online learning algorithm has three different mechanisms which can be applied:

1) **Add** an RBF when the current input is not recognized.
2) **Adapt** to current input.
3) **Remove** an RBF.

Similarities with Self-Organizing Maps (SOMs), or more specifically growing SOMs, might become notable. The concept of SOMs has been used to some extent in order to define mechanisms that are considered useful for the online learning strategy in this work.

1) Adding an RBF: When the input is not recognized, a new RBF is added. Whether an input is recognized or not is determined by the overall response of the RBFN — if it is below a threshold th_{ins}, the input point is assumed not to be recognized. When the learning is initiated with an empty RBFN, the activation for any input will be zero, and an RBF will be added with a center positioned at the location of first input point.

2) Adaption: If an input is recognized, the RBF with the highest activation will adapt to it if its activation is larger than the threshold th_{adapt}. The adaption consists of multiple steps: the position and the width will be adjusted and the output weight will be updated. When an RBF adapts to an input point, its current center $\mathbf{c}$ will be updated to be the weighted mean of $\mathbf{c}$ and the current input point $\mathbf{x}$:

$$\Delta\mathbf{c} = \eta_p(\mathbf{x} - \mathbf{c}) \tag{15}$$

where $\eta_p \in [0; 1]$ is the learning rate controlling the impact of the individual data point. The center will then be updated as $\mathbf{c} = \mathbf{c} + \Delta\mathbf{c}$.

The width $\boldsymbol{\sigma}$ is updated with the aim to make the RBF "cover" the current point. This is a rather intuitive approach, where input points close to the center will reduce the width while input points close to the borders increase it. When multiple input points have been evaluated, the widths are expected to converge to an appropriate value. The RBF should not flatten out to cover the whole feature space. A new width, $\boldsymbol{\sigma}_{goal}$ is defined, that will cause the RBF to have a response of e.g. $r_{goal} = 0.5$ at the current input. The width is then updated as:

$$\Delta\boldsymbol{\sigma} = \eta_\sigma(\boldsymbol{\sigma}_{goal} - \boldsymbol{\sigma}) \tag{16}$$

where $\eta_\sigma \in [0; 1]$ is the learning rate. These widths are computed individually for each of the N dimensions:

$$\exp\left(-1 \cdot \frac{(c_i - x_i)^2}{\sigma_{goal}^i}\right) = r_{goal} \tag{17}$$

$$\Downarrow$$

$$\sigma_{goal}^i = \frac{(c_i - x_i)^2}{-\log(r_{goal})} \tag{18}$$

$$\boldsymbol{\sigma}_{goal} = \{\sigma_{goal}^1, \cdots, \sigma_{goal}^N\} \tag{19}$$

where $i = 1, \cdots, N$, c_i and x_i are the i'th dimension of the center position resp. the input point while σ_{goal}^i denotes i'th dimension of σ_{goal}^i and $r_{goal} \in]0; 1[$.

Although the outlined updating of the widths is straight forward, it suffers from some issues. One of them is the fact that the computation of the new width is non-linear. Points close to the center will have a significantly larger impact than those close to the border — especially as not all input points are desired to cause an adaption. Only points close to the RBF, thus causing an activation higher than some threshold th_σ^l will cause an adaption. If this issue would not be handled, input points close to the center will dominate the updating of the width and cause it only to shrink.

A possible solution is to apply a second threshold th_σ^u as an upper bound for the activation. Input points close to the center, causing a higher activation than th_σ^u, will then be ignored. As the input point is multidimensional and $\boldsymbol{\sigma}_{goal}$ is defined individually on each dimension it is important to apply the threshold on each dimension as well. The value of th_σ^u depends mainly on th_σ^l. The value of th_σ^u is chosen to ensure that input points close to the center have the same net effect as those further away which allows their effects to cancel out. th_σ^u is determined numerically to fulfill:

$$\int_{th_\sigma^l}^{r_{goal}} \Delta\boldsymbol{\sigma} d\boldsymbol{\sigma} = \int_{r_{goal}}^{th_\sigma^u} \Delta\boldsymbol{\sigma} d\boldsymbol{\sigma} \tag{20}$$

The impact of the distribution of the input points has been considered to be negligible as it within reasonable bounds has significantly less impact than the choice th_σ^l. As an alternative to this solution a history can be associated to each RBF and reflect the distribution of the input points. The width can then be adjusted according to the distribution. As only rather short learning phases have been addressed until now, the first solution was found to be sufficient.

The two adaption mechanisms outlined above cause the center of the winning RBF to wander towards the current input point and the width of the RBF to shrink/grow. The only remaining elements to update are the output weights. Only one of the four sets of weights must be adjusted, the one corresponding to the type of the grasp that caused the adaption. As the grasp is considered either successful or not, the weights will either be increased or reduced.

As all RBFs have a response, also all RBFs will contribute to the output of the network. Therefore it is intuitive to update all weights according to their impact in the current estimate. For this purpose the back-propagation mechanism (see e.g. [22]) has been considered as a useful standard tool. The basic concept is to define an error function $E(\mathbf{x})$ and to differentiate this function with respect to the parameter to update. The derivative is then used to adjust the parameter. An appropriate error function is the squared difference between the desired output d_{out}, and the actual output $g(y_s)$ where $s \in \{1, \cdots, 4\}$, depending on the type:

$$y_s = \sum_{i=1}^{M} w_i^s h_i(\mathbf{x}) \tag{21}$$

$$E(\mathbf{x}) = \frac{1}{2}\left(d_{out} - g\left(y_s\right)\right)^2 \tag{22}$$

$$\begin{aligned} \Delta w_i^s &= -\eta_{out}\frac{\delta E(\mathbf{x})}{\delta w_i^s} \\ &= \eta_{out}(d_{out} - g(y_s))\frac{\delta g(y_s)}{\delta w_i^s} \end{aligned} \tag{23}$$

The method to update the weights, defined in (23) where η_{out} is the learning rate, is a common method for updating weights using backpropagation.

3) Removal of RBFs: The need for a mechanism to remove RBFs arises from the fact that the adding-mechanism will add new RBFs whenever an input is not recognized. Thus, if not counteracted, the network will keep growing until the entire search space is covered. A significant amount of the RBFs will most likely not be of any value, as the points that triggered their existence might be unique, or induced by noisy data. Further, a crowded network becomes computationally inefficient and a potential interpretation of *what* has been learned becomes also difficult or even impossible.

A simple strategy for removing RBFs is to remove all those who have a winning ratio lower than some threshold th_{del}. The winning ratio is the ratio between the number of input vectors presented to the RBF and how often this RBF was the one with the highest activation. The value of th_{del} needs to be chosen carefully as it limits the size of the network to $\frac{1}{th_{del}}$. If the distribution of the input points causes some RBFs to win more often than others, the size of the network will be reduced further.

The limited size of the data sets used for online learning in this paper is considered to be a potential issue. This has been partly compensated by careful manual choice of learning parameters:

$$th_{adapt} = 0.37, th_{ins} = 0.15, th_{del} = 0.0$$
$$\eta_\sigma = 0.125, \eta_c = 0.125, \eta_{out} = 0.29$$

Fig. 8. Objects used for the evaluation. Cylindrical objects in the top row, non-cylindrical in the bottom row.

Especially the η_{out} is chosen to be large as the weights are updated individually for each type. For the tests, the removal-mechanism was disabled. As the manual selection of these parameters might not result in an optimal neural network, differential evolution (DE) has been considered to determine an appropriate choice of parameters. Currently the usage of DE was experienced to be very time consuming. Therefore manually selected parameters have been used, especially as any change to the system induces the risk that the parameters determined using DE are not optimal any longer.

VI. Evaluation

In order to test the learning mechanism outlined above, two sets of grasping hypotheses have been recorded. The first set (S_1) is based on partly cylindrical objects (Fig. 8 top row) whereas the second set (S_2) is based on non-cylindrical objects (Fig. 8 bottom row). Each set contains 256 grasp attempts and each attempt is labeled as successful or unsuccessful based on the evaluation discussed in Section III. Note that in this data set creation phase, the grasping hypotheses are chosen without any constraint, as it is done in [1], and a success rate of 38.3% was obtained for cylindrical and 45.7% for non-cylindrical objects.

The effect of learning has been tested by dividing a set of grasps (either S_1, S_2 or a combination of both) into a training and a test set. The test set was ensured to contain different objects than the training set. The training set has been used to train the neural network using offline learning and subsequently the 20% of the grasps in the test set which received the highest estimates have been selected. This procedure has been repeated 500 times and the mean success rates of the selections are shown in Table I.

A similar test has been done using online learning. In this case, the hidden RBF layer of the neural network started empty and became build up while all grasps of the training set are given as input one by one. Once the RBF layer is created by using the training data, the evaluation was done with the test data. Note that, the procedure was repeated 500 times as well and since the order of the grasps in the training set has an influence, they are reshuffled at each iteration. The mean success rates of the grasps selected from the test set are listed in Table II.

Learning Set	Test Set		
	Cylindrical	Non-cylindrical	Combined
Cylindrical	57.6%	54.6%	55.9%
Non-cylindrical	33.3%	51.3%	43.1%
Combined	57.5%	45.7%	51.1%
Without Learning	38.3%	45.7%	42.0%

TABLE I

SUCCESS RATIOS ACHIEVED WITH OFFLINE LEARNING FOR DIFFERENT COMBINATIONS OF TRAINING AND TEST SETS.

Learning Set	Test Set		
	Cylindrical	Non-cylindrical	Combined
Cylindrical	44.4%	52.7%	48.9%
Non-cylindrical	40.1%	47.8%	44.3%
Combined	43.0%	52.1%	47.9%
Without Learning	38.3%	45.7%	42.0%

TABLE II

SUCCESS RATIOS ACHIEVED WITH ONLINE LEARNING FOR DIFFERENT COMBINATIONS OF TRAINING AND TEST SETS.

For the results of online learning in the case when only S_1 was used as training set, it is conspicuous that high success rates where achieved when grasps from S_2 were present in the test set. The two possible reasons that can potentially explain this are: (1) generalization from S_1 to S_2 was possible, and (2) S_2 already achieved higher success ratios initially.

The usage of both online and offline learning creates an improvement of the overall success rate for all combinations of training and test sets. As expected, offline learning in general yields better results.

VII. CONCLUSIONS

We have presented a system to autonomously create and evaluate grasping hypotheses. The integration of learning, both offline and online, enabled us to increase the overall success ratio of grasps by predicting their outcome. Especially the integration of online learning allows to define more complex grasping strategies, addressing the exploration of the feature space. In the future it is considered to expand the feature set, in particular by a feature reflecting the curvature of contours.

ACKNOWLEDGMENTS

This work was supported by the EU project PACO-PLUS (IST-FP6-IP-027657).

REFERENCES

[1] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, and N. Krüger, "An adaptive strategy for grasping unknown objects based on co-planarity and colour information," vol. (submitted), 2009.

[2] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger, "Early reactive grasping with second order 3d feature relations," in *Recent Progress in Robotics: Viable Robotic Service to Human*. Springer Berlin / Heidelberg, 2008, pp. 91–105.

[3] R. Suárez, M. Roa, and J. Cornella, "Grasp quality measures," Institut d'Organització i Control de Sistemes Industrials, Universitat Politècnica de Catalunya., Tech. Rep., 2006.

[4] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the Dice by Dicing the Grasp," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003.

[5] M. Richtsfeld and M. Vincze, "Robotic grasping from a single view," in *18th International Workshop on Robotics in Alpe-Adria-Danube Region*, 2009.

[6] K. I. Easton and A. Martinoli, "Efficiency and optimization of explicit and implicit communication schemes in collaborative robotics experiments." in *2002 IEEE/RSJ international conference on intelligent robots and systems*, 2002.

[7] G. Taylor and L. Kleeman, "Grasping unknown objects with a humanoid Robot," *Proceedings 2002 Australasian Conference on Robotics and Automation*, pp. 191–196, 2002.

[8] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *International Journal of Robotics Research (IJRR)*, 2008.

[9] I. Kamon, T. Flash, and S. Edelman, "Learning to grasp using visual information," in *in Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1994, pp. 2470–2476.

[10] J. Zhang and B. Rössler, "Self-valuing learning and generalization with application in visually guided grasping of complex objects," *Robotics and Autonomous Systems*, vol. 47, no. 2–3, pp. 117–127, 2004, robot Learning from Demonstration.

[11] E. Chinellato, A. Morales, R. B. Fischer, and A. P. d. Pobil, "Visual quality measures for characterizing planar robot grasps," in *IEEE transactions on systems, man and cybernetics*, 2005.

[12] A. Morales, E. Chinellato, H. Fagg, and A. del Pobil, "Using experience for assessing grasp reliability," *International Journal of Humanoid Robotics*, vol. 1(4), pp. 671–691, December 2004.

[13] J. Speth, A. Morales, and P. Sanz, "Vision-based grasp planning of 3d objects by extending 2d contour based algorithms," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008, pp. 2240–2245.

[14] N. Krüger, M. Lappe, and F. Wörgötter, "Biologically Motivated Multi-modal Processing of Visual Primitives," *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, vol. 1, no. 5, pp. 417–428, 2004.

[15] N. Pugeault, F. Wörgötter, and N. Krüger, "Accumulated Visual Representation for Cognitive Vision," *In Proceedings of the British Machine Vision Conference (BMVC)*, 2008.

[16] N. Pugeault, "Early cognitive vision: Feedback mechanisms for the disambiguation of early visual representation," Ph.D. dissertation, Informatics Institute, University of Göttingen, 2008.

[17] E. Başeski, L. Bodenhagen, N. Pugeault, S. Kalkan, J. Piater, and N. Krüger, "Using 3d contours and their relations for cognitive vision and robotics," in *Proceedings of the 24th International Symposium on Computer and Information Sciences (ISCIS 2009), Special Session on Cognitive Cybernetics and Brain Modeling*, 2009.

[18] R. Hunt, *Measuring Colour. 3rd edition*. Fountain Press, Kingston-upon-Thames, 1998.

[19] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, Jul 2002.

[20] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistics," *Royal Statistical Society*, vol. 63, Part 2, pp. 411–423, 2001.

[21] M. Orr, "Introduction to radial basis function networks," Institute for Adaptive and Neural Computation, Devision of Informatics, Edinburgh University, Tech. Rep., April 1996.

[22] G. F. Luger, *Artificial Intelligence - Structures and Strategies for Complex Problem Solving*, 5th ed. Addison Wesley, 2005.

Learning of 2D Grasping Strategies from Box-Based 3D Object Approximations

Sebastian Geidenstam, Kai Huebner, Daniel Banksell and Danica Kragic
Computer Vision & Active Perception Lab
KTH – Royal Institute of Technology, Stockholm, Sweden
Email: {sebbeg,khubner,banksell,danik}@kth.se

***Abstract*— In this paper, we bridge and extend the approaches of 3D shape approximation and 2D grasping strategies. We begin by applying a shape decomposition to an object, i.e. its extracted 3D point data, using a flexible hierarchy of minimum volume bounding boxes. From this representation, we use the projections of points onto each of the valid faces as a basis for finding planar grasps. These grasp hypotheses are evaluated using a set of 2D and 3D heuristic quality measures. Finally on this set of quality measures, we use a neural network to learn good grasps and the relevance of each quality measure for a good grasp. We test and evaluate the algorithm in the GraspIt! simulator.**

I. Introduction

In the field of intelligent grasping and manipulation, a robot may recognize an object first and then reference an internal object model. For unknown objects, however, it needs to evaluate from data it can collect on the spot. How to grasp a novel object is an ongoing field of research. Difficulties in this area include (i) the high dimensionality of the problem, (ii) incomplete information about the environment and the objects to be grasped, and also (iii) generalizable measures of quality for a planned grasp.

Since contacts and forces of the fingers on an object's surface make up a grasp, it is very important to have good information both about the hand and the object to be grasped. Both hand and object constraints together with the constraints for the task to be performed need to be considered [1]. Though there is interesting work on producing grasp hypotheses from 2D image features only, e.g. [2, 3], most techniques rely on 3D data. Due to the complexity of the task, much work has been done for simplifications of 3D shape, such as planar [4] or 3D-contour-based [5] representations. Other approaches involve modelling an object perfectly, i.e. known a-priori, or with high-level shape primitives, such as the use of grasp pre-shapes or Eigengrasps [6, 7, 8]. One work that uses high-level shape primitives, and is similar to ours in terms of learning, but by using an SVM approach, is [9]. Another approach to learning from 2D grasp qualities, using neural networks and genetic algorithms, is presented in [10].

This paper builds on the work of Huebner *et al*. [11, 12], which uses a hierarchy of minimum volume bounding boxes to approximate an object from a set of 3D points delivered by an arbitrary 3D sensor, e.g. laser scanners or stereo camera setups. Grasping is then done by approaching each face of a box until contact, backing up, and then grasping the object. What this work lacked however, was a way to explicitly place the fingers of the hand and to choose the best configuration of the hand. Learning to predict successful grasps was done only with raw data from the projections of points inside a box onto the face to be grasped. Secondly, our work makes use of an algorithm for finding and predicting the success of a grasp, but for planar objects, as proposed by Morales *et al*. [4, 13]. The approach uses 2D image analysis to find contact point configurations that are valid given specific kinematic hand constraints. From the geometrical properties of an object, it then calculates a set of quality measures that can later be used for learning to predict the success of found grasp hypotheses. The limitations of this work lie mainly in the fact that really 'planar' objects and representations are discussed in which information about 3D shape is discarded.

In this paper, we bridge and extend these two methods to enable 2D grasping strategies for 3D object representations.

II. 3D Box Approximation

We will shortly revisit the pre-computation of approximating a 3D point cloud by a constellation of minimum volume bounding boxes (MVBBs). The fit-and-split approach starts with fitting a root bounding box and estimating a best split by using the 2D projections of the enclosed points to each of the box surfaces. Depending on a volume gain parameter t, two child boxes might be produced and then be tested for splitting. To provide an insight to this algorithm as a base for the experiments in this paper, the two core algorithms have been sketched in Fig. 1. For more details and examples, we refer to Huebner *et al*. [11]. However, it is important to note that in that work (i) 2D projections have been used to estimate a split and (ii) only edge-parallel planar splits have been tested.

From these constraints, three main problems were evident relating to the original split estimation. These problems are outlined as follows.

1) Splitting of non-convex regions, e.g. u-shapes: As shown in [11], the presented algorithm will not do any splitting in case of u-shaped 2D projections. This is due to the fact that it uses upper bounds and area minimization, which are constant in such cases. This means that a split does not result in a substantial change in the area of a region. A solution for this problem remains a challenge [11], especially when sparse and noisy data is provided. For 3D data from real vision systems or laser scanners, such distortions are unavoidable, in part because of occlusion or sensor inaccuracies. Thus,

Algorithm II.1: BOXAPPROXIMATE($points^{3D}$)

$box \leftarrow findBoundingBox(points^{3D})$
$faces \leftarrow nonOppositeFaces(box)$
$(p, q) \leftarrow split(\text{FINDBESTSPLIT}(faces, points^{3D}))$
if $(percentualVolume(p + q, box) < t)$
  then { BOXAPPROXIMATE(p); BOXAPPROXIMATE(q)
  else return (box)

Algorithm II.2: FINDBESTSPLIT($faces, points^{3D}$)

for $i \leftarrow 1$ **to** 3
do {
  $p^{2D} \leftarrow project(points^{3D}, faces[i])$
  for $x \leftarrow 1$ **to** $width(faces[i])$
  do {
    $(p1, p2) \leftarrow verticalSplit(p^{2D}, x)$
    $a1 \leftarrow boundArea^{2D}(p1)$
    $a2 \leftarrow boundArea^{2D}(p2)$
    if $(a1 + a2 < minArea)$
      then { $minArea \leftarrow (a1 + a2)$; $bestSplit \leftarrow (i, x)$
  for $y \leftarrow 1$ **to** $height(faces[i])$
  do {
    $(p1, p2) \leftarrow horizontalSplit(p^{2D}, y)$
    $a1 \leftarrow boundArea^{2D}(p1)$
    $a2 \leftarrow boundArea^{2D}(p2)$
    if $(a1 + a2 < minArea)$
      then { $minArea \leftarrow (a1 + a2)$; $bestSplit \leftarrow (i, y)$
return $(bestSplit)$

Fig. 1. Pseudocode (original algorithm): a point set and its bounding box, respectively, are recursively split (II.1). A good split was estimated through analysis of 2D splits of the projected points onto each of the box faces (II.2).

how to distinguish between a real non-convex object region and just incompleteness of the data becomes a critical issue. The models used in [11] were ideal models, extracted from simulated 3D mesh data. As it is our aim to evaluate our algorithm also on real sensory data, we can not generally assume such ideal conditions.

2) Splitting along non-edge-parallel directions: The minimum volume box fitting approach naturally fits extensions of the shape into corners of a box, as this keeps the box smaller. The handle of a cup, for example, will fit best diagonally into one of the box corners. However, such diagonal structures in particular can rarely be cut parallel to one of the box edges as proposed in the previous algorithm.

3) Sensitivity to noise: The box decomposition's robustness showed the splitting to be very sensitive to noise. This is not a main issue in terms of single box or face grasping in general, since any constellation of boxes will produce grasp hypotheses. However, if one would like to take into account and learn from a whole constellation of boxes, then robustness and repeatability are necessary.

A. Improved Split Algorithm using 2D Convex Hulls

For the experiments presented in this paper, we have therefore implemented a new algorithm based on convex hulls. The new algorithm replaces II.2, solving the above mentioned issues, and in addition producing much more confident splitting

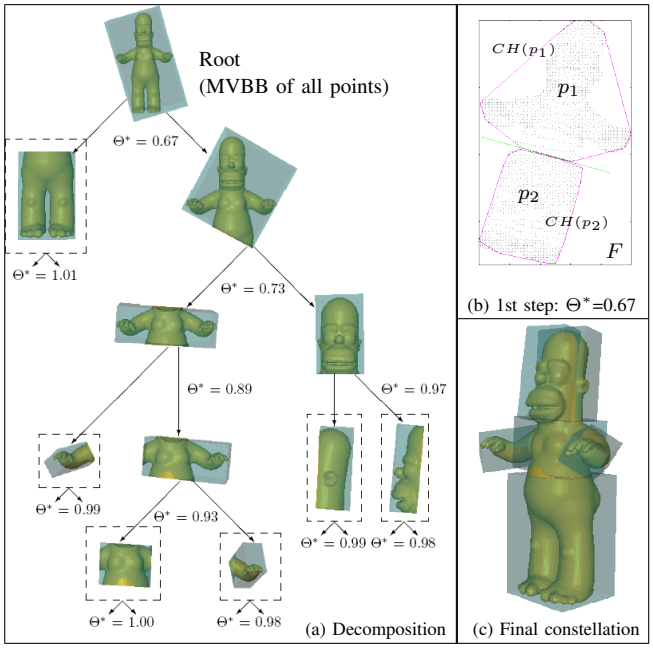


Fig. 2. (a) Example of a decomposition hierarchy, using a gain parameter of t=0.98. With $\Theta^* < t$, a valid cut is detected, as presented for the first step in (b). Otherwise, the box is a leaf box (*dashed*), i.e. a part of the final constellation which is plotted in (c).

results. For efficiently computing convex hulls on a set of 2D points p, like our projections (see Fig. 2), we use a monotone chain algorithm [14]. Starting from the convex hull $CH(p)$ of the whole projection set p, we select those segments of the hull that exceed a given threshold in length. We thereby assume that those segments either span a non-convex region of the outer contour of the data, or that they represent a very straight edge. On these segments, we interpolate a number of sample points. Between each pair of points on each pair of segments, we simulate a cut that splits the point set p into two subsets p_1 and p_2. The two segment points that minimize,

$$\Theta = [A(CH(p_1)) + A(CH(p_2))]/F, \quad (1)$$

where A is the area function for a convex hull and F the overall rectangular area of the face (see Fig. 2b), define our best split. An example of such a decomposition tree produced with the new hull algorithm is presented in Fig. 2.

B. Evaluation

To be able to make a large scale test of the box decompositions stability, an algorithm was developed that estimates if two box decompositions are similar or not. First, the algorithm summarizes the total volume V_i of all boxes which a decomposition i is composed of. Secondly, it calculates the Euclidean distances between the centers of all pairs of leaf boxes and summarizes them as D_i. In order to determine if two compositions i, j are similar, the differences in overall volume and distance measures between the decompositions are simply compared with empirically found thresholds:

if $|D_i - D_j| \leq 0.1 \wedge |V_i - V_j| \leq 0.9$ **then** $similar(i, j)$. (2)

TABLE I

PERCENTAGE OF DECOMPOSITIONS WITH SIMILAR COUNTERPARTS. 19 NOISE LEVELS AND 14 LEVELS OF POINT REMOVAL WERE USED.

3D	Bunny	Car	Cup	Duck	Goblet	Goose	Heart	Homer	Horse	Human	Mug	PaperCup	Pen	Pillow	Radio	Squirrel	ToyDog
Old	78,62	100,00	77,78	61,59	21,21	24,28	100,00	55,07	53,99	19,93	80,43	100,00	91,67	91,67	84,62	59,42	27,54
New	94,00	100,00	97,78	91,67	22,77	46,00	100,00	66,67	30,33	58,00	62,67	74,24	92,00	100,00	100,00	75,00	28,33

To test the stability of the box decomposition algorithms, we simulated 17 different object models and modified them: 19 different levels of close proximity noise and 14 different levels of point removal were used for the experiment to let modified point clouds emerge from each original object point cloud. Both algorithms were then executed (t=0.9) on each of those point clouds before comparing the resulting box decompositions of each unmodified with its modified models. The results presented in Table I show that the previous algorithm is quite sensitive to noise. However, simpler objects like Car or Pen gave very good results. This is mainly because they all produced only one box due to their compact shape. On the other hand, more complex models like the toydog or the human model gave quite poor results. We note that the bound-based algorithm tends to produce a single large box enveloping the whole object also in such cases. This raises the similarity rate significantly, but is not preferred in our application.

The new hull-based algorithm produces much better approximation for the objects, very few single-box decompositions, and a significantly better similarity rate. The models that produced single-box decompositions with the bound-based algorithm produce worse values in some cases. This is caused by better approximations with multiple boxes that are more sensitive to the comparison than a single-box-to-single-box comparison. Since we prefer multi-box decompositions which give better object approximation, this is a good improvement, while the new algorithm is considerably less affected by noise.

The old and the new techniques are also compared to each other in Fig. 3 according to robustness to the change of the gain parameter t (for t, see II.1 and Fig. 2), e.g. the duck model decomposition repeatedly shows the same constellation. Another visible effect is that the decompositions seem more intuitive, e.g. in case of the cup handle.

III. 2D GRASP HYPOTHESES

In this paper, we are concerned with finding 2D grasps for 3D objects. Thus, we need to find a suitable grasping strategy based on the above mentioned box decomposition. We base our grasping hypotheses on the faces of the final box decomposition. The set of hypotheses is further reduced by including geometrical heuristics on which faces are valid in terms of visibility, reachability, and more [12]. For each leaf box in the hierarchy, the points enveloped by it are projected onto the valid faces of the box and stored in a grayscale image. The distance of the closest point to each pixel cell onto which it is projected is stored as a grayscale value between 0 and 255, where 1 is the depth of the box and 255 means zero depth (see Fig. 4a). This provides us with 2.5D representations of the object parts. The decomposition captures symmetries of objects quite well, resulting in faces and thus projections that are often perpendicular to the axes of most variance. This yields suitable information about approach directions of planar grasps and a good dissection of the object. In short, for each of the projections attained, grasps will be planned similarly to a top-view on a planar object. Thus grasp points on the contour of the projection images need to be found.

For grasp hypotheses from 2D contours, we will use an algorithm that is closely related to the work of Morales [4]. This algorithm involves a four step procedure for finding a number of grasp hypotheses, followed by a fifth to disqualify unfeasible grasps and selecting the best of the hypotheses.

A. Finding Good Regions to Grasp

We use the notion of grasp region, as defined in [4] and assume that a good region for grasping is a region that is as straight as possible. The fact that studies have shown that slightly concave curvature may be better suited [15] is left as a possible extension to the work. For this task a combination of the Canny edge detector and the k-angular bending algorithm [16] was used. First, the projection images described above are preprocessed by erosion and dilation steps. By removing pixels with fewer than 2 neighbours the number of outliers in the image is reduced. Expanding each remaining pixel (a projected point from 3D) to its neigbouring 8 pixels, gaps caused by sparse 3D point information are filled. Without these steps internal contours will be found that do not actually exist in the object and many grasp regions will be invalid. By using

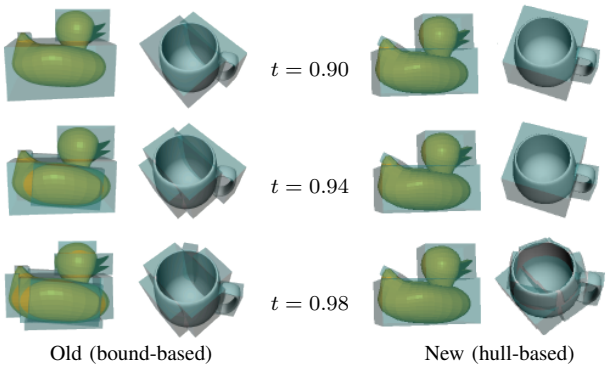


Fig. 3. Results of the box approximation for two models. Compared to the results produced with the bound-based algorithm [11] to the left, new hull-based constellations (right) stay more robust despite of different decomposition granularities (described by gain thresholds t in each row).

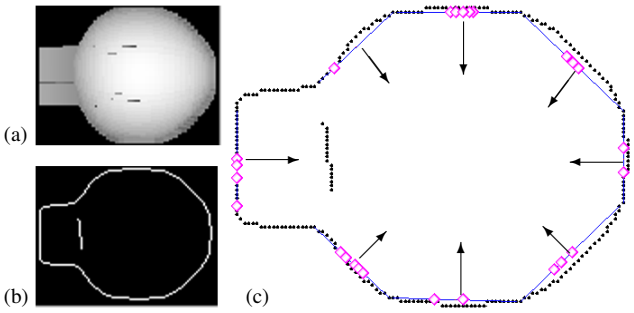


Fig. 4. (a) Projection image of the Duck model's head box (from above). (b) Canny edges image. Size of the Gaussian, lower and higher thresholds are automatically chosen by the Matlab edge algorithm. (c) A set of grasp regions (contour and 2D normal vectors) and grasp points (diamonds). The regions are found with $\sigma = 2.5$, curvature threshold $t_\kappa = 0.4$ (max. angle in radians), accumulated curvature threshold $T_\kappa = 4$, minimum length $l_{min} = 20$mm (similar to Barrett finger width), and maximum length $l_{max} = 50$mm.

the edge detector with comparably high smoothing, see Fig. 4 for an example, we extract edges in the image.[1] These edges correspond to inner and outer contours of the projected object part as well as places where depth is rapidly changing. We assume that for these edges, grasps can be executed similarly to planar objects.

With the discrete edge points detected, these are ordered in a list following the contour. From the contour we will extract regions that satisfy four main conditions:

1) The curvature in any point of the region should be low,
2) the total accumulated curvature of a region should not be too high,
3) a minimum length of a region should be achieved, in order to reduce the number of hypotheses and reduce the effect of positioning errors,
4) a maximum length of a region should not be exceeded, in order to break long straight parts of the contour into several regions such that two fingers can be placed at the same side of an object such as a cube.

