Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D28
Deliverable Title :	Perception/context model providing cues of any object in relation to affordances, action and task context
Type (Internal, Restricted, Public):	PU
Authors	M. Vincze, W. Wohlkinger, A. Aldoma, K. Varadarajan, E. Potapova, D. Fischinger, J. Prankl, M. Zillich; M. Pronobis, D. Kragic;
Contributing Partners	TUW, KTH

Contractual Date of Delivery to the EC:	28-02-2012
Actual Date of Delivery to the EC:	Draft 1-02-2012

Contents

Chapter 1

Executive Summary

This report presents the work of year four in WP4. WP4 is concerned with modelling perceipts and contextual information of objects in relation to affordances, actions and task ocntext. With grasp context we refer to the information relevant to the grasp, which at its core includes the grasp points on the objects but also the relationship to the complete object, the hand, the task, and the attention on the target object. The overall objective is to perceive grasping points on unknown objects by the end of the project. This is planned to be shown in two set-ups. The class-based approach infers object grasp information via object class recognition and a pose alignment. The part-based approach focuses on potential grasp points and locally estimates shape of a part to infer grasp points.

This work relates to the tasks

- **[Task 4.2] - Perceiving task relations and affordances** The objective is to exploit the set of features extracted in Task 4.1 to obtain a set of features relevant to the grasping of objects and to learn the feature relations to the potential grasping behaviours and types.
- **[Task 4.3] - Linking structure, affordance, action and task** The objective is to provide the necessary input to the grasping ontology developed in WP2, which represents knowledge about the task-relations learned. It contains relations and constraints to (1) the object and its properties such as size, shape and weight, to (2) perceived affordances (potentialities for actions) and grasping points, to (3) the task that is executed, e.g., grasping for pick up or to move as cup, and to (4) the context or surrounding of relevance. It is investigated how such a link can be efficiently established and used to obtain task-based grasping of object categories and to achieve extendibility for grasping new objects.

The work in this deliverable relates to the following final year Milestones (project month 48):

- **[Milestone 10]** Linking structure, affordances, actions and tasks and a first evaluation of representations defined by the ontology.

The advance in the last year focused on attention (Section 1.1), to early find potential grasp point, on locally establishing graspable part shapes (Section 1.2), and on learning object categories to generalise grasps to new objects in relation to known object classes (Section 1.3). Finally, we show with task-based grasping how this milestone is reached with an integrated demo on ARMAR (Section 1.4).

1.1 Attention Points for Grasping

When presented with an everyday scene, robots so for are not able to segment the scene into meaningful objects. Segmentation itself works, as shown in the last years, only for separated objects and using a planar support surface assumption.

To be able to grasp an object in a cluttered scene or from a pile or basket (task for year 4), we adopt the method to first create attention points, which may then be used as starting points for grasping. What

we essentially want is the system to segment objects that can be picked up, or, if that is not possible due to clutter or occlusion, we want to at least detect good initial grasp points. These tend to be located somewhere on parts sticking out from the scene. This leads to the problem of identifying good seed points. Inspired by pre-attentive vision theory recent research has suggested the use of attention points, which can be extracted from saliency maps.

While so far primarily 2D cues have been investigated, we exploit stereo of RGB-D images to include 3D pre-attentive cues as also known to be used in humans. We then propose a learning-based approach that extends to top-down search tasks. Using the Microsoft Kinect depth sensor sensor we have created an RGB-D image database, consisting of different types of table scenes that are challenging for segmentation, owing to the presence of fully and partially occluded objects, multi-coloured objects etc. Labelling was done by one person, whose task was to segment objects in the scenes as precisely as possible. The main novelty of the work lies in the understanding how and what pre-attentive cues shall be combined for calculating attention points for segmentation of graspable objects. This work has been published and is presented in Appendix [A] and is used to obtain seed points for potential grasping in the Demo "Empty the Basket" and for "Adaptive Grasping", where the seed point is used for further purely tactile grasping.

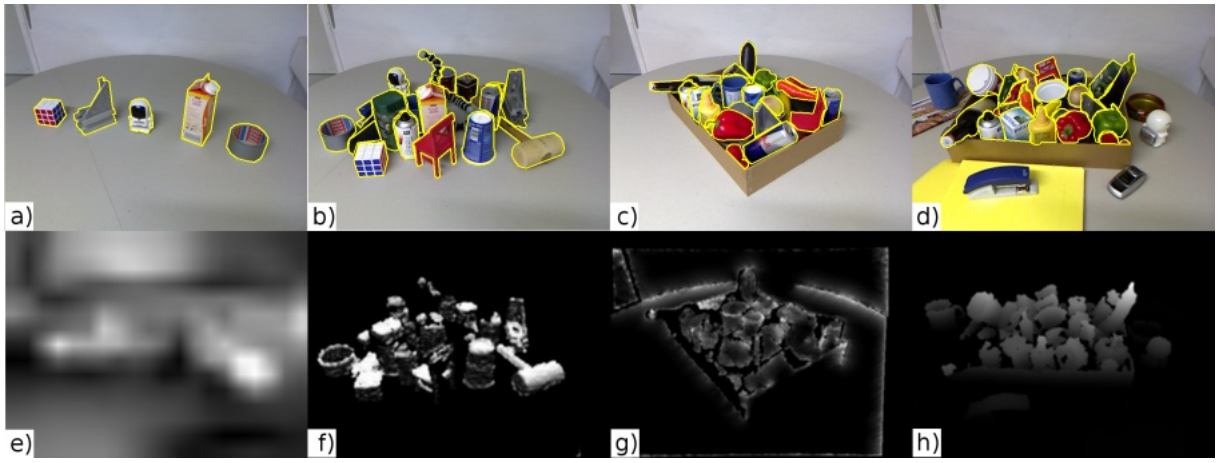


Figure 1.1: Pairs of examples scenes and corresponding saliency maps: a)/e), b)/f), c)/g), and d)/h). Images e)-h) show examples of saliency maps based on 2D cues from [Itti, 1998] and on the 3D cues relative surface orientation, occluded edges and surface height (Appendix [A]).

1.2 Part-based Grasping

Results in the first three years [Tasks 4.1 and 4.2] showed that local image information can be very well used to obtain shape information about objects. Based on this, a novel method for learning grasp points in relation to object parts is investigated. The idea is to link local object part shape with the affordances and tasks formulated in WP2. This enables to break down the detection of new objects to object parts, which in themselves typically indicate where to grasp an object. We attempt to extend the scope of affordance features to define Conceptual Equivalence Classes and to recognize these classes leading to scalable unit (part/ part assembly/ object) recognition system. The advantage is that grasp points from related object classes can then be used for grasping of new objects. A further advantage is that parts and in partiuclar part relations can be used to describe not only grasp affordances but a wide range of affordances, e.g., support, containment, 2D or 3D rolling. We made the affordance features and descriptions widely available with the idea to initiate discussion on how to model affordances beyond the grasping affordance. This work has been published and will be presented in Appendix [B].

The work is combined with the method to obtain attention points as basis for the live demo "Empty the basket". From the attention points an over segmentation is achieved that is used to locally fit parts and evaluate which parts best describe the scene. The parts are parametrised to yield grasp hypotheses, which are then selected in grasp planning.

1.3 Class-based Grasping

The idea is that from the knowledge about the object class it is possible to derive graasp hypothesis. Hence, both KTH and TUW investigated class-based grasping approaches. Furthermore, if the robot is given a specific task, the grasp hypothesis can be selected appropriately. This has been done in a combined demo at KIT on ARMAR (Section 1.4).

The work at TUW looked at the problem on how to learn many object classes efficiently. To this end the approach based on learning from 3D models available on the internet has been continued and it could be shown that the learning is highly efficient while still yielding state-of-the-art detection performance (Appendix [C]). The approach is based on feature histograms that detect the class and best view. However, for grasps the exact object pose and grasp hypotheses are necessary. This has been develop with a pose alignment approach reported in Appendix [D].

As next step we extended the learning from 3D internet models to 200 object classes and made the approach, data and procedure to evaluate object detection freely available on the internet under http://3d-net.org/. Appendix [E] presents the details. Furthermore, we then exploit the pose alignment to show that specific affordances are typically linked to object pose in the scene and can now be superimposed onto novel object once class is detected. Appendix [F] presents the results of detecting available affordances (are present to the robot) and hidden affordances (require a change of object pose to become available) of object in the scene. The Figure below gives an example of a table scene with a few objects and the affordances detected. This approach has also been used to enable pose related task learning in WP2 and is used for the task-based grasp demo.

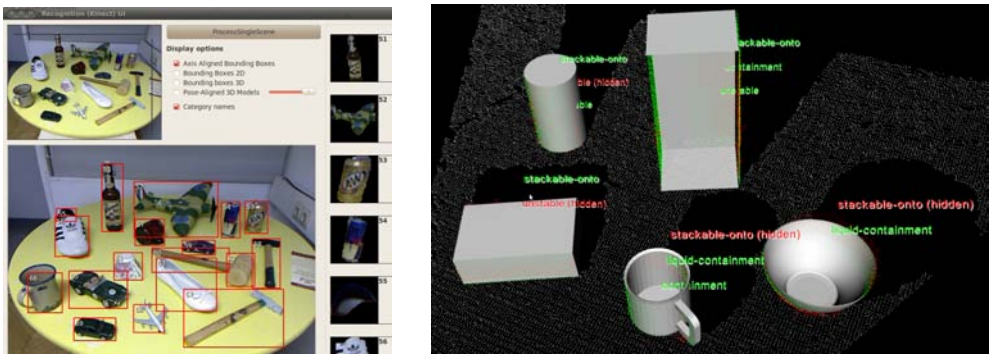


Figure 1.2: Examples of object classification (left) and of pose estimation for determining possible affordances, where a hidden affordance requires a pose change of the object.

Complementary to the approach based on depth data, at KTH we addressed the problem of grasp generation and transfer between objects that share similar geometric properties and functionality. The system models the dependencies between the tasks, actions and objects taking into account the constraints posed by each. For example, when pouring from a cup it should be grasped by its handle and not from the top. The system is built upon an (i) active scene segmentation module [Bjorkman'10], (ii) the object categorization system using integration of 2D and 3D cues, and (iii) probabilistic grasp reasoning system [Song'11]. First, an object hypothesis is generated, categorized and then used as the input to a grasp prediction and transfer system. During the experimental evaluation, we compared individual 2D and 3D categorization approaches with the integrated 2D-3D Object Categorisation system for 14 object categories, and demonstrated its the usefulness of the categorization in task-based grasping and grasp transfer in real settings. Results of this work will be published in IEEE International Conference on Robotics and Automation (ICRA) in May, 2012, see Appendix G. The approaches have been integrated in the task-based grasp demo.

1.4 Task-based grasping

We implemented a demo application in a humanoid robot ARMAR in the Karlsruhe Institute of Technology, Germany in which an object is grasped depending on its category and beforehand specified task. An object is first segmented from the scene using an active segmentation module [Bjorkman'10] and then classified to one of the object categories using the 2D Object Categorization System (developed in KTH)

integrated with the 3D Object Categorization System (developed in TUW). Finally, a robot applies a grasp to an object that is specific to an object category and predefined task. Performance of the integrated systems was demonstrated in a table top scenario for three object categories (mug, bottle, toy-car) assuming arbitrary pose of an object. Results of this work are recorded in the video: "Task-based Grasp Adaptation" (http://www.youtube.com/watch?v=rXNwBurCnTc).

References

[Bjorkman'10] M. Bjrkman and D. Kragic. "Active 3D scene segmentation and detection of unknown objects", in IEEE International Conference on Robotics and Automation (ICRA), 2010.

[Song'11] D. Song, C. H. Ek, K. Huebner, D. Kragic: "Embodiment-Specific Representation of Robot Grasping using Graphical Models and Latent-Space Discretization" in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011.

Appendix A

Appendix A: Attached Papers

A E. Potapova, M. Zillich, M. Vincze: "Learning What Matters: Combining Probabilistic Models of 2D and 3D Saliency Cues"; International Conference on Computer Vision Systems (ICVS), 2011.

B K. Varadarajan, M. Vincze: "Part Grasp Synthesis from Superquadrics based Parameterized Point Clouds", IEEE International Conference on Automation, Robotics and Applications ICARA, 2011.

C W. Wohlkinger, M. Vincze: "Shape-Based Depth Image to 3D Model Matching and Classification with Inter-View Similarity"; The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011.

D A. Aldoma Buchaca, M. Vincze: "CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues"; 3rd IEEE Workshop on 3D Representation and Recognition (3dRR) at ICCV, 2011.

E W. Wohlkinger, A. Aldoma, R.B. Rusu, M. Vincze: 3DNet: Large-Scale Object Class Recognition from CAD Models; IEEE ICRA, accepted, 2012.

F A. Aldoma, F. Tombari, M. Vincze: Supervised Learning of Hidden and Non-Hidden 0-order Affordances and Detection in Real Scenes; IEEE ICRA, accepted, 2012.

G M. Madry, D. Song, D. Kragic: "From Object Categories to Grasp Transfer Using Probabilistic Reasoning"; IEEE ICRA, accepted, 2012.

Learning What Matters: Combining Probabilistic Models of 2D and 3D Saliency Cues

Ekaterina Potapova, Michael Zillich and Markus Vincze *

Automation and Control Institute
Vienna University of Technology
{*potapova,zillich,vincze*}*@acin.tuwien.ac.at*

Abstract. In this paper we address the problem of obtaining meaningful saliency measures that tie in coherently with other methods and modalities within larger robotic systems. We learn probabilistic models of various saliency cues from labeled training data and fuse these into probability maps, which while appearing to be qualitatively similar to traditional saliency maps, represent actual probabilities of detecting salient features. We show that these maps are better suited to pick up task-relevant structures in robotic applications. Moreover, having true probabilities rather than arbitrarily scaled saliency measures allows for deeper, semantically meaningful integration with other parts of the overall system.

Keywords: 3D saliency cues, cue integration, probabilistic learning

1 Introduction

Vision in complex real world scenarios, especially unconstrained segmentation of objects, is a notoriously difficult problem and robotics has realised the importance of attention for robotic systems [23]. Vision in a robot is part of a larger system, which has specific tasks to solve. These tasks allow to derive constraints for the vision system to keep vision problems tractable. These constraints come in the form of attention operators that highlight those parts of the scene most promising for the task at hand.

The range of robotic tasks we consider for this paper includes manipulation, grasping and tracking. We therefore assume objects to appear in various locations and configurations, partly occluded, surrounded by clutter, but typically located on a supporting surface, such as a table or shelf.

What we essentially want is the system to segment objects that can be picked up, or if that is not possible due to clutter or occlusion, we want to at least detect good initial grasp points. These tend to be located somewhere on parts sticking

* The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement IST-FP7-IP-215821 GRASP 2008-2012 and from the Austrian Science Fund (FWF) under project TRP 139-N23, InSitu.

out from the scene. Pre-grasp manipulation of such parts might free the object from the pile.

Scene segmentation is one of the most researched topics in computer vision, and many different approaches have been proposed [3, 4, 17], but no generic solution suitable for every task exists. Recent state-of-the-art research in this field suggests the use of seed points to guide the segmentation process [18, 14, 22]. This leads to the problem of identifying good seed points. Inspired by pre-attentive vision theory recent research has suggested the use of attention points, which can be extracted from saliency maps, using for example a winner-takes-all (WTA) algorithm [15].

Many well-known and widely acknowledged models for computation of saliency maps, such as [10, 13, 12, 11, 1] use only 2D information about the scene. Itti-Koch-Niebur (IKN) [13] is a generic cue inspired by physiological models, and has proven its efficiency in 2D images. Fig. 1,e) shows the saliency map computed by the IKN cue for the image in Fig. 1,a). Several recent extensions to 3D take advantage of the increased availability of 3D sensing equipment, such as inexpensive laser or time-of-flight sensors and RGB-D cameras [9, 16, 21, 2].

However, classical saliency cues indicate only outliers in the scene, while we require regions with specific task-relevant properties to stand out. One can see this problem as the top-down attention task described in [20, 8], while our current goal is to build a bottom-up attention system tuned to identifying particular properties of the visual search space. Finally, given that there is a number of intuitively plausible saliency cues (2D and 3D) there is no model for combining these cues in a principled manner with respect to a given task, without using top-down specific features of required objects or parts of visual space.

We address the above issues with a learning based approach, which can be extended to top-down search tasks in the future. Using the Microsoft Kinect depth sensor sensor we have created an RGB-D image database, consisting of different types of table scenes which are challenging for segmentation, owing to the presence of fully and partially occluded objects, multi-colored objects etc. The database consists of four types of scenes: a) isolated free-standing objects (IFSO), b) occluded objects (OO), c) objects placed in a box (BO) and d) a box containing objects and surrounded by other objects (BOSO). For each type of scenes multiple configurations of objects are presented. In total there are 86 RGB-D images in the database. Task regions were hand-labeled by outlining them with a polygon. In our problem task relevant regions are whole objects. Labeling was done by one person, whose task was to segment objects in the scenes as precisely as possible. For BOSO objects we are interested only in objects situated directly in the box, that is why objects around the box were not labeled at all. Fig. 1, a)-d) show examples of labeled images.

The main novelty of this paper lies in the area of understanding how and what preattentive cues should be combined in a specific robotics task of calculating attention points for segmentation of graspable objects.

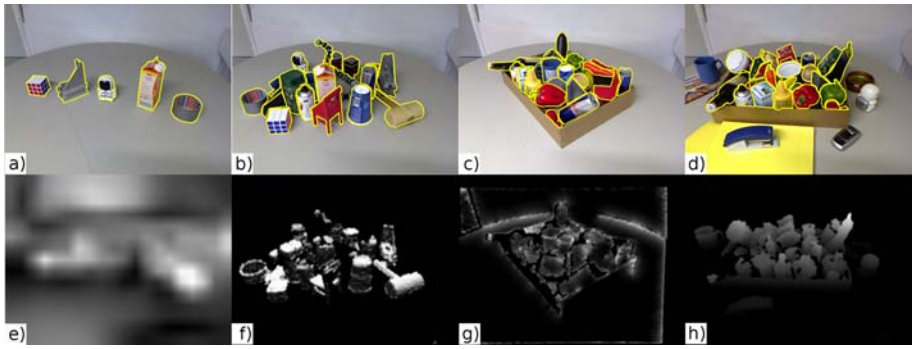


Fig. 1. Four pairs of images and saliency maps (a) and e), b) and f), c) and g), d) and h)). Images a)-d) show examples of images along with labeling for isolated free-standing objects, occluded objects, objects placed in a box and a box containing objects and surrounded by other objects respectively. Images e)-h) show examples of saliency maps based on IKN cue, RSO cue, OE cue and SH cue respectively.

2 Investigated Cues

Inspired by findings from preattentive human vision [6, 5, 19] we investigated several 3D cues, e.g. based on surface height (SH), relative surface orientation (RSO) and occluded edges (OE) and combined them with cues obtained from 2D information (color, orientation and intensity). As input we have a point cloud $P = \{\mathbf{p}_{ij}\}$ of the table scene, arranged as a rectangular array. I.e. for each image pixel i, j we have a 3D point $\mathbf{p}_{ij}$ together with its RGB color value.

2.1 Surface Height Cue

For the task of picking up objects in a cluttered scene, the simplest way to start grasping is first to pick up all objects that stick out from the clutter. These objects are good candidates for initial grasping attempts, and they should therefore be considered more interesting than the rest. These objects can be pointed out by attention points derived from the surface height preattentive cue, which is based on a height map of the scene. Fig. 1,h) shows the saliency map based on the SH cue for the image in Fig. 1,d).

To calculate height we need to determine a reference, i.e. the supporting plane on which objects rest (e.g. a table). We use RANSAC [7] to determine the plane coefficients $Ax + By + Cz + D = 0$. Note that we can assume from the task context of grasping objects from a table that such a single supporting plance exists. For every point $\mathbf{p}_{ij}$ its distance to the supporting plane $d(i, j)$ is calculated. We set d_{max} to be the distance between the ground plane and the most remote point in the point cloud. Values of the SH cue are calculated according to:

$$SH(i, j) = f(d(i, j)) \tag{1}$$

We furthermore scale height values non-linearly according to $f(x) = ax^2$ to obtain more pronounced salient regions, where a is chosen such that $f(d_{max}) = 1$

2.2 Relative Surface Orientation Cue

The surfaces of objects parallel to the supporting plane often present good candidates for first grasping positions, because they usually indicate top-surfaces of simple objects that can be easily grasped. One of our 3D preattentive cues aims to identify top-surfaces based on surface orientation. We calculate relative orientation between local surface normals and supporting plane normal. Fig. 1,f) shows the saliency map based on the RSO cue for the image in Fig. 1,b).

With $\mathbf{n}$ the normal vector of the supporting plane and $\mathbf{n}_{ij}$ the local surface normal vector determined from a plane fitted to the neighborhood of $\mathbf{p}_{ij}$, values of the RSO cue are calculated according to:

$$RSO(i,j) = |\mathbf{n}_{ij} \cdot \mathbf{n}| \tag{2}$$

2.3 Occluded Edges Cue

The success of the segmentation based on seed points depends a lot on the position of the seed point. The more central the location of the seed point with respect to the object, the higher is the probability that the object will be properly segmented. To this end we designed a cue based on occluded edges. The cue is derived from the depth map of the scene. Fig. 1,g) shows an example of the saliency map based on the OE cue for the image in Fig. 1,c). Using the Canny operator an edge map EM is calculated from the depth map. From every point $p(i_0, j_0)$ that belongs to one of the edges we create a potential field $P(\cdot)$ according to:

$$P(d) = a\frac{1}{d} - b \tag{3}$$

where d is the distance from the current point $p(i,j)$ to the initial edge point $p(i_0, j_0)$ whose influence we are calculating, a is set to 0.5 and b is set to 0.01 in our experiments. The influence is expanded only in directions of decreasing values of the depth map, i.e. the object side of the occluding edge. The value of the point $p(i,j)$ in the OE map is equal to:

$$OE(i,j) = \sum_{\forall (i_0,j_0): EM(i_0,j_0) \geq 0} P(\sqrt{(i-i_0)^2 + (j-j_0)^2}) \tag{4}$$

Finally, OE map is linearly normalized to the range [0,1].

2.4 Cue Combination

We investigated two approaches for cue combination to obtain a final saliency map SM. The first approach is similar to cue combination used in IKN method: the final saliency map SM_S is equal to the sum of individual cues:

$$SM_S(i,j) = w_1 IKN(i,j) + w_2 SH(i,j) + w_3 RSO(i,j) + w_4 OE(i,j), \tag{5}$$

where $\sum w_i = 1$ and we set $w_i = 0.25$.

The second combination method uses multiplication instead of summation, so that we obtain SM_M as multiplication of individual cues:

$$SM_M(i,j) = IKN(i,j)SH(i,j)RSO(i,j)OE(i,j). \tag{6}$$

Fig. 6 e)-h) and Fig. 6 m)-p) show examples of SM_S and SM_M combination types for different types of the scenes.

3 Probabilistic Learning

Combining cues according to Eq. 5 or 6 does not take into account the relative importance of cues. One way to address this is to learn weights for individual cues. Another possibility is to directly learn probabilistic models of cues and then combine these. We used a labeled database to train a probabilistic model of relevance for each saliency cue. For each cue c_i we learned the probability of observing that for given cue a pixel was marked as task relevant salient ($s = true$) - situated inside labeled polygons, or non-salient ($s = false$) - situated outside labeled polygons.

$$\begin{matrix} p(c_i \mid s = true) \\ p(c_i \mid s = false) \end{matrix} \tag{7}$$

We estimated parameters for normal distributions for every type of cue separately and for two types of cue combination: addition and multiplication. Note that our labels essentially mark whole objects, with parts of them being salient (different parts for different cues) and other parts not salient, i.e. we use generic labels, rather than labeling for each cue individually. But this means that estimating the above probabilities directly from the labeled images would essentially learn that inside a region labeled as salient, all sorts of saliency values can appear. But we know that inside labeled regions we are only interested in what makes part of that region salient, not the fact that not all of it is salient. To this end, during estimation of the normal distribution, we weight pixels with saliency I according to $w(I) = I^2$. Note that this measure would not be necessary with marked regions, precisely outlining salient regions for each cue individually. We chose this method however, because we want one set of generic labels, applicable to various different cues, picking up saliency somewhere inside those regions.

Fig. 2 shows estimated normal distributions of saliency values (in the range $[0, 1]$) for the IKN cue constructed for occluded objects scenes and for the RSO cue constructed for a box with objects surrounded by other objects (scenes (a) and b) respectively). We can clearly see that distributions are well separated, allowing distinction of salient from non-salient regions. Note that the choice of a normal distribution is strictly speaking not correct, as values are truncated to the interval $[0, 1]$. Further work will investigate the use of a truncated normal distribution or beta distribution on $[0, 1]$.

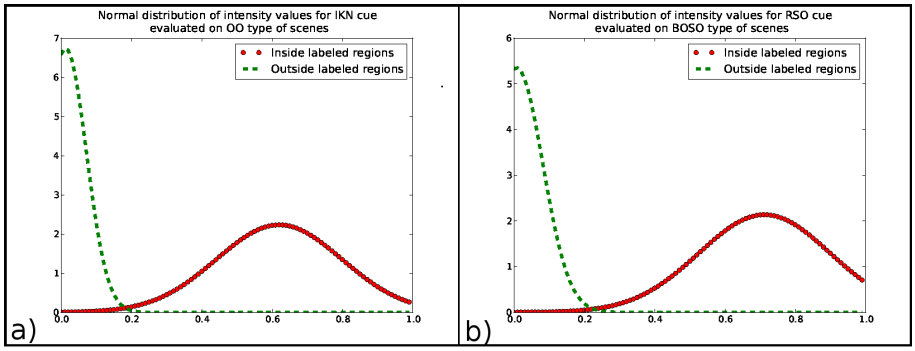


Fig. 2. Normal distribution of saliency values inside and outside labeled regions: a) for IKN cue for occluded objects scenes, b) for RSO cue for a box with objects surrounded by other objects scenes

Following Bayes rule we can then infer the posterior probability of saliency as

$$\begin{aligned} p(s \mid c_i) &= \frac{p(c_i \mid s)\ p(s)}{p(c_i)} \\ &= \frac{p(c_i \mid s)\ p(s)}{\sum_{k \in \{t,f\}} p(c_i \mid s = k)} \end{aligned} \tag{8}$$

with $p(s)$ being the prior probability of saliency. This could be obtained from top level context information, but is simply assumed 1 here, as we are more interested in the relative differences between cues.

Fig. 3 a)-d) shows the posterior probabilities of salient values for different cues and combinations of cues for different types of scenes. The smaller slope of the IKN as well as OE cues over all types of images indicates that for our type of scenes they are less distinctive than the others. This means that these cues cannot precisely distinguish regions belonging to different objects.

Based on evaluated parameters of the normal distributions, posterior probability images were built for a validation set. The relative sizes of training set and test set were 0.8:0.2.

Fig. 4 shows examples of posterior probability images for different types of cues and cue combinations for the image shown in Fig. 1 d). For an ideal probabilistic image regions of different objects should have the highest saliency values (in our case 1) and be separated from each other. As we can see from Fig. 4 among individual cues RH and RSO cues show the best performance, while combination by multiplication performs better than combination by summation.

As can be seen from the Fig. 4 the IKN cue for such complex scenes assigns high probability values to areas, which do not belong to any object. This is because IKN does not take into consideration 3D spatial positions of the objects, and thus cannot distinguish objects with e.g. similar color. Probability images give us insight into how cues can be combined in terms of top-down attention for a specific task of segmentation for grasping.

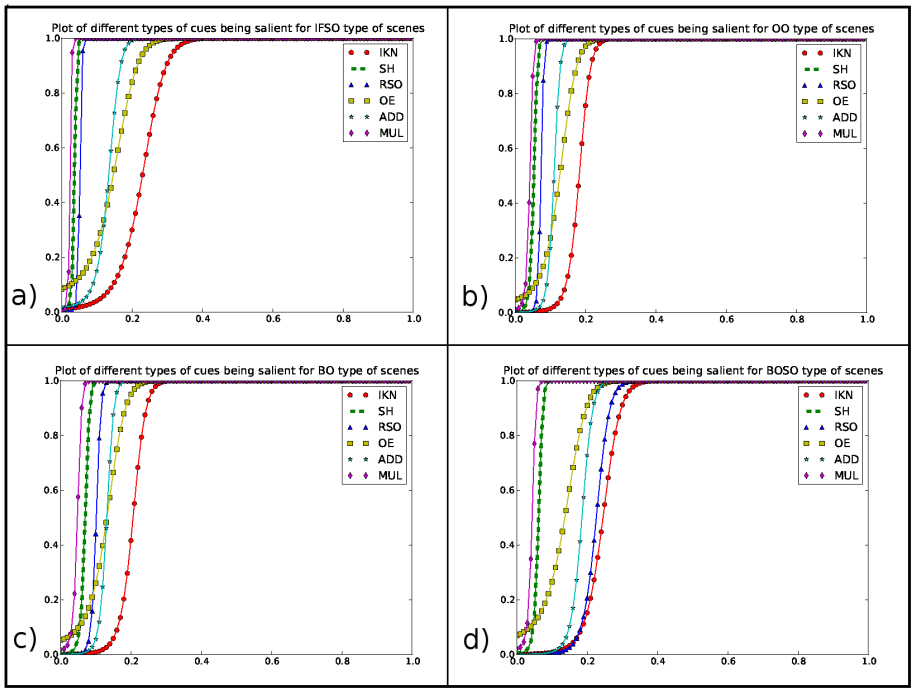


Fig. 3. Probability of salient regions being situated inside labeled regions for different types of scenes: a) isolated free–standing objects b) occluded objects c) objects placed in a box, and d) a box containing objects and surrounded by other objects for different individual cues and cue combinations (for all plots probability via salient value).

