# Exercising Affordances of Objects: A Part-Based Approach

Safoura Rezapour Lakani, Antonio J. Rodríguez-Sánchez and Justus Piater[1]

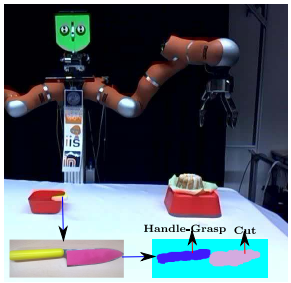
***Abstract*—This study shows how learning relations between affordances facilitates performing robotic tasks. Tasks usually involve multiple affordances. For example, for pounding a nail with a hammer, grasp-ability and pound-ability of the hammer are important for performing the pounding task successfully. Furthermore, these affordances are associated with parts of the hammer. In the pounding task, the head of the hammer affords pounding and the handle of the hammer affords grasping. We propose an RGB-D part-based approach for performing tasks. In our work, affordances are linked to object parts. We learn affordances associated with manipulation and execution of the tasks, i.e. grasping for manipulation and pounding for execution in the task of pounding a nail. Since affordances are associated with parts, tasks can be executed directly on the objects. Our approach is evaluated in six different robotic tasks on a real robot. We obtained an average of 65% task detection rate superior to the baseline methods and an average of 77% task success rate.**

***Index Terms*—RGB-D Perception; Visual Learning; Computer Vision for Other Robotic Applications**

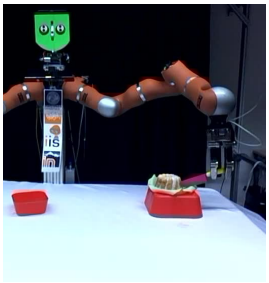
## I. Introduction

LEARNING affordances of objects and the tasks that can be performed given them are important capabilities of robots. Let us consider the robot in Figure 1. For cutting the cake in Fig. 1(b), the robot should detect an object which affords *cutting* like the knife in Fig. 1(a). It should also reason that for performing the cutting task, objects should be grasped from their handles such as the knife's handle in Fig. 1(a). Likewise to strike a ball as shown in Fig. 1(d), an object with *striking* affordance such as the hammer in Fig. 1(c) is needed. Furthermore, they should have a graspable part like a handle for performing the task. These examples show the importance of detecting affordances of objects and relations between them to perform robotics tasks.

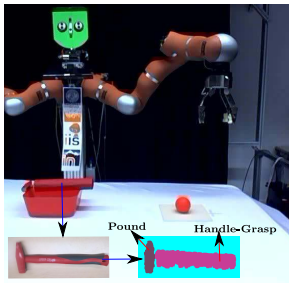
Affordances are defined as the functional properties of objects which are offered to an agent [1], [2]. These properties specify how an agent can use the objects to perform tasks. The concept of affordances has been also widely studied in robotics [3], [4]. Robots need to interact with objects in their environments. Thus, reasoning about affordances of objects is very important for them. In most cases, affordances are associated with certain parts of the objects. For example, the blade of a knife affords *cutting* or the head of a hammer affords *striking*. These affordances alone cannot be exercised in robotic tasks. For example, in order to cut the cake with the knife in Fig. 1(b), not only should the knife afford the *cutting* affordance but it also should afford the *handle-grasp* affordance.



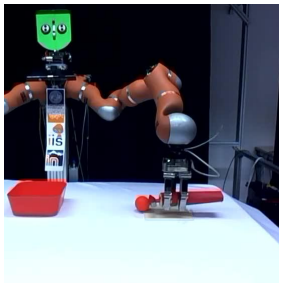

(a) For cutting the cake, an object should afford *cutting* and be graspable from its handle.



(b) The robot grasps the knife from a part which affords *handle-grasp* affordance and performs the cutting task.




(c) To strike the ball, the hammer should afford *striking* and be graspable from its handle.



(d) The robot strikes the ball by grasping the hammer from the part which affords *handle-grasp* and moving the part which affords *striking*.

Fig. 1. The robot is asked to perform two tasks: 1) cutting the cake in Fig. 1(a) and 2) striking the ball in Fig. 1(c). The robot segments objects into their functional parts and detects their affordances. It then executes the tasks based on the affordances of object parts and their relations.

Learning relations between affordances plays an important role for performing tasks. Some affordances are related to manipulation of objects to perform a task and others are related to execution of the tasks. For example, in the task of cutting the cake, the *handle-grasp* affordance is important for manipulating the knife (by grasping) and the *cutting* affordance is important for executing the task.

In this paper, we propose a novel approach for performing tasks using relations among multiple affordances. Most state-of-the-art approaches for performing robotic tasks directly associate single affordances to tasks [5], [6], [7]. The concept

Manuscript received: February, 23rd, 2018; Revised May, 22nd, 2018; Accepted June, 28th, 2018.

This paper was recommended for publication by Editor Jana Kosecka upon evaluation of the Associate Editor and Reviewers' comments.

