2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Learning Grasps for Unknown Objects in Cluttered Scenes

David Fischinger and Markus Vincze
Automation an Control Institute
Vienna University of Technology
Vienna, Austria
Email: {df, vm}@acin.tuwien.ac.at

Yun Jiang
Department of Computer Science
Cornell University
Ithaca, New York, USA
yunjiang@cs.cornell.edu

Abstract—**In this paper, we propose a method for grasping unknown objects from piles or cluttered scenes, given a point cloud from a single depth camera. We introduce a shape-based method – Symmetry Height Accumulated Features (SHAF) – that reduces the scene description complexity such that the use of machine learning techniques becomes feasible. We describe the basic Height Accumulated Features and the Symmetry Features and investigate their quality using an F-score metric. We discuss the gain from Symmetry Features for grasp classification and demonstrate the expressive power of Height Accumulated Features by comparing it to a simple height based learning method. In robotic experiments of grasping single objects, we test 10 novel objects in 150 trials and show significant improvement of 34% over a state-of-the-art method, achieving a success rate of 92%. An improvement of 29% over the competitive method was achieved for a task of clearing a table with 5 to 10 objects and overall 90 trials. Furthermore we show that our approach is easily adaptable for different manipulators by running our experiments on a second platform.**

I. Introduction

Robotic grasping is a well examined research area, but it is by far not solved yet. A large part of related work deals with object recognition and object categorization [1] and then apply known grasps for this type of object. Using known models is one possibility to overcome the issue of incomplete data perceived from single views of a robot. Availability of object models enables use of force and form closure grasp quality metrics ([2], [3], [4]). Even pretended that with sufficiently large model databases and still working recognition algorithms grasps for unknown objects could be interpolated, object recognition approaches would fail in cluttered scenes. State of the art segmentation and recognition methods currently fail to deliver reliable results for scenarios like the one depicted in Fig 1. Another type of approach tries to identify simple geometric forms to enable grasp point detection ([5], [6]), but also requires a proper segmentation to work. A third approach that was popular during the last years is based on learning grasps using features from 2D images ([7], [8], [9], [10]).

In this paper we present our method to abstract shape information from perceived surface of unknown objects, hence enabling to learn how to grasps novel objects in cluttered scenes. Key advantages of our method are the avoidance of segmentation, an efficient way to calculate the introduced Symmetry Height Accumulated Features (SHAF) using Summed Area Tables, an 'integrated path planning' which rules out grasps where no approaching path would be possible and a weighting system to enhance the robustness of the selected grasps.

Fig. 1. PR2 used for experiments grasps from cluttered scene

In the next section we discuss related work. Section III describes the idea of Height Accumulated Features, gives a deeper insight on most efficient features and introduces Symmetry Features. Section IV describes the process how calculated feature values are used to select the best six dimensional grasp configuration. We describe the machine learning methods used, a weighting method to enhance the robustness of grasp hypotheses, our way to explore the entire grasp space using the trained classifier and the fine calculation of grasp points given our grasp representation using a simulation environment. In Section V we show the additional value of Height Accumulated Features by comparing them to learning grasps on height values. We discuss the effect of the introduced Symmetry Features. As practical part of the evaluation section,

978-1-4673-5643-5/13/$31.00 ©2013 IEEE

we present extensive tests of our method compared to a state-of-the-art algorithm, executed on a PR2 robot.

