University of Innsbruck

Institute of Computer Science

Intelligent and Interactive Systems

Affordance-Driven Visual Object Representation

Safoura Rezapour Lakani

Ph.D. Dissertation

Supervisor: Justus Piater

2nd July 2018

To my mother for being a compassionate and wise supporter and for teaching me to be independent. And to my brother for his support and sympathetic company.

Acknowledgments

First, I would like to thank my supervisor, Justus Piater, for giving me an opportunity to work in the intelligent and interactive systems group and for the useful comments.

Furthermore, I thank my colleagues in IIS whom I had the opportunity to work with them. Especially, Antonio for useful discussions on my work, Philipp and Erwan for proof-reading my thesis, and Cornelia for helping in administrative works.

Finally and most importantly, I want to thank my mother and my brother for their support and love in whole my life in general and during my PhD study in particular.

Abstract

Visual object representation has attracted substantial interest during the last decades. Besides being one of the fundamental challenges of computer vision, it also poses a central challenge to many robotic applications. In these applications, we do not only want to recognize objects but we also want to interact with them. In this context, we are specifically interested in applications involving manipulation and grasping in indoor scenarios. In these scenarios, reasoning about affordances of objects such as graspability, pour-ability, or cut-ability are paramount. Thus, making a link between visual object representation and their affordances plays an essential role in these applications.

This thesis deals with the problem of object representation by affordances. The hallmark of our object representation is the notion of parts. We argue that affordances are mostly associated with the parts of objects. For example, the head of a hammer affords pounding or the blade of a knife affords cutting. The distinction of our work compared to current state-of-the-art part based object representations is that in our work parts are driven from the affordances themselves. We present here a number of methods and techniques for a part-based object representation, part-based affordance detection, and actualizing affordances in robotic tasks. We aim at providing methods that robustly generalize to novel objects and are applicable in real robotic scenarios.

In our work, we use RGB-D data obtained from a Kinect sensor. We represent the RGB-D data based on parts which carry functional meaning. We then propose a part-based affordance detection approach. Since parts are shared among objects, affordances can thus be detected in novel objects. We then actualize affordances in robotic tasks which generally involve multiple affordances. As an example, scooping beans from a box with a ladle needs grasping the ladle's handle and scooping with the ladle's mouth. Thus, we learn relations between object parts and their affordances for performing tasks.

The proposed contributions which were integrated in a coherent framework have been evaluated on a number of robotic tasks and a publicly available RGB-D affordance dataset. We obtained a high object segmentation performance compared to the other state-of-the-art part segmentation methods on RGB-D data, even in the presence of clutter. Most importantly, we obtained a high affordance detection performance superior to other baseline methods. We also evaluated our framework in different grasping and manipulation tasks. The evaluation proved the applicability and generalization of our approach in real world scenarios and novel scenes.

Contents

List of Figures

List of Tables

Declaration

By my own signature I declare that I produced this work as the sole author, working independently, and that I did not use any sources and aids other than those referenced in the text. All passages borrowed from external sources, verbatim or by content, are explicitly identified as such.

Signed: Safoura Rezapour Lakani Date: 4th of July 2018

Chapter 1

Introduction

Humans know effortlessly how to use objects in their daily life. For example, looking at a mug, we infer its graspablity or fillability because of having a handle and a container part. Despite the variation in shape and color of the objects, we are still able to detect their functional properties. Looking at the objects in Figure 1.1, we infer the cup, the bowl, and the pitcher being fillable. Even though these are different objects and their containing parts have different shapes, they afford the same functionality.

The functional properties of an object which offer action possibilities to an agent are known as *affordances*. The definition of affordances originated in psychology (Gibson, 1979, 1977). It was initially defined as *the action possibilities in the environment in relation to the action capabilities of the agent* (Gibson, 1979, 1977; Norman, 1988). For most of the human-made objects (e.g. kitchen objects or tools), affordances can also be defined as functional properties (Myers et al., 2015; Nguyen et al., 2016). Following this definition, affordances are present in the objects by their design.

Detecting affordances is also a crucial aspect in robotics. Robots need to understand and interact with their environments. In order to conduct this interaction, they need to perform tasks on and with the objects in their surrondings. For example, as shown in Figure 1.2, the robot might be asked to grasp a cup from the table or to scoop coffee beans with a spoon. In these tasks, reasoning about affordances of objects such as graspability and scoop-ability are important.

Even though detecting affordances is a straightforward task for humans, it is still a challenging problem in robotics. Robots need to generalize affordances to novel objects and scenes. For example, a robot which has to load a dishwasher (Jiang et al., 2012) must learn graspability and place-ability of novel objects on the trays of the machine. In the same way, a robot which uses different kitchen machines (Sung et al., 2018), should generalize the way to use them across novel kitchen appliance.