[1]All authors are with the Intelligent and Interactive Systems group, Department of Computer Science, University of Innsbruck, Innsbruck, Austria firstname.lastname@uibk.ac.at

Digital Object Identifier (DOI): see top of this page.

of part-based affordance detection on the basis of shape from RGB-D data was introduced in [8]. In this paper, we use a different algorithm for part-based affordance classification and we demonstrate it on performing six different tasks. The hallmark of our work is that we distinguish between affordances which are related to single parts and tasks which might be linked to multiple parts and their associated affordances. The contributions of our work are twofold: 1) learning relationships between affordances by performing tasks on a real robot, and 2) associating manipulative and executive affordances for performing tasks.

In our work, affordances are detected following a part-based approach on RGB-D pointclouds (Section III-A). Given the RGB-D pointcloud of a scene, objects are segmented into parts and their affordances are detected subsequently (Section III-B). During execution of the tasks, we learn the probabilities of co-occurrences of manipulative and executive affordances (Section III-C). These probabilities are then used to infer parts for manipulation to perform the tasks.

## II. RELATED WORK

There have been several works on performing robotic tasks based on affordances of objects using visual features. These works are performed either on objects or on parts of objects.

In object-based methods, shape features such as size, convexity, or shape context are approximated from the 3D model of objects and an association between object features and tasks is learned [5], [9], [6], [7]. Most studies have been done on grasping affordance. In the work discussed in [5], [9], task-based grasping for five different tasks (handover, pouring, tool use, dish washing, and playing) is associated with shape features extracted from 3D models of objects. This association is learned with a Bayesian network and is evaluated in a simulated environment. Along these lines, the work discussed in [6] also studies task-based grasping. This work uses selective attention in addition to the visual features. The evaluation is performed on a real humanoid robot. The object-based methods can perform manipulation tasks only if the categories of objects are known. Thus, they cannot generalize to novel objects.

To overcome the generalization problem at the object level, part-based methods have been proposed. In these methods, objects are segmented initially into parts using geometric properties and affordances are associated with them [10], [11], [12]. In the work discussed in [12], objects are segmented into primitive shapes such as cubes and cylinders. The shapes are then linked to grasps for different tasks such as pouring or shaking. In the work discussed in [13], a CNN-based approach is used to segment and classify objects. Objects have also been segmented into parts based on local convexity [14]. In the work discussed in [14], handles of objects segmented in this way are used for task-based grasping. In their work, superquadrics are fitted to the parts and grasps are associated with them. This approach is then evaluated for 3D objects in a simulated environment. The work discussed in [15] uses Reeb graph [16], [17] to obtain parts. The parts are then used for task-based grasping in a simulation environment. Even though the part-based methods have a better generalization than the global approaches, they are mainly limited to grasping affordance and mostly in simulated environments. Moreover, parts used in these methods are obtained independently of affordances of objects.

We propose a part-based method for performing robotic tasks for six different affordances of objects, namely grasping (handle-grasp and wrap-grasp), pouring, scooping, cutting, striking, and placing. Object parts in our work are obtained based on affordances. Thus, useful for detecting affordance. Finally, we applied our method in real robotic scenarios.

## III. METHOD

In this section, we explain our approach for performing tasks based on object parts. The input data to our method is an RGB-D pointcloud. Since affordances are associated with shape and geometrical features, we use only depth information. The pointcloud is then segmented into parts (Section III-A) and affordances are detected on the parts of the objects (Section III-B). We then compute frequency of co-occurrences of affordances for performing tasks (Section III-C) and we use them to infer parts for manipulation.

### A. Object Part Segmentation

The first step in our method is to obtain object parts. Since we want to use parts for predicting affordances, parts should be functional and useful for the affordance detection task. Thus, we use an approach for part segmentation based on affordances of objects [18]. As shown in Figure 2, this work uses affordances to guide the segmentation of objects into functional parts. Object parts are labeled during training based on affordances such as *containing* or *pounding* (Fig. 2a) and grouped into a number of part classes.

The segmentation is a bottom-up approach starting from locally flat patches obtained from pointclouds of objects. The patches are gradually combined using a pairwise Markov Random Field (MRF) (Fig. 2b). The objective of the MRF is to find the best assignment of part classes to the patches. Let us consider $Y = \{y_1, y_2, \ldots, y_N\}$ as the patches. In the MRF, patches are assigned to random variables $X = \{x_1, x_2, \ldots, x_N\}$. Each $x_i$ takes on one of $L$ discrete values, where $l \in L$ represents a part class. The joint probability of a particular assignment of the patches to the part classes can be represented as an energy function

$$E(X,Y) = \sum_i \phi(x_i, y_i; \theta_i) + \sum_{i,j} \psi(x_i, x_j; \Theta_{ij}). \quad (1)$$

The energy function in Eqn. 1 is composed of a sum of unary potentials $\phi$ and a sum of pairwise potentials $\psi$. The unary potential $\phi$ determines the probability of a patch belonging to a part (Fig. 2c). The pairwise potential $\psi$ indicates the probability of two adjacent patches belonging to the same part (Fig. 2d). The parameters of the potential functions $\theta$ and $\Theta$ are estimated by maximizing the likelihood of the training data (i.e. minimizing the energy) over their coefficients by stochastic gradient descent. Since the parts are functional, we can use them directly to detect the affordances. More details on how parts are segmented can be found in [18].