The curvature part is handled by the k-angular bending algorithm [16], that considers k neighbours in each direction of a point to determine its curvature by calculating the angles to these neighbours. Let C be the ordered list of points on the contour, $c_i = (x_i, y_i)$ the i^{th} point in this list, $\vec{a}_{i,k} = c_{i+k} - c_i$ and $\vec{b}_{i,k} = c_{i-k} - c_i$. The angle between these vectors is then calculated as,

$$\kappa_i = \arccos(\vec{a}_{ki} \cdot -\vec{b}_{ki}). \quad (3)$$

Convolving κ with a Gaussian provides smooth curvature values at any point along the contour. This removes remaining noise in the image so that a pixelated straight diagonal will not be discarded because the angle between each pixel and the next is too high. This enables us to assign a threshold on the local curvature, i.e. a region is only chosen considering that no point in it has a curvature value above the threshold.

An additional requirement for a region on the contour to be accepted is that the accumulated curvature of a region is not higher than a chosen threshold. This condition is checked by summing up all κ-values for the region and comparing to the threshold. This takes care of problems with low constant curvature such as for a circle. Without the use of accumulated curvature, the circle would be regarded to have either no feasible regions to grasp or one region going straight through the center. With accumulated curvature, the circle will be broken into several regions. This also applies to other shapes with regions of low curvature with the same sign. Each region is approximated with the line connecting the endpoints of that region. Thus, each region is only represented by these two points and the inwards pointing normal, see Fig. 4.

The two remaining conditions for a region to be considered are the minimum and maximum length of a region. A minimum length is needed in order to account for positioning errors and to have a value close to the finger width. A maximum length is needed so that the representation does not become too simplified. If for a simple object like a cube no maximum length of regions was set, its projection images would only be represented by four regions that would be very hard to combine into a working grasp. By dividing the regions such that none are larger than the assigned maximum length produces more regions and therefore enables the possibility to place two fingers on one single side of the square, for example. Lower maximum length gives more regions and thus higher number of hypotheses, which means more possibilities. One should be cautious, however, since computation time increases rapidly with the number of regions.

B. Determining Finger Positions on the Regions

For each possible triplet (in the case of a 3-finger hand such as the Barrett hand [17] that is used) of regions, two criteria must be met. The normals must positively span the plane and finger placement must be such that all the friction cones of these fingers intersect. In this paper we will assume Coulomb friction and point contacts. By considering the union of all friction cones of one region and looking at the intersection of such 'combined friction cones', one can determine if the intersections of all three regions are empty and the hypothesis discarded, or non-empty and considered. This becomes a geometrical problem for each triplet and can be solved with standard linear programming methods. In the case of non-empty intersections, the centroid of the intersection area is calculated and projected back to the regions. These points will be used for finger positions, as discussed in [4].

C. Determining Hand Configuration

From the finger positions and the Barrett hand kinematics one can test if there is a configuration that can reach the selected points. By varying the angle of the thumb[2] to the surface and searching for those angles that correspond to configuration of the hand that can reach all three grasp points, one can find hypotheses for grasps. The angle of the thumb is varied on the interval (-$\arctan\mu$, $\arctan\mu$) in 100 steps,

[1] We use the Matlab standard function *edge* with parameter 'Canny' and a value for $\sigma = 2.5$ and automatically calculated threshold.

[2] Note that the thumb of the Barrett hand does not allow rotation. Thus, the angle of the thumb to the object is closely connected to the hand orientation.

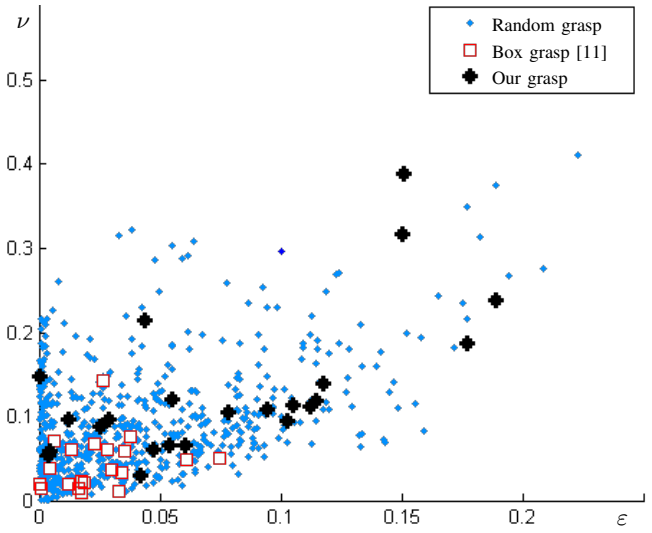


Fig. 5. Plot of 2000 random grasps, hypotheses found with the method from [11], and hypotheses found with our new approach, on the Bunny model. On the axes are the two built-in quality measures from GraspIt!. Only one grasp per finger positioning is chosen randomly and plotted. As can be seen, the best hypotheses are close to the best of the 2000 random grasps, suggesting that with only these 23 hypotheses one could get one or more good grasp.

where μ is the same friction coefficient used for calculating the friction cones in the previous section. Those configurations that satisfy the conditions are stored as grasp hypotheses. For a more detailed description, see [13]. A plot of the quality for grasp hypotheses found for the Bunny compared to 2000 random grasps is shown in Fig. 5.

D. Determining the Quality of a Grasp

From the algorithm presented one can generate a number of grasp hypotheses. However, we still need to determine which grasps are more likely to be successful. This is done by different measures of quality. Firstly however one needs to discard grasp hypotheses that are not reachable. This means that all grasp hypotheses outside of the physical reach of the hand will be discarded. This includes those grasps where one part of the object is in the way for grasping another part of the object. One example for the duck appears when a top grasp is attempted, but with finger positionings on the body of the duck: the head would be occluding the body, thus this grasp is discarded even before attempting it.

Many of the quantitative quality measures are the same as the ones developed by Morales *et al*. [4, 13], and will thus only be mentioned by name. There is one important difference, however: the empirical normalization constants used by Morales *et al*. will not be used here, as an artificial neural network will be used to determine the weights of each measure instead.

The measures derived from Morales are the following:

- q_1: Grasp Triangle Size,
- q_2: Point Arrangement,
- q_3: Force Line,
- q_4: Finger Extension,
- q_5: Finger Spread,
- q_6: Focus Deviation,
- q_7: 2D Force Focus.

These measures however are developed for planar objects. To adapt to non-planar objects to be grasped in our case we add two extra quality measures. These are:

1) Finger Depth Difference: The projection image contains information about the depth of the shape. Thus, it is possible to compare the selected grasp point depths d_i for each finger i with the linear approximations of the real finger extensions $g(e_i)$ by

$$q_8 = (g(e_1) - d_1)^2 + (g(e_2) - d_2)^2 + (g(e_3) - d_3)^2, \quad (4)$$

where $g(\cdot)$ is the linear depth approximation function.[3] This measure depicts how close to the desired grasp points the grasp is likely to be. Note that this measure is the one that explicitely takes into account the 2.5D information provided by the box approximation and projection steps from Section II.

2) 3D Force Focus: Ideally, one would like to measure the distance from the force focus in three dimensions to the actual center of gravity. This is, however, not possible since the information about the object is incomplete and the representations of grasps are only in two dimensions. We provide a rough approximation of this quality by using the center of the root box in the decomposition (containing all points in the point cloud) and the mean of the calculated finger positions in three dimensions,

$$q_9 = ||\bar{p}_{finger} - p_{rootCenter}||. \quad (5)$$

IV. Evaluation

The evaluation of the algorithm has mainly been made with data from simulation. This object data consists of 42 different 3D models, consisting of 14 different objects in 3 different scales to provide more data to train on. First, each model was decomposed with the box decomposition algorithm, using a gain threshold t=0.90. Over all models, this resulted in 570 projections from leaf boxes. 5951 grasp triplets were finally found from those projections and used as the data set for evaluation of grasps. Different types of results for the used models and decompositions are presented in Fig. 6. In a next step, the presented quality measures were computed, and grasp success measures extracted by simulating the grasps in GraspIt! [18]. The correlation between these quality measures and success measures is going to be learned by a neural network. We also explore how different network architectures affect the overall result.

A. Measure for Success

We want to produce a set of grasp hypotheses where the outcome is known in order to supervise the training. Since to do this with a real robot would be both time-consuming and costly in order to get enough data to train on, a more time and cost-efficient simulation option was used. By simulating the grasp hypotheses found for different objects, and by measuring the success for these in the simulator a set of input / output

[3]For the Barrett hand: $g(e) = 0.953 * e + 128.8$, empirically found. Using this linear approximation causes little loss in precision compared to calculating the actual inverse kinematics for the hand.

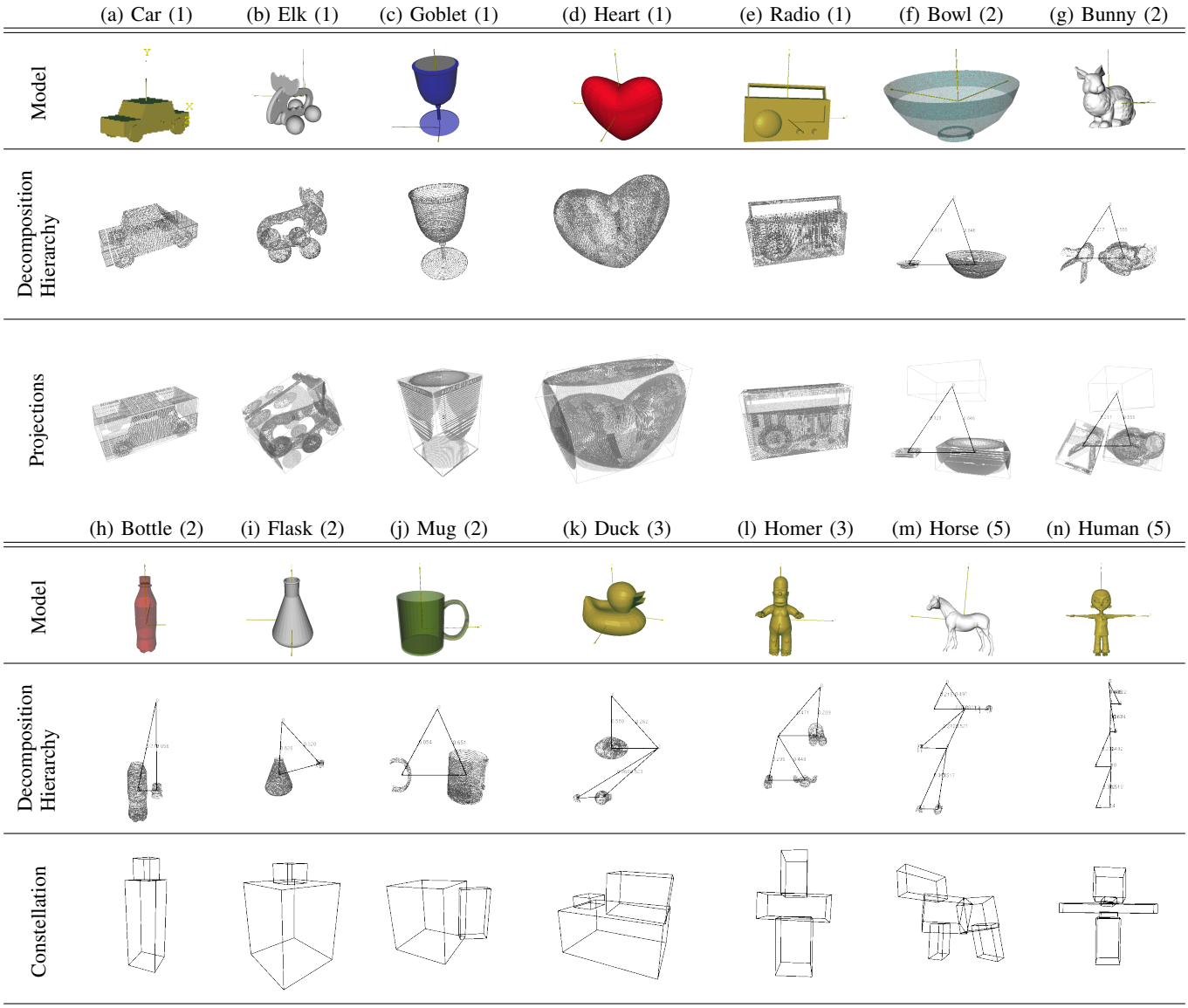


Fig. 6. The 14 models used in the experiment in order of complexity. The number of boxes resulting from the final decomposition hierarchies (2nd and 5th row) is assigned to each model in brackets. Since (a)-(e) are very compact and only the root box is included, the 3rd row visualizes examples of box face projections that will be used for the experiments. For the more complex models (h)-(n), the 6th row depicts the final box constellations. We refrained from showing constellations for (a)-(g), since they would only show 1-2 boxes, as also from showing projections for (h)-(n), since they would be hard to recognize.

pairs was found. The GraspIt! simulator [18] provides two different measures of success, introduced in [19].

The first measure, denoted ν here, measures the volume of the intersection of friction cones from the finger contacts. The second measure, called ε here, is a measure of the radius of the largest sphere that can fit in this space. For the purpose of learning it is more suitable to use only one measure since it is easier to learn a one-dimensional function than a two-dimensional. However, in order to utilize all of the information provided and since there is no firm consensus on which measure is best [20], we use a combination of the two:

$$s_i = (\nu_i/\nu_{max})^2 + (\varepsilon_i/\varepsilon_{max})^2 \quad , \tag{6}$$

where s_i is the success of grasp i, ν_{max} and ε_{max} are the maximum ν and ε values found for the current object and all hypotheses, respectively.

Other measures that could be used would be the division of grasps into two classes, namely successful and unsuccessful, or to train the system using only one of the above measures. Since potentially this could be a waste of useful grasp quality information, the above combination measure was chosen.

Given that the net is used to grasp an unknown object in the final application, the system will continue learning from newly observed hypotheses. However, since we have the possibility to initially gather vast amounts of simulation data to use for training, one additional example will not impact the prediction results noticeably. Combining this with the possibility to retrain an eager learner when the system is offline makes the advantages of a lazy learner, like kNN, diminish. Therefore, we used a feed-forward neural network to implement a supervised eager learning approach. The training algorithm used was the Levenberg-Marquardt backpropagation

algorithm included in the Matlab Neural Networks Toolbox. It is a fast training algorithm with good generalization properties, which is needed for predicting unknown objects.

B. Leave-One-Object-Out Validation

For evaluation, we apply a leave-one-object-out validation. This method validates by picking out one of the 14 objects that is not the test object, while all grasp hypotheses that belong to this object will be used as part of the validation set. There are both advantages and disadvantages to this approach. Considering that the prediction error for an unknown validation object is at a minimum, it would be intuitive to assume that the prediction error for an unknown test object would also be minimal. However, these two objects can be very different, both in complexity and suitable grasps, more than the ones in the training set and the test set. Another drawback with this technique is that the size of the validation set is different for each object used for validation. An advantage to using this approach is that it seldom overfits and thus stops the training when generalization still performs well.

C. Network Architecture

We used a network architecture with 9 input nodes, from the 9 quality measures, and 1 output node corresponding to the success measure s. From here, we still must decide how many hidden layers and how many hidden nodes in each layer to use. To be able to decide what is a good architecture and what is not we need to measure the overall success of the network. This measure should incorporate the s-measure for a grasp, but being as independent of an object as possible, e.g. an easy object to grasp will give better s-measures, but should (with the same prediction success) give the same success of the net, S_{net}. We want to rank the grasp hypotheses and only use the ones highest ranked. This is reflected in the network success measure by using only the top-ranked 10% hypotheses and comparing them with the lowest-ranked 10%:

$$S_{net} = \frac{(s_{high} - s_{low})}{0.1n} = \frac{\left(\sum_{i=1}^{0.1n} r_i - \sum_{i=0.9n}^{n} r_i\right)}{0.1n}, \quad (7)$$

where r_i is the ranked list of hypotheses, the hypothesis with highest predicted success being at index 1, and n is the total number of hypotheses.

To test the effect of adding hidden layers, three different setups were used: the first had only one hidden layer with 10 hidden nodes, the second had two hidden layers each with 10 hidden nodes, and the third had three hidden layers each with 10 hidden nodes. There was no improvement of prediction success for more than one hidden layer. The time for training, however, increased dramatically.

Extensive testing was performed in order to choose the number of nodes in the one hidden layer. This testing was done with the leave-one-object-out validation described above. Tests were made with 1 to 30 hidden nodes. After studying the performance results depicted in Fig. 7, the optimal number of hidden nodes was chosen to be 8. Using this architecture, no training phase in our experiment required more than 100 epochs.

D. Learning to Grasp Unknown Objects

In order to learn to grasp unknown objects, the leave-one-object-out validation method was applied again. Each time the network is trained one object is left out of the training data to be used for testing. This unknown object will be used to determine how well the algorithm has performed. In order to get a reliable result these tests were run 10 times. As such, one can conclude that the method for grasp synthesis, the quality measures and the learning approach used can indeed find and rank a set of grasps for an unknown object. Fig. 8 shows distribution of predicted success measures for each model. Some of the high-ranked grasps are presented in Fig. 9. These were encountered after separately performing a training on all other models in the set (except for one validation object). Note that the input for the overall approach is only a 3D point cloud representation that could also be delivered from real sensor input. The approach therefore does neither need training on every possible object model, nor does it rely on connected surface structure, like triangle meshes.

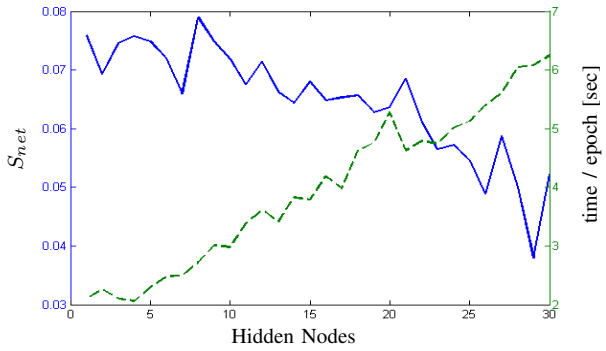


Fig. 7. Success (solid) and training time (dashed) for different number of hidden nodes with the leave-one-object-out validation, averaged over 10 runs. The maximum success value of S_{net} = 0.078 was detected at 8 nodes.

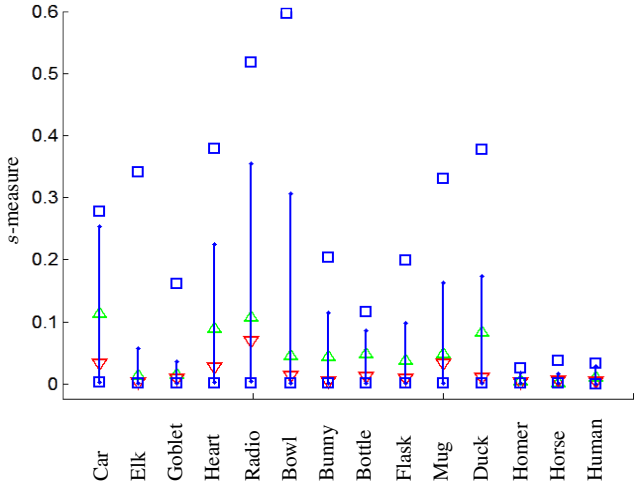


Fig. 8. Prediction results for the 14 models used (see Fig. 6). Upwards pointing triangles represent mean of the best 10% grasps, downwards the worst. Lines correspond to the possible span of predictions, with a perfect prediction of the best in the top and a perfect prediction of the worst in the bottom. The squares represent the best and the worst grasp for each object.

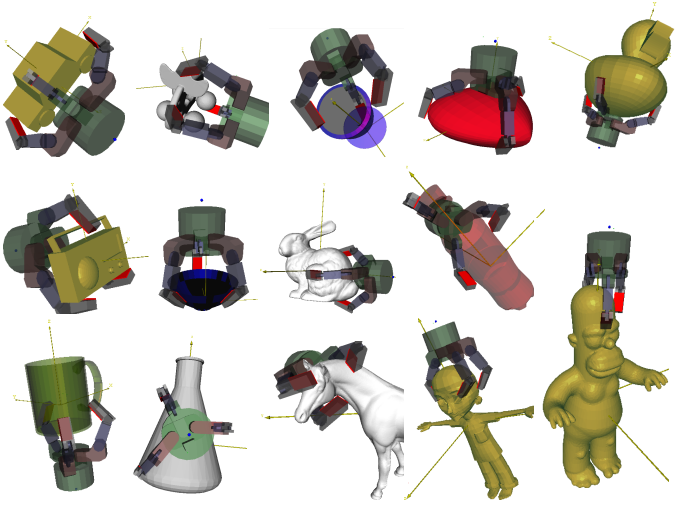


Fig. 9. Visualization of some high-ranked predicted grasps for all models.

V. Conclusions and Future Work

In this paper, we presented an approach for grasping 3D objects by using 2D grasping strategies and heuristics. We applied and extended approaches from each of these domains, namely the 3D box approximation [11] and the 2D grasping quality measures [4]. We showed that, given a point cloud of 3D data, an optimized version of box approximation produced repeatable decompositions in addition to resolving the issues encountered in our previous algorithm. This will contribute to further connected applications based on box constellation representations, e.g. learning from and grasping on whole constellations instead of just single boxes. Learning might also include a classification of the enveloped point cloud of each box and object part as another shape primitive, i.e. cylinders or spheres. Another classification will be approached by learning from the box constellation itself. Not only similarities between constellations could be used, e.g. all 'duck'-like box decompositions afford similar types of grasps, but also finger positioning on more than one face will be enabled.

From a 2.5D representation such as the ones used here, one can produce a set of feasible grasp hypotheses. For these hypotheses one can evaluate a set of physically intuitive quality measures for a 3D object and use them for learning to predict success. It is important to note that representation, synthesis and evaluation are three independent parts and do not need the other parts to be present. The only requirement for a representation is that it has to contain information not only about the position in image space for a point, but also the depth. The grasp synthesis algorithm works independently of the other two and only needs the contour and the kinematics of the hand used. For the last step, most of the quality measures are extendible to all hands with the same or a higher degree of freedom than the Barrett hand used here. This can be done either by the use of virtual fingers, or by an extension of the measures themselves to include sums and differences for more than three fingers. A continuation of the work could include an extension of the quality measures to better take into account 3D shape. With the use of more flexible hands the complete inverse kinematics could be used for finding reachable points in 3D space. A natural extension to the learning part is to include not only data from simulation, but to continue learning from real-world objects. By retraining the network with the increased data set, the evaluation would get more precise and be a useful learning system.

Acknowledgment

This work was supported by EU through PACO-PLUS, IST-FP6-IP-027657.

References

[1] B.-H. Kim, B.-J. Yi, S.-R. Oh, and I. H. Suh, "Non-Dimensionalized Performance Indices based Optimal Grasping for Multi-Fingered Hands," *Mechatronics*, vol. 14, pp. 255–280, 2004.

[2] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic Grasping of Novel Objects using Vision," *International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[3] G. M. Bone and E. Y. Du, "Multi-Metric Comparison of Optimal 2D Grasp Planning Algorithms," in *IEEE International Conference on Robotics & Automation*, 2001, pp. 3061–3066.

[4] A. Morales, P. J. Sanz, A. P. del Pobil, and A. Fagg, "Vision-based three-finger grasp synthesis constrained by hand geometry," *Robotics and Autonomous Systems*, vol. 54, pp. 496–512, 2006.

[5] D. Aarno *et al.*, "Early Reactive Grasping with Second Order 3D Feature Relations," in *From Features to Actions*, 2007, pp. 319–325.

[6] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp Planning Via Decomposition Trees," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4679–4684.

[7] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous Grasping via Eigengrasps: A Low-Dimensional Approach to a High-Complexity Problem," in *Sensing and Adapting to the Real World*, 2007.

[8] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic Grasp Planning Using Shape Primitives," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.

[9] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An SVM Learning Approach to Robotic Grasping," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 3512–3518.

[10] A. Chella, H. Dindo, F. Matraxia, and R. Pirrone, "Real-Time Visual Grasp Synthesis Using Genetic Algorithms and Neural Networks," in *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, 2007, pp. 567–578.

[11] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 1628–1633.

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

[13] A. Morales, "Learning to Predict Grasp Reliability with a Multifinger Robot Hand by using Visual Features," Ph.D. dissertation, Department of Computer and Engineering Science, Universitat Jaume I, 2004.

[14] A. M. Andrew, "Another Efficient Algorithm for Convex Hulls in Two Dimensions," *Information Processing Letters*, vol. 9, pp. 216–219, 1979.

[15] D. Montana, "The Condition for Contact Grasp Stability," in *IEEE Int. Conference on Robotics and Automation*, 1991, pp. 412–417.

[16] A. Rosenfeld and E. Johnston, "Angle Detection on Digital Curves," in *IEEE Transactions on Computers*, vol. C-22, 1973, pp. 875–878.

[17] W. T. Townsend, "The BarrettHand Grasper – Programmably Flexible Part Handling and Assembly," *Industrial Robot: An Int. Journal*, vol. 27, no. 3, pp. 181–188, 2000.

[18] A. T. Miller and P. K. Allen, "Graspit! A Versatile Simulator for Robotic Grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[19] C. Ferrari and J. Canny, "Planning Optimal Grasps," in *IEEE International Conference on Robotics and Automation*, 1992, pp. 2290–2295.

[20] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The Columbia Grasp Database," in *IEEE International Conference on Robotics and Automation*, 2009.

Grasping Known Objects with Humanoid Robots: A Box-Based Approach

Kai Huebner, Kai Welke, Markus Przybylski, Nikolaus Vahrenkamp,
Tamim Asfour, Danica Kragic and Rüdiger Dillmann

Abstract— **Autonomous grasping of household objects is one of the major skills that an intelligent service robot necessarily has to provide in order to interact with the environment. In this paper, we propose a grasping strategy for known objects, comprising an off-line, box-based grasp generation technique on 3D shape representations. The complete system is able to robustly detect an object and estimate its pose, flexibly generate grasp hypotheses from the assigned model and perform such hypotheses using visual servoing. We will present experiments implemented on the humanoid platform ARMAR-III.**

I. INTRODUCTION

Future applications of service robots require advanced object grasping and manipulation capabilities. According to Gibson [1], one of the main properties that characterizes an object is how it can be acted upon, namely what kind of actions it *affords*. In the work presented here, we deal with the problem of object grasping on a humanoid robot.

The development of humanoid robots for human daily environments is an emerging research field of robotics and challenging tasks. Recently, considerable results in this field have been achieved and several humanoid robots have been realized with various capabilities and skills. Integrated humanoid robots for daily-life environment tasks have been successfully presented with various complex behaviors (see e.g. [2], [3]). However, in order for humanoid robots to enter daily environments, it is indispensable to equip them with fundamental capabilities of grasping. This includes manipulating objects encountered in the environment and dealing with kitchen appliances and furniture such as fridges, dishwashers and doors. Research on humanoid grasping and manipulation has been done on humanoid platforms such as the HRP2 [4], ARMAR [3], the NASA Robonaut [5], Justin [6], or Dexter [7], where the problem of grasping has been approached from different perspectives.

The work to be presented in this paper is part of the EU PACO-PLUS project (www.paco-plus.org) and follows the concept of Object-Action Complexes [8], [9], [10]. Although humans master object grasping easily, few suitable representations of the entire process have yet been proposed in the neuroscientific literature. Thus, the development of robotic systems that can mimic human grasping behavior is still a challenging field of research. In addition, the robot embodiment usually does not resemble that of a human, i.e. grasps suitable for a human may not be suitable for a robot, and vice versa.

In our earlier work, we proposed and motivated a flexible framework for object grasping [11]. In this framework, we took advantage of closely connecting grasps to an efficient shape approximation technique based on box primitives and various dependencies that have to be considered in the field of grasping. However, this work was done in simulation only, using the grasp simulator GraspIt! [12].

For real experiments, object grasping with mobile manipulators requires several additional modules to be at place. Our early work demonstrated that it is possible to perform tasks through a careful design and implementation of individual modules [13]. The work presented here will also take into account the system integration aspects and demonstrate object grasping tasks on a humanoid robot. It is an extension of our previous work [3], [14] toward the realization of complexes humanoid manipulation and grasping tasks in a kitchen environment. Another main contribution of this paper will be the transfer of the above mentioned grasping approach from simulated environments to a real-world application.

This paper is organized as follows: in Section II, we will describe the central modules of our system. In Section III, the robot platform will be sketched, before we present experimental grasping results in Section IV.

II. OUR APPROACH

We will now present a strategy for grasping known objects, comprising an off-line, box-based grasp generation technique on 3D shape representations. Since the focus of this paper is the presentation of an integrated system, the applied sub-modules will be described very briefly. We provide references to our related work in which details on technical implementations and algorithms can be found. The subtasks of our system are:

A. An *Object Database*, representing 3D models of known objects,
B. a visual *Object Identification and Pose Estimation* module to recognize such an object in a real scene,
C. a *Shape Approximation* module to transform offline models into primitive shape representations,
D. a *Grasp Generation* module to dynamically generate grasp hypotheses from such representations, and
E. a *Grasp Execution* module, based on visual servoing, to execute such hypotheses on a humanoid robot.

K. Huebner and D. Kragic are with KTH – Royal Institute of Technology, Stockholm, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, e-mail: {khubner,danik}@kth.se.

K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour and R. Dillmann are with the University of Karlsruhe (TH), Karlsruhe, Germany, as members of the Institute for Anthropomatics, e-mail: {welke,przybyls,vahrenkamp,asfour,dillmann}@ira.uka.de.

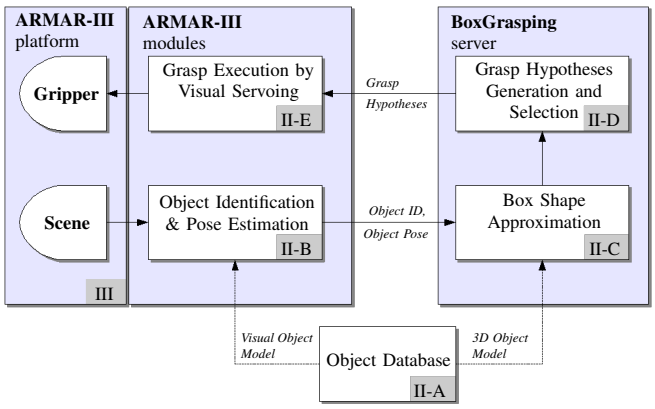


Fig. 1. System architecture for the proposed grasp generation approach.

The architecture of these modules and their interaction, as also links to the following subsections describing each single module, is presented in Fig. 1.

