Learning grasps with topographic features

Article *in* The International Journal of Robotics Research · May 2015
DOI: 10.1177/0278364915577105

CITATIONS
14

READS
680

3 authors, including:

Astrid Weiss
TU Wien
108 PUBLICATIONS **904** CITATIONS
SEE PROFILE

Markus Vincze
TU Wien
369 PUBLICATIONS **3,279** CITATIONS
SEE PROFILE

Article

The International Journal of
Robotics Research
2015, Vol. 34(9) 1167–1194

DOI: 10.1177/0278364915577105
ijr.sagepub.com

Learning grasps with topographic features

David Fischinger, Astrid Weiss and Markus Vincze

Abstract

We present a system for grasping unknown objects, even from piles or cluttered scenes, given a point cloud. Our method is based on the topography of a given scene and abstracts grasp-relevant structures to enable machine learning techniques for grasping tasks. We describe how Height Accumulated Features (HAF) and their extension, Symmetry Height Accumulated Features, extract grasp relevant local shapes. We investigate grasp quality using an F-score metric. We demonstrate the gain and the expressive power of HAF by comparing its trained classifier with one that resulted from training on simple height grids. An efficient way to calculate HAF is presented. We describe how the trained grasp classifier is used to explore the whole grasp space and introduce a heuristic to find the most robust grasp. We show how to use our approach to adapt the gripper opening width before grasping. In robotic experiments we demonstrate different aspects of our system on three robot platforms: a Schunk seven-degree-of-freedom arm, a PR2 and a Kuka LWR arm. We perform tasks to grasp single objects, autonomously unload a box and clear the table. Thereby we show that our approach is easily adaptable and robust with respect to different manipulators. As part of the experiments we compare our algorithm with a state-of-the-art method and show significant improvements. Concrete examples are used to illustrate the benefit of our approach compared with established grasp approaches. Finally, we show advantages of the symbiosis between our approach and object recognition.

Keywords

Grasping in cluttered scenes, topographic features, machine learning, height accumulated features

1. Introduction

Robotic grasping is a well-examined research area, but its full potential is by far not explored yet. Many current approaches rely on object recognition (Morales et al., 2006; Huebner et al., 2009; Papazov et al., 2012) or object categorization (Wohlkinger and Vincze, 2010) and the subsequent application of pre-calculated grasps. Using known models is one approach to overcome the issue of incomplete data perceived from single views of a robot. The availability of object models enables the use of force and form closure grasp quality metrics (Mason and Salisbury, 1985; Li and Sastry, 1988; Pollard, 2004). However, object recognition and in particular segmentation are still open research problems for grasping (Bohg et al., 2014) and cannot guarantee reliable results for scenarios such as those depicted in Figure 1 (e.g. for the black pants with completely variant shape if it is placed on a pile of black clothes).

Other approaches try to identify simple geometric forms to enable grasp point detection (Varadarajan and Vincze, 2011; Miller et al., 2003), but also for these explicit shape abstractions robust segmentation is a necessary prerequisite.

Another approach is based on learning grasps using features from 2D images (Saxena et al., 2008a,b; Le et al., 2010; Jiang et al., 2011). Here the features used for learning are not directly in the grasp space, since the depth dimension resulting in the topographic surface data is not taken into account.

In this article we present features specifically developed for grasping without requiring *a priori* knowledge of the objects: Height Accumulated Features (HAF). Our method abstracts topographical information from perceived surfaces of objects, hence enables to learn how to grasp them, even if they are unknown or in a heap of objects. The presented approach targets parallel grippers of which two opposite fingers are symmetrically approaching the tool center point. This covers a wide range of objects used by

Automation and Control Institute, Vienna University of Technology
Vienna, Austria

Corresponding author:
David Fischinger, Automation and Control Institute, Vienna University of Technology, Gusshausstraße 27, Vienna, 1040, Austria.
Email: david.fischinger@gmail.com

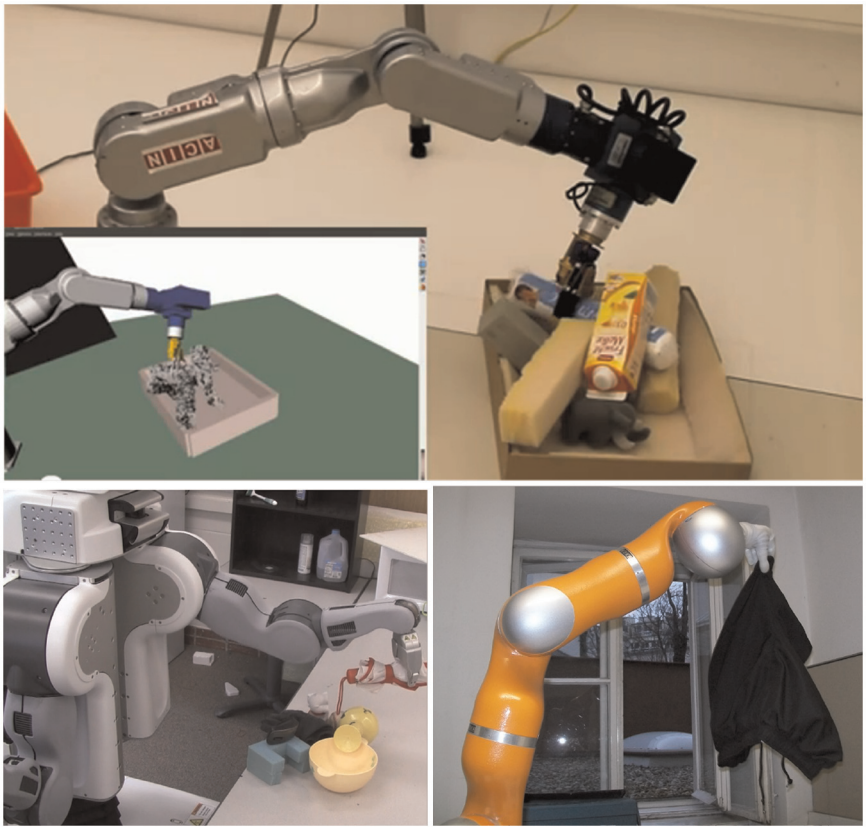


Fig. 1. Robots used for the grasp experiments, clockwise: Schunk seven-degree-of-freedom arm with hand prosthesis emptying a box in reality and simulation; Kuka LBR with Michelangelo hand prosthesis; PR2 clearing the table.

humans because as "for most objects, there is typically a small region that a human (using a two-fingered pinch grasp) would choose to grasp it" (Saxena et al., 2008a). In Section 8.6 we will show how the approach generalizes for more complex hands.

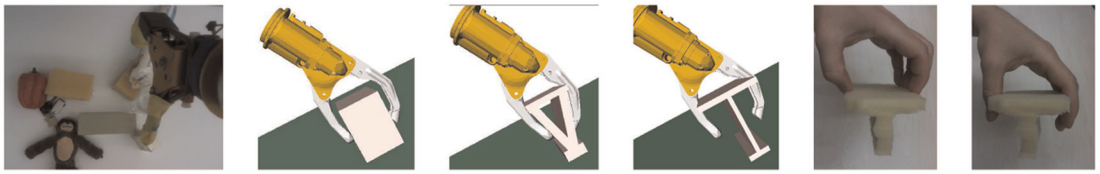
The key advantages of this approach are as follows.

- *Segmentation independent*. Our approach can solve complex tasks such as autonomously emptying a basket filled with unknown objects without the need for segmentation. It can be seen as a complementary approach to methods which need segmented input, such as Superquadric fitting or as a preprocessing module for object recognition by separating one object from a pile of items. Interactive segmentation by manipulation (e.g. examined in Kenney et al., 2009) is not the focus of this work.
- *Integrated path planning*. The majority of recently published grasping algorithms (e.g. Varadarajan and Vincze, 2011) handle grasp planning and path planning independently. Grasp approach directions and grasp points are calculated first, path planning with computation of inverse kinematics and obstacle avoidance is done afterwards. In contrast, our method learns to select grasp hypotheses which result in collision-free local paths for the manipulator used and the given approach vector.
- *Use of known depth regions*. A complete and correct object reconstruction is certainly an advantage for determining grasps, but state-of-the-art algorithms do not work reliably enough for the inherent degree of complexity of the cluttered scenes we aim to explore. Our approach focuses on grasps on perceived surfaces where the manipulator can approach objects without the need to estimate the surface of the object which is facing away from the camera (compare Figure 2).

This article focuses on the task of grasp identification and execution for unknown objects based on perceived 2.5D data. We address the problem of grasping objects with a single direct grasp.

The work presented in this article extends the research presented by Fischinger and Vincze (2012a); Fischinger et al. (2013) which covered the original idea of topographic features for grasping.

The main contribution of this paper is a significant improvement of topographic features and exhaustive evaluation using the *F*-score metric. Furthermore, we present an extension of the system to evaluate the optimal gripper opening width for path planning, a thorough analysis and comparison between our approach and 2D image-based approaches conducted on one state-of-the-art representative (Jiang et al., 2011), experiments on one additional robot platform, and the combination of topographic features with



(a) Top view from robot perspective (b) Box form (c) V-form (d) I-form (e) Fragile Grasp (f) Stable Grasp

Fig. 2. HAF motivation: in Figure 2(a) we see two rectangular surfaces of a gray and a yellow object from a typical robot's top perspective, no side surfaces are visible: (b)–(d) grasps for different object shapes, all of which have the same rectangular top surface shape; (e) an unstable grasp some force closure-based simulation environments would recommend; and (f) a stable top grasp a human would execute even if only the top surface of the object was perceived.

object recognition to aid reliable grasping if object models are available.

In the next section we discuss further work related to grasping unknown objects. Section 3 describes the idea, motivation, and calculation of HAF and Symmetry Height Accumulated Features (SHAF). Section 4 explains the process of how the feature values are used to determine optimal six-dimensional grasp configurations. We describe the machine learning methods used, a weighting heuristic to enhance the robustness of grasps, our technique to explore the entire grasp space using a trained classifier and the fine calculation of grasp points given our grasp representation using a simulation environment. In Section 5, we analyze the most efficient features for grasping, discuss the effect of SHAF and show the additional gain obtained from HAF by a comparison with learning grasps directly on height values. In Section 6 we use our framework to extend grasp options by taking the opening width of the manipulator into account. Section 7 discusses the scalability of the approach for diverse manipulators, elaborates its limitations, and shows options to adapt the approach for more complex grippers. Section 8 shows the evaluation of our approach on three different robotic platforms considering different aspects of each task.

2. Related work

The challenge of grasping unknown objects has been examined in numerous recent works. A great deal of them try to approximate original object shapes. Miller et al. (2003), Huebner and Kragic (2008) and Przybylski et al. (2011) used shape primitives such as boxes, spheres, cylinders and cones to approximate object shapes. Goldfeder et al. (2007) and Varadarajan and Vincze (2011) extended this approach by the use of Superquadrics as a more general basic geometric form for grasping. The resulting shape primitives were used to limit the number of candidate grasps to find the most stable set of grasp hypotheses. Approaches from Bohg et al. (2011) and Rao et al. (2010) are based on the observation that many objects possess symmetries and use this assumption for object completion before grasp calculation.

Another approach related to our work is from Klingbeil et al. (2011). They propose a grasp detection approach for a two-finger gripper to autonomously grasp unknown objects based on raw depth data. Their method tries to find a pattern in the scene that fits into the interior of the end-effector by maximizing the contact area between the robot's gripper and the perceived point cloud. This approach treats grasping as a shape matching problem similar to the work by Li and Pollard (2005), but does not require object models. Another noteworthy shape-matching approach related to our research is the work from Herzog et al. (2012). They proposed a template-based grasp selection algorithm operating on heightmaps which uses demonstrated grasp configurations and generalizes them to grasps novel objects. Katz et al. (2013) deal with the problem of clearing a table. They achieved a grasp success rate of 53% for cluttered scenes and learn push and pull actions in addition.

Saxena et al. (2008a,b) proposed supervised learning with local patch-based image and depth features for grasping novel objects in cluttered environments is proposed. This work is improved by Jiang et al. (2011) by adding the capability to learn optimal gripper opening width. The focus lies on learning features from 2D images, but one of the features is based on a comparison of object heights in predefined rectangular regions (in the remainder of this article, we refer to this method as the "rectangle representation"). This related feature, the performance and popularity of the approach in recent years and the ability to work in cluttered scenes made this work an excellent choice to compare our work with. Le et al. (2010) extended the method from Saxena et al. (2008b) to accommodate grasps with multiple contacts and a success rate of 80% is achieved for desk clearing experiments with 2–8 objects counting success/failure of the first grasp attempt for each object. Kootstra et al. (2012) used edge- and texture-based features on 2D images to build a hierarchical representation in three dimensions and their system is evaluated for scenes with one to three objects with grasp success rates of about 60%.

Most of the approaches in the last paragraph use learning techniques to optimize grasping capabilities based on simple features. In the following, we will describe in more detail powerful features for grasping, which we see as an

enhanced feature type suitable to replace simpler features for established grasping systems.

3. Topographic features

In this section, we describe and motivate two new feature types based on the topography of objects or scenes. HAF were developed specifically for abstracting grasp relevant information. SHAF are an additional feature type, to solve specific problematic cases of the basic classifier.