4 Evaluation and Results

To evaluate individual cues as well as the cue combinations, we calculated the ratio of first five WTA [15] attention points from the saliency map being situated inside labeled regions of a hold-out set of training images. Averaged results are presented in Fig. 5. Results indicate that especially for complicated scenes with occluded objects 3D saliency cues based on surface height and relative surface orientation perform better than simple 2D cues. Furthermore the cue based on occluded edges did not prove to be a useful cue for our tasks.

Evaluation results go along with distributions obtained from probabilistic learning, while there is still an open question what cue combination is the best for the given task and more experiments on that should be provided.

Fig. 6 shows examples of images with first five attention points indicated in blue color and corresponding saliency maps.

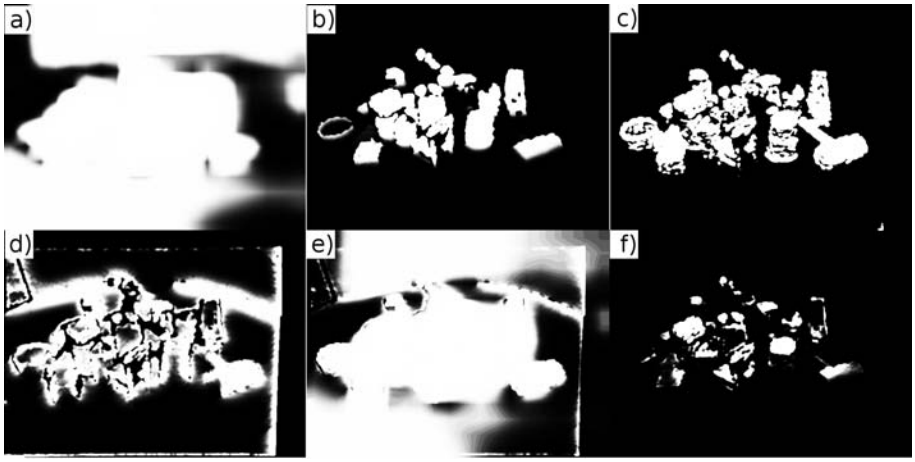


Fig. 4. Posterior probability images for image shown in Fig. 1 d) for a) IKN cue, b) SH cue, c) RSO cue, d) OE cue, e) SM_S cue and f) SM_M cue.

5 Conclusion and Future Work

In this paper we investigated the use of 3D cues to obtain attention points that can be used as seed points for segmentation of objects for robotic grasping tasks. We implemented three 3D cues to compete against the standard IKN model [13]. Scenes with growing complexity (isolated free-standing objects, occluded objects, objects in a box, and a box containing objects and surrounded by other objects) were evaluated against each cue and two types of cue combination – summation and multiplication. We furthermore estimated probabilistic models over the whole set of images for every type of cue. We could show that height and relative surface orientation cues considerably improve performance in calculating attention points on potential objects for grasping over the standard IKN model [13]. In the most complex cases the combination of both 3D cues gives clearly the best results. This indicates that 3D cues deserve more attention when moving out into the real world with robots.

Our future work will lie in the area of implementing and evaluating more types of 3D preattentive cues and using the results in actual grasping scenarios.

References

1. Achanta, R., Estrada, F., Wils, P., Süsstrunk, S.: Salient region detection and segmentation. In: 6th Int. Conf. on Computer Vision Systems. pp. 66–75 (2008)
2. Akman, O., Jonker, P.: Computing saliency map from spatial information in point cloud data. In: Advanced Concepts for Intelligent Vision Systems. pp. 290–299 (2010)
3. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: 8th IEEE Int. Conf. on Computer Vision. pp. 105–112 (2001)

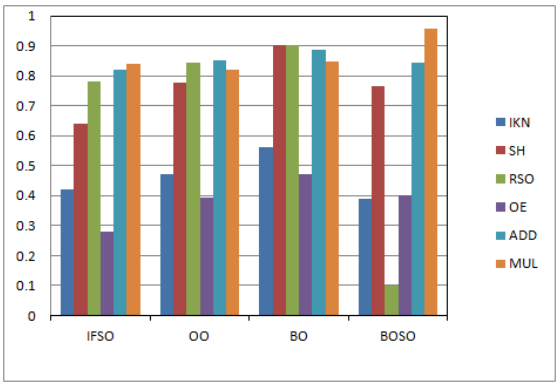


Fig. 5. The ratio of the first five attention points being situated inside different labeled ROIs (IFSO - single standing objects, OO - occluded objects, BO - objects in a box, BOSO - a box with objects which is situated among other objects.

4. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
5. Enns, J.T., Rensink, R.A.: Influence of scene-based properties on visual search. Science 247(4943), 721–723 (Feb 1990)
6. Enns, J.T., Rensink, R.A.: Sensitivity to three-dimensional orientation in visual search. Psychological Science 1/5, 323–326 (1990)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM 24, 381–395 (1981)
8. Frintrop, S., Backer, G., Rome, E.: Goal-directed search with a top-down modulated computational attention system. In: Proc. of the Annual Meeting of the German Association for Pattern Recognition (DAGM). pp. 117–124 (2005)
9. Frintrop, S., Rome, E., Nüchter, A., Surmann, H.: A bimodal laser-based attention system. Computer Vision and Image Understanding 100, 124–151 (2005)
10. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. Advances in Neural Information Processing Systems 19, 545–552 (2007)
11. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 1 –8 (2007)
12. Itti, L., Koch, C.: Computational modelling of visual attention. Nature Reviews Neuroscience 2(3), 194–203 (Mar 2001)
13. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(11), 1254–1259 (1998)
14. Ko, B.C., Nam, J.Y.: Object-of-interest image segmentation based on human attention and semantic region clustering. J. Opt. Soc. Am. A 23(10), 2462–2470 (Oct 2006)
15. Lee, D.K., Itti, L., Koch, C., Braun, J.: Attention activates winner-take-all competition among visual filters. Nature Neuroscience 2(4), 375–381 (Apr 1999)
16. Maki, A., Nordlund, P., Eklundh, J.O.: A computational model of depth-based attention. In: 13th Int. Conf. on Pattern Recognition. pp. 734–739 (1996)

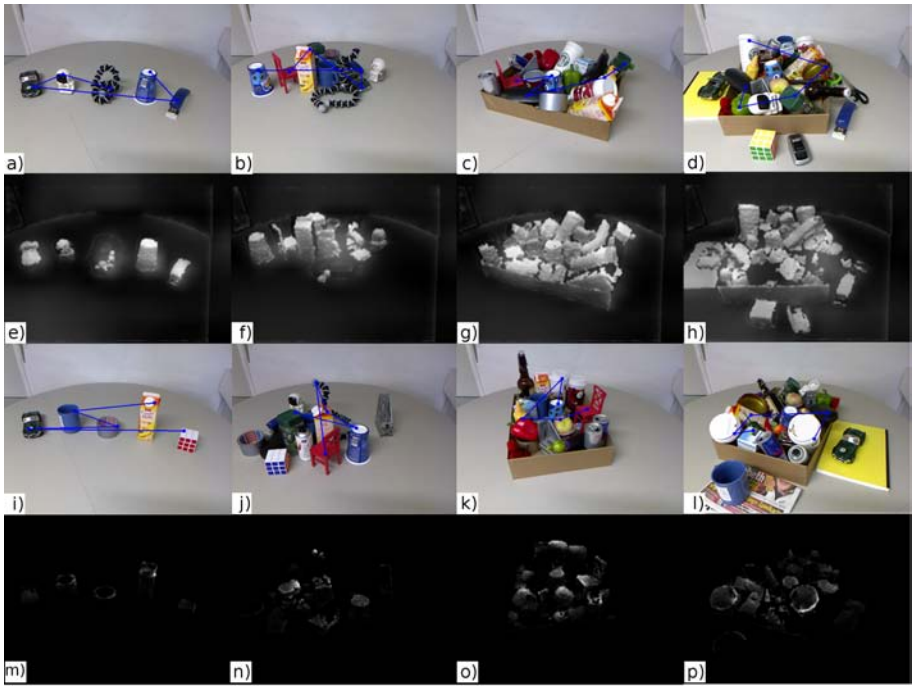


Fig. 6. Results of WTA algorithm for different types of images (left to right: IFSO, OO, BO, BOSO) and corresponding saliency maps: a)-d) present WTA results on SM_S and e)-h) are corresponding saliency maps; i)-l) present WTA results on SM_M and m)-p) are corresponding saliency maps.

17. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. Journal of Computer Vision 43(1), 7–27 (2001)
18. Mishra, A., Aloimonos, Y., Fah, C.L.: Active Segmentation with Fixation. In: Twelfth IEEE Int. Conf. on Computer Vision (2009)
19. Nakayama, K., Silverman, G.H.: Serial and parallel processing of visual feature conjunctions. Nature 320, 264–265 (1986)
20. Navalpakkam, V., Itti, L.: An integrated model of top-down and bottom-up attention for optimizing detection speed. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 2049 – 2056 (2006)
21. Ouerhani, N., Huegli, H.: Computing visual attention from scene depth. In: 15th Int. Conf. on Pattern Recognition. pp. 375–378 (2000)
22. Ouerhani, N., Archip, N., Hügli, H., Erard, P.J.: Visual attention guided seed selection for color image segmentation. In: 9th Int. Conf. on Computer Analysis of Images and Patterns. pp. 630–637 (2001)
23. Tsotsos, J.K., Shubina, K.: Attention and Visual Search : Active Robotic Vision Systems that Search. In: 5th Int. Conf. on Computer Vision Systems (2007)

3D Point Cloud Parametrization for Cognitive Grasping

Karthik Mahesh Varadarajan and Markus Vincze

***Abstract*— Grasping by Components (GBC) is a very important component of any scalable and holistic grasping system that abstracts point cloud object data to work with arbitrary shapes with no apriori data. Superquadric representation of point cloud data is a suitable parametric method for representing and manipulating point cloud data. Most Superquadrics based grasp hypotheses generation methods perform the step of classifying the parametric shapes into one of different simple shapes with apriori established grasp hypotheses. Such a method is suitable for simple scenarios. But for a holistic and scalable grasping system, direct grasp hypothesis generation from Superquadric representation is crucial. In this paper, we present an algorithm to directly estimate grasp points and approach vectors from Superquadric parameters. We also present results for a number of complex Superquadric shapes and show that the results are in line with grasp hypotheses conventionally generated by humans.**

Keywords- Superquadrics, Grasp Hypotheses, Grasp Points, Approach Vectors, Dexterous Manipulation

I. INTRODUCTION

The Grasping by Components (GBC) paradigm, first presented in [7] is a very important component of any scalable and holistic grasping system that abstracts point cloud object data to work with arbitrary shapes with no apriori data. In [5], we present a scalable object segmentation and parametric representation system for grasping. The framework for the system is shown in figure 1. This model uses Superquadrics for parametric representation of object 3D point cloud data. Most Superquadrics based grasp hypotheses generation methods perform the step of classifying the parametric shapes into one of different simple shapes with apriori established grasp hypotheses [1, 2]. These methods are suitable only for simple scenarios. But for a holistic and scalable grasping system, such as the one presented in [5], direct grasp hypothesis generation from Superquadric representation is crucial. In this paper, we present an algorithm to directly estimate grasp points and approach vectors from Superquadric parameters, extending our object segmentation and representation system presented in [5]. We also present results for a number of complex Superquadric shapes and show that the results are in line with grasp hypotheses conventionally generated by humans.

Manuscript received August 10, 2011. . The research leading to these results has received funding from the European Community's Seventh Framework Program under grant agreement no IST-FP7-IP-215821 (GRASP).

Karthik Mahesh Varadarajan and Markus Vincze are with the Vienna University of Technology, Automation and Control Institute, Gusshausstrasse 30 / E376, A-1040 Vienna, Austria (email: {kv,mv} @acin.tuwien.ac.at).

II. ALGORITHM

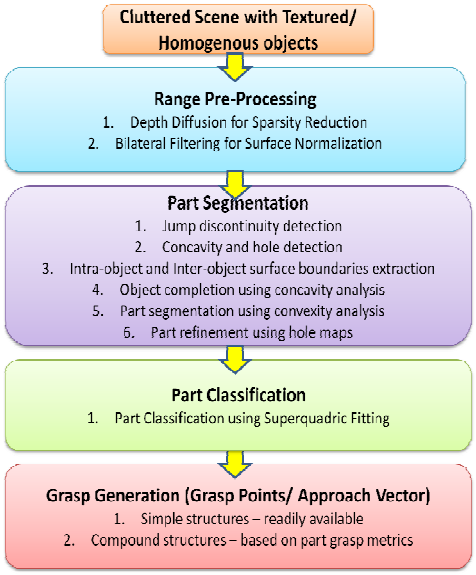


Figure 1. Grasp Synthesis Pipeline (Src: [5])

A. *Parametric Superquadric Representation of 3D Object Parts*

Superquadrics serve as highly efficient generic geometric primitives in order to obtain grasp configurations for parts/ objects with no a-priori model knowledge. Superquadrics can model superellipsoids as well as supertoroids [1, 2]. Most typical symmetrical 3D geometries – such as cubes, cones, cylinders, spheres, cuboids etc. can be modeled using superquadrics. However, super-quadrics are not very efficient in modeling concavities. Hence, we restrict the parameter values of the superquadric fitting process to only convex structures. Noise and sparsity of the 3D point cloud generated can be serious issues in the fitting process. In our framework presented in [5], this issue is resolved in the range pre-processing step. The selected data points are then resampled for use with the superquadric fitting. The convergence rate of the superquadric fitting depends on the minimality of the data size. Furthermore, it is necessary to have a uniform sampling rate in the 3D space of the object. However, the number of data points on surfaces that are tangential to the camera viewpoint is typically very low. In order to alleviate these issues, a content adaptive point-cloud importance resampling based on curvature in depth has been used (with sampling rates varying from low to high towards the edges of parts). Superquadrics can be represented by the following implicit equation:

$$\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}}+\left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}}+\left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}}=1$$

where ϵ_1 and ϵ_2 are squareness parameters and define the transition from a smooth curvature (as in the case of a sphere) to sharp edges (as in the case of a cuboid); a_1, a_2, a_3define the scale of the superquadric along the *x, y, z* dimensions. The fitting of the superquadric is based on the error metric – the inside-outside function (*F*) that evaluates whether a point is inside or outside or on the surface of the superquadric. The error metric is conventionally made independent of ϵ_1, the shape of the superquadric in order to obtain rapid convergence.

$$F^{\epsilon_1}(x,y,z)=\left(\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}}+\left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}}+\left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}}\right)^{\epsilon_1}$$

Furthermore, in order to normalize the convergence rates and directions, the scale factors $a_1a_2a_3$ are introduced in the error metric, resulting in the fitting function,

$$F_s(x,y,z)=\sqrt{a_1a_2a_3}(F^{\epsilon_1}(x,y,z)-1)$$

The final error metric to be minimized for the superquadric fitting is given by

$$\min_{\Lambda}\sum_{i=1}^{n}\left(\sqrt{\lambda_1\lambda_2\lambda_3}(F^{\epsilon_1}(x_i,y_i,z_i;\lambda_1,\lambda_2,\ldots,\lambda_{11})-1)\right)^2$$

where, λ_i are the parameters of the superquadric. Superquadric based 3D point cloud data approximation can be extremely efficient in the identification of stable grasp points. This enables a continuous space parameterization of objects in the scene. These parameters will form the feature vectors for the classification of geometric primitives that serve as discrete space parameterizations. The superquadrics fitting process is accomplished using a Particle Swarm Optimization (PSO) operating on a constrained superquadric equation parameterized as size variables, squareness parameters, coordinate transformation and rotation, tapering and bending parameters - a total of 15 parameters. For most practical scenes, it was sufficient to carry out the fitting process using only 11 parameters, by excluding the bending and tapering forms. Fitting of super-quadrics (based on 15/11 parameters) to pruned 3D data is a relatively easier task due to quantitative nature of the representation. The suitability of initial conditions is very important for rapid convergence.

B. Grasp Points Generation for Simple Shape Primitives

The final step in the pipeline is the generation of grasp points. For a given embodiment, the best set of grasp points for simple geometric primitives is well established (for eg. [3]). For the case of superquadric structures that do not fit into one of the shape descriptions, we use the closest match. For a two finger Otto Bock hand, the following grasping schema is defined:

Cubes/ Cuboids: Cylinder pregrasp shape such that the two fingers contact opposite faces. The palm should be parallel to the face orthogonal to the two opposing faces.

Spheres: Spherical pregrasp shape with the palm approach vector passing through the center of the sphere.

Cylinders/Cones: Based on the initial pose and size of the cylinder, it can be grasped from the side, or from either end.

(a) Side Grasp: Cylindrical pregrasp with the approach vector perpendicular to the side surface.

(b) End Grasp: Spherical pregrasp shape with approach vector perpendicular to end face.

For the case of cones, depending upon the size of the cone, an end grasp may be more stable.

Additional parameters such as number of parallel planes, divisions of 360 degree, grasp rotations and 180 degree rotations [3] together with constraints on time and grasp accuracy or learning of grasping modes from knowledge bases [4] can be used to decide the grasping points.

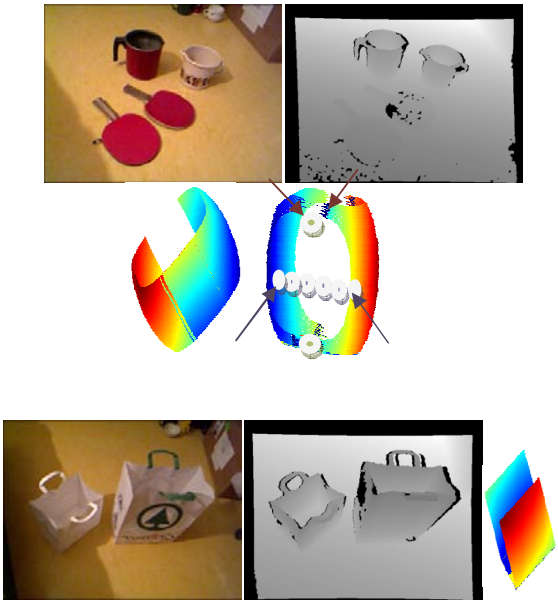


Figure 2. Input image, depth map, generated Superquadrics and grasp points based on classification

C. Grasp Points Generation for Complex Shape Primitives

For the case of complex Superquadric shapes, it is difficult to classify the shapes into one of different types with pre-established grasp points and approach vectors. In this case, it is necessary to generate grasp points from the equation of the Superquadric directly. In our framework, we employ the following approach for generation of grasp points from the superquadric equation.

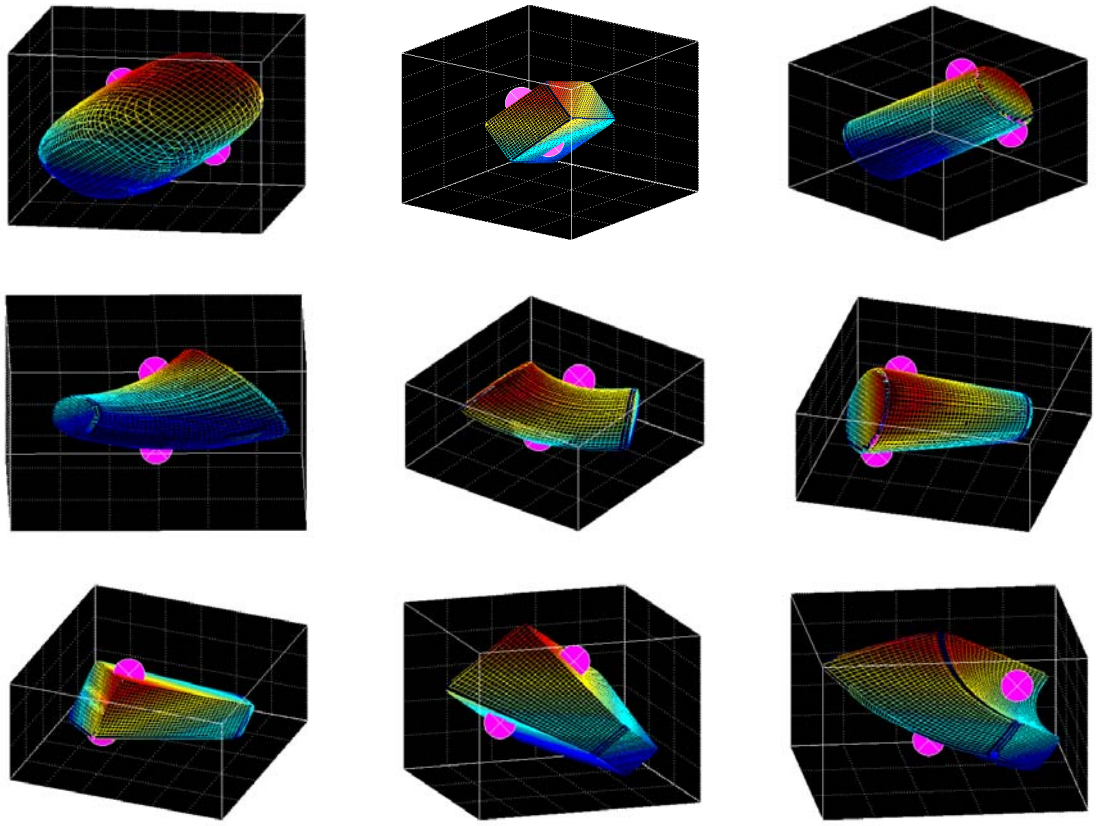
If the value of the tapering parameter is non-zero, the global extrema (minima) for the tapering function is found. Note that, unlike the standard Superquadric equation, which supports only linear taper, it is possible to apply this method for non-linear taper, such as in the case of the equation representing the shape of an hour-glass. This minima is chosen as the first dimension and the location of the minima as the coordinate of contact in the direction of the first dimension.

This minima is then checked for stability. In other words, the minima should have a saddle on either side. In other words, the value of the tapering function should be have a numeric value on either side of the minima (i.e. the minima cannot be at the end of the domain of the function) and should be larger than the minima. This prevents the selection of unstable minima such as the tip of a cone. If the above condition is not satisfied, the global minima is replaced by the next stable local minima. This process is repeated until a stable point is found. For most conventional tapered objects, the chosen extrema value is the maxima. Note that stability check for maxima is not necessary. If the Superquadric equation has zero taper (rounded to fitting precision), then the minimum scale value ($a_1, a_2,$ or a_3) is chosen as the first dimension.

If the values of the bending parameters are non-zero (rounded to fitting precision), the point in the bending function, where the curvature is maximum is estimated. In other words, the kink in the bent surface is determined. If the bending is uniform, the center of the axis of the bend is chosen. This approach ensures that the grasp is along axial coordinates where post-grasp stability of the object part is maximized.

Next, two parallel planes normal to the first dimension are generated. These planes are typically defined by the other two dimensions of the Superquadric and represent the surface of contact of the hand.

A third plane along the first chosen dimension and passing through the center of the Superquadric is generated such that it bisects the two parallel planes along two line segments. This plane is chosen such that the two line segments are parallel to the longest dimension of the Superquadric. The line segments form the locus of points at which an optimal grasp can be made. In figure 3, the red line segments denote the locus of all grasp point pairs for a single approach vector (other loci arising from symmetric grasp point pairs are excluded in the rendering). For the case of tapered Superquadrics, the grasp points loci are reprojected onto the tapered surface of the object.



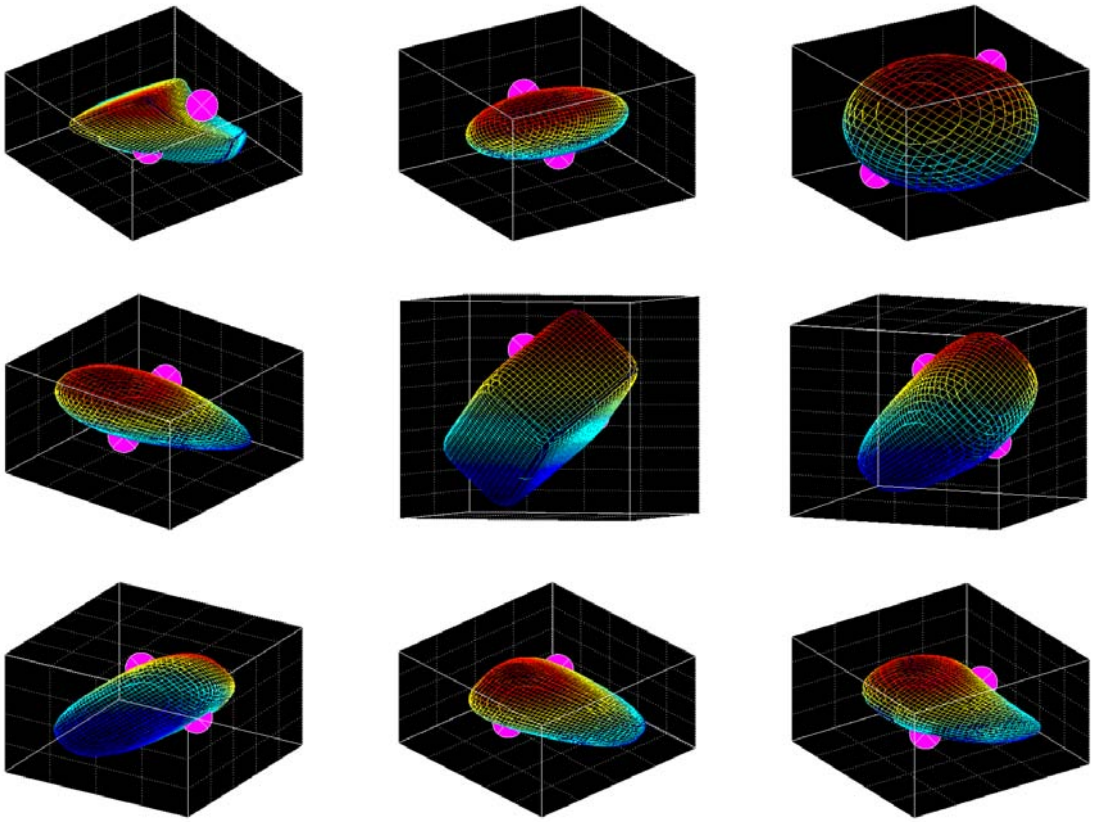


Figure 3. Primary grasp points generated parametrically for a number of arbitrary Superquadric shapes. Grasp points are denoted in magenta. The Superquadric mesh is also shown.

Based on the width of the gripper or the distance between the fingers, the final grasp point is chosen. In figure 3, these points are represented in magenta. The maximal depth (excluding a safety margin) that the hand can grasp determines the grasp point. The approach vectors are shown in blue. As seen from the figure, the approach vectors are normal to the first dimension.

It should be noted that due to the symmetric representation of Superquadrics, more than one pair of grasp points can be generated for the same object. For e.g., in the case of a cuboid, two grasp point pairs are generated at the top and the bottom of the cuboid. In the case of a cylinder, multiple grasp points are generated along the curved surface of the cylinder by symmetry. These grasp points are ranked based on gravity. In other words, since it can be expected that the arm is likely to grasp the object successfully from the top rather than from the bottom, these grasp points are ranked higher. The gravity of the scene is determined from the orientation parameters of the Superquadric. Hence, points on the top of the cube are ranked higher than the points at the bottom.

III. RESULTS

Figure 3 lists grasp points (in magenta) for a number of arbitrary Superquadric structures. It can be seen that the chosen grasp points are in line with typical grasp hypotheses that humans perform while grasping arbitrarily shaped objects. The method is shown to work successfully with a number of complex shapes.

This list can be passed on to a path planning system such as OpenRAVE, which chooses the most appropriate pair of grasp points for grasp execution, based on the priority of the points on the list and based on the 3D occupancy space around the object. In other words, if the path required for the highest ranked grasp hypothesis is occluded by neighboring objects, it is possible to choose the next best grasp hypothesis for execution. Figure 4 shows objects selected in a complex scene for fitting along with OpenRAVE simulation snapshots showing Superquadric approximates of the selected objects and successful grasps using chosen grasp points. Note that in the OpenRAVE simulation snapshots, the objects are depicted with normalized pose to demonstrate the quality of grasps based on the fitting of the Superquadrics and to discard the effects of path planning. The object selected for grasping in

the scene is shaded cyan. The selection was done using an attention driven approach [8], followed by segmentation using the algorithm in [5].

