Affordance Estimation For Vision-Based Object Replacement on a Humanoid Robot

Wail Mustafa[1], Mirko Wächter[2], Sandor Szedmak[3], Alejandro Agostini[4],
Dirk Kraft[1], Tamim Asfour[2], Justus Piater[3], Florentin Wörgötter[4] and Norbert Krüger[1]

[1] The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark.
[2] Institute for Anthropomatics, Karlsruhe Institute of Technology, Karlsruhe, Germany.
[3] Institute of Computer Science, University of Innsbruck, Innsbruck, Austria
[4] Third Institute of Physics and BCCN, University of Göttingen, Germany.

Abstract

In this paper, we address the problem of finding replacements of missing objects, involved in the execution of manipulation tasks. Our approach is based on estimating functional affordances for the unknown objects in order to propose replacements. We use a vision-based affordance estimation system utilizing object-wise global features and a multi-label learning method. This method also associates confidence values to the estimated affordances. We evaluate our approach on kitchen-related manipulation affordances. The evaluation also includes testing different scenarios for training the system using large-scale datasets. The results indicate that the system is able to successfully predict the affordances of novel objects. We also implement our system on a humanoid robot and demonstrate the affordance estimation in a real scene.

1 INTRODUCTION

To perform complex plans on robots in not fully known environments, a large variety of tasks need to be addressed and also different knowledge sources need to be exploited. The usual approach in these scenarios is to manually define the list of objects that would be needed for the plan generation and execution of each possible task. However, these scenarios are not fully predictable. Objects that were originally considered for a given task execution might be missing, preventing the execution of the task and limiting the autonomy of the robot. The alternative would be to provide the robot with mechanisms for the automatic prediction of the functionalities of objects so they can be used for eventually replacing the missing ones. For example it needs to be estimated, with a certain confidence, whether an object can be used as a container or whether it can be used for stirring.

For this kind of prediction, linguistic knowledge could be used for example by looking for similar objects on the Internet and finding on the web-pages, where these objects occur, frequently associated verbs [1]. Another knowledge source for associating such functional roles can be shape information, which is the focus of this paper. More specifically, we utilize the shape representation, based on global 3D descriptors, introduced in Mustafa et al. [2] to predict functional properties of objects.

Following Gibson's definition [3], those properties are referred to as 'affordances'. He defined affordances as the latent "action possibilities" available to an agent, given their capabilities and the environment. This means for a robot, a bowl affords pouring into and a knife affords cutting with. This also means that an object can afford several actions and those affordances can be related. For instance, the bowl affords also stirring items in it. Vision-based replacement allows us also to find a viable replacement in unusual situations. For example, the spoon could also be used for cutting if no cutting tool is available.

Taking this into account, our affordances are learned by using JointSVM [4], which allows us to learn efficiently multi-labl problems while considering the interdependence of the labels. In our case, labels correspond to object affordances. The JointSVM also provides confidences (scores) for the predicted affordances. This allow the system to make replacement decisions based on reliable estimations.

In this paper, we focus on the manipulation actions that can be executed by the humanoid robot ARMAR [5] (see Fig. 4a). Concretely, we consider 11 affordances (TABLE 1) with which we label objects used to train the JointSVM. Considering the large variety of objects that may occur in the scene and the limited set of affordances we address as well as their interdependence, the knowledge the robot needs to develop has to be built from large and diverse sets of objects. This is important in order to derive associations between affordances and abstract shape attributes. For training our system, we make use of the availability of large 3D benchmark datasets of objects, as they provide an easy access to data.

ISBN 978-3-8007-4231-8

We give quantitative results on a test set (called Kitchen set) of 15 objects (Fig. 1) used by ARMAR for kitchen-related manipulation tasks. This quantification takes into account the confidences associated with the affordances estimated by the JointSVM. We compare different scenarios of training. We also evaluate how reliable the prediction confidence is, as a way of estimating affordances (i.e. by thresholding). The results show that our system is able to correctly predict the affordances. Also, the usefulness of the confidence estimate is confirmed. Finally, this estimation mechanism is implemented in this robot and can be invoked by the planner during execution. We also demonstrate this mechanism in a real scene.

This paper is structured as follows. Related work is discussed in Sec. 2. Next, we give a description of affordance estimation method we use in Sec. 3. In Sec. 4, we describe the evaluation procedure and discuss the results. Finally, we conclude in Sec.5.

2 Related work

The automatic estimation of object affordances was used to increase the autonomy of robots using planning architectures that search for replacements when some of the objects needed for plan execution are missing. Agostini et al. [6] introduced an architecture in which a logic-based planner is used to generate a plan from a prototypical problem description. If any of the objects involved in the generated plan is missing, the architecture searches for replacements with similar affordances. This is done by means of a *repository of objects and attributes with roles* (ROAR) [7], which is an intelligent database in which objects can be retrieved by their affordances, and the affordances of novel objects can be inferred. Once replacements are found, the plan is updated accordingly and then executed.

