Autonomous Robots manuscript No.
(will be inserted by the editor)

Towards affordance detection for robot manipulation using affordance for parts and parts for affordance

Safoura Rezapour Lakani · Antonio J. Rodríguez-Sánchez · Justus Piater

Received: date / Accepted: date

Abstract As robots start to interact with their environments, they need to reason about the affordances of objects in those environments. In most cases, affordances can be inferred only from parts of objects, such as the blade of a knife for *cutting* or the head of a hammer for *pounding*. We propose an RGB-D part-based affordance detection method where the parts are obtained based on the affordances as well. We show that affordance detection benefits from a part-based object representation since parts are distinctive and generalizable to novel objects. We compare our method with other state-of-the-art affordance detection methods on a benchmark dataset (Myers et al, 2015), outperforming these methods by an average of 14% on novel object instances. Furthermore, we apply our affordance detection method to a robotic grasping scenario to demonstrate that the robot is able to perform grasps after detecting the affordances.

Keywords Affordances · Part Segmentation · RGB-D perception · Supervised learning

1 Introduction

Learning functional properties of objects is an important objective in robotics. Robots need to understand and interact with their environment; therefore the functional understanding of objects plays an important role for them. For example, the robot in Figure 1 is asked to grasp and remove the pot from the table. To this end, the robot should detect the graspability of the pot (*handle-grasp* or *wrap-grasp*). Likewise, if the robot is asked to fill the pot, it should detect the *containing* functionality of the object. This small example reflects the importance of detecting the functional properties of objects in robotics manipulation scenarios.

The functional properties an object offers an actor (Gibson, 1979, 1977; Norman, 1988), also known as its *affordances*, determine the way the objects can be used. For example, a shovel affords supporting and grasping or a mug affords containing. In robotics, the concept of affordances has been widely investigated. Especially in indoor or kitchens scenarios, reasoning about affordances of objects is important for robots. In particular, we are following a tradition of research in robotics that defines affordances as functional properties of objects (Myers et al, 2015). Following this definition, affordances are present in the objects by design, especially in kitchen objects or tools. For example, a spoon is designed for scooping or a mug is designed for containing. Learning affordances is important for performing robotic tasks. Tasks usually require multiple affordances. Let us consider a task of scooping beans with a kitchen utensil. For this task, a utensil can be used which affords scoopability and graspability. For example, a ladle or a scoop can be used to perform the task but a rolling pin or a whisk can not be used. Thus, learning affordances is the first step for detecting objects which can be used to perform robotic tasks.

As can be seen in Fig. 1, affordances are not necessarily related to entire objects but mostly to their parts. For example, the inside of a pot affords the *containing* functionality, the outside the *wrap-grasping*, and the handle the *handle-grasping* functionalities. In fact, not only the inside of a pot but also most parts with a deep concavity afford the containing functionality. They can have different shapes and exist in different objects such as pots or bowls, but they afford the *containing* functionality. A part-based affordance detection

Safoura Rezapour Lakani
Universität Innsbruck
Tel.: +43 512 507 53268
Fax: +43 (0) 512 / 507 - 53069
E-mail: safoura.rezapour-lakani@uibk.ac.at

Antonio J. Rodríguez-Sánchez
Universität Innsbruck

Justus Piater
Universität Innsbruck

This is a post-peer-review, pre-copyedit version of an article published in the Springer journal Autonomous Robots. *The final authenticated version is available online at* http://dx.doi.org/10.1007/s10514-018-9787-5.

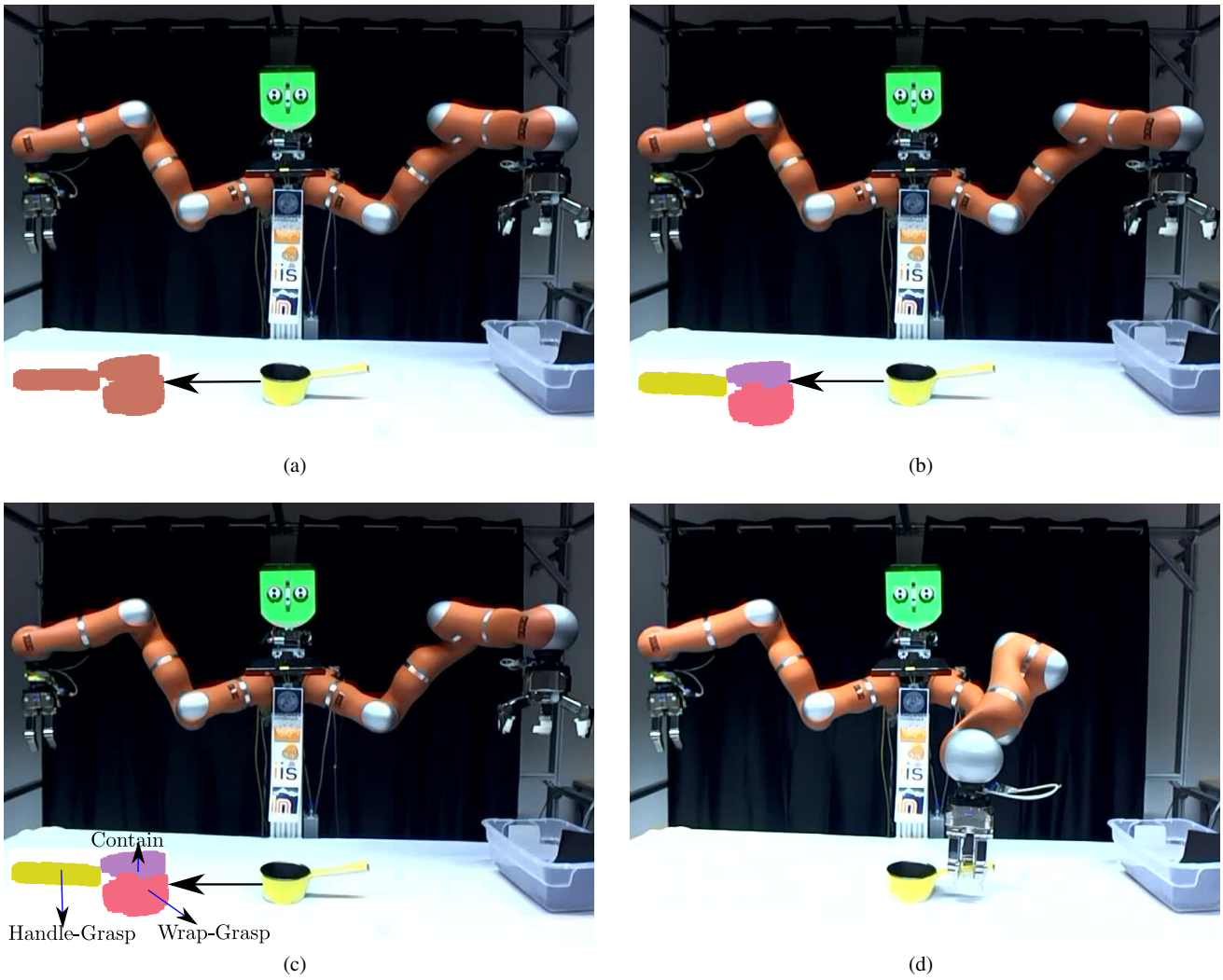

Fig. 1 The robot is asked to grasp the object on the table using the handle-grasp affordance. It gets an input from the kinect on its chest (Fig. 1(a)), segments the object into parts (Fig. 1(b)) and detects its affordances (Fig. 1(c)). It then uses the handle-grasp affordance for grasping the object (Fig. 1(d)). The pointcloud and part segmentation of the object are shown based on the view of the robot's kinect.

method could then generalize better when faced with novel objects. Therefore, decomposing objects into parts can benefit the performance of affordance detection.

Part-Based object representation has been widely studied in computer vision (Felzenszwalb et al, 2010; Wang and Yuille, 2015; Fidler and Leonardis, 2007). Although such methods have shown a good performance in object recognition scenarios, the parts they produce are not necessarily applicable to the affordance detection task. In these methods, objects are segmented into parts based on visual features such that they can discriminate object categories. Therefore, the segmented parts might not be useful for detecting the affordance of objects. In order to overcome this problem, we can use affordances directly for part segmentation. The parts obtained in this way can then be used for detecting the affordances of objects.

In this paper, we address the problem of linking affordances with the visual part-based representation of objects. Using this link for object part segmentation provides us with object parts that can be directly used for predicting the affordances of objects.

Visual representation of parts using surface normals and curvatures can provide us with distinctive information for affordance detection. For example, most convex surfaces (the inside of a pot or a cup) are associated with containing affordance or most concave surfaces (the outside of a pitcher or a pot) afford wrap-grasp affordance. Therefore, we encode surface shape information of parts in an unsupervised manner for detecting affordances (Section 4). Assuming that affordances discussed in this work are independent from each other, we then train an affordance detector for each affordance.

The contributions of this paper are twofold: 1) a part-based segmentation method guided based on affordances of objects and 2) a part-based affordance detection method. We propose a bottom-up segmentation approach using a Markov Random Field (MRF) for object part segmentation from RGB-D pointclouds (Figure 1(b)). Since object parts can have multiple affordances, we use a generative model rather than a discriminative one for object part segmentation. The affordance detection is then performed on the segmented parts as shown in Figure 1(c). In order to show the applicability of our approach, we apply it in a robotic grasping experiment. In the experiment, the robot is asked to grasp those parts that exhibit a particular affordance such as the *handle-grasp* in Figure 1(d).