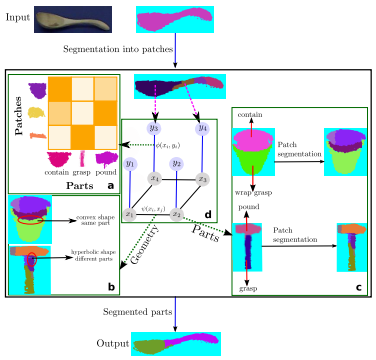



Fig. 2. Object part segmentation based on affordances [18]. Object parts have functional meaning such as pounding, grasping and containing. A graphical model for part segmentation from locally-flat object patches are learned based on two sources of information: 1) the potential of a patch $y_i$ to belong to a part $x_i$, i.e. $\phi(x_i, y_i)$, and 2) the potential of two adjacent patches to belong to the same part $\psi(x_i, x_j)$ based on their pairwise curvature value.

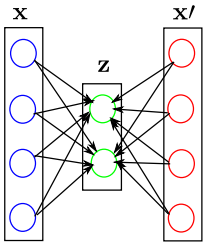



Fig. 3. A schematic diagram of an autoencoder with one hidden layer. It has an input layer $\mathbf{x}$ and an output layer $\mathbf{x}'$ and one hidden layer $\mathbf{z}$. The network attempts to reconstruct the input data. The number of neurons in the input and output layers are the same. The hidden layer compresses the data by applying an activation function.

### *B. Part-Based Affordance Detection*

After obtaining object parts, the next step is to detect their affordances. To this end, we extract features from parts and train binary affordance classifiers using them. Instead of using ad-hoc feature extraction methods, we use an unsupervised approach for learning part features. A good feature descriptor should preserve the most distinctive and frequent properties of the parts. This can be seen as a dimensionality reduction problem, and the reduced-dimensional representation of parts will be the features.

To learn the parts' features, we use autoencoders [19]. An autoencoder is a kind of unsupervised neural network that is used for dimensionality reduction and feature discovery. As shown in Figure 3, an autoencoder is a feedforward neural network with an input layer $\mathbf{x}$, an output layer $\mathbf{x}'$, and multiple hidden layers $\mathbf{z}$. Here, we use a simple autoencoder architecture with only one hidden layer. The hidden layer $\mathbf{z}$ is also considered a *code* or *latent representation*. The purpose of an autoencoder is to reconstruct the input data $\mathbf{x} = \{x_1, \ldots, x_n\}$ with a non-linear dimensionality reduction through the hidden layer. An autoencoder consists of an encoder and a decoder. The encoder maps an input vector $\mathbf{x} \in R^d$ via a nonlinear activation function $\sigma$, such as the logistic sigmoid, to a *code* or *latent representation*

$$\mathbf{z} = \sigma(W\mathbf{x} + b) \in R^p,\ p \leq d,$$

where $W$ is a weight matrix and $b$ is a bias vector. The decoder maps the code $\mathbf{z}$ to the *reconstruction* or output $\mathbf{x}'$. This mapping is done in the same way through an activation function,

$$\mathbf{x}' = \sigma(W'\mathbf{z} + b'),$$

where $W'$ is a weight matrix and $b'$ is a bias vector. Since autoencoders are a kind of neural network, the backpropagation algorithm is used to learn the weights (i.e. $W$ and $W'$) of the model. We use the codes $\mathbf{z}$ of the autoencoder as our features.

The input data to the autoencoder are parts. Since we are interested in shape properties of the parts, we use surface normals computed from their pointclouds. The surface normals are located based on the coordinate of the depth image associated with the pointcloud of the part. The input data to the autoencoder must have the same size. But object parts might have a different number of points and so different sizes. To overcome this problem, we define a local coordinate system for a part. The local coordinate system is a polar coordinate system in the plane of the depth image, centered at the center of the part's image. Each point is then located by its distance from the center and an angle with respect to the center. We then divide the part's image into a fixed number of bins. Within each bin we compute the average surface normal values of the points. Bins not containing any point are set to zero.

We use the codes associated with parts and the affordances associated with them for training the affordance classifiers. Since a part might have multiple affordances, we train binary classifiers using Support Vector Machine (SVM) with a linear kernel. The positive class for each affordance classifier consists of the parts which are labeled with the particular affordance. Likewise, the negative class contains the parts which do not have the particular affordance.