Another method for object substitution for the execution of a logic-based plans is presented in [8]. For finding replacements of missing objects, two elements are defined: a set of classes where attributes and affordances of objects are encoded using a logic-based notation; and a *conceptual space*, a multidimensional space of features such as shape and color intensity, used to evaluate similarities between objects.

In [9] a *categorical knowledge* is introduced, which, similarly to the object classes introduced in [8], encodes categories of objects that can be used to find substitutions. Categorical knowledge is included in the planning domain definition to avoid the *early commitment* problem, i.e. when objects required for plan execution are not found during a plan execution monitoring process.

Affordance estimation has been studied in computer vision and robotic using different approaches. Early work on affordance estimation followed a function-based approach to object recognition for 3D CAD models of objects such as chairs [10].

Another approach considers affordances of objects as a function of interactions, and jointly model both object interactions and activities. For instance, Koppula et al. [11] introduced a model based on Markov Random Field using 3D geometric properties, computed between a tracked human and an object, as features to describe interactions.

Other researchers looked at the problem by defining a subset of attributes that can be considered as transferable knowledge to predict new categories [12, 13, 14]. For example, by learning 2D shape and color patterns, a system was introduced in [15] to recognize attributes for novel object.

Our approach is based on classifiers trained in advance on object-wise features in a way similar to bag-of-words (BoW) approaches for object categorization [16]. In fact, affordances can be regarded as categories if objects are assumed to have single and distinct functional attributes. Following that, methods such as the Hierarchical Matching Pursuit (HMP) [17], which is shown to achieve a state-of-the-art performance on RGB-D benchmarks, can be utilized.

In [18], Myers et al. introduced a local approach to estimating affordances for object parts. In their approach, the image is segmented into superpixels serving as part candidates to which a classier predicts the affordances. In contrary, the approach we use in this paper is based on object-wise, global descriptors derived from full 3D features. Furthermore, the approach employs a multi-label classifier to predict the affordances. This approach was proposed and investigated on a large dataset of objects as generic approach for multi-label object categorization in our previous work [19]. In this work, we go beyond the work in [19], namely by: i) extending the classifier's output to include prediction score, allowing for a confidence-based decision making, ii) testing the system for manipulation affordances on kitchen-related set of objects, and iii) comparing the use of different benchmark object datasets to train the system.

3 Vision-based affordance estimation

Our vision-based affordance estimation approached is composed of two steps: i) extracting object descriptors and ii) using a multi-label learning method for learning and predicting the affordaances. In the following, the two steps are described in more detail.

3.1 Histogram representation for object description

In this paper, object shapes are described using histograms of *relations* between pairs of 3D features. Using the RGB-D data obtained form Kinect sensor, the scene is first segmented. For each segment, planar 3D surface features, hereafter called 3D texlets (see [20]), are extracted. A 3D texlet contains both position and orientation, and provides absolute information (relative to an external reference frame) of objects in 3D space. To describe an object, we compute a set of pairwise relations from all pairs of texlets belonging to the object. One important aspect of relations

ISBN 978-3-8007-4231-8

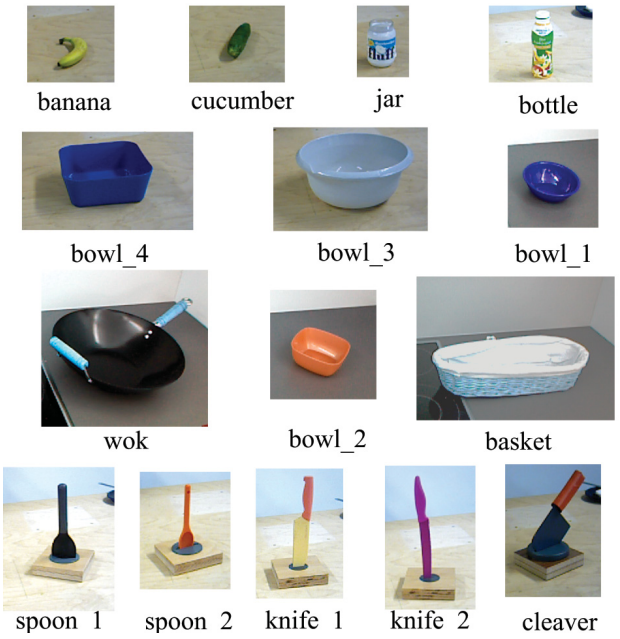


Figure 1 Kitchen Set. Objects considered in the tests as unknown and can potentially replace, based on their affordances, objects expected in an execution plan.

is that they transform an absolute pose-variant representation into a relative pose-invariant one, leading to highly discriminative descriptors [2].

In this paper, geometric relations were defined by two attributes: angle and scale-invariant distance (i.e, normalized relative to the object size) computed between 3D texlets. The distance relation is chosen to be scale-invariant because what defines a object affordance is usually independent of scale. The final object descriptor is obtained by binning these two relations in a 2D histogram, which model the distributions of the relations in fixed-sized feature vectors while considering their co-occurrence. According to previous investigations [2], the binning size is set to 12 in both dimensions resulting in a features vector of 144 dimensions.