2 Related Work

Detecting affordances based on visual features has been studied for decades in robotics. Affordance detection has been performed either at the *global* object level or at the level of *local* object segments.

At the object level, affordances have been associated with global object features (Katz et al, 2014; Koppula and Saxena, 2014). Affordances can also be linked to the 3D geometry of the objects and their pose (Aldoma et al, 2012). In this way, object recognition and 6DOF pose estimation are essential for affordance detection. Since affordances provide action possibilities for a robot, the relation between an object and the robot's end-effector for performing an action can be directly linked to the object model (Hart et al, 2015, 2014). In this way, not only affordances but also manipulation trajectories can be inferred after successfully recognizing objects. Object affordances can also be associated with visual attributes of objects (Hermans et al, 2011). In these works, a set of visual attributes is used for affordance prediction. Affordances can also be associated with the structure of objects (Stark and Bowyer, 1991). Through this association, object categories can be defined with their functional properties shared by all the objects in the category. Affordances can also be associated with the functional regions of objects (Stark et al, 2008; Omrčen et al, 2009). In this way, the visual or spatial relationships between object categories and functional regions are learned. These learned relationships are then used for detecting affordances of objects. For predicting the affordances of objects, first objects are recognized. Then the regions are localized in the objects. Finally, affordances are detected on the regions of objects. Object-Based affordance detection methods perform well when the object categories are known. The main drawback of these methods is not being able to generalize to novel objects.

In order to overcome the deficiencies of the global-level affordance detection methods, local methods have been proposed. At the local level, affordances have been associated with fixed-size object segments (Myers et al, 2015; Nguyen et al, 2016; Yao et al, 2013). Often, state-of-the-art feature extraction methods are used for feature extraction at the patch level (Bo et al, 2013) and combined with a discriminative classifier for affordance detection. Richtsfeld et al (2014) use a hierarchical, bottom-up approach for segmenting RGB-D data into objects. In this approach, the pixels of RGB-D images are initially clustered based on depth and neighborhood information. Then, parametric surfaces and boundaries are fitted to the extracted clusters. These surfaces are subsequently assembled to compose objects. Recently, Convolutional Neural Network (CNN)-based approaches have been used for this purpose as well (Nguyen et al, 2016; Sawatzky et al, 2017).

Affordances can also be assigned to geometrical shapes and surfaces in objects (Desai and Ramanan, 2013; Laga et al, 2013; Fu et al, 2008; Rivlin et al, 1995; Varadarajan and Vincze, 2011). Most of these approaches rely on provided 3D models which are then segmented into regions. The regions are obtained either by extracting geometrical properties such as surface shape from objects (Desai and Ramanan, 2013), or by fitting geometrical shapes (Laga et al, 2013; Fu et al, 2008), or superquadrics (Rivlin et al, 1995; Varadarajan and Vincze, 2011). Affordances are then detected at the segmented regions. Despite the fact that these methods showed better generalization than the global methods, the fixed-size segments used in these works are not necessarily distinctive enough for the affordance detection task. Local representations should have two characteristics to be useful for affordance detection: They should be distinctive, and they should be frequent among novel instances or categories of objects. Segments are frequent among various objects but not distinctive enough. Segmentations using geometrical shapes are distinctive, but they are limited mostly to simulated environments.

We propose here a local representation which is distinctive and frequent in real scenarios for predicting affordances of objects. In this paper, we address these problems by using the relationship between object parts and affordances. We argue that this relationship can boost object decomposition and consequently affordance detection. The geometrical constraints can be obtained directly based on affordance cues rather than predefined constraints. The parts derived in this way are also useful for the affordance detection. Moreover, affordance detection in this manner is more robust and generalizable when faced with novel objects.

3 Affordances for Parts

In this section, we explain our object part segmentation approach (Rezapour Lakani et al, 2017) which is shown in Figure 3. The input data to our system is an RGB-D pointcloud (the top part of Fig. 3) and the output data, segmented object

parts (the bottom part of Fig. 3). As our model uses parts for affordance detection, we will focus on shape and geometrical features neglecting color information (i.e. we will make use of depth only). We want to have a segmentation approach which generalizes to novel objects hence we use a compositional representation. The input data is initially segmented into locally flat surfaces, henceforth *patches*. The patches are at the lowest level of our compositional model (Fig. 3). They are merged subsequently together and form object parts. This merging is guided based on the affordances of the parts. As it is denoted in Fig. 3e, the training data also has manually labeled affordances. A training part is a connected set of patches that share the same set of affordances such as *scooping* and *containing* affordances for the spoon or *pounding* and *wrap-grasping* affordances for the head of the hammer. We then formulate the segmentation problem with a Markov Random Field (MRF) to learn/infer object parts from the patches (the middle part of Fig. 3). We will explain this training procedure in more detail.

3.1 Training a Patch Model

Patches are the lowest component of our part-based compositional model. As mentioned above, they are locally flat surfaces obtained from the pointcloud data and gradually form object parts. In order to be used for the segmentation, they should be frequent and distinctive among novel object parts. Therefore we extract surface normal features from the patches and create a codebook from them.

We used the Region Growing Segmentation algorithm (Rabbani et al, 2006) (available in the Point Cloud Library (Rusu and Cousins, 2011)[1]) for obtaining the patches. This algorithm segments the pointclouds into surfaces based on the angles between normals of adjacent points. Some examples of applying this algorithm to the pointcloud data are shown in Figure 2. Since a patch is a locally flat surface, within a patch surface normals are all similar thus not distinctive for the segmentation purpose. Therefore to represent a patch, we also consider its adjacent patches. We compute surface normals of all the points belonging to a patch and their adjacent points belonging to the neighboring patches. We then quantize these surface normal values in each dimension into a histogram and concatenate them together to represent a patch.

Given the training patches $Y = \{y_1, \ldots, y_n\}$ and their histogram of surface normals, we construct a dictionary (Leung and Malik, 2001; Fei-Fei and Perona, 2005; Lazebnik et al, 2006) from them. We use the K-Means algorithm and cluster the patches based on their features into K clusters. From this, we construct a codebook $C = \{c_1, \ldots, c_K\}$,

[1] http://pointclouds.org/

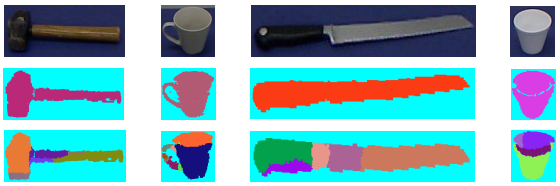

Fig. 2 Patch segmentation of pointclouds. Top row: RGB images; middle row: pointclouds; bottom row: segmented patches. Colors indicate distinct patches.

where the mean cluster values are the codewords. The codewords serve as the *patch types* in our model and we use them in our segmentation algorithm (c.f Section 3.3).

3.2 Training Part Classes

The goal of our segmentation method is to decompose a scene into parts. To this end, we should determine different parts present in a scene. Similar to the patches (Section 3.1), parts should be frequent in different scenes and distinctive. Thus, we follow the same procedure as discussed in Section 3.1 for the training parts. That is, we represent the parts as a histogram of their constituent patch types and make a dictionary from them. This dictionary will then be our *part classes* present in scenes.

Let us consider a part z consisting of n patches $\{y_1, \ldots, y_n\}$. We find the patch types $\{c_{y_1}, \ldots, c_{y_n}\}$ (obtained as explained in Section 3.1) with the minimum Euclidean distances to the patches. We then represent a part by a histogram of its constituent patch types $\{c_{y_1}, \ldots, c_{y_n}\}$. We limit the number of parts in scenes to L and cluster them using the K-Means algorithm to L clusters. From this, we construct a codebook $R = \{r_1, \ldots, r_L\}$, where the mean cluster values are the codewords.

3.3 Training a Markov Random Field (MRF) for Object Part Segmentation

In order to perform our bottom-up object part segmentation, we employ a pairwise MRF (Figure 3c). Let us consider $Y = \{y_1, y_2, \ldots, y_N\}$ as the patches in our model. We want to represent them by random variables $X = \{x_1, x_2, \ldots, x_N\}$ (Fig. 3c). Each x_i takes on one of L discrete values, where $l \in L$ represents a part class. The value of x_i determines probabilistically the label of the patch y_i.