As stated by Saxena et al. (2008a)"because even very different objects can have similar subparts, there are certain visual features that indicate good grasps, and that remain consistent across many different objects". We see protruding object shapes as such subparts and designed the presented (S)HAF features to learn and identify such promising parts for grasping.

The basic concept of HAF was first published in Fischinger and Vincze (2012a) and the concept of SHAF in Fischinger et al. (2013). In Section 5, we show improvements over previous work regarding feature selection and performance.

3.1. Motivation of HAF

In recent years learning of grasps became very popular (Bohg et al., 2014). Applying machine learning directly to RGB-D data is still impractical due to the huge number of perceived points in a point cloud and the six-dimensional grasp output space. To reduce the complexity, numerous approaches learn grasping hypotheses based on 2D images taking into account color and intensity values. Although the human brain can detect potential grasp points from images (and, hence, this approach could be promising in the future), simple features based on 2D image patches seem to have clear limitations for grasping (see Section 8.5.3). Hu et al. (1999) found that kinematic parameters of the human grasp, such as path, and grip aperture, were determined by the 3D geometric structure of the target object, not the 2D projected image of the object. Therefore, we developed a new feature type which reduces the complexity of point cloud input, increases the structural value of input information as shown in Section 5.1 and is well suited for grasp-related machine learning due to the employment of grasp-relevant topographical information. Another challenge for learning grasps is the (at least) six-dimensional grasp output space (three parameters for position, three parameters for orientation) where all six parameters are strongly related. In Section 4.3 we describe our method to explore the whole grasp space using a trained grasp classifier.

The HAF approach is based on the observation that for grasping from top, parts of the end-effector have to enclose an object and, hence, go further down than the top height of the object. Furthermore, unlike other approaches, we do not try to guess the shape of partially visible objects. First, because this can fail in cluttered scenes or for asymmetrically shaped objects. And second, because our approach is based on the observation that a guess of the object shape is often not needed, for example if one surface is known and the manipulator can be placed around that surface: Figure 2(a) shows a picture from a typical view from a robot. For one gray and one yellow object, only the rectangular top surfaces are visible. Figure 2(b)–(d) show different scenarios for grasping when only the top surface is known and a manipulator goes down up to the object and closes: in Figure 2(b) the object is box-shaped and the manipulator touches the object roughly at the body center instead of the rim. In Figure 2(c) the hand more or less adapts to the object form. And even if the object consists primarily of a large top surface and few encompassing surfaces(see Figure 22(c)) a grasp would succeed if there is enough space to place the fingers around it. Despite the knowledge gap, even a human would not grasp only at the visible object parts like depicted in Figure 2(c) (like some force-closure-based simulation environments would recommend, even if the whole object model is available, compare Figure 31), but would go further down with his hand and use tactile feedback to stop the closing movement (Figure 2(f)).

Our idea (also stated in Fischinger and Vincze (2012a); Fischinger et al. (2013)) is to define small regions and compare average heights of these regions using discretized point cloud data. The height differences give an abstraction of the shape of the objects that enables the training of a classifier (using supervised learning) to determine if grasping would succeed for a given scene. Labeling of training scenes was done for parallel two-finger grippers with one DOF, specifically the Otto Bock hand prosthesis "SensorHand Speed" (Figure 2(a)–(d)).

For explanatory reasons consider the special case of top grasps (vertical approach direction of manipulator) of an object on a table. The term *height* can then be used intuitively and measures the perpendicular distance from the table plane to the points on the top surface of the object. A force or form closure grasp can only be achieved if parts of the manipulator will go further down towards the table than the top surface of the object. Hence, the region of the object top will on an average be higher than the area where the manipulator fingers are positioned. To speed up calculation, we discretize the point cloud, i.e. we generate a height grid H where each 1×1 cm^2 cell saves the highest z-value of points with corresponding x-and y-values (see Figure 3). One HAF is now defined as two, three or four regions R_i on the height grid together with a weighting factor w_i for each region. A feature value is defined as the weighted sum of all regions. So the jth HAF value f_j is calculated as

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

with

$$r_{i,j} = \sum_{k,l \in \mathbb{N}:(k,l) \in R_{i,j}} H(k,l) \tag{2}$$

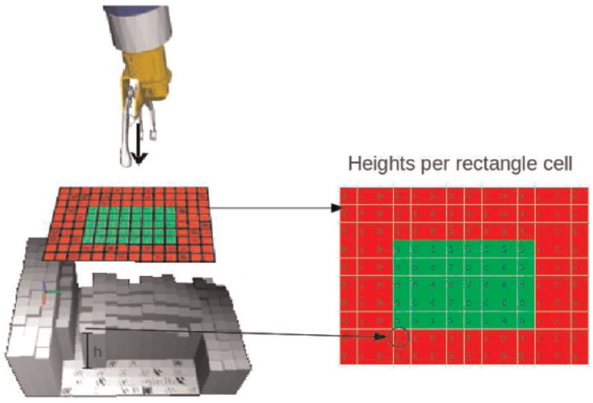


Fig. 3. The gray height grid resulting from point cloud disretization and an example feature with two regions. Region R_1 is the green inner region, region R_2 is composed of the red and the green area. For each grid cell of the regions the height is summed up per region. Each region sum is weighted with an individual factor and the sum (difference) of all regions (here two) gives the feature value.

where $nr\ Regions_j$ is the number of regions for feature f_j. Here $R_{i,j}$ indicates the ith region for the jth feature and is defined by the set of all pairs of height grid cell indices belonging to the region.

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

$$\mathbf{f} = (f_1, f_2, \ldots, f_{nrFeatures}) \tag{3}$$

For the initial set of experiments, we focused on features with two overlapping regions, where one region is completely inside the other region and a weighting factor $w_{i,j}$ is chosen such that the feature value is zero if both regions have the same average height, greater than zero if the inner region is higher, and smaller than zero if the inner region is lower on an average. For these features, the intuitive interpretation mentioned earlier holds. In Section 5, we show that more complex features with three or four regions or symmetry features lead to even higher discriminative results measured in terms of the F-score evaluation (Chen and Lin, 2006) metric. Overall, we tested about 71,000 features (70,000 of them automatically generated) for the experiments in this work and selected the top 300 to 325 with F-score selection to train a classifier weighing up time against detection performance.

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

Instead of a vanilla height grid H, we calculate an accumulated height grid AH, in which each location (x, y) of AH contains the height sum above and to the left of (x, y) in the grid:

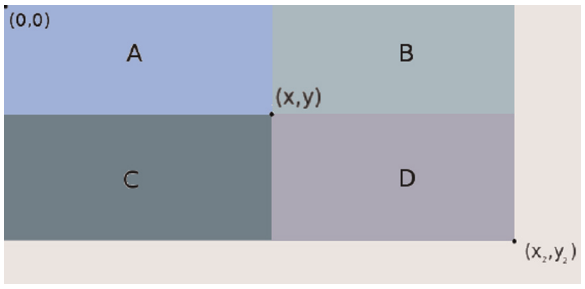


Fig. 4. To calculate the accumulated heights of region A a single AH reference is needed, $AH(A) = AH(x, y)$; area D requires four, $AH(D) = AH(x_2, y_2) - AH(x_2, y) - AH(x, y_2) + AH(x, y)$.

$$AH(x,y) = \sum_{x' \le x, y' \le y} H(x', y') \tag{4}$$

Using height accumulated rectangular regions, each region sum can be computed with four or fewer array references, see Figure 4.

3.2. Symmetry height accumulated features

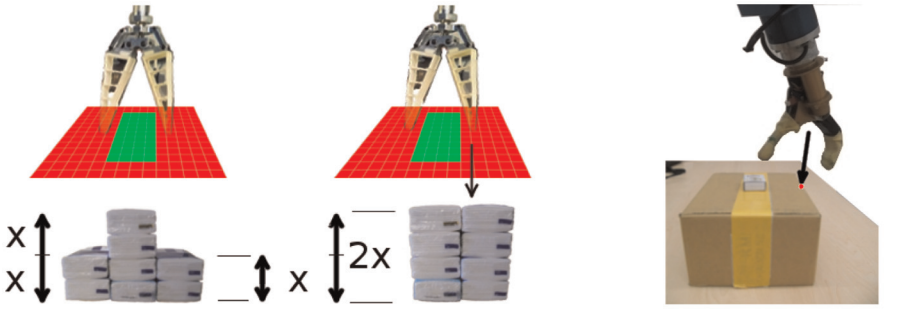
For special constellations such as a small object on top of a box on a table (Figure 5(b)), our HAF approach favors grasps at the edge of the box instead of grasps at the small object. HAF have the drawback that they are based on average heights of nested regions, hence there is no symmetry check when feature values are calculated. The same feature value can be achieved if the center region height exceeds both side region heights by x or if the center region and one side region have equal heights and exceed the second side region by $2x$ (see Figure 5(b) for illustration).

The feature value is no indication if the two fingers of the gripper (see Figure 5(b)) could go deeper than the object center on opposite sides of the object. This leads to false positives when using HAF for grasp detection, e.g. at the edges of a closed box. For "clearing-a-table scenarios" such "wrong" grasps did not occur due to our weighting system and the constellation and size of objects. As long as there is an easily graspable object on the table, this object is grasped first. Nevertheless, there is a need for improvement and it was decided to extend HAF by an additional feature type: SHAF.

Symmetry features have three disjunctive regions of equal size as depicted in Figure 6 where r_r, r_g, r_b are the accumulated heights of the region grid. The feature value f is defined as follows:

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

So we assign the minimal distance of accumulated heights between the center region and the side regions if the average center region height is the largest of the three regions, and -1 otherwise. Note that this function is either positive or -1 and that the regions are not weighted. In



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

Fig. 5. SHAF motivation (left): for the depicted feature with a green and a red region the feature value would be equal for both piles of tissues, but only the left pile is suitable for a grasp with the depicted Fin Ray gripper. The picture on the right shows the impact of this HAF property: a bad grasp being detected at the edge of a box.

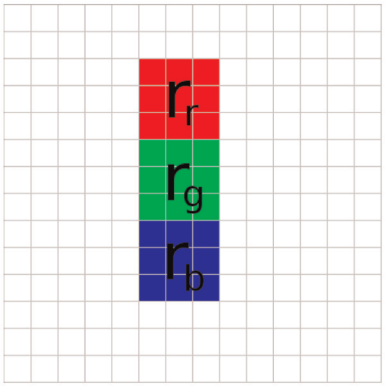


Fig. 6. Symmetry height accumulated feature: a typical example of the new feature type SHAF that solves the shown deficiency. All SHAFs have three equally sized, disjunctive regions.

Section 5.2 we discuss the impact of symmetry features on our grasp classifier.

4. From classification to actual grasp execution

In this section, we describe the process of selecting the best grasp options in the six-dimensional grasp space using our topographic features. In principle, this process is the same as stated in Fischinger and Vincze (2012a), but is summarized here for reasons of completeness, with some clarifications including a graphical system overview. First, we describe the learning process for grasping (Section 4.1). Then, the weighting heuristic to achieve more robust grasps (4.2) is explained. Finally, we describe our method to explore the whole grasp space (4.3) and show how fine calculation of grasp points and path planning is done with the OpenRAVE simulator (4.4).

4.1. Grasp classification training

In order to train a grasp classifier, we gathered 450 positive and 250 negative grasp scenes. A scene is composed of one or more objects on a table with the *z*-axis being perpendicular to the table and the origin located on the table surface. For supervised learning, we labeled the 450 positive examples, to be more specific: we labeled an *x, y* position such that a two-finger gripper (in our case, an Otto Bock hand prosthesis) positioned above the objects (with the tool center point at *x, y*) and oriented in such a way that the line between the tip of the thumb and the tip of the forefinger is aligned with the *x*-axis, would achieve a stable grasp of an object by approaching it (approach vector of the manipulator parallel to the *z*-axis) up to 1 cm and closing the fingers afterwards. Figure 8 illustrates the actual classification task. The 250 negative grasp scenes were labeled at positions with hardly any chance of a successful grasp. We augmented the training set by scaling, mirroring, truncating, and inverting the manually labeled positions to generate in all 8300 positive and 12,800 negative grasp example scenes. HAF and SHAF values were calculated on the 21,100 examples and used to train an SVM classifier with a radial basis function kernel from the implementation of LIBSVM (Chang and Lin, 2011).

Exemplary pictures of positive and negative training examples are shown in Figures 9 and 10. It should be emphasized that our training database contains enough different objects exhibiting similar local shapes to ensure that we do not learn object-specific grasps, but local-shape specific grasps. In other words: we learn local shapes that generalize across all types of objects and indicate graspable regions.

