2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

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 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].

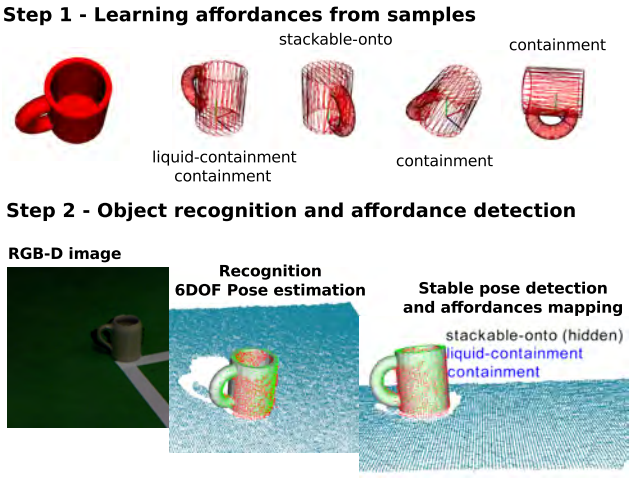


Fig. 1. The two steps of learning and detecting hidden and non-hidden 0-order affordances in real scenes: **Step 1:** affordances are learned using a pool of binary classifiers on the full-3D representations of the objects to be recognized using the methodology presented in Sec. III-E. **Step 2:** objects 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 might roll on a planar under an appropiate external force.
- *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 a 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).

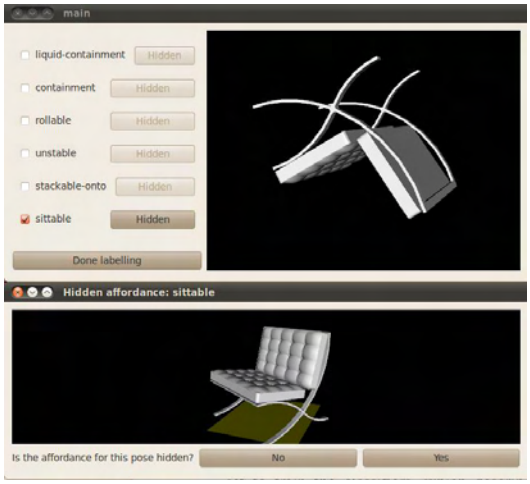


Fig. 2. Screenshot of the labeling tool. In the top figure, the user is given a CAD model to label in terms of its 0-order affordances. When the "hidden" button for a specific affordance is pressed, a window (at the bottom) appears allowing the user to input whether the current affordance is hidden or 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.

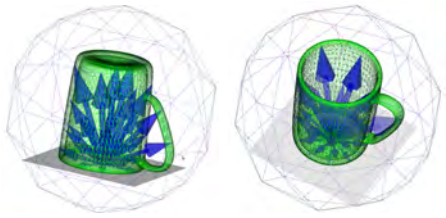


Fig. 3. Visualization of a Spherical Extent Descriptor for a mug found in a upside-down pose (left) and upright (right).

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.

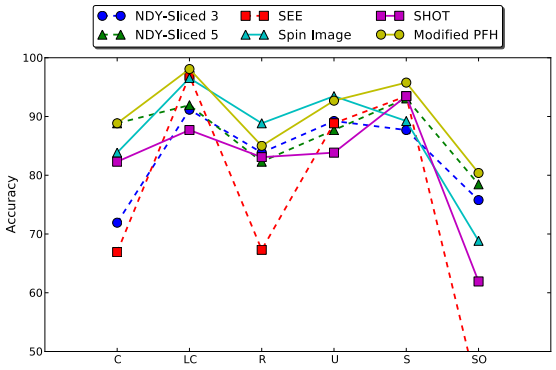
We have used the same training set for all experiments shown throughout this paper. In particular, it is composed of 43 CAD models selected from the *Google Warehouse* dataset [2], which include the following affordance-rich object categories: chairs, sofas, bottles, mugs, bowls, stools, office chairs,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

E. Learning affordances on CAD models - Evaluation

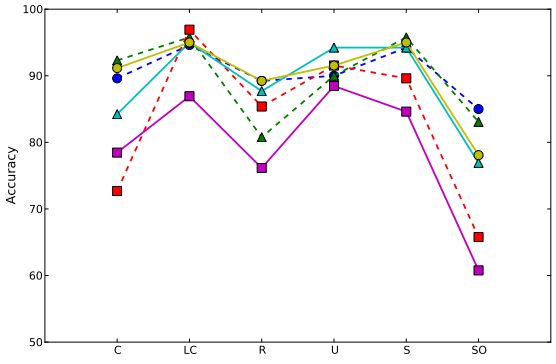
This subsection illustrates an experimental evaluation aimed at demonstrating affordance detection on CAD models based on the descriptor and learning techniques previously introduced. Also, the goal here is to evaluate what is the best performing descriptor-classifier combination among those being evaluated. 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)