The joint probability of a particular assignment of part classes to patches can be represented as an energy function

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

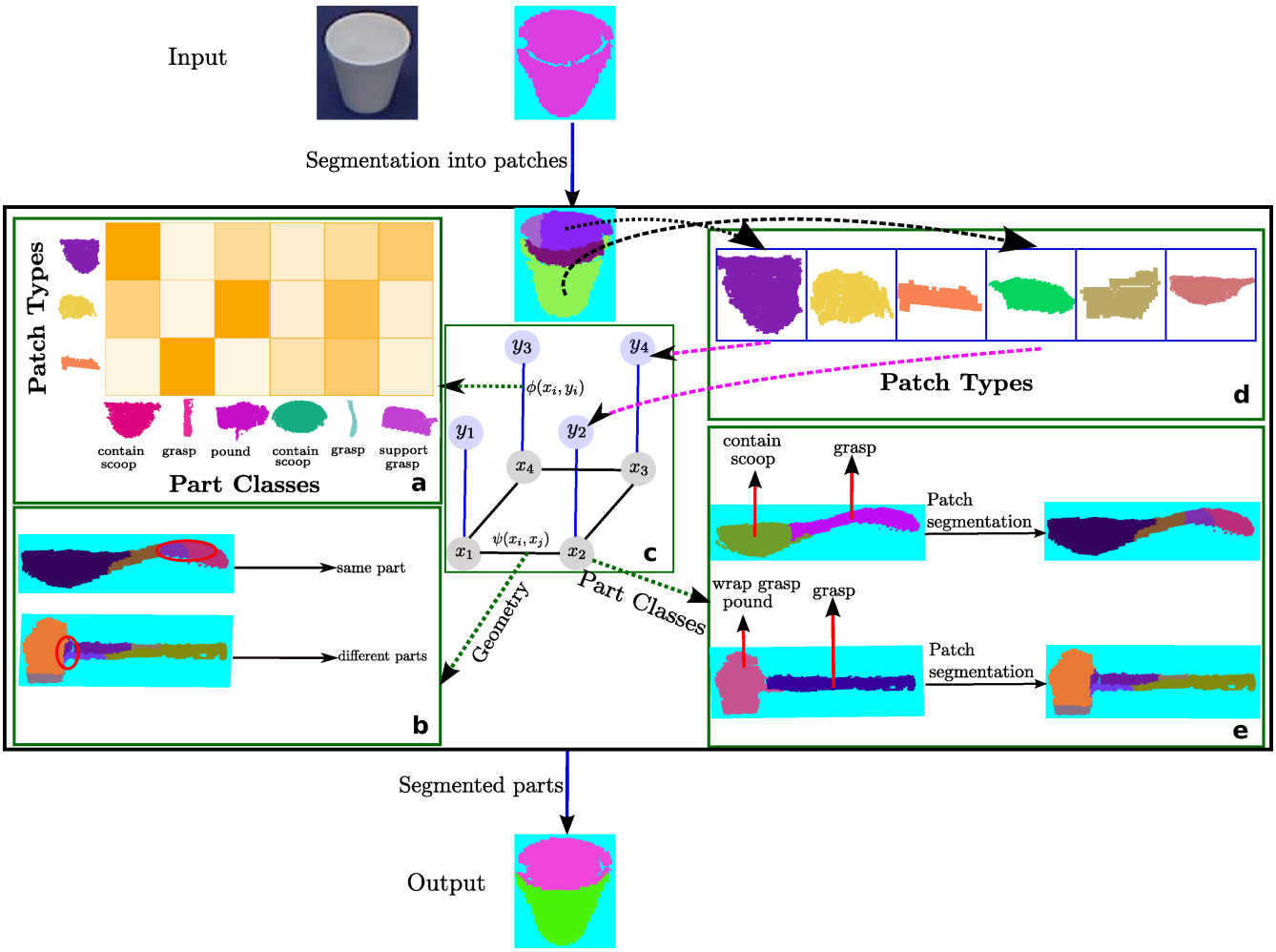

Fig. 3 Object part segmentation based on affordances. Object parts in our model are driven based on their affordances such as pounding, grasping and containing. We learn a graphical model for part segmentation from locally-flat object patches based on two sources of information: 1) the potential of a patch type belong to a part class, 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.

The energy is composed of two terms, a sum of unary potentials ϕ and a sum of pairwise potentials ψ. The unary potential ϕ determines the likelihood that a patch type belongs to a part class (Fig. 3a). In Section 3.3.1, we explain how this potential is computed. As shown in Fig. 3b, the pairwise potential defines the joint probability of pairs of adjacent labels x_i and x_j (Section 3.3.2). The vector θ and the matrix Θ are the parameters of the potential functions. We estimate them by maximizing the likelihood of the training data (i.e. minimizing the energy) over their coefficients by stochastic gradient descent.

3.3.1 Learning the Unary Potentials

The unary potential indicates the conditional likelihood for patch types given part classes. This is computed based on the co-occurrence frequency between the part classes $R = \{r_1, \ldots, r_L\}$ and patch types $C = \{c_1, \ldots, c_M\}$.

Let T be a 2D table storing this co-occurrence frequency where the rows are the patch types and the columns are the part classes. The co-occurrence frequency of each part class r and patch type c is contained in $T(r, c)$. In order to compute this frequency, we use the training parts $Z = \{z_1, \ldots, z_m\}$. Let us consider again the training part z consisting of n patches $\{y_1, \ldots, y_n\}$. The patches are assigned to the patch types $\{c_{y_1}, \ldots, c_{y_n}\}$. In the same way, we assign the part z to the part class r which has the minimum Euclidean distance among the other part classes $R = \{r_1, \ldots, r_L\}$ to z. The probability $p(c|r), r \in R, c \in C$ of a patch type c given the part class r is computed as

$$p(c|r) = \frac{T(r, c)}{\sum_{c_i} T(r, c_i)}. \quad (2)$$

We use this probability to compute the unary potential

$$\phi(x_i, y_i; \theta_i) = \exp(-\theta_i p(c_{y_i}|x_i)) \quad (3)$$

for a particular assignment of x_i in our MRF model. As can be seen, the energy is minimized as the probability gets higher.

3.3.2 Learning the Pairwise Potentials

The pairwise potential in our MRF model is computed based on pairs of neighboring patches. We can see in Fig. 3b that, for the patches belonging to the same part (e.g. the handle part of the spoon in Fig. 3b), the surface of the part changes smoothly. In contrast, this change is substantial for adjacent patches belonging to different parts (e.g. the head and the handle of the hammer in Fig. 3b). Therefore we use surface curvature between adjacent patches for the pairwise relationship.

Let us consider a training object consisting of p patches $\{y_1, \ldots, y_p\}$ and m parts $\{z_1, \ldots, z_m\}$. For each pair of adjacent patches y_i, y_j, we compute the surface curvature γ_{ij} between these patches using fixed-size neighborhoods containing points from both patches. We train a binary Support Vector Machine (SVM) with a Radial Basis kernel (RBF) to predict from a curvature value whether the two patches belong to the same object part or not. We obtain a probabilistic prediction $q(\gamma_{ij})$ of patches y_i, y_j belonging to the same part by transforming the SVM classification score $s(\gamma_{ij})$ by a sigmoid function,

$$q(\gamma_{ij}) = \frac{1}{1 + \exp\left(As(\gamma_{ij}) + B\right)}, \tag{4}$$

where the parameters A and B are learned from the SVM scores of the training data using a two-parameter minimization algorithm (Platt et al, 1999).

We use the trained SVM curvature classifier to compute the pairwise energy term

$$\psi(x_i, x_j; \Theta_{ij}) = \tag{5}$$
$$\begin{cases} 0 & x_i = x_j \\ t & x_i \neq x_j, s(\gamma_{ij}) < 0 \\ \exp\left(-\Theta_{ij} q(\gamma_{ij})\right) & \text{otherwise.} \end{cases}$$

If the patches share the same label $x_i = x_j$, the energy is at its minimum. Otherwise, the classifier is used to predict, based on the curvature γ_{ij} between the patches, whether they belong to the same part. A negative score $s(\gamma_{ij}) < 0$ indicates that they do not. In this case, the energy is set to a maximum value of t, essentially forcing the patches to be assigned to different parts. A nonnegative score $s(\gamma_{ij}) \geq 0$ is an indication that they might belong to the same part. In this case, the pairwise potential is given by the probability $q(\gamma_{ij})$ determined by the classifier.

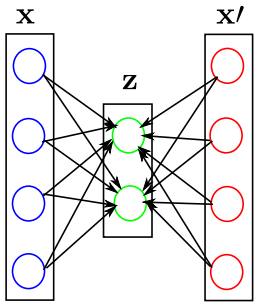

Fig. 4 A schematic diagram of an autoencoder with one hidden layer. It has an input layer $\mathbf{x}$ and an output layer $\mathbf{x}'$ and one hidden layer $\mathbf{z}$. The network attempts to reconstruct the input data. The number of neurons in the input and output layers are the same. The hidden layer compresses the data by applying an activation function.

4 Parts for Affordances

Given the segmentation of objects into parts, the next step is to detect their affordances. To this end, we extract features from the training parts and train affordance classifiers.

In Section 3.2, we explained that the parts are represented as a histogram of their constituent patch types. This representation is sufficient for part segmentation because we need to obtain a relationship between patches and parts. However, this is not enough to detect the affordances of objects. We need a stronger representation that captures the global shape of the parts. Instead of using ad-hoc feature extraction methods, we use an unsupervised approach. 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. In the following, we first explain the approach for the unsupervised feature learning. We then mention how this approach can be used for the parts. Finally, we explain training the affordance classifiers.

Unsupervised Feature Learning In our work, we use autoencoders for feature learning. An autoencoder is a kind of unsupervised neural network that is used for dimensionality reduction and feature discovery (Rumelhart et al, 1985). As shown in Figure 4, it 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*. We use the codes $\mathbf{z}$ of the autoencoder as our features.

Part Representation We use the autoencoder for representing object 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 centroid c of the part's image. Each point is then located by its distance from c and its angle with respect to c. 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 this local representation of surface normals of the parts as the input to the autoencoder. We train the network with the training parts. The trained network is then used to compute the features of the parts. The codes of the network are considered as the features.

Training an Affordance Model The ultimate goal of our work is to detect the affordances of objects. To this end, we train an affordance model on the given data. As mentioned in Section 3, the training data are the pointclouds of objects. Since we use a part-based approach, the training pointclouds are segmented into their parts. The training parts also have affordance labels. We use the parts and the affordances associated with them for training the affordance models. Since a part might have multiple affordances, we train binary classifiers as opposed to a multi-class classifier. We use SVM with a linear kernel. The training data for the SVM are the features computed from the parts. 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. We use the trained affordance classifiers for detecting the affordances of novel objects.