This generalizability can be achieved by associating affordances to parts of objects. For example, most kitchen machines can be manipulated with their handles being an espresso machine or a juice maker. In the same way, the outside of a cup affords grasping or the bowl of a spoon affords scooping. More precisely, not only the bowl of the spoon but most concave object parts afford scooping. Part-based affordance detection has an enormous impact in robotics. Since parts are shared among different objects, their affordances can also be generalized to novel objects.

Learning a part-based object representation is still a difficult problem in computer vision and robotics. Decomposing objects into parts has been studied in computer vision for decades (Fidler and Leonardis, 2007; Wang and Yuille, 2015; Stein et al., 2014a; Laga et al., 2013). These approaches are classified mainly into two categories. They either use a hierarchical object repre-

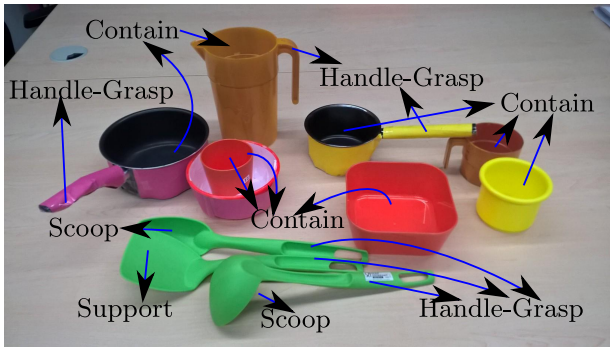
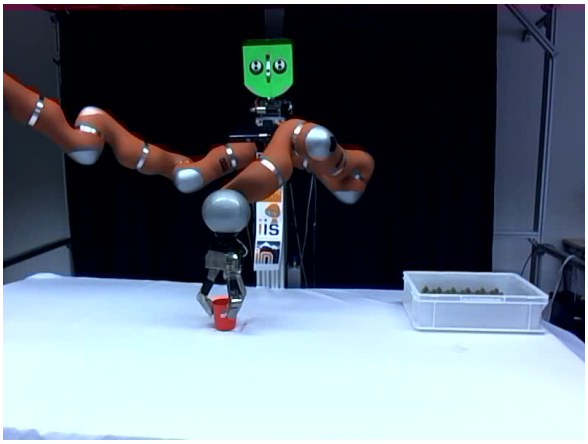
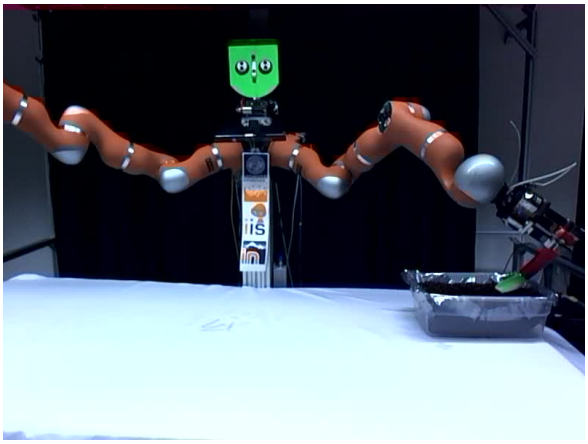


Figure 1.1: Objects and their functionalities.



(a) The robot is asked to grasp a cup from the table.



(b) The robot is asked to scoop coffee beans with the spoon.

Figure 1.2: Two different robotic manipulation scenarios. For grasping the cup in Fig. 1.2(a) or scooping coffee beans in Fig. 1.2(b), reasoning about graspability or scoop-ability of objects are important.

sentation (Fidler and Leonardis, 2007; Wang and Yuille, 2015) or geometrical properties (Stein et al., 2014a; Laga et al., 2013) for decomposing objects into parts. Even though these approaches have shown good performance in object classification, the resulting parts are not necessarily useful for detecting affordances.

Despite many studies in computer vision on object representation, there is a lack of part-based object representations useful for detecting affordances. In this thesis, we focus on learning a visual object representation which can fill this lack. To this end, we use affordances to implicitly guide object decomposition into parts. This guidance ensures that parts will carry functional meaning. The affordances are then detected at the level of object parts. Associating affordances with parts guarantees the generalizability of affordances to novel objects. This part-based approach allows us also to learn the relation between affordances of adjacent parts to perform robotic tasks.