3.2 JointSVM for multi-label learning

In the following, we describe the JointSVM method with the focus on the derivation of the confidence values.

3.2.1 JointSVM

JointSVM was developed with a particular focus on the interdependencies within outputs. Essentially, JointSVM is equivalent to Structural SVM (SSVM) [21], which is an extension of SVM for predicting structured outputs, with a linear output kernel plus a regularization term on the kernel [4]. Therefore, a linear kernel on the outputs is automatically learned to capture the interdependencies within outputs. Furthermore, if prior knowledge about the interdependencies is available, a user-specified output kernel can be straightforwardly mounted in JointSVM as well. Interestingly, the computational complexity of JointSVM is almost the same as a single SVM, in contrary to the exponential complexity in SSVM.

As input kernels, we chose polynomial kernels based on previous tests [19]. The estimation of the kernel parameters is embedded in a cross-validation step, performed prior to training. Cross-validation also includes the estimation of the JointSVM's internal parameters.

3.2.2 Confidence estimation in JointSVM

The output predicted by the JointSVM is given by an indicator vector, a set of binary labels, defined on the objects appearing on a scene. Since the learner handles the full output vector as one entity, i.e. simultaneously predicts all object labels, the component wise confidences requires a special care.

Our approach is based on the assumption that if the predicted vector falling close to a known label vector, then the confidence should be high, close to 1, otherwise it is low, close to 0. This provides us an error measure to which we can assign a probability distribution.

To implement this idea we need two types of predictions. Let the training examples, outputs and inputs, be denoted by $\mathscr{Y}_{training} = \{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$, $\mathscr{X}_{training} = \{x_1, \ldots, x_m\}$ respectively. These examples provide evidences to our estimation. The first type of prediction, "raw", is given by solving the optimization problem of the JointSVM on the available training data, and based on the *Generalized Representer Theorem*, see the derivation in [22], we have for a new unseen x:

$$\mathbf{y}_{raw}(x) = \sum_{i=1}^{m} \alpha_i \mathbf{y}_i \overbrace{\langle \phi(x_i), \phi(x) \rangle}^{K^X(x_i,x) \text{ input kernel}} . \tag{1}$$

The *raw* prediction can be further inferred with the available training examples which yields the second type of prediction,

$$\mathbf{y}_{inferred}(x) = \arg \max_{y \in \mathscr{Y}_{training}} \sum_{i=1}^{m} \alpha_i \overbrace{\langle \mathbf{y}, \mathbf{y}_i \rangle}^{K^Y(y,y_i) \text{ output kernel}} \overbrace{\langle \phi(x_i), \phi(x) \rangle}^{K^X(x_i,x)} \tag{2}$$

where we try to find the most similar training example to a *raw* prediction where the similarity is given by the inner product between a *raw* prediction and the known outputs.

The error can be computed by taking the difference between the "raw" and the "inferred" predictions. Since the raw prediction could contain some bias in predicting the error, we apply a linear regression on the raw prediction to estimate the inferred one to reduce the effect of that bias.

$$(\gamma^*, \mu^*) = \arg \max_{\gamma, \mu} \sum_{x \in \mathscr{X}_{training}} \| \mathbf{y}_{inferred}(x) - \gamma \mathbf{y}_{raw}(x) - \mu \|_2^2 \tag{3}$$

Finally the error is estimated by

$$\eta(x) = \mathbf{y}_{inferred}(x) - \gamma^* \mathbf{y}_{raw}(x) - \mu^* \tag{4}$$

At the end, a probability value can be assigned to all the components of the error vector. It relies on the assumption that those components follow logistic distribution with

ISBN 978-3-8007-4231-8

fixed expected value $\mu_{logistic} = 0.5$ and scale parameter $\sigma = 0.1$ whose meaningful value can be derived by cross-validation. The probability representing the confidence is given to a new test item x by this expression:

$$p(x) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{\eta(x) - \mu_{logistic}}{2\sigma}\right). \quad (5)$$

Applying logistic distribution has advantages, it approximates well the Gaussian distribution, but is computationally much less demanding, see details in [23].
The output of joint SVM is given by:

$$\mathbf{y}(x) = p(x)\mathbf{y}_{inferred}(x). \quad (6)$$

where each entry of the output represents one affordance, expressed by its associated confidence.

4 Evaluation and Results

In the following, we describe the methodology used in evaluating our affordance estimation for object replacement. Then, we show and discuss the results.

4.1 Test set: Kitchen set

The validation is performed on Kitchen set (Fig. 1), which is a test set of 15 objects. This set is composed of a selection of every-day objects (particularly kitchen objects) that can be manipulated by our humanoid robot. For learning, objects are labeled with 12 action affordances. These affordances are derived from the object-action affordance coding as introduced by Agostini et al. in [6] for a salad-making task. In this object-action coding, the object affordance is described with two elements: an action in which the object is involved (e.g, 'cut') and a preposition indicating the specific object's role in that action (e.g, 'with'). In this work however, we regard both elements as a single entity (e.g, 'cut_with') because the preposition is always attached to the action and they cannot be learned separately. The set of affordances we deal with in this paper is described in TABLE 1. Note that last the segment in the affordance name indicates the preposition—'null' indicates the absence thereof.