5 Experimental Results

In this section, we report on the experimental evaluation of our part-based affordance detection method. We compared our method with a number of baseline approaches on a benchmark dataset for affordance detection (Myers et al, 2015),

State-of-the-art Methods We compared our method with two other state-of-the-art affordance detection methods (Myers et al, 2015; Sawatzky et al, 2017). Sawatzky et al (2017) used different CNN architectures with RGB-D features for affordance detection. Myers et al (2015) initially decompose objects into supervoxels. The supervoxels are obtained only from visual data without affordances. They evaluated different features based on RGB, depth, surface shape, and curvatures computed from the supervoxels. We compared with the best results reported by Myers et al (2015). We compared our results with the evaluation results of both methods as reported by Sawatzky et al (2017) since Myers et al (2015) do not give sufficient detail of the evaluation procedure, e.g. how the dataset is split into training and test sets.

Our Method with an RBF Kernel As discussed in Section 4, we use a linear kernel for the affordance classifiers. We also provide experimental results using an RBF kernel for these classifiers.

Our Method with a Linear Kernel for Curvatures We also performed experiments using a linear kernel (instead of an RBF kernel) in the curvature classifier used to compute the pairwise term of the MRF.

Our Method with Histogram Part Features In order to prove the importance of using an unsupervised feature learning method for affordance detection, we also performed experiments when part features are histograms of patch types. We trained affordance classifiers using the SVM with linear and RBF kernels.

Our Method with LCCP Parts (Stein et al, 2014) In order to show the importance of our affordance-driven part segmentation approach, we replaced it with another part segmentation method. We used the Locally Convex Connected Patches (*LCCP*) method which uses only visual information without affordances. LCCP segments objects using local convexity of adjacent supervoxels into parts. We used the recommended parameters of this method for our experiments. The training data are then segmented using LCCP into parts. We followed the same procedure to create the part dictionary using LCCP-segmented parts. This dictionary is then used in the unary potential of our MRF. The segmented LCCP parts are also used for affordance detection. The parts are labeled with ground-truth affordance classes taken from the RGB-D part-affordance dataset (Myers et al, 2015). Parts having inconsistent affordance labels (i.e. oversegmented parts) are not used for training.

Furthermore, we used our method in real robotic scenarios. We performed a grasping experiment based on the detected affordances of object parts. In the following, we explain the procedures for these experiments in more detail.

5.1 Affordance Detection of Tool Parts

We evaluated our part-based affordance detection method on the RGB-D part affordance dataset by Myers et al (2015). The dataset contains RGB-D images for 105 tools. Since our approach works on pointclouds, 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 1. Each pixel of the objects is labeled with an affordance. Since a part can have multiple affordances, there is also a rank of affordance labels for each object pixel. The dataset is split in two ways: novel instances and novel categories. We evaluated our method by two-fold cross validation on both splits of the dataset.

For training, we used the labeled data from the RGB-D part affordance dataset. For fair comparison with Myers et al (2015), we used the first-rank affordances for training(i.e. among the overlapping affordances, we used the first-rank affordances for training.). A part is the continuation of adjacent pixels with the same affordance labels. The parts are subsequently segmented into patches as described in Section 3.1. The threshold for the Region Growing Segmentation algorithm was set to three degrees, the default suggested by its authors. Varying this threshold, we obtain patches of different sizes. In order to find the right threshold value, we computed over-segmentation errors on a sample set of training data, and chose the threshold with the least over-segmentation error. In case of ties, we chose the threshold resulting in the smallest number of patches, reducing inference times in our MRF model. We experimented with different parameters for the bin size of patch features, patch dictionary size, and part dictionary size on novel object instances of the RGB-D part affordance dataset (Myers et al, 2015). The parameters with the best affordance detection performance are then used. For inference and sampling of our MRF model, we used the Undirected Graphical Model package (UGM) by Schmidt (2007). The learned MRF model is used for segmenting parts in novel objects. Finally, the affordances of the segmented parts are detected by the learned affordance classifiers (Section 4).

For the evaluation, we initially remove dominant plane from the pointclouds using the Random Sample Consensus (RANSAC) algorithm provided by the Point Cloud Library (Rusu and Cousins, 2011)[2]) to remove the ground plane. We then apply our part segmentation and affordance detection approaches on the remaining points.

Evaluation Metric The comparison metric used by Myers et al (2015) is the rank weighted F-score R_1^w, an extension of the F-measure

$$F_1 = \frac{2 \cdot tp}{2 \cdot tp + fn + fp}, \tag{6}$$

where tp is the number of true positives, fn is the number of false negatives, and fp is the number of false positives. The weighted F-score F_1^w is computed for evaluating the probabilistic output of a classifier with respect to binary ground truth (Margolin et al, 2014). This metric computes the partial correctness or incorrectness of the output values. Let G denote a binary ground-truth vector and D the corresponding vector of posterior probabilities computed by a classifier. The weighted F-score is computed as

$$F_1^w = \frac{2 \cdot tp'}{2 \cdot tp' + fn' + fp'} \tag{7}$$

$$tp' = D^{\mathrm{T}} G \tag{8}$$

$$fn' = (1 - D)^{\mathrm{T}} G \tag{9}$$

$$fp' = D^{\mathrm{T}} (1 - G), \tag{10}$$

where tp', fn', and fp' stand for weighted true positives, weighted false negatives, and weighted false positives, respectively.

The rank weighted F-score R_1^w introduced by Myers et al (2015) takes into account multiple, ranked affordances. It is computed based on weighted F-scores $F_1^w(r)$ for affordances of different ranks r. We compute weighted F-scores $F_1^w(r)$ for affordance labels of all ranks $r = 1, \ldots, 7$. The rank weighted F-score is then given by

$$R_1^w = \sum_{r=1}^{7} w_r F_1^w(r), \tag{11}$$

where

$$w_r = \frac{1}{\sum_{r'=1}^{7} w_{r'}} 2^{7-r}. \tag{12}$$

This metric weights top-ranked affordances most heavily, and is intended to capture how well the detector generalizes across multiple affordances. We use this metric in our experiments for direct comparison with other recent methods evaluated on the RGB-D part affordance dataset.

Affordance Detection of Novel Object Instances We performed a two-fold cross validation on the novel object instances as provided by Myers et al (2015)[3]. Table 2 shows the affordance detection performance in terms of R_1^w on the novel-instance split of the RGB-D part affordance dataset (Myers et al, 2015). We outperform the other state-of-the-art methods for all the affordances. This shows the robustness of a part-based method. Figure 5 shows some sample results of our experiment. The training and test data used for the objects shown in Fig. 5 are shown in Figure 6. As the figure shows, the affordances are detected properly for the given objects. The main reason is that the object part segmentation is driven by the affordances. Hence they are useful for detecting the affordances themselves.

We also computed F_1^w on first-rank affordances of novel object instances for our method. The results of this evaluation is given in Table 3. We can see that F_1^w for the *contain*

[2] http://pointclouds.org/

[3] Please see Section 6 for a complete list of object instances and their corresponding splits.

Affordance	Description
Grasp	Can be enclosed by a hand for manipulation (handle).
Cut	Used for separating another object (the blade of a knife).
Scoop	A curved surface with a mouth for gathering soft material (trowel).
Contain	With deep cavities to hold liquid (the inside of a bowl).
Pound	Used for striking other objects (the head of a hammer).
Support	Flat parts that can hold loose material (turner/spatula).
Wrap-grasp	Can be held with the hand and palm (the outside of a cup).

Table 1 Affordance descriptions based on Myers et al (2015).

Method	Grasp	Cut	Scoop	Contain	Pound	Support	Wrap-grasp	Average
Our Method	0.31	**0.30**	**0.29**	**0.39**	0.11	**0.27**	**0.29**	**0.28**
Our Method with an RBF Kernel	0.23	0.06	0.09	0.16	0.04	0.06	0.08	0.10
Our Method with a Linear Kernel for Curvatures	**0.32**	0.28	0.23	0.33	**0.13**	0.23	0.21	0.25
Our Method with Histogram Part Features, Linear Kernel	0.26	0.11	0.10	0.22	0.05	0.07	0.21	0.15
Our Method with Histogram Part Features, RBF Kernel	0.25	0.16	0.12	0.33	0.04	0.12	0.23	0.18
Our Method with LCCP Parts (Stein et al, 2014)	0.29	0.01	0.02	0.28	0.00	0.01	0.02	0.09
HMP (Myers et al, 2015)	0.15	0.04	0.05	0.17	0.04	0.03	0.10	0.08
SRF (Myers et al, 2015)	0.13	0.03	0.10	0.14	0.03	0.04	0.09	0.08
VGG (Sawatzky et al, 2017)	0.23	0.08	0.18	0.21	0.04	0.08	0.11	0.13
ResNet (Sawatzky et al, 2017)	0.24	0.08	0.18	0.21	0.04	0.09	0.11	0.14

Table 2 Affordance prediction on novel instances of the RGB-D part affordance dataset (Myers et al, 2015): Rank Weighted F-Measures.

affordance is higher than others. The reason is that object parts labeled as *contain* have deep concavities which make them more discriminative for detection.

In our work, we have three free parameters: bin size for patch histograms, dictionary size for patches, and dictionary size for parts. In this experiment, the bin size for forming histograms for patch representation is 10, the patch dictionary size is 50, and the part dictionary size is 20. We also experimented with other values for these open parameters, to choose the best values and measure the sensitivity of our approach to these parameters. Ranked weighted F-scores R_1^w computed by changing these parameters are given in Table 4. As can be seen, our approach is not sensitive to a particular selection of these parameters. Since these parameters are associated with part segmentation, MRF global optimization is resilient to their specific choice.