### *C. Learning Relationships Between Parts for Exercising Affordances*

Given parts and affordances associated with them, the next step is to exercise them by performing robotics tasks. To this end, we need to learn a relationship between object parts and their affordances with the tasks associated with them. For example, every sharp object part affords cutting. But to exercise them, we need to grasp them from the graspable parts (i.e. handles). Let us consider the affordance associated with the task of interest (*cutting* affordance in the cutting task) as the *executive* affordance and the affordance of the part which needs to be manipulated to perform the task (handles in the cutting example) as the *manipulative* affordance. We then collect frequency of co-occurrences between executive and manipulative affordances for performing tasks during training. Let $T$ be a 2D table storing this co-occurrence frequency where the rows are the executive affordances and the columns are the manipulative affordances. Let $A = \{a_1, \ldots, a_N\}$ be the set of all the affordances. Then, the co-occurrence frequency of each executive affordance $e \in A$ and manipulative affordance

| Affordance | Description |
|---|---|
| Grasp | Can be enclosed by a hand for manipulation. |
| Cut | Used for separating another object. |
| Scoop | A curved surface with a mouth for gathering soft material. |
| Contain | With deep cavities to hold liquid. |
| Pound | Used for striking other objects. |
| Support | Flat parts that can hold loose material. |
| Wrap-grasp | Can be held with the hand and palm. |

TABLE I
AFFORDANCE DESCRIPTIONS BASED ON [8].

| Tasks | Manipulative Affordance | Executive Affordance |
|---|---|---|
| Dropping in a box. | Grasp | Grasp |
| Cutting a cake. | Grasp | Cut |
| Scooping coffee beans. | Grasp | Scoop |
| Pouring coffee beans. | Contain | Contain |
| Striking a ball. | Grasp | Pound |
| Placing a sponge. | Support | Support |

TABLE II
TASK DESCRIPTIONS BASED ON PAIRS OF AFFORDANCES.

$m \in A$ is stored in $T(e, m)$. The probability of a manipulative affordance $m$ given the executive affordance $e$ is computed as

$$p(m|e) = \frac{T(e,m)}{\sum_{m_i} T(e,m_i)}. \quad (2)$$

The manipulative affordance is computed as

$$m^* = \underset{m}{\operatorname{argmax}}\, p(m|e). \quad (3)$$

The part associated with $m^*$ is then used for the manipulation.

## IV. EXPERIMENTAL RESULTS

In this section, we evaluate our proposed approach in two ways. First, we evaluate the perception part of our method. Then, we report on the task success rate after a successful detection.

We used the RGB-D part affordance dataset [8] for training part segmentation and affordance detection of our work. The dataset contains RGB-D images for 105 tools. We construct pointclouds from the RGB-D images. There are seven affordances associated with the surfaces of the tools: grasp, cut, scoop, contain, pound, support, and wrap-grasp. The description of the affordances is given in Table I. *Grasp* affordance here means grasping objects from handles, i.e. *handle-grasp*. Each pixel of each object is labeled with an affordance label.

We trained affordance classifiers on the RGB-D part affordance dataset [8] and applied them on our own novel objects. The objects used for learning the relationships between manipulative and executive affordances are shown in Figure 4. Each object was in 8 different poses for training. The objects used for training of tasks associated with a particular affordance are not included during testing of the same affordance. We used 12 objects in different poses for testing as shown in Figure 5. The object categories, such as pot, pitcher, container, and pasta server are novel and not provided from the RGB-D dataset [8].

Based on the definition of the affordances, we associated them with six different tasks. These tasks and their corresponding affordances are provided in Table II. Figure 6 shows the tasks performed based on the affordances. The hand pre-shape during manipulation for all the tasks is the *rim* grasp. Only for the *wrap-grasp* affordance, the spherical grasp is used.

### A. Experimental Setup

The experimental setup consists of a robot with two KUKA 7-DoF Light-Weight Robot arms with servo-electric 3-Finger Schunk SDH-2 dexterous hands. In our experiment, we only use one arm and hand. There is a Kinect sensor mounted in front of the robot for capturing the RGB-D data.

In our experiment, we obtain pointclouds from the Kinect. For efficiency, we use the Random Sample Consensus (RANSAC) algorithm provided by the Point Cloud Library [20][1] to remove the table plane. Our part segmentation method is then applied to the remaining points after table-plane removal. We used the learned affordance classifiers of the RGB-D part affordance dataset [8] for affordance detection of the segmented parts.

For each task, we perform training on an object in multiple poses. Training objects for each affordance are shown in Figure 4. We label the manipulative and executive affordance associated with the task on the segmented object. From this, we learn frequency of co-occurrences of manipulative and executive affordances.