4.2. Grasp selection: Weighting system

For everyday scenes, the trained grasp classifier typically does not return an isolated grasp position, but a bunch of potential grasp points in a region (i.e. green area in the grasp classification grid of Figure 7, step 5). In general, a point centered at such a grasp region is a good choice for a stable grasp. Therefore, we developed the following weighting system: each point classified as a good grasp position is evaluated by

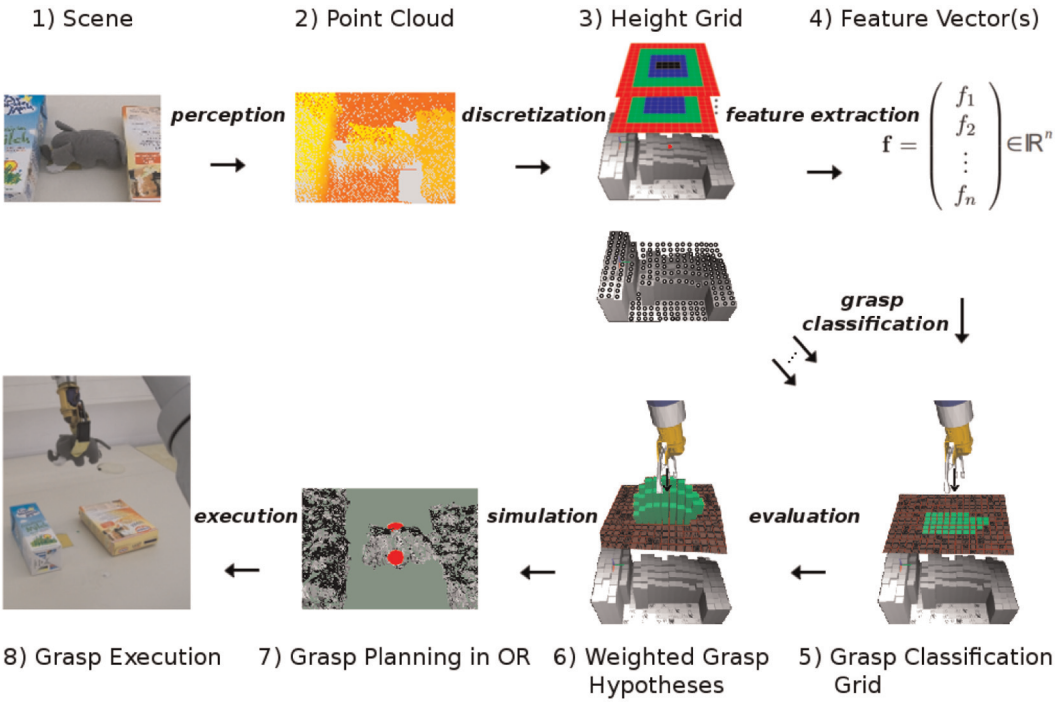


Fig. 7. System overview: this process diagram shows the steps in our grasping pipeline. For simplicity, the presentation shows only a solution for a fixed manipulator rotation, tilt angle, and gripper opening width. The point cloud of a scene is discretized first. To evaluate different manipulator roll/tilt angles and gripper opening widths, the point cloud has to be transformed before discretization (not depicted). For each grasp hypothesis, a feature vector of length n (number of features) is calculated. Then, a trained grasp classifier is used to get the grasp classification grid, whereby green indicates possible grasp positions. A weighting system evaluates the grasp quality, wherein better grasp positions are indicated by higher (green) bars. The overall top-rated grasp (along with estimated roll, tilt, gripper width) is sent to the simulation environment OpenRAVE, in which detailed grasp planning (how close the manipulator can approach the object) is done. For possible grasps, OpenRAVE sends the trajectories for path planning to execute grasping on the actual robot.

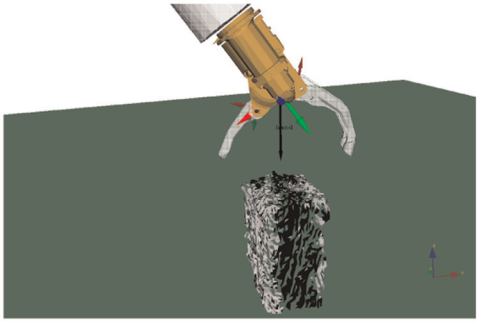


Fig. 8. Our grasp classifier learns voxel configurations if a hand motion in approach direction (black arrow) with subsequent closing of the fingers (1 cm before manipulator–object collision) would result in a stable grasp at the voxel. Note that the type of manipulator only influences the training examples regarding scale. Training examples will not change for other two-finger grippers.

$$v(r,c) = \sum_{x,y \in \mathbb{N}} I_{grasp}(x,y) \cdot w_{r,c}(x,y) \qquad (5)$$

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

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

Table 1 gives the weighting factors $w_{r,c}(x, y)$ for a grasp hypothesis GH.

In Bicchi and Kumar (2000), it was identified that there is a lack of grasp approaches that are robust to positioning errors. This practical weighting method enhances the robustness and stability of grasps. An example outcome of this weighting is shown in Figure 7, column "Weighted grasp hypotheses", wherein the height of the green bars indicates the quality of a potential grasp.

4.3. Grasp space exploration

Using our grasp classifier in combination with the weighting system, we can achieve the best grasp point for a given manipulator orientation and a top grasp. Our technique to explore the whole grasp space is as follows.

4.3.1. Roll. To get grasps for different hand rolls β, i.e. different orientations or manipulator rotations about the manipulator's approach direction, we rotate the initial point cloud iteratively (by *rollStep* = 15°) about the vertical z-

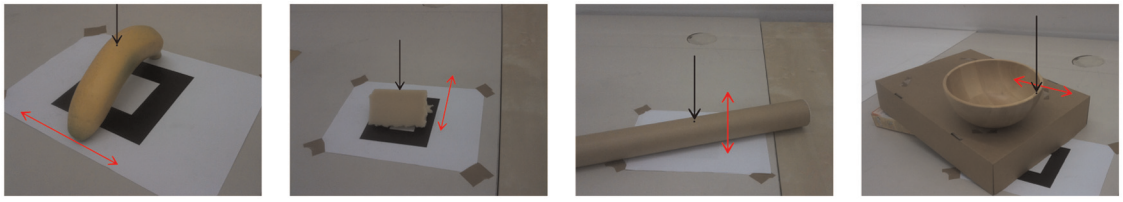


Fig. 9. Positive training examples for the training of the (S)HAF classifier. The black arrow perpendicular to the table plane indicates the approach vector, the red arrow parallel to the table plane indicates the closing direction of the gripper.

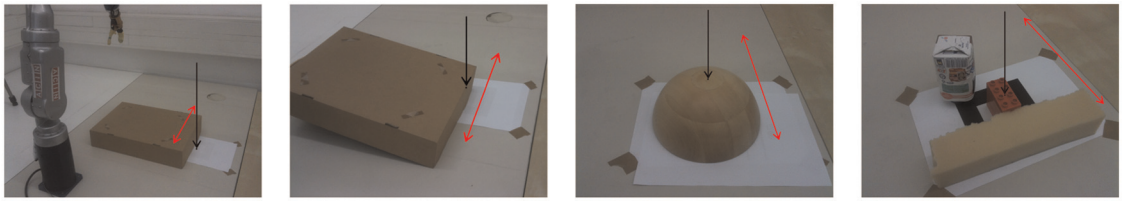


Fig. 10. Negative training examples for the training of the (S)HAF classifier. The black arrow perpendicular to the table plane indicates the approach vector, the red arrow parallel to the table plane indicates the closing direction of the gripper.

Table 1. Weighting values for evaluation of grasp hypothesis *GH*.

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

axis up to 180°, generate a new accumulated height grid and initiate the (S)HAF-based grasp point detection on this data. After selecting the top grasp points for the rotated scene, grasp points are transformed to the original world coordinate system. By using a roll angle range $[\beta - rollStep/2,\ \beta + rollStep/2]$ in the simulation environment and testing with manipulator rotation β and $\beta + 180°$ simultaneously, we achieve a exhaustive exploration of all possible rolls.

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

4.4. Grasp and path planning in simulation

After applying the weighting algorithm from Section 4.2, we select the top grasp hypothesis from all roll–tilt combinations and use the OpenRAVE simulator for path planning including determination of an appropriate distance between the manipulator and the object before closing the manipulator. OpenRAVE tries to approach the object mesh (i.e. an unsegmented mesh of all objects in the scene) using the calculated approach vector and manipulator roll angle until a collision occurs. Then it sets the manipulator position back by a standoff value which is dependent on the object position: in recent work, we do not use a fixed standoff of 1 cm but start with a standoff value of 1 mm. If this standoff leads to a collision of the gripper fingers with the table top or the ground, the standoff is increased until the closing fingers do not collide with the table or the ground anymore. Then the actual grasp points, i.e. contact points of the fingers with the object mesh in the simulation, are calculated. From the resulting hand position, OpenRAVE calculates the manipulator position 7 cm away (based on gripper size and fine tuned using experiments) and searches for a collision free path to place the manipulator there. For the last 7 cm to the object OpenRAVE calculates a straight path to the object if one exists. We chose 7 cm as a practical trade-off between a higher grasping robustness (regarding calibration inaccuracy or incomplete data) achieved by a straight approach trajectory with fixed manipulator orientation and the challenge to find inverse kinematics solutions for such trajectories.

To make the system more flexible, the calculated approach vector and manipulator roll angle are varied by

Table 2. Grasp classification success rate: HAF versus grid heights.

Feature type	Success	Success in %
Grid heights	2516/3928	64.05
HAF	3368/3928	85.74

$\pm\frac{1}{2} \times tiltStep$, respectively $\pm\frac{1}{2} \times rollstep$ degrees, to improve the possibility of finding a kinematic solution.

5. Evaluation of features and classifier

In this section, the feature quality is analyzed with the *F*-score metric from Chen and Lin (2006). Improvement is shown compared with previous work and the top 20 features are presented.

To give a thorough description of our approach, we mention two results from earlier work (Fischinger and Vincze, 2012a; Fischinger et al., 2013): In an evaluation in Section 5.1 we show the information gain obtained with HAF by comparing a grasp classifier trained using HAF with a classifier trained directly on height grids. In Section 5.2, we analyze the impact of symmetry features on the grasp classifier.

5.1. HAF classifier versus heights grid classifier

In Fischinger and Vincze (2012a), we claimed that there is significant additional information value with the use of HAF features. To prove this statement, we compared our SVM classifier trained with HAF against an SVM classifier that was trained using discretized heights only. We used 14 × 14 cm^2 grids for training, hence we had 196 height features. For training purposes, we used clearly distinctive scenarios, i.e. simple grasping situations as positive training examples and impossible grasp execution cases as negative examples. Since the grasp classifier showed an accuracy of more than 99 % for simple scenes, we gathered test scenes which were harder to classify, to provide a dataset which enables meaningful comparison. For example, a positive grasp example was not centered exactly at the rim of a bowl, but with a 2–3 cm offset. In practice, this situation would still lead to a successful grasp, but the classification gets harder. We gathered 50 positive and 50 negative test examples and generated in all 3928 test cases. Results are shown in Table 2.

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

5.2. Classifier combining HAF and SHAF

In this section we present that the combined HAF and SHAF approach is more susceptible to false negatives than HAF only. To examine the impact of symmetry features, we tested our classifier with HAF plus added symmetry features on the test set from Section 5.1, but extended the training data set by negative training instances such as a pile of books or a closed box where a point at the edge of the box was labeled as a bad grasp center.

We achieved an accuracy rate of 85.50 %. So the HAF classifier achieved a higher accuracy rate (85.74 %) than the classifier trained with HAF and SHAF. A thorough analysis exposed that HAF and SHAF classified negative examples with a success rate higher than 90 %, but did perform badly for positive examples. A subsequent analysis showed that HAF and SHAF are overall more sensitive to grasp situations: for example, a rim of a bowl with 2–3 cm offset with respect to the manipulator center easily results in a negative classification, although a small offset for a bowl would still be sufficient for grasping. Since we regularly achieved several potential grasps during experiments, tougher constraints for selecting positive grasps should not be considered a drawback, but as an improvement for detecting even more reliable grasps.

5.3. Top feature analysis for HAF and SHAF

For the experiments in Sections 8.2 and 8.3 302 out of 35,000 features were selected balancing time performance against classification quality. Feature values were calculated by (1). For the tests presented in Section 8.5 21 additional symmetry features were added to improve classification results. By improving a generation function for features with two regions and manually defining further features with three and four regions, including symmetry features, we could select new features with significantly higher *F*-score values compared with previous work.