II. Related Work

The problem of grasping unknown objects has been tackled in several works. Miller et al. [5] used shape primitives like spheres, cones and boxes to approximate object shape and used the simulation environment GraspIt [11] for grasp stability tests. Varadarajan and Vincze [6] proposed a similar approach with the more general Superquadrics as basic geometric form. Wohlkinger et al. [1] used object recognition and categorization in connection with a large model database [12] to obtain object models for later use in simulation to calculate force closure grasps. Saxena et al. ([7], [8]) proposed supervised learning with local patch-based image and depth features for grasping novel objects in cluttered environments. Jiang et al. [9] improved this work and added the capability to learn optimal gripper opening width. The focus of learned features is on 2D images, but one feature is based on a comparison of object heights in predefined rectangle regions (in further we refer to this method as “Rectangle Representation”) and similar to Height Accumulated Features used in this work. This related feature, the popularity and performance 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. [10] extended method from [8] to accommodate grasps with multiple contacts and achieved a success rate of 80% for desk clearing experiments with 2 to 8 objects counting success/failure of the first grasp attempt per object. Closest to our work is [13]. Klingbeil et al. proposed an approach for grasp detection for a two finger gripper to autonomously grasp unknown objects based on raw depth data. Their approach is based on finding a good end-effector grasp position by maximizing the contact area between the robots gripper and the perceived point cloud. This approach does not require either object models or a learning phase. Furthermore, they have integrated their grasping methods to create an autonomous checkout robot by reading barcodes. Close to our work is also the recent approach of Herzog et al. [14]. They proposed a template-based grasp selection algorithm operating on heightmaps which uses demonstrated grasp configurations and generalizes them to grasps for novel objects.

III. Height Accumulated Features

In this section we give the idea of learning grasps using Height Accumulated Features (HAF). The basic idea of HAF was first published in [15] and used to grasp unknown objects from inside a box. For reasons of readability we repeat the essence here. Furthermore we introduce an improvement of the method to solve specific problem cases of the old classifier using Symmetry Features. In addition we analyze feature performance and show the top 10 features using F-score evaluation.

A. Motivation of HAF

Objects are often perceived as point clouds with tens of thousands points. Applying machine learning directly on the raw data is impractical. Not only due to the large space of points, even more critical is the 6 dimensional grasp output space (3 parameters for manipulator position, 3 parameters for manipulator orientation) where all 6 parameters are strongly related. In Section IV-C we describe our method to explore the whole grasp space using a trained grasp classifier. We state the idea and intuitive interpretation of HAF, which reduce the complexity of point cloud input, increase the value of given information as shown in Section V-A and hence enabling the use of machine learning for grasp detection of unknown objects in cluttered and non cluttered scenes.

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

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

with

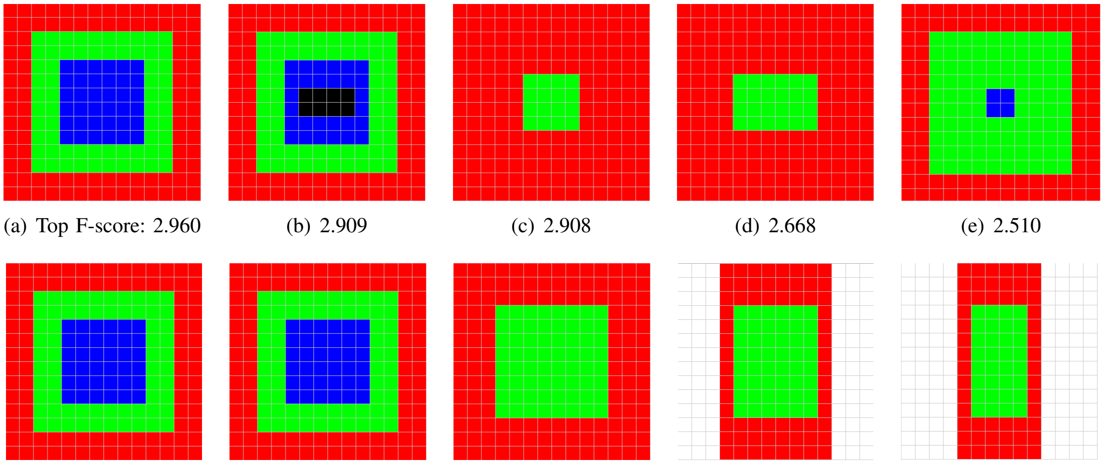
$$r_{i,j} = \sum_{k,l \in \mathbb{N} : (k,l) \in R_{i,j}} H(k,l) \tag{2}$$

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

The HAF vector **f** is the sequence of HAF values:

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

For 267 out of 302 used features for the classifier in Section V-C two overlapping regions were used where one region was completely inside the other one and a weighting factor $w_{i,j}$