A. An Object Model Database

The grasping experiments we will present in this paper are performed on household objects with known geometry. The respective object models are part of the public available KIT ObjectModels Web Database [15]. In order to obtain such models, we use the interactive object modeling system introduced in [16],[17]. To acquire a 3D model, the respective object is placed on a rotation plate which is situated in front of a Minolta VI-9OO laser scanner. The scanner uses an active triangulation measurement method, providing a resolution of 640×480 measurement points and an accuracy of less than 0.2mm. Different aspects of the object are generated using different rotation angles of the plate. The measurement process results in a registered and triangulated mesh which is available in OpenInventor, VRML and Wavefront OBJ formats. In addititon, an Allied Vision Marlin stereo camera pair mounted on a rotating rig takes images of the object during the process described above. These images are used to generate texture information for the object model. The meshes from the database are registered with the recognition system (see Section II-B) and made available for box decomposition (see Section II-C).

B. Object Identification and Pose Estimation

A two-step approach using local features is applied in order to identify and localize textured objects in a scene, as presented in [18]. First, the object is recognized including 2D localization, which is accomplished using 2D feature correspondences between the image of the scene and images in the database. 2D localization is computed from a homography based on SIFT descriptor correspondences. Based on the 2D localization result, a 6D pose estimate of the object is computed by making use of the stereo camera system. For 6D pose estimation, interest points within the localized 2D area of the object are collected and correlated with the second camera image, yielding a sparse depth map. The resulting point cloud is registered with the object model.

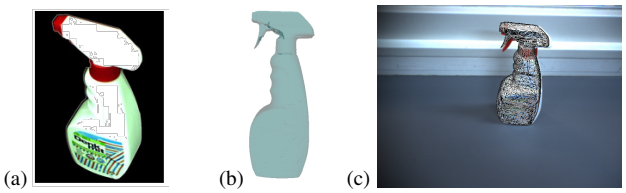


Fig. 2. Visual representations and 3D models, like (a-b), are used to describe objects in the database. (c) Result of the final pose estimate for an example scene, after application of the calibrated rigid body transformation.

To later associate object-centered grasps with objects on the basis of 3D meshes generated in the object modeling step, the fixed rigid body transformation between the object mesh and the estimated object pose has to be determined. For this purpose, we developed a tool which computes the pose of the object of interest by using the recognition module for one given scene. In parallel, the scanned model is mapped into the stereo image pair of this scene, and its pose is adjusted manually so that the model projection matches the stereo views. The desired rigid body transformation is then given by the transformation between the automatically computed pose estimate and the manually adjusted pose. An exemplary result of a final pose estimate, as also corresponding samples from the database, are shown in Fig. 2.

C. Shape Approximation through Box Decomposition

We base the generation of grasp hypotheses on a box-based 3D shape approximation technique that we presented in [19] and recently optimized in [20]. Originating from an arbitrary 3D point set and the computation of its oriented minimum volume bounding box (MVBB) [21], our method recursively splits a set of boxes to tightly envelop the point set by a set of MVBBs. By this split-and-fit strategy we aim at approximating the object shape with a minimum number of tight fitting MVBBs. The main parameter for a decomposition is a volume gain value. In case a box split will not result in a sufficient cutting-off of unoccupied space, it will not be split any further. For more details on the algorithm, we refer to [19], [20]. It is important to note that we can approximate a set of points by a box constellation. In the application here, we first decompose all 3D models (like the one in Fig. 2b) by extracting the point data from the meshes in the database in an offline step and store their respective box constellations.

D. Grasp Hypotheses Generation and Selection

Grasp hypotheses directly emerge from each face of the final box decomposition, where the approach of the gripper is aligned to the face's normal, and orientations aligned to the face's edges. The set of valid faces is reduced by mainly applying geometrical heuristics that decribe various dependencies, e.g. like spatial constellation, visibility or task at hand, as presented in [11]. To include grasp quality learning, we earlier presented two approaches based on supervised neural networks that use the grasp simulator GraspIt! [12] for learning stable grasps from 2.5D representations of object parts. Applying boxes as shape primitives efficiently allows us to generate such part-based 2.5D 'depth maps' from the

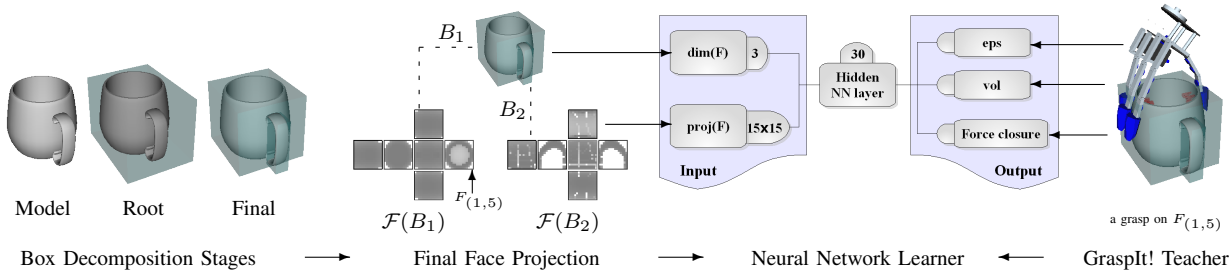


Fig. 3. The applied neural network structure holds 228 input, 30 hidden and 3 output neurons. As an input, a face projection F plus its box dimensions *dim(F)* are fed into the network, since faces are normalized to 15×15. *eps* and *vol* are the grasp quality measures that GraspIt! delivers. The force closure is also learned separately even if it equals ($eps > 0$). From this model, off-line learning of grasp qualities from face representations is possible.

3D data and the box constellation (see Fig. 3). In this paper, we will use only one specific grasp pre-shape, namely a power-grasp with a model of the robot hand that we will use. How an additional dependency concerning the gripper kinematics can be introduced in order to control finger fine-positioning was presented in [20].

E. Grasp Execution

The grasp execution on ARMAR-III [22] (see also Section III) comprises three different stages: the first two stages describe the approach of the end-effector to the final grasp pose, while in the third stage the object is grasped by closing the five-fingered hand. For approaching, two sequential poses are generated for the end-effector: (i) a pre-grasp pose which assures a collision free approach towards the grasp pose, and (ii) the grasp pose itself, which determines the final position and orientation of the end-effector before closing the fingers. While reaching for the grasp pose requires high accuracy in order to guarantee a stable grasp execution, the approach of the pre-grasp pose does not demand high accuracy. Consequently, the approach of the pre-grasp pose is realized by solving the inverse kinematics (IK) problem, while reaching for the final grasp pose is accomplished using a visual servoing approach. For both stages, the 7 joints of either the left or the right arm and the torso yaw of ARMAR-III (around the body axis) are considered.

In order to find a solution to the IK problem, we use a probabilistic approach which randomly samples start configurations. Using a Jacobian pseudoinverse method, the end-effector is moved from the sampled configurations towards the desired target pose. Thus local minima resulting from the numerical approach can be overcome and invalid postures resulting from joint limits and self-collisions can be handled. For providing natural postures as solutions for the IK problem, the resulting configuration is rated using the distance from a pre-defined grasp posture in joint space, e.g. grasping an object from the right hand side when using the right arm. The generated rating together with the ability to find a solution for the IK problem is used to rate grasp hypotheses with respect to the embodiment.

In order to execute a grasp, the torso and arm have to be moved from the pre-grasp pose to the final grasping pose. Since there are inaccuracies both in the perception of the object pose and in the execution of arm movements, we make use of a visual servoing approach to achieve exact alignment of the end effector and the object [14]. With this approach it is possible to track the hand in a robust manner and thus to adjust the pose of the hand to the feasible grasping pose.

III. EXPERIMENTAL PLATFORM

As already mentioned, we integrated the system presented in the last section on a humanoid platform, ARMAR-III. The humanoid robot ARMAR-III (see Fig. 4) was designed under a comprehensive view so that a wide range of tasks can be performed. From the kinematics control point of view, the robot consists of seven subsystems: head, left arm, right arm, left hand, right hand, torso, and a mobile platform.

The head has seven degrees-of-freedom (DoF) and is equipped with two eyes. The eyes have a common tilt and can pan independently. Each eye is equipped with two color cameras, one with a wide-angle lens for peripheral vision and one with a narrow-angle lens for foveal vision. The visual system is mounted on a four DoF neck mechanism (lower pitch, roll, yaw, upper pitch). For the acoustic localization, the head is equipped with a microphone array consisting of six microphones (two in the ears, two in the front and two in back of the head). Furthermore, an inertial sensor is installed in the head for stabilization control of the camera images.

The upper body of the robot provides 33 DoF: 14 for the arms, 16 for the hands and 3 for the torso. The arms are designed in an anthropomorphic way: 3 DoF in the shoulder, 2 DoF in the elbow and 2 DoF in the wrist. Each arm is equipped with a five-fingered hand with 8 DoF (see [23]). Each joint of the arms is equipped with a motor encoder, an axis sensor and a joint torque sensor to allow for position, velocity and torque control. In the wrists, 6D force/torque sensors are used for hybrid position and force control. Four planar skin pads (see [24]) are mounted to the front and back side of each shoulder, thus also serving as a protective cover

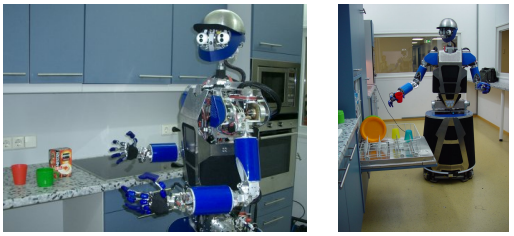


Fig. 4. ARMAR-III in the experimental kitchen environment. The robot is equipped with an active head including peripheral and foveated vision, two arms, two five-fingered hands and a holonomic mobile platform.

for the shoulder joints. Similarly, cylindrical skin pads are mounted to the upper and lower arms respectively.

The locomotion of the robot is realized using a wheel-based holonomic platform, where the wheels are equipped with passive rolls at the circumference (Mecanum wheels or Omniwheels). In addition, a spring-damper combination is used to reduce vibrations. The sensor system of the platform consists of a combination of three laser range finders and optical encoders to localize the platform. The platform hosts the power supply of the robot and the main part of the robot computer system.

For detailed information the reader is referred to [3], as also to [25] for a detailed description of the mechanics.

IV. GRASPING EXPERIMENTS

In this section, we will demonstrate the proposed method using the ARMAR-III humanoid platform in a kitchen environment, grasping common household objects.

A. Experimental Setup

For the experiments, meshes for all database objects were generated using the interactive modeling center. We will present results for three of those objects: a zwieback box, a cylindrical salt container and a complex shaped detergent sprayer bottle (see Fig. 5). In the end, the process steps for the experiment resemble the architecture modeled in Fig. 1.

In the offline preparation, all objects were registered with the recognition system as described in II-B. In order to generate a set of grasp hypotheses on each object, the decomposition of the high quality meshes (generated in the modeling step) into boxes was performed. These hypotheses are reduced using constellation and gripper embodiment dependencies, i.e. grasp hypotheses on blocked or too large surfaces will be removed. In order to rate the hypotheses related to grasp stability, grasp quality learning from a different set of training objects was performed for the left and the right hand of ARMAR-III using the GraspIt! simulator.

For the online experiments, each object is placed on the kitchen sideboard, in the field of view of the robot, and localized using the recognition system. The resulting object pose is used to transform the object-centered grasp hypotheses to the current scene. The resulting grasp hypotheses comprise approach direction, pre-grasp pose and grasp pose. The inverse kinematics solver is deployed in order to derive a rating for the reachability of the generated hypotheses. Reachable grasp hypotheses are then executable using the configuration resulting from the solver in order to align with the pre-grasp position. Finally, we manually select three of those valid grasps for each object. To perform each of them, the final poses are approached using visual servoing with a red colored ball at the wrists of both hands. Once the final grasp pose has been reached, the robot closes the hand in order to lift the object.

Fig. 5. Objects used in the experiment: zwieback, salt, sprayer bottle.

B. Experimental Results

The experimental results are depicted in Tab. I. In the first column, the corresponding models and their box approximations are shown, along with some statistics about the decomposition. The database point meshes were generated as described in Section II-A. Since both the zwieback and the salt are compact shapes, only one box was found to be necessary to suit the shape. In the case of the detergent bottle, the decomposition procedure yielded an approximation consisting of five boxes. The recursive fitting-and-splitting strategy is also reason for the higher effort in offline computation time for this object. Also note that here, though 6 boxes originally yield 30 facets, 9 of them were automatically removed because of occlusion in the constellation.

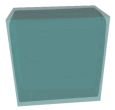
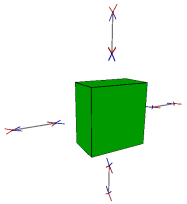
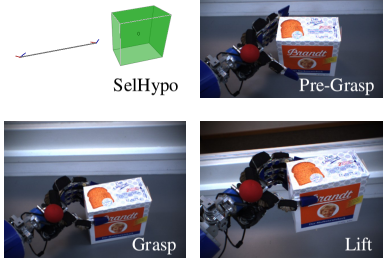
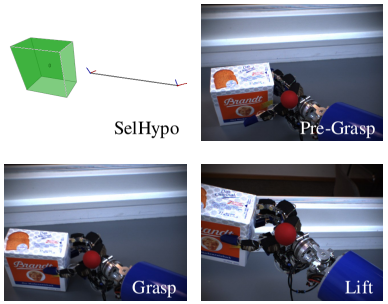
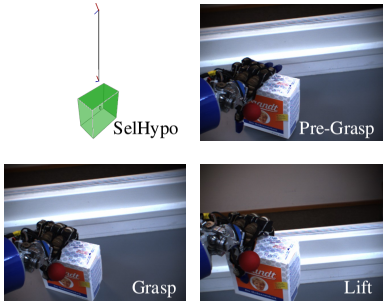
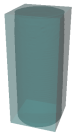
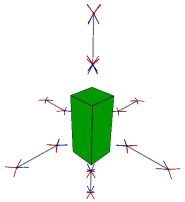
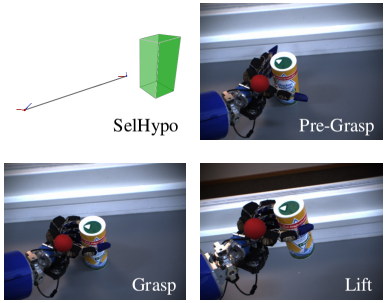
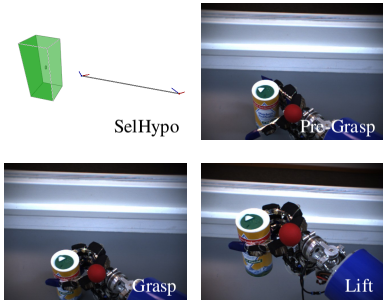
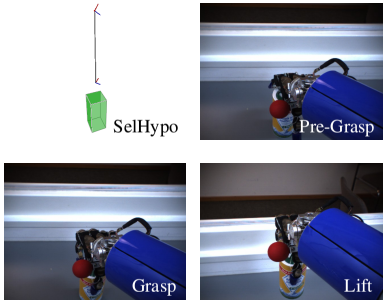
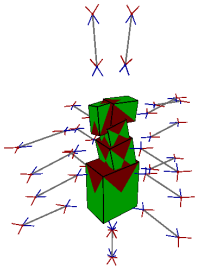
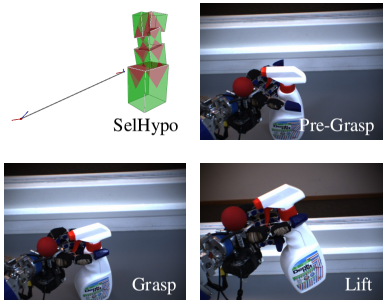
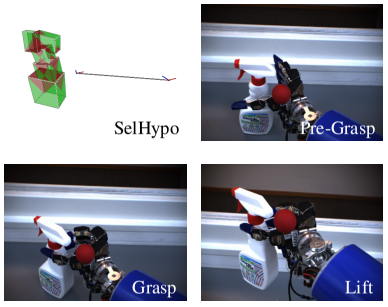
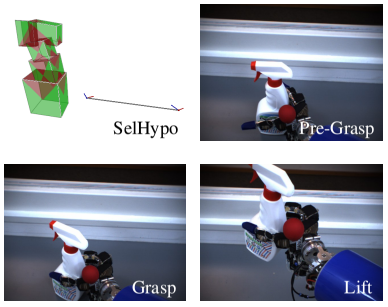
In the second column, the complete sets of generated hypotheses are depicted. The visual representations also include the grasp hypotheses removed from constellation (dark triangles). While 4 hypothesis (orientation-aligned to the four edges) emerge from each of the valid facets, some of them are removed by further constellation or gripper dependencies. Note, for example, that the zwieback box provides no grasp hypotheses from the back or front side, since the dimensions of these facets exceed the gripper aperture.

As one can see from the selected grasps in the third column, the zwieback box was successfully grasped from the left hand side, the right hand side and from the top. Similar grasps were performed on the salt can. It should be noted that it is a more difficult task to grasp the salt can from the top because of its circular lid. The box approximation of the object yielded a successful grasp even for this difficult case. It also has to be mentioned that grasps are selected manually since representations of the supporting table or other distracting objects have not been considered in this experiment, i.e. grasp hypotheses from the bottom of the object would theoretically be valid, too. Grasps on the detergent bottle were performed on two different boxes of the set: the bottom and the central box, approaching the object from the left hand side as well as from the right hand side, also resulting in stable grasps.

Despite the fact that no haptic sensor feedback is used, and that all objects turn more or less when force is applied to them during the gripping phase, all grasps are stable even during the lifting, as also they look quite intuitive and natural. It is also emphasized how grasp hypotheses selection and grasp planning can point to task-dependent grasping. For example, a power-grasp on a large box (e.g. sprayer, bottom) is suitable one for a ‘transport’ action, while a pinch-grasp on a small box (e.g. sprayer, middle) is suitable for a ‘show’ or ‘hand-over’. However, the current hand model does not support a lot of such grasp pre-shapes.

TABLE I

EXPERIMENTAL GRASPING RESULTS, INCLUDING (A) 3D DATABASE MODEL AND DECOMPOSITION, (B) GRASP HYPOTHESES AND (C) THREE SELECTED GRASPS FOR THREE OBJECTS.

Offline Generation of Grasp Hypotheses		Online Grasp Generation by Hypotheses Selection		
		(C) Grasps		
(A) Model and Decomposition	(B) Grasp Hypotheses	#1	#2	#3
 209,003 points Leaf Boxes: 1 Valid Faces: 6 Time: 4.43 sec	 8 Final Hypotheses			
 188,460 points Leaf Boxes: 1 Valid Faces: 6 Time: 4.26 sec	 16 Final Hypotheses			
 123,191 points Leaf Boxes: 5 Valid Faces: 21 Time: 26.43 sec	 36 Final Hypotheses			

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a grasping strategy for known objects, comprising an off-line, box-based grasp generation technique on 3D shape representations. The complete system is able to robustly detect an object and estimate its pose, flexibly generate grasp hypotheses from the assigned model and perform such hypotheses using visual servoing. Through the presented systems integration approach, we showed for the first time that grasp hypotheses delivered from box approximations of object models are well applicable on a real robot system. Throughout the presented experiments, object pose changes dependent on the force applied to it. Though grasps are generally stable and look human-like, we keep in mind the issue of what one can call the grip component. For the sake of efficiency and intuitive motivation, we are aware that our approach is a pre-grip component on very robust shape information. A sophisticated grip component would greatly contribute in terms of corrective movements by analyzing haptic feedback.

The box representation of an object is simple. However, the projection of an object onto the box faces ignores the real 3D shape of the object in the box, not considering the correct surface normals of the object in the grasp planning. Thus, there is a possibility that planned grasps are infeasible, which addresses the limitation of the grasp planning. In [20], we examined the integration of gripper kinematics using finger positioning estimates on the described projection patterns. However, and as future work, one can imagine higher-level part classification from point sets of the model that have been segmented through decomposition. This topic relates to work on part-based shape representations. Classification of shape is a beneficial, but also complex task, as additionally, the box constellation might be very different and unstable as influenced by noise, perspective view and uncertainties. For the purpose of grasp hypothesis generation, this is not a severe problem, while it will be in part and object classification tasks. Finally, the evaluation of the proposed method on unknown, i.e. unmodeled, objects based on 3D input perceived from a real vision system will be a challenging future work task, due to the same uncertainties.

VI. ACKNOWLEDGMENTS

This work was supported by EU through the projects PACO-PLUS, IST-FP6-IP-027657 and GRASP, IST-FP7-IP-215821, and the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft).

REFERENCES

[1] J. Gibson, "The Theory of Affordances," in *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, R. Shaw and J. Bransford, Eds. Erlbaum, NJ, 1977, pp. 67–82.

[2] K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato, and M. Inaba, "Vision Based Behavior Verification System of Humanoid Robot for Daily Environment Tasks," in *IEEE/RAS International Conference on Humanoid Robots*, 2006, pp. 7–12.

[3] 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*, vol. 56, no. 1, pp. 54–65, 2008.

[4] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba, "Humanoid Motion Generation System on HRP2-JSK for Daily Life Environment," in *IEEE International Conference Mechatronics and Automation*, 2005, pp. 1772–1777.

[5] T. Martin, R. Ambrose, M. Diftler, R. Platt, and M. Butzer, "Tactile Gloves for Autonomous Grasping with the NASA/DARPA Robonaut," in *IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 1713–1718.

[6] T. Wimbock, C. Ott, and H. Hirzinger, "Impedance Behaviors for Two-handed Manipulation: Design and Experiments," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4182–4189.

[7] R. Platt, "Learning and Generalizing Control Based Grasping and Manipulation Skills," Ph.D. dissertation, PhD Dissertation, Department of Computer Science, University of Massachusetts Amherst, 2006.

[8] D. Kraft, E. Baseski, M. Popovic, N. Krüger, N. Pugeault, D. Kragic, S. Kalkan, and F. Wörgötter, "Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes," *Humanoid Robotics*, vol. 5, pp. 247–265, 2008.

[9] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, "Cognitive Agents - a Procedural Perspective Relying on the Predictability of Object-Action-Complexes," *Robotics and Autonomous Systems*, 2008.

[10] C. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krüger, and F. Wörgötter, "Object Action Complexes as an Interface for Planning and Robot Control," in *IEEE/RAS International Conference on Humanoid Robots*, 2006.

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

[12] A. T. Miller and P. K. Allen, "Graspit! A Versatile Simulator for Robotic Grasping," *Robotics & Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110–122, 2004.

[13] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic, and H. I. Christensen, "Systems Integration for Real-World Manipulation Tasks," in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 2500–2505.

[14] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez, T. Asfour, and R. Dillmann, "Visual Servoing for Humanoid Grasping and Manipulation Tasks," in *IEEE/RAS International Conference on Humanoid Robots*, 2008, pp. 406–412.

[15] KIT ObjectModels Web Database: Object Models of Household Items, see http://i61p109.ira.uka.de/ObjectModelsWebUI.

[16] R. Becher, P. Steinhaus, R. Zöllner, and R. Dillmann, "Design and Implementation of an Interactive Object Modelling System," in *International Symposium on Robotics*, 2006.

[17] A. Kasper, R. Becher, P. Steinhaus, and R. Dillmann, "Developing and Analyzing Intuitive Modes for Interactive Object Modeling," in *International Conference on Multimodal Interfaces*, 2007.

[18] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, USA, 2007, pp. 919–924.

[19] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 1628–1633.

[20] S. Geidenstam, K. Huebner, D. Banksell, and D. Kragic, "Learning of 2D Grasping Strategies from Box-Based 3D Object Approximations," in *2009 Robotics: Science and Systems Conference*, 2009, to appear.

[21] G. Barequet and S. Har-Peled, "Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions," *Journal of Algorithms*, vol. 38, pp. 91–109, 2001.

[22] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An Integrated Humanoid Plattfrom for Sensory-Motor Control," in *IEEE/RAS International Conference on Humanoid Robots*, 2006, pp. 169–175.

[23] S. Schulz, C. Pylatiuk, A. Kargov, R. Oberle, and G. Bretthauer, "Progress in the Development of Anthropomorphic Fluidic Hands for a Humanoid Robot," in *IEEE/RAS International Conference on Humanoid Robots*, 2004, pp. 566–575.

[24] D. Göger, K. Weiß, K. Burghart, and H. Wörn, "Sensitive Skin for a Humanoid Robot," in *International Workshop on Human-Centered Robotic Systems (HCRS'06)*, Munich, 2006.

[25] A. Albers, S. Brudniok, and W. Burger, "Design and Development Process of a Humanoid Robot upper Body through Experimentation," in *IEEE/RAS Int. Conference on Humanoid Robots*, 2004, pp. 77–92.

Grasping by Parts: Robot Grasp Generation from 3D Box Primitives

Kai Huebner and Danica Kragic

January 20, 2010

Abstract

One of the core challenges in the field of robotics is to equip robots with the ability to intelligently interact with the world. To achieve this, a robot necessarily needs to perceive and interpret the environment in a proper way and understand the situations it is engaged in. The robot thus has to be able to gather and interpret the sensory information in new, unforeseen situations being provided some minimal knowledge in advance. For service robot applications, one of the key requirements is to be able to detect, recognize and manipulate objects, autonomously or in collaboration with humans and other robots.

These capabilities should also include the generation of stable grasps to safely handle even objects unknown to the robot. We believe that the key to this ability is not to select a good grasp depending on the identification of an object (e.g. as a cup), but on its shape (e.g. as a composition of shape primitives). In this paper, we envelop our previous work on shape approximation by box primitives for the goal of simple and efficient grasping, and extend it with a deeper investigation of methods and robot experiments.

1 Introduction

Researchers and programmers are working on providing robots with tasks to fulfill in order to aid and support humans in everyday situations, e.g. cleaning a table, filling a dishwasher. As only one example, such tasks are analyzed in the project PACO-PLUS[1] which our work is embedded in. The goal in the development of such cognitive systems is oriented towards enabling the robot to form useful object representations or categories by being actively engaged in the environment. This means that the robot should, like a human, learn about objects and their representations through interaction. It has been recognized that in order to achieve this, we need to integrate findings from disciplines such as neuroscience, cognitive science, robotics, multi-modal perception and machine learning. In cognitive systems, representation of objects plays a major role. A robot's local world model is built of objects that are ought to be recognized,

[1] http://www.paco-plus.org

classified, interpreted and manipulated. The representations and aims may be either hard-coded or learned in an active, on-line manner. However, whether in an office, in health care or in a domestic scenario, robots should finally operate independently to satisfy various goals. The robot thus has to be able to gather and interpret the sensory information in new, unforeseen situations being provided some minimal knowledge in advance. In order to evolve this basic knowledge, learning skills, for example from exploration or observation and imitation are widely investigated and applied.

Robot grasping capabilities are essential for perceiving, interpreting and acting in arbitrary and dynamic environments. While classical computer vision and visual interpretation of scenes focus on the robot's internal representation of the world rather passively, robot grasping capabilities are needed to actively execute tasks, modify scenarios and thereby reach versatile goals. Grasping is a central issue of various robot applications, especially when unknown objects have to be manipulated by the system.

Due to the demerit of model-based recognition, a closely related approach which is focussed on the functionalities or affordances of objects is motivated. The affordance theory by Gibson (1977) is quite attractive in current research on intertwining objects and actions, claiming that actions are directly perceived from being applied on objects, e.g. *filling* a *cup*. A cup is solid, it can stand stable and it is hollow so it can keep fluid in it. Maybe each object that holds the same attributes can also be used as a *filling device*, which humans might name a *cup*. However, the *filling device* property is alone more general and allows to put in flowers, which would make one name it a *vase* instead. This line of argument from objects to actions is also formalized into an upcoming concept, the Object Action Complexes (Geib *et al.*, 2006; Kraft *et al.*, 2008; Krüger *et al.*, 2009; Wörgötter *et al.*, 2009). Learning object categories by interacting based on such Object Action Complexes (OACs) is part of our long-term goal in PACO-PLUS. However, this concept depends on and strongly demands three core abilities from the system:

- the O-ability to recognize object properties like color or shape;
- the A-ability to perform actions; and
- the C-ability to put such observations into a higher level context, allowing learning and evolving of OACs.

In this paper, we present an approach aimed at all three items. We will mostly focus on the object description, but constrain it by performable actions. In particular, we will connect box-like representations of objects with grasping, and motivate this approach in a number of ways. Implicitly, and to show the applicability of the concept, we have some preliminary C-abilities in our system. In the entire PACO-PLUS project, C-components are investigated in a wider scope, where we refer to the OAC references mentioned above. We briefly introduce our basic work as also other related work in Section 2.

2 Related Work

The work presented here relates to grasping in robotics, as also to methods for shape approximation in computer vision and computer graphics. Related works from those fields will be summarized in the following subsections.

2.1 Grasping in Robotics

In robotic object grasping there has been a lot of effort during the past few decades (e.g. see Siciliano and Khatib (2008) for a survey). However, the existing artificial systems performing grasping and manipulation of objects are still far away from closely emulating the human perception-action system. One of the reasons is the hardware: most robot hands feature very simple contact surfaces not comparable to human hands having five soft fingers with high dexterity and compliance. Apart from the hands, robotic arms also have less dexterity and flexibility. Current robotic systems dealing with object grasping and manipulation rarely take into account task dependency, planning or exception handling, especially when the whole eye-hand coordination problem is considered.

On the processing side, it has been widely recognized that high-level task-related grasp planning is difficult due to the large search space resulting from all possible hand configurations, grasp types, and object properties that occur in realistic settings. Innovative work in this field included kinematic constraints of the hand in order to prune the search space, e.g. (Borst *et al.*, 2003; Miller *et al.*, 2003). The most common way to approach the problem has been the model-based approach. Different grasp-related components such as objects, surfaces, contacts, forces, etc., are modeled according to very specific physical laws assuming a good knowledge of the environment. Thus, the research has mainly focused on (i) *grasp analysis*, i.e. the study of the physical properties of a given grasp (Bicchi and Kumar, 2000; Zhu *et al.*, 2003), and (ii) *grasp synthesis*, the computation of grasps that meet certain pre-defined properties (Liu *et al.*, 2004; Miller *et al.*, 2003; Morales *et al.*, 2004; Pollard, 1994, 2004; Shimoga, 1996). Unfortunately, these approaches have failed to deliver practical implementations that can be implemented on different platforms independent of the hardware properties. The most crucial problem has been that the methods mostly rely on assumptions that are not satisfied in complex environments with a high degree of uncertainty. Examples of these assumptions are the use of accurate object models, contact surface properties, or disregard of dynamic properties of grasps.

Early work on contact-level grasp synthesis focused mainly on finding a fixed number of contact locations without regarding hand geometry (Liu *et al.*, 2004). Considering specifically object manipulation tasks, the work on automatic grasp synthesis and planning is of significant relevance (Miller *et al.*, 2003; Morales *et al.*, 2004; Okamura *et al.*, 2000; Shimoga, 1996). In these approaches, finger contact locations, forces and grasp wrench spaces can be simulated. Different criterions can be defined to rate grasp configurations, e.g. force closure, dexterity, equilibrium, stability and dynamic behavior (Shimoga, 1996). However, the dependency on a-priori known or dense and detailed object models is apparent.