4.2 Datasets for training

In this paper, we investigate the use of different benchmark datasets for training our object replacement system. This allows us to determine which dataset is best suited for such a learning task. It also allows us to study the impact of having more data for training (by combining the datasets) on two aspects: i) the performance ii) the reliability of the prediction confidence. Finally, learning from a dataset of objects recorded under different configuration provides an indication of the 'learning transfer' ability of our system. In the tests, we use the following datasets:

- YCB dataset[1]: This dataset was recently made available as a benchmark for robotic grasping and manipulation [24]. The dataset contains 88 objects and provides high-resolution RGB-D scans of the object along with other physical properties. For our system, we generate RGB-D data from the scanned models.
- BigBird dataset[2]: This dataset was released as a large-scale benchmark for object instance recognition [25]. The set contains 125 objects and, similar to the YCB dataset, provides RGB-D scanned models from which we generate the RGB-D data. Generating raw RGB-D data from the scanned models is performed by capturing views from artificially-positioned sensors with spherical distribution (Fig. 2). This is implemented using the visualization tool provided by the Point Cloud Library (PCL) [26].
- SDU object dataset[3]: This dataset contains 100 objects with 30 different samples (random poses) per object captured using Kinect sensors in a real robot setup [2].

[1] http://ycb-benchmarks.opensourcerobots.org/

Affordance	Description
pour_into	pouring contents inside the object.
drop_into	dropping contents inside the object.
stir_null	stirring the contents of the object.
cut_null	cutting the object.
drop_null	dropping the object.
pick_null	picking the object.
pick_place_null	picking and placing the object.
place_null	placing the object.
pour_from	pour contents from the object.
cut_with	using the object for cutting.
stir_with	using the object for stirring.

Table 1 Description of affordances considered in this paper. The affordance is composed of two elements: action and preposition. 'null' indicates no preposition exists for this affordance. The assumed affordances of the objects in Fig. 1 are indicated in Table 2 with gray shades.

4.3 Experimental Procedure

To evaluate the performance of the system on estimating the affordances of the objects in Kitchen set, each object in the set is considered a novel object. Thus in all test cases, the object under validation is excluded from the set passed for training.
To obtain a rather balanced training set, we draw 50 random samples (of random objects) for each affordance. Due to affordance co-occurrence however, this number would be higher for some affordances.

4.4 Evaluation metrics

In the following experiments, two ways of evaluation are considered:

[2] http://rll.berkeley.edu/bigbird/
[3] http://caro.sdu.dk/sdu-dataset/

	pour_into	stir_null	drop_into	cut_null	drop_null	pick_null	pick_place_null	place_null	pour_from	cut_with	stir_with	Perfromance (object similarity)	
basket	0.3	0.3	0.3	-	-	-	-	-	-	-	-	0.3	
bowl_1	0.56	0.56	0.56	-	-	-	-	-	-	-	-	0.56	
bowl_2	1	1	1	-	-	-	-	-	-	-	-	1	
wok	0.2	0.2	0.2	-	-	-	-	-	-	-	-	0.2	
banana	-	-	-	0.16	0.16	0.16	0.16	0.16	0.07	-	-	0.09	
bottle	-	-	-	0.93	0.93	0.93	0.93	0.93	0.07	-	-	-0.86	
bowl_3	1	1	1	-	-	-	-	-	-	-	-	1	
bowl_4	0.9	0.9	0.9	0.1	0.1	0.1	0.1	0.1	-	-	-	0.8	
cleaver	-	-	-	-	-	-	-	-	-	0.13	0.13	0	
cucumber	-	-	-	1	1	1	1	1	-	-	-	1	
knife_1	-	-	-	0.05	0.05	0.05	0.05	0.05	-	0.95	0.95	0.9	
knife_2	-	-	-	-	-	-	-	-	-	0.96	0.96	0.96	
spoon_1	-	-	-	-	-	-	-	-	-	0.95	0.95	0	
spoon_2	-	-	-	-	-	-	-	-	-	0.95	0.95	0	
Perfromance (affordance similarity)	0.66	0.66	0.66	-0.35	-0.35	-0.35	-0.35	-0.35	0	-0.27	0.82	**overall mean**	0.07
												0.43	

Table 2 Affordance prediction results on Kitchen set. The values represent the average confidence (score). The average is computed from 10-20 test samples per object. The gray shadows indicate the ground-truth labeling. '-' indicates no prediction.

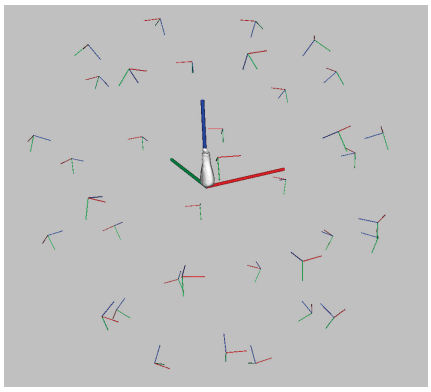


Figure 2 Generation of training samples (for YCB and BigBird datasets). The spherically-distributed coordinates shows the poses of the sensors.