(a) Top F-score: 2.960 (b) 2.909 (c) 2.908 (d) 2.668 (e) 2.510

(f) 2.233 (g) 2.233 (h) 2.220 (i) 2.148 (j) 2.124

Fig. 2. Top 10 HAF-Features with F-score values (from Table I)

such that the feature value is zero if both regions have the same average height, bigger than zero if the region inside is higher, and smaller than zero if the smaller region has in average a lower height. 35,000 Features were automatically generated with these constraints and added to 500 manually generated features where this intuitive interpretation is not applicable. Using F-score evaluation [16] we selected the top 302 features out of these 35,500 weighing up time constraints against detection outcome. The HAF top 10 list with respective F-score values are shown in Figure 2. From the top ten features (F-score value) only 2 features (ranked 9 and 10) are from the 35,000 features with the intuitive interpretation ('if the center is higher, it is good to grasp there'). From the top seven features, four features have three regions and one 4 regions. Table I summarizes the top 10 features with the weights of each region. Each region has a rectangular form, hence also including overlapping regions indicated by other colors. Region size can be observed from Fig. 2.

TABLE I
TOP TEN FEATURES RANKED BY F-SCORE VALUE

Rank	F-score	#Reg.	w_{red}	w_{green}	w_{blue}	w_{black}
1	2.960	3	1	-1.0	-2	-
2	2.909	4	1	0.5	-5	-8.25
3	2.908	2	1	-10	-	-
4	2.668	2	1	-7	-	-
5	2.510	3	1	-1	-10	-
6	2.233	3	1	-1	-1	-
7	2.233	3	1	-1	-1	-
8	2.220	2	1	-2	-	-
9	2.148	2	-1	2.33	-	-
10	2.124	2	-1	2.62	-	-

B. Symmetry Height Accumulated Features

The basic form of learning grasps with HAF has one drawback. The manipulator in Fig. 3 shows where HAF detected a grasp: at the edge of an upside down box. Since 90 percent of the features in earlier work [15] are based on the average heights of two nested regions, 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 $2x$. The feature value is no indication if two fingers of the gripper 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 edges of an upside down box. For clearing a table with objects directly on the table this behavior was tolerated if not appreciated: due to the weighting system in Section IV-B single misclassifications of grasp position did not influence the chosen grasp point, since as long as there was an easily graspable object on the table, this object was grasped before trying to grasp, say at a boarder of a big flat lying book. On the other hand, if there would only be a flat lying book at a table, we preferred to get bad grasps to no grasps for having at least the chance to try grasping. Execution of bad grasps can lead to perturbed position and orientation of objects in a hard scene, hence making grasps in later tries possible or easier.

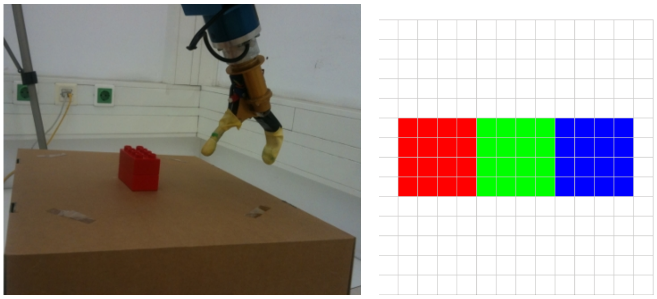


Fig. 3. SHAF-Motivation: HAF detects a bad grasp at the edge of a box, right: shows regions for one SHAF example

It turned out that for special constellations like a small object on top of an upside down positioned box on a table (Fig. 3) even with our weighting system sometimes the edge of the box was picked as grasping destination instead of the object on top. Hence we decided to extend HAF by an additional feature type: Symmetry Features. Symmetry Features have three disjunctive regions of equal size as depicted in Fig. 3 (right) where r_r,r_g,r_b are the accumulated heights on the region grid. The feature value f is defined as follows:

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

So we assign the minimal distance of accumulated heights between center region and side regions if the center region is in average the highest, and -1 otherwise. Note that this function is either positive or -1. In V-B we discuss the impact of Symmetry Features for our grasp classifier.