Miller *et al.* (2003) therefore proposed grasp planning on simple shape primitives like spheres, cylinders and cones, clearly demanding a pre-classification of object shape. Dependent on the primitive shape, one can test several grasp configurations for their static stability. Ekvall and Kragic (2007) showed how a robot system can learn grasping by human demonstration using a grasp experience database. The human grasp is recognized with the help of a magnetic tracking system and mapped to the kinematics of the robot hand using a predefined lookup-table. Other than in (Miller *et al.*, 2003), the system can distinguish between ten different human grasps, adapted from Cutkosky's grasp taxonomy (Cutkosky, 1989). The grasp controller takes into account not only the object pose and the kind of grasp to be executed, but also the approach strategy of the human demonstrator. The training for the system has also been proceeded on models approximating the real objects by shape primitives. However, the classical work on contact-level planning concentrates on known primitives, or known shape, from the very beginning. In our work, we do only take into account the contact-level at the very end, in order to check how far we can get with simple shape approximation.

Hence, control and object models are commonly tailored for specific tasks, e.g. for ball catching (Namiki *et al.*, 2003). The main issue here is the automatic generation of stable grasps assuming that the model of the hand is known and/or that certain assumptions about the object (e.g. shape, pose) can be made, e.g. see (Pollard, 2004). An important question is: how can we equip robots with capabilities of gathering and interpreting the necessary information for novel tasks through interaction with the environment, but in combination with minimal prior knowledge?

In order to overcome these difficulties, machine vision has been proposed as a solution to obtain the lacking information about object shapes, or contact information to explore the object. Another trend has focused on machine learning approaches to determine the relevant features indicating a successful grasp (Morales *et al.*, 2004; Saxena *et al.*, 2008). Finally, there have been efforts to use human demonstrations for learning grasp tasks. Problematically, these approaches also commonly consider grasps as a fixed number of contact locations without any regard of hand geometry and hand kinematics (Bicchi and Kumar, 2000). An alternative paradigm, often motivated by studies on human grasping, is the so-called *knowledge-based* approach. It tries to simplify the visual grasp planning problem by reasoning on a more symbolic level. In this paradigm, object shapes are often described using shape primitives, like constellations of cubes or ellipsoids. Grasp prototypes are defined in terms of purposeful hand pre-shapes, e.g. power-grasp or pinch-grasp, and planning and selection of grasps is made according to programmed decision rules. Taking into account both hand kinematics and a-priori knowledge about the feasible grasps has been acknowledged as a more flexible and natural approach towards automatic grasp planning (Miller *et al.*, 2003). It is obvious that knowledge about the object shape and task is quite meaningful for grasp planning (Borst *et al.*, 2004), which thereby motivates a more particular view on shape approximation in context of grasping.

2.2 Shape Approximation

When addressing robotic grasping of unknown objects, one has to think about a representation that not only eases grasping, but which can also be efficiently delivered from the sensor data. Though there is interesting work on producing grasp hypotheses by visual features from 2D images only, e.g. (Saxena *et al.*, 2008), most techniques rely on 3D data. 3D data, which in its simplest form may be a set of 3D points belonging to an object's surface, can be produced by several kinds of sensors and techniques, e.g. distance imaging cameras, laser scanners or stereo camera systems. Since the last solution is cheap, easy to integrate and close to the human sensory system, a multitude of concepts in the area use 3D point distance data for feature points from stereo disparity. Entire point clouds originating from a scene are usually affected with sensor noise, distortion and uncertainties, and thus scattered cloud of points of the scenario, which has to be taken into account for precise shape approximation of such data. A higher-level representation of these points as a set of shape primitives (e.g. planes, boxes, spheres or cylinders) thus gives more valuable clues for object recognition and grasping by compressing information to their core.

Most approaches that consider this problem are likewise bottom-up, starting from point-clouds and synthesizing object shapes by using *superquadrics* (SQs). Superquadrics are parameterizable models that offer a large variety of different shapes. In the problem of 3D volume approximation stated here, only superellipsoids are used out of the group of SQs, as these are the only ones representing closed shapes. There is a multitude of state-of-the-art approaches based on parameterized superellipsoids for modeling 3D range data with shape primitives (Biegelbauer and Vincze, 2007; Chevalier *et al.*, 2003; Goldfeder *et al.*, 2007; Katsoulas, 2003; Solina and Bajcsy, 1990). If we assume that an arbitrary point cloud has to be approximated, one SQ is obviously not enough for most objects. The more complex the shape is, the more SQs have to be used to conveniently represent its different parts. Just for such cases, good generality is not possible using SQs with few parameters (Biegelbauer and Vincze, 2007). Besides the advantages of immense parameterization capabilities with at least 11 parameters, intensive research on SQs has also yielded disadvantages in two common strategies for SQ approximation. The first strategy is region-growing, starting with a set of hypotheses, the *seeds*, and let these adapt to the point set. However, this approach has not proved to be effective (Chevalier *et al.*, 2003) and suffers from refinement problem of the seeds (Katsoulas, 2003). The second strategy uses a split-and-merge technique splitting up an overall shape and merging parts again, which is more adapted to unorganized and irregular data (Chevalier *et al.*, 2003). Independent of the strategy used, the models and seeds, respectively, have to be fitted to the 3D data. This is usually done by least square minimization of an inside-outside fitting function, as there is no analytical method to compute the distance between a point and a superquadric (Goldfeder *et al.*, 2007). Thus, SQs are though a good trade-off between flexibility and computational simplicity, but sensitive to noise and outliers that will cause imperfect approximations. This is an important issue, as our work will be

based on dense stereo data, which results in more distorted and incomplete data in contrast to data points provided by range scanners which are mainly applied in related work.

The work of (Lopez-Damian, 2006; Lopez-Damian *et al.*, 2005) is related to ours in terms of object decomposition and grasping. Additionally, they propose a grasp planner to find a stable grasp. However, their concept uses polygonal structures instead of 3D points. Though one could produce polygonal surfaces from 3D point data, for example by the Power Crust algorithm (Amenta *et al.*, 2001), this introduces another step causing additional effort both in processing time and noise handling.

2.3 Previous Work, Paper Structure and Contributions

On the issue of shape approximation, we first presented a bounding box decomposition approach for arbitrary object shape approximation and robot grasping in (Huebner *et al.*, 2008). The initial technique based on Minimum Volume Bounding Boxes from 3D point clouds proposed in this work will be revisited in Section 4, *Box Decomposition*, in particular in Section 4.3. We have further improved the decomposition algorithm to be more robust under influence of noise and clutter. The new technique will be presented in Section 4.4. The content of that section holds the box decomposition itself, leading from an arbitrary point cloud to a constellation of 3D boxes. In Section 5, we will summarize and discuss how we continue with such a box constellation for the purpose of grasping. The basic ideas were introduced in (Huebner and Kragic, 2008): in the work presented here, we decribe an improved algorithm and extend it with additional detail and experiments. In Section 6, we present an experiment demonstrating the framework capabilities, and then conclude our work in Section 7. We first start with an outline of our system.

The contributions of work presented here are two-fold: in terms of shape approximation, we provide an algorithm for a 3D box primitive representation to identify object parts from 3D point clouds. We motivate and evaluate this choice particularly toward the task of grasping. For this purpose, and as a contribution in the field of grasping, we additionally provide a grasp hypothesis generation framework that utilizes the chosen box presentation in a highly flexible manner.

3 Outline of the System

We have observed that modeling 3D data by shape primitives is a valuable step for object representation. Sets of such primitives can be used to describe instances of the same object classes, e.g. cups or tables. However, it is not our aim to focus on such high-level classifications or identification of objects, but specifically on grasping. As discussed, we moreover approach a deeper understanding of objects by *interaction instead of observation* for that purpose, e.g., if there is an object that can be picked up and filled, it can be used as a cup.

The very basic features we work with are 3D point clouds. From a stereo vision system like ours (described in Section 6.1), these are typically, but very dependent on image resolution, disparity processing method, or an object's segmentation, between 20.000 and 200.000 points per scene. Processing an enormous number of data points takes time, both in approaches that use raw points for grasp hypotheses and in those that try to approximate them as good as possible by shape primitives. Thus, the question remains how rudimentary a model of an object can be in order to be handled successfully and efficiently. While comparable work uses pairs of primitive feature points, e.g. (Aarno *et al.*, 2007), or a-priori known models for each object (Tegin *et al.*, 2006), we are interested in looking into which primitive shape representations might be sufficient for the task of grasping arbitrary, unseen objects.

We believe that a mid-level solution is a promising trade-off between good approximation and efficiency for this purpose. Complex shapes are difficult to process, while simple ones will give bad approximations, resulting in unsuccessful grasps. However, we can keep in mind the capabilities of accessible methods to handle immanent approximation inaccuracies for grasping: e.g., haptic feedback, visual servoing and advanced grasp controllers for online correction of grasps. We prefer general fast online techniques instead of pre-learned offline examples, thus the algorithm's efficiency is the most important. Unknown objects are difficult to parameterize but need real-time application for robot grasping. A computation in terms of minutes for a superquadric approximation is therefore not feasible.

We aim for simplicity stating the question: Do humans approach an apple for grasping with their hand in another way as they approach a cup, or a pen in another way as a fork? While there are surely differences in fine grasping and task dependencies, differences in approaching these objects seem quite marginal, but lead us to the analysis of steps that are involved in a manipulative action.

3.1 Components of Manipulative Actions

While classical contact-level solutions include a merge of both *transport* (leading the hand to the grasp position) and *grip* (closing the fingers to perform the grasp), we see a benefit in loosely decoupling these two components. The psychophysical shortcomings of completely decoupling the grip from the transport component have been discussed in (Smeets and Brenner, 1999), even if that is described to be the classical approach. It is also hardly questioned that the transport component refers to extrinsic object properties only (e.g. position, orientation) while the grip component depends on intrinsic properties (e.g. size, shape, weight). Derbyshire *et al.* (2006) even motivate action to be an intrinsic property. As we will later see, our work here will neither separate nor combine these two components. In contrast, and as a result of putting focus on rough shape representation, it can be seen as a connecting module in-between which we call the *pre-grip* component. We will briefly describe the components that we consider to be important for any grasping action:

The *pre-grip* component is responsible for the hand configuration (wrist position, approach vector, or even finger configuration) of the gripper before an action execution like grasp or push is initialized. The *pre-grip* component might be constrained explicitly by kinematics of the gripper. In our case, this is generally not the case due to the use of grasp pre-shapes (e.g. following (Cutkosky, 1989)). The final location of the hand is also clearly dependent on the task at hand, making the task another extrinsic property.

The *transport* component is a temporal pre-decessor in the sequence of actions for the *pre-grip* component. However, the pre-grip component can be used to initialize a manipulation action by providing a goal position to which the gripper shall be transported. It has to be noted that it would demand grasp planning and collision detection in a definition of successful robot hand transport, making this problem a research area for itself.

The *grip* component is the direct successor of the grasp approach vector generation in the *pre-grip* phase. It is not handled in a comparable way to classical contact-level grasp planning, as that one connects directly to all perceptually sensed intrinsic properties. In our definition, the *grip* is the step from the initial pre-grip configuration to the state where all effectors included in the manipulation are in contact with the object. No additional object-centered properties are introduced for this component, but haptic or visual feedback will allow handling of inaccuracies and lacks of the preceding components.

The *post-grip* component is ought to be realized as a fine-controller based on tactile feedback and corrective movements, like included in (Tegin *et al.*, 2009). Precise shape, weight or surface texture properties, as information that has not been accessible before will be introduced with this component. In our work, we abstract from higher-level *manipulative* components, e.g. turning the lock for opening a bottle. We will mainly focus on very elementary grasping actions.

Grasp Component	Extrinsic Properties	Intrinsic Properties	Methods
Transport	position	–	Path Planning
Pre-Grip	orientation, task	size, rough shape	3D Shape Approximation
Grip	–	–	Haptic Feedback, Visual Servoing
Post-Grip	–	precise shape, weight, surface texture	Haptic Exploration, Corrective Movements

Table 1: Grasp components, involved object properties and methods.

These considerations are concluded and visualized in Tab. 1, showing the components, as also properties and methods connected to them. In our framework, we see the *pre-grip* component as the initial trigger for each object manipulation, considering a number of basic, both extrinsic and intrinsic, object properties. Accordingly, our main clue to approach this component is a basic 3D shape approximation which is capable to connect to the necessary properties.

4 Box Decomposition

4.1 Minimum Volume Bounding Box Algorithm

We base our algorithm on the minimum volume bounding box computation proposed by Barequet and Har-Peled (2001). Given a set of n 3D points, the implementation of the algorithm computes their Minimum Volume Bounding Box (MVBB) in $O(n \log n + n/\varepsilon^3)$ time, where ε is a factor of approximation. The algorithm is quite efficient and parameterizable by sample and grid optimizations, as also performing the computation on an arbitrary point cloud yields one tight-fitting, oriented MVBB enclosing the data points.

Our aim is now to iteratively split the box and the data points, respectively, in such a way that the new point sets yield a better box approximation of the shape. Iterative splitting of a root box corresponds to the build-up of a hierarchy of boxes. Gottschalk *et al.* (1996) present the OBBTree (Oriented Bounding Box Tree) for this purpose. The goal is to efficiently detect collisions between polygonal objects by the OBBTree representation. The realization of the splitting step is quite straightforward: each box is cut at the gravity center point of the vertices, perpendicular to the longest axis. This is done iteratively, until a box cannot be divided any further.

4.2 Fit-and-Split Adaptation

In our case, the above commonly used strategy is suboptimal. We want to conveniently approximate a shape with as few boxes as possible, thus a splitting into as many small boxes as possible is against our overall aim, if we refrain from merging them again. Additionally, though the MVBB algorithm is efficient, a fitting step after each splitting consumes valuable computation time. On the other hand, splitting at the central point is then not optimal. A heuristic to find a 'good' split is needed. Therefore, we will have to define what a 'good' split is.

Fig. 1a shows a central cut, similar to the ones used by Gottschalk's algorithm. It is obvious that this one is not optimal for our task, as it does not improve the approximation by the boxes of both new halves. It is neither intuitive, since it does not divide the bunny in semantic parts, e.g. head and body, as is shown in Fig. 1b. Such a semantic division is hard to find. Due to efficiency, we yet restricted to planar instead of non-linear cuts in the examples. But even with planes, finding the best intuitive cut would correspond to an extensive search and comparison of a lot of planes, differing in position and orientation.

Thus, we decide to test only those planes parallel to the MVBB's faces.

As a measure of a good split, we consult the relation of the box volume before and after performing the split. A split of the parent box is the better, the less volume the two child MVBBs include. Intuitively, this is clear, as shape approximation is better with highly tight-fitting boxes. We proposed the following efficient algorithm to find the best split:

4.3 Simple Best Split Computation

In (Huebner *et al.*, 2008), we tested planes parallel to the box surfaces for the best splitting plane. Each MVBB has six sides, whereof opposing pairs are parallel and symmetric. In-between each of these pairs, we can shift a cutting plane. Fig. 1d depicts this restriction on a splitting parallel to $\overline{A}$, shifted by a distance a, and $\overline{B}$ by b and $\overline{C}$ by c, respectively. A computation of new MVBBs for each value of the split parameters a, b and c would take a lot of computational effort. Therefore, we estimated the best cut by projecting the data on 2D grids which correspond to the surfaces $\overline{A}$, $\overline{B}$ and $\overline{C}$. The bunny sample data projection onto the three surface grids of the root MVBB are shown in Fig. 2. For the sake of efficiency, it is thereby abstracted from the real 3D volume of the shape.

We defined the best split as the one that minimizes the summed volume of the two partitions. Thus, we now test each discretized grid split along the six axes, using the split parameters. We define a split measure $\theta(\overline{\mathcal{F}}, \overline{f}, i)$ with $\overline{\mathcal{F}} \in \{\overline{A}, \overline{B}, \overline{C}\}$ being the projection plane to split, $\overline{f}$ being one of the two axes that span $\overline{\mathcal{F}}$, and i as the grid value on this axis that defines the current split. Consequently, we have six possible split measures

$$\begin{aligned}
&\theta_1(\overline{A}, \overline{c}, i_1), \;\; i_1 \in \mathbb{N}^{<c_{\max}}, \qquad \theta_2(\overline{A}, \overline{b}, i_2), \;\; i_2 \in \mathbb{N}^{<b_{\max}}, \\
&\theta_3(\overline{B}, \overline{c}, i_3), \;\; i_3 \in \mathbb{N}^{<c_{\max}}, \qquad \theta_4(\overline{B}, \overline{a}, i_4), \;\; i_4 \in \mathbb{N}^{<a_{\max}}, \\
&\theta_5(\overline{C}, \overline{a}, i_5), \;\; i_5 \in \mathbb{N}^{<a_{\max}}, \qquad \theta_6(\overline{C}, \overline{b}, i_6), \;\; i_6 \in \mathbb{N}^{<b_{\max}}, \qquad (1)
\end{aligned}$$

to compare, of which the minimum is supposed to lead to the best split.

The minimization of each $\theta(\overline{\mathcal{F}}, \overline{f}, i)$ was implemented as follows. For each i that cuts $\overline{\mathcal{F}}$ perpendicular to $\overline{f}$ in two rectangular shapes, we compute the two resulting minimal volumes by lower and upper bounds. The i that yields the minimum value is the best cut of $\theta(\overline{\mathcal{F}}, \overline{f}, i)$. $\theta(\overline{\mathcal{F}}, \overline{f}, i)$ was defined as the fraction between the sum of the two best cut rectangles and the whole projection rectangular. The two core algorithms have been sketched in Fig. 3, Alg. 4.1 and 4.2. Though this is a very approximative method, it is quite fast, as rectangle volume and bound computation are easy to process. Fig. 2 shows the best cuts for which rectangular volume and the corresponding values $\theta_{1\ldots6}$ are minimal. However, several problems arose from the proposed kind of split estimation and led us to later improve the performance:

Cutting Non-Convex Shapes. In projections with a 'valley' between two equally high 'hills', both upper bounds and thus the volume to minimize will be constant. For example, in Fig. 2 (θ_6) it would intuitively be a valuable next

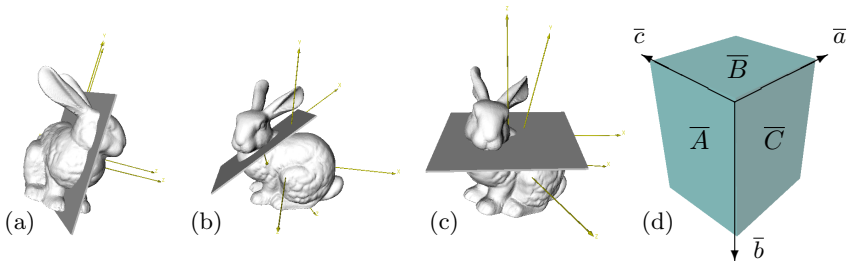


Figure 1: Exemplary cuts of the bunny: (a) a central cut, (b) an intuitively best cut and (c) a good cut parallel to one of the root MVBB planes. (d) Restriction to surface-parallel cutting planes in the simple approach.

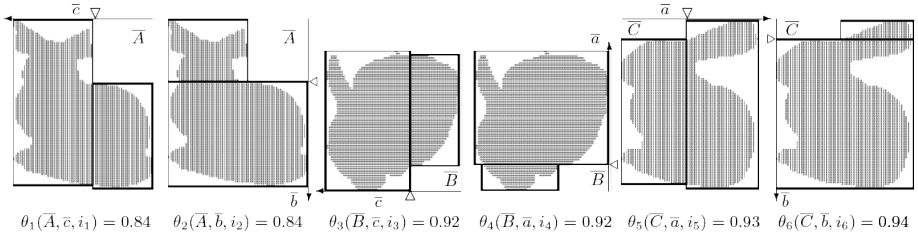


Figure 2: Best cuts along the six box directions and cut positions i (*triangles*). The corresponding volume values $\theta(\overline{\mathcal{F}}, \overline{f}, i)$ (according to (1)) are also presented.

cut below the bunny's ear. Nevertheless, the computation presented will not detect this cut. In earlier work, we stated that finding such is not that simple, especially when distorted, sparse and insecure data is provided. The issue is how to distinguish between a real 'valley' and just incompleteness of the data. An add-on for the solution of this problem would therefore be more complex and time-consuming. The bunny is a very ideal model, as it is artificial, complete, and data points are very dense. Since it is our aim to evaluate our algorithm also on real sensory data, we can not generally assume such ideal conditions.

Cutting Shape Extremities. The minimum volume box fitting approach naturally prefers fitting extremities of the shape into the corners of the boxes, as this keeps the box smaller. The bunny's ear is again an example for this, since it is almost diagonally suited into one of the box corners. However, especially such extremities can rarely be nicely cut by a face-parallel cut as proposed.

Sensitivity to Noise. A third reason is the result of the box decomposition's robustness evaluation that we presented in (Geidenstam *et al.*, 2009). Briefly summarizing, the evaluation shows that face-parallel splitting is very sensitive to each kind of inaccuracy that can emerge from a real 3D scene and sensors: noise, outliers, shape incompleteness due to viewpoint or viewpoint change, etc.

Algorithm 4.1: BoxApproximate($points^{3d}$)

$P \leftarrow findBoundingBox(points^{3d})$
$\{\overline{A}, \overline{B}, \overline{C}\} \leftarrow nonOppositeFaces(P)$
$(p_1^{3d}, p_2^{3d}) \leftarrow split(\text{FindBestSplit}(\{\overline{A}, \overline{B}, \overline{C}\}, points^{3d}))$
$(C_1, C_2) \leftarrow (findBoundingBox(p_1^{3d}), findBoundingBox(p_2^{3d})$
if $(percentualVolume(C_1 + C_2, P) < t)$ ← see (3)
  then BoxApproximate(p_1^{3d}) **and** BoxApproximate(p_2^{3d})
  else return (P)

Algorithm 4.2: FindBestSplit_Bound($\{\overline{A}, \overline{B}, \overline{C}\}, points^{3d}$)

for $\overline{F} \leftarrow \overline{A}$ **to** $\overline{C}$
do
- $p^{2d} \leftarrow project(points^{3d}, \overline{F})$
- **for** $i \leftarrow 1$ **to** $\overline{f^x}_{max}$
  - **do**
    - $(p_1^{2d}, p_2^{2d}) \leftarrow verticalSplit(p^{2D}, i)$
    - $\theta = \frac{boundArea(p_1^{2d}) + boundArea(p_2^{2d})}{area(\overline{F})}$
    - **if** $(\theta < \theta^*)$
      - **then** $(\theta^* \leftarrow \theta)$ **and** $(bestSplit \leftarrow (\overline{F}, \overline{f^x}, i))$
- **for** $i \leftarrow 1$ **to** $\overline{f^y}_{max}$
  - **do**
    - $(p_1^{2d}, p_2^{2d}) \leftarrow horizontalSplit(p^{2D}, i)$
    - $\theta = \frac{boundArea(p_1^{2d}) + boundArea(p_2^{2d})}{area(\overline{F})}$
    - **if** $(\theta < \theta^*)$
      - **then** $(\theta^* \leftarrow \theta)$ **and** $(bestSplit \leftarrow (\overline{F}, \overline{f^y}, i))$

return $(bestSplit)$

Algorithm 4.3: FindBestSplit_ConvexHull($\{\overline{A}, \overline{B}, \overline{C}\}, points^{3d}$)

for $\overline{F} \leftarrow \overline{A}$ **to** $\overline{C}$
do
- $points^{2D} \leftarrow project(points^{3d}, \overline{F})$
- $S \leftarrow longSegments(CH(points^{2D}), lengthThreshold)$
- **for each** $(S_{(i,j)}, S_{(k,l)})$
  - **with** $(i, k = segmentIndex; i \neq k), (j, l = pointIndex; j, l < n)$
  - **do**
    - $(p1, p2) \leftarrow split(p^{2D}, S_{(i,j)}, S_{(k,l)})$
    - $\theta' = \frac{area(CH(p1)) + area(CH(p2))}{area(CH(points^{2D}))}$ ← see (2)
    - **if** $(\theta' < \theta^*)$
      - **then** $(\theta^* \leftarrow \theta')$ **and** $(bestSplit \leftarrow (S_{(i,j)}, S_{(k,l)}))$

return $(bestSplit)$

Figure 3: Pseudocode: a point set and its bounding box, respectively, are recursively split (Alg. 4.1). A good split is estimated through analysis of 2D splits of the projected points onto each of the box faces, either using edge-parallel cuts (Alg. 4.2) or convex hull computations (Alg. 4.3).

4.4 Advanced Best Split Computation

These issues prompted us to revisit the split computation. Finally, we developed an algorithm based on convex hulls that solves all the three issues of the simple best splitting and additionally presents much more confident splitting results. For efficiently computing convex hulls on a set of 2D points p, like our projections, we use a Monotone Chain Algorithm (Andrew, 1979). Starting from the convex hull $CH(p)$ of the whole projection, we select those segments S_i of the hull which exceed a given threshold in length. We thereby assume that those either span a 'valley' of the outer contour of the data, or they represent a very straight edge. On these segments, we interpolate a number n of sample points $S_{i,j}, j < n$. Between each pair of points $(S_{i,j}, S_{k,l})$ with $i \neq k$, we simulate a cut that splits the point set p into two subsets p_1 and p_2. The two segment points that minimize

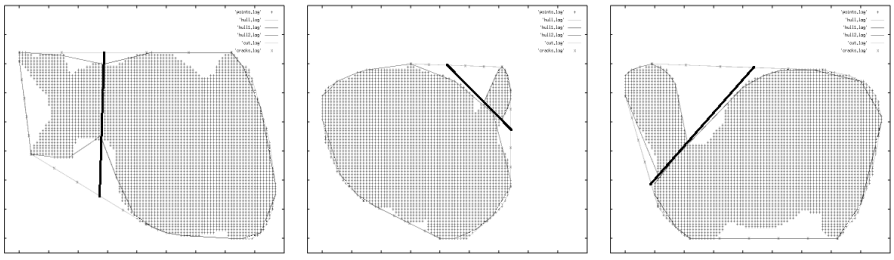
$$\theta' = \frac{area(CH(p_1)) + area(CH(p_2))}{area(CH(p))} \tag{2}$$

define our best split, where A is the area function for a convex hull. Practically, we use $n = 6$ (see also Fig. 4). Increasing n might produce more precise cuts, but for the price of additional convex hull computation cost. Pseudocode of this algorithm is sketched in Fig. 3, Alg. 4.3.

Performance of both the old and the new technique can be compared taking a look at Fig. 5. The new technique is more robust to the change of the gain threshold t which will be discussed below. The duck model (Fig. 5f) is not even affected in the cases tested, but stays with the same constellation of three boxes. Another visible effect is that the decompositions seem more intuitive, e.g. the cuts of the handle of the cup (Fig. 5e), or the ears of the bunny (Fig. 5h). This is due to the now unrestricted pose of the 2D cutting lines.

4.5 Fit-and-Split Hierarchy Building

According to the best split θ^*, which would be θ_1 or θ_2 and $\theta'_{\overline{A}}$ respectively in our examples above, the original point cloud can be divided into two subsets of



(a) $\theta'_{\overline{A}} = 0.69$ (b) $\theta'_{\overline{B}} = 0.71$ (c) $\theta'_{\overline{C}} = 0.72$

Figure 4: The three best projection splits when using the advanced convex hull algorithm. The result is much more confident than those derived from Fig. 1, while the additional computational effort is still acceptable.

the data points. These can be used as inputs to the MVBB algorithm again and will produce two child MVBBs of the root MVBB. In this way, the complete technique of fit-and-split can iteratively be performed. It is important to note that by MVBB computation, the MVBBs are not axis-aligned, but will greatly differ from the box cuts depicted in Fig. 2 in both orientation and scale.

Additionally, the previous step of 2D cutting is just equal to computing an approximative gain value, for the purpose of efficiency. As an iteration breaking criterion, we subsequently test the real MVBB volume gain Θ^* of the resulting best split measure θ^*. Therefore, we compute the gain in volume defining

$$\Theta^* = \frac{volume(C_1) + volume(C_2) + V(A^{\setminus P})}{volume(P) + volume(A^{\setminus P})}, \tag{3}$$

where A is the overall set of boxes in the current hierarchy, P is the current (parent) box, C_1, C_2 are the two child boxes that might be produced by the split, and *volume* being a volume function.
We decide further process on two constraints:

- If the gain is too low, a split is not valuable. For this purpose, we include a threshold value t that can also be used as a parameter. The precision of the whole approximation can be parameterized by simply preventing a split if Θ^* exceeds t. A threshold between $t = 0.90$ and $t = 0.95$ has given good results in most of our experiments.

- We do not preserve boxes in the hierarchy that include a very low number of points. By this process, noise in the point data can be handled to a certain extent.

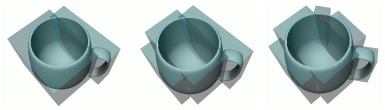
Note that by t, both the depth of the hierarchy, the number of leaf boxes, and thereby the detail of approximation can be controlled. Where a split is done is not dependent on t. Thus, we can later on easily let t evolve, e.g. from a rough root box approximation in the beginning that already can be used for transport, size attribution or grasping, to a higher degree of decomposition into parts.

An example of a decomposition hierarchy can be seen in Fig. 6 which led to the rightmost result in Fig. 5g, using a gain threshold $t = 0.98$. Each time, a best volume split value Θ^* is below t, a valid cut is detected and it is continued separately with two new point clouds. Otherwise, the treated box is a leaf box and thereby part of the final constellation. Besides the result for $t = 0.98$, which is the whole binary tree presented, sub-graphs represent the state of lower thresholds, e.g. those presented for $t = 0.90, 0.94$ in Fig. 5g.

5 Pre-Grasp Heuristics on Box Representations

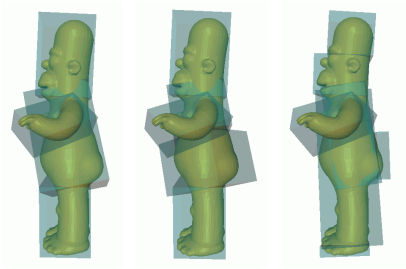
The common way to evaluate grasping strategies is extensive evaluation which is practically possible only in simulation (Goldfeder *et al.*, 2007; Miller *et al.*, 2003). Miller *et al.* have simulated pre-models and shape primitives using their public grasp simulation environment GraspIt! (Miller *et al.*, 2003). We also base

Simple Best Split

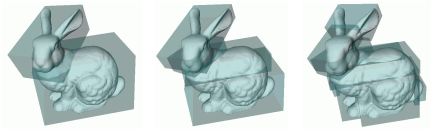


(a) Mug: MVBBs (2,3,5) produced with t=0.90, 0.94, 0.98.

(b) Duck: MVBBs (3,5,9) produced with t=0.90, 0.94, 0.98.

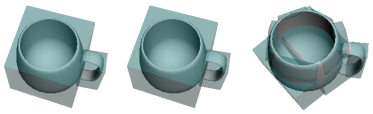


(c) Homer: MVBBs (4,5,7) produced with t=0.90, 0.94, 0.98.



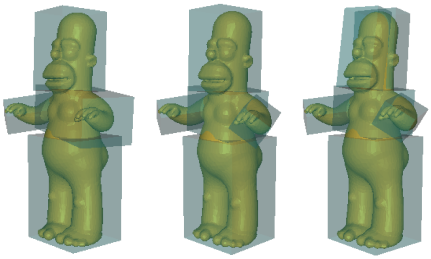
(d) Bunny: MVBBs (2,4,11) produced with t=0.90, 0.94, 0.98.