In order to learn grasping for object manipulation, kinesthetic teaching is performed. The procedure is as follows. First, we segment the object into parts and compute their poses. To compute the pose of object parts, we perform Principle Component Analysis (PCA) on the pointcloud of the parts.The eigen-vectors of PCA form the rotation matrix. The mean of the part's pointcloud is the translation vector. We then guide the robotic arm to the manipulative object part and grasp the part using a predefined hand preshape. For example, to scoop coffee beans from the container (Fig. 6), the robot is guided to grasp the handle using a rim grasp. We record the 6D pose of the robot's end-effector (i.e., 3D position and 3D orientation) and the pose of the manipulative object part. Let us consider the pose of the robot's end-effector as $T_r$ and the pose of the part as $T_p$. Then $T_{pr}$ is the relative transformation between the manipulative part and the robot's end-effector. This relative transformation is computed for each task and is used to compute the robot's end-effector pose for grasping novel objects for similar tasks.

In the testing phase for a given task, we use the probabilities of co-occurrences of affordances to find the manipulative affordance. The manipulative part is the part associated with this affordance. We then compute the pose of the manipulative part using PCA. The end-effector pose is then computed by applying the relative transformation $T_{pr}$ to the pose of the manipulative part. After computing the end-effector pose, we perform the requested task.

[1] http://pointclouds.org/

(a) Dropping in a box. (b) Cutting a cake.

(c) Striking a ball. (d) Scooping coffee beans. (e) Placing a sponge on an object. (f) Pouring coffee beans into an object.

Fig. 4. Training objects used for learning relationships between affordances to perform tasks. Each object was in 8 different poses during training.

(a) Dropping in a box. (b) Pouring coffee beans into an object.

(c) Cutting a cake. (d) Striking a ball. (e) Scooping coffee beans. (f) Placing a sponge on an object.

Fig. 5. Test objects used in the experiments. We used 12 objects in four different poses for the evaluation. The knives and the bowl and the cup were provided only in one pose.

### *B. Task Detection Performance*

In this section, we report on the evaluation results for the perception part of our system. As mentioned earlier, affordance classifiers are trained on the RGB-D part affordance dataset [8]. We compared our affordance detection approach with the other state-of-the-art methods reported in [21] on novel object instances and categories of the RGB-D part-affordance dataset. For this experiment, we first segment objects provided from the RGB-D dataset into parts using the part segmentation approach described in Section III-A. We then apply our affordance detection method on the segmented object parts. In [8] a ranked weighted F-measure was proposed for measuring the accuracy for affordance detection. The measure takes into account that a pixel can have multiple labels, but assumes that the labels can be ranked. Table III shows our evaluation results using this metric compared to the other state-of-the-art methods. HMP [8] and SRF [8] use state-of-the-art feature extraction methods for detecting affordances of object pixels. VGG [21] and ResNet [21] use Convolutional Networks (CNN) for predicting affordances of object pixels. As can be seen in Table III, we obtain substantially higher performance than the other methods. The results show the importance of using a part-based approach for detecting affordances. Furthermore, since parts are shared among objects, we also can robustly detect affordances of novel object categories.

The trained affordance classifiers are then applied on our test objects. We compared our method with two baseline approaches on six robotic tasks,

*a) Random Selection of Manipulative Parts:* As discussed in Section III-C, we learn co-occurrence frequency between affordances to select the manipulative parts. We replaced this by randomly selecting the manipulative parts for performing the tasks.

*b) Random Selection of Parts:* We also replaced our affordance detection approach with a method which randomly selects executive and manipulative parts for performing the tasks.

*c) VFH [23] as Part Features:* As discussed in Section III-B, we use autoencoder for extracting part features. We also performed experiments by using other state-of-the-art