IV. From Grasp Classifier to Actual Grasps

This section describes the process of generating actual grasps given HAF. The process description is stated in [15] in more detail and summarized here for reasons of readability. First we describe the learning process for grasping (Section IV-A). Then we explain our weighting method to achieve more robust grasps (IV-B). After showing our method for exploring the whole grasping space (IV-C), we show how the fine calculation of grasping points and path planning is done for simulation (IV-D).

A. SVM Learning

For training purposes we gathered 700 point clouds of a variety of objects on a table with z-axis perpendicular to the table and origin at the table surface. For supervised learning we labeled 450 times the most promising grasp of the scene. To be more specific: we labeled a x,y position such that an Otto Bock hand prosthesis positioned above the objects (with tool center point at x,y) and oriented in a way that the line between the tip of the thump and the tip of the forefinger is aligned with the x-axis, would get a stable grasp of an object after approaching the object (approach vector of the manipulator parallel to z-axis) up to 1cm and closing the fingers afterwards. 250 times scenes were labeled at positions with hardly any chance of grasp success. Using methods like scaling, mirroring and inverting we generated overall 8300 positive and 12,800 negative examples. For all examples Symmetry Height Accumulated Features (SHAF) were calculated and a SVM classifier with radial basis function kernel in the implementation of [17] was trained.

B. Weighting System

For a real life scene the trained grasp classifier normally does not return an isolated grasp position, but a bunch of grasp points in a region (see green area in Fig 4). Generally 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 good grasp position is evaluated by

$$v(r,c) = \sum_{x,y \in \mathbb{N}} I_{grasp}(x,y) \cdot w_{r,c}(x,y) \qquad (4)$$

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

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

The following table gives the weighting factors for a grasp hypothesis GH.

TABLE II
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		

This simple heuristic enhances the robustness and stability of grasp in practice immensely.

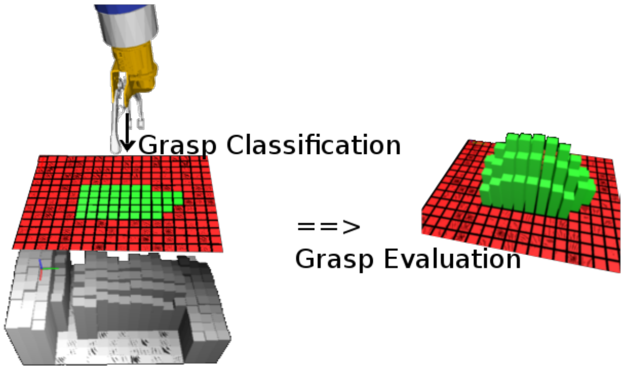


Fig. 4. Example of grasp classification (left: green color indicates possible grasp positions) and weighting of outcome (right: height of green bars indicates grasp quality evaluation)

C. Exploration of Grasp Space

Using the SVM grasp classifier and our weighting system we receive the best grasp point for a given hand orientation and a top grasp. To explore the complete grasp space we successively rotate the point cloud, detect the best grasp on the newly calculated height grid and transform the coordninates back to the original coordinate frame.

D. Fine calculation of Grasps

Calculation of optimal distance between manipulator and object before closing manipulator, calculation of touching points between manipulator and object as well as path planning are done using a robot model in the simulation environment OpenRAVE. For a more comprehensive description of our learning system, weighting heuristic, grasp space exploration and fine calculation see [15].

V. Experiments and Evaluation

Evaluations are divided into two main parts. The first part evaluates the grasp classifier. Here we discuss the information gain obtained by HAF by comparing them with direct learning on height grids. In addition we analyze the impact of Symmetry Features for the grasp classifier. In the second part the whole system is tested on a real robot. In contrast to the first part we use the system to get the best grasp, instead of testing if the center of a scene is a promising grasp. We use our algorithm and a state-of-the-art method on a PR2 platform and test grasping for single objects as well as grasping in cluttered scenes with 5 to 10 objects.