Advanced Best Split

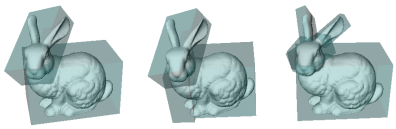


(e) Mug: MVBBs (2,2,17) produced with t=0.90, 0.94, 0.98.

(f) Duck: MVBBs (3,3,3) produced with t=0.90, 0.94, 0.98.



(g) Homer: MVBBs (4,5,6) produced with t=0.90, 0.94, 0.98.



(h) Bunny: MVBBs (2,3,4) produced with t=0.90, 0.94, 0.98.

Figure 5: Examples of box decomposition using different gain thresholds t=0.90, 0.94, 0.98, where numbers in brackets correspond to numbers of boxes. (a)-(d) show the results with the simple cutting proposed in Section 4.3, while (e)-(h) shows the advanced cutting proposed in Section 4.4.

most of our following experiments on model-based grasping in the GraspIt! simulator. For the evaluation, we create lots of GraspIt! worlds, each of which contains a model of a gripper mounted on a freely movable 'Euler' robot, since the hand is not able to move itself in free space. Additionally, an object that is to be grasped is included into the world, being the only difference between the world files (one for each object).

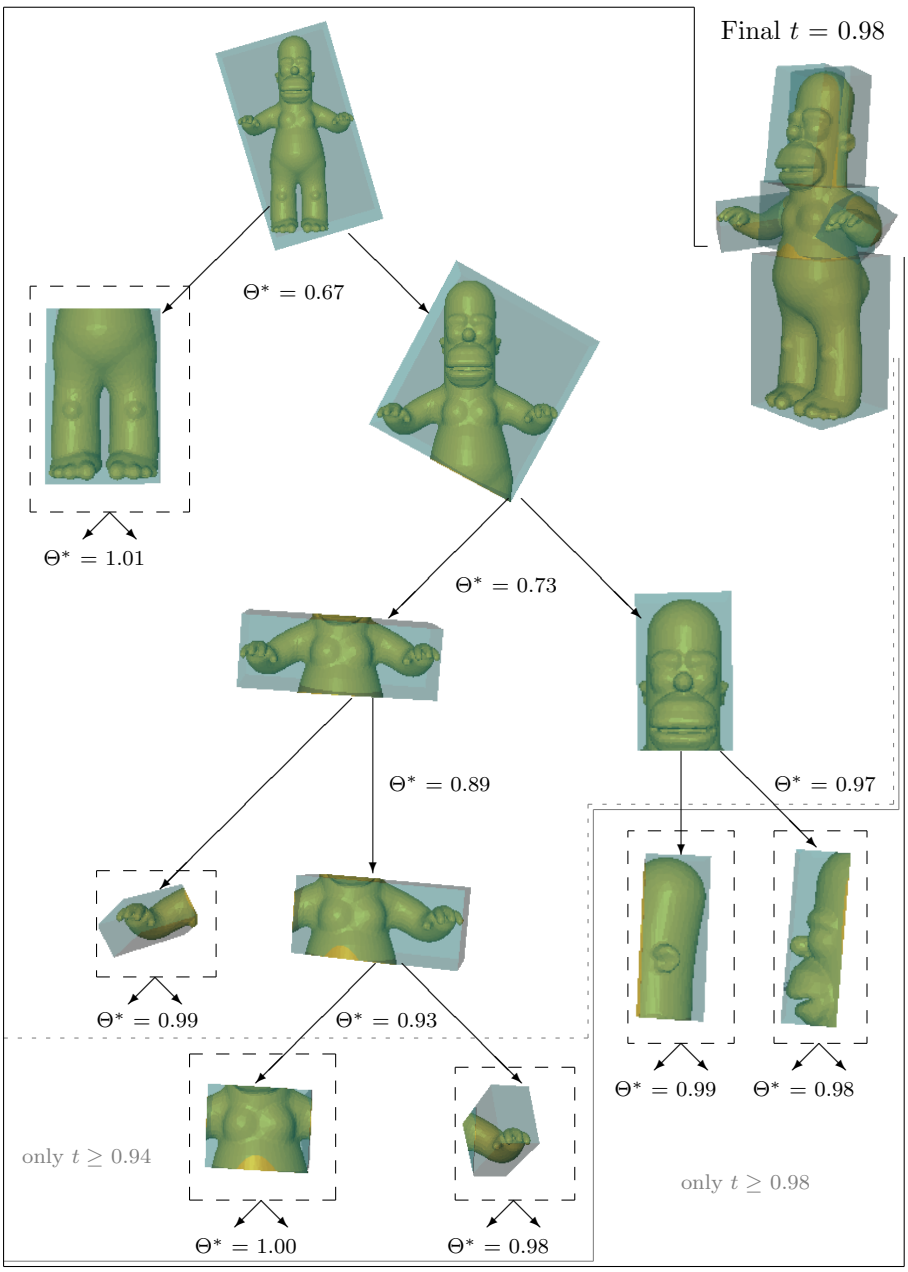


Figure 6: Visualization of a decomposition hierarchy. The example shows a model decomposition, using a gain threshold $t = 0.98$. Is the best volume split value Θ^* below this threshold, a valid cut is detected. Otherwise, the box is a leaf box (*dashed*), a part of the final constellation. One can also trace the results with lower thresholds ($t = 0.9$ dotted gray; $t = 0.94$ solid gray) from Fig. 5g.

5.1 Connecting Box Faces to Grasp Hypotheses

After an experiment has been initialized, the first iteration of the MVBB algorithm is performed as proposed in Section 4. All levels of generated boxes are subsequently inserted into the hierarchy. The first iteration will produce a root box with six faces. It is important to note that faces will be very closely connected to our grasp hypotheses. In our experiments, each face will be used for grasp hypotheses parallel to the edges spanning it. There are two types of approach techniques that we will apply: a (backup) power-grasp and a pinch-grasp.

The **Power Grasp (Backup)** will put each pre-grasp position to a constant distance from the face's center aligned to its normal. We let the hand approach the object along the normal until an arbitrary contact is detected. Afterwards, the hand retreats a small distance (the backup) to call the autograsping function (see below). The *backup* is mainly used due to technical constraints of the simulator, and based on contact instead of shape representation. Despite this technical discrepance, we will call this type of grasp a *power grasp*, to be in line with common grasp taxonomies.

The **Pinch Grasp** will force a grasp towards the center of mass by approximating the distance to the box center the current face belongs to. The *pinch* is therefore based on the shape representation of the object. In contrast to the power grasp, this technique is assumed better for small part grasping, as our power grasp will usually retreat due to contact with another object first (e.g. a table under a pen). From the approximated distance, the autograsp is called.

The Autograsp is a built-in grasping technique of GraspIt! which closes all fingers of a gripper simultaneously. We apply just one initial posture for each hand and do not consider different grasp pre-shapes at this point. Of course, the boxes themselves are not only transparent in the simulator, but also physically penetrable for each object in the scene. Thus, the fingers will grasp through the box representation and perform contact on the real object model.

When all fingers are in contact, GraspIt! provides computation of two different grasp quality measures, namely eps_{L1} and vol_{L1}. Both measures rely on the intersection of force cones derivable from the contact points. While vol_{L1} describes the volume of intersection as a measure of grasp quality, eps_{L1} corresponds the the radius of the maximum sphere that can be placed inside this volume. After the grasps on the root box have been performed and evaluated, we continue the proposed fit-and-split algorithm until it is finished. We collect all faces of boxes from the final approximation and remove occluded, ungraspable faces from the set. Finally, the same grasping process is made for each remaining face as described for the root box.

5.1.1 Experimental Results

In this work, we will evaluate the box decomposition with two robotic hand models: a three-finger, 4 DOF Barrett Hand (Townsend, 2000), and a five-finger, 7 DOF Karlsruhe Hand (Fukaya *et al.*, 2000). For each of the models from the complete model data set, we go through this grasping evaluation for the root box and the boxes produced with gain parameters 0.90, 0.94 and 0.98. The boxes are computed according to the MVBB fit-and-split algorithm proposed in Section 4.5 and resulting faces grasped with power and pinch grasp, respectively. For each try, we take a look at the final grasp qualities and the one grasp that is best rated according to these quality measures. The results can be seen in Tab. 2 and 3 for the two different hand models.

For each object model, the number of overall faces is 24 times the number of boxes, since each box has 6 faces. Here, we use 4 different grasp orientations parallel to the face edges. Geometrical detection of blocked faces reduces the number of graspable faces drastically, as also the consideration of maximum width that the hand can grasp between its fingers. For example, there are no valid grasps for the Bunny root box using the Karlsruhe hand, as this is a too large model (see best grasps in Fig. 7 and 8). Sometimes, mainly for the Karlsruhe hand and the pinch grasp, setting the hand near to the box directly causes a collision with another object part. As these cases are also defined as invalid, the number of valid faces for a pinch grasp are often much fewer than those for the power grasp (compare Tab. 3).

For most objects, the best grasp type in terms of the level of decomposition is quite comparable, independent of the chosen hand model. The main difference can be seen in the face that both hands select for the most stable grasp. A clear difference can be seen for the homer model. The most stable grasp using the Karlsruhe hand is a pinch grasp on the highest decomposition level (0.98), while with the Barrett hand, it is a power grasp on the root box. We can make the following observations from the experiment:

- Considering the grasp quality values only, more stable grasps are generated for the five-finger hand. This is due to a higher number of contact points from more fingers, different material of the hands in the simulator, and the grasp pre-shape of the Barrett hand, which is very similar to a two-finger grasp. This motivates a deeper investigation on grasp pre-shapes. However, such pre-shapes are clearly task-dependent.

- The same observation also relates to the percentage of force closure grasps, which is equal to $eps_{L1} > 0$. While with the Barrett hand, only 15% of the hypotheses result in force-closure grasps (Tab. 2), the Karlsruhe hand experiments show that 56% of grasp hypotheses produced by our box decomposition approach are successful force-closure grasps (Tab. 3).

- The eps_{L1} quality measure seems to yield even more intuitive results. While a human probably would rate the best eps_{L1} grasps in Fig. 7 and 8 as quite stable, the best vol_{L1} look unstable in many cases.

Box set	Grasp	#FC	#VF	#OF	eps^*_{L1}	vol^*_{L1}
Duck Root	Power	0	24	24	———	0.00942
Duck Root	Pinch	2	24	24	0.01050	0.00879
Duck 0.90	Power	0	24	72	———	**0.00944**
Duck 0.90	Pinch	6	24	72	**0.02675**	0.00529
Duck 0.94	Power	0	24	72	———	**0.00944**
Duck 0.94	Pinch	6	24	72	**0.02675**	0.00529
Duck 0.98	Power	0	24	72	———	**0.00944**
Duck 0.98	Pinch	6	24	72	**0.02675**	0.00529
Mug Root	Power	8	24	24	0.01330	0.00122
Mug Root	Pinch	10	24	24	0.01811	0.00144
Mug 0.90	Power	6	24	48	0.01044	0.00057
Mug 0.90	Pinch	7	24	48	0.01670	0.00043
Mug 0.94	Power	6	24	48	0.01044	0.00057
Mug 0.94	Pinch	7	24	48	0.01670	0.00043
Mug 0.98	Power	1	38	408	**0.02075**	0.00114
Mug 0.98	Pinch	0	34	408	———	**0.00201**
Bunny Root	Power	2	24	24	0.01590	0.00670
Bunny Root	Pinch	0	24	24	———	0.00233
Bunny 0.90	Power	5	28	48	0.00491	**0.00689**
Bunny 0.90	Pinch	6	28	48	0.03268	0.00517
Bunny 0.94	Power	3	20	72	0.00943	0.00599
Bunny 0.94	Pinch	5	20	72	**0.05053**	0.00446
Bunny 0.98	Power	5	32	96	0.00491	**0.00689**
Bunny 0.98	Pinch	6	32	96	0.03268	0.00517
Homer Root	Power	6	24	24	**0.03257**	**0.01086**
Homer Root	Pinch	3	24	24	0.00458	0.00446
Homer 0.90	Power	2	30	96	0.00355	0.00208
Homer 0.90	Pinch	6	30	96	0.02553	0.00126
Homer 0.94	Power	2	32	120	0.00355	0.00208
Homer 0.94	Pinch	6	32	120	0.02553	0.00126
Homer 0.98	Power	2	36	144	0.01031	0.00149
Homer 0.98	Pinch	5	36	144	0.02553	0.00266
Average		129	860		= 15%	

Table 2: Table of the experimental grasping results (Barrett hand). FC = Force Closure Grasps ($eps_{\mathrm{L1}} > 0$). VF = Valid Faces (after heuristical selection). OF = Overall Faces (after box decomposition).

Box set	Grasp	#FC	#VF	#OF	eps^*_{L1}	vol^*_{L1}
Duck Root	Power	8	16	24	**0.35045**	1.32220
Duck Root	Pinch	5	8	24	0.15746	1.07563
Duck 0.90	Power	11	18	72	0.15366	**1.44921**
Duck 0.90	Pinch	10	18	72	0.31731	0.88817
Duck 0.94	Power	11	18	72	0.15366	**1.44921**
Duck 0.94	Pinch	10	14	72	0.31731	0.88817
Duck 0.98	Power	11	18	72	0.15366	**1.44921**
Duck 0.98	Pinch	10	14	72	0.31731	0.88817
Mug Root	Power	0	0	24	———	———
Mug Root	Pinch	0	0	24	———	———
Mug 0.90	Power	3	12	48	0.04535	0.14132
Mug 0.90	Pinch	2	5	48	0.04486	0.16683
Mug 0.94	Power	3	12	48	0.04535	0.14132
Mug 0.94	Pinch	2	5	48	0.04486	0.16683
Mug 0.98	Power	7	30	408	**0.08689**	0.15867
Mug 0.98	Pinch	5	20	408	0.05113	**0.31882**
Bunny Root	Power	0	0	24	———	———
Bunny Root	Pinch	0	0	24	———	———
Bunny 0.90	Power	5	20	48	**0.15010**	**0.46843**
Bunny 0.90	Pinch	4	4	48	0.11063	0.33039
Bunny 0.94	Power	5	20	72	0.08698	0.22224
Bunny 0.94	Pinch	6	6	72	0.11063	0.33039
Bunny 0.98	Power	5	20	96	**0.15010**	**0.46843**
Bunny 0.98	Pinch	3	3	96	0.05245	0.03133
Homer Root	Power	7	8	24	0.15218	0.19584
Homer Root	Pinch	6	6	24	0.15996	0.37877
Homer 0.90	Power	24	30	96	0.13771	0.35117
Homer 0.90	Pinch	20	25	96	0.15044	0.35336
Homer 0.94	Power	24	32	120	0.13771	0.35117
Homer 0.94	Pinch	20	26	120	0.15044	0.35336
Homer 0.98	Power	20	36	144	0.13771	0.35117
Homer 0.98	Pinch	17	29	144	**0.20816**	**1.03122**
Average		264	473	= 56%		

Table 3: Table of the experimental grasping results (Karlsruhe hand). FC = Force Closure Grasps ($eps_{\mathrm{L1}} > 0$). VF = Valid Faces (after heuristical selection). OF = Overall Faces (after box decomposition).

- The experiments have been performed in static simulation using GraspIt!, i.e. when contact appears between a finger and the object, that finger is stopped at this position. In reality or dynamic simulation, forces would naturally draw the object towards the palm or opposing fingers. More contact points would be available, improving the quality measures. The implementation of dynamic experiments in GraspIt! is our ongoing work (Tegin *et al.*, 2009).

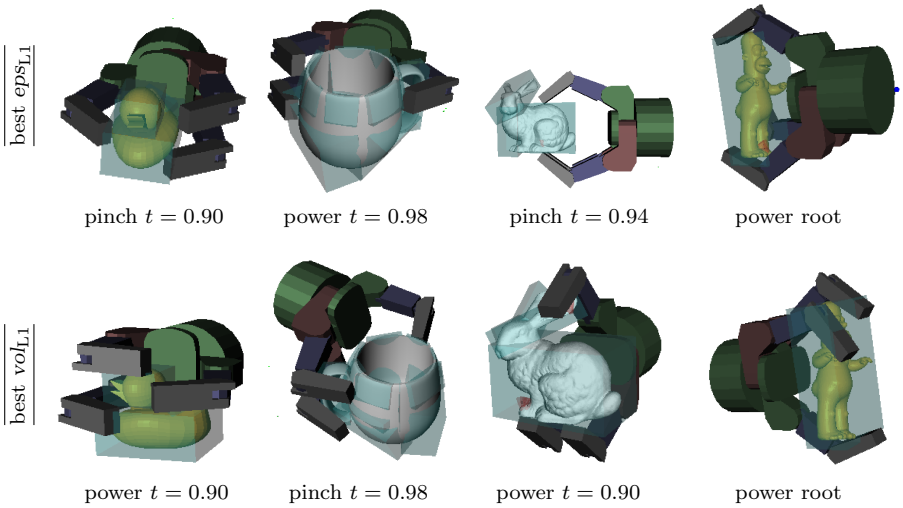


Figure 7: Best grasps for the experimental Barrett hand grasping results in Tab. 2. Upper row: best eps_{L1} grasps. Bottom row: best vol_{L1} grasps.

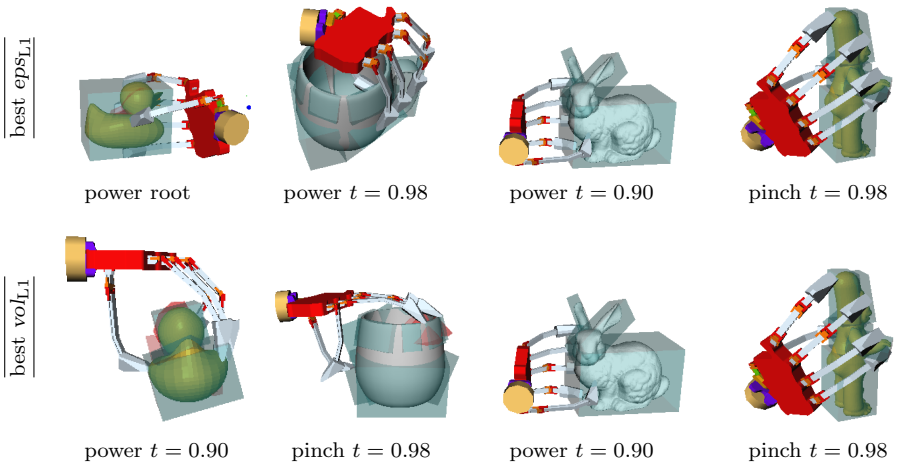


Figure 8: Best grasps for the experimental Karlsruhe hand grasping results in Tab. 3. Upper row: best eps_{L1} grasps. Bottom row: best vol_{L1} grasps.

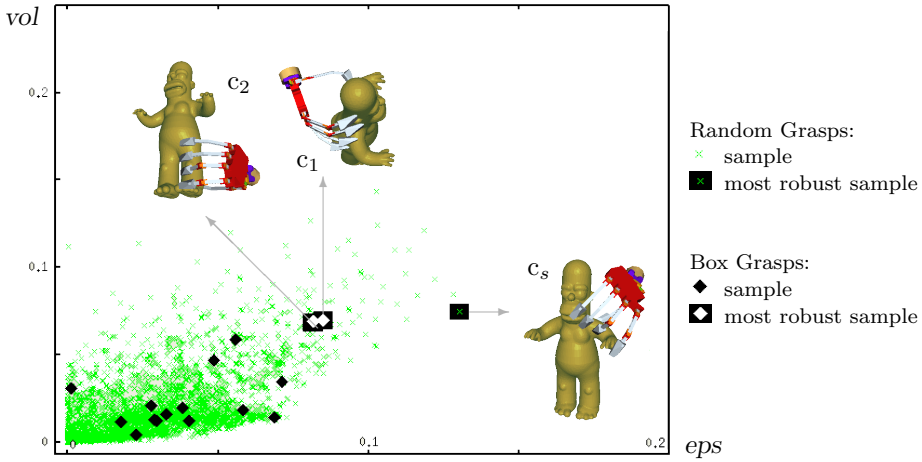


Figure 9: Grasp quality space for the Homer model. Note the depicted samples as a contrast between box grasps and random grasps.

We conclude this experiment with the observation that generation of pre-grasp hypotheses from box-based face representations reduces the number of hypotheses drastically (Huebner *et al.*, 2008). In that paper, a random pre-grasp generation included 22104 hypotheses. Though box decomposition effectively produced only very few valid hypotheses (usually <50), they still feature good grasp quality. A demonstration of the comparison can be seen in Fig. 9.

5.2 Introducing Higher-Level Dependencies

It has been shown hereby that box shapes give efficient clues for planning grasps on arbitrary objects or object parts. For most of the robot tasks we envision, it should be sufficient to find one of the stable grasps, not necessarily the most stable one. Additionally, the part-describing box concept enables grasp semantics to be integrated in the representation, e.g. 'approach the biggest part to stably move the object' or 'approach the smallest part to show a most unoccluded object to a viewer.' The description of an object by a shape-based part representation, which is claimed to be necessary for this kind of task-dependent grasping, is made available, as also needed as a criterion what grasp is the 'best' in terms of a given task.

To briefly refer to the box decomposition approach, a compact box set

$$\mathcal{B} = \{B_1, \ldots, B_n\} \tag{4}$$

encloses a set of 3D points and thereby offers a primitive shape approximation. For each box B_i in the set, we focus on its six rectangular faces

$$\mathcal{F}_i = \mathcal{F}(B_i) = \{F_{(i,1)}, \ldots, F_{(i,6)}\}. \tag{5}$$

In the last section, each face spawned up to four grasp hypotheses by using the face normal as approach vector and the four edges as orientation vectors, using a pre-defined grasp. Thus, we can define the overall set of hypotheses emerging from the box representation as

$$\mathcal{H} = \{H(F)|F \in \mathcal{F}(\mathcal{B})\}, \quad \text{with} \quad H(F_{(i,j)}) = \{F^0_{(i,j)}, F^{90}_{(i,j)}, F^{180}_{(i,j)}, F^{270}_{(i,j)}\} \tag{6}$$

In this section, we will extend this framework by introducing several grasp selection criteria, whereof each is based on a different dependency. In each of them, the matter of 'good' is connected to very different dependencies, e.g. a task dependency might vote for a particular box, or a view-point dependency might vote for particular faces only. Technically, all the dependencies will filter out certain $F^k_{(i,j)}$ from $\mathcal{H}$ to aim for an even smaller set of valid grasp hypotheses.

5.2.1 Task Dependencies – restricting $\mathcal{B}$

The dependency on a given task is a most important issue in grasping, demonstrating that 'best' grasps do not have to be the most stable ones. Picking up a cup from above will be unsuitable for the task of filling the cup, in the same way as a very stable full-enclosing grasp will be unsuitable for handing over or presenting the cup to someone else. Application of such re-usability semantics by defined keep-out zones has been proposed in (Baier and Zhang, 2006). Object properties like hollowness are hard to detect by the state-of-the-art vision systems, as also are high-level properties like *filled* or *empty*. Our box set method allows intuitive mapping of less complex actions to simple box properties.

Given a box set (Eq. 3), one can easily compute criteria like the overall mass center (assuming uniformly distributed mass density), volume and dimension of a box, or the relations between boxes. For example, we can define the *outermost* or *innermost*, the *largest* or *smallest*, the *top* or the *bottom*, etc., or even rank the boxes according to these criteria. Given a task, we can easily map an action like *pick-up*, *push*, *show*, *rotate*, etc., to a selected box. For example, in order to *pick-up* something to place it somewhere else, it may intuitively be a good choice to grasp the *largest* box. When showing the same object to a viewer, it may be better to grasp the *outermost* box instead.

Similarly, different tasks can be mapped to grasp configurations. We already introduced two of these in Section 5.1: the power grasp, which approaches a box until contact, retreats a bit and than closes fingers simultaneously, and the pinch-grasp, which approaches the box until it is in position to close fingers and contact the box most centrally. One might extend this idea towards the selection of different grasp pre-shapes (Cutkosky, 1989), or even the selection of controllers for different tasks. In fact, Prats *et al.* (2007) also use box representations for task-oriented grasping with hand pre-shapes and task frames. However, they assume geometrical knowledge about each object (using a database of 3D models) and structural and mechanical knowledge about a task, e.g. 'turning' a door handle.

5.2.2 Box Face Visibility – restricting $\mathcal{F}$

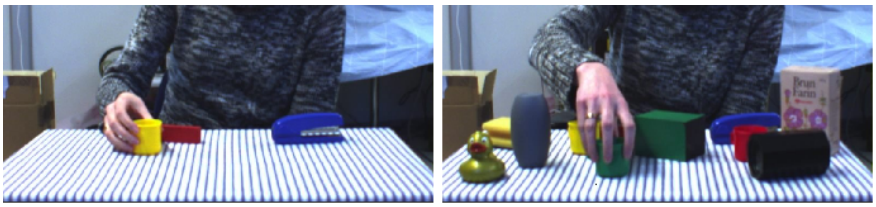
Each box provides six rectangular faces in 3D space (Eq. 4). We have to consider that incomplete data is produced by a single sensor view of an object, since the back of the object is not visible. Thus, box decompositions are clearly view-dependent and do only envelop visible data points. For this reason, it may be helpful to take into account only those faces that are visible from the viewpoint. Here, 'visibility' is defined as the face being oriented towards a viewpoint, not being visible in case of occlusion by other objects. By definition, objects placed on a table will never be approached from the bottom, as this face is generally not oriented towards an external viewpoint. We see another motivation for a face visibility check considering between an end-effector, i.e. a gripper, and the object. Intuitively, humans tend to use grasping movements that involve minimum energy effort (Alexander, 1997).

We have performed a simple experiment that showed some evidence for this: Test persons had to grasp various objects on a table to describe their appearance, thus the task of grasping was implicit. It showed up that in case of cups, the handle was mostly pinch-grasped when it was orientated towards the human hand, while otherwise the cup body was power-grasped (see Fig. 10). Though this experiment is not compelling in terms of a psychophysical evaluation and will therefore not be described any further, it is intuitive in the same way as the viewpoint face check. Most people would not grasp an unoccluded object from the backside, even if this might produce the most stable grasp. Introducing viewpoints for the end-effectors of the robot can handle this issue. Valid faces can thereby be selected by being accessible from a given end-effector viewpoint, even if one end-effector might be busy, e.g. holding another object.

5.2.3 Box Face Occlusion and Blocking – restricting $\mathcal{F}$ and $\mathcal{H}$

While the visibility criterion is a check for orientation of faces towards a camera's or an end-effector's viewpoint, occlusions and blockings between faces in the box set are also considered. As an example, grasping the head box of the Homer model (revisit Fig. 6) from the bottom is not profitable, since this face is 'occluded' by the body box. The corresponding face is then removed from $\mathcal{F}$. One may also classify other grasps on the head as being unprofitable. Imagine a grasp towards the head box B_1 from one of the sides. The fingers will not contact the approached face $F_{(1,a)}$, but two of its neighbors, $F_{(1,b)}$ and $F_{(1,c)}$, depending on the hand orientation $k \in \{0, 90, 180, 270\}$. We then define a grasp hypothesis $F^k_{(1,a)}$ as 'blocked' in this grasp orientation, if $F_{(1,b)}$ or $F_{(1,c)}$ is occluded, and remove it from $\mathcal{H}$.

This technique has proven to be very effective in reducing the number of hypotheses. Technically, the detection of opposing faces is more complex than the visibility check, since each face of a box has to be compared to each other valid face in $\mathcal{F}$. Therefore, it may form the end of the heuristical selection sequence. However, even if two faces face each other, this is usually not a sufficient condition to mark the face as occluded, since a finger may fit in-between. The



(a) An example of no occlusion / occlusion.

(b) An example of orientation (handle towards / averted).

(c) An example of position (closer to left hand).

Figure 10: Different test persons were asked to describe objects on a table, whereof some were cups. The grasping task itself was implicit. The dependency of the grasp on extrinsic attributes, like occlusion (by other objects), orientation, or position (also relative to the person) is clearly traceable. The first two examples (left, center) show different runs and setups, respectively, in the third one, the right-hand grasp was even rejected in favor of a left-hand grasp.

handling of such situations would demand additional computational effort. For this reason, and since a more extensive restriction reduces the number of hypotheses drastically, we currently remove all occluded and blocked hypotheses from our selection.

5.2.4 Reachability and Graspability – restricting $\mathcal{H}$

If there is information available about the embodiment and the kinematics of the robot platform, i.e. its arm and gripper, it is possible to use graspability and reachability criteria to further reduce the number of hypotheses. In terms of graspability, our approach already compares the gripper aperture with the width of the approached face. In terms of reachability, an inverse kinematics solver might be applied for dropping hypotheses that are not reachable with one of the available hands. Practically, the integration of an IK solver is still an issue of future work.

5.2.5 Experimental Results

To demonstrate the heuristical hypothesis reduction techniques discussed in Sections 5.2.1 to 5.2.4, we will apply them on the Homer model in GraspIt!. The results of a 'show' instruction on the Homer model can be tracked in Fig. 11, given the following sequence:

Section 5.2.1: Rank the boxes B_i and select grasp pre-shape according to a given task,

Section 5.2.2: restrict the produced face set $\mathcal{F}(B_i)$ according to a given viewpoint,

Section 5.2.3: and restrict the thereby produced hypothesis set $\mathcal{H}$ by removing occluded faces and blocked hypotheses.

Section 5.2.4: Also reduce the new hypotheses set $\mathcal{H}$ by removing those that are wider than the gripper aperture.

Section 5.1.1: For the hypotheses that 'survive' the restriction process, we repeat the simulated evaluation process discussed in 5.1.1, i.e. grasping them in simulation and select the best one for visualization.

In Fig. 11b, one can see the process of grasp hypotheses reduction from occlusion and blocking in the box constellation, e.g., all four grasp hypotheses towards the chest have been rejected due to blocking, since there are occluding

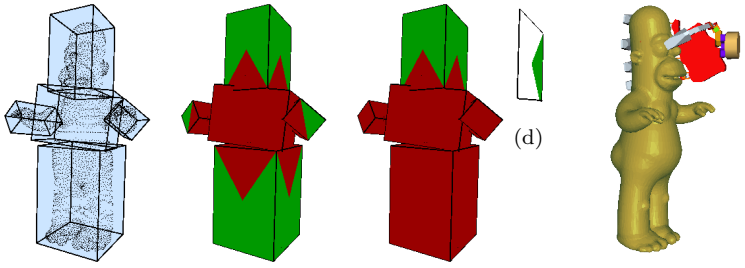


Figure 11: (a) Point cloud and box constellation of the Homer model. (b) Resulting hypothesis restrictions from viewpoint, occlusion and blocking. Invalid hypotheses are depicted by dark (red) triangles, valid ones by light (green), where the inward vertex correlates to the hand orientation (the thumb). (c) According to an *outermost* ranking, the head box is prior. Only valid hypotheses of this box are kept. (d) The highlighted triangle corresponds to the most stable hypothesis of those in (c). (e) Visualization of the final grasp on the original model. Viewpoint has been changed to better show finger contact locations.

boxes at the sides (head, arms and legs). Fig. 11c shows the influence of ranking, introducing the task. We simply link tasks to certain rankings and grasp preshapes, e.g.

$$\begin{aligned} &(\text{T1}) \quad task : pick \quad &&\rightarrow box : largest, grasp : power, \\ &(\text{T2}) \quad task : show \quad &&\rightarrow box : outermost, grasp : pinch. \end{aligned}$$

Since in the example, we chose a 'show' instruction, boxes are ranked according to an 'outermost' criterion. Beginning with the highest ranked box, the head in the example, valid hypotheses are tested in GraspIt!. Out of those, the one that gave the best grasp quality values is shown in 11d-e.