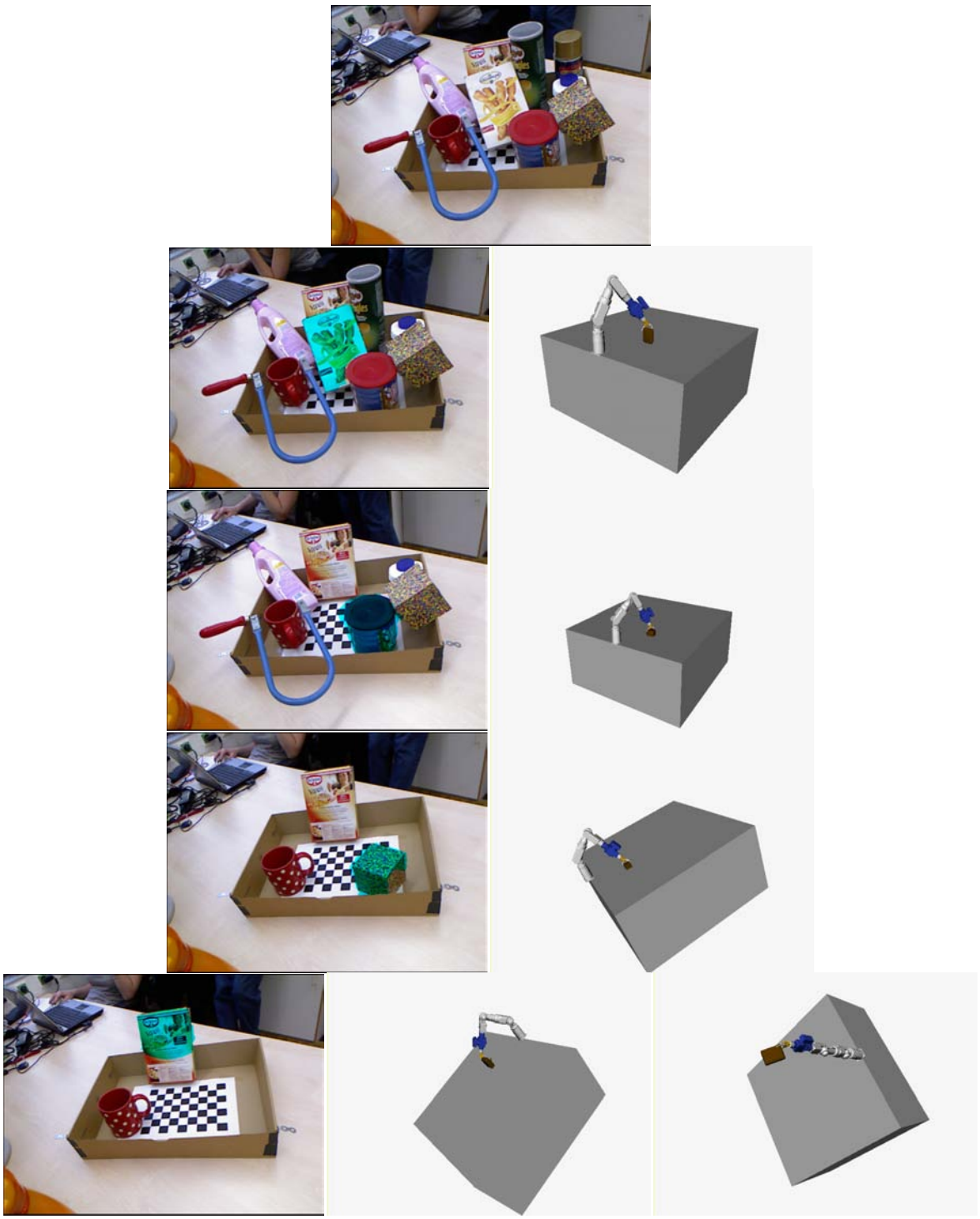


Figure 4. Row1. Input scene. Rows2-5: Objects selected for grasping in each scenario (shaded cyan), along with OpenRAVE simulation snapshots showing Superquadric approximates of the selected objects and successful grasps using chosen grasp points

IV. CONCLUSION

In this paper, we have presented a scalable grasp hypothesis generation system from parametric 3D point cloud representation. The actual integration of the framework with our system in [5] and evaluation of the hypotheses on real objects form future work.

REFERENCES

[1] D. Katsoulas, CC. Bastidas, D. Kosmopoulos, "Superquadric Segmentation in Range Images via Fusion of Region and Boundary Information", PAMI, Vol. 30, No. 5, May 2008.

[2] F. Huang, X.P. Fan, "Reconstruction of Superquadric 3D Models by Parallel Particle Swarm Optimization Algorithm with Island Model", ICIC 2005, Part I, LNCS 3644, pp. 757-766, Springer Verlag, 2005.

[3] AT Miller, S Knoop, HI Christensen, PK Allen, "Automatic grasp planning using shape primitives", ICRA 2003.

[4] Curtis, N. Jing Xiao, "Efficient and effective grasping of novel objects through learning and adapting a knowledge base", IROS 2008.

[5] KM. Varadarajan, M. Vincze, "Object Part Segmentation and Classification in Range Images for Grasping", ICAR 2011.

[6] C. Raju, KM. Varadarajan, N. Krishnamurthi, S. Xu, I. Biederman, T. Kelly, 'Cognitive Object Recognition System', Perception, Unmanned Systems Technology, SPIE (2010).

[7] KM. Varadarajan, M. Vincze, 'Affordance based Part Recognition for Grasping and Manipulation', IEEE International Conference on Robotics and Automation (ICRA) 2011, Workshop on Autonomous Grasping (2011).

[8] E.Potapova, M. Zillich, M Vincze,"Learning What Matters: Combining Probabilistic Models of 2D and 3D Saliency Cues", International Conference on Computer Vision Systems (ICVS), 2011.

CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues

Aitor Aldoma and Markus Vincze
ACIN - Technische Universitat Wien
aldoma,vincze@acin.tuwien.ac.at

Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu and Gary Bradski
Willow Garage

Abstract

This paper focuses on developing a fast and accurate 3D feature for use in object recognition and pose estimation for rigid objects. More specifically, given a set of CAD models of different objects representing our knoweledge of the world - obtained using high-precission scanners that deliver accurate and noiseless data - our goal is to identify and estimate their pose in a real scene obtained by a depth sensor like the Microsoft Kinect. Borrowing ideas from the Viewpoint Feature Histogram (VFH) due to its computational efficiency and recognition performance, we describe the Clustered Viewpoint Feature Histogram (CVFH) and the cameras roll histogram together with our recognition framework to show that it can be effectively used to recognize objects and 6DOF pose in real environments dealing with partial occlusion, noise and different sensors atributes for training and recognition data. We show that CVFH outperforms VFH and present recognition results using the Microsoft Kinect Sensor on an object set of 44 objects.

1. Introduction and related work

Object recognition and pose estimation is a well studied problem in computer vision due to its endless applications in scene understanding, robotics, virtual reality, *etc*. Several feature descriptors for object recognition have been presented in the literature, both in 2D (e.g. [6]) and 3D (e.g. [12]). However, they still can not manage to resolve the full object recognition problem, especially when faced with hard problems such as textureless objects noise or missing parts of the objects. For both 2D and 3D, there are mainly two different approaches to the object recognition problem: local (e.g. [1],[2],[4]), or global descriptors (e.g. [5],[8]).

The latter and most relevant in the scope of the paper, describe the geometry, appearance or both of a whole partial view of an object and are more robust to noise than local features, specially in the 3D domain but they require the notion of object before recognition which is normally given by a prior segmentation procedure. Because of its global nature they have problems dealing with missing parts which are caused by partial occlusions, sensor limitations or segmentation artifacts (see Figure 1).

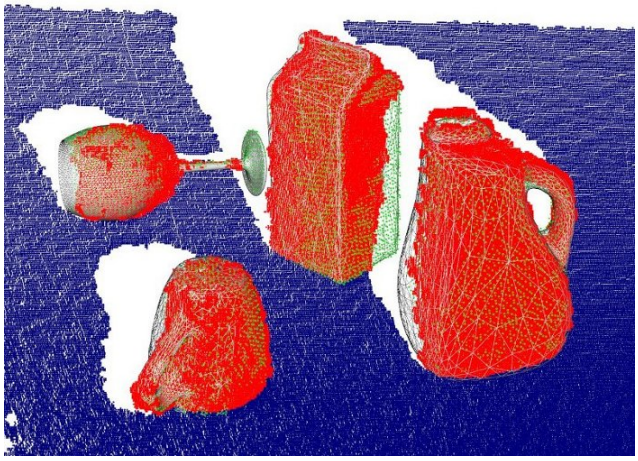


Figure 1. The figure shows how CVFH can deal with limited amounts of occlusion. The support plane is shown in blue, the segmented object candidates from the current scene in red, the recognized views from the database in green and the corresponding models overlapped as grey meshes.

These artifacts increase the complexity of the problem, mostly in our specific scenario where we want to develop a feature that can be trained on 3D CAD models and yet perform recognition on real data. Almost none of the descriptors presented in the literature (excepting [13]) have tackled the problem of training on synthetic data and matching on real data. Creating training databases for a reasonable number of objects using real devices can be a cumbersome task, even very difficult if one would like to have all different viewpoints and poses of an object. On the other hand, there are publicly available databases of CAD mod-

els and accurate 3D meshes for thousands of objects found in our daily life (e.g., Google Warehouse). Given a 3D CAD model and a rendering system, it is straightforward to place a virtual camera around the object and obtain all desired viewpoints without the need of calibrated systems and a time-consuming capturing process. We believe this is a crucial factor for cost and ease of scaling the set of objects the robot can learn and manipulate.

With the advent of the Kinect, depth information at ranges of 0.8-3.5 meters can be obtained at framerate and at a moderate price. This cost breakthrough is an enabler for vision systems and robotics and new low cost depth sensors such as the WAVI Xtion [9] are following rapidly. Now that depth information is cheap and easy to get, there is a need to develop efficient 3D features that will work effectively with this data in order to support robot object recognition plus 6DOF pose for manipulation.

We decided to build on the VFH feature which is efficient to compute, and already showed high discriminability in previous work [8]. As described below, VFH has shortcomings to perform recognition on real data when trained on synthetic data.

The rest of the paper is organized as follows: In section 2 the Viewpoint Feature Histogram is reviewed and used to motivate the Clustered Viewpoint Feature Histogram (CVFH), presented in section 3, that meets our goal of allowing for training on 3D CAD models and yet performing well on real world data. In section 4, the Camera's Roll Histogram is presented as an efficient way to deal with the invariance to rotations around the camera axis that appear in 3D global descriptors based on partial views. In section 5, we present the recognition framework allowing for training and recognition which includes the histogram metric used for nearest neighbor searches together with the post-processing applied after CVFH recognition to refine the results. In section 6, we compare CVFH against VFH and show that CVFH outperforms it. Finally, we conclude in section 7 and present our future work lines.

2. The Viewpoint Feature Histogram

The VFH descriptor is a compound histogram representing four different angular distributions of surface normals. Let p_c and n_c be the centroids of all surface points and their normals of a given object partial view in the camera coordinate system (with $||\mathsf{n_c}|| = 1$). Then $(\mathsf{u}_i, \mathsf{v}_i, \mathsf{w}_i)$ defines a Darboux coordinate frame for each point $\mathsf{p_i}$ (see [10]):

$$\begin{aligned} \mathsf{u}_i &= \mathsf{n_c} \\ \mathsf{v}_i &= \frac{\mathsf{p_i} - \mathsf{p_c}}{||\mathsf{p_i} - \mathsf{p_c}||} \times \mathsf{u}_i \\ \mathsf{w}_i &= \mathsf{u}_i \times \mathsf{v}_i \end{aligned} \tag{1}$$

The normal angular deviations $\cos(\alpha_i)$, $\cos(\beta_i)$, $\cos(\phi_i)$ and θ_i for each point $\mathsf{p_i}$ and its normal $\mathsf{n_i}$ are given by:

$$\begin{aligned} \cos(\alpha_i) &= \mathsf{v}_i \cdot \mathsf{n_i} \\ \cos(\beta_i) &= \mathsf{n_i} \cdot \frac{\mathsf{p_c}}{||\mathsf{p_c}||} \\ \cos(\phi_i) &= \mathsf{u}_i \cdot \frac{\mathsf{p_i} - \mathsf{p_c}}{||\mathsf{p_i} - \mathsf{p_c}||} \\ \theta_i &= \mathrm{atan2}(\mathsf{w}_i \cdot \mathsf{n_i}, \mathsf{u}_i \cdot \mathsf{n_i}) \end{aligned} \tag{2}$$

Note that $\cos(\alpha_i)$, $\cos(\phi_i)$ and θ_i are invariant to viewpoint changes, given that the set of visible points does not change. For $\cos(\alpha_i)$, $\cos(\phi_i)$ and θ_i histograms with 45 bins each are computed and a histogram of 128 bins for $\cos(\beta_i)$, thus the VFH descriptor has 263 dimensions.

Though VFH showed promising results in [10], it has a few shortcomings:

- it is invariant to the size of the object as the compound histogram is normalized by the total number of points in the partial view;
- it is invariant to rotations around the camera's view direction, so it does not allow full pose estimation;
- using the centroid and average normals ($\mathsf{p_c}$ and $\mathsf{n_c}$) to build the Darboux coordinate system, makes VFH sensitive to missing parts of the object caused by partial occlusions, segmentation or sensor artifacts.

3. The Clustered Viewpoint Feature Histogram

As outlined in section 2, the major flaws to VFH are its sensitivity to noise and occlusions (e.g. missing parts of the object) and the fact that it is invariant to rotations about the camera axis. By analyzing the data obtained from the Kinect, we noticed that surfaces that are at a steep angle relative to the sensor as well as parts that are close to object borders contain more noise or even miss a few depth estimates (see Figure 2).

These effects can result in unstable estimations of the object points and normals centroid (p_c and n_c from Eq. 1), thus affecting the resulting VFH and making it unsuitable to match against the corresponding synthetic view that will not present these artifacts.

The main idea behind CVFH is to take advantage from the object parts that can be robustly estimated by the depth sensor and use them to build the Darboux coordinate system while still using the whole partial view to compute the descriptor.

Formally, we propose to describe a partial view of an object, represented by a set of points $\mathcal{P}$, as a set $\mathcal{H}$ of Clustered Viewpoint Feature Histograms. The cardinality of $\mathcal{H}$ is the same as the cardinality of $\mathcal{S}$, where $\mathcal{S}$ is the set of stable regions found on $\mathcal{P}$ using the procedure defined in the upcoming section 3.1.

Figure 2. Example of an incomplete surface due to limitations of the sensor.

Taking $\mathsf{s}_i \in \mathcal{S}$ with $\mathsf{s}_i \subseteq \mathcal{P}$, we can define a Darboux coordinate system $\mathcal{D} = (\mathsf{u}_i, \mathsf{v}_i, \mathsf{w}_i)$ like in Eq. 1 but in this case p_c and n_c represent the euclidean centroid and normal centroid of s_i and not of the whole partial view $\mathcal{P}$. Given $\mathcal{D}$ and using Eq. 2, the normal angular deviations for all points in $\mathcal{P}$ can be computed.

Let then $(\alpha, \phi, \theta, \beta)$ represent the normal angular deviations already bined in (45,45,45,128) bins, the CVFH histogram $\mathsf{h}_i \in \mathcal{H}$ is defined as the following concatenation:

$$(\alpha, \phi, \theta, \mathcal{SDC}, \beta) \tag{3}$$

where $\mathcal{SDC}$ represents the Shape Distribution Component of CVFH computed as follows:

$$\mathcal{SDC} = \frac{(\mathsf{p}_c - \mathsf{p}_i)^2}{\max((\mathsf{p}_c - \mathsf{p}_i)^2)} \tag{4}$$

The number of bins used for this component is again 45 thus making a total size of 308 for CVFH. This component allows to differentiate surfaces that have very similar normal distributions and sizes but their points are distributed differently. For instance we could differentiate an elongated planar surface from a more compact planar surface.

To avoid scale invariance, each bin in CVFH count the absolute number of points falling in that bin. To reduce ambiguities, we first construct a voxel grid over our point cloud data with a fixed voxel size, and reduce the cloud to the set of voxel centroids. Because the actual size of the object is given by the 3D sensor, the amount of points for a given view will be the same no matter what the distance to the camera is. Avoiding the normalization step allows us to distinguish between objects of different size but identical shape. It also makes the descriptor more robust to missing parts of the object, as this will only influence local parts of the descriptor (compare Figure 3). Normalizing the histogram by the total number of points would increase the bins height under the presence of occlusion.

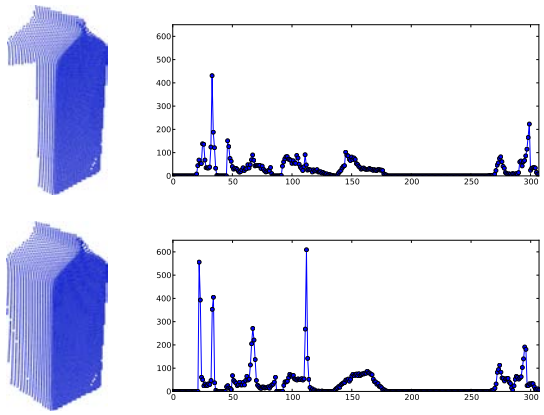


Figure 3. The CVFH histograms become additive when the centroids are consistent. *top:* Missing part on the view and the correspondent CVFH signature. *bottom:* Whole view and and the correspondent CVFH signature.

The advantages of CVFH are two-fold: (i) the coordinate system is more likely to resemble the one obtained from the synthetic view making the descriptor more stable and (ii) because the set of CVFHs represent a multivariate description of the partial view, we can better handle occlusions as long as any of the stable region is visible. Please note that the CVFH histograms in H are independent from each other and not complementary as they describe the same geometry but encode them differently. To understand how CVFH is used for recognition, we refer the reader to the next section (Recognition Framework).

3.1. Stable regions clustering

To overcome the instability caused by missing object parts and local noise artifacts, we first identify stable regions in partial view obtained by the depth sensor. To do so, we apply a smooth region growing algorithm on the points obtained from a partial view of an object after removing points with high curvature (caused by noise, object edges or non-planar patches).

Each new cluster is initialized with a random point. A point p_i with normal n_i is added to a cluster C_k if the cluster contains a point p_j with normal n_j in the direct neigbourhood of p_i with a similar normal, i.e. the following constraint is fulfilled:

$$\exists p_j \in C_k : ||p_i - p_j|| < t_d \wedge n_i \cdot n_j > t_n \tag{5}$$

For our experiments, t_d is set to three times the voxel grid size and t_n to $\cos(10°)$. For each stable region, a CVFH descriptor is computed as outlined in the previous section. The number of stable regions for a specific partial view defines the cardinality of the descriptor set $\mathcal{H}$.

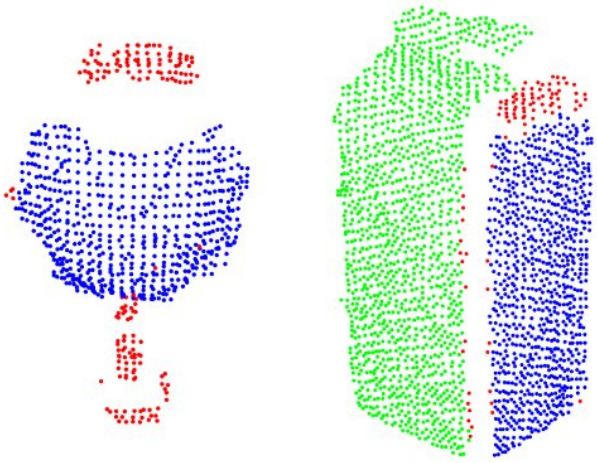


Figure 4. Free shape smooth clustering. *Left:* a wine glass, and *right:* a milk carton. Smooth surfaces are clustered together. Points in red do not belong to any cluster and points with high curvature are not shown, e.g. at the edges of the milk carton.

Only regions with more than 50 points in total are considered to be stable and taken into account. In the case that no regions are found that fulfill these conditions, the CVFH centroids are computed using all points in the partial view.

Intuitively, we are trying to define a stable location to base the computation of the CVFH descriptor even when parts of the objects are missing. For instance, the base and stem of the wine glass in Figure 4 is partly missing, which usually happens due to oversegmentation of its support plane in an earlier processing step. This will affect the descriptor centroids if the complete partial view is used, but the stable region shown in blue remains unchanged. In the case of the milk carton where 2 stable regions are found, the centroid for one of the dominant surfaces stays stable when the other one is occluded and thus the stable CVFH will allow for a positive recognition (see Figure 1 where part of the milk cartoon is occluded).

4. Camera roll histogram and 6DOF pose

Most descriptors based on views of an object like VFH, CVFH, CAP-SIFT [3] are unable to deliver a complete 6-DOF pose. Due to the this invariance of CVFH with respect to rotations along the view direction of the camera (roll), the object and viewpoint recognition is determined up to an unknown rotation. To determine the correct orientation of the object, we introduce a new descriptor that is not invariant to the roll angle. To avoid a higher dimensionality in the overall descriptor by extending it, which would decrease the performance of the object/viewpoint recognition noticeably, we use a final optimization step to find the correct roll angle. Since the computation of the roll angle is only done for the best N candidates from the CVFH matching step and furthermore is efficient to calculate, the overall performance is not affected drastically.

For each CVFH descriptor in $\mathcal{H}$, an additional histogram is computed - *the camera's roll histogram*. We project the normals at each point onto a plane that is orthogonal to the vector given by the camera center and the centroid of the stable region used to compute CVFH. For the projection, we compute a rotation-axis v and a rotation angle θ using Eq. 6 that transforms the CVFH centroid $\mathsf{p_c}$ to coincide with the camera's z-axis. Since we use an orthographic projection, the projected normals are given by the first two components of the transformed normals n_i.

$$\begin{aligned} v &= \frac{p_c \times z}{||p_c||} \\ \theta &= -\arcsin\left(||v||\right) \end{aligned} \tag{6}$$

The roll histogram is then computed by taking the angle of the projected normal relative to the up-view vector of the camera on the plane. The histogram contains 90 bins giving an angular resolution of 4 degrees. The number of bins for the camera-roll-histogram is selected from our empirical evaluations to provide a reasonable trade off between efficiency and accuracy. Due to noise in the input data, we weight the projected normals by their magnitudes. This removes most of the equally distributed noise in the histogram, resulting from unstable projections of normals that are almost parallel to the roll axis of the camera.

Figure 5 shows two histograms of the same object. The upper one is from the object in upright orientation, whereas the bottom histogram is computed from the object rotated around the roll axis by $44°$.

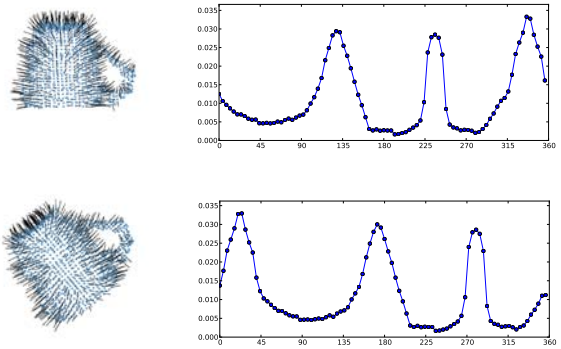


Figure 5. The camera roll histograms of the same object in different orientations.

In order to estimate the object's rotation around the roll axis, we need to find an orientation where the two roll histograms match best according to a metric. This can be considered a correlation maximization problem. Therefore, we apply a Discrete Fourier Transform for both histograms, and multiply the complex coefficients of the database view with

the complex conjugate coefficients, and perform the inverse transform to compute the cross power spectrum R. The peaks of this spectrum appear at rotation angles that align the two roll histograms well.

There are cases where the power spectrum of two roll histograms can have multiple high peaks due to different kinds of symmetries. Also, partial occlusions or sensor noise might deteriorate the roll histograms, so it is generally not sufficient to rely solely on the maximal peak in R.

In order to select a set of orientations that can be pruned in a subsequent test, we select a minimum threshold t_p for peaks, and add peaks with higher magnitude to the set. We start with the highest peak, adding peaks if their corresponding rotation angles do not fall within a certain distance band t_d of any of the previously added peaks. This ensures that the set of orientations does not contain multiple entries for very similar alignments, but captures local maxima that are distributed over the whole set of rotations, if they indicate a good alignment.

In our experiments, we set $t_d = 12°$ and chose a relatively high value for t_p in order to keep the size of the rotation set small. We found a value of $t_p = 0.9 * \max(R)$ to yield a low number of peaks - typically up to 10 peaks - while still capturing corner cases.

5. Recognition framework

In this section, we concentrate on the recognition framework which consists of two different parts: an offline training stage where the CVFH descriptors are computed for the models in our training set, and an online recognition stage, in which the real scene is processed. The recognition stage includes segmentation, recognition and pose estimation using CVFH and final refinement of the recognition results. Please note, that in this case, segmentation refers to finding possible objects candidates in the scene and not to the stable regions clustering step presented before.

5.1. Training stage

Our training data is generated from a set of CAD models. Because CVFH works on views from object, our first step is to take each of the CAD models and generate a set of distinguishable views. We place a virtual camera on the vertices of a tesselated sphere looking at the CAD model of the object and render the object seen from that viewpoint into a depth buffer from which we can efficiently extract a partial pointcloud.

For each view, the CVFH descriptor and roll histogram is computed. Views that are not distinguishable, like symmetric objects as bottles or bowls, are not considered, reducing the initial number of 80 view to about 12 views per object.

To decide which views can be removed, we align two different views of the same object using the camera's roll histogram and compute the overlapping between the aligned point clouds by searching for each point in one of the views the nearest neighbor in the other view. A point is considered not to overlap if the nearest neighbor is not within a range of twice the voxel grid size. If more than 2.5% do not overlap the view is considered to be different and the next view is checked.

5.2. Recognition stage

The recognition stage runs on a raw pointcloud from a depth sensor, which in our case is the Kinect. We proceed first with a segmentation of the scene using dominant plane extraction and Euclidean segmentation on the remaining points [11]. The segmented groups of points represent the objects to be recognized. Independently for each object in the scene:

1. Compute a set of CVFH descriptors (H) and camera's roll histograms. Please note, that each CVFH descriptors is paired with a camera roll histogram.

2. For each CVFH in H, a nearest neighbor (NN) search is performed to find the N closest CVFH descriptors in the training set, giving a set of views from the trained objects.

3. As we have performed as many NN-searches as elements in H, the best N candidates according to the metric given in Eq. (7) are selected.

4. For the resulting N view candidates the roll angle is determined using the roll histogram matching and 6DOF pose estimation (as detailed in section 4).

5. After aligning the views using the pose and roll information gathered so far, an additional ICP [14] step is used to refine the alignment.

6. Finally the N best view candidates are sorted using the number of inliers from the last iteration of ICP using a distance threshold of twice the voxel grid size.

Because of its efficiency, we use the FLANN library [7] to perform the nearest neighbor search. FLANN includes different search and indexing methods such as linear search, randomized kd-trees or hierarchical k-means indexing. Moreover, it provides different distance and histogram comparison metrics for high dimensional spaces, including e.g. L1, L2, Histogram Intersection, and ChiSquare.

We have performed different empirical experiments to determine which is the best metric for our needs. The major problem with metrics like L1 and L2 are its sensitivity to outliers. Dealing with partial occlusions implies that the histograms will have outliers due to missing parts of the objects even if the rest of the histogram is shaped correctly. Therefore, we use the following metric:

$$d(A,B) = 1 - \frac{1 + \sum_{i=1}^{308} \min(A_i, B_i)}{1 + \sum_{i=1}^{308} \max(A_i, B_i)}, \quad (7)$$

where A and B represent two CVFH descriptors. This metric is not element-wise addivite, making it unsuitable for kd-tree search but suitable for hierarchical k-means indexing. At the moment, we are using linear search to retrieve the nearest neighbor since our database contains only 1704 CVFH descriptors for the 44 objects in our training set. The computation time for finding the nearest neighbor is below $2ms$ in our experiments, and using other search methods such as hierarchical indexing requires an addiotional overhead to construct the appropriate search structure, which is not necessary for linear search.

6. Results

For the evaluation of CVFH, we perform a different set of experiments and compare the results to VFH. First, we evaluate the performance on our training set for noise. We also evaluate the performance of both descriptors in matching single objects in real scenes obtained with the Kinect sensor. Finally, we show some scenes with the aligned models overlapped as a qualitative evaluation, see Figure 7.

The criteria we use to evaluate performance in synthetic data are multiple:

- Correct view and correct object id, respectively, in the first result.
- Correct view and correct object id, respectively, in the first N results.
- To test the performance of the camera's roll histogram, all views from the training set are randomly rotated along the virtual camera's roll axis. Because of discretization errors, we assume the result to be correct if the computed angle is off by 4° or less from the applied rotation.

6.1. Noise

Each view in the training set is noisified by applying a Gaussian kernel to each point. We use different standard deviations to test robustness to noise, ranging from 0.5mm to 2mm for each point in the view. After a view is noisified, we compute the CVFH and VFH and perform a search for the nearest neighbors in our descriptors database obtained from non-noisified views and compute the metrics listed above. Table. 1 and Table. 2 show respectively the results for VFH and for CVFH.

Table. 1 and Table. 2 show that with this kind of uniform noise and without missing parts VFH performs better than CVFH. Because in CVFH, the amount of points used for the computation of the centroid and the average of the normals which are used to build the Darboux coordinate system is usually smaller than in VFH, CVFH becomes more sensitive to this noise applied uniformly over the whole partial view. Another reason is that when the amount of noise increases, the estimation of stable regions becomes very unstable thus making the CVFH descriptor also unstable.

It is interesting to note that the roll orientation performs extremely well (over 90% with 1.5mm noise) when CVFH or VFH return the correct view.

	Noise levels (Stdev in mm)			
	0.5mm	1mm	1.5mm	2mm
View (1st)	99.32%	97.25%	92.76%	86.39%
View (N-1st)	100%	100%	99.93%	99.60%
Roll	98.32%	96.51%	93.89%	90.84%
Id (1st)	99.53%	97.98%	94.63%	89.87%
Id (N-1st)	100%	100%	100%	99.79%

Table 1. Recognition rates and roll angles correctness with different amount of noise applied on the training data using VFH.

	Noise levels (Stdev in mm)			
	0.5mm	1mm	1.5mm	2mm
View (1st)	93.89%	94.63%	86.92%	41.51%
View (N-1st)	97.38%	97.58%	93.62%	59.96%
Roll	97.10%	97.52%	94.48%	79.53%
Id (1st)	97.45%	97.38%	94.29%	58.62%
Id (N-1st)	99.53%	99.73%	99.46%	77.33%

Table 2. Recognition rates and roll angles correctness with different amount of noise applied on the training data using CVFH.

6.2. Recognition and pose evaluation on real scenes

We have performed recognition experiments on 18 of our 44 objects in the database to estimate the recognition rate of CVFH, VFH and CVFH + post-processing using Kinect data. Here, we refer to CVFH + post-processing as the steps 5) and 6) outlined in section 5.2. Because ground truth data for pose is not easily obtained, we decided to evaluate the recognition results manually.

To do so we have taken each of the 18 objects independently and placed them in the field of view of the sensor. The cluster of points representing the view of the object is extracted using Euclidean segmentation and recognized using CVFH, VFH and CVFH + post-processing to refine the recognition results. The recognition of the three pipelines are displayed together with the matching view in the database and the CAD model overlayed. All three recognitions include the computation of the roll orientation for a full 6DOF pose. We visually inspected the results and annotated independently for each 3 results set at which posi-

tion the correct object and pose is found. For each recognition, 14 nearest neighbors were retrieved. Each object was recognized 10 times in different stable poses.