List of Papers	Contributions
Safoura Rezapour Lakani, Mirela Popa, Antonio Rodríguez-Sánchez, and Justus Piater. CPS: 3D Compositional Part Segmentation through Grasping. In 12th Conference on Computer and Robot Vision, 2015, Best robot vision paper award.	Part segmentation guided with grasping (Chapter 4).
Safoura Rezapour Lakani, Antonio Rodríguez-Sánchez, and Justus Piater. Can Affordances Guide Object Decomposition Into Semantically Meaningful Parts?. In IEEE Winter Conference on Applications of Computer Vision, 2017.	Part segmentation guided with affordances (Chapter 4).
Safoura Rezapour Lakani, Antonio Rodríguez-Sánchez, and Justus Piater. Towards affordance detection for robot manipulation using affordance for parts and parts for affordance. Submitted to Autonomous Robots, 2017.	Part-based affordance detection (Chapter 5).
Safoura Rezapour Lakani, Antonio Rodríguez-Sánchez, and Justus Piater. Exercising Affordances of Objects: A Part-Based Approach. Accepted for publication for IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018.	Performing tasks based on relations between affordances (Chapter 6).

Table 1.1: Contributions included in this thesis.

List of Papers	Contributions
Philipp Zech, Simon Haller, Safoura Rezapour Lakani, Barry Ridge, Emre Ugur, and Justus Piater. Computational models of affordance in robotics: a taxonomy and systematic classification. In Adaptive Behavior, 2017.	Literature review on the use of affordances in robotics (Chapter 2.2).
Safoura Rezapour Lakani, Mirela Popa, Antonio Rodríguez-Sánchez, and Justus Piater. Scale-Invariant, Unsupervised Part Decomposition of 3D Objects. In Parts and Attributes Workshop at ECCV, 2014.	Part segmentation by fitting primitive shapes.

Table 1.2: Contributions not included in this thesis.

1.1 Overview of Contributions

The novel contributions of this thesis are either published (Rezapour Lakani et al., 2015, 2017b; Zech et al., 2017), accepted for publication (Rezapour Lakani et al., 2018), or submitted (Rezapour Lakani et al., 2017a) to scientific conferences or journals. Table 1.1 shows the publications which are included in this thesis. The publications which are not included in this thesis are shown in Table 1.2. Our contributions described in this thesis are as follows.

Affordance-Based Object Part Segmentation (Section 4) In this thesis, we focus on an object representation which can be used for detecting affordances and performing tasks that exercise them. Thus the first contribution of our thesis is guiding object representation by affordances. We initially used the grasping affordance (Rezapour Lakani et al., 2015) to decompose objects into parts. We then extended this idea to seven different affordances (Rezapour Lakani et al., 2017b) using the RGB-D part affordance dataset (Myers et al., 2015). The main contributions of this works are:

- using affordances for object part segmentation, and

- a new Markov Random Field (MRF)-based part segmentation method using affordance cues.

Part-Based Affordance Detection (Chapter 5) Since object parts are associated with affordances, the next step in our work is to detect the affordances of object parts. This work is conditionally accepted for publication in Autonomous Robots (Rezapour Lakani et al., 2017a). The main contributions of this work are:

- detecting affordances at the level of object parts, and
- an application of our affordance detection approach in real robotic grasping scenarios.

Learning Relations Between Affordances by Performing Tasks (Chapter 6) Affordances determine action possibilities of objects. After detecting the parts and affordances associated with them, the next step is to perform tasks that exercise them (Chapter 6). The corresponding work is conditionally accepted for publication for Autonomous Robots (Rezapour Lakani et al., 2017a) and accepted for publication for IEEE Robotics and Automation Letters and IROS 2018 (Rezapour Lakani et al., 2018). The main contributions of these works are:

- associating affordances with different grasp types, and
- making a link between affordances of object parts and tasks associated with them.

Chapter 2

Literature Review

Over the last decades, both object representation in computer vision and affordance detection in robotics have shown great importance. In this chapter, we give an overview of literature related to object representation and affordance detection. Furthermore, we explain how our work can link these two research domains together.

2.1 Literature Review in Object Representation

There have been several works on object representation in computer vision. Object representation determines what kind of features are extracted from objects and how these features are grouped together (Dickinson, 1999). The choice of these features and the mechanisms for grouping them highly depend on the application. These applications have been mostly related to object recognition, e.g. optical character recognition, object classification, and scene understanding. In this thesis, we are interested in robotic manipulation scenarios using visual represenations of objects. As mentioned in Chapter 1, in such scenarios, detecting affordances of objects is important. Thus, we are interested in representations which are useful for detecting affordances. In this section, we briefly review object representation approaches and discuss their applicability for affordance detection.