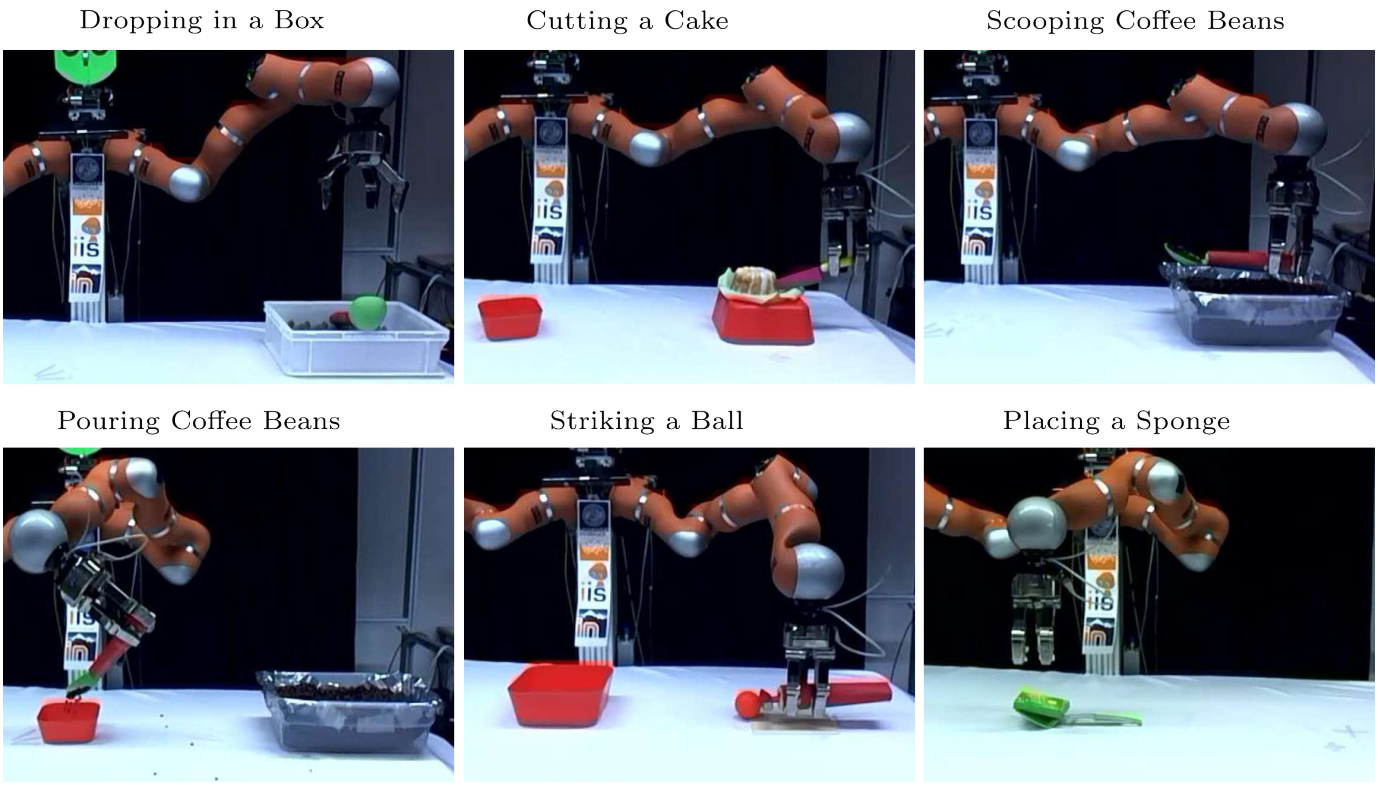



Fig. 6. Tasks performed based on affordances. Based on the definition of the affordances, each affordance is mapped to a particular task. To be used for manipulations, some thin object handles are covered.

| Method | Grasp | Cut | Scoop | Contain | Pound | Support | Wrap-grasp | Average |
|---|---|---|---|---|---|---|---|---|
| **Affordance prediction on novel instances** | | | | | | | | |
| Our Method | **0.31** | **0.30** | **0.29** | **0.39** | **0.11** | **0.27** | **0.29** | **0.28** |
| HMP [8] | 0.15 | 0.04 | 0.05 | 0.17 | 0.04 | 0.03 | 0.10 | 0.08 |
| SRF [8] | 0.13 | 0.03 | 0.10 | 0.14 | 0.03 | 0.04 | 0.09 | 0.08 |
| VGG [21] | 0.23 | 0.08 | 0.18 | 0.21 | 0.04 | 0.08 | 0.11 | 0.13 |
| ResNet [21] | 0.24 | 0.08 | 0.18 | 0.21 | 0.04 | 0.09 | 0.11 | 0.14 |
| **Affordance prediction on novel categories** | | | | | | | | |
| Our Method | **0.19** | **0.18** | **0.28** | **0.32** | **0.08** | **0.11** | **0.32** | **0.21** |
| HMP [8] | 0.16 | 0.02 | 0.15 | 0.18 | 0.02 | 0.05 | 0.10 | 0.10 |
| SRF [8] | 0.05 | 0.01 | 0.04 | 0.07 | 0.02 | 0.01 | 0.07 | 0.04 |
| VGG [21] | 0.18 | 0.05 | 0.18 | 0.20 | 0.03 | 0.07 | 0.11 | 0.12 |
| ResNet [21] | 0.16 | 0.05 | 0.18 | 0.19 | 0.02 | 0.06 | 0.11 | 0.11 |

TABLE III
AFFORDANCE PREDICTION ON NOVEL INSTANCES AND CATEGORIES OF THE RGB-D PART AFFORDANCE DATASET [8]: RANK WEIGHTED F-MEASURES [22].

features such as viewpoint feature histogram (VFH).

The evaluation results of our method compared with the other baseline approaches is given in Table IV. Table V shows evaluation results per object for different tasks. As mentioned earlier, there are four novel object categories in this experiment, namely pot, pasta server, ladle, and bowl. As it can be seen, we obtain a higher detection rate than other methods for all the objects. This shows the strength of using a part-based approach where parts are functional and distinctive. The performance for cutting task is lower than the other tasks for our method. The reason is that blades of knives contain only few points. This makes it difficult for the classification of the *cutting* affordance.

Table VI shows the running time of our task detection experiments. We reported running times of different components of our detection system. As can be seen, most time is spent on computing features. In order to compute patches and parts features, we need to compute neighborhoods for each point which is an expensive operation. Developing efficient algorithms for neighborhood estimation is not considered in this paper.