4.4.1 Similarity measure

For evaluating our results, we define $S\,(\in[-1,1])$ as a similarity measure between the predicted affordances (based on the JointSVM output given in equation 6) and the expected (ground truth) ones for novel objects. This measure is composed of two metrics such that:

$$S(y(x)) = S_p(y(x)) - S_n(y(x)) \tag{7}$$

where $S_p(y(x))$ and $S_n(y(x))$ are the positive and the negative similarity metrics, respectively. The positive similarity accounts for the true positive predictions, $\mathbf{y}_p(x)$, while the negative similarity accounts for the false positive predictions, $\mathbf{y}_n(x)$. Note that both $\mathbf{y}_p(x)$ and $\mathbf{y}_n(x)$ are taken from $\mathbf{y}(x)$ by considering the expected affordances. Regarding the above, $S_p(y(x))$ and $S_n(y(x))$ are given by:

$$S_p(y(x)) = AVG(\mathbf{y}_p(x)) \cdot TPR(\mathbf{y}(x)) \tag{8}$$

$$S_n(y(x)) = AVG(\mathbf{y}_n(x)) \cdot FPR(\mathbf{y}(x)) \tag{9}$$

where AVG indicates the mean value and TPR and FPR indicate the true positive rate (recall) and the false positive rate (fall-out), respectively. Again, these rates are computed considering the expected affordances.
This similarity measure provides quantification for the system performance on the individual objects as well as on the affordances (see Subsec. 4.5). It can also be used by the robot's planner (as we discuss in Subsec. 4.6) to determine whether a novel object does constitute a viable replacement for a missing object.

4.4.2 Receiver operating characteristic (ROC) curves

ROC curves provide a graphical evaluation depicting the trade-off between the true positive rate and false positive rate, obtained by incrementally thresholding the confidence values. Thus, the ROC allow us to select a grounded threshold value for the specific scenario. In addition, the area under the curve (AUC) provides a quantitative measure for comparison. In the experiments, we derive ROC curves showing the system performance on different training scenarios.

4.5 Results and discussion

TABLE 2 shows the affordance prediction outcome on Kitchen set (Fig. 1). In this experiment, we do not use an external benchmark dataset for training but we instead permute Kitchen set itself to serve as a training set for the individual object under evaluation. This is performed by

ISBN 978-3-8007-4231-8

	pour_into	stir_nul	drop_into	cut_null	drop_null	pick_null	pick_place_null	place_null	pour_from	cut_with	stir_with	mean
Kitchen set	0.66	0.66	0.66	-0.35	-0.35	-0.35	-0.35	-0.35	0	-0.27	0.82	0.07
sdu	0.02	0.02	0.02	0.38	0.38	0.39	0.39	0.39	-0.57	0.46	0.27	0.20
ycb	0.16	0.16	0.16	0.32	0.32	0.32	0.32	0.32	-0.47	0.01	0.55	0.20
bigbird	0.4	0.4	0.4	0.3	0.58	0.62	0.62	0.62	-0.56	0.41	0.43	0.38
sdu+bigbird+ycb	-0.11	-0.11	-0.11	0.22	0.61	0.58	0.58	0.58	-0.49	0.48	0.42	0.24
mean	0.23	0.23	0.23	0.17	0.31	0.31	0.31	0.31	-0.42	0.22	0.50	0.22

Table 4 The affordance prediction evaluation on the datasets. 'Kitchen set' corresponds to the results in TABLE 2.

	Kitchen set	sdu	ycb	bigbird	sdu+bigbird+ycb	mean
basket	0.3	0.04	-0.1	-0.09	0.4	0.11
bowl_1	0.56	0.81	-0.28	1	0.27	0.472
bowl_2	1	-0.13	-0.1	-0.22	-0.02	0.106
wok	0.2	-0.03	0.01	0.02	-0.11	0.018
banana	0.09	0.8	0.74	0.8	0.81	0.648
bottle	-0.86	-0.97	-0.85	-0.97	-0.97	-0.924
bowl_3	1	0.96	1	0.9	0.9	0.952
bowl_4	0.8	0.6	0.81	0.8	0.43	0.688
cleaver	-0.56	0.01	0.12	0.08	0.2	-0.03
cucumber	1	0.7	0.52	0.77	0.86	0.77
knife_1	0.9	0.65	0.47	0.77	0.58	0.674
knife_2	0.96	0.87	0.8	0.94	0.58	0.83
spoon_1	0	0	0.56	0.6	0.69	0.37
spoon_2	0	0.09	0.1	0.03	-0.1	0.024
mean	0.39	0.31	0.27	0.39	0.32	

Table 3 The performance on Kitchen set when training is performed using various datasets. 'Kitchen set' corresponds to the results in TABLE 2.

iteratively taking out each object and train the system with the rest.