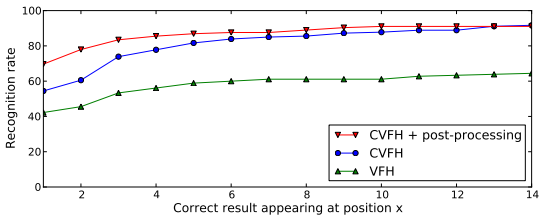


Figure 6. Recognition rate for CVFH, VFH and CVFH including the recognition framework. We show how often the correct solution appeared within the first x results.

Figure 6 shows the result of the experiment, where each point represents how many times we identified the right object with in the first x results. It can be seen that CVFH outperforms VFH both in recognition rate for the first result and for the accumulated recognition rate over the first 14 results. This is to show that although the recognition rate just by looking at the first result of $CVFH$ is below 60% we obtain the correct solution in the top-10 results in 90% of the cases. If we take into account the top-10 results of CVFH which include the right solution in 90% of the cases and sort these with post-processing, we increase the recognition rate in the first result to 70% of the cases. Ideally, we would like the recognition rate after post-processing to be 90% meaning that the post-processing can always identify the right solution if available in the candidates given by CVFH. In this case, for a desired recognition rate of 90%, the number of candidates is reduced using CVFH from 1409 (number of views) to 10.

7. Conclusions and future work

We have presented the Clustered Viewpoint Feature Histogram (CVFH) and shown that it can be robustly used to recognize objects and detect their poses in real scenes even when the training data source has different properties. In the scope of the paper, 44 objects were trained using CAD models and recognized in real scenes using the Kinect sensor.

Being able to determine a stable normal and a stable centroid on the objects allows us to deal with partial occlusions and handle properly the different properties of the training and recognition sensors. In our experiments we have shown that CVFH returns in 90% of the cases the correct view in the first 10 results reducing the number of candidates that need to be processed from approx. 1409 (number of views) to 10 in less than 2ms.

We have also presented the camera's roll histogram that can efficiently compute the rotation about the roll axis of the camera to which CVFH is invariant.

Future work includes dealing more robustly with higher degrees of clutter and occlusion, larger object databases and taking advantage of the semi-global nature of CVFH be able to solve undersegmentation issues.

References

[1] M. B. Ajmal S. Mian and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Ieee Transactions On Pattern Analysis And Machine Intelligence, Vol. 28, No. 10, October*, 2006. 1

[2] K. K. B. Steder, R. B. Rusu and W. Burgard. Narf: 3d range image features for object recognition. *In Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010. 1

[3] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen. Data-driven grasping with partial sensor data. 4

[4] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 1999. 1

[5] X. R. Kevin Lai, Liefeng Bo and D. Fox. A large-scale hierarchical multi-view rgb-d object. *IEEE International Conference on Robotics and Automation (ICRA)*, 2011. 1

[6] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 1

[7] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009. 5

[8] M. Muja, R. B. Rusu, G. Bradski, and D. G. Lowe. Rein - a fast, robust, scalable recognition infrastructure. In *ICRA 2011*, Shanghai, China, May 2011. 1, 2

[9] PrimeSense. Primesense teams up with asus to bring intuitive pc entertainment to the living room with wavi xtion. In *Business Wire*, 2011. 2

[10] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010. 2

[11] R. B. Rusu, A. Holzbach, M. Beetz, and G. Bradski. Detecting and segmenting objects for mobile manipulation. In *ICCV S3DV workshop*, 2009. 5

[12] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Learning Informative Point Classes for the Acquisition of Object Model Maps. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), Hanoi, Vietnam, December 17-20*, 2008. 1

[13] W. Wohlkinger and M. Vincze. 3d object classification for mobile robots in home-environments using web-data. *19th International Workshop on Robotics in Alpe-Adria-Danube Region RAAD*, 2010. 1

[14] Z. Zhang. Iterative point matching for registration of free-form curves. 1992. 5

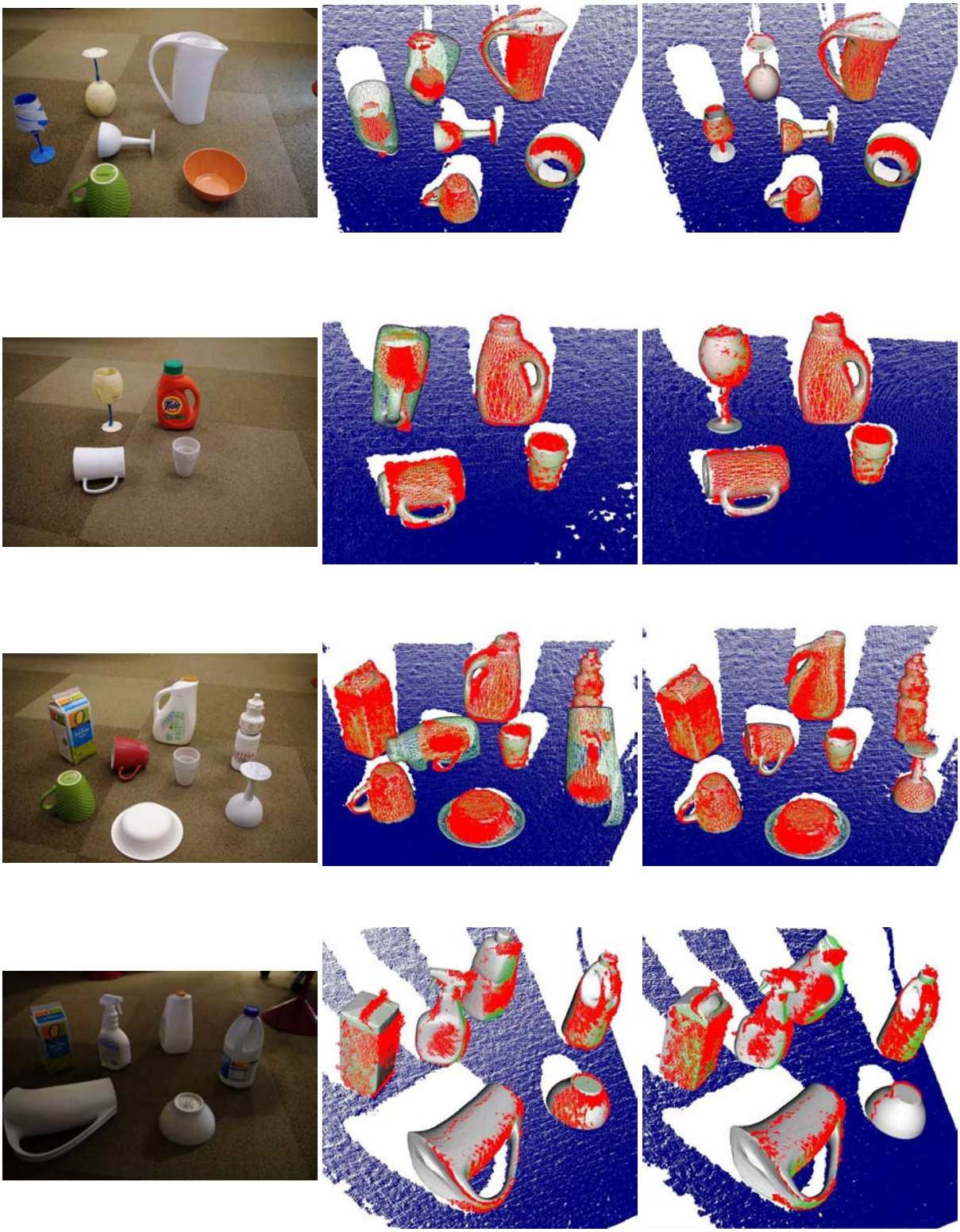


Figure 7. First column: Image of the scene, second column: results obtained using VFH and the third column using CVFH. Both VFH and CVFH results include the camera's roll histogram and the post-processing step to refine results.

Shape-Based Depth Image to 3D Model Matching and Classification with Inter-View Similarity

Walter Wohlkinger and Markus Vincze

***Abstract*— Object recognition and especially object class recognition is and will be a key capability in home robotics when robots have to tackle manipulation tasks and grasp new objects or just have to search for objects. The goal is to have a robot classify 'never before seen objects' at first occurrence in a single view in a fast and robust manner. The classification task can be seen as a matching problem, finding the most appropriate 3D model and view with respect to a given depth image. We introduce a single-view shape model based classification approach using RGB-D sensors and a novel matching procedure for depth image to 3D model matching leading inherently to object classification. Utilizing the inter-view similarity of the 3D models for enhanced matching, the average precision of our descriptors is increased of up to 15% resulting in high classification accuracy. The presented adaptation of 3D shape descriptors to 2.5D data enables us to calculate the features in real time, directly from the 3D points of the sensor, without any calculation of normals or generating a mesh from it which is typical of state-of-art methods. Furthermore, we introduce a semi-automatic, user-centric approach to utilize the Internet for acquiring the required training data in the form of 3D models which significantly reduces the time for teaching new categories.**

I. Introduction

For service robots to enter real-world home environments, they require adaptation to changing environments and knowledge transfer from one setting to another. One of the key elements for robots to fulfil meaningful tasks like object search and retrieval or object manipulation is object and object class recognition. Human robot interaction, robot localization and mapping, and robotic manipulation can greatly benefit from a vision system which is able to categorize even 'never before seen objects' at first glance.

The domestic setting with its plethora of categories and their huge intraclass variety demands a great deal of generalization skill from a service robot. These categories are characterized by their shape ranging from low intraclass diversification as in the case of fruits and simple objects like bottles up to high intraclass variety such as for liquid containers, furniture and especially toys. The scenario is aggravated by restrictions on traversability of task space or on the number of views that can be obtained of the object of interest in a given amount of time.

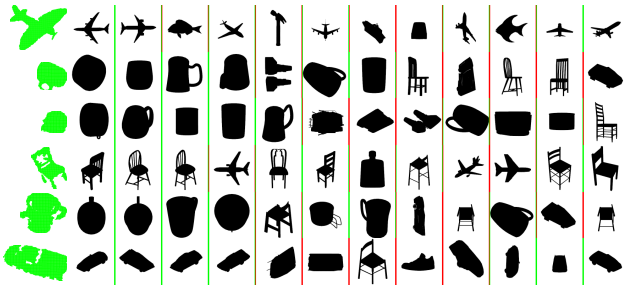
Our contribution consists of a 2-fold strategy to tackle the problems of learning new categories in a fast and semi-automatic manner and of reliable classification of objects from a single view.

This work was conducted within the EU Cognitive Systems project GRASP (FP7-215821) funded by the European Commission.

Vision4Robotics Group, Automation and Control Institute, Vienna University of Technology, Austria [ww,vm]@acin.tuwien.ac.at

(a) A mobile robot equipped with a Kinect sensor inspecting a table scene.

(b) A close-up of the scene with the segmented point cloud clusters including three mugs in different poses, a toy-car, a toy-plane and a toy-chair.



(c) Ranked list of matching 3D models, green representing correct category, red wrong category, best twelve matches.

Fig. 1. The overview of our system: The robot capturing the data using its RGB-D sensor, segmentation of the depth image and classification of the point cloud clusters by matching against a database of 3D models.

First, the robot is provided with access to 3D model repositories on the Internet to use the information found there to cope with the intraclass variation in classification. By using 3D models from Google Warehouse[1] the problem of coping with a large intraclass variety is inherently addressed, as the number of available models is found to be proportional to the intraclass variety, reducing the problem from classification to nearest neighbor matching.

Second, we use a single-view shape model based approach for depth image to 3D model matching to give the system its required speed. Our methodology works directly on the 3D data without a need for time-consuming and sensor noise dependent operations such as normals calculation and mesh-generation from the point clouds. As descriptors we present adaptations of three commonly used 3D descriptors to work in real time on depth images. For increased matching perfor-

[1] http://sketchup.google.com/3dwarehouse/

mance we suggest to utilize inter-view similarity of the 3D models to discard false positives. This new matching scheme can easily be adapted to work with other global, affine invariant 3D descriptor to also increase their performance.

The classification is performed with two frames per second against a database of 3D models which can be generated and altered semi-automatically by a non-expert. Robust classification is achieved in our distributed ROS[2]-based framework by choosing multiple complementary feature descriptors, selecting the appropriate similarity measures, using our proposed inter-view matching scheme and combining the descriptors. An illustration of our approach is presented in Figure 1. The source code for the descriptors is available online within the Point Cloud Library PCL [3].

II. Related Work

Automatically accessing a internet database with 3D models for robotics applications was presented by [10] where they use Shape Distributions [15] on the 3D models to discard wrongly downloaded models, do morphing to increase the number of models and use 2D contours for image based object recognition. Spin Images are used for a 3D to range image matching on 3D LIDAR point clouds by [5] and by [11] who also uses 3D models from the web to match against. Range image to 3D model matching for robotic grasping was done by [4] using a dense SIFT [12] based descriptor first introduced by [14] which is the top performer in the SHREC shape retrieval contest of range images[4] [3]. The authors of[4] achieve promising results and improved the matching by extending the visible area of the objects with a movable sensor head. Object categorization from multiple views using a humanoid robot by was demonstrated in [6]. Spin Images [8], D2 Shape Distribution [15] and geometric properties like bounding box and volume of the real-world sized 3D model were used for categorization, thus requiring acquisition of a specialized database for training with a structured light sensor. A global descriptor based on histograms of normals was introduced by [16] which delivers excellent results on range scans for container-like objects, but requires calculation of normals. The intuitive idea that objects are similar if they also look similar from different view points was stated by [18] who introduced the so called light field descriptor for 3D model matching, which uses projections from views around the model and encodes the images by Zernike moments and Fourier descriptors. Although this descriptor works only on images, it is the top performer on the Princeton Shape Benchmark. Adaptive view clustering to reduce the views around the model to a optimal set of distinct views was shown in [1]. Learning viewpoint detection models for viewpoint planning was introduced by [13] where they used the system for finding the next view with most information gain for the robot. Our approach adapts the ideas of [18], [1] and [13] for improving matching performance when matching one view of a model to similar models where multiple views around the object are available. The idea is to use 3D descriptors which are capable of interpolating between neighbouring views. Having multiple descriptors for the full 3D model, there exists more than one view matching to the depth image coming from the sensor. We use this characteristic for pruning wrong matches and improve matching.

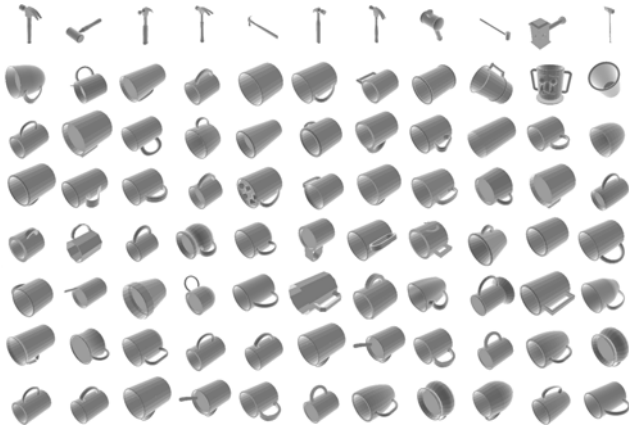


Fig. 2. Models acquired from Google's 3D Warehouse. There is no common coordinate system among the models or a common scale, but most of the mugs share a main orientation. The intraclass variety of a common class like 'mug' can easily be handled given this large number of examples for this class.

III. Methodology

The classification is based on matching depth images against a database of 3D models and a subsequent probabilistic voting scheme. The stages of the system include the acquisition of the database, object segmentation and matching against the database. model database acquisition from internet, 3D object segmentation, calculation of the descriptors and matching, calculation of the descriptor confidences and a final voting stage.

A. Knowledge Acquisition & Model Preparation

The input into our model acquisition system is the name of the new object class, which can be entered by the user via voice or via keyboard. With this keyword, we query the lexical database WordNet[5] to disambiguate the keyword by presenting the different meanings to the user to select the appropriate one. Once the correct meaning of the keyword is known, the synonyms and hyponyms (words sharing a 'type-of' relationship with the keyword) provided by WordNet are used for the 3D model search on 3D Warehouse[6]. After download of the models, the user selects one of the models as the reference model to enable a subsequent process of discarding wrong models from the database using a similarity criterion to the reference model. Shape Distributions is used as our similarity descriptor. A set of models is shown in Figure 2. Having a semantic meaning and an index for the word in the hierarchy provided by WordNet enables further

[2]http://www.ros.org

[3]http://pointclouds.org

[4]http://www.itl.nist.gov/iad/vug/sharp/contest/2010/RangeScans/

[5]http://wordnet.princeton.edu/

[6]http://sketchup.google.com/3dwarehouse/

semantic meaningful manipulation applications like pouring something into a container-like object.

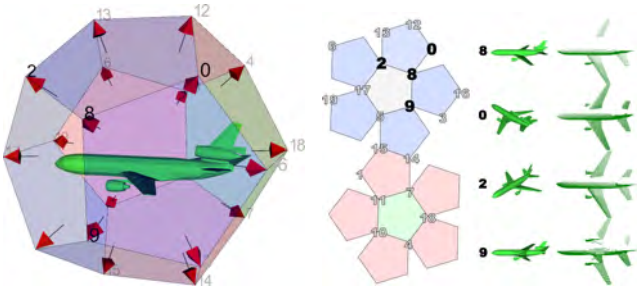
One way of matching depth images to full 3D models is to equate the problem to finding the appropriate view of the 3D model. This can be achieved by formulating the problem as a *partial-view to partial-view* matching problem. To use the models from the web for depth image to depth image matching, we generate synthetic depth images by rendering the 3D models and sampling the z-buffer from 20 equally spaced views around the model using the vertices of a dodecahedron as done in the light field descriptor [18] and depicted in Figure 4. These 20 views are sufficiently dense for the type of descriptors used to interpolate between views. To discard details and therefore improve generalization of the models, we sample the models by rendering them in 150x150 pixel images which leads to around 5000 data points for the typical model. Finally, for each of the 20 views of the model the 3D descriptors are calculated and stored in the database. The best partial-view out of the 3D models can be found by comparing the descriptors calculated from the depth image delivered by the sensor to all descriptors in the database. As our approach is orientation and scale invariant, we can detect objects in any pose and and any size, making no distinction between toy-chairs and real-sized chairs.

B. Matching with Inter-View Similarities

The basic idea is to match the depth image not only to one single view of the 3D model, but to several views as nearby views share some similarity. This approach is grounded in the similarity matrices in Figure 3(a) to Figure 3(c) which indicate that there are views of an object that are similar. The number of these similar views depends on the shape complexity and symmetry of the object which can be clearly seen on these three example models and their similarity matrices. Not only neighboring views share some similarity but also opposing views. The similarity across the views depends on the type of descriptor used, but for rotational invariant 3D shape descriptors the difference to similar views is less dramatic than with 2D descriptors on the viewpoint images as depicted in Figure 4(b). Using the constraint that there have to be multiple matching views the false positive rate can significantly be reduced. After evaluation on a synthetic database, the matching constraint for our system was set to three views out of the twenty views generated by the dodecahedron. Figure 7 in the experimental evaluation section depicts the improvement of using inter-view similarities with three views over matching against a single view.

C. Shape Descriptors

To find the similarity between two depth images, descriptors are calculated from the data and compared against each other. Our requirements to a descriptor are its computational efficiency, its size to store and its affine invariance as we only have a fixed set of views and the descriptor has to interpolate between the views. The use of multiple descriptors leads to an increased recognition rate as



(a) Views around an object are equally arranged at the vertices of a dodecahedron.

(b) The unfolded dodecahedron with images and depth images taken at four neighbouring views. Despite the viewpoint change, the range images (right) share more similarity than the 2D images(mid column).

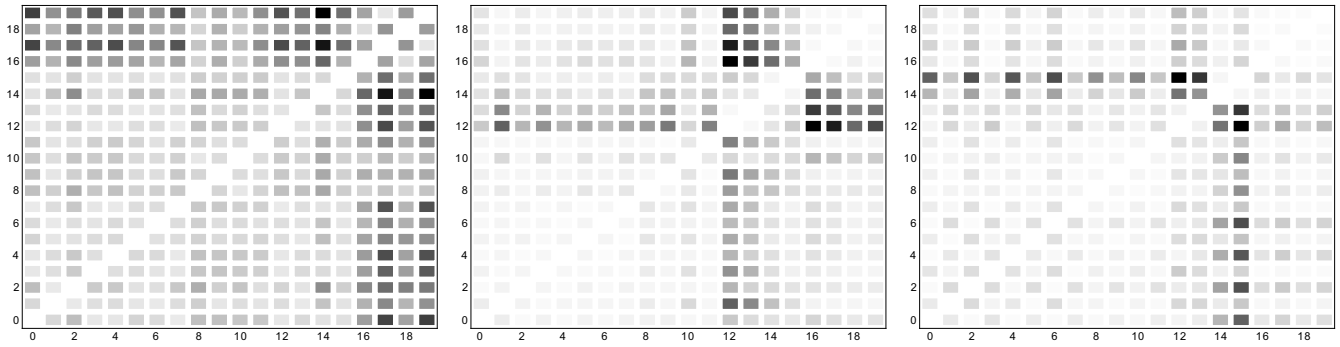
Fig. 4. A 3D model of the category "commercial plane" with the 20 viewpoints enumerated at the vertices of a dodecahedron. The roll of the cameras around the view-axis is arbitrarily set, as the descriptors are rotationally invariant.

the performance of descriptors varies with category. When carefully chosen, the calculation overhead can be kept to a minimum by memoizing intermediate results. We utilize the shape distributions [15], moment invariants [17] and spherical harmonics [9] as descriptors as presented in the next sections.

1) D2 Shape Distribution: We use a multi-resolution version of the D2 shape distribution descriptor of [15] who introduced this descriptor for full 3D model matching and was also utilized in [6]. The advantage of this descriptor is that the histogram of distances between randomly sampled points can be calculated directly from the point cloud. To capture coarse structures and fine details, the best bin-size of the distance histogram has to be chosen. We avoid this by combining multiple bin resolutions into one histogram. Figure 5 depicts the multi-resolution D2 shape distribution histogram for three classes with 32, 64, 128 and 256 bins, normalized inside each sub-histogram. The varying characteristics are clearly visible for these three classes, but one can imagine how the discrepancy diminishes when the number of classes is vastly increased, justifying the use of multiple descriptors with orthogonal/opposed characteristics. The performance of this descriptor on depth images is obviously inferior to the performance on full 3D models, as the distances now capture less than half of the object but is good enough to distinguish among most classes and used in conjunction with other descriptors. As our choice of distance measure we use the Taneja [2] similarity measure given in (1) with P and Q the histograms to be compared, d the bin size, and d_T the Taneja distance. Taneja performes best for this descriptor across all classes in our evaluation of all the similarity measures given in [2].

$$d_T = \sum_{i=1}^{d} \left(\frac{P_i + Q_i}{2} \right) \ln \left(\frac{P_i + Qi}{2\sqrt{P_i Q_i}} \right) \qquad (1)$$

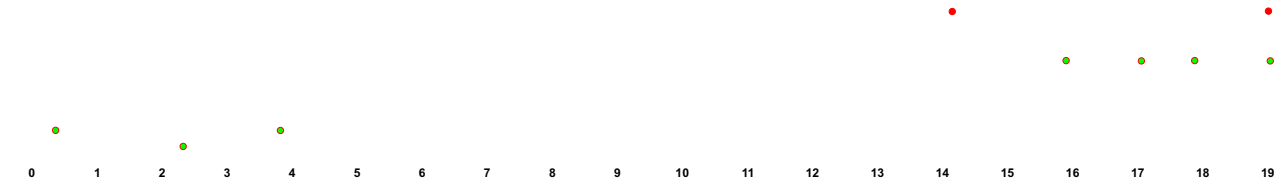
2) Moment Invariants: For a coarse classification we use the 3D moment invariants presented in [17] which are the



(a) Air plane view similarity matrix with white showing similar views, black dissimilarity.

(b) Chair view similarity matrix with self similarity on the diagonal.

(c) Mug view similarity matrix. Views 0, 2 and 4 are very similar.



(d) The views of the objects from view 0 to view 19 depicted as images. Numbering as introduced in Figure 4. Green dots mark similar views, red dots dissimilar views for better illustration (see similarity matrix).

Fig. 3. The similarity matrix of the views around a dodecahedron for three models (descriptor similarity on depth image). Depending on the symmetry of the object, there are clearly some similar views per model noticeable.

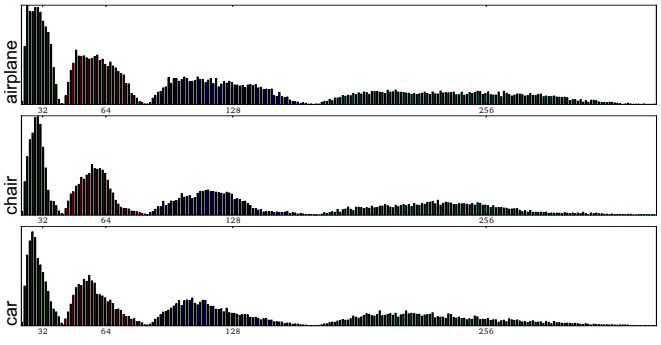


Fig. 5. The multi-resolution shape distribution histograms for the classes plane, chair and car with bin sizes 32(green), 64(red), 128(blue) and 256(cyan) combined into a single descriptor.

3D equivalent to the 2D Hu Moment Invariants [7]. The invariants are calculated directly from the point cloud and the time necessary for computing is negligible. The invariants are stacked into a single vector and the similarity measure of choice for this descriptor is Wave Hedges [2] which is given in (2) and was chosen as best performing similarity measure after empirically evaluation. P and Q are the vectors to be compared, d the size of the vector, and d_W the Wave Hedges distance.

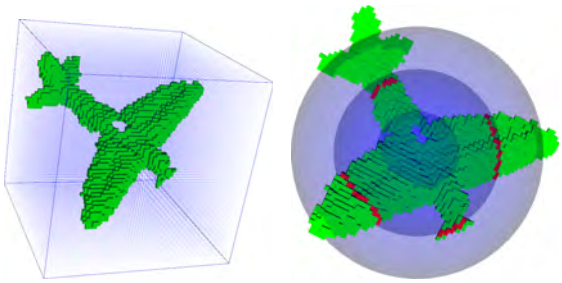
$$d_W = \sum_{i=1}^{d} \frac{|P_i - Q_i|}{\max(P_i, Q_i)} \tag{2}$$

3) Voxel based Spherical Harmonics: This descriptor is among the best performing 3D descriptors for full 3D models on the Princeton Shape Benchmark [19]. Despite its computational expensive formulation, this descriptor can be adopted to function in a real-time robotics environment. To compute this descriptor the point cloud has to be scaled to fit into a cube with side length 64 and is then converted into a voxel representation as depicted in Figure 6(a). The spherical harmonics representation is then calculated for 32 concentric spheres and 32 frequency bands. Figure 6(b) shows three of the 32 concentric spheres with the voxels marked red falling into the sphere with radius 20. Calculation of the descriptor is done by evaluating the spherical harmonics function for each voxel and building up the 32x32 histogram. Using a fixed sized voxel grid to work with, all the computational expensive calculations for each voxel can be done offline and stored in a look-up-table, resulting in fast descriptor calculation at run-time which only consists of iterating over the voxels and creating the histogram with the aid of the LUT. This enables the descriptor to be computed in a fixed amount of time. For the resulting 32 by 32 histogram we use KDivergence [2] as the similarity measure given in (3) with P and Q the two histograms, d the histogram size, and $d_K div$ the K divergence which is similar to Kullback Leibler divergence but gave slightly better results in our evaluation.

$$d_{Kdiv} = \sum_{i=1}^{d} P_i \ln \frac{2P_i}{P_i + Q_i} \tag{3}$$

D. Classification

Our framework conveys the use of multiple descriptors to increase classification performance by using a voting scheme. In order to combine descriptors with different characteristics and weight them accordingly, we introduce two confidence measures. The first confidence measure is a bias calculated for each descriptor and each class offline on the database



(a) Aircraft in voxel representation in a 64-cube. (b) Spherical harmonics are calculated on concentric spheres.

Fig. 6. Schematics of the preprocessing and calculation of the spherical harmonics descriptor. Slight viewpoint variation can be interpolated with this descriptor.

of 3D models with the reference model each time a new category is added. The second confidence measure is calculated online for each descriptor and each query by calculating First Tier and Second Tier and calculating the ratio of best guess to second best guess. This enables us to combine the results of descriptors running in a distributed system without the need to know the other detectors/descriptors running in the system. Calculation of the descriptors takes less than 30 ms on a 2 GHz dual-core laptop. For better scalability to a higher number of models, we use a hashing approach which enables us to perform classification in less than one second, regardless the number of models we have to match against.

IV. Experimental Evaluation

A. Matching Impact

We demonstrate the performance increase for independent descriptors with a sample query on the Princeton Shape Benchmark to clearly single out the advantage of using our proposed matching scheme. Figure 7 depicts the average precision (AP) for our three descriptors on a single query with a range scan of a plane model into the database consisting of 20 categories. The green curve shows the increased matching performance utilizing inter-view similarity to matching with a single view, shown in red. The performance increases up to 15%, false positives are decreased in general and true positives are pushed forward in the ranked list.

B. Classification with Kinect Data

We introduce a new database for testing object classification acquired with a RGB-D camera. The database provides tools for capturing scenes from a Kinect sensor or a stereo camera, annotation of the scenes with bounding boxes and labels and to replay selected scenes including ground truth to ease testing. The database together with the tools is available as a ROS-package at our repository (svn.acin.tuwien.ac.at/ros). The database consists of 774 scenes of single objects on a flat surface as the purpose of this database is classification rather than segmentation.