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

$$F(i) = \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-} x_{k,i}^{(-)} - \bar{x}_i^{(-)2}} \quad (6)$$

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

In Table 3 we compare *F*-scores of the previously used 302 features to the new top 302 features after feature selection. We evaluated the *F*-score on two datasets. The first dataset had 13,692 very clear (easily distinguishable

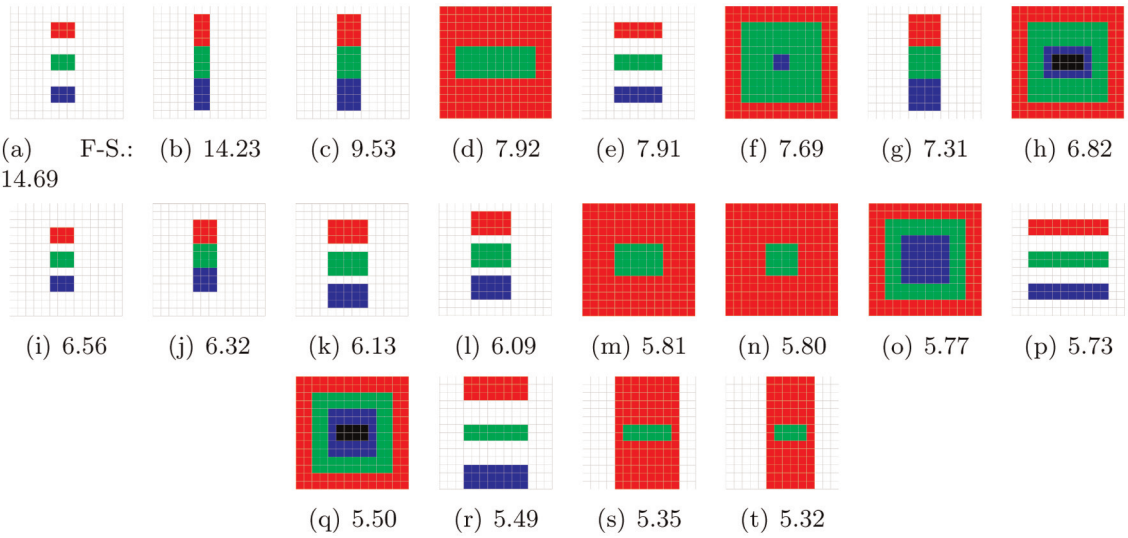


Fig. 11. Top 20 topographic features (HAF, SHAF) with *F*-score values (from Table 5).

Table 3. *F* -score comparison between new top 302 features and previous 302 features from Fischinger et al. (2013) on two data sets.

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

Table 4. Grasp classification success rate for top-ranked features tested on dataset 2.

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

between possible and impossible grasp) instances. The second dataset included a total of 17,620 positive and negative examples that were more difficult to classify. Thereby, the average *F*-score could be increased from 2.04 to 3.78 on the first dataset and from 0.54 to 1.57 on the second one.

Using cross-validation, we could show (Table 4) that even for the second dataset the 20 top ranked (new) features are sufficient for a 100% success rate in classification.

In Table 5, we list the top 20 (S)HAF, which are also depicted with their respective *F*-score values in Figure 11. The necessary weighting factors for regions of HAF (SHAF have no weighting factors) can be found in Table 5. From the top 20 features, 10 are symmetry features (9 of which are among the top 12). Furthermore, from the top 20 features, only two (ranked 19 and 20) were generated automatically with two regions and allow an intuitive interpretation ("if the center is higher, it is good to grasp there"). For three other features with two regions (4, 13, and 14) the weighting emphasis was on the larger region, meaning that the inner region had to be higher than the outer region to achieve a feature value of zero. From the remaining features, 6, 15, and 18 have three regions and 8 and 17 have four. In summary, these results demonstrate that a higher complexity of features achieves better results, even though an intuitive interpretation is not that easy anymore.

6. Pre-grasp gripper width calculation

In the previous sections, we presented an approach to calculate grasps in a six-dimensional grasp space (defined by the position and the orientation of the gripper). We assumed an initially fully opened gripper. In Figure 12 a situation is depicted where grasping with an initially fully opened gripper would not succeed because the gripper cannot reach the grasp position due to collisions with obstacles. In this section we present how we extended our system to learn grasps in a seven-dimensional grasp space, showing how to determine a suitable opening width for target approaching of the manipulator by iterative use of our approach.

Table 5. Top twenty features ranked by F -score value.

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

The initial idea of HAF was to learn and detect areas where parts of a robotic gripper can enclose the center of the object parts. The classifier and the weighting system identify suitable grasping positions for a gripper with known opening width. To test if a partly opened gripper can enclose an object we use our approach with only small adaptations: by scaling the point cloud with respect to the degree of gripper closing, we can simulate different opening widths. For example, to test a half-opened gripper (opening width = maximum opening width/2), we scale the point cloud (after rotation and tilt) by the factor 2. If the system detects a grasp with a high evaluation score, grasping at that position with a half-opened gripper will probably succeed. In other words, to determine the best opening width, we iteratively determine the best grasp hypothesis for different opening widths using the scaling factor S for the point cloud by the means of

$$S = \frac{1}{(\textit{opening width as a fraction} \in (0, 1])} \tag{7}$$

and finally select the grasp hypothesis with the overall best score.

In experiments with a newly developed household robot (Fischinger et al., 2014), we tested the procedure with a Festo Fin Ray gripper that is able to partially close its fingers and showed that this improvement enabled grasping of objects in scenes where grasping was not possible without optimizing the opening width.

Our grasp detection was tested along with variable opening widths of the gripper before it closes. The roll angle of the gripper was fixed for the proof-of-concept experiments.

In order to test the opening width parameter, we used opening widths of 1, 0.5 and 0.33 times the maximal width.

(a) (b)

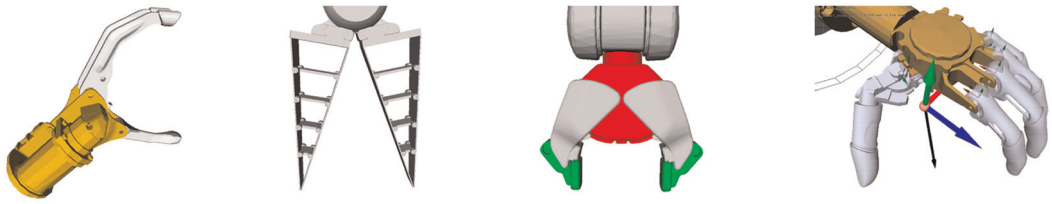
Fig. 12. Motivation of the pre-grasp gripper opening width: the pictures show a scenario where an initially fully opened gripper could not succeed, because gripper–object collisions would occur while approaching the final grasp position.

Point cloud scaling (to simulate a partly opened gripper) was done only for the gripper closing axis since the gripper has only two anti-podal fingers. In scenarios especially arranged for testing the gripper width, we verified the functioning of our approach.

We could achieve valid solutions for setup scenarios such as that depicted in Figure 12, where the calculation of a proper gripper opening width is crucial, since a fully opened gripper could not approach the object in the required way for grasping.

7. Scalability for diverse manipulators

The presented approach is intended and motivated for 1-DOF parallel grippers where two opposite fingers are symmetrically approaching a protruding object sub-part from opposite directions. However, many robotic grippers of that type can be used in a similar way.



(a) SensorHand Speed (b) FinRay Gripper (c) PR2 Gripper (d) Michelangelo Hand

Fig. 13. Overview of manipulators used in experiments.

One reason for an evaluation of the approach on four different hardware platforms is to demonstrate its scalability for different grippers with zero or only minor changes. For all experiments in the following sections no gripper-specific classifier training was done. Thereby, we show that the grasp classifier is usable for different manipulators. However, for considerably bigger or smaller manipulators, scaling of the input point cloud prior to the execution of the grasp classifier enables the usage of the approach without altering the classifier (according to the gripper opening with in Section 6).

As mentioned earlier we do not calculate specific grasp points on objects (or actually the incompletely perceived object surface data of objects), instead we calculate an approach vector for the object. For grasping, this approach vector is aligned with the manipulator specific approach vector. This alignment with subsequent grasp planning in simulation enables our method to work with different manipulators. Crucial for the grasp performance is the definition of the gripper's approach vector (position and direction).

An advantage of this approach is its division into two parts: (a) detection of promising grasp positions related to an object with a HAF/SHAF classifier and (b) the gripper-specific alignment, done in simulation using the gripper-specific approach vector and the TCP). Even for more complex grippers, there is no need for changing the classifier. Instead heuristics and simulation can be used in a later state of the approach to adapt the alignment process. A valid pose for the robotic hand can be found, given the object approach vector indicating the protruding object part or completely discarding a position and trying the next best one identified by the classifier.

This article presents research on grippers with simple hand morphologies, except the Michelangelo hand. This gripper is the most complex hand used in our experiments (see Sections 8.6 and 8.7). We used it with one DOF, controlling open–close motion of all fingers simultaneously. Due to the complex finger trajectories a single (gripper-specific) approach vector cannot guarantee perfect positioning of the manipulator for both objects depicted in Figure 14 (assuming that the calculated approach vectors intersect with the center point of each object). Simplified trajectories for the forefinger and the thumb in two dimensions are depicted in Figure 14.

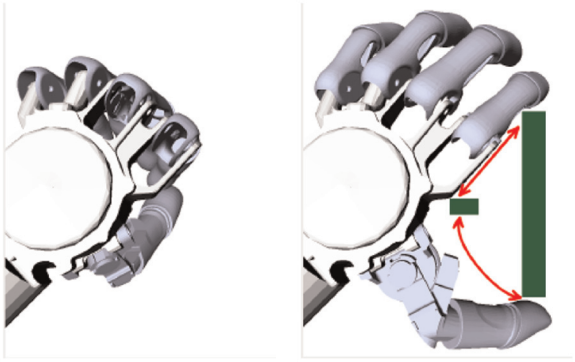


Fig. 14. Michelangelo hand in closed and open position. Right: Michelangelo hand in perfect position for grasping a small and a long rectangular shaped object (both green). The red arrows indicate a simplified finger trajectory when the hand is closed.

The green rectangles represent two objects. For both objects, the hand is optimally positioned (for a fixed hand rotation) such that closing the fingers should result in valid grasps. If the objects were swapped in position, grasps would fail for both of them. This exhibits a problem: our approach (ideally) defines the center of these objects as the center of a grasp (which is equal to the mid point of two grasp points). Thus, defining one approach vector for the manipulator can only position the hand ideally for grasping one of the objects, but not both.

The analysis of the Michelangelo hand showed an insufficiency of the approach regarding a complex hand. To cope with this insufficiency we want to shortly discuss three approaches.

- *Ignoring the problem as a side effect*. Ignoring the insufficiency was what we did in the experiments in Section 8.6 and 8.7. By choosing a trade-off approach vector for the gripper and using our approach without adaptation (the classifier was not trained again, training data and features stayed the same) we showed that the approach is robust enough that even with theoretical proven insufficiency for constructed cases the grasping process is still working well for a complex gripper such as the Michelangelo hand.
- *Relearning the classifier*. In order to achieve valid grasps for more complex grippers, one could relearn

Table 6. Overview of experiments.

Experiment	Section	Name	Robot Arm	DOF	Manipulator
1	8.2	Clear Table	Schunk	7	OB Hand protheses
2	8.3	Empty Basket	Schunk	7	OB Hand protheses
3	8.4	SingleObjects	PR2	7	PR2 Gripper
4	8.5	Clear Table	PR2	7	PR2 Gripper
5	8.6	SingleObjects	Kuka LBR	7	Michelangelo Hand
6	8.7	ObjectRecog.	Kuka LBR	7	Michelangelo Hand

the classifier. However, this would be a very laborious work, as a completely new labeling for all training data would be needed and the HAF and SHAF classifier would be limited for specific hands, which is against the intention to have a generalist approach.

- *Heuristic adaptation of the gripper in the simulation*. A third way to proceed is the use of a heuristic which takes the local surface into account and adapts the hand position with respect to the width of the object mesh in the space between the opened fingers. Such a heuristic would have to be developed gripper-specific, but could integrate the exact hand kinematic, the quality of given data (e.g. if an complete object model is given), and could use advantages of simulation environments.

8. Robotic experiments and evaluation

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

8.1. Goals of experiments

In the following sections, we present a series of experiments, in which we evaluated our approach with different robots (see Table 6). In each experiment, we focused on specific aspects (for an overview see Table 7).

In Section 8.2, we present an experiment, in which a table with objects was completely autonomously cleared (aspect: "*Autonomous*" from Table 7) using HAF ("*HAF*"), without any user input. For this experiment in clutter ("*Clutter*"), we verified the use of our grasp space exploration method ("*6D Exploration*"). In Section 8.3, an experiment is presented in which a box of objects ("*Box obstacle*") was autonomously unloaded. Sections 8.4 and 8.5 present experiments performed on a PR2 in which we compared our approach ("*HAF*", "*SHAF*") to two other grasp detection methods and give a detailed comparison between our method and another state-of-the-art approach ("*Comparison SOTA*"). In Sections 8.6 and 8.7, we use our method with a more complex manipulator ("*Scal. Manipulator*") and show the gain in combining our method with object recognition ("*Recognition gain*").

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

8.2. Clearing a table with a Schunk arm

In the first experiment, we demonstrate the capability of our approach by grasping objects from a table. The grasp classifier was trained with Height Accumulated Features. The focus of the experiments are on the capability to grasp in clutter and the autonomy of the system (no user interaction after placing all objects and starting the system). Tests for 11 different scenarios (Figures 15(a)–15(j)) with 5–9 objects were done. The experiments and results are an extension of work previously published in Fischinger and Vincze (2012b). Figure 16 shows all 19 objects used. Most

Table 7. Overview of aspects focused in the experiments (x). A dot "." indicates relevance of the aspect for the experiment, but not as a primary focus.

ASPECT \ Exp.	1	2	3	4	5	6
HAF	x	·	·	·	·	·
6D Exploration	x	·				
Autonomous	x	x				
Box obstacle		x				
Clutter	x	x		x		
Comparison SOTA			x	x		
Scalable manipulator			·	·	x	·
Deformable objects	·		·	·	x	
Recognition gain						x

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

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

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

Fig. 15. Test cases for clearing the table.

of them are graspable from any configuration. The two bowls become non-graspable for the manipulator if grasp manipulations result in an upside down position.