The table rows show the average output of the predicted affordances (given by equation 6) for each object. This average is computed from 10-20 test samples. Note that the gray shadows indicate the ground-truth affordances. Also note that '-' indicates no prediction. Object-wise similarity is computed, using equation 7, and averaged across all samples. Affordance-wise similarity is obtained the same way. However in this case, y(x) is replaced with the predictions of the respective affordance for the whole Kitchen set.

The table shows that for most objects, the system is able to correctly predict the affordances. False positive predictions do occur, but they tend to be associated with low confidence—except for 'bottle'. For 'spoon_1' and 'spoon_2', false positive predictions of the 'cut_with' affordance also occur with high confidence. Practically speaking however, one could use a spoon for cutting when no cutting tool is available. This suggests that, based on shape alone, the system can choose such a viable alternative despite being incorrect according to our labeling.

TABLE 3 summarizes the system performance—using the similarity measure across affordances—on Kitchen set for different training cases. Those cases include the training scenario in TABLE 2 as well as using the benchmark datasets (Subsec. 4.2), both separately and when merged, for training. Contrary to what we expect, the table shows that training the system with more data does not necessarily improve the performance. This may be due to our sampling procedure (Subsec. 4.3) that aims at getting rather balanced data and which may be, on the downside, prohibiting learning of more varieties of shape. The table also shows that training with the YCB dataset leads to the lowest performance. This may be due to the sparse variety of objects occurring in that set.

TABLE 4 shows the system performance on the individual affordances using the similarity measure across objects. The table shows that this performance significantly varies for the different training cases. For 'pour_from', low performance is obtained in all cases. This suggests that no shape attributes could be learned to associate with this affordance.

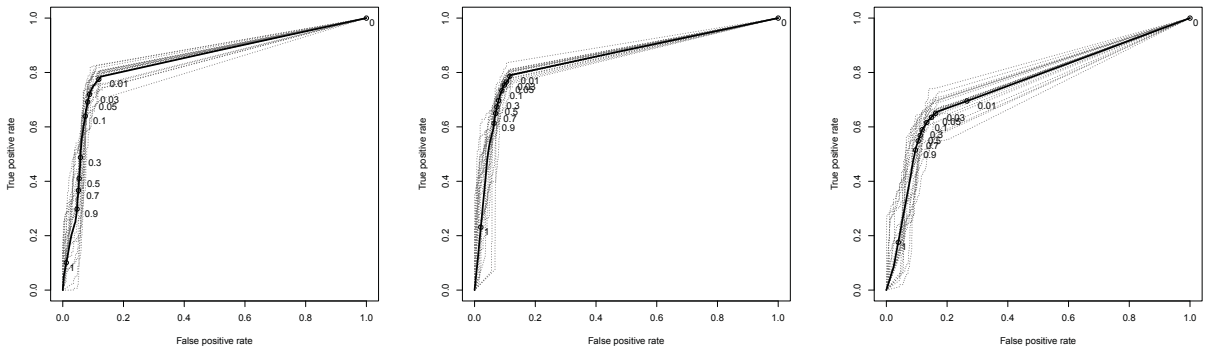
Fig. 3 shows ROC curves depicting the system performance for the different combinations of training. The curves are averaged over 20 runs (curves of the respective runs are shown in gray). In each run, we draw a random training samples (see Subsec. 4.3). All curves have points approaching the top-left corner. This suggests that we can select threshold values corresponding to a good trade off between the expected true positive rate and false positive rate. Thus, this implies that the confidence can be used as reliable way for estimating the affordances of a novel object and, consequently, for making decision on object replacement. When considering the relative performance of the system, the curves indicate similar result as in TABLE 3.

4.6 Scene interpretation on a humanoid robot

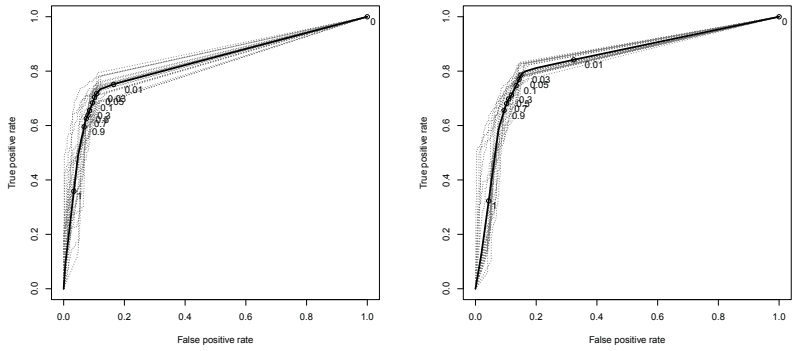
For our experiments, we used the humanoid robot platform ARMAR III [5] (see Fig. 4a), a kitchen assistant robot. The robot has two 7-DoF arms with five finger pneumatic hands. For visual perception it is equipped with a 7-DoF head and two stereo camera system, one for foveal and one for peripheral perception, and a RGB-D sensor, which we use in this scenario.