A. HAF vs. Heights

In [15] we claimed the additional information value of HAF. To prove this statement, we compare our SVM classifier trained with HAF against a SVM classifier that was trained using simple heights after discretization. We used 14x14cm grids for training, hence we had 196 Height Features. For training purposes we used clearly distinctive scenarios, i.e. positive training examples presented easy grasp situations, and negative examples reflected situations where grasp execution seemed impossible. Since the grasp classifier showed an accuracy of more than 99 percent (compare to [15]) for simple scenes, we gathered test scenes which are harder to classify, to provide a dataset which enables a meaningful comparison. E.g. a positive grasp example was not exactly centered 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 overall 3928 test cases. Results are shown in Table III.

TABLE III
GRASP CLASSIFICATION ACCURACY: HAF VS. GRID HEIGHTS

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

Note that all HAF values are completely calculated out of the 196 height values, still the data processing results in a 21.69% improvement of classification accuracy on a hard test data set.

B. HAF vs. SHAF

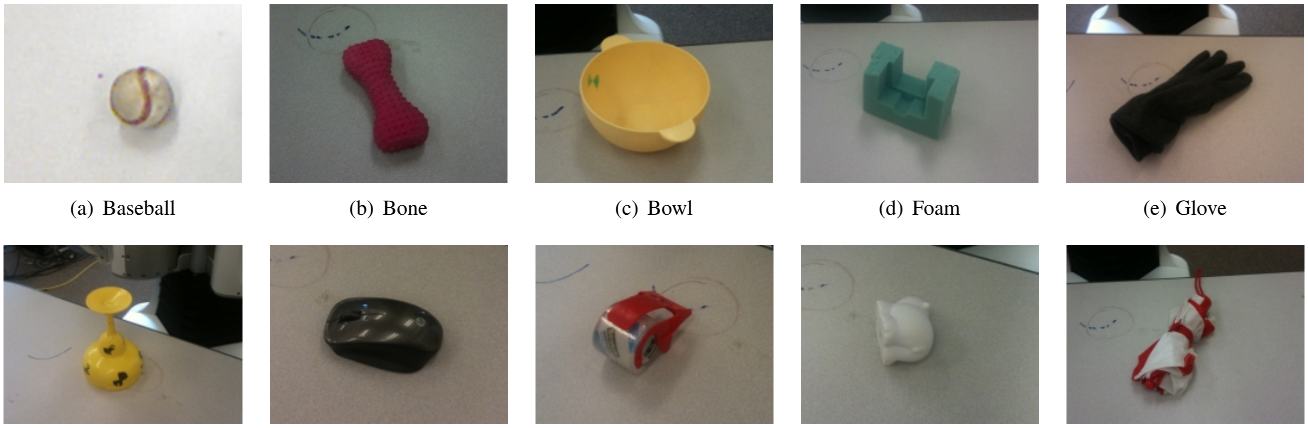
To examine the impact of Symmetry Features we tested our SHAF-Classifier (HAF plus Symmetry Features) on the same test set. An accuracy rate of 74.31% resulted. The main reason for this disappointing outcome was the difference of scenarios in test and training examples. For the test data set we added many test cases like the one motivating us to add symmetry features (e.g. upside down box). The training examples were not changed and did not include similar scenes. After adding training examples with similar scenes (one sided gaps), we achieved an accuracy rate of 85.50%. Instead of adding further training examples to beat the accuracy rate of the default HAF classifier, we analyzed the outcome (since missing training data is no sufficient explanation why HAF had a higher accuracy rate then SHAF). Our analysis exposed that SHAF classified negative examples with success rates higher 90% but performed badly regarding positive examples. Deeper analysis showed that SHAF is more sensitive on grasp situations. A rim of a bowl with 2 to 3 centimeters offset with respect to a manipulator center results easily in negative classification, although few centimeters offset for a bowl would still be sufficient for grasping. Since getting at least one grasp point was never a problem in earlier experiments, we see tougher constraints for selecting positive grasps not as a drawback. So the detected grasps are even more reliable.

C. Robotic Experiments

Because we believe that the most significant way of testing grasp algorithms is to test it on a physical robot, we perform two sets of experiments. The first series tests grasp success rates for single standing objects for 3 different grasp algorithms. Test series two is about grasping in cluttered scenes with 5 to 10 objects.