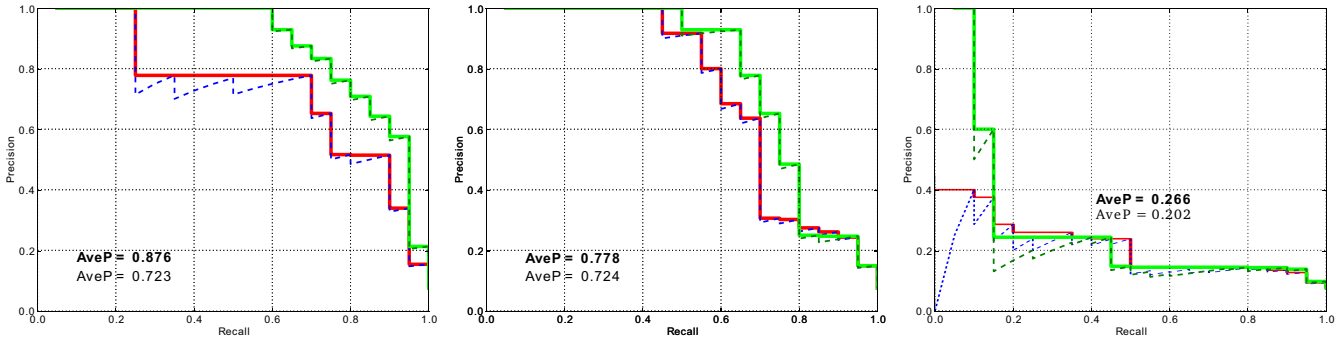
The 20 categories for testing were taken from the Princeton Shape Benchmark rather than from the web to have a fixed set of models for better comparability to other approaches. The categories used for this test include hammer, mug, bottle, sedan(car), shoe, commercial(airplane), biplane, fish, knife, dining chair, helicopter, ship, shovel, jeep, couch, screwdriver, wrench, military tank, handgun and hourglass.

Fig. 8. The objects in the V4R2011 database: Each object is captured with a Kinect sensor on a turntable with multiple views around the object.

Class	Views	Classification Rate
hammer	37	76%
mug	174	86%
airplane	59	85%
bottle	14	64%
car	24	75%
shoe	41	68%

We achieve a high overall classification rate matching against 20 categories. Open challenges are depicted in Figure 9, where missing data due to material properties lead to false segmentation and classification which can be seen on the right for the hammer and (glass) bottle. If there is model in the database resembling the query object, the system picks the most similar model, which in case of the closed mug in Figure 9 is a car model.

Fig. 9. Positive examples are shown on the left side, failed classifications are shown on the right side with the point cloud colored in red.



(a) Improved AP (green) with inter-view matching over single view (red) with SH.

(b) Improved AP (green) with inter-view matching over single view (red) with D2.

(c) Improved AP (green) with inter-view matching over single view (red) with MI.

(d) ranked list for single view matching with moment invariants

(e) ranked list for multiple views matching with moment invariants

Fig. 7. Improvement using inter-view matching. One important fact is the improved ordering of the results: More correct matches are at the beginning of the ranked list, noticeable in the right-shifting of the green curve.

V. Conclusion

In this paper we investigated the use of web-learned models to detect object classes in depth images from actual scenes. The intention was to use the object class relations to derive grasp points for the respective objects. We implemented a scheme to learn view-based 3D models given the web data. These models can be used for matching with the depth data provided by a state-of-the-art RGB-D sensor such as the PrimeSense sensor. The results clearly indicate that the mixture of features used to describe the object models achieve high recognition rates. We further showed that by using multiple views of the 3D models at the matching stage, the performance of the individual descriptors and of the whole system can be considerably improved.

The advantage of this approach is that new object class models can be very efficiently learned from web data and that matching is robust and fast using the depth images. Future work comprises the investigation of more and alternative features and a deeper analysis of the cases where pure matching of 3D data is misleading and should be complemented by adding appearance data, 2D features and scale to the object class models. As these results in this framework are achieved without incorporating the specific sensor modalities, the performance could be boosted by adapting the synthetic views to the sensor characteristics.

References

[1] Tarik Filali Ansary, Mohamed Daoudi, and Jean-Phillipe Vandeborre. 3d model retrieval based on adaptive views clustering. In *International Conference on Advances in Pattern Recognition (ICAPR)*, 2005.

[2] Sung-Hyuk Cha. Taxonomy of nominal type histogram distance measures. In *Proceedings of the American Conference on Applied Mathematics*, pages 325–330, Stevens Point, Wisconsin, USA, 2008.

[3] H. Dutagaci, A. Godil, C. P. Cheung, T. Furuya, U. Hillenbrand, and R. Ohbuchi. Shrec 2010 - shape retrieval contest of range scans. In *Eurographics Workshop on 3D Object Retrieval*, 2010.

[4] C. Goldfeder, M. Ciocarlie, J. Peretzman, Hao Dang, and P.K. Allen. Data-driven grasping with partial sensor data. In *International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[5] Aleksey Golovinskiy, Vladimir G. Kim, and Thomas Funkhouser. Shape-based recognition of 3d point clouds in urban environments. *International Conference on Computer Vision (ICCV)*, 2009.

[6] D. Gonzalez-Aguirre, J. Hoch, S. Roehl, T. Asfour, E. Bayro-Corrochano, and R. Dillmann. Towards shape-based visual object categorization for humanoid robots. In *International Conference on Robotics and Automation (ICRA)*, 2011.

[7] Ming-Kuei Hu. Visual pattern recognition by moment invariants. In *IEEE Transactions on Information Theory*, 1962.

[8] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligences*, 21(5):433–449, 1999.

[9] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. *SGP*, pages 156–164, 2003.

[10] Ulrich Klank, Muhammad Zeeshan Zia, and Michael Beetz. 3d model selection from an internet database for robotic vision. In *International Conference on Robotics and Automation (ICRA)*, 2009.

[11] Kevin Lai and Dieter Fox. Object detection in 3d point clouds using web data and domain adaptation. *International Journal of Robotics Research*, 2010.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[13] David Meger, Ankur Gupta, and James J. Little. Viewpoint detection models for sequential embodied object category recognition. *International Conference on Robotics and Automation (ICRA)*, 2010.

[14] R. Ohbuchi and T. Furuya. Scale-weighted dense bag of visual features for 3d model retrieval from a partial view 3d model. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 63 –70, 2009.

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 154 –166, May 2001.

[16] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2155 –2162, 2010.

[17] Firooz A. Sadjadi and Ernest L. Hall. Three-dimensional moment invariants. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-2(2):127 –136, 1980.

[18] Yu-Te Shen, Ding-Yun Chen, Xiao-Pei Tian, and Ming Ouhyoung. 3d model search engine based on lightfield descriptors. In *Eurographics*, 2003.

[19] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Shape Modeling International, Genova, Italy*, 2004.

3DNet: Large-Scale Object Class Recognition from CAD Models

Walter Wohlkinger and Aitor Aldoma and Radu B. Rusu and Markus Vincze

***Abstract*— 3D object and object class recognition gained momentum with the arrival of low-cost RGB-D sensors and enables robotics tasks not feasible years ago. Scaling object class recognition to hundreds of classes still requires extensive time and many objects for learning. To overcome the training issue, we introduce a methodology for learning 3D descriptors from synthetic CAD-models and classification of never-before-seen objects at the first glance, where classification rates and speed are suited for robotics tasks. We provide this in 3DNet (www.3d-net.org), a free resource for object class recognition and pose estimation from point cloud data. 3DNet provides a large-scale hierarchical CAD-model databases with increasing numbers of classes and difficulty with 10, 50, 100 and 200 object classes together with evaluation datasets that contain thousands of scenes captured with a RGB-D sensor. 3DNet further provides an open-source framework based on PCL for testing new descriptors and benchmarking of state-of-the-art descriptors together with pose estimation procedures to enable robotics tasks such as search and grasping.**

I. INTRODUCTION

Central tasks for robots are to find, grasp and manipulate objects. While an industrial robot helper needs to know about the specific objects in production, home robots should know about all the object classes typically found in human living space. And certainly, the user expects that the robot can learn novel objects and object classes.

Especially the domestic setting with its plethora of categories and their intraclass variety demands great generalization skills from a service robot. The categories are characterized mostly by their shape ranging from low intraclass diversification as in the case of fruits and simple objects like bottles up to high intraclass variety of classes such as liquid containers, furniture, and especially toys. With robots starting to tackle real-word scenarios, we require fast and reliable object and object class recognition. Especially in robotics manipulation, where object recognition and object classification have to work from all possible viewpoints of an object, data collection for training becomes a bottleneck. Especially for classes with high intraclass variability it is required to obtain a very large number of objects in the training phase.

With the arrival of an affordable RGB-D sensor, the Kinect, and the increasing number of mobile manipulators, e.g., WillowGarage's PR2, learning classes and objects for each one of the objects and robots seems like a waste of resources. The goal should be to have a common knowledge database shared between the robots. So when one robot in place A is trained on novel objects or classes, another robot at place B can update the reference database and detect the new object classes (maybe with the exception if objects differing too greatly from country to country). This also holds true for the introduction of new features and descriptors: once introduced and integrated, everyone should be able to use these algorithms. This is especially true for researchers not working in the field of object and object class recognition, as for them, classification is a necessary step to achieve their own research to provide the robot with new functionalities.

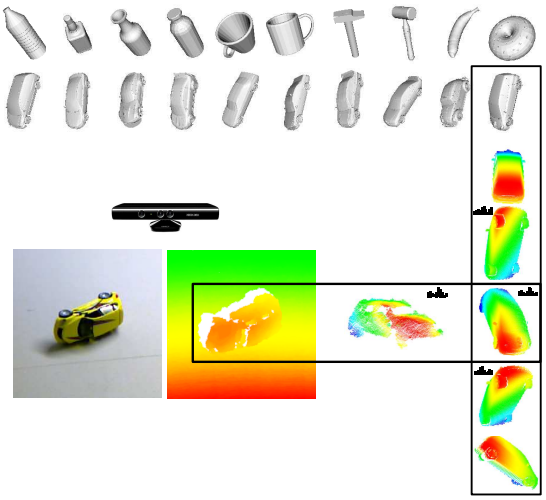


Fig. 1. System overview: For classification, the RGB-D image is segmented to obtain a point cloud cluster and to calculate a cluster descriptor. The descriptor is compared to synthetically rendered views of CAD-models downloaded from the web to model object classes. The most similar view delivers the best 3D model and class label.

To gain momentum towards that vision, we introduce 3DNet (www.3d-net.org), a free resource providing training data in the form of 3D CAD models, a framework for implementing and evaluating existing and new 3D shape descriptors, and out-of-the-box object recognition and object classification, see Fig. 1. We encourage the community to exploit and add to this open framework, which presents state-of-the-art performance compared to Lai [9], see SectionV. beginning and the features and descriptor possibilities not fully exploited yet. Our contributions encompass three distinct but related areas:

- First, we propose to use synthetic CAD models from the web that are organized according to WordNet and are provided through 3DNet (www.3dnet.org). This enables the robotic community to fast and easily train many

This work was conducted within the EU Cognitive Systems project GRASP (FP7-215821) funded by the European Commission.

Wohlkinger, Aldoma and Vincze are with Vision4Robotics Group, Automation and Control Institute, Vienna University of Technology Vienna, Austria [ww,aa,mv] @ acin.tuwien.ac.at

Rusu is with Willow Garage, Menlo Park, USA rusu @ willowgarage.com

object and object class recognition algorithms without tedious object scanning.
- Secondly, we provide an open-source framework based on PCL[15] with state of the art descriptors for use with the 3D model database. These descriptors are automatically trained on the 3d models and enable real-time, high performance object and object class recognition to be integrated into common robotics frameworks such as ROS[13]. The framework provides templates to easily integrate new descriptors.
- And Third, we propose benchmarks with increasing complexity to be used as test environment to enable objective comparison of descriptor performance. The benchmark datasets are in addition to the RGB-D dataset by Lai[10] and the SHREC Range Image Retrieval Contest[4] where we provide out-of-the-box evaluation scripts to be tested on these already available datasets. To provide an unbiased test dataset for our 200 category model database, we start to collect datasets from the community via 3DNet.

After reviewing related work we present the 3DNet database in in Section III and the classification framework in Section IV and present benchmarks and evaluation results in Section V.

II. State Of the Art

Our reference is the hierarchical RGB-D object test dataset that was made available to the community by Lai [9]. It presents 51 object classes also organised according to WordNet relations. In the accompanied approach [10] multiple features are combined, trained, and evaluated on this dataset and the authors showed that shape together with color leads to improved object recognition. Although the authors collected a large dataset from multiple viewpoints, the authors did not make the code nor evaluation tools available to the community. The KIT Object Models Web Database[1] is also a free resource of 3D models with texture scanned with a structured light setup representing mostly household items. The closest benchmark to our system is the SHREC Shape Retrieval Contest of Range Scans[2] where a set of 800 3D models in 40 classes is given as target set and 120 range scans captured with a Minolta Laser Scanner and converted to meshes are given as query set. The results were presented in [4] where the top performer reached a nearest neighbor classification rate of 67.5% with a bag of words approach with of depth-sift features.

Regarding the development of datasets, an interesting issue was brought up by Torralba and Efros [18]: datasets (e.g., Caltech-101 or the Pascal VOC) for measuring and comparing competing algorithms are biased. This also halts true for the RGB-D dataset of [9], which has a selected set of objects, poses, lighting conditions and objects on a small turntable. The authors of [18] provide suggestions to minimize the bias in datasets which include:

- Selection Bias: to avoid a bias towards human-selected images, data should be collected automatically from multiple sources, using multiple search engines from multiple countries, or use a large set of not annotated images and label them by crowd-sourcing as done with ImageNet[3].
- Capture Bias: as objects almost always appear in the center of the image and objects tend to have a standard position (mugs upright with handle to the right). In a robotic-centered RGB-D context, the capture bias could be resolved by capturing failed manipulation attempts which lead to objects in random pose and distance to the camera and thus avoiding human-biased viewpoints (e.g., looking down 45 degrees).
- Negative Set Bias: is reduced if we add scenes to the database that do not contain any of the database objects.

These suggestions motivated us to create a publicly available community-built test dataset for the unbiased, objective and extendible comparison of classification and recognition algorithms for robotics.

III. DATABASE

The intention is to build up and maintain a steadily growing database of object classes for robotic applications. We propose to adopt the paradigm of learning models of classes from the web to easily capture intra-class variability and simplify data gathering. And we link the classes to actual scenes with (new) samples of these object classes.

As start we provide four CAD-model databases with increasing size and complexity accompanied with corresponding test databases. The model databases are constructed by semi-automatically downloading models from Google's 3D Warehouse and various smaller, free online repositories for CAD models[3]. The models are linked to the WordNet [5] structure, which provides a hierarchical semantic organization of the classes. The idea to use 3D models from the web has an additional advantage: By using 3D models the problem of coping with a large intraclass variety is inherently addressed, as the number of available models is found to be proportional to the intraclass variety.

The classes are organized in four increasingly challenging datasets as more sophisticated descriptors and additional cues are necessary to differentiate between 200 classes in the largest dataset. The test-databases contain only a single object per scene. Segmentation is provided as a preprocessing step in the framework. The datasets are introduced in the following sections.

A. Cat10: Basic Object Classes

The basic dataset consists of common, simple, geometrically distinguishable but partially similar objects. Object classes were chosen to also be suitable for robotic manipulation. The database consists of 360 3D CAD models in the classes apple, banana, bottle, bowl, car, doughnut, hammer, mug, tetra-pak and toilet-paper. The test-database consists of

[1] http://i61p109.ira.uka.de/ObjectModelsWebUI/

[2] http://www.itl.nist.gov/iad/vug/sharp/contest/2010/RangeScans/

[3] www.123dapp.com, turbosquid uvm.

TABLE I

HIERARCHICAL ORGANIZATION OF THE MODELS IN CAT50.

coarse categories (hypernyms)	shape categories (hyponym)
animal	camel, cow, dinosaur, elephant horse, shark
musical instrument	banjo, guitar
container	bottle, can, mug, tetra pack
edible fruit	apple, banana, lemon, pear starfruit, pineapple, strawberry
motor vehicle	car, convertible, locomotive monster truck, pickup, race car, suv tank, truck
food	donut, pretzel, croissant
aircraft	airplane, biplane, fighter jet, helicopter
seat	armchair, chair, office chair, stool
footwear	boot, sandals, shoe, heels, ski boot,
hand tool	hammer, pliers, screwdriver, wrench

1600 scenes of single objects on a flat surface in multiple poses and multiple instances per class. For each scene a color image, a point cloud and a bounding box with the class label is provided. In Figure 3, a representative sample of the Cat10 model and test database is given.

The challenges in these classes are twofold: Firstly the intra-class variance of the classes hammer, mug and bottle, as these three classes are to be found in hundreds of shape variations in the real world. Secondly, the inter-class similarity of the classes (mug,toilet paper),(apple,donut) and (bottle,banana,car) when given only a partial view as depicted in Figure2.

B. Cat50: Super-Classes

The Cat50 model database consists of the Cat10 database with forty additional classes. The classes in this database are still distinguishable by shape only, but also include sub-categories (chair, office-chair, armchair and car, convertible, pickup, formula-car). Table I gives an overview of the classes sharing the same hypernym, i.e., belonging to the same superclass.

From the point of view of object classification, organizing objects in a tree has an implicit advantage regarding evaluation: The level of misclassification of an object can be measured as the length between the nodes in the tree. Clearly, misclassification inside a subtree – convertible as car, or airplane as fighter jet) – is better than outside a subtree, especially when robustness and user acceptance is of importance as in home robotics.

The according test database for the Cat50MDB adds another 1600 scenes which adds up to 3200 test scenes for the 50 categories.

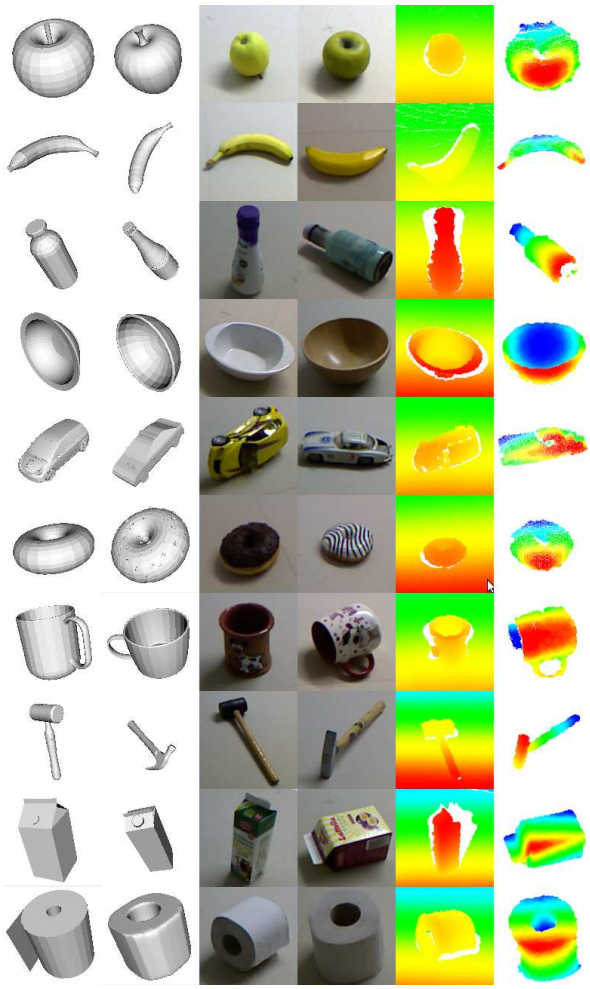


Fig. 3. CAD models of the ten classes with selected test scenes. First two columns present two typical cad-models from the according class followed by two object instances from the testset with the whole scene and segmented scene in point cloud representation.

The challenges in this database include coping with large shape differences although from the same class (paper airplane test object to real model airplanes), similar objects from super-classes and accidentally matching views – as already present in the Cat10 database – as a direct result of scaling the number of CAD models to (exact number here). Example views of the challenges are depicted in Figure 4.

This database adds objects which are similar in shape but can be uniquely distinguished when using color as an additional cue. As stated in the work of Lai [10], color together with shape leads to improved recognition of objects and object classes. As depicted in Figure 5, color is not only improving object class recognition, in these one hundred

Fig. 2. Similar partial views of the classes mug vs. toilet paper and and donut vs. apple

Fig. 4. Challenges when matching real objects like inflatable and paper airplanes to CAD models of planes which only share overall shape.

object classes it is crucial to have color as an additional cue to differentiate between the newly added classes. The database now contains many natural objects like fruits and vegetables, which share a common primitive shape such as orange, apple, lemon, lime, watermelon, carrot-radish, etc.

Fig. 5. Some classes are almost identical in shape but differ in color. Lemon and lime are two obvious examples, but most roundish shaped fruits and vegetables having color as distinct cue.

Man made objects are largely excluded from adding to this database, as color can not be assumed fixed, even with common objects such as a tennis ball for example, as it comes in additional colors to the standard yellow.

C. Cat200: Size

One important aspect of objects and object classes was not used and not needed in the previous category databases: size. To successfully distinguish among our 200 categories database, the real world size of the objects becomes important. As classification of objects is subjective – assume a tennis ball with 30 cm diameter, is it still a tennis ball? – we advocate a functional viewpoint on classification: If the object affords the intended function, it is part of the class, otherwise it is a new class, e.g. toy-tennis ball. Following this schematic, a huge part of man-made objects depends on the size cue, e.g., example in Figure 6.

Fig. 6. Size matters: Depending on the real size of the depicted object it can be waste-bin, a mug or a thimble. Shape, color and texture are not sufficient any more to classify this object, which is a thimble.

Real world scale information is not yet present in the database, as CAD models do not come with a common real word size information and therefore it has to be acquired from other resources on the web or learned by the system during successful detections of objects.

D. Community-built Test Database

The test database for these two hundred object classes are open to be extended by the community to provide a large unbiased test database. The test database will be fixed once a minimum of five test objects per category are available for evaluation in the database. We provide tools and web-space for uploading test scenes to 3DNet. A test scene is defined as binary pcd file including X,Y,Z,RGB values captured with a Kinect-like sensor. Capturing can be done using standard PCL tools or our provided ROS-based capturing and annotation tools. To ease segmentation, objects have to be on a flat surface, e.g. on the floor. Annotation is done by 3DNet according to the classes available. For the follow-up database Cat300, classes can be requested by the community.

IV. FRAMEWORK

The proposed open-source PCL-based framework targets real-time classification and object instance 6D0F pose recognition for robotics and provides an easy way of training descriptors, adding new classes or specific objects. Adding new descriptors is supported and encouraged by providing code-templates for an easy transition of C++ code into the framework. Evaluation and benchmarking are also part of the framework, as is 6D0F pose estimation and object recognition.

Usage of the proposed framework for object classification requires the following steps:

1) SVN check-out framework provided on 3DNet
2) Download CAD models from 3DNet
3) Download test database from 3DNet
4) Use present descriptor or implement own using provided template
5) Run the program to fully automatically train on the CAD models and evaluate on test set
6) Plug in a Kinect and classify objects

A. View Generation

The training on CAD models is done by rendering and sampling the z-buffer from views around the model and storing the generated partial views as point clouds. Descriptors are computed on these partial views. The number of views can be chosen from as few as 12 to several hundreds, depending on the descriptor and application in mind. The standard number of views used for the experiments in this paper is 80, as this number provides sufficient views even for complex objects.

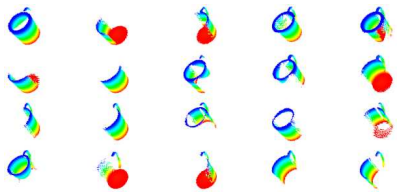


Fig. 7. Partial views of a mug generated by sampling the depth-buffer while rendering views around the object.

B. Entropy

Having synthetic views and the original model at hand enables the calculation of the entropy of each view i.e. the expected value of the information contained in a view. This follows the idea of using the different levels of information in views as shown in [2], where an optimal set of views(images) of a 3D model is found by adaptive clustering.

These entropy values for each view can be used in a post-processing step to filter accidental views: Given a model of a bottle, the view directly from the bottom only represents

a small portion of the object and thus has a low entropy value assigned. If this view is matched against something round and curved like an apple or donut, it can be filtered as real world scenes are rarely represent such extremal views of objects.

The entropy is calculated as the ratio of the surface area of the whole model and the visible surface area. Experimental evaluation of view filtering the nearest neighbor list is given in SectionV. Another available post-processing step for filtering is available in the framework using the approach proposed in [20], where the similarity of nearby views is used to filter accidental matches.

C. Pose Estimation

Given real-scale models – e.g. by scanning the objects – as input to the system, pose estimation using the Camera's Roll Histogram [1] is used together with any of the descriptors to calculate the pose of the model and align the 3D model with the scan from the sensor as depicted in Figure 8 which enables robotic manipulation tasks. We are currently working on methods that will be able to deliver the pose and the scale of the 3D models so that the whole 3DNet database can be used to recognize objects and estimate their pose targeting at virtual reality and robotic applications like grasping.

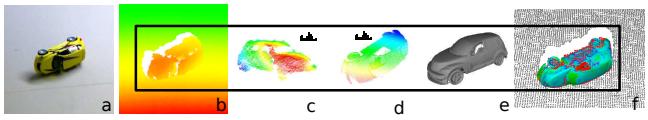


Fig. 8. Pose Estimation: Given a scan from a Kinect (a,b), the segmented point cloud (c) is matched against synthetic views of the model (e). Using the best matching view (d), the model is aligned to the scan(f).

D. Extensibility

Adding a new object class is easily achievable by following the following steps

1) Download 3D models of new object class from the various sources from the web or scan your objects
2) Convert the 3d models to PLY-format (we suggest meshconv[4])
3) Put the 3d models into a subdirectory of the already existing database and start the framework for view generation
4) Optional: A XML-file in each class directory provides the link to WordNet and additional attributes to the class.

E. DESCRIPTORS

The proposed framework comes with a set of available descriptors. The choice of descriptors is based on speed, availability and stability. Therefore global 3D descriptor are the first to be entering the framework such as VFH, CVFH, SHOT and shape distributions based descriptors as these provide the needed speed for robotics applications. Reimplementation and adaptation of Spherical Harmonics [8] and Spin Images [7] as global descriptors are the next to be put into action. Local descriptors with Bag-of-Words approaches as used in [4], [11] and [6] require an extra step of learning the visual-words which is not yet available in the framework.

1) VFH: The Viewpoint Feature Histogram is a descriptor based on normal vectors and was introduced in [14]. The descriptor is designed to be robust with respect to surface noise and missing depth information and the main focus is on recognition of objects learned beforehand. The average time for calculation and matching is approximately 70 ms.

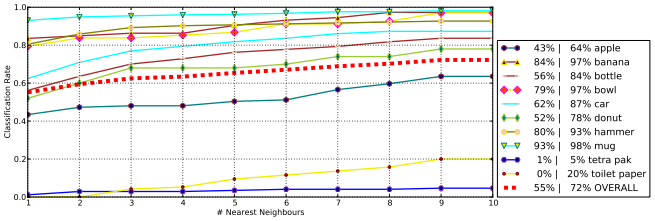


Fig. 9. VFH rank plot on the 10 classes test database against 10 Classes. VFH produces good results but fails on two classes.

The Viewpoint Feature Histogram (VFH) introduced in [14] is a viewpoint global descriptor based on angular normal distributions extracted from the surface normals and a reference coordinate system obtained by averaging the normals and points on the whole surface. It was designed to robustly describe the geometry of objects seen from a certain viewpoint using the same depth sensor for training and detection. The average time for calculation and matching is approximately 70 ms.

2) CVFH: The Clustered Viewpoint Feature Histogram [1] is a semi-global view based descriptor based on VFH. Because of its semi-global nature, only certain parts of the objects are used to build the reference systems on which the computation is based but uses the whole available view information to build the angular normal distribution histograms. Because of its multivariate representation of a partial view, it can deal with partial occlusions and cope with different data characteristics between training and detection. The parts of the object used to build the coordinate systems are obtained by a smooth region growing stage aiming to detect stable regions which are robustly estimated by the depth sensor. The descriptor computation time depends strongly on the region growing step, both in the number of points and the number of stable regions found. The average time for computation and search is approx. 208 ms, ranging from 50 ms and 300 ms.

3) SDVS: The Shape Distribution on Voxel Surfaces descriptor is a descriptor based on histograms of point-to-point distances and was introduced in [19]. The point distances are classified to be either on the surface of the partial view, off or mixed. This descriptor is calculated directly on the point cloud and does not need any normals to be computed and takes an average of 25 ms for calculation and matching.

4) ESF: The Ensemble of Shape Functions descriptor is based on the SDVS descriptor and includes multiple shape

[4]http://www.cs.princeton.edu/ min/meshconv/

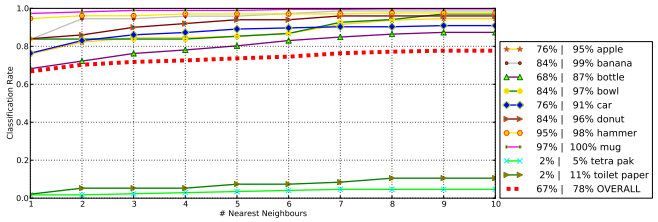


Fig. 10. CVFH rank plot shows improvement over VFH on 10 classes, but also has problems with two classes.

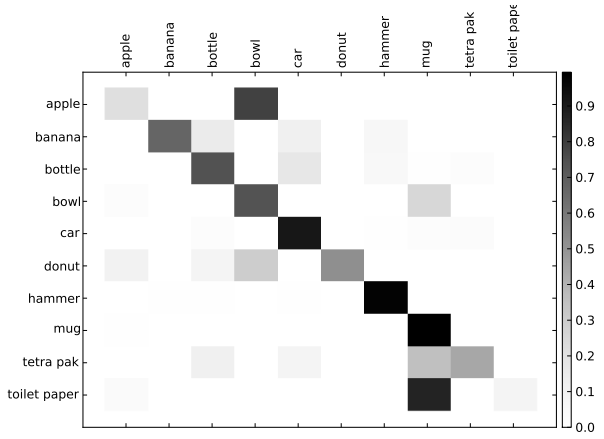


Fig. 11. SDVS rank plot on the 10 classes test database against 10 Classes. Most confusion is between the classes mug and toilet paper and between bowl and apple as partial views of these classes resemble parts of the other class.

function as described in Osada [12], such as A3(angles), D2(lengths) and D3(areas) and requires 75 ms for calculation and matching. A supplemental video of classifying object with this descriptor can be found on 3DNet (3dnet.org/video).