The robot is operated using the robot development environment ArmarX [27], a framework especially designed for complex multi-sensor robot systems. Besides the common control components, the framework contains memory and

ISBN 978-3-8007-4231-8



(a) Kitchen set (AUC=0.84) **(b)** SDU (AUC=0.85) **(c)** YCB (AUC=0.75)



(d) BigBird (AUC=0.83) **(e)** SDU+YCB+BigBird (AUC=0.84)

Figure 3 ROC figures for the training cases. The curves are averaged over 20 runs . The gray line depicts the curves for the individual runs. Numbers in the line shows the locations of 10 threshold values between 0 and 1.

planning systems. The object replacement presented in this paper enriches these two systems.

The visual affordance estimation (Sec. 3) requires object segmentation. To this end, the perceived point cloud needs to be segmented and clustered. To achieve this, we start with removing all the big planes using a random sample consensus (RANSAC) model approach, assuming that those planes are no objects. Then, we apply Euclidean clustering. This results in an individual point cloud for each object, which are passed on to the visual affordance estimation.

The work presented here is part of a system that is capable of creating symbolic plans for a given task. The system has the ability to search for replacements of missing objects based on several replacement-strategies. Replacement based on affordance estimation is one of these strategies. Essentially, the robot creates an assumed memory state (see Fig. 4b) from previous experience, which serves as a basis for the planner [28]. If the robot encounters that an object of a plan is missing, the replacement component is consulted for a valid replacement.

In the scenario shown in Fig. 4 for instance, the assumed memory state of the robot contains the object in Fig. 4b of which the known affordances are: 'pour_in', 'stir_null' and 'drop_into'. However, another object with unknown affordances is visible (i.e., the red bowl shown in Fig. 4a and Fig. 4c). Therefore, the approach presented in this paper is called to estimate the affordance of this object. Note that the system is trained with object samples collected from the three datasets, discussed in Subsec. 4.2, i.e. the SDU+YCB+BigBird case in the results. The outcome is shown in Fig. 4d.

In this scenario, the system predicted affordances similar to the known ones of the expected object. The associated confidence is 0.63. This corresponds to a similarity value of 0.63, according to equation 7. This similarity value could be used to decide whether this object is a valid replacement (i.e., based on a designed threshold or in comparison with predictions from other sources). The decision could also be made by thresholding the confidence values. In this case, Fig. 3e indicates that a threshold of 0.03 corresponds to a good trade-off. According to this, we obtain affordances identical to those of the expected object.

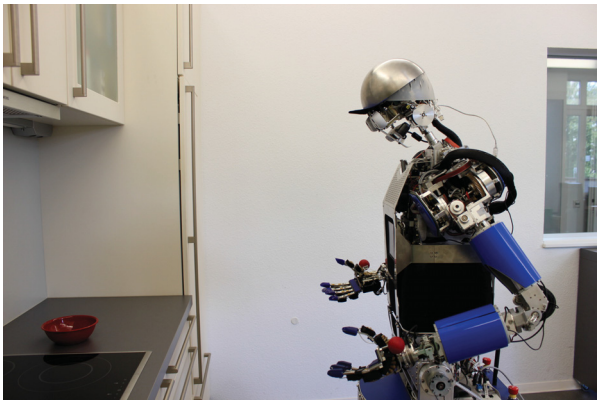
ISBN 978-3-8007-4231-8

5 Conclusion

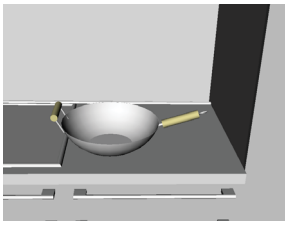
In this paper, we addressed the problem of finding replacements of missing objects by providing a mechanism that estimates the functional affordances of the unknown objects. This estimation can be used by a higher-level planning module to propose replacements. Our evaluation showed that we can correctly estimate such affordances by training the system with data obtained, e.g., from external datasets. We also demonstrated the usefulness of deriving confidence values for the affordance estimation. This is important because it enables the planner to make reliable decisions.

The approach presented and investigated in this paper serves as a basis for integrating this approach in a robot planning system compromised of several replacement strategies (based on different knowledge sources). Such a system is particularly useful for robots, especially humanoid robots, performing human-like tasks. In such cases, it is likely that some of the required objects are unavailable, making the object replacement necessary.

In this paper, we restricted our selves to learning affordances for kitchen-based manipulations executed by a particular type of robots. Also, this set of affordances, defined by a human, exhibits high co-occurrence among the affordances, which may limit the system to learn to associate affordances with more abstract visual attributes. In the future, this work can be expanded to include learning affordances not limited to such scenarios. Moreover, those affordances can be defined automatically according to a long-term memory of action outcomes, allowing the robot to learn object affordances by experience.