Concluding this experiment, we have shown how heuristical dependencies can be integrated into the box framework. It may be mentioned that all calculations necessary are purely geometrical problems on faces, points and volumes. Like the whole grasp selection process, visibility, occlusion and blocking are currently computed software-based, one might think about taking advantage of graphical processors to speed up and optimize the geometrical operations.

The simplicity of the presented operations and rule-based connections between tasks and parts still offers space for optimizations. The rules might be optimized towards other criteria and other attributes, e.g. the faces themselves, such that the system might be triggered to approach an arm for a 'show' instruction. The computations of blocked faces might be optimized in such a way that it could be checked if a finger fits between two opposing boxes.

Finally, the described heuristical and geometrical processes are restrictions of the hypotheses, and thus only further reduce the set $\mathcal{H}$ of all hypotheses to a smaller set. The final grasp in Fig. 11 resulted by comparing simulated grasps on the model and taking the most stable one. To approach this issue, and find a best grasp hypothesis H^* from the box representation only, we will now extend the system by enriching each face representation with a 2.5D projection image. This will allow for learning of grasp qualities instead of grasp simulation.

5.3 Projection Grids and Learning – finding H^*

We can now use the presented box decomposition algorithm to perform a box approximation of the point cloud and reduced hypotheses according the previously presented heuristics. Those were aiming at reducing the number of grasp hypotheses according to the task, and 3D orientation or 3D shape of the object. Also the size, i.e. the dimensions of a face, was considered. However, there is usually a set of remaining hypotheses $\mathcal{H}'$ after the restriction steps, from which we would like to select one final 'best' grasp H^*. Our current approach to this issue is learning of grasp qualities from 2.5D shape projections.

Considering a box and the points that it envelopes, each face produces a projection of the points onto the face plane. In fact, these projections were already computed for best cut detection (see Section 4.3 and 4.4). Discretization was made by dividing each face into equally sized cells, thus projections were represented as dynamically sized binary grids. Additionally, opposing faces shared the same projection grid. These grids kept binary information and were

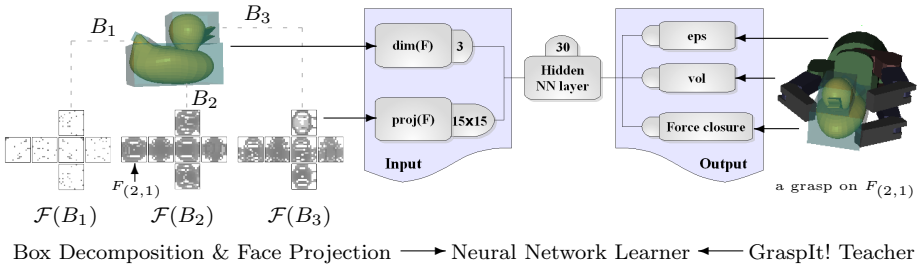


Figure 12: The applied neural network structure holds 228 input, 30 hidden and 3 output neurons. As an input, a face projection F plus its box dimensions *dim(F)* are fed into the network, since faces are normalized to 15×15. *eps* and *vol* are the grasp quality measures that GraspIt! delivers. The force closure is also learned separately even if it equals ($eps > 0$). From this model, off-line learning of grasp qualities from face representations is made available.

dynamically sized. To adapt this representation and enrich it, we now compute linear information, i.e. minimum distance information to the face plane, in a normalized, fixed-sized grid. Thus, all six projections have to be stored instead of three, since opposing faces do not share the same projection anymore. In return, this representation both allows analyzing the 2.5D depth map of each face and fulfills the input space conditions of a classical neural network learner like the one we will use here (see Fig. 12).

By providing the two quality measures *eps* and *vol* which were introduced in Section 5.1, GraspIt! (Miller and Allen, 2004) is used as a teacher for a supervised network, estimating the stability of a grasp from a given face F and its 2.5D projection grid *proj(F)*, respectively. Since due to normalization in width, height and depth, information about the dimension of F is lost, the box dimensions *dim(F)* are added in terms of three additional neural network inputs. A sample of projections and the neural network structure is shown in Fig. 12.

5.3.1 Final Grasp Decision and Learning

Finally, we have to decide *where* and *how* to grasp after initially having reduced the hypotheses to a smaller set. The *'where'* component equals a decision on grasping one of the faces with one orientation. To do this, we apply the neural network approach presented above. The face projections of the remaining hypotheses are fed into the net that has been previously off-line trained with artificial examples. In our experiments, these are mostly complete models which have been processed by the algorithm and their projections grasped in the grasp simulator. By providing the two quality measures, GraspIt! was automatically used as a teacher for the supervised network, estimating the stability of a grasp on a given 2.5D projection grid. After sorting out hypotheses that do not result in good force-closure response (third network output) larger than 0.5, we decide for the one hypothesis with maximum *vol* grasp quality (second network output). According to the definition of grasp hypotheses in (5), this is

$$\mathcal{H}^* = \underset{F^{\phi}_{(i,j)} \in \mathcal{H}' \wedge \mathcal{N}_{fc}(F^{\phi}_{(i,j)}) \geq 0.5}{\arg\max} \mathcal{N}_{vol}(F^{\phi}_{(i,j)}), \tag{7}$$

where $\mathcal{N}_x$ is the corresponding output of our neural net $\mathcal{N}$ and $\mathcal{H}'$ the set of hypotheses after the heuristical selection processes.

As briefly described above, the *'how'* component is currently a direct mapping between a manually given task description (e.g. pick, show) to a grasp pre-shape (e.g. power, pinch). An extension which approaches the kinematic properties of the applied gripper and connects them to the projection, in order to estimate good finger contact positions by a set of quality measures, has been proposed in (Geidenstam *et al.*, 2009). However, this approach will not be used here to keep the experiments clear and independent of any gripper kinematics.

At this point, we have completed a method of selecting a 'best pre-grasp hypothesis' $\mathcal{H}^*$ from a 3D point cloud, using box decomposition. In terms of 'best', this not only considers 'good' stability, which is to some extent learned and supported by the neural network, but also the proposed heuristical dependencies, i.e. being 'good' in relation to the task at hand, the gripper embodiment, or the pose of the object.

6 Implementation

In this section, we will present an experiment showing the capabilities of the presented techniques beyond those of artificial models. Earlier described experiments have been performed in simulation, thus one interest is to test the box decomposition on real 3D data which is influenced by natural dense stereo noise and incompleteness.

6.1 Experimental Setup

Our experimental 3D data will be produced from disparity using the four-camera Armar-III stereo head shown in Fig. 13b. More information about the whole Armar-III robot, a humanoid platform at the University of Karlsruhe, can be found in (Asfour *et al.*, 2006) and on `www.paco-plus.org`. The whole system consists of two foveal cameras for recognition and pose estimation, and two wide field cameras for attention. We proposed a grasping strategy for known objects, comprising an off-line, box-based grasp generation technique on 3D shape representations on the complete platform in (Huebner *et al.*, 2009). A general outline of our system going beyond the scope of this paper, is presented in (Bohg *et al.*, 2009). our interest here will be the practical processing of the proposed heuristical selection and a learning mechanism, including the considered decisions on task, view-point, shape and size properties on unknown objects. Along the presentation of the experiments, we will point out connections to grasp learning and shape classification.

As is depicted in Fig. 13c, 2D object segmentation from a single image, as an optional part, will boost further performance of the perception system since

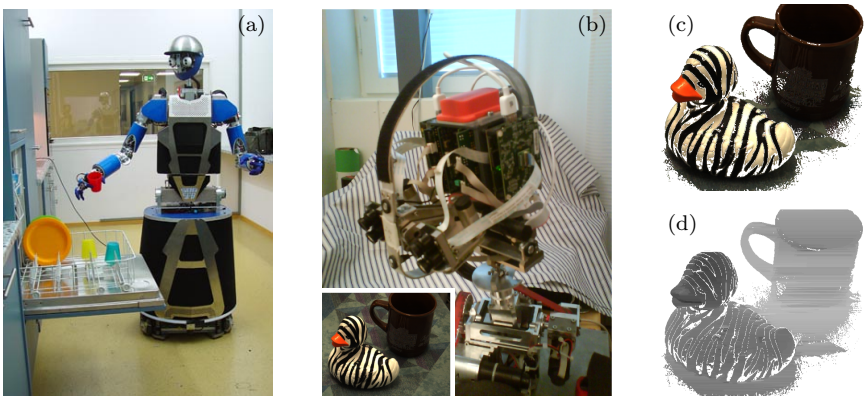


Figure 13: (a) The Armar-III humanoid platform (Asfour *et al.*, 2006) used in the PACO-PLUS project. (b) A duplicate of the Armar-III stereo head, used in our lab, including a clipped region of an acquired rectified image. (c) Result of image differencing related to an image without the objects. This mask is applied in (d) to results from the disparity processor. Note that apart from white being the mask region, intensity corresponds to distance to the viewpoint.

local (2D) shape information also supports the binocular fixation of the system. As an example, we assume the background image to be given in a static head scenario. The object is segmented by image differencing and the 3D point cloud from stereo can be masked easily to include only these points, as common uncertainties and noise in the environment can be removed. Additionally, an estimate of mean disparity can be computed from which the disparity algorithm benefits. More sophisticated methods for object segmentation in the 2D image have not been implemented in this system yet, but are clearly available in the literature. Promising in this context are techniques like object segmentation from attention or object segmentation from manipulation. However, even a simple differencing subtraction method already demonstrates that a step of 2D segmentation is factually valuable for the whole system.

For the purpose of grasping based on 3D shape, such 2D segmentation may be helpful, but not sufficient. In general, and as long as there is no high-level reasoning system to infer 3D shape properties for unknown objects from a 2D image only, a mug on the cover of a magazine will not be distinguishable from a real cup on the table without any further analysis of 3D data. Additionally, estimation of an object's size or shape in three dimensions is intuitively valuable for its manipulation. 2.5D segmentation will help us to distinguish between objects in three dimensions with options beyond those of 2D segmentation.

General high-dimensional segmentation, be it in 3D space or even enriched with color space information, has high complexity and drawbacks. However, efficiently shortcutting this problem was successfully demonstrated through the assumption of planar surfaces (Rusu *et al.*, 2008; Triebel *et al.*, 2007). In a number of manipulation scenarios, as also in ours, we can assume that manipulable objects are very commonly placed on a horizontal plane, e.g., a table. In our

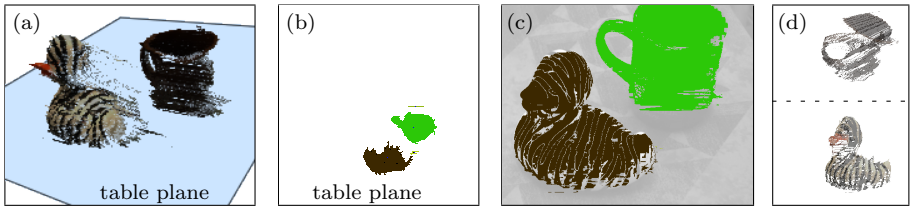


Figure 14: (a) Scene reconstructed from Fig. 13d, purged by table plane assumption. (b) Objects deduced from 2.5D segmentation on table plane projection. Note that the table is not detected, but just the (infinite) table plane visualized. (c) Reprojection of the segmented 3D objects (d) to the image.

current system and scenario, where there is only one table for reasons of simplicity, detecting the table plane can either be done by Hough Transformation in 3D, or, and both much more efficiently and online, by integrating the vector of gravity. The vector of gravity corresponds to a good estimate of most table planes' normals, and can be deduced with minor effort from either the acceleration sensor or the kinematic chain of the head (see also Fig. 13b). Given a table plane, the 3D scene can further be purged by removing points lying on or below this plane. See Fig. 14 for an example.

In Fig. 14, the 3D data processing from stereo images, image differencing, disparity processing, table plane reduction and 2.5D segmentation is visualized. Both detected 'objects' are clearly influenced by incompleteness, observable by some holes and by the backsides which are not visible. Additionally, and due to disparity processing, there is noise in disparities, as also some false assumptions, e.g. the uniformly colored top of the mug has been interpolated to a flat surface. Most effects of these uncertainties become clear in Fig. 14d, where the 3D model of both objects are shown from a different viewpoint. Of course, if objects are standing close to each other, 2.5D segmentation will detect them as one. We see this issue to be approached by a more sophisticated 2D segmentation module or even as a trigger for explorative manipulation through expectation and surprise.

6.2 Technical Issues and Data Structure

In this subsection, we will briefly describe the technical data structure that has emerged during the development of the framework proposed in Sections 4 and 5. While an input to the box decomposition can be an Inventor file format (supported by GraspIt!) or a Coordinate file format (list of 3D values), such Coordinate files can optionally be extended by assigning XY image coordinates and corresponding color values to each 3D point.

The box decomposition can be controlled by a simple configuration file. When running an experiment, the framework connects to GraspIt!, executes the box decomposition and optionally grasping experiments, before saving the final decomposition as a result.

Due to re-usability reasons, this output is divided in both an .xml file for representation of shape, and .pgm image files for projection images. Projection images will be described in detail in the next section. Concluding, the .xml file

carries the following information for each ⟨*BoxDecomposition*⟩:

- For each ⟨**Box**⟩ in a ⟨*BoxDecomposition*⟩, its ⟨*ID*⟩, ⟨*HierarchyLevel*⟩, ⟨*Parent*⟩ and ⟨*Child*⟩ box IDs and a flag if it is a leaf box, ⟨*IsLeaf*⟩, is stored. Additionally, the representation carries information about the ⟨*Volume*⟩, the ⟨*Center*⟩ point, the 3 box ⟨*Dimensions*⟩, the 8 box ⟨*Corners*⟩ and the 6 ⟨*Faces*⟩.
- For each ⟨**Face**⟩ in a ⟨*Box*⟩, its ⟨*ID*⟩, ⟨*Center*⟩ point, outward-directed ⟨*Normal*⟩ vector, the 3 face ⟨*Dimensions*⟩, and a flag if it is an occluded face, ⟨*Occluded*⟩, is stored. Additionally, the representation carries information about the 4 face ⟨*Corners*⟩ and if the edges between them are blocked, ⟨*EdgeBlock*⟩.
- In case a ⟨**Face**⟩ belongs to a leaf box, a path to the ⟨*ProjectionFile*⟩ is provided, as also a ⟨*ShapeClassID*⟩ and a flag if the projection is hollow, ⟨*ShapeClassHollow*⟩. The first will be used for different purposes of grasp learning that will be discussed in the next section. The latter two tags are prepared for evolving shape classification which is still under examination. In addition, ⟨*GraspHypotheses*⟩ are assigned to each leaf box.
- Optionally, the representation can carry information about the original ⟨**Points**⟩, including 3D position (in model or camera frame), 2D origin in the image, as color in the image.

Note that the representation provides information about the whole decomposition process and hierarchy. The box with ID 0 will ever be the origin of the hierarchy tree, being the root box.

To be able to visualize already processed decompositions and their properties, a graphical interface has been developed. This interface which only reads the above mentioned data, namely .xml and .pgm images, will be used in the following to visualize most of the experiments.

6.3 Experimental Box Decomposition and Grasping

Starting from the two segmented object point clouds in Fig. 14d, we trigger the framework modules (decomposition, hypotheses generation and pre-grasp generation) with the parameters shown in Fig. 15.

Decomposition results for the treated examples are presented in Fig. 16, each (a) and (b). As one can see, both examples are decomposed in a very similar way, resulting in three leaf boxes each. Though not ideally, handle of the cup and head of the duck are separated from the other parts. Decomposition time strongly depends on the complexity of the model, but is linear in relation to splits. In our cases at hand, each split step takes around 4 seconds.

Hypotheses results for the treated examples are presented in Fig. 16, each (c) to (f). (c) shows the occluded and blocked components opposed to the valid

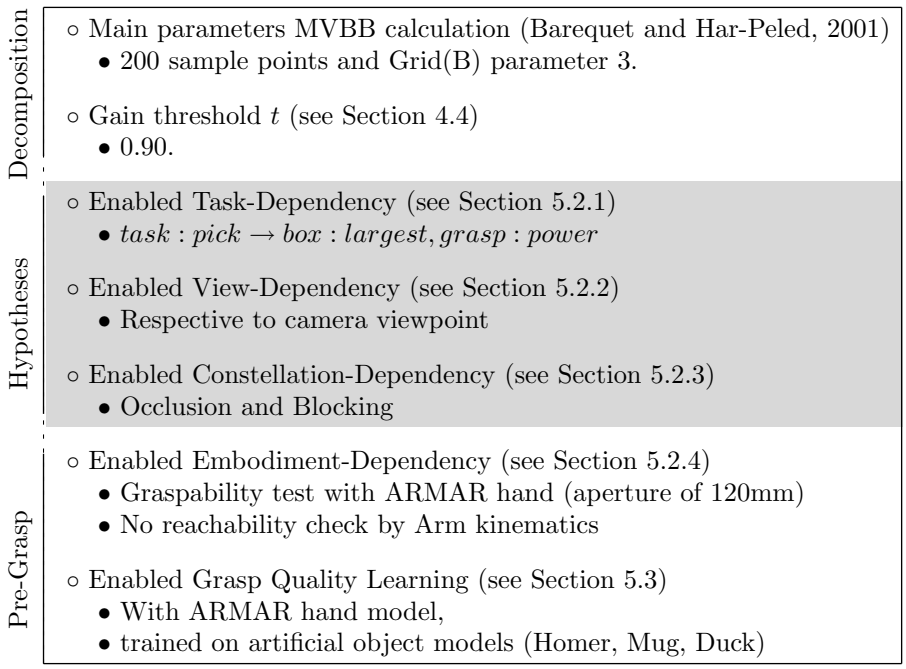


Figure 15: Algorithm parameters for the experiment.

ones. Note that also a lot of backsides are invalid, since the viewpoint in the scene is almost equal to the one in the sketches. (d) provides a view on the face projections of all faces (also invalid). On those, the network is tested at a later stage. While in (e), all valid pre-grasp hypotheses are depicted, (f) only shows those that relate to the chosen task-dependency $box : largest$.

The final **Pre-Grasp** is determined by applying the trained network on the hypothesis set in (f). The pre-grasp that results in best grasp stability estimate is chosen and performed (in GraspIt!). (g) shows the approach position with fully opened hand, while (h) shows the state after approach and grasp.

6.4 Discussion

The presented results point to a couple of issues to discuss. As one can see, the decomposition of objects into parts is only partially convincing. This is caused by general features of using vision and dense stereo disparity, using a dynamic programming approach from (Scharstein and Szeliski, 2002, 2007) for point cloud generation.

In particular, artifacts appear in the point cloud for uniformly colored regions, since unmatched image points can only be interpolated. This issue makes the cup (Fig. 16b) appear to be closed at the top surface, as also the front shape is quite erroneous. As an additional issue, the only data observable is the one

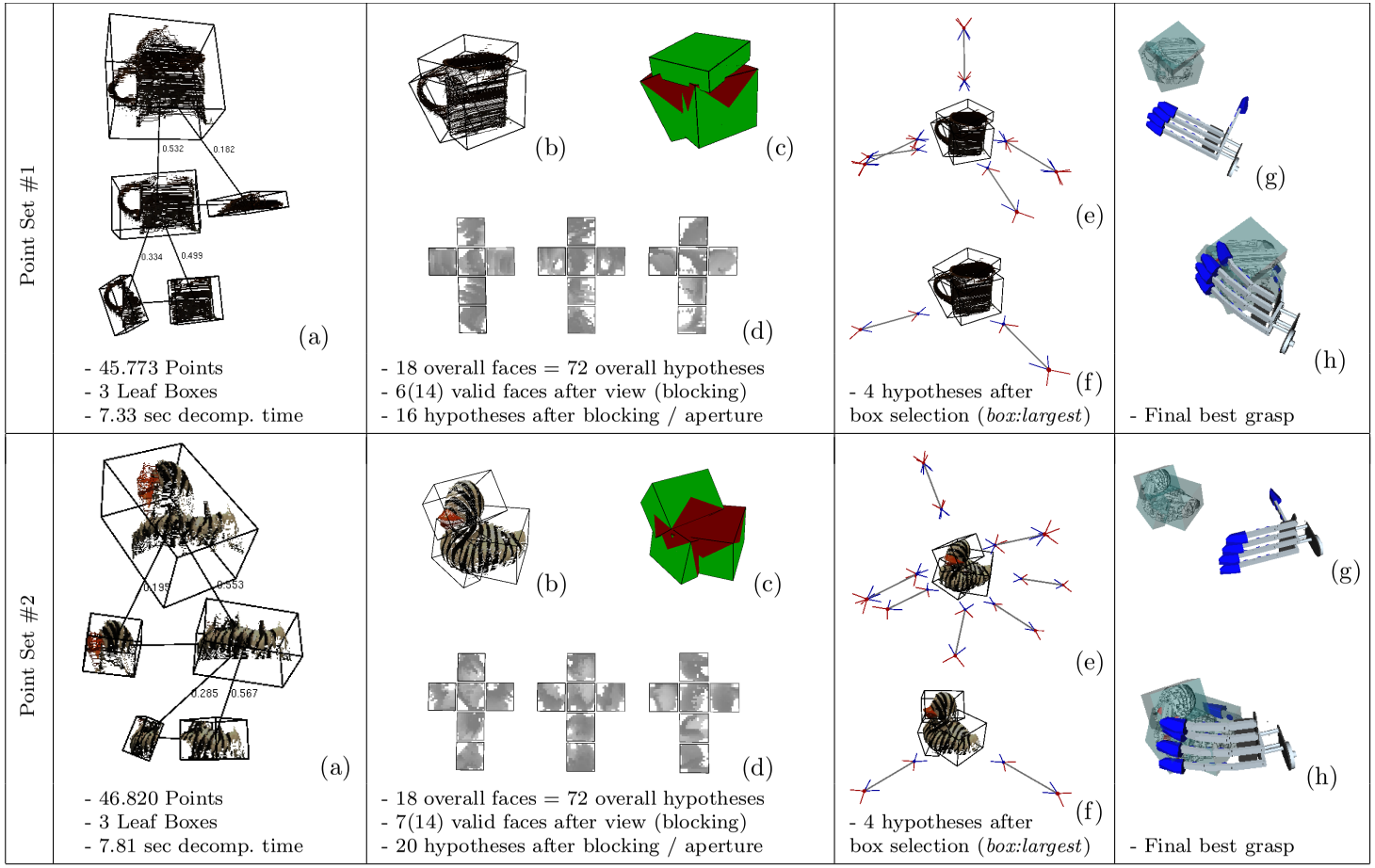


Figure 16: Process for the two point clouds produced in Fig. 14d. For description, see text in Section 6.3.

seen from only one view, causing the point cloud to be highly incomplete. Due to effect that more a 3D surface than a 3D model is considered by this, also the decomposition fits boxes to those surfaces mainly and thereby looses valuable information about 3D shape. As an example for this, the cup's 'cylindrical' part (Fig. 16b) is finally represented by two orthogonal surfaces which are separated by the decomposition. In comparison to experiments which applied complete models from a database of known objects (Huebner *et al.*, 2009), performance is therefore not as good in terms of shape approximation.

The various steps of hypotheses reduction were intentionally set relatively strictly in the experiment. The view-point heuristic removes all hypotheses that are not visible from the view-point, relating to the issues above. However, in case of complete models, applying this heuristic would be rather unliked: grasps from the backside are valid and interesting, especially when front-hypotheses are not well-rated. The task-related box selection is clearly influenced by the unfavorable box decomposition. Note, however, though in Fig. 16f only hypotheses connected to one box are visualized, all of the hypotheses in Fig. 16e are considered through ranking both related to size of box and grasp quality estimate. This will prevent an empty result if there is no good hypothesis for, e.g. the largest, box by switching to the second-largest, and so on. It can be noticed that the hypotheses set is very restricted. However, the flexibility of the framework allows to disable or enable heuristics to control the size of the final hypotheses set.

For the final pre-grasp generation from a set of hypotheses, the trained network estimates grasp quality from the samples shown in Fig. 16d. The best ranked are performed in Fig. 16g and 16h with a right-hand gripper. As one can see, the grasps are awkward having in mind the embodiment of the right hand on the right arm of the humanoid platform. This problem can be solved by integrating an inverse kinematics solver which is then able to rate hypotheses by their reachability, as also if the left or the right hand can be used. This technique was applied in (Huebner *et al.*, 2009). Another issue in this context is that one separate neural network has to be trained for each hand and each type of grasp pre-shape type (e.g. power-grasp, pinch-grasp). It has not been analyzed yet if, and up to which degree, grasp quality measures are generalizable over such options. In the experiment, only one network was trained for the right hand gripper model and the power grasp (see Section 5.1).

Despite these issues, the framework presented in this paper is one of few that approach 3D shape approximation from dense stereo data instead of 3D range data or 3D meshes for the purpose of grasping. The source of data for the presented algorithms is arbitrary, as long as it represents 3D point clouds. Nevertheless, the high complexity and manifold difficulties of a vision-based approach have been pointed out. However, we believe that the proposed framework is flexible enough to be extended toward such issues.

7 Conclusions

We presented the continuation of box approximation for the purpose of robot grasping. We specified the core algorithm and specific extensions of connecting box shape approximation and grasp hypotheses generation in earlier work (Geidenstam *et al.*, 2009; Huebner and Kragic, 2008; Huebner *et al.*, 2008, 2009). In our approach, we combined several motivations known from the shape approximation and grasping literature. In short, we prune the search space of possible approximations and grasp hypotheses by rating and decomposing very basic shapes, which intuitively corresponds to the 'grasping-by-parts' strategy. In this paper, we focused in greater detail on all the parts of an entire framework taking advantage of the very simple shape representation of boxes. Starting from boxes and their faces that the core algorithm produces, we extended the idea of 'grasping on boxes' towards an applicable grasping strategy. This strategy includes various heuristical selection criteria based on efficient geometrical calculations, as also learning from off-line simulation. Basic task-dependencies have been included in this process. We see the strength of our approach in its simplicity and its modularity. The simplicity is obvious by using boxes and faces in 3D space. Geometric calculations are much more easy to do in contrast to more sophisticated shape primitives like superquadrics. As presented, boxes and faces can additionally take advantage of linear shape projections. The modularity is established by mostly independent criteria and heuristics that complement each other and flexibly leave space for adaptation and extension.

There are many possibilities to extend and optimize the current framework. Considerations have to be made for the neural net structure, e.g. if it is better to extend the learning to grasp qualities dependent on the chosen grasp pre-shape, i.e. setting three quality outputs for *each* available grasp pre-shape. Additionally, the simulation part for learning is currently done using static simulation. Thus, contact will stay static between gripper and object, while in dynamics, and reality, the object pose will change dependent on the force applied to it. We are working on this issue also with regard to what we called the *grip component* (see Section 3.1). As discussed, we are aware that our approach is a *pre-grip component* based on very robust shape information. The grip component, as an additional module, would contribute in terms of fine correction based on haptic feedback (Tegin *et al.*, 2009). We see haptic feedback and exploration also as a solution to approach the problem of incomplete models acquired from stereo vision. Merging the 3D data from stereo with 3D data from haptic contact points along the backsides of objects might therefore be an issue of future work.

As another issue, the projection of an object onto the box faces ignores to some extent the real 3D shape of the object, disregarding correct surface normals of the object in the grasp planning. Thus, there is a possibility that planned grasps are infeasible, which addresses the limitation of the proposed planning. In (Geidenstam *et al.*, 2009), we tried to approach this issue using explicit gripper kinematics in order to analyze finger position estimations on the projections, extending work of Morales *et al.* (2003).

As future work, one could also imagine higher-level part classification to

infer suitable grasp pre-shapes from a wider variety of primitives. Given all three projections of a box or the enclosed point cloud itself, one could try to classify the represented shape, which is ought to correspond to an object part. This relates to work on view-based object (part) representation. Classification of shape is a beneficial, but also complex task, as additionally, the box constellation might be very different as influenced by noise, perspective view and uncertainties. For the purpose of grasping on faces, this is not a very severe problem, while in part and object classification, it probably will be. Evaluations of these high-level ideas are not a topic of our short-term goal. However, we are planning to evaluate a model-based part-matching technique like in (Detry *et al.*, 2008), matching 3D data of shape primitives (e.g. cylinders, spheres, cones) to the point subsets generated from the box decomposition.

Another high-level issue is task dependency. There are different task types on which a grasp might depend. Just to pick up a cup and place it somewhere else might yield a different grasping action as picking up the cup to show it or hand it over. These grasp semantics can be mapped to boxes in the set, e.g. 'grasp the largest box for a good force grasp to securely move the object', 'grasp the smallest box for a good pinch grasp to show a most unoccluded object to a viewer/camera' or 'grasp a very outlying box so that another human / robot hand can overtake the object easily'. The latter semantics are quite valuable for applications that are based upon interacting with objects *before* the exploration and recognition stage, such as Ude *et al.* (2007). An issue like this has to be analyzed in a wider scope than the one given in this paper, preferably in a complete system architecture as proposed in (Bohg *et al.*, 2009).

Acknowledgements

This work was supported by EU through PACO-PLUS, IST-FP6-IP-027657.

References

Aarno, D., Sommerfeld, J., Kragic, D., Pugeault, N., Kalkan, S., Wörgötter, F., Kraft, D., and Krüger, N. (2007). Early Reactive Grasping with Second Order 3D Feature Relations. In *ICRA Workshop: From Features to Actions*, pages 319–325.

Alexander, R. M. (1997). A Minimum Energy Cost Hypothesis for Human Arm Trajectories. *Biological Cybernetics*, **76**(2), 97–105.

Amenta, N., Choi, S., and Kolluri, R. (2001). The Power Crust. In *6th ACM Symposium on Solid Modeling and Applications*, pages 249–260.

Andrew, A. M. (1979). Another Efficient Algorithm for Convex Hulls in Two Dimensions. *Information Processing Letters*, **9**, 216–219.

Asfour, T., Regenstein, K., Azad, P., Schröder, J., Bierbaum, A., Vahrenkamp, N., and Dillmann, R. (2006). ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 169–175.

Baier, T. and Zhang, J. (2006). Reusability-based Semantics for Grasp Evaluation in Context of Service Robotics. In *International Conference on Robotics and Biomimetics*, pages 703–708.

Barequet, G. and Har-Peled, S. (2001). Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions. *Journal of Algorithms*, **38**, 91–109.

Bicchi, A. and Kumar, V. (2000). Robotic Grasping and Contact: A Review. In *IEEE International Conference on Robotics and Automation*, pages 348–353.

Biegelbauer, G. and Vincze, M. (2007). Efficient 3D Object Detection by Fitting Superquadrics to Range Image Data for Robot's Object Manipulation. *Int. Conf. on Robotics and Automation*, pages 1086–1091.