1) Test Setup: Training and initial tests for our SVM classifier was done for an Otto Bock hand prosthesis and a Schunk/Amtec 7 DOF arm. For test purposes 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 2 finger manipulator. We didn't adapt the grasp classifier for the PR2 gripper, thereby showing that our method works without changes for different manipulators. Even for numerous manipulators with bigger deviation little code changes (such as scaling of the relevant point cloud as preprocessing step) enable usage of our classifier without retraining from scratch. The Robot Operating System ROS (www.ros.org) was used as basic framework and for module communication. Point cloud perception of scenes was done using a Microsoft Kinect camera mounted at the head of the PR2 as shown in Fig. 1. For point cloud processing the Point Cloud Library PCL (www.pointclouds.org) was used. After perception of a point cloud, points of the table surface were deleted automatically as well as points not relevant for grasping, e.g. outside kinematic reachability of the used left PR2 arm. From the remaining point cloud a mesh was generated which is then used in OpenRAVE for path planning and grasp simulation. The Rectangle Represantion method from [9] and SHAF grasping use the same function for grasp simulation in OpenRAVE from were the physical robot is also controlled, so the system is not aware of from which method the grasp hypothesis is coming from.

2) Single standing objects: The first test series is about grasping single objects on a table. Three methods for calculating grasps (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 used method can be found at the OpenRAVE documentation about the grasping module ([18]). We clearly want to state that



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

(f) Martini Glass (g) Computer Mouse (h) Splicing Tape (i) Toy Bear (j) Umbrella

Fig. 5. Objects used for test scenarios

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).

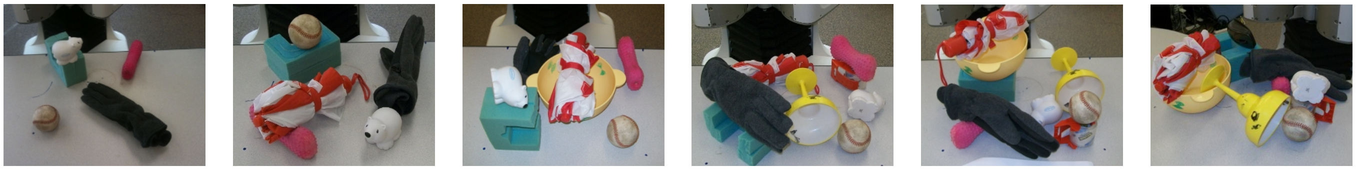
Grasp method two is the Rectangle Representation from [9]. Method three is our method based on SHAF. For the first test series 10 objects were chosen. Nine from an object box at Cornell University, with no object ever involved in training of the SHAF classifier or any other part of our system. To pick the tenth object we asked a not involved person to pick any object from the lab that fits between the PR2 gripper, which resulted in picking a slippery computer mouse. All objects are depicted in Fig. 5 in one of the grasp poses used for this test. For test method 2 and 3 we used top grasps (with vertical approach direction) only, due to 3 reasons: First, for a given manipulator orientation and a straight approach trajectory at the last centimeters to an object, it is already hard to find an area of 35x35cm for top grasps where inverse kinematic solutions are possible. Each allowed deviation from vertical grasps reduce the size of this object placing region where grasps can be executed. Second, for test method two no code was available to calculate grasp hypothesis other than from the direction of the camera view. And third, in our test scenario in contrast to earlier work [15] point cloud perception is done by a single camera. Due to incomplete point cloud data (especially occlusions), path planning gets more unreliable the more 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 35x40 cm region where inverse kinematic solutions for vertical grasps were generally found. Lacking of uninvolved human resources on the first day of test runs, we did this work ourselves for the first five test objects, for the last five objects we asked non involved persons to do so. 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. Results are summarized in Table IV. The listed average time in seconds for OpenRAVE algorithm is the time for grasp calculation. For algorithms 2 and 3 the time is measured from receiving the point cloud data (and image data for method 2) to the output of the grasp hypothesis. Grasp and path planing time is for the later 2 methods about one second. Being aware that our algorithm will still be superior regarding time performance, we choose a very high quality threshold parameter for our algorithm. This quality threshold stops our algorithm as soon as a grasp solution evaluation is better than the threshold (so other rolls are not tested anymore). Although with smaller threshold values good results are achieved, we decided to go for more grasp quality than faster performance.