### C. Task Success Performance

We provided the task success rate of our experiment in Table VII. Each object-affordance combination was tested 10 times for each task. As it can be seen, we obtain a high

| Method | **Dropping in a Box** | **Cutting a Cake** | **Scooping Coffee Beans** | **Pouring Coffee Beans into an Object** | **Striking a Ball** | **Placing a Sponge on an Object** | **Average** |
|---|---|---|---|---|---|---|---|
| Our Method | **96** | **24** | **72** | **100** | **34** | **62** | **65** |
| Random Selection of Manipulative Parts | 44 | 14 | 38 | 46 | 13 | 35 | 32 |
| Random Selection of Parts | 25 | 5 | 14 | 18 | 5 | 20 | 14 |
| VFH [23] as Part Features | 52 | 15 | 11 | 71 | 4 | 55 | 34.7 |

TABLE IV
TASK DETECTION RATE COMPUTED ON SIX DIFFERENT TASKS: OUR METHOD IS COMPARED WITH OTHER BASELINE APPROACHES.

| Objects | Our Method | Random Selection of Manip-ulative Parts | Random Selection of Parts | VFH [23] as Part Features |
|---|---|---|---|---|
| **Dropping in a Box** | | | | |
| Ladle | 100 | 80 | 30 | 100 |
| Pasta server | 100 | 40 | 25 | 100 |
| Pot | 88 | 18 | 7 | 36 |
| Angled turner | 100 | 60 | 35 | 100 |
| Nylon turner | 86 | 51 | 37 | 86 |
| Bowl | 100 | 33 | 0 | 67 |
| Cup | 100 | 60 | 40 | 0 |
| **Pouring Coffee Beans into an Object** | | | | |
| Bowl | 100 | 30 | 20 | 100 |
| Cup | 100 | 80 | 10 | 70 |
| Pitcher | 100 | 40 | 40 | 60 |
| Pot | 100 | 33 | 0 | 56 |
| **Cutting a Cake** | | | | |
| Paring Knife | 22 | 11 | 7 | 30 |
| Ceramic Knife | 25 | 18 | 4 | 0 |
| **Scooping Coffee Beans** | | | | |
| Ladle | 77 | 43 | 19 | 9 |
| Pasta server | 66 | 33 | 9 | 14 |
| **Striking a Ball** | | | | |
| Chipping hammer | 35 | 16 | 8 | 0 |
| Ball peen hammer | 34 | 10 | 2 | 8 |
| **Placing a Sponge on an Object** | | | | |
| Angled turner | 63 | 37 | 16 | 32 |
| Nylon turner | 61 | 33 | 25 | 88 |

TABLE V
TASK DETECTION RATE COMPUTED ON SIX DIFFERENT TASKS AND 12 OBJECTS: OUR METHOD IS COMPARED WITH OTHER BASELINE APPROACHES.

success rate for most objects during testing. This shows that the affordances detected on the parts of objects can be robustly exercised in a real scenario which proves the applicability of our method. For striking a ball, we obtained a lower success rate. The reason is that objects used for this experiment (such as hammers) are heavy and need to be grasped precisely to be stable. Thus, in some cases, the robot cannot hold them during the entire experiment.

## V. CONCLUSIONS

We presented here a novel part-based approach for detecting and exercising affordances of objects on RGB-D data. We showed that a part-based representation where the parts are functional results in a high affordance detection performance. To show the generalization capabilities of our approach, we applied it on novel object categories. We obtained a good affordance prediction on these object categories (Section IV-B).

To prove the applicability of our part-based affordance detection approach, we applied it in real robotic scenarios. We learned the probability of co-occurrence of affordances for adjacent object parts in performing six different robotic tasks. Since parts are distinctive and their affordances are detected robustly, we obtained a high task success rate (Section IV-C). This proves the robustness of our approach in real scenarios.

## APPENDIX

To justify certain design and parameter choices, we here provide experimental results of our affordance detection approach on novel object instances of the RGB-D part-affordance dataset [8] under various parameter settings. We give results of using RBF and linear kernels for the affordance classifiers. As can be seen in Table VIII, a linear kernel gives us a better performance. We also changed the bin size for the patch features as well as the patch and part dictionary sizes. As shown, our method is robust to changes of these parameters. The reason is that these parameters concern object segmentation, but since we use an MRF for object segmentation, the global optimization of the MRF compensates for different values of these parameters.