Bohg, J., Barck-Holst, C., Huebner, K., Ralph, M., Rasolzadeh, B., Song, D., and Kragic, D. (2009). Towards Grasp-Oriented Visual Perception for Humanoid Robots. *International Journal of Humanoid Robotics*, **6**(3), 387–434.

Borst, C., Fischer, M., Haidacher, S., Liu, H., and Hirzinger, G. (2003). DLR Hand II: Experiments and Experiences with an Antropomorphic Hand. In *IEEE Int. Conf. on Robotics and Automation*, pages 702–707.

Borst, C., Fischer, M., and Hirzinger, G. (2004). Grasp Planning: How to Choose a Suitable Task Wrench Space. In *IEEE International Conference on Robotics and Automation*, pages 319–325.

Chevalier, L., Jaillet, F., and Baskurt, A. (2003). Segmentation and Superquadric Modeling of 3D Objects. *Journal of Winter School of Computer Graphics, WSCG'03*.

Cutkosky, M. (1989). On Grasp Choice, Grasp Models and the Design of Hands for Manufacturing Tasks. *IEEE Transactions on Robotics and Automation*, **5**, 269–279.

Derbyshire, N., Ellis, R., and Tucker, M. (2006). The Potentiation of Two Components of the Reach-to-Grasp Action during Object Categorisation in Visual Memory. *Acta Psychologica*, **122**, 74–98.

Detry, R., Pugeault, N., and Piater, J. H. (2008). Probabilistic Pose Recovery Using Learned Hierarchical Object Models. In *6th International Conference on Vision Systems, International Cognitive Vision Workshop*.

Ekvall, S. and Kragic, D. (2007). Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning. In *IEEE Int. Conference on Robotics and Automation*, pages 4715–4720.

Fukaya, N., Toyama, S., Asfour, T., and Dillmann, R. (2000). Design of the TUAT/Karlsruhe Humanoid Hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1754–1759.

Geib, C., Mourao, K., Petrick, R., Pugeault, N., Steedman, M., Krüger, N., and Wörgötter, F. (2006). Object Action Complexes as an Interface for Planning and Robot Control. In *IEEE-RAS Int. Conference on Humanoid Robots*.

Geidenstam, S., Huebner, K., Banksell, D., and Kragic, D. (2009). Learning of 2D Grasping Strategies from Box-Based 3D Object Approximations. In *2009 Robotics: Science and Systems Conference*, pages 9–16.

Gibson, J. J. (1977). The Theory of Affordances. *Perceiving, Acting and Knowing: Toward an Ecological Psychology*, pages 67–82.

Goldfeder, C., Allen, P. K., Lackner, C., and Pelossof, R. (2007). Grasp Planning Via Decomposition Trees. In *IEEE International Conference on Robotics and Automation*, pages 4679–4684.

Gottschalk, S., Lin, M. C., and Manocha, D. (1996). OBBTree: A Hierarchical Structure for Rapid Interference Detection. *Computer Graphics*, **30**(Annual Conference Series), 171–180.

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

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

Huebner, K., Welke, K., Przybylski, M., Vahrenkamp, N., Asfour, T., Kragic, D., and Dillmann, R. (2009). Grasping Known Objects with Humanoid Robots: A Box-Based Approach. In *14th International Conference on Advanced Robotics*.

Katsoulas, D. (2003). Reliable Recovery of Piled Box-like Objects via Parabolically Deformable Superquadrics. In *9th IEEE International Conference on Computer Vision*, volume 2, pages 931–938.

Kraft, D., Baseski, E., Popovic, M., Krüger, N., Pugeault, N., Kragic, D., Kalkan, S., and Wörgötter, F. (2008). Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes. *Humanoid Robotics*, **5**, 247–265.

Krüger, N., Piater, J., Wörgötter, F., Geib, C., Petrick, R., Steedman, M., Ude, A., Asfour, T., Kraft, D., Omrcen, D., Hommel, B., Agostino, A., Kragic, D., Eklundh, J., Krüger, V., and Dillmann, R. (2009). A Formal Definition of Object Action Complexes and Examples at different Levels of the Process Hierarchy. Technical report, EU project PACO-PLUS.

Liu, Y. H., Lam, M., and Ding, D. (2004). A Complete and Efficient Algorithm for Searching 3-D Form-Closure Grasps in Discrete Domain. *IEEE Transactions on Robotics*, **20**(5), 805–816.

Lopez-Damian, E. (2006). *Grasp Planning for Object Manipulation by an Autonomous Robot*. Ph.D. thesis, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS.

Lopez-Damian, E., Sidobre, D., and Alami, R. (2005). Grasp Planning for Non-Convex Objects. In *36th International Symposium on Robotics*.

Miller, A. T. and Allen, P. K. (2004). Graspit! A Versatile Simulator for Robotic Grasping. *Robotics and Automation*, **11**(4), 110–122.

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

Morales, A., Chinellato, E., Fagg, A. H., and del Pobil, A. P. (2003). Experimental Prediction of the Performance of Grasp Tasks from Visual Features. In *IEEE/RSJ Int. Conference on Robots and Systems*, pages 3423–3428.

Morales, A., Chinellato, E., Fagg, A. H., and del Pobil, A. P. (2004). Using Experience for Assessing Grasp Reliability. *International Journal of Humanoid Robotics*, **1**(4), 671–691.

Namiki, A., Imai, Y., Ishikawa, M., and Kaneko, M. (2003). Development of a High-Speed Multi-Fingered Hand System and its Application to Catching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2666–2671.

Okamura, A. M., Smaby, N., and Cutkosky, M. R. (2000). An Overview of Dexterous Manipulation. In *IEEE International Conference on Robotics and Automation*, pages 255–262.

Pollard, N. S. (1994). *Parallel Methods for Synthesizing Whole-Hand Grasps from Generalized Prototypes*. Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Pollard, N. S. (2004). Closure and Quality Equivalence for Efficient Synthesis of Grasps from Examples. *Journal of Robotic Research*, **23**(6), 595–613.

Prats, M., Sanz, P. J., and Del Pobil, A. P. (2007). Task-Oriented Grasping using Hand Preshapes and Task Frames. In *IEEE International Conference on Robotics and Automation*, pages 1794–1799.

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M. E., and Beetz, M. (2008). Functional Object Mapping of Kitchen Environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3525–3532.

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

Scharstein, D. and Szeliski, R. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, **47**(1/2/3), 7–42. Microsoft Research Technical Report MSR-TR-2001-81.

Scharstein, D. and Szeliski, R. (2007). StereoMatcher C++ implementation, `http://vision.middlebury.edu/stereo/`.

Shimoga, K. (1996). Robot Grasp Synthesis Algorithms: A Survey. *International Journal of Robotic Research*, **15**(3), 230–266.

Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*. Springer. ISBN 978-3-540-23957-4.

Smeets, J. B. J. and Brenner, E. (1999). A New View on Grasping. *Motor Control*, **3**, 237–271.

Solina, F. and Bajcsy, R. (1990). Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations. *Pattern Analysis and Machine Intelligence*, **12**(2), 131–147.

Tegin, J., Ekvall, S., Kragic, D., Iliev, B., and Wikander, J. (2006). Experience based Learning and Control of Robotic Grasping. In *Workshop: Towards Cognitive Humanoid Robots, IEEE-RAS Int. Conference on Humanoid Robots*.

Tegin, J., Ekvall, S., Kragic, D., Iliev, B., and Wikander, J. (2009). Demonstration-based Learning and Control for Automatic Grasping. *Intelligent Service Robotics*, **2**, 23–30.

Townsend, W. T. (2000). The BarrettHand Grasper – Programmably Flexible Part Handling and Assembly. *Industrial Robot: An International Journal*, **27**(3), 181–188.

Triebel, R., Schmidt, R., Mozos, O. M., and Burgard, W. (2007). Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data. In *International Joint Conference on Artificial Intelligence*, pages 2225–2230.

Ude, A., Welke, K., Hale, J., and Cheng, G. (2007). Data Acquisition for Building Object Representations: Discerning the Manipulated Objects from the Background. In *Unpublished*.

Wörgötter, F., Agostini, A., Krüger, N., Shylo, N., and Porr, B. (2009). Cognitive Agents: a Procedural Perspective Relying on the Predictability of Object-Action-Complexes. *Robotics and Autonomous Systems*, **57**(4), 420–432.

Zhu, X., Ding, H., and Wang, J. (2003). Grasp Analysis and Synthesis based on a New Quantitative Measure. *IEEE Transactions on Robotics and Automation*, **19**(6), 942–953.

Grasp Affordances from Multi-Fingered Tactile Exploration using Dynamic Potential Fields

Alexander Bierbaum, Matthias Rambow,
Tamim Asfour and Rüdiger Dillmann
Institute for Anthropomatics
University of Karlsruhe (TH)
Karlsruhe, Germany
{bierbaum,rambow,asfour,dillmann}@ira.uka.de

***Abstract*— In this paper, we address the problem of tactile exploration and subsequent extraction of grasp hypotheses for unknown objects with a multi-fingered anthropomorphic robot hand. We present extensions on our tactile exploration strategy for unknown objects based on a dynamic potential field approach resulting in selective exploration in regions of interest. In the subsequent feature extraction, faces found in the object model are considered to generate grasp affordances. Candidate grasps are validated in a four stage filtering pipeline to eliminate impossible grasps. To evaluate our approach, experiments were carried out in a detailed physics simulation using models of the five-finger hand and the test objects.**

I. Introduction

Robotic grasping using multi-fingered hand constitutes a complex task and introduces challenging problems. For well known scenes a grasping or other manipulation process may be pre-programmed when using todays robots. On the other hand, adaptation of a grasping algorithm to formerly unknown or only partially known scenes remains a difficult task, to which different approaches have been investigated.

A classical approach consists in grasp analysis and planning, based on a geometric scene model. In force based grasp planning the forces and moments at selected grasping points are analyzed and matched against a grasp quality criterion considering e.g. force closure. This approach is usually independent of the hand kinematics. In contrast mere geometry based algorithms are tailored to specific gripper designs, especially in the context of multi-fingered hands. Comprehensive overviews on grasp planning are given in [1], [2]. Using grasp planning for previously unknown objects consequently introduces the difficulty of model building from sensor data which is delivered by robot perception. As alternatives to the mere planning approach online control algorithms driven by tactile information have been developed, which make use of *a priori* assumptions on the object to grasp, and control the grasping process by displacing robot fingers. Different control goals have been formulated for grasping convex objects in [3], [4] and later [5], where contact displacements are calculated in order to minimize a grasp quality cost function. The function values are computed using estimation of local surface parameters from haptic feedback, thus resulting in an online control scheme. A further extension capable of dealing with concavities on an object's surface was presented in [6]. Online grasping approaches using a discrete set of hand postures or motions have also been presented [7], [8].

Beside vision based methods tactile exploration may be used for 3D reconstruction of an unknown object, as tactile sensing solves some severe limitations of computer vision, such as sensitivity to illumination and limited perspective. A reconstructed 3D object model may be used for grasp planning and execution as shown e.g. in [9].

Single finger tactile exploration strategies for recognizing polyhedral objects have been presented and evaluated in simulation, see [10] and [11]. In [12] a method for reconstructing shape and motion of an unknown convex object using three sensing fingers is presented. In this approach friction properties must be known in advance and the surface is required to be smooth, i.e. it must have no corners or edges. Further, multiple simultaneous sensor contacts points are required resulting in additional geometric constraints for the setup.

In general, previous approaches in robot tactile exploration for surface reconstruction did not cover the problem of controlling multi-finger robot hands during the exploration process. Also, real world constraints such as manipulator limits or robustness over measurement errors have not been considered. In [13] we have presented first results on the application of a dynamic potential field control technique for guiding a multi-finger robot hand across the surface of an unknown object and simultaneously building a 3D model from contact data.

In this paper we extend our approach in tactile exploration to serve the purpose of extracting grasp affordances for a previously unknown object. Therefore, we have added modifications to our exploration strategy which lead to a homogenous exploration process and prevent sparsely explored regions in the acquired 3D model. We have added a grasp planning system based on a comprehensive geometric reasoning approach as initially reported in [14]. We chose a geometric reasoning approach here as object modelling from tactile exploration currently does not deliver the details required for force analysis and contact modelling, as it is performed in force-based grasp planners, e.g. [15]. As we believe that robustness and applicability of tactile exploration

and robotic grasping algorithms depend significantly upon the deployed hardware configuration, we have evaluated our approach in the framework of a physical simulator, reflecting non-neglectable physical effects such as manipulator kinematics, joint constraints or contact friction. As in related approaches we initially limit our scope to the exploration of static scenes, which means the objects are fixated during exploration and may not move during interaction, although we wish later to develop means of pose estimation and tracking for objects in dynamic scenes.

This paper is organized as follows. In the next section a short introduction to the potential field technique is given and the relevant details of the robot model are described. In Sec. IV-A we present the tactile exploration process and in Sec. IV-B grasp planning and execution. We give details on our simulation scenario and exploration results in Sec. V. Finally, our conclusions and outlook on our future work may be found in Sec. VI.

II. Potential field control

Artificial potential fields have originally been introduced for the purpose of on-line collision avoidance in the context of robot path planning [16]. In the original approach, real-time efficiency was emphasized over obtaining a complete planner. The basic idea is that the robot behaves like a particle influenced in motion by a force field. The field is generated by artificial potentials Φ_i, where obstacles are represented as repulsive potentials $\Phi_r(x) > 0$ and goal regions are represented as attractive potentials $\Phi_a(x) < 0$. The superposition property allows to combine potentials in an additive manner,

$$\Phi(x) = \sum_i \Phi_{r,i}(x) + \sum_j \Phi_{a,j}(x) \quad .$$

The force vector field or potential field F, which influences a *Robot Control Point* (RCP) at position x is defined as

$$F = -\nabla\Phi(x) \quad .$$

A major drawback of potential fields is the existence of local minima outside the goal configurations in which the imaginary force exerted on an RCP is zero. By applying harmonic potential functions it is possible to construct potential fields without spurious local minima for point-like robots. This is not the case with robots that can not be approximated by a point, e.g. a manipulator arm. These are likely to exhibit structural local minima which need to be treated by dedicated escaping strategies [17].

III. Robot hand kinematics, control and sensors

For exploration and grasping we consider a setup comprising a 6-DoF manipulator arm with a five finger robot hand attached to its *Tool Center Point* (TCP). The manipulator arm was modelled according to the Mitsubishi RM-501 five axis small-scale industrial manipulator, which is currently used as a research platform for dexterous haptic exploration in our lab. The model was augmented with a sixth DoF before the TCP to provide a larger configuration space. In

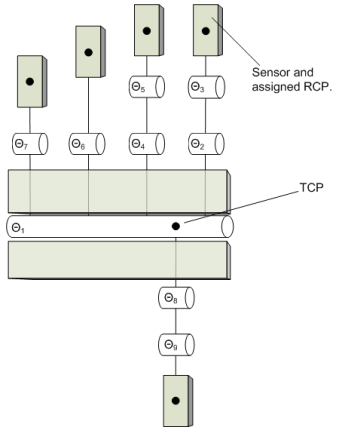


Fig. 1. Kinematics of the robot hand with joint axes, contact sensor locations (grey shaded) with assigned RCPs (black dots) and the TCP.

our exploration control scheme we apply controller outputs to a set of five RCPs, located at the fingertips of the robot hand and to the TCP of the manipulator. The kinematic model of the robot hand is shown in Fig. 1. The hand model provides nine degrees of freedom and is modelled according to the FRH-4 anthropomorphic robot hand presented in [18].

During haptic exploration we are interested in controlling the velocity vectors of the RCP's, which is a different task compared to trajectory control. In trajectory control the end-effector is commanded to follow a desired trajectory with the motion control goal of asymptotic tracking. Yet, the given exploration task does not induce specific trajectories due to the uncertainty in the environment. In our approach we compute the velocity vector applied to an RCP directly from the dynamic potential field, which guides the exploration process. In order to evaluate our concept in a physics simulation environment it was not required to develop a solution to the multipoint end effector inverse kinematic problem. Instead we chose to take advantage of the physical model of the robot system and directly specify velocity vectors to the RCPs by using a virtual actuator which is commonly available in physics simulation frameworks. The joint angles are then determined by solving the constrained rigid body system and a stable and consistent configuration of the robot hand is maintained. In general, this approach is known as *Virtual Model Control* (VMC), which is described in detail in [19]. In our case we specify joint constraints and joint friction for the robot model for achieving an appropriate force distribution over the joint serial paths, while we do not model a compliant behavior. The physics simulation is solved by using the *Inventor Physics Modeling API* (IPSA) which was introduced in [20].

We also make use of the dynamic potential field concept during initialization and grasp execution by placing attractive sources at desired target locations.

For haptic exploration and contact sensing during grasping, tactile sensors are required which we have modelled in our physics simulation. Of course the simulation environment itself may be regarded as omniscient and therefore it is

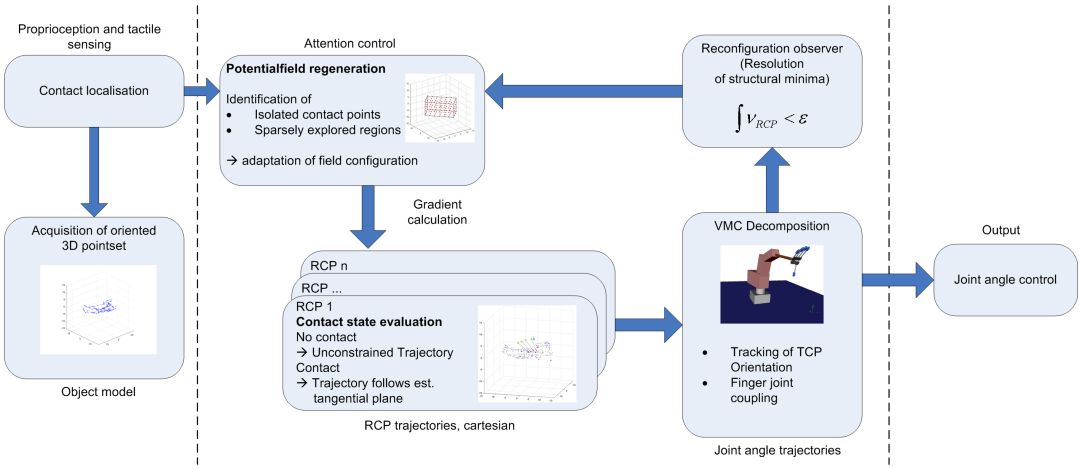


Fig. 2. Overview tactile exploration module.

possible to query all contact locations and force vectors during the interaction of modelled physical bodies. We have restricted contact sensing to dedicated sensor areas which cover the fingertips and the palm of the robot hand, see also Fig. 1. Further, we did not consider the contact force vector but only the contact location on the sensor area to provide a more realistic sensor model. This complies with current tactile sensor technology which in general can not provide both types of information. It is also possible to model more specific sensor characteristics such as a certain resolution in contact location or contact force tresholding, which we did not yet consider in our experiments.

IV. Exploration and Grasping System

The goal of our work is a system enabling a robot with a multi-fingered hand to explore an unknown object using tactile sensing and subsequentially find suitable grasps. Therefore, our system comprises a module for tactile exploration as depicted in Fig. 2. In the following we will describe the exploration and grasp planning process and transition between both modes of operation. Tactile exploration is executed in closed-loop and online in simulation. In contrast, the extraction of grasp affordances is an offline planning process executed subsequently to exploration. Please note that major details of the dextrous tactile exploration process have been reported in [13]. Therefore we will summarize the basic concept and point out the improvements to the original algorithm.

A. Dexterous tactile exploration

As a prerequisite the system requires a rough initial estimate about the objects position, orientation and dimension. In simulation we introduce this information to the system, while this information will be provided by a stereo camera system in the real application. From this information an initial potential field containing only attractive sources is constructed. The trajectories for the RCPs are continuously calculated from the field gradient, while contact point locations and normals are sensed and stored as oriented 3D point set. The normal vectors are estimated by averaging the finger sensor orientations within a spherical neighborhood around a contact point. The RCP trajectories are constrained depending on the contact state of the sensor associated with each RCP, which aims to produce tangential motion during contact.

The potential field is updated from the tactile sensor information as follows. If a contact is detected, a repelling source is inserted at the corresponding location in the potential field. Otherwise, if no contact is found in the circumference of an attractive source, this source becomes deleted from the field. The robot system is likely to reach structural minima during potential field motion. We therefore introduced a reconfiguration observer which detects when the TCP velocity and the mean velocity of all RCPs fall below predefined minimum velocity values. This situation leads to a so called *small reconfiguration* which is performed by temporarily inverting the attractive sources to repulsive sources. This forces the robot into a new configuration from which previously unexplored goal regions may be explored. As this method is not guaranteed to be free of limit cycles we further perform a *large reconfiguration* if subsequent small reconfigurations remain ineffective, i.e. the robot does not escape the structural minimum. During a large configuration the robot is moved to its initial configuration.

Our approach to extract grasp affordances relies on identifying suitable opposite and parallel faces for grasping. Therefore, we needed to improve the tactile exploration process as described above to explore the object surface in a dense scheme and prevent sparsely explored regions. The faces become extracted after applying a triangulation algorithm [21] upon the acquired 3D point set. Triangulation naturally generates large polygons in regions with a low contact point count. We use this property to introduce new attractive sources and guide the exploration process to fill the contact information gaps. Within fixed time step intervals we execute a full triangulation of the point cloud and rank the calculated faces by their size of area. We then add an attractive source at the centers of the ten largest faces. This

leads to preferred exploration of sparsely explored regions, i.e. regions that need further exploration, and conseqently to a more reliable estimate for the objects surface.
We apply a similar scheme to isolated contact points, i.e. contacts that have no further contact points in their immediate neighborhood. We surround these by eight cubically arranged attractive charges. This leads to the effect that once an isolated contact is added, the according RCP now explores its neighborhood instead of being repelled to a more distant unexplored region.

B. Grasping Phase

As an exemplary application for our exploration procedure we have implemented a method for identifying grasp affordances from the oriented point set.
We did not choose a traditional force-based grasp planning algorithm as this would require to calculate a triangulated geometric object model from the 3D point set. The point set delivered by tactile exploration is inherently sparse and irregular and we found that most triangulation algorithms would fail to produce results in a usable way. Instead we found that extraction of local features from the point set is more robust than triangulation. We therefore chose a subset of a geometric reasoning approach as proposed in [14] in order to compute grasp affordances based on the acquired object information.

1) Extraction of grasping features: A grasp affordance contains a pair of object features from which the grasping points are determined in subsequent steps. In general, planar faces, edges and vertices of a polygonal object representation may be used as object features. We only consider planar faces in our implementation, as estimation and extraction of planar faces from the given 3D point set is much more reliable than that of edges or vertices. Therefore, we investigate the oriented 3D point set for neighboured contact points with similar normal vectors. Using a region growing method the contact points in adequate dense regions are assigned to faces. The original method is designed for parallel robot grippers therefore the grasp affordances found are consequently of a parallel type with opposing planar faces for grasping. We apply a mapping scheme as described below in Sec. IV-B.3 to compute the five finger tip target locations for the robot hand within each face.

2) Geometric feature filters: Initially every possible face pairing is considered as a potential grasp affordance. In a sequential geometric filtering process all grasps unlikely to be executed successfully with the given robot hand are eliminated from the set of all pairings. The filter parameters are chosen for the FRH-4 hand. We use a four stage filtering pipeline in our approach. The results of the filter stages are summed up to a score for each grasp affordance. Each filter is designed to return a value of 0 when disqualifying a pairing and value $1 \leq o \leq 1.1$ for accepting a pairing. As only grasp affordances with filter score ≥ 4 are considered valid this automatically implies that valid grasps have to pass all filter stages successfully.

- *Parallelism:* This filter tests the two faces for parallelism. Let $\vec{n}_1$ and $\vec{n}_2$ be the normal vectors of the two faces f_1 and f_2, ϕ the angle between $\vec{n}_1$ and $\vec{n}_2$ and ϕ_{max} the maximum angle for acceptance. The output o of the filter is:
$$o = \begin{cases} 0, & \text{if } \phi > \phi_{max} \\ 1 + \frac{(\phi_{max} - \phi)}{\phi_{max}} \cdot 0.1, & \text{otherwise.} \end{cases}$$
- *Minimum Face Size:* This filter tests the two faces for adequate size of area. Let a_1 and a_2 be the areas of the faces f_1 and f_2. The minimum area for acceptance is a_{min} , k_a is a normalization factor. Then the output o of this filter is:
$$o = \begin{cases} 0, \text{if } (a_1 < a_{min}) \vee (a_2 < a_{min}) \\ 1 + \min(\min(\frac{a_1}{k_a}, \frac{a_2}{k_a}), 0.1), & \text{otherwise.} \end{cases}$$
- *Mutual Visibility:* With this filter the two faces are projected into the grasping plane gp, which is the plane with the mean normal vector $\vec{n}_{gp}$ situated in the middle of the two faces f_1 and f_2. So let $f_{1\downarrow gp}$ and $f_{2\downarrow gp}$ be the projections of f_1 and f_2 onto gp. Then, a_{int} is the intersection area of $f_{1\downarrow gp}$ and $f_{2\downarrow gp}$. The minimum intersection area for acceptance is a_{min} , k_{mv} is a normalization factor. The filter's output is:
$$o = \begin{cases} 0, & \text{if } a_{int} < a_{min} \\ 1 + \min(\frac{a_{int}}{k_{mv}}, 0.1), & \text{otherwise.} \end{cases}$$
- *Face Distance:* The last filter incorporates the characteristics of the used manipulator tool, i.e. the robot hand. The filter checks if the robot hands spreading capability matches the distance of the faces. Let d be the distance between the centers of the faces f_1 and f_2, d_{min} and d_{max} are the minimum respectively maximum admitted distance values. Then the filters output is
$$o = \begin{cases} 0, & \text{For } d \notin [d_{min}, d_{max}] \\ 1, & \text{otherwise.} \end{cases}$$

3) Grasp execution: The grasp affordance with the highest score is used as the candidate for grasp execution. In a first step we compute the grasping position $\vec{p}_{tcp,a}$ of the TCP and the grasping approach direction as depicted in Fig. 3.

Initially we estimate the centers $\vec{c_1}$, $\vec{c_2}$ of the two faces f_1, f_2 as the centers of gravity of all contact points assigned to each face. From this we determine the center point $\vec{gp} = \frac{\vec{c_1} + \vec{c_2}}{2}$ on the line connecting the centers of the two faces. Then we analyse the first principle component $\vec{pc}$ of the acquired 3D point cloud and calculate the grasping position as

$$\vec{p}_{tcp,a} = \vec{gp} + (\vec{n}_{gp} \times \vec{pc}) \cdot d,$$

where d is a distance which considers the fingers length of the robot hand. The cross product $(\vec{n}_{gp} \times \vec{pc})$ becomes the approach direction. We only consider grasping the object from top. Therefore, in the case the coordinate $\vec{p}_{tcp,a}$ is below the object to grasp, we mirror its location across the center

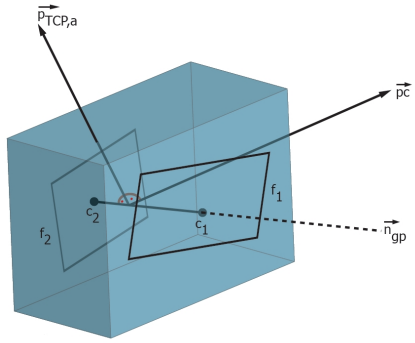


Fig. 3. Calculation of the grasp center point and approach direction.

between c_1 and c_2 and along the approach direction to a location above the object. Clearly, we make an assumption about the object's extension here. From the face pair of the grasp affordance finger tip target locations need to be computed. This is achieved by the following mapping scheme.

The target for the thumb $\vec{p}_{thumb,a}$ is set to be the center of the smaller of the two faces. We choose target locations $\vec{p}_{index,a}, \vec{p}_{middle,a}, \vec{p}_{ring,a}$ and $\vec{p}_{pinkie,a}$ for the opponent fingers around the center and in the plane of the larger of the two faces. The arrangement is chosen, so that it is perpendicular to the approach direction in the plane of target face. If the target location of ring finger or pinkie is not situated within the face area the fingers will not be used for grasping. This way the number fingers involved during grasping is automatically adapted.

Motion execution starts with the hand in an initial pose, as it is always reached after a large reconfiguration. From here we apply the potential field control to the RCPs and the TCP. Unlike during the exploration phase, the TCP and the RCPs share the set of repulsive potential sources while having individual attractive potential sources as mentioned above. Repulsive sources located in the target planes become deleted.

As long as the TCP is distant from its target $\vec{p}_{tcp,a}$ the potential field velocity control is only applied to the TCP while the finger joints remain open via direct joint control. When the TCP is close to its target we additionally apply the potential field control to the RCPs. If an RCP is not in use because the finger is not involved in grasping, the associated finger joints are still kept open. Further, the palm normal $\vec{n}$ is aligned towards $\vec{gp}$ by controlling forces acting on the hand's pitch and roll DoFs.

If the RCPs in use have approached the finger target locations, the fingers are closed and the corresponding sensors are checked for contact. Once all assigned RCP sensors are in contact with the object, potential field control is turned off and the finger joints are closed directly. The virtual fixture of the object then becomes disabled in the simulation and the robot arm moves back to its initial position with the object grasped and lifted.