Affordance Detection of Novel Object Categories In order to prove the generalization ability of our method, we evaluated it on novel object categories. We used the novel category split of the RGB-D part affordance dataset (Myers et al, 2015). The dataset is split into two parts, which allows a two-fold cross validation[4]. The evaluation results in terms of R_1^w are provided in Table 5. As can be observed, our method performed better than the other state-of-the-art methods for all the affordances. It shows the strength of using a bottom-up approach for object part segmentation which proves its use for the affordance detection task. Figure 7 shows some qualitative results of our experiment. Objects used for training are shown in Fig. 8. We are interested in detecting the affordances of the objects which are shown in the first row of the figure. As can be noticed in the second row of the figure, objects tend to be segmented into functional parts. The segmented parts are then used for the affordance detection. The third row in Fig. 7 shows the results of the affordance detection. The object parts highlighted in red are those that afford the functionalities given above the objects.

We also provided the evaluation results of our method on novel object categories in terms of F_1^w for first-rank affordances in Table 6. We can see that F_1^w for *support* affordance is lower than other affordances. The reason is that there are objects of only two classes associated with this affordance, namely *shovel* and *turner*. Thus it makes it difficult for the *support* classifier to generalize to a novel object category.

Affordance Detection of Cluttered Scenes To show the applicability of our approach in occluded environments, we applied it on cluttered scenes of the RGB-D part affordance dataset (Myers et al, 2015). This dataset contains three different scenes. Each scene is captured in multiple views. We used the trained affordance classifiers of novel instance splits for this experiment. The evaluation is performed on the objects in the scenes after removing the table plane using the RANSAC algorithm. The quantitative results of our experiment based on rank weighted F-measures R_1^w is given in Table 7. Some qualitative results of our experiment are also

[4] The reader may refer to Section 6 for a complete list of objects and their corresponding category splits.

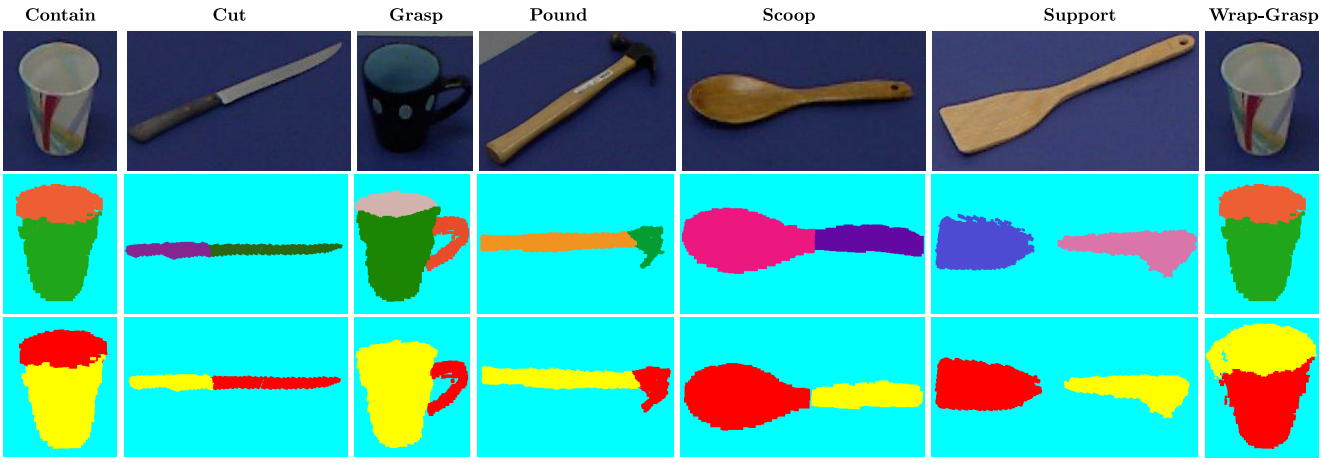

Fig. 5 Qualitative results on novel object instances of the RGB-D part affordance dataset. Labels: affordances to be detected for the objects in the first row. First row: RGB images of the objects. Second row: segmented object parts. Third row: results of the affordance detection on the object parts. The object parts highlighted in red afford the functionalities given by the labels.

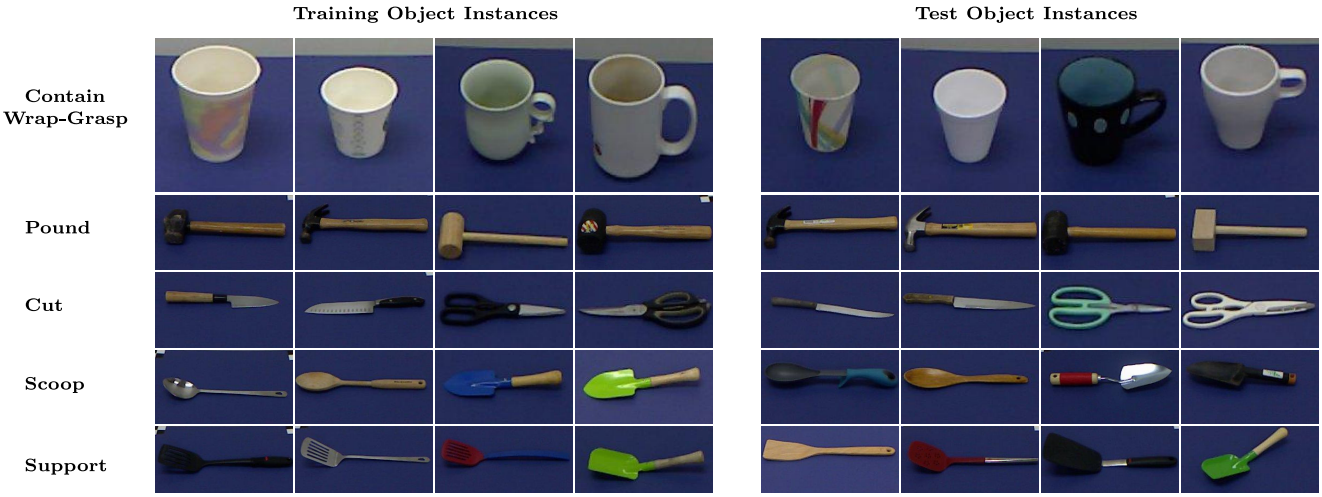

Fig. 6 Examples of training and test objects used for novel object instances in Fig 5. For simplicity only two instances of each category are shown.

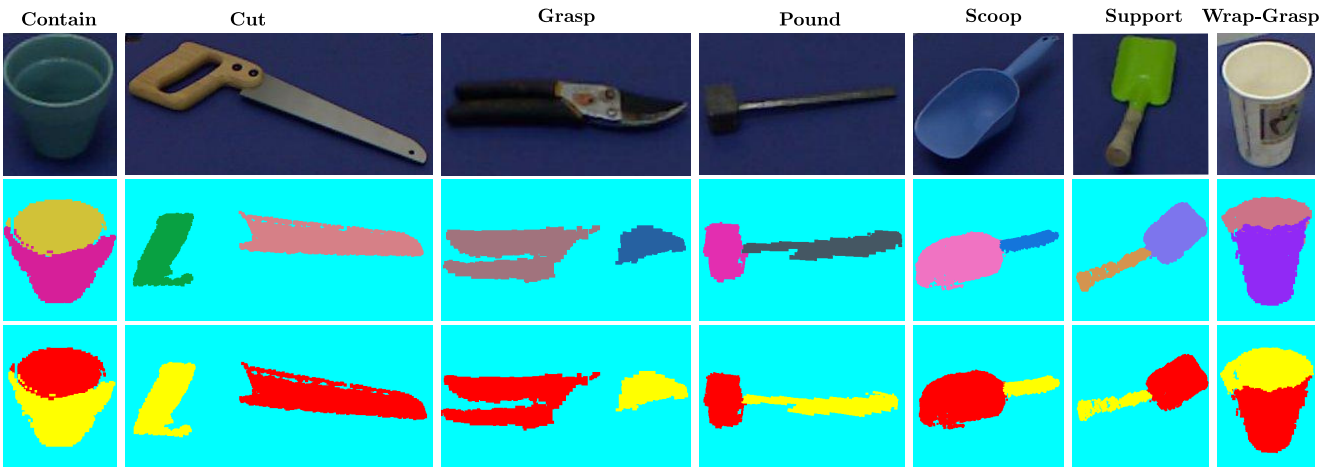

Fig. 7 Qualitative results on novel object categories of the RGB-D part affordance dataset. Labels: affordances to be detected for the objects in the first row. First row: RGB images of the objects. Second row: segmented object parts. Third row: results of the affordance detection on the object parts. The red-highlighted object parts afford the functionalities given in the labels.

Method	Grasp	Cut	Scoop	Contain	Pound	Support	Wrap-grasp	Average
Our Method	0.50	0.57	0.37	0.68	0.23	0.49	0.36	0.46

Table 3 Affordance prediction on novel instances of the RGB-D part affordance dataset (Myers et al, 2015): Weighted F-Measures.