8.2.1. Test setup. For the grasp execution, we use a Schunk 7-DOF robot arm with an Otto Bock hand prosthesis "SensorHand Speed" with one DOF. For perception of data we used two Microsoft Kinect cameras with PrimeSense sensors positioned on opposite sides of the object pile. The two cameras were triggered with a time offset to overcome overlapping laser pattern projections that lead to lower data quality.

8.2.2. Results. Table 8 shows results from the 11 executed trials for clearing the table. In all cases, the table was successfully cleared after placing the objects and starting the system without any further intervention from the experimenter. A grasp was rated as successful if an object was grasped, lifted, and delivered to a plastic box 1 m away from the original position of the object pile. Since object

Fig. 16. Objects used for clearing the table with a Schunk arm.

Table 8. Clearing the table results for all trials.

Run	Objects removed	Table cleared	Grasp failures
1	5/5	yes	0
2	5/5	yes	0
3	6/6	yes	1
4	6/6	yes	0
5	7/7	yes	0
6	7/7	yes	0
7	8/8	yes	1
8	8/8	yes	0
9	9/9	yes	0
10	9/9	yes	0
11	7/7	yes	3
Sum	**77/77**	**11/11**	**5**

manipulation for a number of objects in cluttered scenes can lead to situations where grasps are not possible any more (due to kinematic reachability after moving an object out of the graspable area or due to missing grasps because of an upside-down bowl) it was a priori not granted that the table will be cleared completely in each run. Videos of the test runs are available at: http://www.youtube.com/user/clearingthetable and in Multimedia Extension 1.

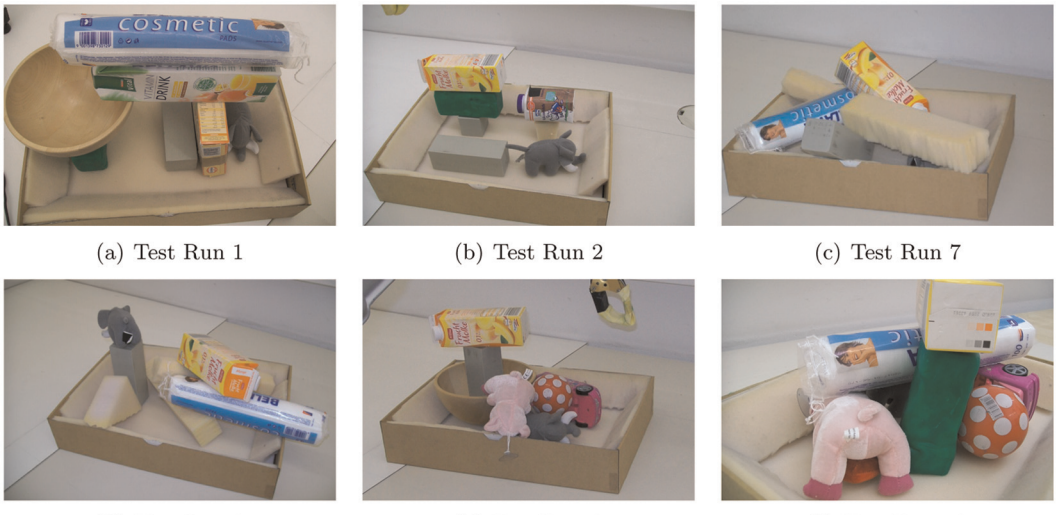
Table 9 gives a detailed overview of grasp failures per trial and object. In test run 7, the plastic bowl was grasped together with the headset. Since the objective of this experiment was to clear the table top without segmentation of objects, this grasp was assessed as to have successfully grasped both objects. For the implementation used, it takes 2–3 seconds to calculate the top grasp and about 1 second for grasp and path planning in OpenRAVE. Overall, we could achieve 77 successful grasps out of 82 tries (93.9%) for grasping from piles of unknown objects.

8.3. Emptying a basket with a Schunk arm

The goal of this follow-up experiment was also to prove the feasibility of our HAF approach, but this time in a scenario where a basket filled with unknown objects should be autonomously emptied. See Multimedia Extension 2. The basket as a non-graspable obstacle with variable position and orientation significantly increases the complexity of the task compared with the previous experiment of clearing a table, as the number of executable grasps decreases. Tests for 10 different scenarios (see Figure 17) were performed, details of the test setting can also be found in Fischinger and Vincze (2012a).

Table 9. Grasp failures per object for 10 trials. Entry of last column is number of failures divided by number of tries. A dash "—" indicates that the object was not used for this run.

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



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

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

Fig. 17. Examples of test scenarios for empty the basket.

Table 10. Grasp failures per object for 10 trial runs. The last column shows the number of failures divided by the number of tries. A dash "—" indicates that the object was not used for this run.

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

8.3.1. Test setup. For grasp execution, again a Schunk 7-DOF robot arm with an Otto Bock hand prosthesis "SensorHand Speed" was used. An accurate and robust basket detection for position and orientation was crucial for these tests. Developing this module with 100 % reliability and coping with basket occlusions is challenging, but not the focus of this work. Perception was again done using two Microsoft Kinect cameras positioned on opposite sides of the basket.

8.3.2. Results. Table 10 gives a detailed overview of grasp failures per test run and grasped object type for the 10 trials of emptying a basket. Although the basket increases the complexity of the task significantly, in all cases the basket was successfully emptied after placing it at a random position in the graspable area (limited only by the kinematics of the arm) and starting the system without any further experimenter intervention. In 5 out of 10 test runs, the basket was emptied without a single grasp failure. Regarding only first tries to grasp an object, our approach succeeded in 61 out of 70 cases, giving a success rate of 87.1 %. The used implementation needed 2–3 seconds for grasp calculation and about 1 second for grasp and path planning with OpenRAVE.

Table 11 shows grasp error analysis. Three main issues were identified as causing grasp failures: incomplete point cloud data, path planning errors in the simulation, and

Table 11. Analysis for grasp failures per object and test run. Failures are caused by insufficient point cloud data (Data), wrong path planning (PP) or unstable grasp points (HAF).

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

Fig. 18. Objects used for emptying the basket experiments.

suboptimal grasp points. For a deeper failure analysis we refer to (Fischinger and Vincze, 2012a), but we wish to note that out of 22 failed grasps, 18 happened when only one or two objects where left in the basket. Two particularly challenging object constellations, which caused seven and four grasp failures in a row were responsible for that. However, this fact also demonstrates that our weighting system for grasp selection is capable of identifying easily graspable objects first. It also shows that the basket brings a complication which should not be underestimated, since most of these 18 grasp failures were related to objects adjacent to the basket border. It also reveals potentials for enhancements of the grasp learning system. To avoid obstacles (i.e. basket borders or currently skipped objects), the system chooses grasp points near the edges of an object, which can result in objects slipping out of the manipulator's fingers. Figure 19 shows the scenario where grasping failed seven times in a row due to a combination of grasps selected near object edges (also because of missing alternatives) and insufficient point cloud data.

8.4. Grasping single objects with PR2

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

8.4.1. Test setup. SHAF as well as basic HAF were used to train the grasp classifier for this experiment. We implemented the demo scenario on a PR2 robot platform at Cornell University. For grasp execution, we used the left 7-DOF arm, with a two-finger manipulator. We did not adapt the grasp classifier (training and initial tests for our grasp classifier were done for an Otto Bock 1-DOF hand prosthesis) for the PR2 gripper, demonstrating the scalability of our approach for diverse manipulators (see Section 7 for explanation). Point cloud perception of scenes was done using one Microsoft Kinect camera mounted at the head of the PR2. After perception of a point cloud, points of the table surface were deleted automatically as well as points not relevant for grasping, e.g. points outside of the kinematic reachability of the PR2 arm. From the remaining point cloud a mesh was generated which was then used in OpenRAVE for path planning and grasp simulation. The Rectangle Representation method from Jiang et al. (2011) and our topography-based grasping used the same function for grasp simulation in OpenRAVE from which the physical robot was also controlled. In other words, the system was not aware of the method that generated the grasp hypothesis.

In this experiment on grasping single objects on a table, three methods for calculating grasps (where a grasp is represented by a 3D point, an approach vector, and a roll angle for the manipulator) were compared. The first method is the default grasp planner from the robotics simulation environment OpenRAVE. A detailed description of the method used can be found in the OpenRAVE documentation about

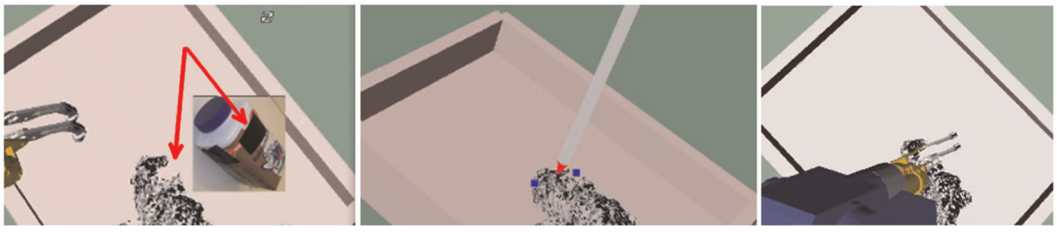


Fig. 19. Unstable grasp points due to insufficient data. Left: misleading hole in a data mesh due to black tape on the milk carton (to cover brand label). Center: calculated grasp points and approach direction. Right: grasp execution in simulation; objects slipped out of manipulator repeatedly when executed on a real robot.

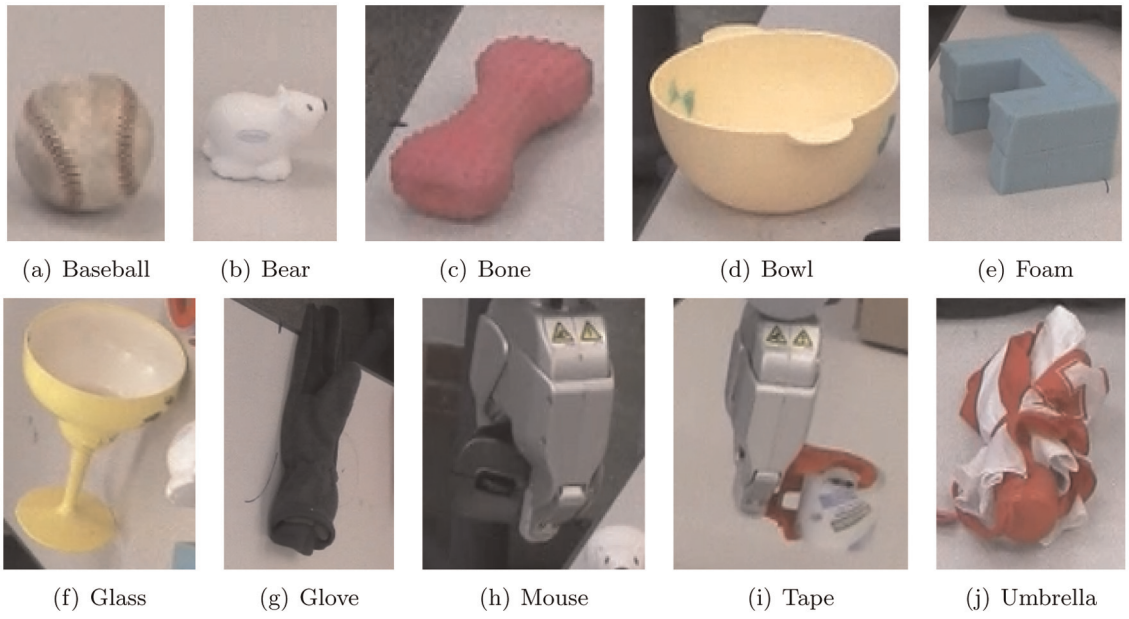
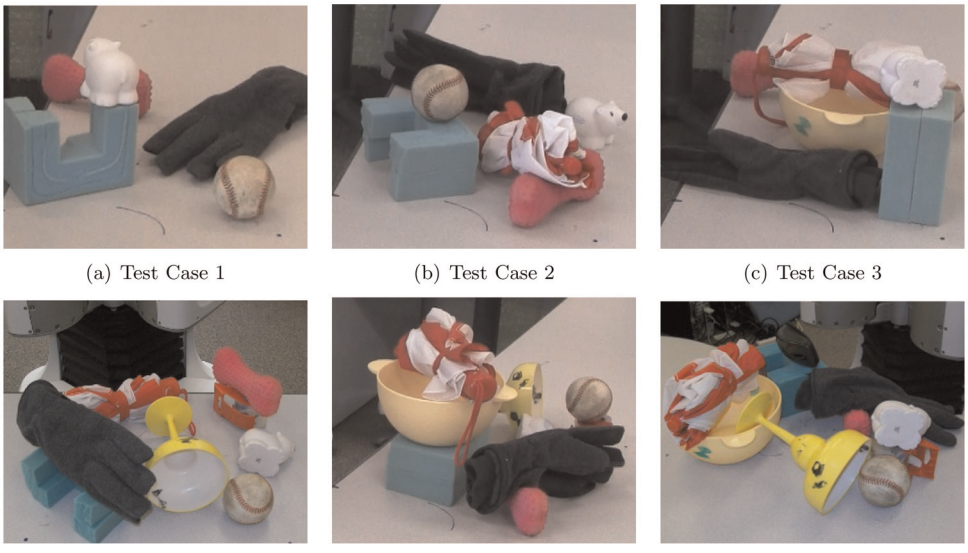


Fig. 20. Test objects used for clearing the table with a PR2.

the grasping module (Diankov, 2012). We expressly state that the grasp calculation with this default OpenRAVE method is not completely appropriate, since force closure calculation assumes a complete 3D object model. Despite this shortcoming, we preferred this method to a more random generation of grasp points and approach vectors as a basic benchmark algorithm. Due to path planning, inverse kinematic, and performance reasons we also had to restrict the OpenRAVE grasp selection method to grasps with a mainly vertical approach direction (70 % vertical). The second grasp method was the Rectangle Representation from Jiang et al. (2011). The third method was ours, using HAF and SHAF topographic features.