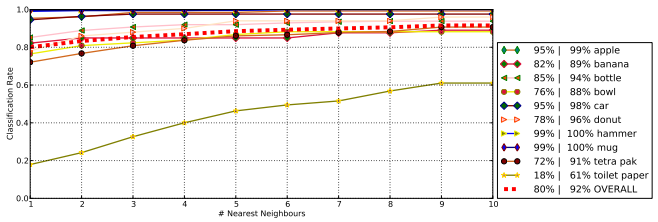


Fig. 12. ESF rank plot on the 10 classes test database against 10 Classes. The tetra pak class is working for this descriptor, but it still has problems with the similarity of mug and toilet paper classes.

5) SHOT: The SHOT descriptor introduced in [17] is aimed at surface matching with local descriptors, but is used here as a global descriptor for the whole object. The descriptor showcases a high classification rate, but compared to the other approaches the calculation time is up to than 10 magnitudes larger, so the feature calculation and matching takes from 130 ms to 4 sec on our test database.

F. Weight Learning on Synthetic Views

Parameters and descriptor weights can be learned on the synthetic views without having to see a single real scene. The improvement of the descriptor performance is showcased

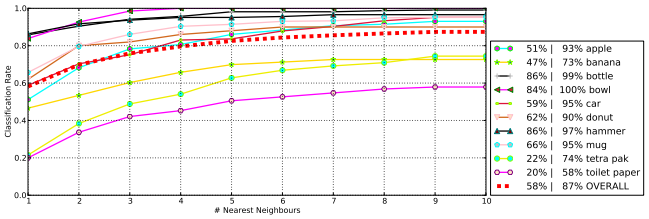
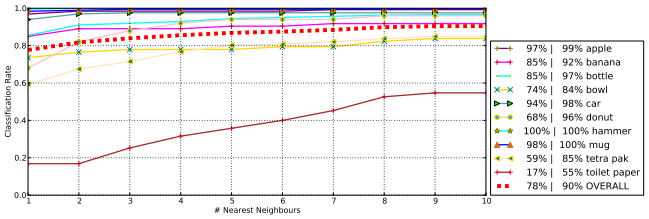
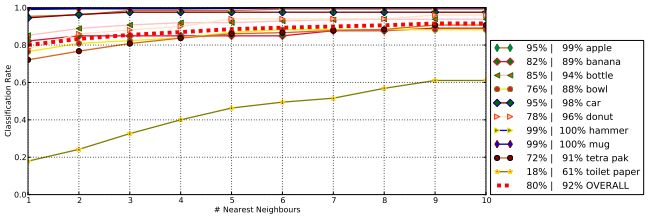


Fig. 13. SHOT rank plot on 10 classes provides good results with no class less then 20%, but is also the slowest descriptor in this benchmark.



(a) ESF descriptor with equal weights, no training.



(b) ESF descriptor with learned weights from synthetic views shows 2 % performance improvement.

Fig. 14. Weights learned on synthetic views for increased classification rate.

in Figure 15 where a 2 % improvement was achieved by learning the descriptor sub-histogram weights on a sample of the synthetic views. This method for tuning descriptors can be accomplished with any descriptor having sub-parts in its histogram and therefore weights can be learned. The big advantage here is that this can be done offline, without having a test database to split in training and evaluation parts.

V. BENCHMARK & EVALUATION

3DNET's intention is to provide benchmarks for 3D shape descriptors on the test databases in a similar way the Middlebury Stereo Benchmark [16] is for dense stereo.

For every descriptor rank-plots, confusion matrices and overview statistics are generated for the test sets against the model databases, e.g. 10-10, 10-50, 10-200, to provide insight and conclusions on descriptor performances. A sample benchmark is given for the ESF descriptor for 200 classes in Figure 14 and Table II.

As speed is a key issue in addition do classification performance for robotics, we do not follow the approach of the Middlebury Benchmark providing user to submit benchmarks. To foster sharing open-source code and enabling comparable performance measures, users are invited to include their descriptor in the framework, add test scenes to the test databases and add new categories, but benchmarking and

providing benchmarking results on 3d-net.org is done by the 3DNet itself.

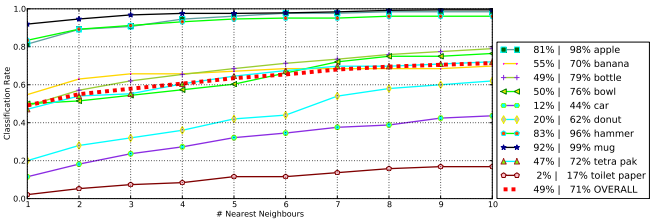


Fig. 15. ESF rank plot on Cat10 test database against 200 Classes.

TABLE II

NEAREST NEIGHBOR CLASSIFICATION AND MOST CONFUSING CLASS

class name	1-NN	10-NN	confusing class
per scenes OVERALL	58.22 %	78.23 %	
per class OVERALL	49.10 %	71.39 %	
apple	81.40 %	98.45 %	pumpkin
banana	54.79 %	69.86 %	pistol
bottle	48.77 %	79.01 %	suv
bowl	50.00 %	76.47 %	hat
car	11.52 %	43.64 %	suv
donut	20.00 %	62.00 %	cap
hammer	83.41 %	96.10 %	axe
mug	91.96 %	99.46 %	watch
tetra pak	47.09 %	72.09 %	mug
toilet paper	2.11 %	16.84 %	armchair

VI. CONCLUSIONS

A novel methodology is presented for rapid and scalable training of 3D shape descriptors using CAD models. To accomplish objective comparison of shape descriptors, 3DNet (3d-net.org) is presented as a free resource providing an open-source framework and test databases for benchmarking. Model databases with CAD models in 10,50,100 and 200 categories are presented as a common training resource. 3DNet offers to be extended by the community by adding new categories, creating a common test database and sharing new shape descriptors. 3DNet provides all necessary resources to process scenes as depicted in Figure 16. At the current state, segmentation is the main performance bottleneck, detaining us from having frame-rate classification. This leaves a lot of scope for future improvements in the challenging areas of handling touching objects, occlusions and speed and we hope with 3DNET, progress is accelerated.

Fig. 16. Classification with ESF with nearest neighbor on a scene with multiple objects. Challenges are wrong and missing segmentations, sensor noise and missing data on shiny and transparent objects and parts and descriptor flaws, which cause mis-classification.

REFERENCES

[1] A. Aldoma, N. Blodow, D. Gossow, S. Gedikli, R.B. Rusu, M. Vincze, and G. Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. *3rd IEEE Workshop on 3D Representation and Recognition*, 2011.

[2] Tarik Filali Ansary, Mohamed Daoudi, and Jean-Phillipe Vandeborre. 3d model retrieval based on adaptive views clustering. In *International Conference on Advances in Pattern Recognition (ICAPR)*, 2005.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009.

[4] H. Dutagaci, A. Godil, C. P. Cheung, T. Furuya, U. Hillenbrand, and R. Ohbuchi. Shrec 2010 - shape retrieval contest of range scans. In *Eurographics Workshop on 3D Object Retrieval*, 2010.

[5] Christiane Fellbaum. Wordnet: An electronic lexical database. *Cambridge, MA: MIT Press*, 1998.

[6] C. Goldfeder, M. Ciocarlie, J. Peretzman, Hao Dang, and P.K. Allen. Data-driven grasping with partial sensor data. In *International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[7] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligences*, 21(5):433–449, 1999.

[8] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. *SGP*, pages 156–164, 2003.

[9] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical rgb-d object dataset. *International Conference on Robotics and Automation (ICRA)*, 2011.

[10] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Sparse distance learning for object recognition combining rgb and depth information. *International Conference on Robotics and Automation (ICRA)*, 2011.

[11] R. Ohbuchi and T. Furuya. Scale-weighted dense bag of visual features for 3d model retrieval from a partial view 3d model. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 63 –70, 2009.

[12] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 154 –166, May 2001.

[13] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[14] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2155 –2162, 2010.

[15] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *International Conference on Robotics and Automation*, Shanghai, China, 2011 2011.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002.

[17] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. *11th European Conference on Computer Vision*, 2010.

[18] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. *IEEE Computer Vision and Pattern Recognition*, 2011.

[19] W. Wohlkinger and M. Vincze. Shape distributions on voxel surfaces for 3d object classification from depth images. *IEEE International Conference on Signal and Image Processing Applications*, 2011.

[20] Walter Wohlkinger and Markus Vincze. Shape-based depth image to 3d model matching and classification with inter-view similarity. *International Conference on Intelligent Robots and Systems*, 2011.

Supervised Learning of Hidden and Non-Hidden 0-order Affordances and Detection in Real Scenes

Aitor Aldoma, Federico Tombari and Markus Vincze

Abstract— **The ability to perceive possible interactions with the environment is a key capability of task-guided robotic agents. An important subset of possible interactions depends solely on the objects of interest and their position and orientation in the scene. We call these object-based interactions 0-order affordances and divide them among non-hidden and hidden whether the current configuration of an object in the scene renders its affordance directly usable or not. Conversely to other works, we propose that detecting affordances that are not directly perceivable increase the usefulness of robotic agents with manipulation capabilities, so that by appropriate manipulation they can modify the object configuration until the seeked affordance becomes available. In this paper we show how 0-order affordances depending on the geometry of the objects and their pose can be learned using a supervised learning strategy on 3D mesh representations of the objects allowing the use of the whole object geometry. Moreover, we show how the learned affordances can be detected in real scenes obtained with a low-cost depth sensor like the Microsoft Kinect through object recognition and 6D0F pose estimation and present results for both learning on meshes and detection on real scenes to demonstrate the practical application of the presented approach.**

I. INTRODUCTION

From a robotic perspective, the ability of understanding a specific environment together with the interaction possibilities provided in it represents a key capability for most autonomous agents. What an environment potentially affords depends strongly on two factors: (i) the objects and their configuration in the environment and (ii) the interaction capabilities embodied on a specific agent. The combination of both factors is coined under the term affordance [1]:

> "Affordances relate the utility of things, events, and places to the needs of animals and their actions in fulfilling them [...]. Affordances themselves are perceived and, in fact, are the essence of what we perceive."

In robotics, affordances have been primarily exploited in grasping or action-behaviour learning of objects, where 2D motion or colour cues have been related to object shape, e.g., [2] [3]. However, objects provide several more affordances, which we refer to as 0-*order affordances*, that are supported by geometrical properties of the object. For instance, objects like chairs or sofas can be used for sitting, because they provide a surface parallel to the ground and an attached vertical surface to lean back. Mugs, bowls, and in general containers, are used for liquid-containment because they provide a closed concavity. 0-order affordances do not depend solely on the geometry of the objects but also on their configuration in the world. Liquid containers can only fulfill their function if they are in an upright pose, while objects like sofas and chairs can be used for sitting only when found in a specific pose. We term *hidden* 0-*order affordances* those affordances that can be found on an object but not in the current pose, e.g., a chair or mug upside down. [1]

Given a certain task, e.g., fetch a container or prepare coffee, it becomes necessary for the robot to detect objects and their affordances. Placing the robot in a house or in an industrial setting provides structural information to the robot. Moreover, man-made objects are usually designed to fulfill their function(s) when placed in a certain pose(s) which due to the structured man-made world is expected to be *stable* on a planar surface. Hence, detecting the current pose of an object is particularly important to understand whether the current affordance is hidden or not.

Consequently, in this paper we propose an approach to learn 0-order affordances for objects modelled as 3D meshes by discretizing the space of possible orientations using their stable poses. The learned affordances are detected in real scenes by recognizing the objects of interest that are currently present in them and estimating their pose, see Fig. 1. Object recognition allows a direct mapping to both hidden and non-hidden affordances, which in turn enables the robot to either directly interact with the object or to plan interactions with the environment (e.g. manipulations) to make a hidden affordance available.

After reviewing related work, we present in Section III how hidden and non-hidden 0-order affordances can be learned on 3D mesh models where the whole geometry is available and therefore stronger cues can be exploited. An evaluation of several 3D descriptors and classifiers to capture affordances is also presented. Section IV demonstrates how through object recognition and 6DOF pose estimation, we are able to detect both non-hidden and hidden 0-order affordances in real scenes obtained with a low-cost depth sensors like the Kinect, which is valuable in the context of robotic platforms and task-guided agents that have the ability of manipulating the environment. In Section V we present an evaluation of the whole pipeline (see Fig. 1) and finally conclude with several future research directions.

Aldoma, Wohlkinger and Vincze are with Vision4Robotics Group, Automation and Control Institute, Vienna University of Technology, Austria [aa,mv] @ acin.tuwien.ac.at

F. Tombari is with the Computer Vision Lab, DEIS - ARCES, University of Bologna, Italy federico.tombari@unibo.it

[1] Following this terminology, 1^{st}-order affordances relate the object to the specific robot embodiment, e.g., chair height in respect to humanoid size. And 2^{nd}-order affordances represent what the embodiment handling on object affords, e.g., placing the object onto a table [4].

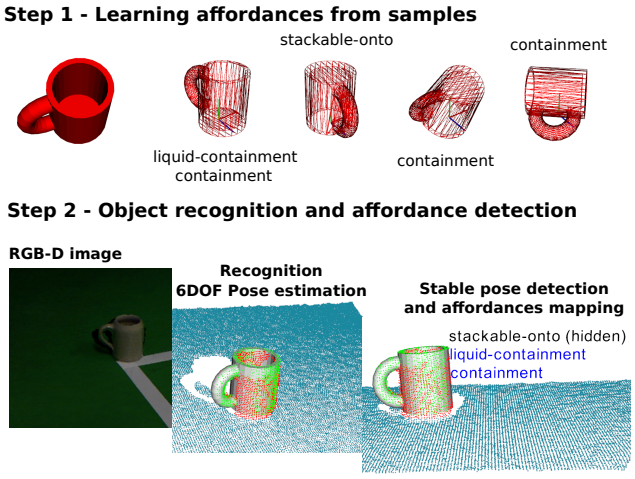


Fig. 1. The two steps of learning and detecting hidden and non-hidden 0-order affordances in real scenes: **Step 1:** affordances are learned using a pool of binary classifiers on the full-3D representations of the objects to be recognized using the methodology presented in Sec. III-E. **Step 2:** objects are recognized in range images and their 6DOF pose is aligned to map affordances based on the stable poses.

II. RELATED WORK

First work on object affordances used full-3D object models to describe the functions provided by specific object and object categories and the geometrical attributes or parts affording that functionality [5]. However, it is difficult to obtain such accurate 3D data from a robot under realistic settings, especially with real-time constraints. Hence, several attempts used 2D images to learn and detect affordances based on shape, texture and color features, e.g., [6] [3]. The main disadvantage of these approaches are the low-level features on which the learning is based. As a matter of fact, 2D features are often not adequate to learn geometrical attributes of objects, which are eventually used to detect specific affordances. Only recently 3D cues have been used in a similar approach [7] to learn the difference between container and non-container affordances based on robot-object interaction and depth images.

Probably the best studied affordance in robotics is grasping, i.e., *graspable*. Several authors have presented data-driven grasping techniques based on recognition and 6-DOF pose estimation [8], [9], [10]. These approaches are similar to our in that grasp hypothesis are learned from mesh representations of the object and applied to real objects after positive recognition and pose estimation. However, we decide to exclude the graspable affordance from our analysis for two reasons: (i) it is already a well-studied problem and (ii) we consider graspable to be a 1-order affordance as it depends strongly on the agent and, theoretically, all objects might be grasped with the appropriate embodiment.

Affordance-driven recognition has also been investigated in related fields. In [11] the authors perform 3D object categorization based on the definition of a *canonical form* of an object. Although not explicitly taking into account affordances, they (and citations therein) define categories by grouping objects based on their "main purpose/function". Recently, a similar approach is exploited in [12], where object affordances and grasping is used as an additional feature to aid object recognition. Also recently, Grabner et al. [13] learns the sittable affordance by matching a human sitting figure to depth images.

III. LEARNING 0-ORDER AFFORDANCES

The ultimate goal of this work is to detect hidden and non-hidden 0-order affordances by recognizing objects in the scene and estimating their 3D pose. Also, the objects used to train the recognition module are represented as 3D meshes obtained from CAD models or high-precision scanners. Once the pose is detected by the recognition module, the stable pose of the object (if any) is used to map the affordances of the recognized 3D mesh to the current scene. In [4], we show how this mapping can be obtained, although in that work a human operator had to manually insert affordances for each stable pose of the object, this seriously limiting the scalability of the method.

In this paper, we try to remove - or at least to greatly loosen up - the dependency from the human operator. By means of a initial training stage, we adopt a learning approach to automatically infer affordances on novel meshes. More specifically, we tackle this problem using a supervised learning approach to train independent binary classifiers, each one specialized on a single affordance. Thanks to this approach, we are then able to associate a set of pre-defined affordances to any given mesh depicting an object in a stable pose, by classifying it through the set of trained classifiers.

The set of 0-order affordances we consider are:

- *rollable*: the object can roll if pushed.
- *containment*: the object can contain other objects.
- *liquid-containment*: the object can contain liquids.
- *unstable*: the stability of the pose is compromised if pushed.
- *stackable-onto*: objects can be stacked onto the object.
- *sittable*: an agent can sit on it like a human would do.

Since in real world most man-made object categories are designed to fulfill their intended functionality when they are placed on a stable pose, stable poses are a perfect candidate to discretize the 0-order affordances space. The forthcoming sections will investigate the following points: i) how to compute a set of stable poses of an object; ii) how to label the models to obtain an initial training set for classifiers; iii) which descriptors are most adequate to capture the geometrical attributes of the affordances: iv) how the proposed approach performs with different, general-purpose machine learning algorithms. Finally, we conclude this section by comparing several state-of-the-art descriptors and classifiers in order to determine the best descriptor-classifier combination for each affordance.

A. Stable pose computation

An object is in a stable pose if it will persist in that same pose when not disturbed by external agents. As described in

[14], the stable planes of a model are a subset of the tangent planes enclosing a model - the planar faces of the convex hull. The triangle faces of the convex hull can be grouped in planar faces by performing a hierarchical clustering [15]. The final planar faces represent the tangent planes Π that need to be further analyzed for stability. We refer the reader to [4] for a detailed explanation on how we compute the stable planes of a 3D mesh.

B. Labeling of training models

Given a set of object affordances, $\mathcal{A}$, and an set of objects, $\mathcal{O}_0$, we start by creating supervised object - affordance relationships inserted by a human operator. We have a CAD model representation of each object in the initial object set. An object $o \in \mathcal{O}_0$ is displayed to the operator together with a list of all possible affordances $\mathcal{A}$. At this point, the operator inputs the affordances belonging to the current object, in addition he inputs whether any of these affordances might be hidden when the object is found in the environment by a robotic agent. If there is any, the operator is shown the object in different stable poses and for each of them, he types in whether the affordance is hidden or not (see Fig. 2).

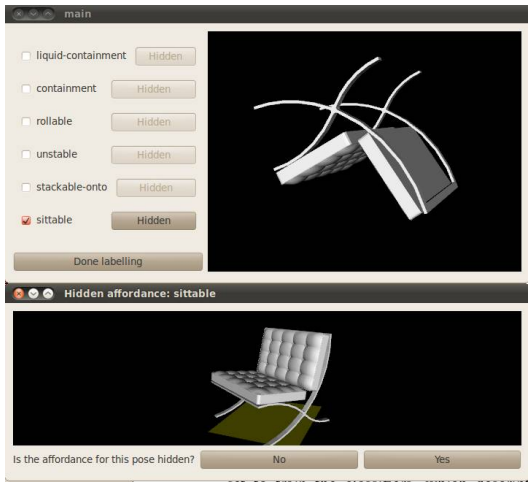


Fig. 2. Screenshot of the labeling tool. In the top figure, the user is given a CAD model to label in terms of its 0-order affordances. When the "hidden" button for a specific affordance is pressed, a window (at the bottom) appears allowing the user to input whether the current affordance is hidden or not.

C. Descriptors

In machine learning approaches the representation of the input data given to the learning algorithms is a key factor for the accomplishment of the final application. Thus, we have tested a pool of 5 3D descriptors, some of them taken from the literature and others specifically tailored for our needs. In general, and conversely to most works in literature, we are looking here for 3D descriptors that are pose dependant to capture the affordances of the objects depending on the specific stable pose. A brief review of the evaluated descriptors is now given. In the following, we will refer to C_p as the projection of the centroid of the mesh on the stable plane, as well as to N_p as the normal on that plane.

Spherical Extent Descriptor (SEE) [16] — Our implementation is based on computing the length of the last N intersections between the mesh and N rays running from C_p to N points sampled on a tesselated sphere where the mesh is circumscribed (as shown in Fig. 3) These lengths are binned into a histogram of N elements, where N depends in turn on the number of tesselations performed on an initial icosahedron ($N = 20 * 4 * K$) used to approximate the sphere. For the experiments, we use a tesselation level of 2, yielding a total of 320 bins.

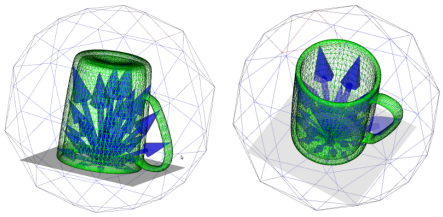


Fig. 3. Visualization of a Spherical Extent Descriptor for a mug found in a upside-down pose (left) and upright (right).

Normal Distributions Sliced (NDS) — This descriptor aims at capturing the distribution of the differences between the normals on the mesh and N_p. Specifically, we take the dot product between N_p and n_i, which ranges between $[-1, 1]$. To capture the spatial distribution of the normals along N_p, the mesh is sampled with $20K$ points and the sampled points are sliced along this direction (see Fig. 4). The normal n_i is computed on the mesh triangle where each point has been sampled from. The normal distribution of all points in a specific slice is binned into an histogram with 45 bins. We use 3 (NDS3) or 5 (NDS5) slices and the histograms relative to each slice are finally concatenated giving a total length of $45 * \#slices$. Hence, we aim to render this descriptor particularly discriminative with regards to affordances such as stackable-onto or sittable, where several normals of the sampled points are parallel to N_p). In addition, it should be particularly descriptive also for the "rollable" affordance, since when this affordance is accomplished the lowest slice (i.e. that closest to the ground) should accumulate mostly negative and uniformly distributed (i.e. without peaks) dot products, as they would represent a rounded face.

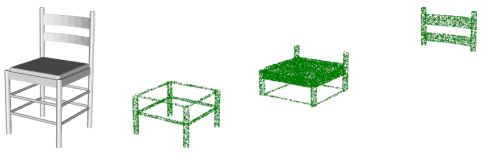


Fig. 4. Slices along N_p used by NDS for a chair standing up-right.

SHOT — The SHOT descriptor [17] was originally proposed as a local descriptor, encoding a signature of histograms of topological traits. A 3D spherical grid centered on the feature to be described and oriented according to a

unique local Reference Frame defines the elements of the signature. Each element is in turn a histogram, accumulating the cosine between the normal of the center point and the normal of each point falling in the current spherical sector of the grid. For better robustness a quadrilinear interpolation and a normalization step are also applied.

Spin Images [18] — The Spin Image descriptor is based on sweeping a discretized plane (the Spin Image itself) around the normal of the point being described, and accumulating at each bin the number of intersections with the points of the object through all sweeps. We place the spin image plane to be perpendicular to N_p, spanning from C_p to C_p plus the height of the object and from C_p to the farthest away point projected on the stable plane. We then sweep with an angular resolution of 10 degrees and the accumulation result represents the spin image with a size $32x64$.

Point Feature Histogram [19] — This descriptor is a modification of PFH. The normal angular distributions of the normals are computed using the normals of all points on the mesh (p_i, n_i) and (C_p, N_p) as follows:

$$\begin{aligned} \mathsf{u}_i &= \mathsf{N_p} \\ \mathsf{v}_i &= \frac{\mathsf{p_i} - \mathsf{C_p}}{||\mathsf{p_i} - \mathsf{C_p}||} \times \mathsf{u}_i \\ \mathsf{w}_i &= \mathsf{u}_i \times \mathsf{v}_i \end{aligned} \tag{1}$$

The normal angular deviations $\cos(\alpha_i)$, $\cos(\phi_i)$ and θ_i for each point $\mathsf{p_i}$ and its normal $\mathsf{n_i}$ are given by:

$$\begin{aligned} \cos(\alpha_i) &= \mathsf{v}_i \cdot \mathsf{n_i} \\ \cos(\phi_i) &= \mathsf{u}_i \cdot \frac{\mathsf{p_i} - \mathsf{p_c}}{||\mathsf{p_i} - \mathsf{p_c}||} \\ \theta_i &= \mathrm{atan2}(\mathsf{w}_i \cdot \mathsf{n_i}, \mathsf{u}_i \cdot \mathsf{n_i}) \end{aligned} \tag{2}$$

Finally, the spatial distributions of the points is computed using the distance from each point to C_p and binned into two different histograms, one along N_p (capturing the height of p_i relative to the stable plane) and the other representing the distance of the projected points on the plane. The normal angular distributions are binned into 3 histograms, each 45 bins and the spatial distributions into 2 histograms, also 45 bins. The final histogram is obtained by concatenating the 5 histograms giving a total size of 225.

D. Classifiers

As previously mentioned, in order to automatically associate affordances to a CAD model in a specific pose, we deploy a pool of binary classifiers, each trained on a specific affordance. Thanks to this approach, we are able to determine whether each evaluated affordance is actually hidden or not in the current pose of the object. For this aim, we propose to use, as the input sample for the classifier, a global descriptor computed in a pose-dependent way (thus, explicitly avoiding rotational invariance).

In our experiments, we have used different popular classifier methods in order to evaluate the generality of our approach. More specifically, we have employed Support Vector Machines (SVM) [20], Boosting [21] and Random Forests [22]. All implementations were provided by the open source library OpenCV.

During the training stage, a set of global descriptors is computed on several models (each one in a different pose) so as to populate the training set. For parameter selection, a *k-fold* cross-validation approach is used, by dividing the training set in k parts and using in different permutations $k-1$ folds for training and the remaining one for validation. Given the specific characteristics of our training set, i.e. small due to the limited number of objects and poses used for training, and unbalanced due to a limited number of positive samples, we have decided to set $k = 2$. Finally, in our approach we haven't used any particular dimensionality reduction approach, although for certain descriptors this could have been beneficial given their cardinality (on the order of a few hundreds): this analysis currently represents a future direction of this work.

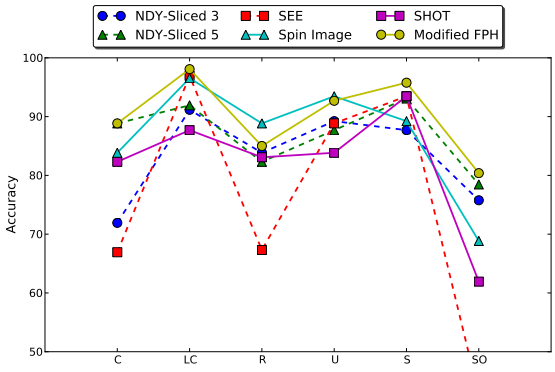
We have used the same training set for all experiments shown throughout this paper. In particular, it is composed of 43 CAD models selected from the *Google Warehouse* dataset [2], which include the following affordance-rich object categories: chairs, sofas, bottles, mugs, bowls, stools, office chairs, toilet paper and tetra pacs.

E. Learning affordances on CAD models - Evaluation

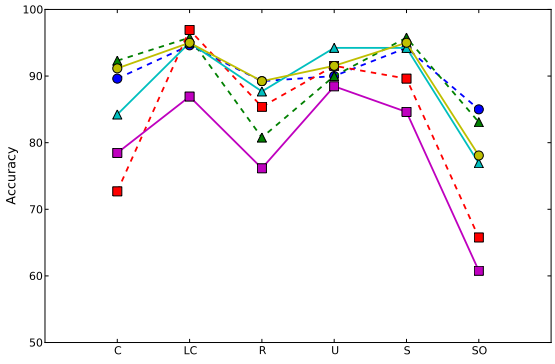
This subsection illustrates an experimental evaluation aimed at demonstrating affordance detection on CAD models based on the descriptor and learning techniques previously introduced. Also, the goal here is to evaluate what is the best performing descriptor-classifier combination among those being evaluated. In our experiments, we have selected 45 CAD models from the Princeton Shape Benchmark (PSB) dataset [23] (obviously not included in the training set) to form a test set. Ground truth for this set has been obtained by manual labelling following the same tool and rules used for the training set. As for the selection of the models composing our test set, for the sake of the evaluation we favoured objects having multiple affordances, and included chairs, benches, mugs, bottles, wheels, sofas, table, glasses, shelves, beds and stools, which are good representatives for the set of affordances we take into account.

From the results presented in Fig. 5 we can point out the following aspects: (i) for certain affordances, learning is particularly challenging, e.g., stackable-onto and liquid-containment report a classifications rate below 90%, (ii) there is no descriptor that clearly outperforms the others over the evaluated affordance set and (iii) SVM and boost classifiers seem to outperform random forests. Therefore, as a main guideline to learn affordances on CAD models, we suggest to select the best descriptor combined with the best learning algorithm for each single affordance. Furthermore, we wish to point out that there are several ways that could help increasing the classification rates, which we regard here as future work: (i) increase the size of the training set, improve the balance between the population of the two classes (ii)

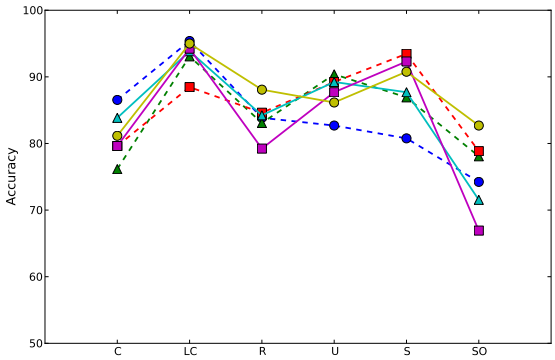
[2] http://sketchup.google.com/3dwarehouse/



(a) SVM classifier



(b) Boost classifier



(c) Random Forests classifier.