Patch Dictionary Size								
Patch Dictionary Size	**Grasp**	**Cut**	**Scoop**	**Contain**	**Pound**	**Support**	**Wrap-grasp**	**Average**
10	**0.33**	0.28	0.22	0.32	0.12	0.23	0.16	0.24
30	**0.33**	0.29	0.24	0.35	0.11	0.09	0.28	0.24
50	0.31	**0.30**	**0.29**	**0.39**	0.11	**0.27**	**0.29**	**0.28**
70	0.32	0.28	0.24	0.32	**0.13**	0.23	0.19	0.24
Part Dictionary Size								
Part Dictionary Size	**Grasp**	**Cut**	**Scoop**	**Contain**	**Pound**	**Support**	**Wrap-grasp**	**Average**
10	0.32	0.28	0.22	0.32	0.12	0.23	0.18	0.24
20	0.31	**0.30**	**0.29**	**0.39**	0.11	**0.27**	**0.29**	**0.28**
30	**0.33**	0.28	0.23	0.35	0.12	0.23	**0.29**	0.26
40	**0.33**	0.28	0.25	0.36	**0.14**	0.23	0.33	0.27
Patch Feature Bin Size								
Patch Feature Bin Size	**Grasp**	**Cut**	**Scoop**	**Contain**	**Pound**	**Support**	**Wrap-grasp**	**Average**
5	0.32	0.28	0.23	0.33	**0.12**	0.23	0.23	0.25
10	0.31	**0.30**	**0.29**	**0.39**	0.11	**0.27**	**0.29**	**0.28**
20	**0.33**	0.28	0.22	0.34	**0.12**	0.23	0.25	0.25

Table 4 Affordance prediction on novel object instances of our method for different values of free parameters: patch dictionary size, part dictionary size, and patch feature bin size.

Method	Grasp	Cut	Scoop	Contain	Pound	Support	Wrap-grasp	Average
Our Method	0.19	**0.18**	**0.28**	**0.32**	**0.08**	0.11	**0.32**	**0.21**
Our Method with an RBF Kernel	0.22	0.06	0.10	0.18	0.04	0.06	0.07	0.10
Our Method with a Linear Kernel for Curvatures	**0.27**	0.15	0.20	0.30	0.05	0.09	0.29	0.19
Our Method with Histogram Part Features, Linear Kernel	0.24	0.08	0.23	0.24	0.04	**0.15**	0.21	0.17
Our Method with Histogram Part Features, RBF Kernel	0.25	0.05	0.16	0.25	0.03	0.09	0.19	0.15
Our Method with LCCP Parts (Stein et al, 2014)	0.26	0.00	0.00	0.17	0	0	0	0.06
HMP (Myers et al, 2015)	0.16	0.02	0.15	0.18	0.02	0.05	0.10	0.10
SRF (Myers et al, 2015)	0.05	0.01	0.04	0.07	0.02	0.01	0.07	0.04
VGG (Sawatzky et al, 2017)	0.18	0.05	0.18	0.20	0.03	0.07	0.11	0.12
ResNet (Sawatzky et al, 2017)	0.16	0.05	0.18	0.19	0.02	0.06	0.11	0.11

Table 5 Affordance prediction on novel categories of the RGB-D part affordance dataset (Myers et al, 2015): Rank Weighted F-Measures.

shown in Figure 9. It can be seen that our method performs better than other methods on average, and detects most of the affordances in presence of clutter in the scenes. This emphasizes the value of using a part-based affordance detection approach. In some cases, e.g. for the *support* affordance, we obtain more false positives. The reason is that if object parts are small or largely occluded, the estimation of surface normals is noisy (Fig. 10), which affects affordance detection. This can be alleviated by integrating multiple views, which is worth exploring in the future. Furthermore, in our approach, the affordances are detected on single parts. This may result in false positives especially in occluded scenes (Fig. 11). This false positive rate can be reduced by integrating affordances of neighboring parts. For example, given that the bowl of the ladle in Fig. 11 affords containing, its handle cannot afford supporting. Learning the relationships between adjacent affordances is a promising avenue for future work.

Discussion As can be seen from the evaluations, the affordance class *pound* has the lowest performance. One reason is that the training set contains only instances of the two object classes hammer and mallet that are marked with the *pound* class. In the test data, the affordance appears for the object classes tenderizer, cup, and saw. Moreover, rank affordance labeling for the two object classes hammer and mallet is not consistent. For the first-rank affordance, parts of objects are labeled as the *pound* class and other parts of

Method	Grasp	Cut	Scoop	Contain	Pound	Support	Wrap-grasp	Average
Our Method	0.46	0.30	0.22	0.47	0.08	0.03	0.47	0.29

Table 6 Affordance prediction on novel categories of the RGB-D part affordance dataset (Myers et al, 2015): Weighted F-Measures.

Fig. 8 Examples of training and test objects used for affordance detection on novel object categories in Fig 7. For simplicity up to two instances of each category are shown. All objects with handles are labeled with the *grasp* affordance. For the test object categories tenderizer and shovel, two views of the same instance are shown in the fourth and fifth columns.

Method	Grasp	Cut	Scoop	Contain	Pound	Support	Wrap-grasp	Average
Our Method	**0.27**	**0.15**	**0.21**	**0.21**	0.04	0.03	**0.13**	**0.15**
HMP (Myers et al, 2015)	0.12	0.07	0.08	0.22	**0.08**	**0.05**	0.11	0.11
SRF (Myers et al, 2015)	0.12	0.03	0.11	0.18	0.02	0.02	0.10	0.08

Table 7 Affordance prediction on cluttered scenes of the RGB-D part affordance dataset (Myers et al, 2015): Rank Weighted F-Measures.

objects as other affordance classes. These labels are opposite for the second-rank affordances of the same objects. Since we trained the *pound* classifier on first-rank affordances of object parts, this artifact of the dataset affects the performance numbers especially for these two object classes.

5.2 Robotic Grasping Experiment

In order to show the applicability of our approach, we applied it in real robotic scenarios. The grasping affordances, i.e. handle-grasp or wrap-grasp, are associated with a grasping action. For the other affordances, we needed to consider more than a single part. For example, to perform scooping, objects need to be grasped by their handles to be used for scooping. Since learning pairwise relations between affordances is beyond our current work, we validated affordance detection by affordance-specific grasps as a proxy for the real affordance. We associated grasp types to four different affordances, namely, rim grasp for contain, scoop and grasp and spherical grasp for the wrap-grasp affordance. Pound, cut, and support affordances were not used in this grasping experiment because the parts associated with them cannot be grasped by our robot.

The experimental setup for grasping objects consists of a robot with two KUKA 7-DoF Light-Weight Robot 4+ arms with servo-electric 3-Finger Schunk SDH-2 dexterous hands. There is a Kinect sensor mounted in front of the robot for capturing the RGB-D data. We used 11 objects in our experiment as shown in Figure 12. As can be seen in Table 8, objects might consist of multiple parts and have multiple affordances associated with them. Each scene-affordance combination was tested 10 times for grasping. We evaluated our approach on single objects as well as multiple objects in different scenes by computing the grasp success rate.

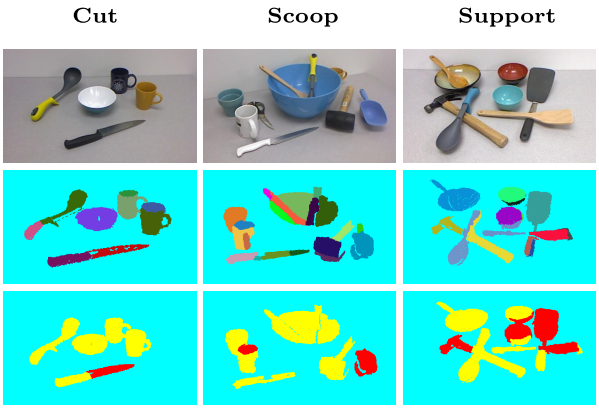

Fig. 9 Qualitative results on cluttered scenes of the RGB-D part affordance dataset. Labels: affordances to be detected for the objects in the first row. First row: RGB images of the objects. Second row: segmented object parts. Third row: results of the affordance detection on the object parts. The red-highlighted object parts afford the functionalities given in the labels.

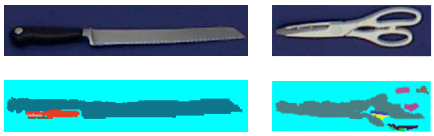

Fig. 10 Segmentation error for patches. Top row: RGB image of objects; bottom row: segmented patches based on Region Growing Segmentation. Each patch is shown in a different color. Patch segmentation uses surface normals of adjacent points. This results in a false segmentation in disconnected areas or areas with too few points.

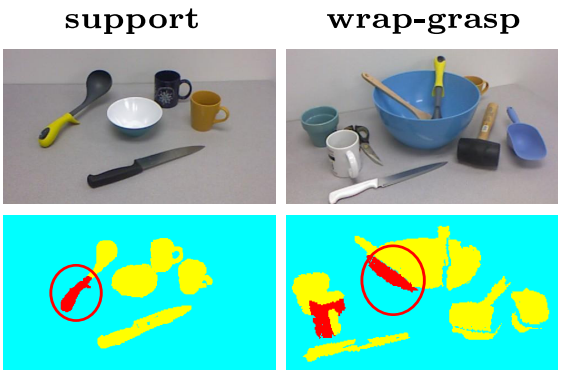

Fig. 11 False positives for support and wrap-grasp affordances. Affordances are detected on single parts. This may result in false positives especially in occluded scenes as indicated by the red circles. Using affordances of neighboring parts can reduce the false positive rate.