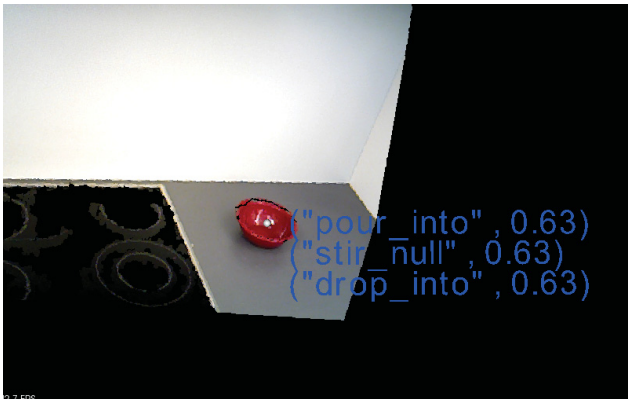


(a) Armar III encounters a small bowl while expecting the object in (b).



(b) Visualization of the robot's assumed state of the world based on it's memory content. The known affordances of the object in the state are: 'pour_in', 'stir_null' and 'drop_into'.

(c) RGB camera image from the robot's perspective.



(d) Point cloud visualization (snapshot) of the scene, showing the predicted affordances of the unexpected object.

Figure 4 Affordacne estiatmion in ARMAR III.

ACKNOWLEDGMENT

This work has been funded by the EU project Xperience (FP7-ICT-270273).

6 Literature

[1] R. P. A. P. T. A. P. Kaiser, M. Lewis and M. Steedman, "Extracting common sense knowledge from text for robot planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2014)*, 2014, pp. 3749–3756.

[2] W. Mustafa, N. Pugeault, A. Buch, and N. Krüger, "Multi-view object instance recognition in an industrial context," *Robotica*, pp. 1–22, 2015.

[3] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, 1977.

[4] H. Xiong, S. Szedmak, and J. Piater, "Scalable, accurate image annotation with joint {SVMs} and output kernels," *Neurocomputing*, vol. 169, pp. 205 – 214, 2015.

[5] 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 *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE, 2006, pp. 169–175.

[6] A. Agostini, M. J. Aein, S. Szedmak, E. E. Aksoy, J. Piater, and F. Wörgötter, "Using structural bootstrapping for object substitution in robotic executions of human-like manipulation tasks," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

[7] S. Szedmak, E. Ugor, and J. Piater, "Knowledge propagation and relation learning for predicting action effects," in *Proceedings of the IEEE Intl. Conf. on In-*

telligent Robots and Systems (IROS 2014), Chicago, 2014.

[8] I. Awaad, G. Kraetzschmar, and J. Hertzberg, "Finding ways to get the job done: An affordance-based approach," in *Twenty-Fourth International Conference on Automated Planning and Scheduling*, 2014.

[9] S. Konecnỳ, S. Stock, F. Pecora, and A. Saffiotti, "Planning domain+ execution semantics: a way towards robust execution?" in *Qualitative Representations for Robots, AAAI Spring Symposium*, 2014.

[10] L. Stark and K. Bowyer, "Function-based generic recognition for multiple object categories," *CVGIP: Image Understanding*, vol. 59, no. 1, pp. 1–21, 1994.

[11] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013.

[12] D. Parikh and K. Grauman, "Relative attributes," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 503–510.

[13] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 951–958.

[14] X. Yu and Y. Aloimonos, "Attribute-based transfer learning for object categorization with zero/one training example," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 127–140.

[15] V. Ferrari and A. Zisserman, "Learning visual attributes," in *Advances in Neural Information Processing Systems*, 2007, pp. 433–440.

[16] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[17] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *International Symposium on Experimental Robotics (ISER)*, 2012.

[18] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[19] W. Mustafa, H. Xiong, D. Kraft, S. Szedmak, J. Piater, and N. Krüger, "Multi-label object categorization using histograms of global relations," in *3D Vision (3DV), 2015 International Conference on*, 2015, pp. 309–317.

[20] D. Kraft, W. Mustafa, M. Popović, J. B. Jessen, A. G. Buch, T. R. Savarimuthu, N. Pugeault, and N. Krüger, "Using surfaces and surface relations in an early cognitive vision system," *Machine Vision and Applications*, vol. 26, no. 7, pp. 933–954, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00138-015-0705-y

[21] T. Joachims, T. Finley, and C.-N. Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[22] B. Schölkopf, R. Herbrich, and A. Smola, "A generalized representer theorem," in *Computational Learning Theory. Lecture Notes in Computer Science*. Springer, 2001, vol. 2111, pp. 416–426.

[23] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, 2nd ed. Wiley, 1995, vol. 2.

[24] B. Çalli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *CoRR*, vol. abs/1502.03143, 2015.

[25] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 509–516.

[26] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[27] M. K. K. W. N. Vahrenkamp, M. Wächter and T. Asfour, "The robot software framework armarx," *Information Technology*, vol. 57, no. 2, pp. 99–111, 2015.

[28] V. W. E. Ovchinnikova, M. Wächter and T. Asfour, "Multi-purpose natural language understanding linked to sensorimotor experience in humanoid robots," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids) (accepted)*, 2015, pp. 0–0.

ISBN 978-3-8007-4231-8