Object representation approaches can be classified mainly into two categories, flat and compositional approaches. In flat object representation approaches, state-of-the-art feature extraction methods are applied on the entire object (Pontil and Verri, 1998; Swain and Ballard, 1991; Jones and Rehg, 2002; Schiele and Crowley, 2000; Rusu et al., 2010). Objects are then classified based on the extracted features. These methods have shown good performance when objects are completely visible. But they are not robust to occlusion and clutter which makes them difficult to be used in real scenarios.

In order to make object representation methods more robust to clutter, compositional methods have been proposed which can be further categorized into hierarchical methods and part-based approaches. In hierarchical methods, objects are represented as a hierarchy of primitive features such as edges (Fidler and Leonardis, 2007; Wang and Yuille, 2015; Ommer and Buhmann, 2010; Wu et al., 2010; Si and Zhu, 2012, 2013), segmented image regions (Todorovic and Ahuja, 2008; Pepik et al., 2012; Scalzo and Piater, 2007), Histogram of Oriented Gradients (HOG) (Schnitzspan et al., 2009), or interest points (Sudderth et al., 2008; Zhu et al., 2009). These primitive features are subsequently combined by their frequency of co-occurrence obtained from multiple objects. These approaches represent objects at different levels of abstraction, starting from primitive features to the entire object. Hierarchical approaches are robust to occlusion, but there is no explicit notion of object parts. Furthermore, combinations of primitive features might not result in functional object parts useful for affordance detection.

In order to have an explicit notion of parts, part-based methods have been proposed. In these methods, part are either learned in a supervised or an unsupervised manner. In supervised part-based methods, regions of objects are manually labeled as parts (Kreavoy et al., 2007; Kalogerakis et al., 2010). These parts are then linked together based on probabilistic methods and in particular graphical models. These methods are usually trained on CAD models of objects and are not applicable in real scenarios. Moreover, the parts in these approaches must be completely visible to be detected. In unsupervised methods, objects are segmented into parts based on geometrical properties such as harmonics (Zheng et al., 2013), planes and geometric primitives (Richtsfeld et al., 2012), or fixed-size image regions (Pepik et al., 2012). Since affordances are associated with shape and geometrical properties, these methods have been applied especially in robotic grasping scenarios. Even though these methods have shown applicability for grasping, the assumed geometrical constraints do not necessarily hold in cluttered environments. Furthermore, the assumed geometrical properties do not necessarily result in functional object parts. We argue that these geometrical features should not be presumed but should be learned from the affordances.

In this thesis, we propose an object representation approach which can be used for detecting affordances. Object parts are obtained from affordances rather than only visual properties. We also follow a compositional approach for forming object parts to be robust to clutter.

2.2 Literature Review in Affordance Detection

Affordance detection has been studied extensively in robotics (Min et al., 2016; Yamanobe et al., 2017). As discussed in Chapter 1, affordances are defined as functional properties of objects that provide action possibilities to robots. In this thesis, we are interested in predicting affordances from visual representation of objects. Thus, we review only vision-based affordance detection methods. For a more detailed literature review on affordance detection please refer to our survey paper (Zech et al., 2017).

Affordances have been associated to different levels of abstraction in object representation, from a small group of pixels to the entire objects. These affordance prediction approaches can be categorized into three groups: object level methods predicting affordances on full object models, region-based approaches associating affordances to functional regions of objects, and local methods linking affordances to a fixed set of object pixels.

At the object level, affordances have been primarily linked to the structure of objects driven from 3D object models. Stark and Bowyer (1991) presented a structure-based representation of objects based on 3D surfaces and faces. Affordances are then linked with these surfaces and their relation with the entire object. Such structure-driven approaches perform well when precise 3D models of objects are provided. But they fail when we do not have precise object models. In order to reduce the effort of modeling objects, feature-based methods have been proposed. Aldoma et al. (2012) described an approach for linking affordances to features extracted from 3D object models and object poses which enables the robot to manipulate objects (e.g. grasping). But pose estimation in this approach also depends on CAD object models. For reducing the dependency on CAD object models, feature-based methods on partial views of objects have been used. Katz et al. (2014) described a method for associating affordances with geometrical features extracted from RGB-D pointclouds of objects. These extracted features are then combined with state-of-the-art classification methods such as Support Vector Machine (SVM) for affordance prediction. These feature-based affordance detection methods consider one type of features. But affordances might be related to multiple visual cues, such as object size, diameter, and shape features. Probabilistic assignment of multiple object features with affordances has been also studied in robotics. Hermans et al. (2011) linked affordances with visual attributes of