The grasping task proceeds as follows: Given a scene and a particular affordance, all the parts in the scene that have the affordance should be grasped by the robot. A grasp is considered successful if the robot can successfully grasp and lift the object.

In our experiment, we obtain pointclouds from the Kinect. As mentioned in Section 5.1, we use RANSAC to remove the ground plane. Our part segmentation method

Fig. 12 Objects used in the grasping experiment.

is then applied to the remaining points after ground-plane removal. We used the learned affordance classifiers of the novel-instance split of the RGB-D part affordance dataset (Myers et al, 2015) for affordance detection of the segmented parts. There are three novel objects in our experiment that do not exist in the dataset, namely the pitcher, the pot, and the container. The parts are grasped at their centers with a fixed gripper orientation.

Robotic Grasping Experiment on Single Objects We performed robotic grasping experiment on objects as shown in Fig. 12. The robot is asked to grasp the objects based on their affordances (Table 8). The grasp success rates of our experiment for scenes consisting of one object are shown in Table 8. We also provide quantitative results of affordance detection on single objects in Table 9. The results are the average of 10 grasp trials. Some qualitative results of our experiment are shown in Figure 13. As can be seen, the contain and scoop affordance classes have a high grasp success rate. This is due to the fact that these affordances are associated with surfaces of deep concavities. Since the robot performs rim grasps on these parts, it has enough free space for grasping.

The grasp success rates for the *wrap-grasp* and *grasp* affordance classes are a bit lower than the others. For the *wrap-grasp* affordance class, the reason is that we use a view of object and not a full 3D object model for grasping. Thus, the grasp associated with this affordance is not well centered on the object, reducing the robustness of the grasp.

Most parts associated with the *grasp* affordance (namely handles) cannot be picked up from the table by the robot (e.g. the handle of turners or ladles). To be graspable, such handles must be held up into free space by supporting them with other objects such as containers or bowls. These parts must be grasped with high precision to avoid collisions.

Robotic Grasping Experiment on Scenes We evaluated the grasp success rates for scenes consisting of multiple objects which have the same affordances. For each affordance, we evaluated three different scenes as shown in Figure 14. Each scene contains two object parts that are associated with the same affordance. For these scenes we followed the same experimental procedure as for the single objects. Table 10

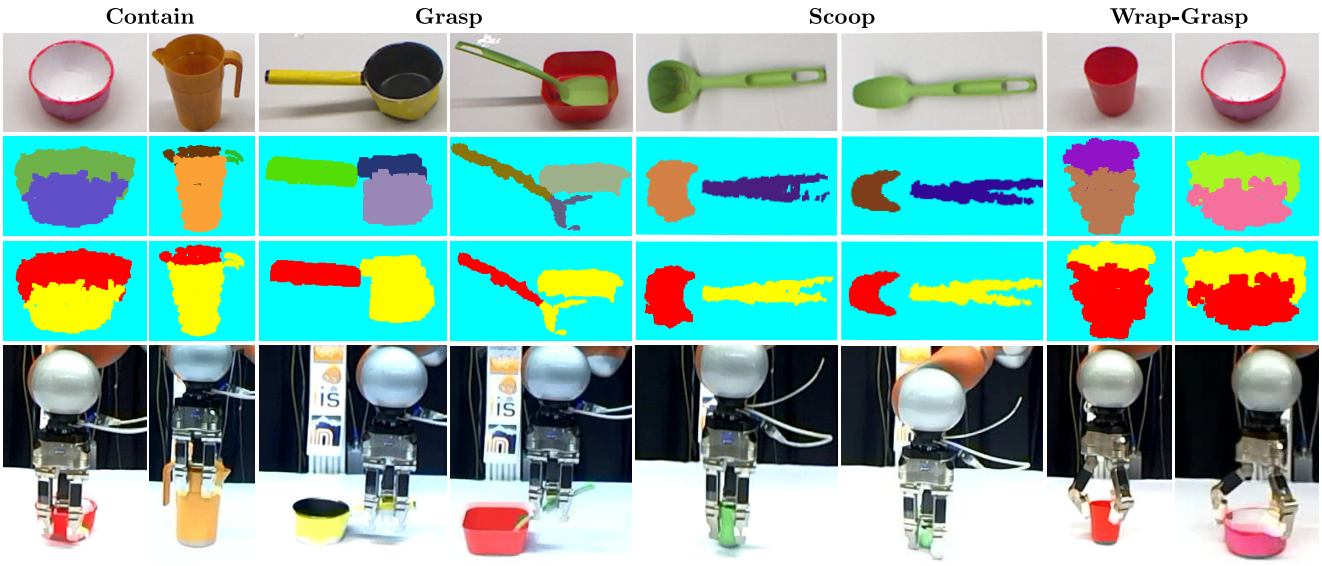

Fig. 13 Robotic grasping experiment on single objects. The robot is asked to grasp objects based on the given affordances shown above the objects. First row: RGB images of the objects, Second row: segmented object parts, Third row: results of the affordance detection on the object parts. Parts that afford the given functionalities are highlighted in red. Fourth row: grasp execution on the detected object parts.

	Grasp	Scoop	Contain	Wrap-Grasp	Average
Bowl	-	100	100	70	90
Container	-	-	100	-	100
Cup	-	-	100	100	100
Pitcher	-	-	100	70	85
Pot	90	-	80	80	83
Turner	100	-	-	-	100
Scoop	80	90	60	-	77
Ladle	70	100	100	-	90
Average	85	97	91	85	91

Table 8 Grasp Success Rate for single objects in %. The dashes indicate that the objects did not have the respective affordance.

	Grasp	Scoop	Contain	Wrap-Grasp	Average
Bowl	-	100	100	90	97
Container	-	-	100	-	100
Cup	-	-	100	100	100
Pitcher	-	-	100	78	89
Pot	100	-	80	100	93
Turner	100	-	-	-	100
Scoop	90	90	60	-	80
Ladle	78	100	100	-	93
Average	92	97	91	92	94

Table 9 Accuracy for Affordance Detection of single objects in %. The dashes indicate that the objects did not have the respective affordance.

shows the results of the grasping evaluation on the scenes as averages of 10 grasp trials. The grasping success rate is computed based on single objects in the scenes. We also provide accuracy of affordance detection for the objects in these scenes in Table 11. Figure 15 shows some qualitative results from our experiment. The results emphasize again that our approach performs well in the presence of clutter thanks to the part-based representation. Furthermore, we can see that the evaluation results are similar to the single-object experiments. This indicates the stability of our method across different objects and scenes.

Fig. 14 Scenes that are used in the grasping experiment.

Affordances	Grasp Success Rate
Grasp	78
Scoop	100
Contain	94
Wrap-Grasp	88
Average	90

Table 10 Grasp Success Rate for Scenes in %.

Discussion The robotic experiment showed that our approach can be used in real scenarios and cluttered scenes especially when objects are different than training data. Our

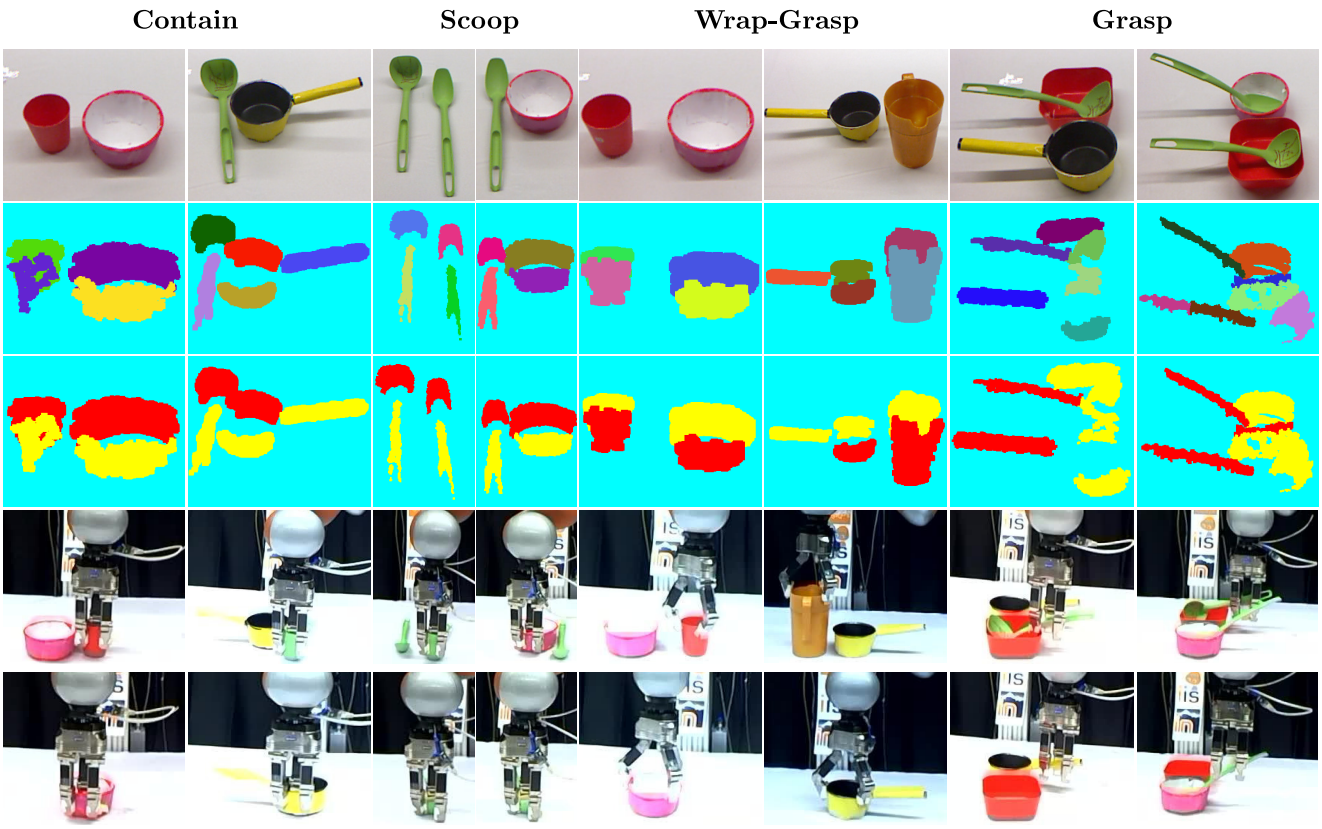

Fig. 15 Robotic grasping experiment on scenes. The robot is asked to grasp objects based on the given affordances on top of the scenes. First row: RGB images of the scenes, Second row: segmented object parts, Third row: results of the affordance detection on the object parts. Parts that afford the given functionalities are highlighted in red. Fourth-Fifth row: grasp executions on the detected object parts.