Nine out of ten objects for this experiment were chosen from an object box at Cornell University, of which no object was ever used before for training our classifier or any other part of our approach. To pick the tenth object, we asked an uninvolved person to pick an arbitrary object from the lab that fits between the PR2 gripper, which resulted in the picking of a computer mouse. All objects are depicted in Figure 20 in one of the grasp poses used for this test. For test methods 2 and 3, we used top grasps (with vertical approach direction) only, due to three reasons. First, for a given manipulator orientation and a straight approach trajectory in the final few centimeters approaching an object, it is hard to find an area of 35×40 cm^2 for top grasps where inverse kinematic solutions are possible for all manipulator roll angles. Each allowed deviation from vertical grasp reduces the size of this object region where grasps can be executed. Second, for test method two no code was available to calculate grasp hypothesis other than those from the direction of the camera view. And third, in this test scenario, in contrast to experiments in Sections 8.2 and 8.3, point cloud perception is done by a single camera. Due to incomplete point cloud data (especially occlusions), path planning gets more unreliable as the approach direction deviates from the camera view direction.

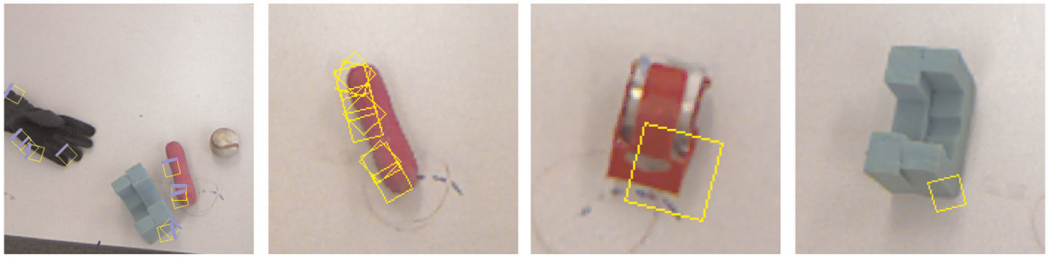
For testing, each object was placed five times in different poses (i.e. varying orientation and position) in a 35×40 cm^2 region where inverse kinematic solutions for vertical grasps were generally found. This was done manually by the experimenters for the first five test objects, for



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

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

Fig. 21. Test cases with 5, 6, 7, 8, 9 and 10 objects for clearing the table.



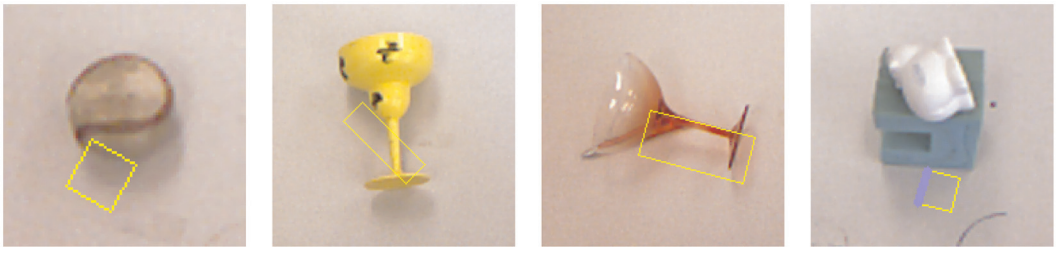
(a) Top 10 grasps with gripper closing direction indicated violet (b) Top 10 grasps (c) Top result (d) Top result

Fig. 22. Calculated gripper width for Rectangle Representation. Examples of the grasp learning approach Rectangle Representation which is mainly based on 2D images. If the manipulator approaches an object with an opening width corresponding to the Rectangle Representation, the gripper would not be able to encompass the object, but would touch it and hence would be stopped by the path planning routine before it has reached a position at which closing the gripper would succeed as grasp.

the last five objects, this was done by an uninvolved person. After placing an object in the marked area, photos were taken from different angles to replicate the scene for all three test methods. A grasp is defined as successful if the robot arm lifts the object and holds it for at least 15 seconds.

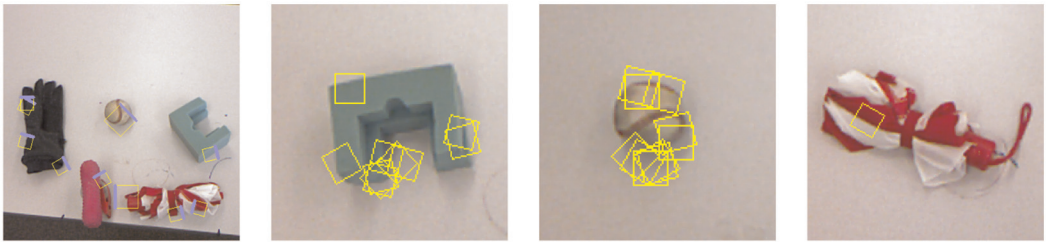
8.4.2. Comments on available Rectangle Representation code. We express our gratitude to Professor Ashutosh Saxena and his group at Cornell University for sharing their grasping code with the community and giving us the opportunity to work with their resources to enable an objective comparison of grasp methods.

The provided code (available from http://pr.cs.cornell.edu/grasping/rect_data/data.php/) needs as input an image of the grasping area without objects to grasp and limits the grasp detection to image regions where objects were placed afterwards by comparing the current and the former image. There is a need to know how the empty grasp area looks like without objects from a fixed camera view (to trigger a "background subtraction" mechanism), which makes the approach inflexible with respect to camera or robot movements, and hence unsuitable for mobile robotics. We enhanced the code by using the path planning of OpenRAVE to find an appropriate distance between the object and the manipulator while grasping instead of using a fixed offset from the detected 3D point in the center of the rectangle from the Rectangle Representation. Using the original approach would often have led to a collision between the manipulator and grasped object because the learned opening width was too narrow (see Figure 22) or there was simply no space for the manipulator fingers since no object collision implementation was available to recalculate the fixed offset. To obtain better results for



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

Fig. 23. Shadow: Rectangle Representation is sensitive to shadows. It happens that grasps are detected at shadow boarders.



(a) Top 10 grasps with closing direction (b) Top 10 grasps (c) Top 10 grasps (d) Top result

(e) Top result

Fig. 24. Edge focused: Rectangle Representation often relies on edges in 2D images. Color changes often correlate with object boundaries and thereby indicate potential grasp positions, however, as the examples show this does not necessarily result in stable grasping points.

Rectangle Representation, we therefore grasped with maximal opening width (which was still not wide enough to grasp a baseball where the rectangle was centered usually at the edge of the image and the manipulator touched the ball when approaching: see Figure 24(c)).

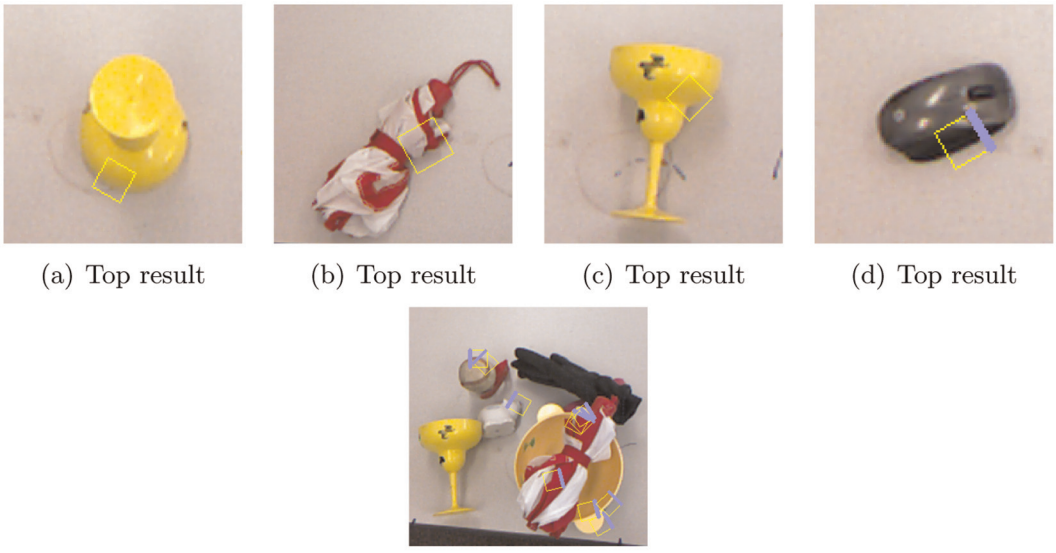
8.4.3. Grasping single objects with PR2: Results. Results are summarized in Table 12. The listed average time in seconds for the OpenRAVE algorithm is the time for grasp calculation. For Algorithms 2 and 3, the time is measured from the point of receiving the point cloud data (and image data for method 2) to the output of the grasp hypothesis. Grasp and path planning time for the latter two methods is about one second. Being aware that our algorithm will still be superior regarding time performance, we chose a very high-quality threshold parameter for our algorithm. This quality threshold stops our algorithm as soon as a grasp evaluation is better than the threshold (so other grasps are not evaluated anymore). Although with smaller threshold values good results are achieved, we decided to go for higher grasp quality than faster performance.

Our method succeeded in 46 out of 50 trials, giving a success rate of 92% compared with 58% for Rectangle Representation and 20% for the OpenRAVE force closure grasp selection. Overall, 8 out of 10 objects were grasped 5 times without a single failure. The Rectangle Representation only achieved a higher success rate for the Martini glass. For this object the (S)HAF grasps were not optimal and during the approaching of the manipulator, a slight touch of the object caused the lying Martini glass to roll away. Reasons why the Rectangle Representation performed considerably worse are depicted in Figures 22–26. Often grasps were detected at object edges, which was the main reason for the 34% gap in grasp success rate. Further grasp quality analyses of the main methods is done in Section 8.5.3.

Although we invested significant time in finding optimal parameters for the OpenRAVE method, the algorithm

Table 12. Grasp success rate (Success) in % and performance time in seconds: OpenRave versus Rectangle Representation versus (S)HAF.

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



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

(e) Top 10 grasps with closing direction

Fig. 25. Surface independent: Rectangle Representation detects grasps at positions the manipulator cannot reach because it would touch parts of the object before the manipulator can successfully close. In Figure 25(e)Form5 detected for the baseball and the white toy bear.

(a) Height dependence (b) Orientation (c) Perception (d) Path planning (e) Grasp center not on object

Fig. 26. Examples of Rectangle Representation learning (with top evaluated grasp), images from left to right showing (a) a weak impact of height features, (b) a bad result for grasp orientation, (c) a scene in which perception of a white toy bear was only possible after augmenting with supporting texture, (d) a scene which illustrates a potential drawback when grasp detection relies too much on 2D features: a grasp center detected at a position not belonging to any object.

Table 13. Grasp success rate (Success) in % and performance time in seconds: Rectangle Representation versus (S)HAF.

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

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

8.5. Clearing table with PR2

The goal of this follow-up experiment was to compare our topography-based approach using HAF and SHAF to the same Rectangle Representation approach used in the previous experiment, but this time for a cluttered pile of 5 to 10 objects on a table. Furthermore, we can show with this experiment that one camera is not only sufficient for clearing single objects, but also for objects in cluttered scenes, which is in contrast to the experiment in Section 8.2 where we used two cameras with different view angles for enhanced data perception.

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

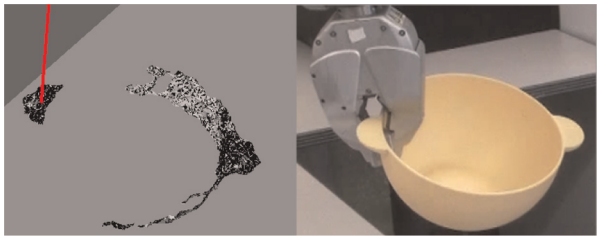


Fig. 27. Incomplete perceived point cloud data (depicted is the result of a plastic bowl from a camera mounted at the robots head) is one reason for failed grasps.

8.5.2. Results. Table 13 shows success rates for all six test cases and the average calculation time per grasp in seconds. Pictures of all test cases are shown in Figure 21.