objects in a graphical model. In the same line of works, graphical models also have been used to link affordances with actions (Montesano et al., 2008; Lopes et al., 2007; Montesano et al., 2007; Moldovan et al., 2012). In these works, objects are usually represented by multiple visual features. The relations between objects and affordances are then learned by performing actions on objects. In these approaches not only affordances but also actions performed given them can be predicted. Linking action-object-affordances with manipulation trajectories has been investigated by Hart et al. (2015, 2014). In these works, the relations between objects and the robot's end-effector for performing actions are directly linked to the object model. The robot is able to predict affordances and manipulation trajectories after successfully recognizing objects. In these methods, affordances are detected using handcrafted visual features. In order to reduce the effort of designing features, Convolutional Neural Network (CNN)-based methods have been recently applied for affordance detection (Nguyen et al., 2017). Nguyen et al. (2017) trained a CNN for object and affordance detection. Conditional Random Fields (CRF) have been used for refining affordance detection. In all these methods, object categories need to be known beforehand. This limits the generalization of these approaches to novel object categories.

In order to overcome the generalization problem of object-based affordance detection methods, affordances have been detected on functional regions of objects. Often geometrical shapes or superquadrics have been used to obtain these functional regions (Laga et al., 2013; Fu et al., 2008; Varadarajan and Vincze, 2011; Rivlin et al., 1995). Affordances are then associated with these regions. These methods require CAD models of objects for fitting the geometrical shapes. As an alternative, feature-based methods have been used for detecting functional object regions. Stark et al. (2008) presented an approach for obtaining regions with conventional features such as Scale Invariant Feature Transform (SIFT). These regions are demonstrated by humans during training. Affordances are then linked to these regions and objects. In the same line of works, RGB-D data has been used for detecting functional surfaces of objects. These surfaces are then associated with affordances while performing actions (Omrčen et al., 2009; Stein et al., 2014b; Desai and Ramanan, 2013). In these methods, objects are detected and segmented into surfaces using geometrical properties such as convexity. Affordance prediction is then performed on the segmented surfaces. In these works, regions are detected independent of precise object models.

In order to have an approach applicable in cluttered environments, affordances have been detected only on a fixed set of object pixels i.e. *segments*. Often state-of-the-art feature extraction methods (such as Hierarchical Matching Pursuit (HMP) (Bo et al., 2013)) are combined with classification approaches such as an SVM to detect these segments (Myers et al., 2015). Recently, CNN-based approaches have been used for detecting affordances at the level of object segments (Nguyen et al., 2016; Sawatzky et al., 2017). Even though these approaches have shown a high generalization to novel objects, the object segments used in these approaches are not distinctive.

In this thesis, we aim to find the right level of abstraction in representing objects for associating affordances. Parts in our approach are driven based on affordances. This affordance-driven representation distinguishes our work from other studies to date. We argue that in this way, object decomposition results in parts useful for robotic manipulation.

Chapter 3

Technical Background

This chapter gives background on methods used in this thesis. We explain the learning methods used for object part segmentation and learning features for affordance detection. We also review the pairwise Markov Random Fields (MRF) as used for our part segmentation approach. We then describe the unsupervised feature learning algorithm based on autoencoders used for learning features from object parts for affordance detection.

3.1 Pairwise Markov Random Fields

In our part segmentation as we will discuss in Chapter 4, we follow a compositional approach. Parts are composed of a configuration of locally flat patches. In order to learn and infer a configuration of patches, we use a pairwise MRF.

An MRF is an undirected graphical model consisting of a set of random variables $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$. Each random variable can take one discrete value. We have observations $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$ and we would like to represent them by the random variables $\mathbf{x}$. The joint distribution of a certain assignment of random variables is computed as

$$p(\mathbf{x}|\theta) = \frac{1}{Z(\theta)} \prod_{c \in C} \psi_c(x_c|\theta_c) \tag{3.1}$$

$$Z(\theta) = \sum_{\mathbf{x}} \prod_{c \in C} \psi_c(x_c|\theta_c). \tag{3.2}$$

In Eqn. 3.1, C is the set of cliques in the graph (i.e. a subset of nodes such that there exists a link between all pairs of nodes in the subset), θ are the parameters of the model, x_c denotes the nodes in the clique c, and ψ_c are the potential functions. In our work, we use a pairwise MRF, that is, we consider only cliques of a maximum size of two. It is convenient to represent the potential function in an exponential form

$$\psi_c(x_c|\theta_c) = \exp\left(-E(x_c|\theta_c)\right), \tag{3.3}$$

where $E(x_c|\theta_c)$ is an energy function. The energy can have a linear form of features weighted by parameters θ_c

$$E(x_c|\theta_c) = \phi_c(x_c)\theta_c^T. \tag{3.4}$$

In Eqn. 3.4, ϕ_c is a function (e.g. a feature vector) applied on the nodes of the clique c. In this case the logarithmic form of the potential function can be written as

$$\log \psi_c(x_c|\theta_c) = \phi_c(x_c)\theta_c^T. \tag{3.5}$$

We use this logarithmic form of the potential functions for learning and inference in an MRF.