Affordances	Accuracy of Affordance Detection
Grasp	78
Scoop	100
Contain	95
Wrap-Grasp	97
Average	93

Table 11 Accuracy of Affordance Detection for Scenes in %.

grasping experiment serves as an indication of successful affordance detection. In this experiment, as the focus is not on elaborate grasp strategies, grasping is simplified by placing objects at known orientations. Practical grasping would require pose estimation of graspable parts and consideration of clutter.

6 Conclusions

We presented here a novel method for part-based affordance detection on RGB-D data. We showed that a part-based representation (where parts are driven from affordances) improves affordance detection performance (Section 5.1) and can generalize better when faced with novel objects.

We aimed to create a link between object part segmentation and affordance detection to improve the affordance detection performance. This can be seen as a step towards learning a functional representation of objects. Our work opens new avenues for future work in functional representation of objects. In the following, we discuss some possible future directions.

Refining Affordances Using Neighboring Parts We applied affordance detection on single object parts. As mentioned in Section 5.1, integrating affordances of adjacent parts can improve affordance detection of single parts especially in the presence of occlusion and clutter.

Using a Multi-View Object Representation In this paper, we used a single-view approach. As shown in Fig. 10, the estimation of surface normals which is required for object segmentation is error-prone in areas with too few points. Using a multi-view approach can alleviate this problem and subsequently improve object segmentation and affordance detection.

Guiding Object Representation with Robotic Tasks In this work we focused on individual object parts and the affor-

dances associated with them. Taking this one step further, one might ask how *tasks*, acting on affordances, can give rise to object representations. Tasks generally involve multiple affordances in combination (*grasping* a handle of a hammer to *pound* its head onto a nail) and in sequence. Thus, relations between multiple object parts and their affordances will be important. Analogously to this work, two complementary research questions are how task demands can drive the visual characterization of objects in terms of their parts, and how opportunities of task execution can be inferred from perceptual data.

Appendix

In Table 12, we include the list of object categories and instances used for a two-fold cross validation for novel object instances and categories in Section 5 and available from the RGB-D part affordance dataset (Myers et al, 2015). Second column of Tab. 12 shows category split for each object category used for affordance detection of novel object categories. The third and forth columns show the split number of object instances used for affordance detection of novel object instances.

References

Aldoma A, Tombari F, Vincze M (2012) Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 1732–1739

Bo L, Ren X, Fox D (2013) Unsupervised feature learning for RGB-D based object recognition. In: Experimental Robotics, Springer, pp 387–402

Desai C, Ramanan D (2013) Predicting functional regions on objects. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops

Fei-Fei L, Perona P (2005) A bayesian hierarchical model for learning natural scene categories. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, IEEE, vol 2, pp 524–531

Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 32(9):1627–1645

Fidler S, Leonardis A (2007) Towards scalable representations of object categories: Learning a hierarchy of parts. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp 1–8

Fu H, Cohen-Or D, Dror G, Sheffer A (2008) Upright orientation of man-made objects. In: ACM transactions on graphics (TOG), ACM, vol 27, p 42

Gibson JJ (1977) The Theory of Affordances. Perceiving, Acting, and Knowing: Toward an Ecological Psychology pp 67–82

Gibson JJ (1979) The Ecological Approach to Visual Perception. Psychology Press

Hart S, Dinh P, Hambuchen K (2014) Affordance templates for shared robot control. In: Artificial Intelligence and Human-Robot Interaction, AAAI Fall Symposium Series, Arlington, VA. USA

Hart S, Dinh P, Hambuchen K (2015) The affordance template ros package for robot task programming. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, pp 6227–6234

Hermans T, Rehg JM, Bobick A (2011) Affordance prediction via learned object attributes. In: IEEE International Conference on Robotics and Automation (ICRA): Workshop on Semantic Perception, Mapping, and Exploration, pp 181–184

Katz D, Venkatraman A, Kazemi M, Bagnell JA, Stentz A (2014) Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. Autonomous Robots 37(4):369–382

Koppula HS, Saxena A (2014) Physically grounded spatio-temporal object affordances. In: European Conference on Computer Vision, Springer, pp 831–847

Laga H, Mortara M, Spagnuolo M (2013) Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes. ACM Transactions on Graphics (TOG) 32(5):150

Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer vision and pattern recognition, 2006 IEEE computer society conference on, IEEE, vol 2, pp 2169–2178

Leung T, Malik J (2001) Representing and recognizing the visual appearance of materials using three-dimensional textons. International journal of computer vision 43(1):29–44

Margolin R, Zelnik-Manor L, Tal A (2014) How to evaluate foreground maps? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 248–255

Myers A, Teo CL, Fermüller C, Aloimonos Y (2015) Affordance detection of tool parts from geometric features. In: International Conference on Robotics and Automation (ICRA)

Nguyen A, Kanoulas D, Caldwell DG, Tsagarakis NG (2016) Detecting object affordances with convolutional neural networks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 2765–2770

Norman DA (1988) The psychology of everyday things. Basic books

Object Category	Category Split Number	Instance Numbers in the First Split	Instance Numbers in the Second Split
Bowl	1	1, 2, 4, 6, 8	3, 5, 7, 9, 10
Cup	1	1, 2, 3	4, 5, 6
Hammer	1	2, 4	1, 3
Knife	1	2, 3, 4, 6, 8, 12	1, 5, 7, 9, 10, 11
Ladle	2	1, 3, 4	2, 5
Mallet	1	2, 4	1, 3
Mug	1	3, 4, 9, 10, 12, 15, 16, 17, 18, 20	1, 2, 5, 6, 7, 8, 11, 13, 14, 19
Pot	2	2	1
Saw	2	1, 2	3
Scissors	1	1, 3, 5, 6	2, 4, 7, 8
Scoop	2	2	1
Shears	2	2	1
Shovel	2	1	2
Spoon	1	2, 3, 4, 7, 8	1, 5, 6, 9, 10
Tenderizer	2	2	1
Trowel	2	1, 2, 5	3, 4
Turner	1	1, 2, 3, 4	5, 6, 7, 8

Table 12 List of object instances for a two-fold cross validation on novel object instances obtained from the RGB-D part affordance dataset (Myers et al, 2015).

Omrčen D, Böge C, Asfour T, Ude A, Dillmann R (2009) Autonomous acquisition of pushing actions to support object grasping with a humanoid robot. In: 9th IEEE-RAS International Conference on Humanoid Robots, IEEE, pp 277–283

Platt J, et al (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers 10(3):61–74

Rabbani T, Van Den Heuvel F, Vosselmann G (2006) Segmentation of point clouds using smoothness constraint. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36(5):248–253

Rezapour Lakani S, Rodríguez-Sánchez A, Piater J (2017) Can Affordances Guide Object Decomposition Into Semantically Meaningful Parts? In: IEEE Winter Conference on Applications of Computer Vision (WACV)

Richtsfeld A, Mörwald T, Prankl J, Zillich M, Vincze M (2014) Learning of perceptual grouping for object segmentation on rgb-d data. Journal of visual communication and image representation 25(1):64–73

Rivlin E, Dickinson SJ, Rosenfeld A (1995) Recognition by functional parts. Computer Vision and Image Understanding 62(2):164–176

Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. Tech. rep., DTIC Document

Rusu RB, Cousins S (2011) 3D is here: Point cloud library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 1–4

Sawatzky J, Srikantha A, Gall J (2017) Weakly supervised affordance detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Schmidt M (2007) UGM: A matlab toolbox for probabilistic undirected graphical models. `http://www.cs.ubc.ca/~schmidtm/Software/UGM.html`

Stark L, Bowyer K (1991) Achieving generalized object recognition through reasoning about association of function to structure. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(10):1097–1104

Stark M, Lies P, Zillich M, Wyatt J, Schiele B (2008) Functional object class detection based on learned affordance cues. Computer Vision Systems pp 435–444

Stein CS, Schoeler M, Papon J, Wörgötter F (2014) Object partitioning using local convexity. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Varadarajan KM, Vincze M (2011) Affordance based part recognition for grasping and manipulation. In: Workshop on Autonomous Grasping, ICRA

Wang J, Yuille AL (2015) Semantic part segmentation using compositional model combining shape and appearance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1788–1797

Yao B, Ma J, Fei-Fei L (2013) Discovering object functionality. In: The IEEE International Conference on Computer Vision (ICCV)