V. Simulation Results

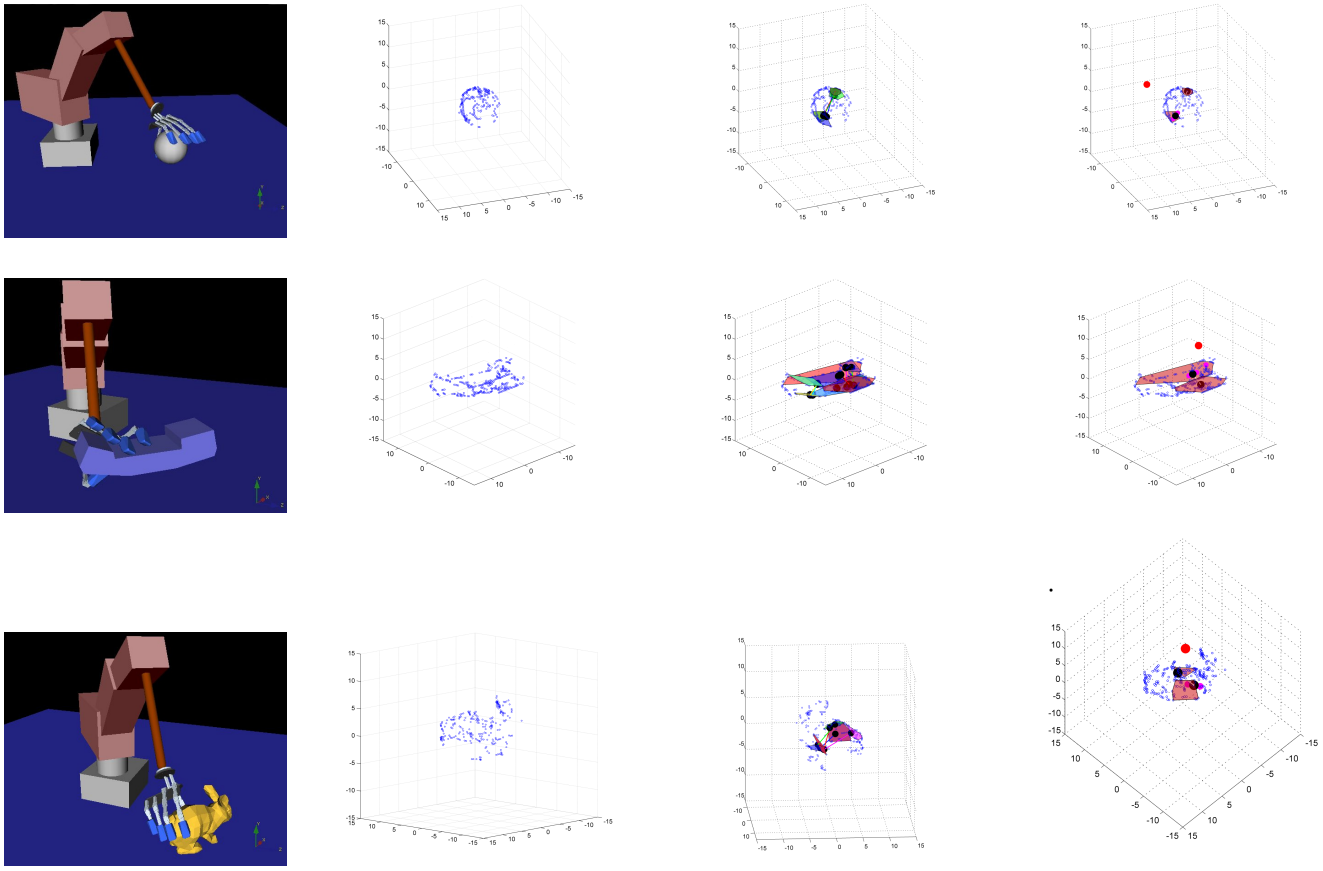
We evaluated our exploration and grasping system in several virtual scenes using our physics simulator with standard earth gravity $g_N = 9.81$ applied. For contacts Coulomb friction with a friction coeefficient $\mu = 0.5$ is considered. The virtual scenes were set up with different rigid objects of suitable size for grasping by the hand: a sphere, a telephone receiver and a rabbit. The objects are placed approximately in the center of the robots workspace. All objects are fixated floating above the simulators virtual ground to avoid interference, as we currently do not differ between contact between the object of interest and any other obstacle in the workspace. As described in Sec. IV-A the cubical bounding box of the object is computed from position and space occupancy estimates and used to initialize the exploration potential field. Grasp affordances are extracted after a fixed number of 2000 control time steps, whereby each control time step comprises ten simulation time steps with a temporal resolution of $T = 0.04s$.

Fig. 4 shows typical results. Here figures in column (c) show the 6 best candidate faces for grasping. The color indicates score ranking in following order: red, green, blue, magenta, cyan, yellow. Black dots indicate the center of a face, which is calculated as mean value of all points in the face. Colored lines connect corresponding centers of corresponding faces. In colum (d) the grasp affordance with the highest score is shown. Purple dots indicate grasping points for index, middle, ring and pinkie finger. Ring and pinkie grasping points are only plotted if they are used. The red dot marks the location of the attractive potential source for the TCP at start of the approaching phase. Naturally, the algorithm performs worse with objects exposing curved regions as the algorithm searches for planar faces. Therefore, only one grasp affordance was found for the sphere in the given exploration interval. The exploration of the rabbit shows similar results. Still, successful grasps can be performed with the grasp affordances identified.

In contrast, several affordances could be identified with the model of the telephone receiver consisting of large polygons. In general, the number of found grasp affordances increases with exploration time. The video accompanying this paper shows examples of tactile exploration and grasp execution for the rabbit.

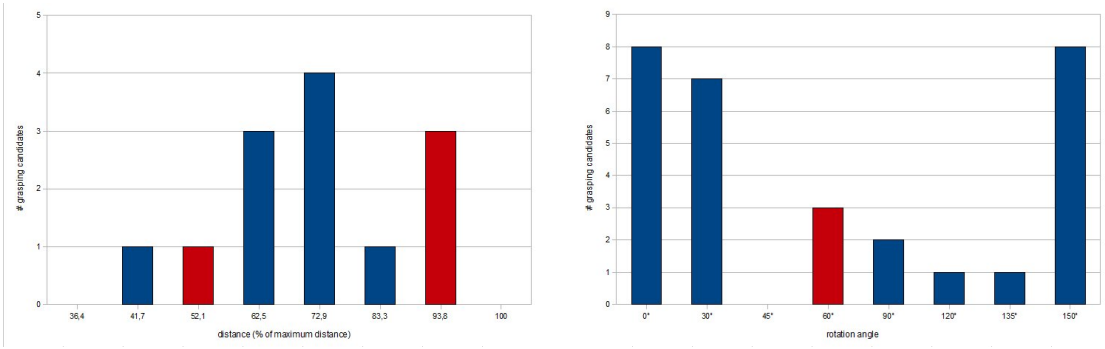
Beside experiments with different objects we also investigated performance of the system with objects placed at different positions and orientations in the workspace. For the experiments a grasp is considered successful if the manipulator can grasp and lift the object in simulation. We believe this is still a good approximation for reality as the simulator only calculates with rigid body dynamics and assumes point contacts. In reality such a robot system would be equipped with deformable rubber finger tips which will provide a significant larger contact area leading to higher tangential forces. Therefore we assume that a real robot system could execute the simulated successful grasps.

In a first experiment we placed the sphere, which is naturally



(a) (b) (c) (d)

Fig. 4. Typical simulation results from top to bottom: Sphere, telephone receiver, rabbit. Column (a) shows a virtual scene snapshot during exploration,(b) final point cloud, (c) grasp affordances, (d) best grasp and grasping points.



(a) Sphere at different distances

(b) Telephone receiver at different orientations

Fig. 5. Number of identified grasp affordances. Blue: successful grasp execution, red: failed grasp execution with best candidate.

invariant to rotations, at different distances ranging from minimum to maximum reaching distance for the manipulator arm in the workspace. Fig. 5(a) shows the number of found grasp affordances after $N = 2000$ exploration steps. After generation the grasp affordance with the highest score is executed as desribed in Sec. IV-B.3. In the figure a red bar indicates a failed grasp execution, a blue bar indicates a successful grasp execution, both with the best candidate grasp applied. The failed grasps may be deduced to the error between the estimated grasping plane and the local tangential plane of the sphere in combination with an inappropriate situation of the sphere within the robots workspace. This could be improved by increasing the exploration time in order to collect more contact data points.

In a second experiment we investigated the scheme with the robot model for sensitivity towards different orientations of an elongated object as the telephone receiver. Therefore, the receiver is placed in the scene with different orientations

around the Y-axis (direction of gravity). The initial configuration can be seen in the mid image of Fig. 4(a). The receiver was situated in the workspace center area of the manipulator arm. The results are depicted in Fig. 5(b) and indicate that the receiver provides less features to extract grasp affordances from with its longer axis pointing toward the manipulator. The reasons for the failed grasp agree with those from experiment 1. Note that the receiver is not a symmetric object, therefore the number of grasping candidates is also not symmetric over rotation.

VI. Conclusions

In this paper we have presented a control scheme for tactile exploration and subsequent extraction and execution of grasp affordances for previously unknown objects using an anthropomorphic multi-fingered robot hand. Our approach is based on dynamic potential fields for motion guidance of the fingers. We have shown that grasp affordances may be generated from geometric features extracted from the contact point set resulting from tactile exploration. The complete control scheme was evaluated in a detailed physics simulation of the robot system with test objects of different shape and presented the results of the grasp planner based on the exploration data. Finally, we tested the best grasp candidate by executing the grasp within the physics simulation. In further experiments we have reported results for different object locations and orientations in the manipulator workspace.

For the future we are working on an extension of the presented set of geometric filters in order to further improve the success rate upon grasp execution with our robot hand. Further we will consider the incorporation of the palm during grasp execution, which would enable power grasps.

Concluding, we are confident that the dynamic potential field based approach presented may be used for real world tactile exploration and grasping with an anthropomorphic robot hand, as it appears robust enough to autonomously control interaction of the robot hand with a previously unknown object using tactile information. We assume that the proposed scheme is transferable to different manipulator and robot hand kinematics by adapting filter parameters, number of RCPs and RCP locations. We further plan to investigate possibilities of combination with exploration methods based on sensors of different modalities than haptics, e.g. vision based object exploration. The developed control scheme based on VMC and dynamic potential fields is currently subject to implementation on our real world robot system equipped with five-finger hands [22].

Acknowledgement

The work described in this paper was conducted within the EU Cognitive Systems projects PACO-PLUS (FP6-027657) and GRASP (FP7-215821) funded by the European Commission.

References

[1] J. Pertin-Troccaz, "Grasping: A state of the art," in *The Robotics Review*, O. Khatib, J. J. Craig, and T. Lozano-Perez, Eds. The MIT Press, 1989, vol. 1.

[2] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 24-28 April 2000, pp. 348–353vol.1.

[3] M. Teichmann and B. Mishra, "Reactive algorithms for 2 and 3 finger grasping," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems, Grenoble, France*, 1994.

[4] J. Coelho and R. Grupen, "A control basis for learning multifingered grasps," *Journal of Robotic Systems*, vol. 14, no. 7, pp. 545–557, 1997.

[5] J. Platt, R., A. Fagg, and R. Grupen, "Nullspace composition of control laws for grasping," in *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, vol. 2, 30 Sept.-5 Oct. 2002, pp. 1717–1723vol.2.

[6] D. Wang, B. T. Watson, and A. H. Fagg, "A switching control approach to haptic exploration for quality grasps," in *Robotic Science and Systems*, 2007.

[7] R. Platt, "Learning grasp strategies composed of contact relative motions," in *IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, PA*, Dec 2007.

[8] J. Steffen, R. Haschke, and H. Ritter, "Experience-based and tactile-driven dynamic grasp control," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Oct 2007, pp. 2938–2943.

[9] B. Wang, L. Jiang, J. LI, and H. Cai, "Grasping unknown objects based on 3d model reconstruction," in *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005, pp. 461–466.

[10] K. Roberts, "Robot active touch exploration: constraints and strategies," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 13-18 May 1990, pp. 980–985 vol.2.

[11] S. Caselli, C. Magnanini, F. Zanichelli, and E. Caraffi, "Efficient exploration and recognition of convex objects based on haptic perception," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 22-28 April 1996, pp. 3508 – 3513 vol.4.

[12] M. Moll and M. A. Erdmann, *Reconstructing the Shape and Motion of Unknown Objects with Active Tactile Sensors*, ser. Springer Tracts in Advanced Robotics. Springer Verlag Berlin/Heidelberg, 2003, ch. 17, pp. 293–310.

[13] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann, "A potential field approach to dexterous tactile exploration," in *International Conference on Humanoid Robots 2008, Daejeon, Korea*, 2008.

[14] J. Pertin-Troccaz, "Geometric reasoning for grasping: a computational point of view," in *CAD Based Programming for Sensory Robots*, ser. NATO ASI Series, B. Ravani, Ed. Springer Verlag, 1988, vol. 50, iSBN 3-540-50415-X.

[15] A. T. Miller and P. K. Allen, "Graspit!: A versatile simulator for grasp analysis," in *Proceedings ASME International Mechanical Engineering Congress & Exposition, Orlando*, Nov. 2000, pp. 1251–1258.

[16] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[17] J.-O. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," in *Proc. IEEE International Conference on Robotics and Automation*, 1991, pp. 790–796 vol.1.

[18] I. Gaiser, S. Schulz, A. Kargov, H. Klosek, A. Bierbaum, C. Pylatiuk, R. Oberle, T. Werner, T. Asfour, G. Bretthauer, and R. Dillmann, "A new anthropomorphic robotic hand," in *IEEE-RAS International Conference on Humanoid Robots*, 2008.

[19] J. Pratt, A. Torres, P. Dilworth, and G. Pratt, "Virtual actuator control," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, vol. 3, 1996, pp. 1219–1226 vol.3.

[20] A. Bierbaum, T. Asfour, and R. Dillmann, "Ipsa - inventor physics modeling api for dynamics simulation in manipulation," in *IROS - Workshop on Robot Simulation, 22. Sept. 2008, Nice, France*, 2008.

[21] N. Amenta, S. Choi, and R. Kolluri, "The power crust," in *Sixth ACM Symposium on Solid Modeling and Applications*, 2001, pp. 249–260.

[22] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-III: An integrated humanoid platform for sensory-motor control," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, Dec. 2006, pp. 169–175.

Dynamic Potential Fields for Dexterous Tactile Exploration

Alexander Bierbaum, Tamim Asfour and Rüdiger Dillmann

Abstract Haptic exploration of unknown objects is of great importance for acquiring multimodal object representations, which enable a humanoid robot to autonomously execute grasping and manipulation tasks. In this paper we present our ongoing work on tactile object exploration with an anthropomorphic five-finger robot hand. In particular we present a method for guiding the hand along the surface of an unknown object to acquire a 3D object representation from tactile contact data. The proposed method is based on the dynamic potential fields which have originally been suggested in the context of mobile robot navigation. In addition we give first results on how to extract grasp affordances of unknown objects and how to perform object recognition based on the acquired 3D point sets.

1 Introduction

Humans make use of different types of haptic exploratory procedures for perceiving physical object properties such as weight, size, rigidity, texture and shape [12]. For executing subsequent tasks on previously unknown objects such as grasping and also for non-ambiguous object identification the shape property is of outmost importance. In robotics this information is usually obtained by means of computer vision where known weaknesses such as changing lightning conditions and reflections seriously limit the scope of application. For robots and especially for humanoid robots, tactile perception is supplemental to the shape information given by visual perception and may directly be exploited to augment and stabilize a spatial representation of real world objects. In the following we will give a short overview on the state of the art in the field of robot tactile exploration and related approaches.

A. Bierbaum, T. Asfour, R. Dillmann
University of Karlsruhe (TH), Institute for Anthropomatics, Humanoids and Intelligence Systems Laboratories.
e-mail: {bierbaum,asfour,dillmann}@ira.uka.de

Different strategies for creating polyhedral object models from single finger tactile exploration have been presented with simulation results in [19] and [5]. Experimental shape recovery results from a surface tracking strategy for a single robot finger have been presented in [6]. A different approach concentrates on the detection of local surface features [15] from tactile sensing. In [13] a method for reconstructing shape and motion of an unknown convex object using three sensing fingers is presented. In this approach friction properties must be known in advance and the surface is required to be smooth, i.e. must have no corners or edges. Further, multiple simultaneous sensor contacts points are required resulting in additional geometric constraints for the setup.
In the works mentioned above the exploration process is based on dynamic interaction between the finger and object, in which a sensing finger tracks the contour of a surface. Other approaches are based on a static exploration scheme in which the object gets enclosed by the fingers and the shape is estimated from the robot finger configuration. In [14], [9] and [20] the finger joint angle values acquired during enclosure are fed to an appropriately trained SOM-type neural network which classifies the objects according to their shape. Although this approach gives good results in terms of shape classification, it is naturally limited in resolution and therefore does not provide sufficient information for general object identification as with dynamic tactile exploration.
In this work we will present the current state and components of our system for acquiring a 3D shape model of an unknown object using multi-fingered tactile exploration based on dynamic potential fields. In addition we give first results on how to extract grasp affordances of unknown objects and how to perform object recognition based on the acquired 3D point sets.

2 Dynamic potential fields for exploration

We have transferred the idea of potential field based exploration to tactile exploration for surface recovery using an anthropomorphic robot hand. Potential field techniques have a long history in robot motion planning [11]. Here, the manipulator follows the streamlines of a field where the target position is modelled by an attractive potential and obstacles are modelled as repulsive potentials. By assigning regions of interest to attractive sources and already known space to repulsive sources this scheme may also be exploited for spatial exploration purposes with mobile robots [18]. The notion of dynamic potential fields evolves as the regions of interest and therefore the field configuration changes over time due to the progress in exploration. Yet, this method has not been reported for application in multifingered tactile exploration. For this purpose we have defined a set of *Robot Control Points* (RCPs) at the robot hand to which we apply velocity vectors calculated from the local field gradient

$$\mathbf{v} = -k_v \nabla \Phi(x) \quad .$$

The potential $\Phi(x)$ is calculated from superposition of all sources. We use harmonic potential functions to avoid the formation of local minima in which the imaginary force exerted on an RCP is zero. Further, we deploy a dedicated escape strategy to resolve structural minima, which naturally evoke from the multiple end-effector problem given by the fingers of the hand. The velocity vectors applied to the RCPs are computed in the cartesian coordinate frame therefore an inverse kinematic scheme is required to calculate joint angles for the robot hand and arm. In our case we have chosen Virtual Model Control (VMC) [17] to solve for the joint angles, as it links the potential field approach to inverse kinematics in an intuitive way.
Initially we have evaluated our approach in a detailed physical simulation using the model of our humanoid robot hand [8]. During exploration the contact location and estimated contact normals are acquired from the robot hands tactile sensor system and stored as a oriented 3D point set. We have modelled tactile sensors in the simulation environment which determine contact location. The contact normals are estimated from the sensor orientation to reflect the fact that current sensor technology can not measure contact normals reliably. The object representation may be used for further applications such as grasping and object recognition as we will describe in the following sections.

3 Tactile Exploration

Fig. 1 gives an overview on our tactile exploration module. An initial version of this method has been presented in [3]. As prerequisite the system requires a rough initial estimate about the objects position, orientation and dimension. In simulation we introduce the information to the system, while this information will be provided by a stereo camera system in the real application. From this information an initial potential field containing only attractive sources is constructed in a uniform grid which covers the exploration space in which the object is situated.

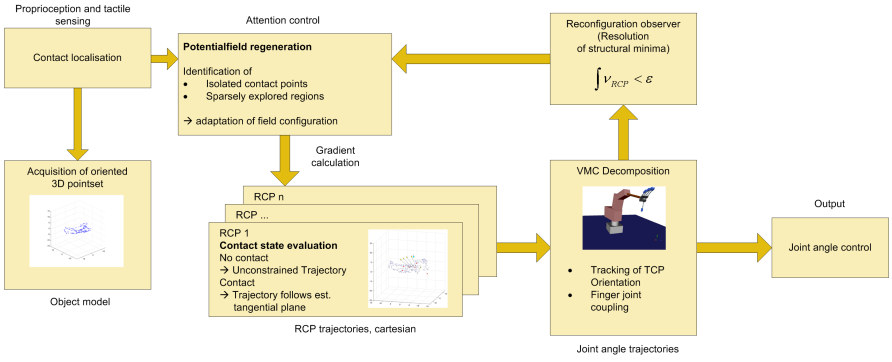


Fig. 1 Tactile exploration scheme based on dynamic potential field.

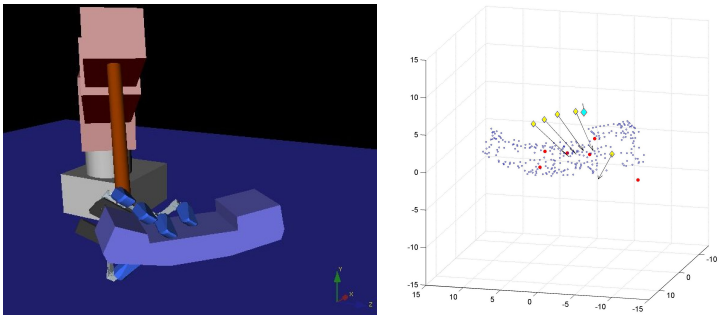


Fig. 2 Tactile exploration of a phone receiver (left) and acquired 3D point set (right).

During exploration it is required to fixate the object as contact points are acquired in world reference frame. The trajectories for the RCPs become continuously calculated from the field gradient, while contact point locations and normals are sensed and stored as oriented 3D point set. The normal vectors are estimated from finger sensor orientations. The RCP trajectories are constrained depending on the contact state of the sensor associated with each RCP, which aims to produce tangential motion during contact.

The potential field is updated from the tactile sensor information as follows. If no contact is found in the circumference of an attractive source it becomes deleted. If a contact is detected a repelling source is inserted at the corresponding location in the grid.

The robot system is likely to reach structural minima during potential field motion. We therefore introduced a reconfiguration observer which detects when the TCP velocity and the mean velocity of all RCPs fall below predefined minimum velocity values. This situation leads to a so called *small reconfiguration* which is performed by temporarily inverting the attractive sources to repulsive sources and thus forcing the robot into a new configuration which allows to explore previously unexplored goal regions. As this method is not guaranteed to be free of limit cycles we further perform a *large reconfiguration* if subsequent small reconfigurations remain ineffective, i.e. the robot does not escape the structural minimum. During a large configuration the robot is moved to its initial configuration.

Our approach to extract grasp affordances relies on identifying suitable opposing and parallel faces for grasping. Therefore, we needed to improve the original tactile exploration process to explore the object surface with preferably homogenous density and prevent sparsely explored regions. The faces become extracted after applying a triangulation algorithm upon the acquired 3D point set. Triangulation naturally generates large polygons in regions with low contact point count. We use this property in our improved exploration scheme to introduce new attractive sources and guide the exploration process to fill contact information gaps. Within fixed time step intervals we execute a full triangulation of the point cloud and rank the calculated faces by their size of area. In our modification we add an attractive source each at the centers of the ten largest faces. This leads to preferred exploration of sparsely

explored regions, i.e. regions that need further exploration, and consequently lead to a more reliable estimate for the objects surface. As further improvement we apply a similar scheme to isolated contact points, i.e. contacts which have no further contact points in their immediate neighborhood, by surrounding these points with eight cubically arranged attractive charges. This leads to the effect that once an isolated contact is added, the according RCP now explores its neighborhood instead of being repelled to a more distant unexplored region.

4 Extraction of Grasp Affordances

As an exemplary application for our exploration procedure we have implemented a subset of the automatic robot grasp planner proposed in [16] in order to compute possible grasps based on the acquired oriented 3D point set, we call *grasp affordances*. A grasp affordance contains a pair of object features which refer to grasping points of a promising grasp candidate using a parallel grasp. We prefered to investigate this geometrical planning approach in contrast to grasp planning algorithms using force closure criteria, e.g. [7], due to its robustness when planning with incomplete geometric object models as they arise from the described exploration scheme. In our case we only consider planar face pairings from the given 3D point set as features for grasping, which we extract from the contact normal vector information using a region growing algorithm. Initially every possible face pairing is

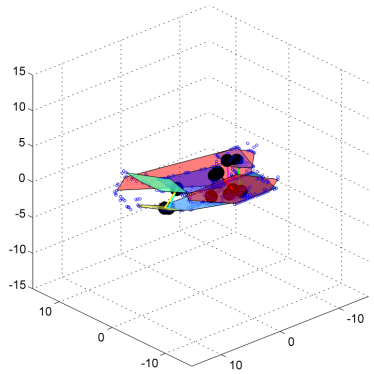


Fig. 3 Extracted grasp affordances for the telephone receiver.

considered as a potential symbolic grasp. All candidates are submitted to a geometric filter pipeline which eliminates impossible grasps from this set. The individual filter j returns a value of $f_o, j = 0$ when disqualifying and a value $f_o, j > 0$ for accepting a pairing. For accepted pairings the individual filter outputs are summed to

a score for each symbolic grasp, where the filter pairing with the highest score is the most promising candidate for execution.
The filter pipeline comprises the following stages in order of their application.

- *Parallelism:* This filter tests the two faces for parallelism and exports a measure indicating the angle between the two faces.
- *Minimum Face Size:* This filter compares the two faces to minimum and maximum thresholds. Selection of these values depends on the dimensions of the robot hand and fingers.
- *Mutual Visibility:* This filter determines the size of overlapping area when the two faces are projected into the so called grasping plane, which resides in parallel in the middle between the faces.
- *Face Distance:* This filter tests the distance of the two faces which must match the spreading capability of the robot hand. Therefore, this filter is also parameterized by the dimensions of the robot hand.

Fig. 3 shows symbolic grasps found for the receiver from Fig. 2. Face pairings are indicated by faces of the same color, the black spots mark the centers of the overlapping region of opposing faces in respect to the grasping plane. These points will later become the finger tip target locations during grasp execution.

5 Future concepts for object recognition

The oriented 3D point set acquired from tactile exploration is inherently sparse and of irregular density which makes shape matching a difficult task. In a first approach we have investigated a superquadric fitting technique which allows to estimate a super quadric function from tactile contacts in a robust manner [2]. Fig. 4 (left) shows a superquadric recovered from tactile exploration data using a hybrid approach where a genetic algorithm is used to identify the global minimum region

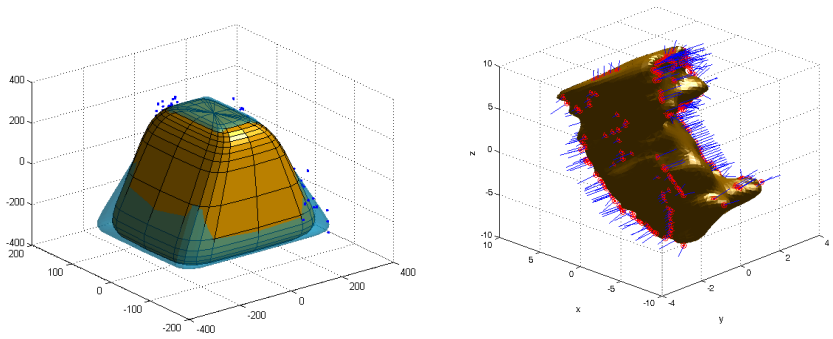


Fig. 4 Superquadric reconstructed from a tactile point set (left). A surface reconstructed using 3D Fourier transform (right).

and a least-squares-method converges to an optimum solution. Yet, this method is limited to representing and recognizing shapes only from a set of major geometric primitives such as spheres, cylinders, boxes or pyramids. For representing more complex shapes, different shape descriptors which may also become applied to partial models have been investigated in the research fields of computer vision and 3D similarity search [4]. The methods reported are mainly designed for large 3d data sets with uniform sampling density. Therefore, we have focused on investigating suitable point set processing methods which may interpolate the tactile contact data in order to compute robust shape descriptors. Fig. 4 (right) shows an oriented point set from tactile exploration which has been interpolated by using an algorithm for reconstruction of solid models [10]. From uniform density point sets stable shape descriptors may be computed using methods developed in the context of computer vision. Promising candidates for distinct shape descriptors here are geometric hash tables and spectra from spherical harmonic transforms. Both provide means for translational and rotational invariance, which is essential in object recognition from exploration data in human environments.

6 Discussion

In this paper we presented an overview on our system for tactile exploration. Our approach is based on dynamic potential fields for motion guidance of the fingers of a humanoid hand along the contours of an unknown object. We added a potential field based reconfiguration strategy to eliminate structural minima which may arise from limitations in configuration space. During the exploration process oriented point sets from tactile contact information are acquired in terms of a 3D object model. Further, we presented concepts and preliminary results for applying the geometric object model to extract grasp affordances from the data. The grasp affordances comprise grasping points of promising configurations which may be executed by a robot using parallel-grasps. For object recognition we have outlined our approach which relies on transforming the sparse and non-uniform pointset from tactile exploration to a model representation appropriate for 3D shape recognition methods known from computer vision.

We believe that the underlying 3D object representation of our concept is a major advantage as it provides a common basis for multimodal sensor fusion with a stereo vision system and other 3D sensors. As finger motion control during exploration is directly influenced from the current model state via the potential field, this approach becomes a promising starting point for developing visuo-haptic exploration strategies.

Currently we extend our work in several ways. In a next step we will transfer the developed tactile exploration scheme to our robot system Armar-III [1] which is equipped with five-finger hands and evaluate the concept in a real world scenario. Further, we are developing and implementing a motion controller which is capable to execute and verify the grasp affordances extracted from exploration. For object

recognition we will continue to investigate suitable shape descriptors and evaluate them with simulated and real world data from tactile exploration.

References

1. T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. Armar-iii: An integrated humanoid platform for sensory-motor control. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 169–175, Dec. 2006.
2. A. Bierbaum, I. Gubarev, and R. Dillmann. Robust shape recovery for sparse contact location and normal data from haptic exploration. In *IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, Nice, France*, pages 3200 – 3205, 2008.
3. A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. A potential field approach to dexterous tactile exploration. In *International Conference on Humanoid Robots 2008, Daejeon, Korea*, 2008.
4. Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. Feature-based similarity search in 3d object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.
5. S. Caselli, C. Magnanini, F. Zanichelli, and E. Caraffi. Efficient exploration and recognition of convex objects based on haptic perception. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, pages 3508 – 3513 vol.4, 22-28 April 1996.
6. N. Chen, R. Rink, and H. Zhang. Local object shape from tactile sensing. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3496–3501vol.4, 22-28 April 1996.
7. C. Ferrari and J. Canny. Planning optimal grasps. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2290–2295vol.3, 12-14 May 1992.
8. I. Gaiser, S. Schulz, A. Kargov, H. Klosek, A. Bierbaum, C. Pylatiuk, R. Oberle, T. Werner, T. Asfour, G. Bretthauer, and R. Dillmann. A new anthropomorphic robotic hand. In *IEEE-RAS International Conference on Humanoid Robots*, 2008.
9. M. Johnsson and C. Balkenius. Experiments with proprioception in a self-organizing system for haptic perception. In M. S. Wilson, F. Labrosse, U. Nehmzow, C. Melhuish, and M. Witkowski, editors, *Towards Autonomous Robotic Systems*, pages 239–245, Aberystwyth, UK, 2007. University of Wales.
10. Michael Kazhdan. Reconstruction of solid models from oriented point sets. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 73, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
11. Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.
12. Susan J. Lederman and Roberta L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19(3):342–368, 1987.
13. Mark Moll and Michael A. Erdmann. *Reconstructing the Shape and Motion of Unknown Objects with Active Tactile Sensors*, chapter 17, pages 293–310. Springer Tracts in Advanced Robotics. Springer Verlag Berlin/Heidelberg, 2003.
14. L. Natale, G. Metta, and G. Sandini. Learning haptic representation of objects. In *International Conference on Intelligent Manipulation and Grasping, Genoa, Italy.*, July 2004.
15. A.M. Okamura and M.R. Cutkosky. Haptic exploration of fine surface features. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, pages 2930–2936, 10-15 May 1999.
16. Jocelyne Pertin-Troccaz. Geometric reasoning for grasping: a computational point of view. In Bahram Ravani, editor, *CAD Based Programming for Sensory Robots*, volume 50 of *NATO ASI Series*. Springer Verlag, 1988. ISBN 3-540-50415-X.

17. J. Pratt, A. Torres, P. Dilworth, and G. Pratt. Virtual actuator control. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, volume 3, pages 1219–1226 vol.3, 1996.
18. Edson Prestes e Silva, Paulo M. Engel, Marcelo Trevisan, and Marco A. P. Idiart. Exploration method using harmonic functions. *Robotics and Autonomous Systems*, 40(1):25–42, July 2002.
19. K.S. Roberts. Robot active touch exploration: constraints and strategies. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 980–985 vol.2, 13-18 May 1990.
20. Shinya Takamuku, Atsushi Fukuda, and Koh Hosoda. Repetitive grasping with anthropomorphic skin-covered hand enables robust haptic recognition. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 3212–3217, 2008.