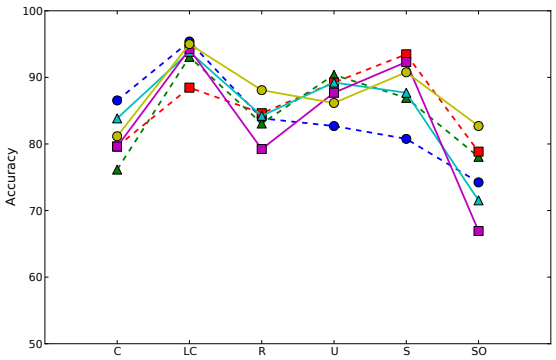
[2] http://sketchup.google.com/3dwarehouse/



(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 typically 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 the 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 therefore, 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

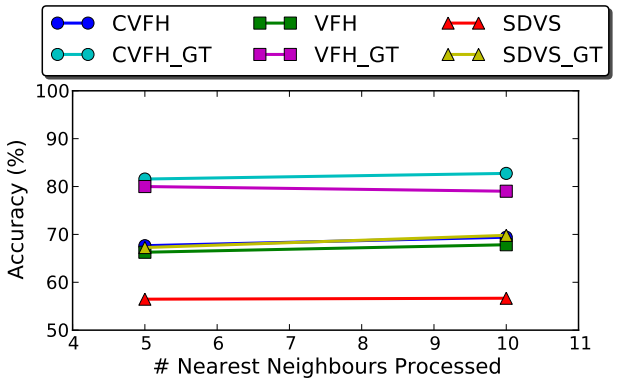


Fig. 7. Evaluation of the accuracy in the affordance detection using several descriptors and as a function of the number of nearest neighbours that are post-processed. All nearest neighbours are post-processed in the same way.

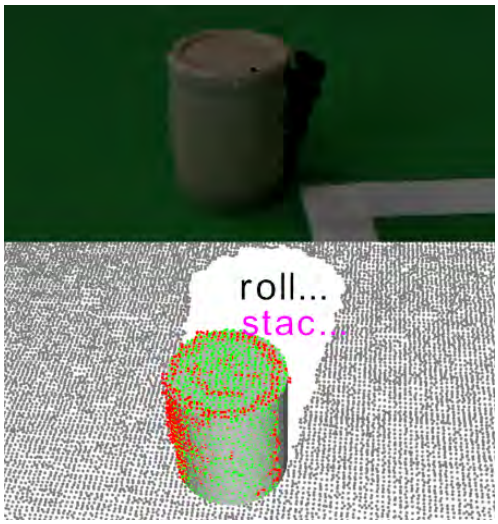


Fig. 8. A mug seen from a viewpoint where the handle is not visible and in an upside-down pose. The mug gets recognized as a cylinder and hidden affordances are incorrectly detected. Green points depict the view rendered from the CAD model, while red points depict the segmented clusters in the current scene. Observe how both red and green points are accurately registered and therefore its challenging to decide if the mug is in fact a mug in an upside-down pose or a cylinder.

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 *Workshop: Active Semantic Perception and Object Search in the Real World, IROS*, 2011.

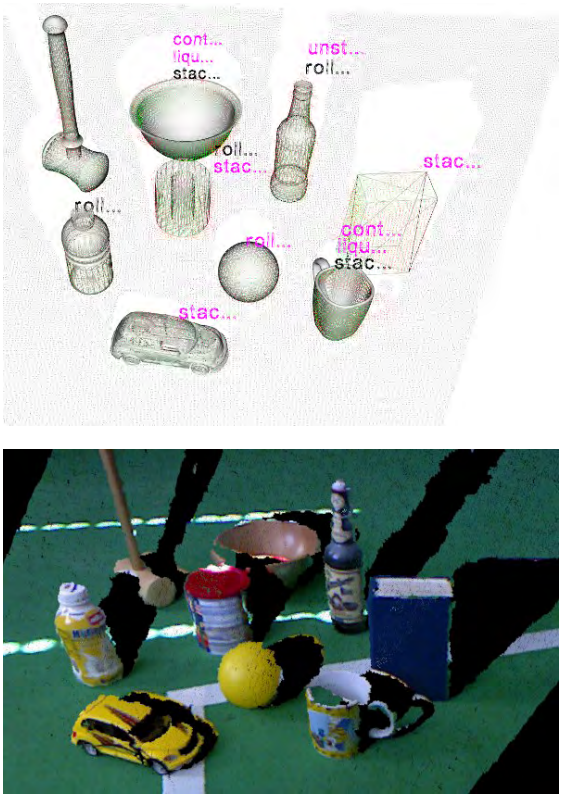


Fig. 9. Top: A scene recognized with CVFH. The CAD models are overlapped on the scene together with the 0-order affordances that each object provides. Hidden affordances are shown in black and non-hidden in magenta. Green points represent the best-matching rendered view from the training set, while red 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 *Workshop: 3rd IEEE Workshop on 3D Representation and Recognition, ICCV*, 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.