Using our approach the system successfully grasped 39 objects out of 45 tries, giving an overall success rate of 86.7%. Rectangle Representation achieved 26 out of 45 successful tries, giving an overall success rate of 57.8%. Although, as mentioned above, our parameter setting was suboptimal with respect to time performance, our algorithm was three times faster than the Rectangle Representation algorithm. The six grasp failures for our approach had different underlying reasons. In test case 5, we failed to grasp the glove because only for this single case the "integrated path planning" failed and parts of the bowl prevented the manipulator from approaching the glove as far as needed in the simulation environment and hence in the real grasp execution. For the other five failures, non-optimal HAF grasps were each time partly responsible. But in each of these cases, other factors also contributed to the failure: three times the object (toy bear in TC 3, pink bone in TC 4, Martini glass in TC 5) was touched and moved by the manipulator out of the initial position. One reason for premature touching of objects is incomplete data. For the Martini glass and the bowl the interior region was in general badly perceived as shown in an example of the bowl in Figure 27. Hence, path planning resulted in paths wherein the manipulator collided with the (missing in simulation) interior surface of the bowl or Martini glass when grasping at the rim.

Videos of all test cases of the clearing the table scenario with PR2 can be found at http://www.youtube.com/user/

clearingthetable/and partly in Multimedia Extension 1. The demo code used for the experiments (along with whole framework, in addition to the core algorithms) for grasping unknown objects with a PR2 (Rectangle Representation and (S)HAF) is available as ROS packages. This contribution enables a valid comparison of other grasp detection algorithms with the two methods presented in this section. The code is available at http://pr.cs.cornell.edu/grasping/rect_data/data.php and in Multimedia Extension 4.

8.5.3. Comparison: topographic features versus 2D image features. In Figures 22–26, we give examples to explain why our approach performed considerably better than the Rectangle Representation with respect to the grasp success rate. We show results of the Rectangle Representation mostly taken from above test cases, wherein the rectangle represents position and opening width of the gripper. The pictures show the top grasp or the top 10 grasps in a scene. A thick violet line is used to indicates the manipulator closing direction.

- **Opening width**.
  Figure 22 and the following images demonstrate that using the opening parameter may negatively affect the grasping and that an initially fully opened gripper leads to better results for Rectangle Representation. Therefore, in our experiments we skipped the calculation of the gripper opening.
- **Shadow sensitive**.
  Figure 23 shows that shadows can have an impact on the Rectangle Representation. The (S)HAF approach is completely robust to shadows because they have no (topographical) impact on objects.
- **Edge focused**.
  Images from Figure 24 illustrate that grasp learning of Rectangle Representation often relies on the existence of edges in the image. For slim objects such as screwdrivers and pens or objects with thin boarders such as bowls this approach is beneficial, but for bigger objects the center of the grasp may be placed at the border of the object, which is suboptimal. The approach is also sensitive to color changes within the given object (see Figure 24(d)). (S)HAF does not rely on color intensity values.
- **Surface independent**.
  Figure 25 shows examples of detected grasp hypotheses that illustrates that it is hard for the features in Rectangle Representation to take the object surface into account. There is no possibility to grasp an object when the gripper touches the object before any part of the object is in the area such that the manipulator can surround the object when closing. Our approach takes object surfaces and obstacles into account.
- **Height dependent**.
  Figure 26(a) shows an example of the top grasp with the Rectangle Representation of an umbrella at its cord. It makes it clear that the impact of height related features in this approach is not very strong which we consider as a drawback since executing robot grasps at strings on planes is very hard. Our approach learns to prefer grasps at positions with large height differences (for top grasps) between objects and its surroundings.
- **Orientation**.
  Figure 26(b) shows an example where Rectangle Representation delivered a grasp orientation 90° off to the optimal orientation (see the violet line) because of the color crossing of a screwdriver handle. As the (S)HAF approach tries to surround objects with the manipulator as far as possible before closing, an appropriate roll angle for slim rectangular objects is achieved.
- **Perception**.
  In some cases, the white toy bear depicted in Figure 26(c) could not be detected on the white background although the Rectangle Representation algorithms compare the current image with the image of an empty table and the bear possesses contrasting black spots for nose and eyes. In comparison, the (S)HAF approach relies on perceived point cloud data. With new devices developed in recent years high-quality data for a favorable price can be achieved. But there are still limits for 3D perception which suggests the combination of 3D and 2D data to overcome perception problems for shiny or transparent objects (see Section 8.7).
- **Obstacle avoidance and path planing**. In Figure 26(d), the center of the rectangle is not on the glass, but on the pen behind it. For the path planning, the glass is an unsurmountable obstacle for grasping the pen with a predefined vertical approach direction and rotation. The (S)HAF approach avoids grasps blocked by obstacles and tends to pick highest objects first, due to its "integrated path planning".
- **Object-related grasp points**.
  Figure 26(e) shows an example where the 2D image features identify a grasp point (center of rectangle) at a position belonging to the table surface instead of the rim of a bowl. The 2D to 3D mapping would deliver undesirable coordinates, which is especially a problem if for grasp planning a fixed grasp point offset is used. For our method, we cannot remember any grasp center positioned on the table surface. Furthermore, our implementation uses perceived object meshes in simulation to calculate the final grasp position given an approach vector, thereby minimizing collisions. This beneficial schema was also used for testing the Rectangle Representation approach.

8.6. Grasping unknown objects with a Kuka Arm

The goal of this experiment was to demonstrate the applicability of our approach on a Kuka arm as a fourth hardware setup with a manipulator suitable to validate our scalability

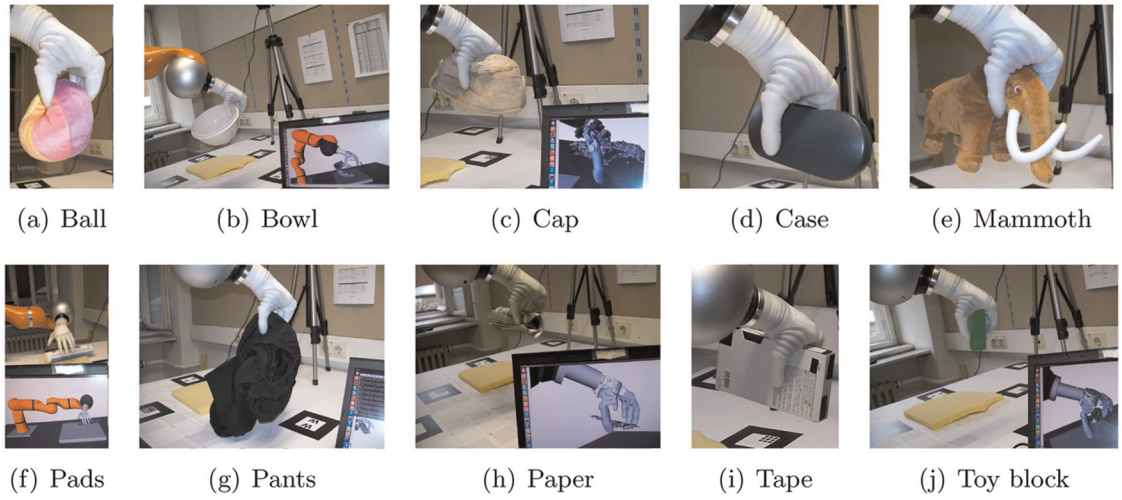


Fig. 28. Test objects for grasping with a Kuka arm and Michelangelo hand during experiments.

claim (see Section 7). Furthermore, we emphasize an error analysis and potential improvements for grasp detection.

8.6.1. Test setup. The experiment was executed with a 7-DOF Kuka LWR arm and the Michelangelo hand from Otto Bock with 2-DOF. We used one Kinect device for data acquisition.

To show that our grasp detection is also usable with more complex manipulators, we perform a test series with 10 objects (depicted in Figure 28) and 10 tries per object. For each try, the object was placed in front of the robot arm with different orientations. Due to safety reasons, we positioned the objects on foam and started the execution of each trajectory by pressing enter as only interaction after starting the grasp processing pipeline. For practical reasons (e.g. to improve robustness against calibration errors or missing point cloud data) we always attempt at approaching the final 7 cm in a straight line retaining a constant orientation of the manipulator. This limits the probability for finding (inverse) kinematic solutions for specific grasps. This is also the main reason why we restricted the grasp approach direction in these experiments to mainly vertical approach rays (apart from small variations to find possible kinematic solutions). In this way, we can execute the top-rated grasps instead of using first possible grasps. The manipulator roll steps were chosen to be 15° and grasp calculation was done for all rotations. In other words, the grasp estimation was not terminated if a good grasp has already exceeded a threshold value, which could enhance time performance significantly depending on the selection of the threshold.

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

8.6.2. Results. We achieved a grasp success rate of 85%. For detailed success rates for each object see Table 14, a more detailed failure analysis is done in Section 8.6.3.

Table 14. Grasp success rate in % and performance time in seconds with Michelangelo hand. For each object 10 trials were executed.

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

8.6.3. Error analysis and potential improvements. Nine out of 15 failed grasps happened when grasping the spectacle case or the small wooden toy block. Furthermore, the spectacle case slipped out of the manipulator in 5 out of 10 trials for successful pre-grasps. The rigid convex shape of the object in combination with relatively strong gripper force and a week friction due to the object surface material made the spectacle case slip out of the hand in each of the failure cases.

For the soft pads, the video tape and the newspaper (with the elastic band) the grasps were sometimes not centered but at the side of the object (see Figure 29). The resulting torque led in some cases to unstable grasps and grasp failures during the delivery of the object (e.g. Figures 29(c)–29(e) where the tape slid out of the hand shortly before the delivery position). Therefore, we investigated the problem of these suboptimal grasps and could detect two sources of error.

In Figure 30 the mesh of the video tape is shown. The camera could only perceive the upper surface of the object. This surface is actually flat in real world, but the mesh

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

Fig. 29. Weak grasps at the side of paper, soft pads and video tape.

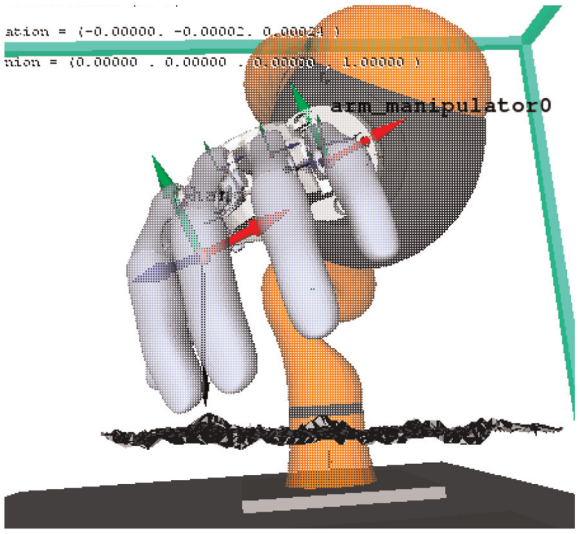


Fig. 30. Grasp position detected at the side of an object (video tape) due to inaccurate point cloud data and extremely sensitive grasp classifier with respect to height differences.

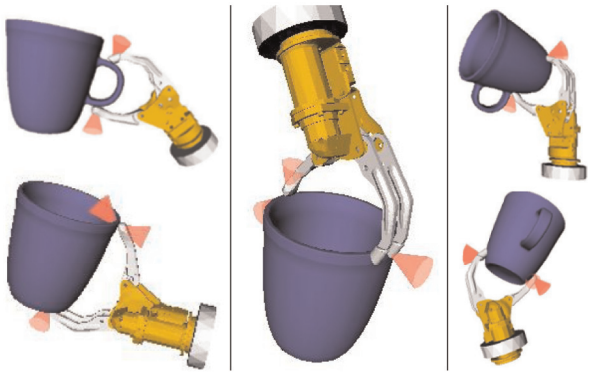


Fig. 31. Complete object models do not guarantee good grasps. The depicted grasps calculated for complete object models and an Otto Bock "SensorHand Speed" fulfill force closure criteria evaluated in OpenRAVE. Still, the grasps are not stable in practice.

generated from the perceived point cloud is rather bumpy. In the depicted case the grasp was chosen above one of the peeks in the surface. This problem is related with the second source of error which becomes also apparent if objects like the soft pads lie on a slope: the grasp classifier does not always accept grasp positions if there are higher (with respect to top grasps) surface points next to the position to classify. Therefore, we get grasps on the sides of objects. Adding training examples focusing on these surface constellations should improve the performance of the classifier and will be part of future work.

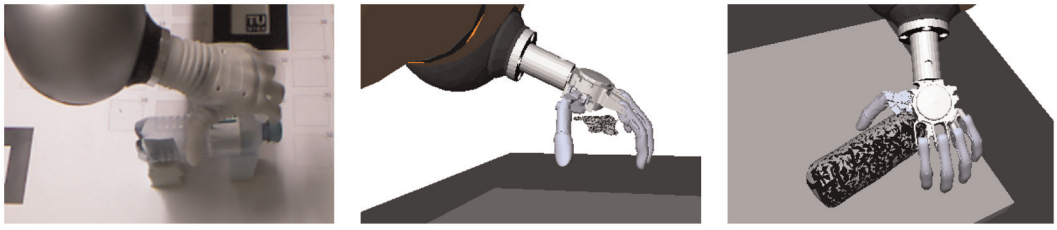
8.7. Grasping known objects with a Kuka Arm

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

State-of-the-art object recognition (e.g. for homogeneously colored, deformable objects such as the trousers in Figure 28(g) and related object segmentation as prerequisite for grasping are quite limiting (e.g. in very cluttered environments) from our experience. On the other hand, object specific grasp force adaptation or task based grasping and manipulation are needed for future mobile service robots that should at one time be able to support people in their daily lives.