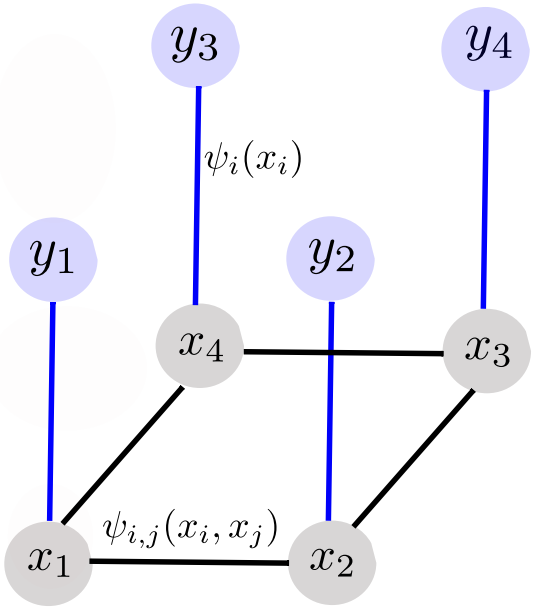


Figure 3.1: A grid view of a pairwise MRF. $x_1, \ldots, x_4$ are the random variables and $y_1, \ldots, y_4$ are the observations. $\psi_i(x_i)$ and $\psi_{i,j}(x_i, x_j)$ are the potential functions of the MRF.

3.1.1 Learning an MRF

The goal in learning an MRF is to estimate the parameters of the potential functions θ. One way for estimating these parameters is to maximiz the likelihood of the training data (i.e. minimizing the energy) over their coefficients by stochastic gradient descent. We follow the stochastic maximum likelihood algorithm as discussed by Murphy (2012) for learning an MRF. Let us consider the MRF from Eqn. 3.1 in log-linear form

$$p(\mathbf{x}|\theta) = \frac{1}{Z(\theta)} \exp(-\sum_c \theta_c^T \phi_c(x_c)). \tag{3.6}$$

The scaled log-likelihood is given by

$$l(\theta) = \frac{1}{N} \sum_i^N \log p(x_i|\theta), \tag{3.7}$$

where N indicates the number of training samples. Integrating Eqn. 3.6 into Eqn. 3.7 yields

$$l(\theta) = \frac{1}{N} \sum_i^N (\sum_c \theta_c^T \phi_c(x_i) - Z(\theta)). \tag{3.8}$$

Since we want to estimate the optimal parameters, we compute the gradient of the log-likelihood function given by Eqn. 3.8 with respect to the parameters θ. Omitting the details of calculation [1], the gradient has the form

$$\nabla_\theta l(\theta) = \frac{1}{N} \sum_i [\phi(x_i) - \mathbb{E}(\phi(\mathbf{x}))] \tag{3.9}$$

[1] For more detailed calculations, please consult (Murphy, 2012)

where $\mathbb{E}(\phi(\mathbf{x}))$ is the model's expectation of the feature vector. Computing this expectation is expensive because we need to consider all the possible combinations of variable assignments. Thus we use sampling based algorithm using Markov Chain Monte Carlo (MCMC). In our work, we used the Gibbs Sampling algorithm (Section 3.1.3).

In the stochastic maximum likelihood algorithm for fitting an MRF as shown in Algorithm 1, parameters θ are initialized randomly. The algorithm then iterates in multiple epochs. In each epoch, a minibatch of size B from the input data is used. For each minibatch, we compute the gradient as mentioned in Eqn. 3.9. The parameters are then updated based on the computed gradient and the learning rate η.

Algorithm 1 Stochastic maximum likelihood algorithm for fitting an MRF (Murphy, 2012)

Initialize θ randomly.
$k = 0, \eta = 1$
for each epoch **do**
  for each minibatch of size B **do**
    for each sample $s = 1 : S$ **do**
      sample $x^{s,k} \sim p(\mathbf{x}|\theta_k)$
    end for
    $\hat{\mathbb{E}}(\phi(\mathbf{x})) = \frac{1}{S}\sum_s \phi(x^{s,k})$
    for each training case i in minibatch **do**
      $g_{ik} = \phi(x_i) - \hat{\mathbb{E}}(\phi(\mathbf{x}))$
    end for
    $g_k = \frac{1}{B}\sum_{i \in B} g_{ik}$
    $\theta_{k+1} = \theta_k - \eta g_{ik}$
    $k = k + 1$
    Decrease step size η
  end for
end for

3.1.2 Inference in an MRF

An essential problem regarding the application of MRF models is how to infer the optimal configuration of nodes. Finding the optimal configuration is equivalent to finding the configuration with the minimum energy. Inferring such a configuration is an NP hard problem. In our work, we use the Loopy Belief Propagation (LBP) algorithm for inference in an MRF.