TABLE IV
GRASP SUCCESS RATE (ACC.) IN % AND PERFORMANCE TIME IN SECONDS: OPENRAVE VS. RECTANGLE REPRESENTATION VS. SHAF

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

SHAF succeded in 46 out of 50 trials, giving a success rate of 92% compared to 58% for Rectangle Representation and 20% for the OpenRAVE heuristic. 8 out of 10 objects were grasped 5 times without a single failure. The Rectangle Representation only achieved a higher success rate for the mar-



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

tini glass. For this object the SHAF grasps were not optimal, and when approaching with the manipulator slight touching of the object leaded to rolling away of the lying Martini Glass. The Rectangle Representation often detected grasps at object edges, which was a main reason for the 34% gap in grasp success rate. Although we invested significant time finding optimal parameters for the OpenRAVE method, the algorithm did not grasp more than 1 out of 5 objects successfully. 12 out of 50 grasp tries, this method could not find a force closure grasp. For additional 5 tries, path planning failed for all found grasp solutions (although we restricted the grasps to manly vertical directions). Force closure detection in simulation for two finger grippers does not always return promising solutions even if complete object models are available. Because of this, we decided to skip the OpenRAVE algorithm for the next test scenario due to the bad performance without available object models and also due to calculation times which dramatically increases with the size of the object mesh (which is way bigger for the clearing the table scenario).

3) Clearing Table: Test two is about clearing the table for 5 to 10 objects. In this case after each failed grasp the object with the center nearest to the tool center point of the gripper at time of closing is removed such that each algorithm has only one try for one object. Each of the two methods led to one grasp where two objects where removed simultaneously. In these two cases, the object with object center further away from the tool center point of the manipulator was replaced at its original position and the grasp for the other object was assessed as successful. After each grasp the initial position of the remaining objects was reestablished using photographs from different angles of the initial scene. Table V shows the success rates for the six test cases and the average calculation time per grasp in seconds. Pictures of all test cases are shown in Fig. 6.

TABLE V
GRASP SUCCESS RATE (ACC.) IN % AND PERFORMANCE TIME IN SECONDS: RECTANGLE REPRESENTATION VS. SHAF

Method:		**Rec. Repr.**		**SHAF**	
Test Case	**#Obj**	**Acc.**	**Time**	**Acc.**	**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**

SHAF successfully grasped 39 objects out of 45 tries, giving an overall success rate of 86.7%. Rectangle Representation achieved an overall success rate of 57.8% (26 out of 45). Although, as mentioned above, our parameter setting was suboptimal for time performance, our algorithm was three times as fast as the Rectangle Representation algorithm. The 6 grasp failures for our approach had different reasons. In test case 5, we failed to grasp the glove because this single time 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 5 failures non optimal SHAF grasps were each time partly responsible. But in each case other factors contributed to the failure: 3 times the object (toy bear in TC 3, pink bone in TC 4, Martini Glass in TC 5) was touched and moved by the manipulator out of the initial position. Premature touching had two reasons. One is incomplete data. For the Martini Glass and the bowl the inside region was generally badly perceived as shown on an example of the bowl in Fig. 7. Hence path planning calculated paths where the manipulator collided with the (in simulation not existing) inside of the bowl or Martini Glass when grasping at the rim.

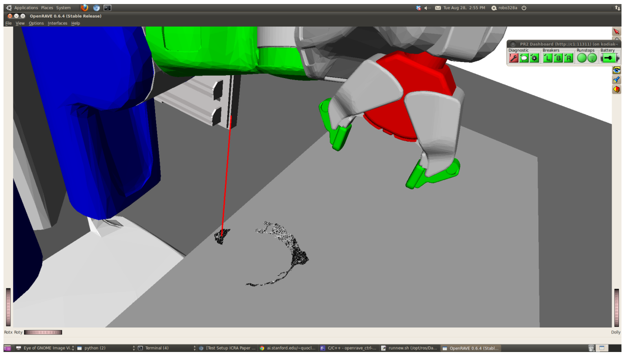


Fig. 7. Incomplete perceived point cloud data (depicted the result of a plastic bowl from a camera mounted at the robots head) is one reason for failed grasps