## REFERENCES

[1] J. J. Gibson, "The Theory of Affordances." Hillsdale, NJ: Erlbaum, 1977, pp. 67–82.
[2] ——, *The Ecological Approach to Visual Perception*. Psychology Press, 1979.
[3] H. Min, C. Yi, R. Luo, J. Zhu, and S. Bi, "Affordance research in developmental robotics: a survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 237–255, 2016.
[4] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada, "A brief review of affordance in robotic manipulation research," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1086–1101, 2017.
[5] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 1579–1585.

| Object Segmentation | | Affordance Detection | |
|---|---|---|---|
| **Computing Patch Features** | **MRF Inference** | **Computing Part Features** | **Affordance Classification** |
| 3.97 | 0.02 | 3.27 | 0.00 |

TABLE VI
RUNNING TIMES OF TASK DETECTION. OBJECT SEGMENTATION AND AFFORDANCE DETECTION TIMES ARE PROVIDED IN SECONDS.

| Object | Task Success Rate |
|---|---|
| **Dropping in a Box** | |
| Ladle | 100 |
| Pasta server | 100 |
| Pot | 94 |
| Angled turner | 95 |
| Nylon turner | 72 |
| Bowl | 100 |
| Cup | 100 |
| **Average** | **94** |
| **Cutting a Cake** | |
| Paring Knife | 60 |
| Ceramic Knife | 70 |
| **Average** | **65** |
| **Scooping Coffee Beans** | |
| Ladle | 70 |
| Pasta server | 70 |
| **Average** | **70** |
| **Pouring Coffee Beans into an Object** | |
| Bowl | 100 |
| Cup | 100 |
| Pitcher | 90 |
| Pot | 100 |
| **Average** | **98** |
| **Striking a Ball** | |
| Chipping hammer | 75 |
| Ball peen hammer | 55 |
| **Average** | **65** |
| **Placing a sponge on an Object** | |
| Angled turner | 70 |
| Nylon turner | 70 |
| **Average** | **70** |
| **Average of All Tasks** | **77** |

TABLE VII
TASK SUCCESS RATE COMPUTED ON SIX DIFFERENT TASKS ON 12 DIFFERENT OBJECTS IN MULTIPLE POSES.

| Parameters | **Affordance Prediction** |
|---|---|
| **SVM Kernel for Affordance Detection** | |
| Linear | **0.28** |
| RBF | 0.10 |
| **Patch Dictionary Size** | |
| 10 | 0.24 |
| 30 | 0.24 |
| 50 | **0.28** |
| 70 | 0.24 |
| **Part Dictionary Size** | |
| 10 | 0.24 |
| 20 | **0.28** |
| 30 | 0.26 |
| 40 | 0.27 |
| **Patch Feature Bin Size** | |
| 5 | 0.25 |
| 10 | **0.28** |
| 20 | 0.25 |

TABLE VIII
AFFORDANCE PREDICTION USING RANK WEIGHTED F-MEASURE ON NOVEL OBJECT INSTANCES OF OUR METHOD FOR DIFFERENT VALUES OF FREE PARAMETERS: CHANGING SVM KERNEL, PATCH DICTIONARY SIZE, PART DICTIONARY SIZE, AND PATCH FEATURE BIN SIZE [22].


[6] J. Bohg, K. Welke, B. León, M. Do, D. Song, W. Wohlkinger, M. Madry, A. Aldóma, M. Przybylski, T. Asfour, *et al.*, "Task-based grasp adaptation on a humanoid robot," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 779–786, 2012.

[7] H. Dang and P. K. Allen, "Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 1311–1317.

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

[9] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Task-based robot grasp planning using probabilistic inference," *IEEE transactions on robotics*, vol. 31, no. 3, pp. 546–561, 2015.

[10] M. Tenorth, S. Profanter, F. Balint-Benczedi, and M. Beetz, "Decomposing cad models of objects of daily use and reasoning about their functional parts," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5943–5949.

[11] P. Abelha Ferreira and F. Guerin, "Learning how a tool affords by simulating 3d models from the web," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS 2017)*. IEEE Press, 2017.

[12] M. Hjelm, R. Detry, C. H. Ek, and D. Kragic, "Representations for cross-task, cross-object grasp transfer," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5699–5704.

[13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, pp. 77–85, 2017.

[14] S. El-Khoury and A. Sahbani, "A new strategy combining empirical and analytical approaches for grasping unknown 3d objects," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 497–507, 2010.

[15] J. Aleotti and S. Caselli, "Part-based robot grasp planning from human demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 4554–4560.

[16] S. Berretti, A. Del Bimbo, and P. Pala, "3d mesh decomposition using reeb graphs," *Image and Vision Computing*, vol. 27, no. 10, pp. 1540–1554, 2009.

[17] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno, "Reeb graphs for shape analysis and applications," *Theoretical Computer Science*, vol. 392, no. 1-3, pp. 5–22, 2008.

[18] S. Rezapour Lakani, A. Rodríguez-Sánchez, and J. Piater, "Can Affordances Guide Object Decomposition Into Semantically Meaningful Parts?" in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. Cambridge, MA: MIT Press, 1986, pp. 318–362.

[20] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 1–4.

[21] J. Sawatzky, A. Srikantha, and J. Gall, "Weakly supervised affordance detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5197–5206.

[22] S. Rezapour Lakani, A. Rodríguez-Sánchez, and J. Piater, "Affordances for Parts, Parts for Affordances," *Autonomous Robots*, 2018.

[23] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2155–2162.