In the BP algorithm, we want to assign each node x to a state s which has the maximum belief among the other states for this particular node given the assignment of the other nodes. In a pairwise MRF, the probability distribution of a node x given the other nodes $\mathbf{x}$ can be written as

$$p(x|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\prod_s \psi_s(x_s)\prod_{s,t}\psi_{s,t}(x_s, x_t). \tag{3.10}$$

In Eqn. 3.10, ψ_s is the unary potential and $\psi_{s,t}$ is the pairwise potential. We would like to compute the probability of assignment for each node x in a particular state s. This probability is considered as the belief of the node x in the state s. Considering Eqn. 3.10, this belief can be

computed as

$$b_s(x_s) = p(x_s = s|\mathbf{x}) \tag{3.11}$$

$$\propto \psi_s(x_s) \prod_{s,t} \psi_{s,t}(x_s, x_t). \tag{3.12}$$

In order to compute this belief, we need to know the state of the other nodes as well. We can then propagate their states to compute this belief. This propagation is done through message passing. The message passed from the node x_t to the node x is computed as

$$m_{s\to t}(x_t) = \sum_{x_s} \psi_s(x_s)\psi_{st}(x_s, x_t) \prod_{u \in nbr_s \setminus t} m_{u\to s}(x). \tag{3.13}$$

After passing the messages, the belief of each node is updated. Algorithm 2 shows the belief propagation algorithm. As it can be seen, initially all the messages and beliefs are set to one. The algorithm is then run in multiple iterations. In each iteration the messages are propagated and the beliefs are updated accordingly. The algorithm stops when the beliefs have settled.

Algorithm 2 Loopy belief propagation for a pairwise MRF (Murphy, 2012)

1: Input: node potentials $\psi_s(x_s), \psi_{st}(x_s, x_t)$
2: Initialize messages $m_{s\to t}(x_t) = 1$ for all edges
3: Initialize beliefs $bel_s(x_s) = 1$ for all nodes
4: **while** beliefs don't change significantly **do**
5: Send message on each edge
6: $m_{s\to t}(x_t) = \sum_{x_s} \psi_s(x_s)\psi_{st}(x_s, x_t) \prod_{u\in nbr_s\setminus t} m_{u\to s}(x_s)$
7: Update belief of each node $bel_s(x_s) \propto \psi_s(x_s) \prod_{t\in nbr_s} m_{t\to s}(x_s)$
8: **end while**
9: Return marginal beliefs $bel_s(x_s)$;

After convergence of the LBP algorithm, each node x_s is assigned to the state s which has the maximum belief among the other states. These assignments determine the configuration of the nodes.

3.1.3 Markov Chain Monte Carlo Sampling

In Section 3.1.1, we explained that for learning an MRF, we need to compute the model's expectation of the feature vector $\mathbb{E}(\phi(\mathbf{x}))$. As mentioned earlier, computing this expectation is expensive because we need to consider all possible combinations of variable assignments. Thus, we use a sampling approach to estimate the model's expectation. In our work, we use Gibbs sampling, an efficient variant of the popular MCMC sampler. MCMC techniques are often applied to solve integration and optimization problems in large dimensional spaces (Andrieu et al., 2003). It is particularly used in Bayesian inference and learning (e.g. for computing the marginal posterior or the expectation of a posterior probability distribution) or for computing the partition function in a system with multiple states.

Let us consider that we have a set of D random variables $\mathbf{x} = \{x_1, \ldots, x_D\}$. The idea of the Monte Carlo algorithm is to draw an i.i.d. set of samples $\{\mathbf{x}^s\}_{s=1}^N$ from a target density $p(\mathbf{x})$ defined on a high dimensional space e.g. the set of possible configurations of a system or the space on which the posterior is defined. These N samples can then be used to approximate the target density.

The basic idea behind MCMC is to generate samples x^s while exploring the state space $\mathbf{x}$ using a Markov chain mechanism. By drawing correlated samples from the chain, we can perform Monte Carlo integration with respect to the target distribution $p(\mathbf{x})$.