VI. Conclusion

Beyond the advantages of HAF already stated in [15] like 'integrated path planning', no need of segmentation, quality boost through grasp evaluation system and an efficient feature calculation process using Summed Area Tables, we verified that one camera delivers sufficient input for our method to achieve an excellent grasp performance as stated but not proven in earlier work. We implemented and executed tests to prove the generated information gain of HAF compared to grasp learning on height grids (Section V-A). The easy extensibility of our system was shown by extending HAF with Symmetry Features. The introduced Symmetry Features were improving the classification of grasps for a specific constellation where earlier work had significant problems. At the same time Symmetry Features increase the requirements for accepted grasps, enhancing the reliability of the final outcome. After experiments on a Schunk 7-DOF arm with hand prosthesis we performed our practical test for this work on a PR2 platform without any change for our grasp classifier, thereby indicating the hardware scalability of our approach. Most importantly, we showed the superiority of our approach compared to a state-of-the-art method in several test cases, achieving success rates about 30% higher than the benchmark algorithm. Videos of all test cases of the clearing the table scenario for SHAF can be found at www.youtube.com/user/clearingthetable/. In addition one video shows clearing the table with 10 objects in a harder constellation then all test cases which was not part of the test trials without a single grasp failure. Another main contribution is the availability of the executed demo code (whole framework, not only core algorithms) for grasping unknown objects with a PR2 (Rectangle Representation and SHAF) as ROS stack. This contribution enables a serious comparison of other grasp detection algorithms with the two methods used in this paper. The code is available at: http://pr.cs.cornell.edu/grasping/rect_data/data.php

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 288146, Hobbit.

References

[1] W. Wohlkinger and M. Vincze, "3D Object Classification for Mobile Robots in Home-Environments Using Web-Data," *International Workshop on Robotics in AlpeAdriaDanube Region RAAD*, pp. 247–252, 2010.

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

[3] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 32–44, 1988.

[4] N. S. Pollard, "Closure and Quality Equivalence for Efficient Synthesis of Grasps from Examples," *The International Journal of Robotics Research*, vol. 23, no. 6, pp. 595–613, 2004.

[5] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," *ICRA*, vol. 2, pp. 1824–1829, 2003.

[6] K. M. Varadarajan and M. Vincze, "Object Part Segmentation and Classification in Range Images for Grasping," in *International Conference On Advanced Robotics ICAR 2011*, 2011, pp. 21–27.

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

[8] A. Saxena, L. L. S. Wong, and A. Y. Ng, "Learning Grasp Strategies with Partial Shape Information," *AAAI*, vol. 3, no. 2, pp. 1491–1494, 2008.

[9] Y. Jiang, S. Moseson, and A. Saxena, "Efficient Grasping from RGBD Images : Learning using a new Rectangle Representation," *ICRA*, pp. 3304–3311, 2011.

[10] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," pp. 5062–5069, 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5509508

[11] B. Y. A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[12] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3DNet: Large-Scale Object Class Recognition from CAD Models," *ICRA*, pp. 5384–5391, 2012.

[13] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib, "Grasping with Application to an Autonomous Checkout Robot," *ICRA*, pp. 2837–2844, 2011.

[14] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal, "Template-based learning of grasp selection," *2012 IEEE International Conference on Robotics and Automation*, pp. 2379–2384, May 2012.

[15] D. Fischinger and M. Vincze, "Empty the Basket - A Shape Based Learning Approach for Grasping Piles of Unknown Objects," *IROS*, 2012.

[16] Y.-w. Chen and C.-j. Lin, "Combining SVMs with Various Feature Selection Strategies," *Strategies*, vol. 324, no. 1, pp. 1–10, 2006.

[17] C.-c. Chang and C.-j. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1—-27:27, 2011.

[18] R. Diankov, "OpenRAVE Documentation," 2012. [Online]. Available: http://openrave.org/docs/latest_stable/openravepy/databases.grasping/