Fig. 5. Accuracy rates for all descriptors and all affordances. Each chart is relative to a different classifier: from top to bottom, SVM, Boost, Random Forests. C,LC,R,U,S,SO stand respectively for containment, liquid-containment, rollable, unstable, sittable and stackable-onto

use dimensionality reduction techniques to face the sparsity due to the typically high dimensions of the descriptors and (iii) combine together multiple descriptors.

One interesting final remark for this section is that for some object categories — or, more appropriately, for objects sharing the same functionality — our affordance detector based on stable poses is able to compute, as by-product, a semantic alignment of the object up to a rotation about the stable plane normal (see Fig. 6). This might be of interest for task-based applications that require objects to

Fig. 6. Princeton Shape Benchmark [23] models displayed in the pose where, according to our approach, the sittable affordance was detected as fulfilled (detected using a SVM classifier with a PFH descriptor). Alignment on the plane is obtained by maximizing symmetry along the z-axis. The sittable affordance allows for a semantic alignment (up to a rotation over the plane normal) of objects that can be used for sitting, even when their geometry is completely different.

be semantically aligned like in [24].

IV. DETECTING HIDDEN AND NON-HIDDEN AFFORDANCES IN REAL SCENES

As stated throughout the paper, our ultimate goal is to detect hidden and non-hidden 0-order affordances in real environments using sensors tipically available on mobile platforms. We have shown how affordances of object models represented in the form of full-3D meshes can be learned using a supervised learning strategy. Now, the challenge is represented by matching our models, where 0-order affordances have been automatically detected, to objects in real scenes where the data is represented by partial views and acquired with a depth sensor (we focus our attention on low-cost sensors such as the recently released Microsoft Kinect). In addition, we also aim at estimating their 3D pose and, finally, estimating the stable pose (if any) on which the objects are found in the real world so to obtain, by association, the hidden and non-hidden 0-order affordances.

A. *Object recognition*

The object recognition module is probably the most interchangeable module in the whole pipeline. But, due to the fact that our models are represented as noiseless meshes — CAD models downloaded from the Internet or obtained with high-precision scanners — we need to deploy object recognition techniques able to deal with the significant differences in the 3D data characteristics among training and test. Also, our algorithms cannot rely on color information to improve their recognition capabilities due to two main reasons: (i) most CAD models present in public datasets are provided without texture information (ii) within the scope of thi paper, we have considered only affordances that can be perceived (and discriminated among each other) using only shape cues, therefore it would be interesting to limit the object recognition module as well to exploit this cue only.

Because of these contraints, we decided to use the Clustered Viewpoint Feature Histogram (CVFH) descriptor and the recognition pipeline presented in [25], which has been

shown to carry out good performance in a scenario similar to the one we are facing here. CVFH is a semi-global view-based descriptor composed by several histograms based on the normal distributions of the object surface. Because of its multivariate representation, it can deal with occlusions and "holes" typically present in the data due to the limited quality of the deployed 3D sensor (see Fig. 9). Moreover, combined with the Camera's Roll Histogram (CRH) [25], aimed at determining the final degree of freedom over the camera axis and a post-processing step, it is able to accurately determine the object poses.

Besides, we also employ, in addition to the CVFH descriptor, two other view-based descriptors: the Viewpoint Feature Histogram (VFH [19]) and Shape Distributions on Voxel Surfaces (SDVS [26]), which will be altogether included in our experimental evaluation. All evaluated descriptors are used in combination with the same CRH stage and post-processing stage for a full 6DOF pose estimation. Please note that the descriptors used in the recognition module are designed to recognize objects using the depth data obtained from a certain viewpoint and threfefore, the descriptors presented in III-C aimed to describe the whole geometry of an object are not adequate for this problem.

B. Stable pose estimation

Once object recognition and pose estimation have been carried out, for those objects which one or more hidden 0-order affordances were detected for, we need to further evaluate if their current pose in the scene makes any hidden 0-order affordance usable. Let $\mathcal{M}_1$ represent the object in camera coordinates once it has been aligned using the procedure explained in Section IV-A and let n_{dp} represent the normal of the dominant plane in the scene. Let $\mathcal{M}_2$ represent the same object in object coordinates together with the set of stable planes Π, where each $\pi \in \Pi$ has been labeled to have the specific 0-order affordance hidden or not hidden using the learning mechanisms presented in Section III. The problem can then be expressed in the following way: find $\pi_i \in \Pi$ that best aligns $\mathcal{M}_2$ with $\mathcal{M}_1$ and check if the affordance for the stable pose based on π_i is hidden or not.

We use the method presented in [27] to align $\mathcal{M}_2$ and $\mathcal{M}_1$ (assumed to stand on the plane with normal n_{dp}). Since the method is based on stable planes, the best alignment yields a certain π_i from $\mathcal{M}_2$. By looking at the labeled information associated with π_i, we can then understand whether, in the current configuration, the hidden 0-order affordance is hidden or not. Note that in our representation, a hidden 0-order affordance is a boolean variable and we do not consider poses where the object might partially fulfill the affordances. In the case that the pose retrieved by the procedure in Section IV-A does not represent a stable pose, the system will consider all pose-dependant affordances to be hidden. The absence of a stable pose is detected by thresholding a similarity measure computed between both meshes after the best stable pose is found.

V. EVALUATION

We already presented, in section III-E, an experimental evaluation of descriptors and classifiers aimed at learning affordances on a training set of CAD models extracted from the PSB dataset. In this section, we propose an evaluation of the whole pipeline aimed at detecting hidden and non-hidden 0-order affordances in real objects acquired with a Microsoft Kinect sensor. In particular, as depicted in Fig. 1, the following aspects are being evaluated:

- Learning 0-order affordances on the CAD models representing the real objects (see Section III).
- Object recognition and 6DOF pose estimation (see Section IV-A).
- Stable pose detection (see Section IV-B).

For this purpose, 20 objects are selected and placed in front of the camera. Several snapshots are acquired for each object, each one referred to a different stable pose. We take 5 snapshots per object, except for highly symmetrical object, in which case only 2 (spherical objects) or 3 (cylindrical objects) snapshots are taken, yielding a total of 85 scenes. Each scene is manually labeled with hidden and non-hidden 0-order affordances depending on each object and its configuration.

Obviously, since the latter stages of our pipeline highly depend on the outcome of the previous ones, errors add up and even with a perfect recognition and pose estimation, affordances might be incorrectly detected if the learning algorithms failed to properly classify the affordances on the mesh. In order to better estimate the performance of each main stage of the pipeline, the affordances on the 20 mesh models used in these experiments are also manually labeled so that the errors given by the recognition and pose estimation methods, together with the stable pose detection, can be evaluated independently.

As previously mentioned, we use the best combination of descriptors and classifiers according to the evaluation in Sec. III-E to learn the affordances for the 20 meshes representing the real objects. We carry out an experimental evaluation of our approach by reporting a standard accuracy metric based on the number of true positives, false positives, true negatives and false negatives between the detection results and the labeled scenes used as ground truth. The different descriptors presented in Sec. IV-A are independently evaluated together with the number of nearest neighbours that are post-processed. In Fig. 7 the results are presented, both using the learned affordances and also using manually labeled affordances in order to isolate the error source. Thus, the evaluations $*\_GT$ use the manually labeled affordances and therefore quantify only the error of the recognition method and 6DOF pose estimation plus the detection of the stable pose. Approximately, half of the error is caused by the learning mechanism and the other half by subsequent stages of the pipeline.

Even though the test scenes do not present occlusions, CVFH outperforms both VFH and SDVS, giving an affordance detection rate of 70% using learned affordances

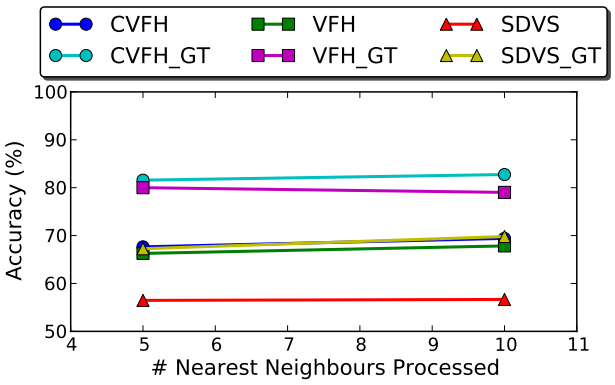


Fig. 7. Evaluation of the accuracy in the affordance detection using several descriptors and as a function of the number of nearest neighbours that are post-processed. All nearest neighbours are post-processed in the same way.

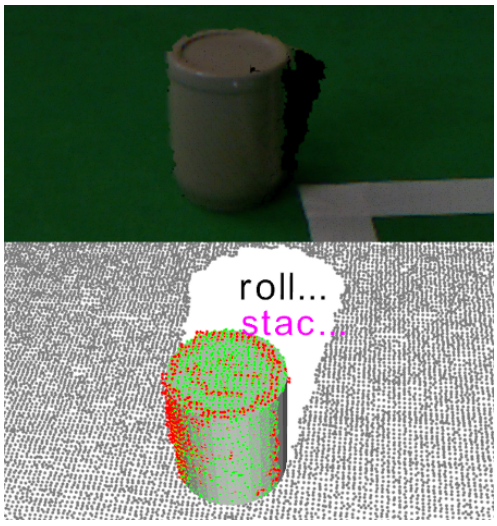


Fig. 8. A mug seen from a viewpoint where the handle is not visible and in an upside-down pose. The mug gets recognized as a cylinder and hidden affordances are incorrectly detected. Green points depict the view rendered from the CAD model, while red points depict the segmented clusters in the current scene. Observe how both red and green points match almost perfectly despite the challenging scenario, where color information is completely discarded.

and 84% for the ground truth affordances. Fig. 9 shows recognition results and affordance detections using CVFH on a scene where several objects are partially occluded.

It is important to note that in our evaluation we also take into account hidden affordances and therefore, in some situations like scenes where a mug is upside-down and the handle is not seen, the object might be recognized as a cylinder (if there is a cylinder model with a size similar to that of the mug), therefore the hidden containment and liquid-containment affordances will be counted as false negatives and the hidden rollable affordance (detected in the cylinder model) will account for a false positive (see Fig. 8). To demonstrate the effect of these circumstances on the performance, in Table I we report the accuracy rates for non-hidden affordance using CVFH and we can see how the performance is significantly improved.

	# nearest neighbours	
	5	10
CVFH	85.2	86.4
CVFH_GT	93.5	94.1

TABLE I

ACCURACY RATES OF NON-HIDDEN AFFORDANCE DETECTION USING CVFH, BOTH FOR LEARNED AND MANUALLY LABELED AFFORDANCES.

VI. CONCLUSIONS AND FUTURE WORKS

We have proposed a method to infer 0-order affordances on 3D models using supervised learning algorithms, where 3D surface descriptors are employed as a representation of affordances. Moreover, we have shown how object recognition methods providing 6DOF pose can be used to detect affordances in real scenes obtained with the Kinect sensor by mapping the estimated object and pose to the learned affordances on the mesh model.

In the evaluation section, some challenges have been presented demonstrating the difficulty of the task. We believe that extended representations using texture, color and materials will have to be used to discriminate affordances as the number of considered affordances keep growing. For instance, the openable affordance will have to use a texture representation combined with material in order to be classified. Material might help as well to discriminate between affordances like containment and liquid-containment. A human would never use an opened shoe box made of carton as liquid-container although in the correct pose and based on geometrical properties, a classification system like the one we presented here might. Yet, we still believe that geometrical classifier based on 3D mesh representation can be very helpful and provide accurate classifiers when combined with different cues.

Our future research line includes, among others, the integration of different cues to represent a broader set of affordances and integration with robotic platforms to show the practical application of affordances in task-based scenarios where 0-order affordances will be used as starting interaction chances filtered by the task and higher order affordances.

REFERENCES

[1] J. Gibson, *The ecological approach to visual perception*. Boston, MA, USA: Houghton Mifflin, 1979.

[2] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning About Objects Through Action - Initial Steps Towards Artificial Cognition," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 3140–3145.

[3] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation," *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 15 –26, feb. 2008.

[4] A. Aldoma, R. B. Rusu, and M. Vincze, "0-Order Affordances through CAD-Model Recognition and 6DOF Pose Estimation," in *In Workshop: Active Semantic Perception and Object Search in the Real World, IROS Yet to appear, see http://3d-net.org/privatepapers/IROSWorkshop.pdf*, 2011.

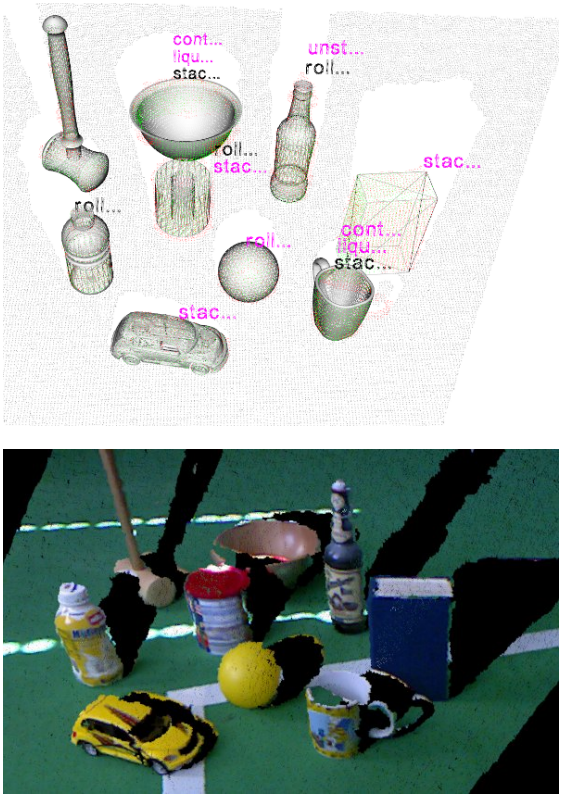


Fig. 9. Top: A scene recognized with CVFH. The CAD models are overlapped on the scene together with the 0-order affordances that each object provides. Hidden affordances are shown in black and non-hidden in magenta. Green points represent the best-matching rendered view from the training set, while red points represent the segmented clusters in the current scene acquired by the depth sensor (best viewed in color). Bottom: Same scene from the Microsoft Kinect sensor with color image overlayed. Note the difference between the CAD models used to trained CVFH and the data obtained from the Kinect, as well as the partial occlusions present in the dictionary, the hammer and the bowl.

[5] M. Sutton, L. Stark, and K. Bowyer, "Gruff-3: Generalizing the domain of a function-based recognition system," *Pattern Recognition*, vol. 27, pp. 1743–1766, 1994.

[6] G. Fritz, L. Paletta, M. Kumar, G. Dorffner, R. Breithaupt, and E. Rome, "Visual Learning of Affordance Based Cues," in *From Animals to Animats 9*, ser. Lecture Notes in Computer Science, S. Nolfi, G. Baldassarre, R. Calabretta, J. Hallam, D. Marocco, J.-A. Meyer, O. Miglino, and D. Parisi, Eds., vol. 4095. Springer Berlin / Heidelberg, 2006, pp. 52–64.

[7] S. Griffith and A. Stoytchev, "Interactive Categorization of Containers and Non-Containers by Unifying Categorizations Derived From Multiple Exploratory Behaviors," *Association for the Advancement of Artificial Intelligence (AAAI). Atlanta, Georgia.*, 2010.

[8] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative Grasp Planning with Multiple Object Representations," 2011.

[9] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, "Data-Driven Grasping with Partial Sensor Data."

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

[11] G. Somanath and C. Kambhamettu, "Abstraction and generalization of 3D structure for recognition in large intra-class variation," in *Proc. Asian Conf. on Computer Vision (ACCV 10)*, 2010.

[12] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and C. Barbara, "Using object affordances to improve object recognition," *IEEE Trans. Autonomous Mental Development*, 2011.

[13] H. Grabner, J. Gall, and L. J. V. Gool, "What makes a chair a chair?" in *CVPR*, 2011, pp. 1529–1536.

[14] H. Fu, D. Cohen-or, G. Dror, and A. Sheffer, "Upright orientation of man-made objects," *ACM Trans. Graphics*, pp. 1–7, 2008.

[15] M. Attene, B. Falcidieno, and M. Spagnuolo, "M.: Hierarchical mesh segmentation based on fitting primitives," *The Visual Computer*, vol. 22, pp. 181–193, 2006.

[16] D. Saupe and D. V. Vranic, "3d model retrieval with spherical harmonics and moments," in *DAGM*. Springer-Verlag, 2001, pp. 392–397.

[17] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of Histograms for local surface description," in *Proc. 11th European Conference on Computer Vision (ECCV 10)*, 2010.

[18] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 21, no. 5, pp. 433–449, 1999.

[19] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010.

[20] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[21] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. EuroCOLT*, 1995, pp. 23–37.

[22] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[23] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *In Shape Modeling International*, 2004, pp. 167–178.

[24] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning Task Constraints for Robot Grasping using Graphical Models," in *In IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS'10)*, 2010.

[25] A. Aldoma, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, M. Vincze, and G. Bradski, "CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues," in *In Workshop: 3rd IEEE Workshop on 3D Representation and Recognition, ICCV Yet to appear, see http://3dnet.org/privatepapers/ICCVWorkshop.pdf*, 2011.

[26] W. Wohlkinger and M. Vincze, "Shape Distributions on Voxel Surfaces for 3D Object Classification From Depth Images," 2011.

[27] A. Aldoma and M. Vincze, "Pose Alignment for 3D Models and Single View Stereo Point Clouds Based on Stable Planes," *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2011.

From Object Categories to Grasp Transfer Using Probabilistic Reasoning

Marianna Madry, Dan Song and Danica Kragic

***Abstract*— In this paper we address the problem of grasp generation and grasp transfer between objects using categorical knowledge. The system is built upon an i) active scene segmentation module, able of generating object hypotheses and segmenting them from the background in real-time, ii) object categorization system using integration of 2D and 3D cues, and iii) probabilistic grasp reasoning system. Individual object hypotheses are first generated, categorized and then used as the input to a grasp generation and transfer system that encodes task, object and action properties. The experimental evaluation compares individual 2D and 3D categorization approaches with the integrated system, and it demonstrates the usefulness of the categorization in task-based grasping and grasp transfer.**

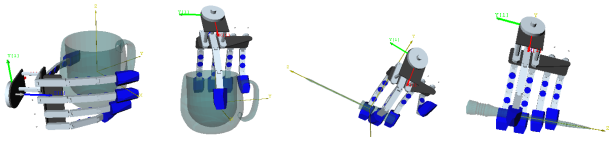
I. INTRODUCTION

Household environments pose serious challenges to robotic perception and manipulation: objects are difficult to locate and manipulate due to the unstructured settings, variable lighting conditions and complex appearance properties. Although some excellent examples of finding and manipulating a *specific* object in a scene have been reported in the literature [1][2], the aspect of *generalization* have not been addressed seriously: there are no systems that can flexibly and robustly, in realistic settings, find objects that *fulfill* a certain functionality thus executing tasks such as "**Robot, give me something to hammer with.**" or "**Robot, bring me something to drink from.**"

The aspect of function is related to that of affordances [3], [4] and has been addressed frequently in works that learn relations between objects and actions [5], [6], [7], [8], [9]. However, none of these consider the aspect of *task* in their model: what the agent is required to do with an object will affect the type of action (grasp) to apply. In this case, the task will be constraining the action space - not just any grasp can be applied on the object, see Fig. 1. Another closely related example is finding something to *hammer-with* or *pour-to* that relates to the notion of functional categories that have been addressed in a limited fashion in computer vision [10], [11].

In this paper, we present work on encoding object categorical knowledge with task and action related reasoning. Knowledge of object category facilitates action (grasp) transfer: i) detecting an object that affords pouring may be easily pursued at the categorical level, or ii) knowing how to grasp an object that affords pouring may be transferred to another object that belongs to the same category. We build upon our previous work, [12], [13], [14] where we developed a probabilistic grasp reasoning system. The system models the conditional dependencies between the tasks, actions and objects taking into account the constraints posed by each. However, in previous work we concentrated on theoretical problems of structure learning in graphical models without considering the aspect of *real* sensory information extracted in natural scenes.

The authors are with Computer Vision and Active Perception Lab, Center for Autonomous Systems, KTH-Royal Institute of Technology, Sweden, madry,dsong,danik@csc.kth.se. This work was supported by GRASP, IST-FP7-IP-215821 and Swedish Foundation for Strategic Research. We thank Jeannette Bohg and Carl Henrik Ek for their help.



(a) *pouring* (b) *hand-over* (c) *tool-use* (d) *hand-over*

Fig. 1. Grasping a cup: (a) *pouring* and (b) *hand-over* task (hand should not block the opening), and a screwdriver: (c) *tool-use* (hand should grasp the handle) and (d) *hand-over* task.

We present an integrated approach to task-oriented grasp reasoning and categorization, with the novel aspect of grasp transfer. The contributions of the proposed system are that:

- we enable a robot to choose the objects in a 3D scene that afford the assigned task while
- planning the grasp that satisfies the constraints posed by the task;
- grasp knowledge can be transferred between objects that belong to the same category, even under considerable differences in appearance and physical properties.

Our system integrates 2D and 3D visual information and captures different object properties (appearance, color, shape) what makes the categorization process robust in real-world scenes. We show that the system can successfully discriminate between objects sharing similar properties but affording different tasks, such as a carrot and a screwdriver that are structurally similar but fulfill different functions.

The paper is organized as follows: In Sec. II we present the probabilistic reasoning framework and in Sec. III object categorization system. Sec. IV outlines the experimental evaluation and Sec. V concludes the paper.

A. The system

Our system consists of three main parts, see Fig. 2:

- Visual Front End: here, an active robot head equipped with foveal and peripheral cameras provides input to the real-time scene segmentation system [15];
- Categorization: the system provides information about object class using various object properties such as appearance, color and shape;
- Reasoning system: the probabilistic grasp reasoning system, that encodes task-related grasping [12][13].

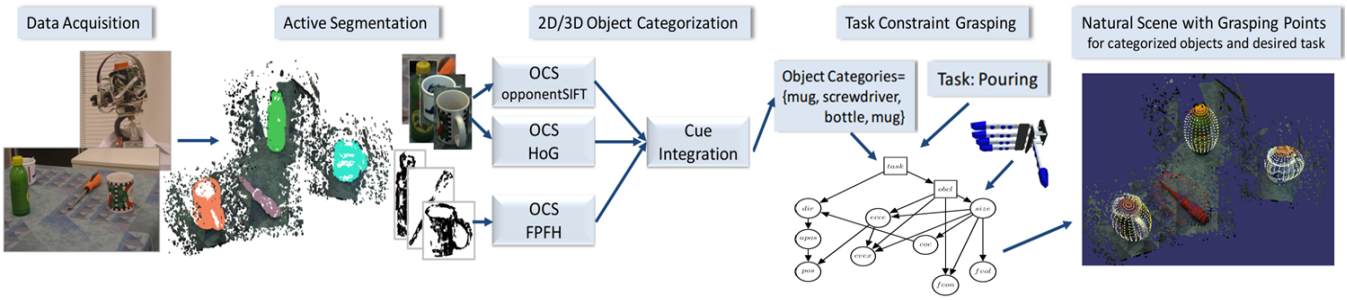


Fig. 2. Visual Object Category-based grasp generation for an arbitrary scene: objects are first segmented and categorized using our 2D-3D Object Categorization Systems (OCSs). Then, grasping hypotheses are generated taking the task into account. The image is best viewed in color.

We start by providing the necessary details for our probabilistic reasoning system.

II. Encoding Task Constraints

In the previous work [12], [13], [14], we have developed a probabilistic framework for embodiment-specific grasp representation. We model the conceptual task requirements using a Bayesian network through conditional dependencies between task, object, action and constraints posed by each. The model is trained using a synthetic database of objects, grasps generated on them, and the task labels provided by a human. The data generation is based on the toolbox BADGr [16]. BADGr provides 3D object shape approximation, grasp planning, execution and also grasp-related feature extraction and task labeling. We refer the reader for the detailed process of data generation to [12].

Both the structure and the parameters of the BN are learned from the database. The BN structure encodes dependencies among the set of task-related variables, and the parameters encode their conditional probability distributions. Fig. 3 shows the learned structure of the BN with the features listed in Table I. Once trained, the model can be used to infer distribution of a small set of variables based on a partial or complete observation of others. This property is used to generate a likelihood map on a set of grasp position around each object.

Our previous work was done in simulation and the inference engine assumed the object class unknown. Learning of the network structure in [13] revealed the importance of the categorical information. This motivated us to integrate the object categorization module with the task-constraint grasp reasoning system.

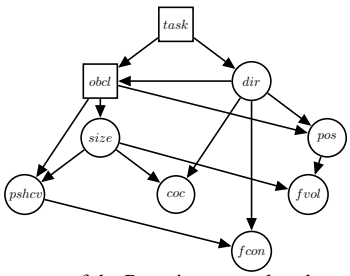


Fig. 3. The structure of the Bayesian network task constraint model.

III. 2D-3D Object Categorization System

Many household objects that afford different tasks have similar shape or appearance properties making them hard to discriminate, e.g. a mug and a roll of toilet paper are alike in shape but only the former object affords pouring a liquid to. Thus, our Object Categorization System (OCS) integrates visual descriptors capturing different object properties such as appearance, color, shape and does this using both images (2D) and reconstructed stereo data (3D).

As shown in Fig. 1, we first build a single cue OCS for each feature descriptor which are then integrated for the final categorization. All single cue OCSs implement the following methodology: (a) data acquisition (Sec. III-A), (b) feature extraction (Sec. III-B), and (c) classification (Sec. III-C). The methods used to integrate these single cue OCSs will be described in Sec. III-D.

A. Scene Segmentation

Prior to categorization, object hypotheses are first generated using a multi-cue scene segmentation system [15]. The method relies on attentional mechanisms to direct cameras towards regions of interest, subsequently grouping areas close to the center of fixation as the foreground. Points of the disparity maps, computed using the Stable Matching [17], are then labeled as either the object (foreground), supporting plane (flat surface) or the background. Important aspect is that the system generates object hypotheses *without* relying on information about object category which is the common approach in the literature.

The segmented point cloud is further processed to remove outliers and equalize point density. We rely on the statistical outlier removal and voxel grid filters from the ROS PCL [18]. The resulting point cloud contains approx. 2000 points representing the visible part of the object. Our system does not require reconstruction of the whole object from

TABLE I

Features used for the Task Constraint Bayesian network.

Name	Dimension	States	Description
$task$	-	5	Task Identifier
$obcl$	-	7	Object Category
$size$	3	6	Object Dimensions
dir	4	15	Approach Direction (Quaternion)
pos	3	17	Grasp Position
$fcon$	11	3	Final Hand Configuration
$pshcv$	3	3	Grasp Part Shape Vector
coc	3	8	Center of Contacts
$fvol$	1	4	Free Volume

its partial view as in [19][20]. Such reconstruction methods often assume objects to be symmetrical which is not always the case.

B. Feature Extraction

The object representation is crucial for achieving robust categorization. Several descriptors have been proposed in the field of computer vision to encode object appearance (SIFT [21], textones [22]), color (opponentSIFT [23]) and contour shape (HoG [24]). Studies on 2D cue integration [25] show that contour- and shape-based methods are adequate for handling the generalization requirements needed for object categorization however they are not robust to occlusions. On the other hand, appearance- and color-based descriptors have been successfully applied in object (instance) recognition and detection [21], [22]. However, their performance drops significantly in case of clutter and illumination changes. In object retrieval and computer graphics, a number of 3D shape descriptors have been proposed [26]. Only a few of them are applicable to real 3D data that covers only the visible part of the object: spin images [27], RSD [19], FPFH [28],[29].

Motivated by the fact that the object representation should have high discriminative power, be robust to real world condition and diverse for cue integration, we extract from a segmented part of an image multiple 2D descriptors encoding different object attributes: appearance (SIFT), color (opponentSIFT), contour shape (HoG). The final object representation for 2D descriptors follows a concept of the spatial pyramid [30]. The 3D shape properties of an object are obtained by applying the FPFH descriptor [28] to each 3D point in the segmented point cloud. It was shown that the normal-based descriptors obtain high performance for the task [29]. To obtain the final object representation, a *bag-of-words* BoW model [31] is employed.

C. Classification

Motivated by the histogram-based object representation (BoW), we use for classification SVMs with a χ^2 kernel successfully applied in previous studies [23][24][20]. For the purpose of cue integration, we need information about the confidence with which an object is assigned to a particular class. Several studies were devoted to find confidence estimates for large margin classifiers [32], [33]. In principle, they interpret the value of the discriminative function as a distance of a sample to the optimal hyperplane. The closer the sample is to the hyperplane the lower is the probability (confidence) of a correct classification. In this work, we use the One-against-All strategy for M-class SVMs and the confidence measure for a sample $\mathbf{x}$ is calculated as [34]:

$$C(\mathbf{x}) = D_j * (\mathbf{x}) - \max_{j=1\ldots M, j\neq j*} \{D_j(\mathbf{x})\} \qquad (1)$$

where $D_j(\mathbf{x})$ is equal to the difference between the average distance of the training samples to the hyperplane and the distance from $\mathbf{x}$ to the hyperplane. Experimentally, this approach shown to be superior to the Platt's method [32].

D. Cue Integration

Various cue integration approaches have been applied to object recognition and categorization based on 2D data. These methods can be divided into: *low level integration* and *high level integration*. Low level integration operates directly on feature vectors. Due to the curse of dimensionality [35, p.170] its applications are mostly limited to the early work in object recognition [36]. High level integration have been shown to be more robust to noisy cues and is is commonly accomplished by an ensemble of classifiers or experts. The most common techniques include [37]: majority voting of classifiers [25] and methods based on algebraic combination of classifier outputs. The classifier outputs can be combined using linear [33] or nonlinear [38] techniques.