In Section 8.5.3, we mentioned the limits of point cloud perception of current sensors for transparent objects. Figure 32 shows a transparent plastic bottle with a label and its perceived point cloud data (i.e. the generated mesh). Insufficient object data leads to grasps in which the manipulator would collide with the object endangering the intactness of the hardware or the object. Using object recognition from Aldoma et al. (2013) implemented with scale-invariant feature transform (SIFT) on 2D data, we can achieve a complete pre-learned mesh of the object. Complete object models are no guarantee for good grasps as Figure 31 indicates, where valid grasps for force closure evaluation are shown.

8.7.1. Test setup. To show the benefit of combining (S)HAF with object recognition in specific cases, we performed an experiment on grasping the bottle 10 times using the (S)HAF method: (1) without object recognition and (2) using an object model perceived by object



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

Fig. 32. Bottle experiments: the left picture shows a bottle seen from the camera, the center picture shows the incomplete mesh/point cloud data received from a Kinect laser sensor and the resulting hand orientation for grasping that would lead to a collision between manipulator and object that cannot be determined by the system. The right picture shows the resulting object mesh if the object is recognized (SIFT) and an object model is used as input for grasping.

Table 15. (S)HAF versus (S)HAF & object recognition: grasp success rate and performance time in seconds. *Restart of laptop.

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

Fig. 33. Object recognition failed in some cases due to light reflections and structural degeneration from wear and tear. After the eighth trial we had to exchange the bottle for better recognition performance.

recognition. The general test setup is identical to that stated in Section 8.6.1 for both conditions.

8.7.2. Results. The results are listed in Table 15: with our conventional approach we could only grasp the bottle in 3 out of 10 tries. Using object recognition and complete models of the bottle, each grasp succeeded, compare Multimedia Extension 3. For the latter, we considered only tries where the object recognition module delivered a model. This was not the case for every try due to reflections from the shiny surface of the bottle (see Figure 33) and abrasion of the bottles surface during grasping. These preliminary experiments could show advantages of combining our method with object recognition. Object recognition not only improves success rates in grasping using our approach, but also results in better quality of grasps.

9. Conclusion

We presented an approach for enabling robots to grasp unknown objects in clutter. Our algorithm does not rely on object models or segmentation and works robustly even in cluttered scenes. The core contribution of this work is a powerful feature type called HAF particularly suited for the task of robot grasping and manipulation in domestic environments. We demonstrated the advantages of our approach compared to other approaches, especially 2D color feature-based ones. We conducted experiments in relation to a state-of-the-art representative and demonstrated the superiority as well as advantages of our topographic surface abstracting features based on numerous examples. The focus of the research presented here was to develop an approach, which is robust to the numerous and dynamically changing task scenarios in the real world and hence is subsequently useful for mobile robotics, being not just based on theoretical simulation. We used "integrated path planning" to guarantee that the selected grasps do not fail due to collisions with other objects and developed a heuristic to ensure robustness of the grasps with respect to positioning errors. We showed the abstraction power and information gain obtained from HAF by comparing it to a classifier trained on point clouds discretized to height grids. The enhancement of the approach with SHAF showed easy extensibility with considerable gain. A thorough analysis of the improved features showed that more complex features lead to better grasp classification results than basic features. We introduced our methods to explore a seven-dimensional grasp space (position, orientation, manipulator gripper width), given our grasp classifier. We implemented tests on three different robot platforms for different tasks such as grasping single objects, emptying a box and clearing a table without manipulator specific classifier training, thereby indicating the hardware scalability of our approach (which also relies on the use of a robot model for final grasp calculation in simulation). Finally, we showed how to combine our approach with object recognition to overcome the problem of incomplete point cloud data and exploit the advantages of our approach if complete object models are available. Videos of the experiments and the code from the

experiments presented in Sections 8.4 and 8.5 have been made available at http://www.youtube.com/user/ijrrmultimedia.

Acknowledgement

We would like to thank Michael Zillich for the fruitful discussions and his feedback on how to improve the article.

Funding

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013; grant agreement numbers 288146 (Hobbit), 215821 (GRASP) and 610532 (SQUIRREL)).

References

Aldoma A, Tombari F, Prankl J, Richtsfeld A, Stefano LD and Vincze M (2013) Multimodal cue integration through hypotheses verification for RGB-D object recognition and 6DOF pose estimation. In: *IEEE international conference on robotics and automation (ICRA)*, pp. 2096–2103.

Balasubramanian R, Xu L, Brook PD, Smith JR and Matsuoka Y (2012) Physical human interactive guidance: identifying grasping principles from human-planned grasps. *IEEE Transactions on Robotics* 28(4): 899–910.

Bicchi A and Kumar V (2000) Robotic grasping and contact: a review. In: *IEEE international conference on robotics and automation (ICRA)*, vol. 10. Sandia National Laboratories/IEEE, pp. 348–353.

Bohg J, Johnson-Roberson M, Leon B, et al. (2011) Mind the gap - robotic grasping under incomplete observation. In: *IEEE international conference on robotics and automation (ICRA)*, pp. 686–693.

Bohg J, Morales A, Asfour T and Kragic D (2014) Data-driven grasp synthesis - a survey. *IEEE Transactions on Robotics* (accepted).

Chang Cc and Lin Cj (2011) LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3): 27:1–27:27.

Chen Yw and Lin Cj (2006) *Combining SVMs with Various Feature Selection Strategies* (*Studies in Fuzziness and Soft Computing*, vol. 324). New York: Springer.

Crow FC (1984) Summed-area tables for texture mapping. In: *Proceedings of SIGGRAPH* 18(3): 207–212.

Diankov R (2010) *Automated Construction of Robotic Manipulation Programs*. PhD Thesis, Carnegie Mellon University. Available at: http://www.programmingvision.com/rosen_diankov_thesis.pdf.

Diankov R (2012) OpenRAVE Documentation. Available at: http://openrave.org/docs/latest_stable/openravepy/databases.grasping/.

Diankov R and Kuffner J (2008) *OpenRAVE: A Planning Architecture for Autonomous Robotics*. Technical Report CMU-RI-TR-08-34, Robotics Institute, Carnegie Mellon University.

Ferrari C and Canny J (1992) Planning optimal grasps. In: *Proceedings 1992 IEEE international conference on robotics and automation*, pp. 2290–2295.

Fischinger D, Einramhof P, Papoutsakis K, et al. (2014) Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems*. DOI:10.1016/j.robot.2014.09.029.

Fischinger D, Jiang Y and Vincze M (2013) Learning grasps for unknown objects in cluttered scenes. In: *IEEE international conference on robotics and automation (ICRA)*, pp. 609–616.

Fischinger D and Vincze M (2012a) Empty the basket - a shape based learning approach for grasping piles of unknown objects. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 2051–2057.

Fischinger D and Vincze M (2012b) Shape based learning for grasping novel objects in cluttered scenes. In: *10th IFAC symposium on robot control (SYROCO)*, pp. 787–792.

Goldfeder C, Allen PK, Lackner C and Pelossof R (2007) Grasp planning via decomposition trees. In: *Proceedings 2007 IEEE international conference on robotics and automation*, April. IEEE, pp. 4679–4684.

Goldfeder C, Ciocarlie M, Dang H and Allen PK (2009) The Columbia grasp database. In: *Proceedings of the IEEE international conference on robotics and automation*. IEEE, pp. 1710–1716.

Herzog A, Pastor P, Kalakrishnan M, Righetti L, Asfour T and Schaal S (2012) Template-based learning of grasp selection. In: *IEEE international conference on robotics and automation*, pp. 2379–2384.

Hu Y, Eagleson R and Goodale MA (1999) Human visual servoing for reaching and grasping: the role of 3-D geometric features. In: *Proceedings of the 1999 IEEE international conference on robotics and automation (ICRA99)*, vol. 4, pp. 3209–3216.

Huebner K and Kragic D (2008) Selection of robot pre-grasps using box-based shape approximation. In: *International conference on intelligent robots and systems*. IEEE, pp. 1765–1770.

Huebner K, Welke K, Przybylski M, et al. (2009) Grasping known objects with humanoid robots: A box-based approach. In: *International conference on advanced robotics*. IEEE, pp. 1–6.

Jiang Y, Moseson S and Saxena A (2011) Efficient grasping from RGBD images: learning using a new Rectangle Representation. In: *Proceedings of the IEEE international conference on robotics and automation*. IEEE, pp. 3304–3311.

Katz D, Venkatraman A, Kazemi M, Bagnell JA and Stent A (2013) Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. In: *Robotics: science and systems conference*.

Kenney J, Buckley T and Brock O (2009) Interactive segmentation for manipulation in unstructured environments. In: *IEEE international conference on robotics and automation*.

Klingbeil E, Rao D, Carpenter B, Ganapathi V, Ng AY and Khatib O (2011) Grasping with application to an autonomous checkout robot. In: *Proceedings of the IEEE international conference on robotics and automation*. IEEE, pp. 2837–2844.

Kootstra G, Popovic M, Jorgensen JA, et al. Enabling grasping of unknown objects through a synergistic use of edge and surface information. *The International Journal of Robotics Research* 31(10): 1190–1213.

Le QV, Kamm D, Kara AF and Ng AY (2010) Learning to grasp objects with multiple contact points. In: *IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 5062–5069.

Li Y and Pollard NS (2005) A shape matching algorithm for synthesizing humanlike enveloping grasps. In: *5th IEEE/RAS international conference on humanoid robots*. IEEE, pp. 442–449.

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

Mason MT and Salisbury JKJr (1985) *Robot Hands and the Mechanics of Manipulation*. Cambridge, MA: The MIT Press.

Miller AT and Allen PK (2004) Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Automation Magazine* 11(4): 110–122.

Miller AT, Knoop S, Christensen HI and Allen PK (2003) Automatic grasp planning using shape primitives. In: *Proceedings of the IEEE international conference on robotics and automation*, volume 2. IEEE, pp. 1824–1829.

Morales A, Asfour T, Azad P, Knoop S and Dillmann R (2006) Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, pp. 5663–5668.

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

Pollard NS (2004) Closure and quality equivalence for efficient synthesis of grasps from examples. *The International Journal of Robotics Research* 23(6): 595–613.

Przybylski M, Asfour T and Dillmann R (2011) Planning grasps for robotic hands using a novel object representation based on the medial axis transform. In: *IEEE/RSJ international conference on intelligent robots and systems*, Institute for Anthropomatics, Karlsruhe Institute of Technology, Germany. IEEE, pp. 1781–1788.

Rao D, Le QV, Phoka T, Quigley M, Sudsang A and Ng AY (2010) Grasping novel objects with depth segmentation. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 2578–2585.

Saxena A, Driemeyer J and Ng AY (2008a) Robotic grasping of novel objects using vision. *The International Journal of Robotics Research* 27(2): 157–173.

Saxena A, Wong LLS and Ng AY (2008b) Learning grasp strategies with partial shape information. In: *Proceedings of the AAAI*, vol. 3. AAAI Press, pp. 1491–1494.

Varadarajan KM and Vincze M (2011) Object part segmentation and classification in range images for grasping. In: *International conference on advanced robotics (ICAR 2011)*, pp. 21–27.

Viola P and Jones MJ (2004) Robust real-time face detection. *International Journal of Computer Vision* 57(2): 137–154.

Weisz J and Allen PK (2012) Pose error robust grasping from contact wrench space metrics. In: *2012 IEEE international conference on robotics and automation*. IEEE, pp. 557–562.

Wohlkinger W and Vincze M (2010) 3D object classification for mobile robots in home-environments using web-data. In: *International workshop on robotics in AlpeAdriaDanube Region (RAAD)*. IEEE, pp. 247–252.

Appendix: Index to Multimedia Extensions

Archives of IJRR multimedia extensions published prior to 2014 can be found at http://www.ijrr.org, after 2014 all videos are available on the IJRR YouTube channel at http://www.youtube.com/user/ijrrmultimedia

Table of Multimedia Extensions

Extension	Media type	Description
1	Video	Clearing the table. The video shows examples of clearing the table using (S)HAF with a Schunk robot arm (test cases 6, 8, 10 from Section 8.2) and a PR2 (compare Section 8.5)
2	Video	Emptying the box. The Video shows examples of emptying a box with HAF executed on a Schunk robot with the hand prosthesis "SensorHand Speed" (Section 8.3)
3	Video	Grasping known objects. The video shows how the presented approach can grasp objects hardly visible with depth sensors using object recognition and a model of the object (Section 8.7)
4	Code	Features, Grasp-Classifier and Framework as ROS package: http://wiki.ros.org/haf_grasping