Our object categorization system takes a high level approach integrating evidences from the single cue OCSs. We use methods based on an algebraic combination of classifier outputs and we evaluate both the linear and nonlinear algebraic techniques.

In case of the linear techniques, the total support for each class is obtained as a linear weighted sum, product or max function $F(\cdot)$ of the evidences provided by individual classifiers. The final decision is made by choosing the class with the strongest support. Let us assume that d_{ij} is an evidence provided by classifier i for a category j, and w_i is a weight for classifier i (both are normalized to sum up to one for all L classifiers and M categories), then the class with the strongest support $j_0 \in \{1, \ldots, M\}$ is chosen as:

$$j_0 = \arg \max_{j=1,\ldots,M} \frac{F(d_{1j}, \ldots, d_{Lj}; w_1, \ldots, w_L)}{\sum_{j=1}^{M} F(d_{1j}, \ldots, d_{Lj}; w_1, \ldots, w_L)}. \qquad (2)$$

The weights $w_i|_{i=1,\ldots,L}$ are estimated during training. In this setup, the sum rule is equivalent to the Discriminative Accumulation Scheme (DAS) proposed in [33].

In case of the nonlinear techniques, we have used an approach where an additional SVM classifier is trained to model the relation between evidences provided by the different single cue OCSs [38]. The outputs from the single cue OCSs are concatenated to build a feature vector that is fed to the subsequent SVM classifier. During training, parameters of the nonlinear function $F(\cdot)$, equal to the classifier kernel function, are estimated. We have evaluated the performance of the following three nonlinear function: (a) radial basis function (RBF), (b) χ^2 function, and (c) histogram intersection.

Linear methods are simple and have low computational complexity. However, to infer weights $w_i|_{i=1,\ldots,L}$, an exhaustive search over parameter values is needed which becomes an intractable task for a large number of cues. The nonlinear methods owing to more complex function may better adapt to the varying properties of the cues. However, they also require a larger training dataset which may be infeasible for real world scenarios.

IV. EXPERIMENTAL EVALUATION

First, we present the dataset and experimental setup in Sec. IV-A and IV-B. Then, we study robustness of different

Fig. 4. Examples of objects used to create the database presented in Section IV-A. Different objects were chosen for each category in order to capture variations in appearance, shape and size within each class. The data for all the 140 objects can be viewed at our web site `http://www.csc.kth.se/~madry/research/stereo_database/index.php`.

(a) (b)

Fig. 5. Examples of imperfect segmentation in both 2D and 3D: (a) only a part of an object is detected, or (b) the segmentation mask contains background points (background points are marked in red).

2D and 3D descriptors in Sec. IV-C followed by a systematic evaluation of several 2D-3D integration strategies in Sec. IV-D. We then demonstrate grasp generation on novel objects based on categorical information. We also show how the grasp knowledge can be transfered between objects that belong to the same category. Finally, we study performance of the integrated system in realistic scenario for multiple objects, scenes and tasks in Sec. IV-E.

A. Database

Most of the available object categorization databases store only 2D image information [39] or 3D object structure [40], and other 2D-3D datasets [20], [41] contain unsuitable categories to demonstrate the task-directed grasping. We collected a new database with objects chosen from everyday categories. This is very challenging for a category-based, task-oriented grasping system. The dataset contains a number of objects that are similar in shape and appearance, but afford different tasks (e.g. ball/orange, orange/carrot, carrot/screwdriver). There are 14 categories: *ball, bottle, box, can, car-statuette, citrus, mug, 4-legged animal-statuette, mobile, screwdriver, tissue, toilet-paper, tube and root-vegetable*, each with 10 different object instances per category (in total 140 objects, examples of objects for each category are presented in Fig. 4). For each object, the 2D (RGB image) and 3D (point cloud) data were collected from 16 different views of the object (separated by 22.5°) using the 7-joint Armar III robotic head, see Fig. 2. To differentiate the object and background we used the active segmentation method [15] that generated good results in ca. 90% of cases. For some object categories, such as *car-statuette*, *animal-statuette* and *screwdriver*, segmentation was more challenging, see Figure 5. In order to evaluate performance of the categorization and grasp generation systems in the real environment, we collected data for 10 natural scenes. Five subjects were asked to randomly place between 10 to 15 objects from 14 different categories on a table. In the scenes, different lightning condition and occlusions of objects are present. Several scenes are shown in Fig. 10 and 12.

B. Experimental Setup

The database was divided into four sets used for: (1) training, (2) validation of OCS parameters, (3) validation of the cue integration parameters, and (4) testing. Objects were randomly selected for each set with the ratio 4:1:1:4 objects per category. In total, data for 56 objects were used for training and testing, and data for 14 objects for subsequent validations. Due to the fact that we aim to test performance of the system for the object categorization and not object instance recognition, an object that was presented to the system during the training phase was never used later to evaluate the performance.

For testing, we selected 8 views per object separated by 45° (Fig. 6 top row). We also used 8 images per object, however we varied a number of *unknown* viewpoints between 0 and 8. Fig. 6 (bottom row) presents a test setup where half of the views is unknown. This setup reflects the best the real condition and we called it *Setup-50*. The results are reported for a single object view and information provided by different views was not fused. To average the results each experiment was repeated five times for randomly chosen object instances. We report the average categorization rate and standard deviation (σ).

C. Feature Selection for Object Categorization

We built four identical single cue OCSs, one for each descriptor, to evaluated performance of descriptors encoding different object properties: appearance (SIFT), color (opponentSIFT), contour shape (HoG) and 3D shape (FPFH). The SIFT and opponentSIFT were extracted using a grid detector, and HoG descriptor using the Canny edge detector. The final object representation for the 2D descriptors follows a concept of the spatial pyramid, and for the 3D descriptor BoW model.

In order to assess the performance of the descriptors under different viewpoints, we varied a number of *unknown* viewpoints in the test set between 0 and 8. The results are illustrated in Fig. 7. All 2D descriptors obtained rather high categorization rate when the viewpoint was known (0 views), but the performance dropped significantly when as the viewpoint varies. The highest performance was obtained for opponentSIFT which indicates that color information is less influenced by the viewpoint changes than shape informa-

Fig. 6. *Setup-50*. Objects from eight different viewpoints selected to train the system (top row) and evaluate its performance (bottom row).

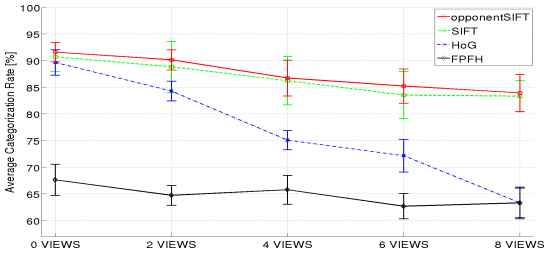


Fig. 7. Performance of descriptors under varying viewpoint.

TABLE II

RESULTS FOR THE FEATURE SELECTION EXPERIMENTS FOR *Setup-50*.

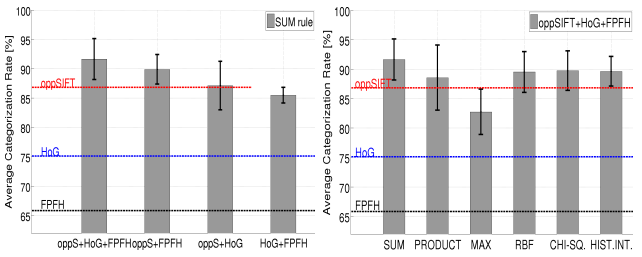
Descriptor	SIFT	opponentSIFT	HoG	FPFH
Av.Categ.Rate	86.2%	**86.8%**	75.1%	65.8%
σ	4.5%	3.3%	1.8%	2.7%

tion (HoG). The 2D descriptors yielded higher categorization rates than the 3D descriptor. It can be related to the quality of stereo data. However, the performance of the 3D descriptor is only slightly affected by the viewpoint changes. Additionally, we attach the numerical results for *Setup-50* in Table II.

D. Cue Integration

In this section, we present results from combining 2D and 3D categorization. The best performance of **92%** was obtained for integration of the three descriptors: opponentSIFT+HoG+FPFH using the linear combination method. When comparing to the best single cue OCS (based on opponetSIFT), the combination of 2D and 3D features improved performance of the system in average by **5%**. The confusion matrix obtained for this experiment is presented in Figure IV-D (d). The results show that capturing diverse object properties (appearance, contour shape and 3D shape) and integration of information from different visual sensors (2D and 3D) not only significantly improve robustness of the categorization system, but are essential to discriminate between similar objects that afford different tasks. The integrated system is able to correctly classify objects that are alike in shape or appearance, but are to be used for different purpose. For example, it correctly categorizes objects of similar: (a) shape, such as *screwdriver* and *root-vegetable* where only the former can be used as a tool, *ball* and *citrus* where only the former affords playing, or *mug*, *can* and *toilet-paper* where only the former affords pouring a liquid; (b) appearance: *citrus* vs. *root-vegetable*, *bottle* vs. *can*. Such classification is very challenging for a system based on a single cue.

1) Detailed Results: The categorization results for different choice of features and cue integration methods are presented in Fig. 9. The results confirm that descriptors need to be complementary, i.e. capture different object properties and originate from different sensors. The best categorization rate is obtained for fusion of all three features (opponentSIFT+HoG+FPFH). The second best for the combination of descriptors that capture different object attributes and originate from different channels: 2D color and 3D shape descriptor (opponentSIFT+FPFH). Further, for the color and shape descriptor from the same channel



(a) Features (b) Integration methods

Fig. 9. Average categorization rate for: (a) different pairs/triples of features (for linear combination method, sum rule), (b) different linear and nonlinear algebraic combination methods (for opponentSIFT+HoG+FPFH).

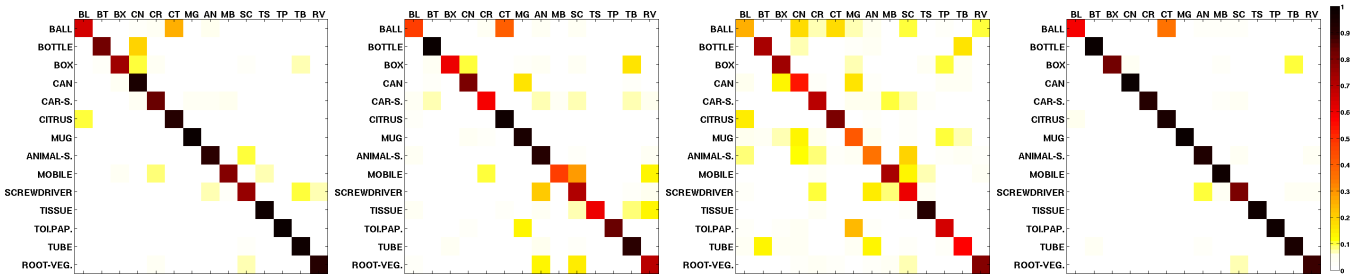
(opponentSIFT+HoG) and for the two shape descriptors (HoG+FPFH). The same trend in performance is observed for both the linear and nonlinear combination methods. This is evidence of selective properties of our system.

In case of the linear algebraic methods, we tested the weighted sum, product and max rule. For all combinations of features, the approach based on the sum and product rule improved the performance of the system in comparison to the best single cue OCS (based on opponentSIFT), and the sum rule was superior to the product rule. The max rule that in case of two classifiers is equivalent to the majority voting, yielded the lowest categorization rate further supporting the notion of complementarity. In case of the nonlinear algebraic methods, we evaluated the RBF, χ^2 and histogram integration functions. All the nonlinear functions provided a comparable performance. In our study, the linear algebraic integration methods outperformed the nonlinear methods. A small set of data was used to train the SVM classifier for the nonlinear methods. We can draw the conclusion that in case of a limited amount of data, the simpler fusion methods are more efficient.

2) Natural Scenes: We evaluated performance of the 2D-3D integrated OCS on 10 natural scenes where each contains 10-15 objects randomly placed on a table. To categorize objects, we chose the best classifier trained following the procedure from Section IV-B. For each object in the scene, we estimated a confidence vector over the 14 object categories. The final label was found by choosing a category with the highest support. We obtained a high categorization rate of **91.7%**. The categorization results for a few scenes together with a confidence vector for each object are presented in Fig. 10. The confidence values reflect the same trend as presented in the confusion matrices in Fig. IV-D. The most difficult remained the differentiation between the *ball* and *citrus* category (see Scene: 3, Object: 8). *Mug*s are likely to be confused with *can*s when a part of an object is not visible due to occlusion (S: 3, O: 11) or inaccurate segmentation (S: 2, O: 6). We showed that the system is capable to operate in a very challenging scenario.

E. Object Category-based Task-constrained Grasping

In this section, we summarize the results of an integrated system considering categorization for task-constrained object



(a) opponentSIFT (b) HoG (c) FPFH (d) opponentSIFT+HoG+FPFH

Fig. 8. Confusion matrices obtained for: (a) color (opponentSIFT), (b) contour shape (HoG), (c) 3D shape (FPFH) descriptor, and (d) integrated opponentSIFT+HoG+FPFH (linear combination method, sum rule). The images are best viewed in color.

grasping. Our experimental scenario considers multiple objects grasp planning constrained by the assigned tasks. In addition, we take the robot embodiment into account. The robot is presented with a scene containing several unknown objects, see Fig. 11. First, object hypothesis are segmented from the background. Secondly, each hypothesis is fed into our object categorization system. In the given scene, 13 objects are found and they are all correctly classified. The categorization confidence value of each object provides evidence for which object to grasp first.

Next, given the assigned task, the robot needs to decide: (1) which object should be grasped, and (2) how to grasp it to fulfill the task requirements. For this purpose, we use the embodiment-specific task constraint model. The model is trained on a grasp database that includes stable grasps generated on a set of synthetic object models using the hand model from the humanoid robot Armar [42]. The object models are extracted from the Princeton Shape Benchmark [43] (3-8 models per category), each of which includes 4 different object shapes scaled to 2 sizes – small and average. Five tasks were labeled: *hand-over*, *pouring*, *dishwashing*, *playing* and *tool-use*. The total training set includes 1227 cases with 409 cases per grasping task.

1) Grasp Transfer: Our goal is to infer the most suitable grasp position pos for an object in the 3D scene given the assigned task $task$ and the categories of the objects $obcl$. Since, the BN allows to infer local distribution of a small set of variables, based on partial or complete observation of others, we can create a likelihood map on a set of grasp position around each object, i.e. $P(pos|obcl, task)$. An example of such a grasp map for an $obcl = mug$ and $task = pouring$ is presented in Figure 11. The point that has the highest P (indicated by the brightest color) implies the best grasp position for the task.

The pos variable in the BN is represented in the synthetic object local coordinate system. In order to transfer grasp information to an arbitrary object in the scene, it is necessary to convert the pos data from the local object frame to the world coordinates. This transformation requires the knowledge of object size, position and orientation in a scene. In this paper, we assume that the orientation of the object is known. The size and position determined by estimating a minimum bounding sphere for the the filtered 3D point cloud (outliers and background points are removed). We assume that a diameter of the sphere corresponds to the longest object dimension. Several examples of grasp transfer to the real objects are presented in Fig. 11. For each object in the scene that was classified as a *mug*, the grasp map is presented in the front (camera), top and back view. It is important to note that by transferring the grasp map, we are able to generate grasp points for the back (not visible) part of an object without reconstructing the full object shape.

2) Task-constrained Grasping in a Real Scene: Fig. 12 shows the results of the experiment for natural scenes. We show the likelihood maps for each object using colored sample points of $P(pos|task, obcl)$ and for five tasks: *hand-over*, *tool-use*, *pouring*, *playing* and *dishwashing*. For each scene, the joint probability of an object and task $P(obcl, task)$ is used together with the categorization confidence to specify which objects should be grasped first giving priority to objects that are categorized with a high confidence and affords a task (have high $P(obcl, task)$).

In Fig. 12 (Scene 3, Column 2), we see that for the *pouring* task, the likelihoods of the sample points around mugs and bottles are clearly higher than for other objects indicating that they are the only objects affording the task. Similarly, *screwdriver*s are the only objects that can be used as a tool (Sc. 3, Col. 3), and *cars* and *balls* to play (Sc. 2, Col. 4). For the *hand-over* task, all objects have high likelihood. This indicate that using the object category information and the task constraint BN, we can successfully select the object according to their task affordance.

For the object that affords *pouring*, for example mugs in Scene 3 (Col. 2, Objects 6 and 9) the likelihood maps show darker color on the top of the object. This is because the robot hand should not block the opening of an object when pouring a liquid. When using the screwdriver as a tool (Sc. 2, Col. 3, Obj. 2), the network favors the position around the tip of the screwdriver whereas leaving the handle part for regrasp.

V. CONCLUSIONS AND FUTURE WORK

Robots grasping objects in unstructured environments need the ability to select grasps for unknown objects and transfer this knowledge to other objects based on their category and functionality. Although for pure categorization 2D information may be sufficient, 3D information is a must for grasping and manipulation of objects and can thus also be used for

categorization. The categorization system is integrated with a task constrained model for goal-directed grasp planning. We showed that the object category information can be efficiently used to infer the task affordance of the observed objects. The proposed system allows for reasoning and planning of goal-directed grasps in real-world scenes with multiple objects.

We have presented a 2D-3D object categorization system that is built upon an active scene segmentation module. The system allows generating object hypotheses and segmenting them from the background in real-time. Experimental evaluation showed that the proposed system achieved high categorization rate (up to 92%), significantly better than the classic single cue SVM for the same task. Moreover, cue integration method proposed in this paper is very efficient and capable to model situations where limited amount of data is available. The results show that capturing diverse object properties (appearance, contour shape and 3D shape) and integration of information from different visual sensors (2D and 3D) not only significantly improve robustness of the categorization system, but are essential to discriminate between similar objects that afford different tasks.

One avenue for the future research is the integration of the proposed system with the on-line stability estimation system proposed in [44]. The aim will be to condition the choice of grasps based on the perceptions available to a robot prior to and while lifting and transporting an object.

REFERENCES

[1] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, "Contact-reactive grasping of objects with partial shape information," in *IROS*, 2010.

[2] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann, "Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot," in *ICRA*, 2010.

[3] J. J. Gibson, "The theory of affordances," in *Perceiving, Acting, and Knowing*, 1977.

[4] J. G. Greeno, "Gibson's Affordances," *Psychological Review*, vol. 101, no. 2, pp. 336–342, 1994.

[5] G. Fritz, L. Paletta, R. Breithaupt, E. Rome, and G. Dorffner, "Learning predictive features in affordance-based robotic systems," in *IROS*, 2006.

[6] E. Sahin, M. Cakmak, M. Dogar, E. Ugur, and G. Ucoluk, "To afford or not to afford: A new formalization of affordances towards affordance-based robot control," *ISAB*, vol. 15, no. 4, pp. 447–472, 2007.

[7] A. Stoytchev, "Learning the affordances of tools using a behavior-grounded approach," in *Affordance-Based Robot Control*. LNAI, 2008, pp. 140–158.

[8] D. Kraft, E. Baseski, M. Popovic, N. Kruger, N. Pugeault, D. Kragic, S. Kalkan, and F. Worgotter, "Birth of the object: Detection of objectness and extraction of object shape through object action complexes," *IJHR*, vol. 5, no. 2, pp. 247–265, 2008.

[9] H. Kjellstrom, J. Romero, and D. Kragic, "Visual object-action recognition: Inferring object affordances from human demonstration," *CVIU*, vol. 114, no. 1, pp. 81–90, 2011.

[10] L. Stark and K. Bowyer, "Achieving generalized object recognition through reasoning about association of function to structure," *PAMI*, vol. 13, pp. 1097–1104, 1991.

[11] S. Oh, A. Hoogs, M. Turek, and R. Collins, "Content-based retrieval of functional objects in video using scene context," in *ECCV*, 2010.

[12] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning Task Constraints for Robot Grasping using Graphical Models," in *IROS*, 2010.

[13] D. Song, C.-H. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for bayesian network structure learning in robot grasping," in *ICRA*, May 2011.

[14] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Embodiment-Specific Representation of Robot Grasping using Graphical Models and Latent-Space Discretization," in *IROS*, 2011.

[15] M. Bjorkman and D. Kragic, "Active 3D scene segmentation and detection of unknown objects," in *ICRA*, May 2010.

[16] K. Huebner, "BADGr - A Toolbox for Box-based Approximation, Decomposition and GRasping," in *Workshop on Grasp Planning and Task Learning by Imitation (IROS)*, 2010.

[17] R. Sara, "Finding the largest unambiguous component of stereo matching," in *ECCV*, vol. 2, May 2002.

[18] *ROS Point Cloud Library*, http://www.ros.org/wiki/pcl, Last visited: Feb 2011.

[19] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3D modelling of novel objects from a single view," in *IROS*, 2010.

[20] D. Marton, Z-C.and Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz, "Hierarchical object geometric categorization and appearance classification for mobile manipulation," in *Humanoids*, 2010.

[21] G. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 91–110, 2004.

[22] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*, 2008, pp. 1–8.

[23] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1, 2005, pp. 886–893.

[25] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *CVPR*, 2003, pp. 409–415.

[26] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," in *SMI*, 2004, pp. 145–156.

[27] A. Johnson, "Spin-images: A representation for 3-D surface matching," Ph.D. dissertation, Carnegie Mellon University, 1997.

[28] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *ICRA*, May 2009.

[29] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *IROS*, 2010.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, vol. 2, 2006.

[31] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision (ECCV)*, 2004, pp. 1–22.

[32] J. C. Platt, "Probabilistic outputs for SVMs and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, pp. 61–74, 1999.

[33] M. E. Nilsback and B. Caputo, "Cue integration through discriminative accumulation," in *CVPR*, 2004.

[34] A. Pronobis and B. Caputo, "Confidence-based cue integration for visual place recognition," in *IROS*, 2007, pp. 2394–2401.

[35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, 2001.

[36] J. Matas, R. Marik, and J. Kittler, "On representation and matching of multi-coloured objects," in *ICCV*, 1995, p. 726.

[37] R. Polikar, "Ensemble based systems in decision making," *Circuits And Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[38] A. Pronobis, O. M. Mozos, and B. Caputo, "SVM-based discriminative accumulation scheme for place recognition," in *ICRA*, 2008.

[39] *Caltech101 Database*, Last visited: Feb 2011, http://www.vision.caltech.edu/Image_Datasets/Caltech101.

[40] *Princeton Shape Benchmark*, http://shape.cs.princeton.edu/benchmark, Last visited: Feb 2011.

[41] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *ICRA*, 2011.

[42] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Humanoids*, 2006.

[43] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *SMI*, 2004, pp. 167–178.

[44] Y. Bekiroglu, K. Huebner, and D. Kragic, "Integrating grasp planning with online stability assessment using tactile sensing," in *ICRA*, 2011.

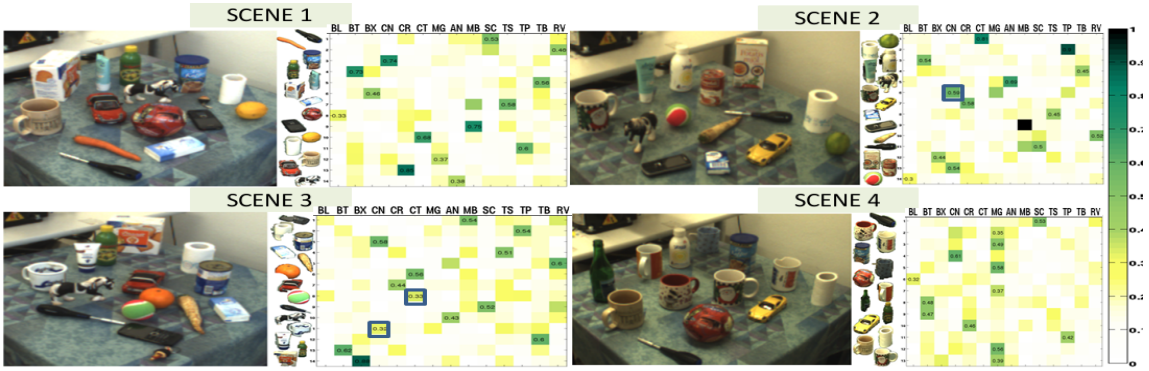


Fig. 10. Categorization results for natural scenes. For each object in a scene, confidence values over 14 categories are shown. All objects were correctly classified except three objects marked using a blue square in confidence vector.

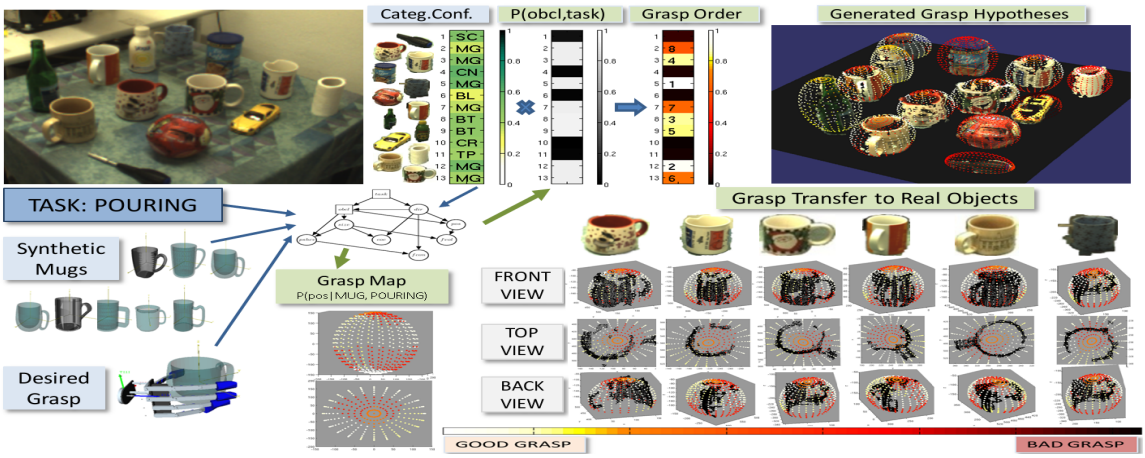


Fig. 11. Grasp transfer from a synthetic object model to real objects in a scene. The grasping points with a high value of $P(pos|obcl, task)$ (good grasping points) are represented by bright color in the heat maps.

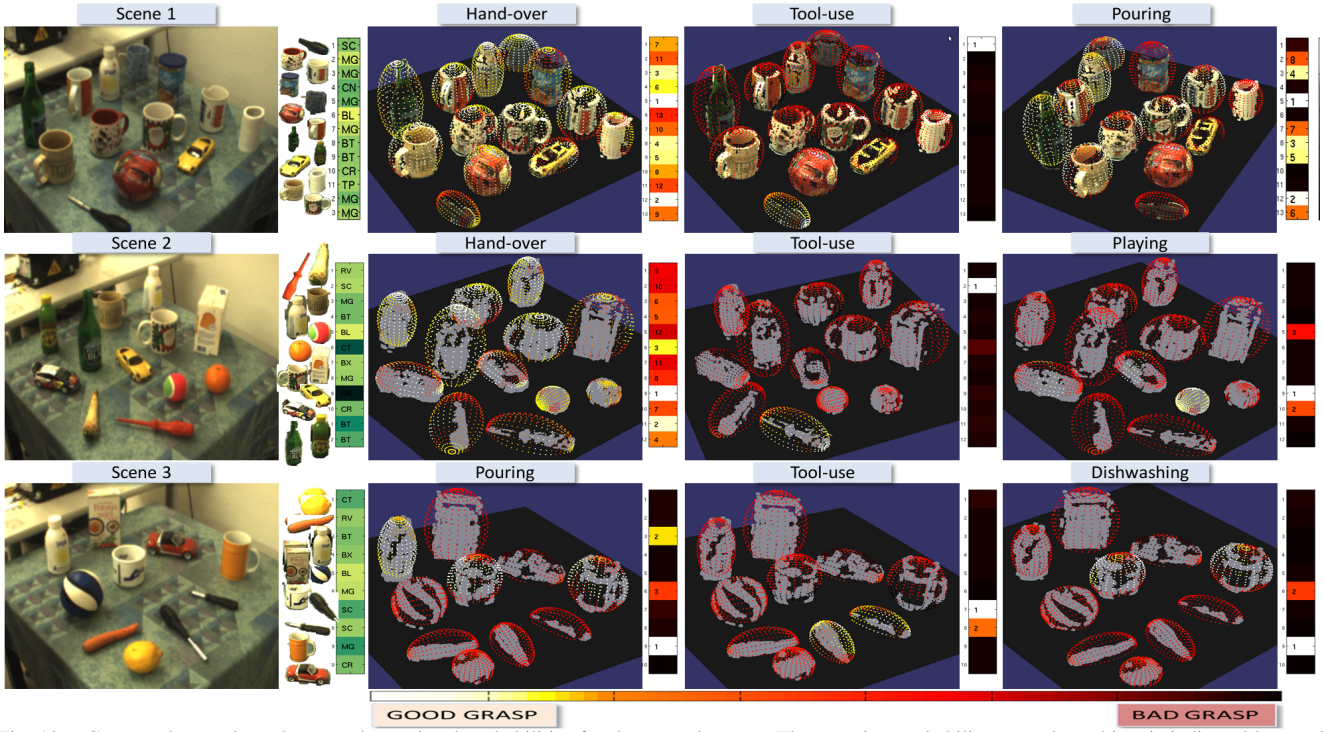


Fig. 12. Generated grasp hypotheses and associated probabilities for the natural scenes. The grasping probability around an object is indicated by a color of the point (the brighter is the point, the higher is the probability). For each scene, we specify which objects should be grasped first (bar on the right side of a scene grasp map). Objects in Scene 2 and 3 are displayed in gray color to provide a better contrast for grasp maps. The images are best viewed in color. For the accurate 3D information, we kindly direct the reader to our web site `http://www.csc.kth.se/~madry/research/madry12icra` where the movies with the experimental results are available.