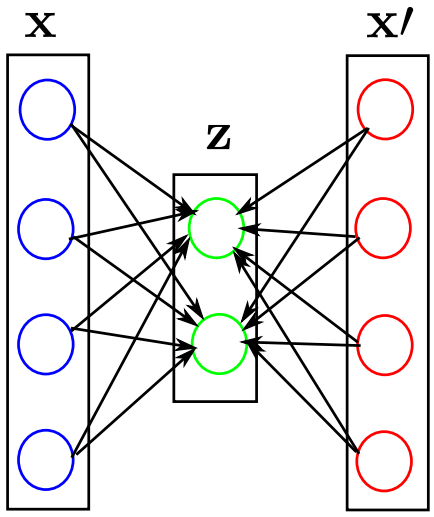


Figure 3.2: A schematic diagram of an autoencoder with one hidden layer. It has an input layer $\mathbf{x}$, an output layer $\mathbf{x}'$ and one hidden layer $\mathbf{z}$.

In Gibbs sampling, each variable is sampled sequentially conditioned on the value of all the other variables in the distribution. That is, given a joint sample $\mathbf{x}^s$ of all the variables, we generate a new sample $\mathbf{x}^{s+1}$ by sampling each component in turn, based on the most recent values of the other variables. For example, the new sample of a variable x_i^{s+1} is computed as $x_i^{s+1} \propto p(x_i|x_1^{s+1}, \ldots, x_{i-1}^{s+1}, x_{i+1}^s, \ldots, x_D^s)$.

In general, we do not need to compute the full conditional i.e. $p(x_i|\mathbf{x}_{-i})$ for variable i. We only use the variables that the variable i depends on them. In learning an MRF, we just consider the neighboring variables to the variable i. The Gibbs sampling algorithm is run for a few iterations. The generated samples $\{\mathbf{x}^s\}_{s=1}^N$ are then used for computing the model's expectation $\mathbb{E}(\phi(\mathbf{x}))$ of the feature vector.

3.2 Unsupervised Feature Learning with Autoencoders

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). It is a feedforward neural network with an input layer $\mathbf{x}$, an output layer $\mathbf{x}'$, and one or multiple hidden layers $\mathbf{z}$. Here, we use an autoencoder architecture with only one hidden layer (Figure 3.2). 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.

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 σ, such as the logistic sigmoid, to a *code* or *latent representation*

$$\mathbf{z} = \sigma(W\mathbf{x} + b) \in R^p, \; p \leq d, \tag{3.14}$$

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,

$$\mathbf{x}' = \sigma(W'\mathbf{z} + b'), \tag{3.15}$$

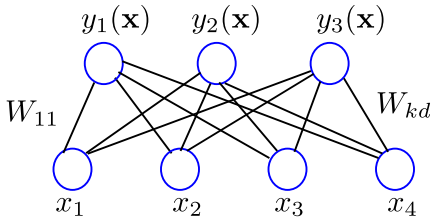


Figure 3.3: A schematic view of a neural network. Each layer consists of a set of nodes. There are connections between layers. In learning a neural network, we want to learn the weights $W_{11}, \ldots, W_{kd}$ of these connections.

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.

3.2.1 Learning Autoencoders

The goal in learning autoencoders is to estimate the weights of the mapping functions (i.e. W in Eqn. 3.14 and W' in Eqn. 3.15). Since autoencoders are a kind of neural networks, the backpropagation algorithm is used for learning these weights.

In the backpropagation algorithm, the gradients of the weights are computed layer by layer. Starting from the output layer, each layer below builds upon the results of the layer above. The gradients are then propagated backward through the layers.

Weights are updated iteratively by applying Stochastic Gradient Descent which uses the computed gradients of the backpropagation algorithm. Let us consider the example of a two layer neural network shown in Figure 3.3. The weights connecting the node y_k to the node x_j in the previous layer are updated as

$$w_{kj}^{\tau+1} = w_{kj}^{\tau} - \eta \frac{\partial E(\mathbf{w})}{\partial w_{kj}}|_{\mathbf{w}^\tau} \tag{3.16}$$

$$E(\mathbf{w}) = \sum_{m=1}^{M} E_m(\mathbf{w}). \tag{3.17}$$

where w_{kj} is the weight of connection between nodes k and j, M is the number of nodes at the layer below (i.e. d in Fig. 3.3), η is the learning rate, $\frac{\partial E(\mathbf{w})}{\partial w_{kj}}|_{\mathbf{w}^\tau}$ is the computed gradient applying the backpropagation algorithm using the weights of the previous iteration $\mathbf{w}^\tau$, and E is the error function. The most common-used error function for neural networks is the least square function,

$$E(\mathbf{w}) = \sum_{m=1}^{M} (y(x_n; \mathbf{w}) - t_{kn})^2. \tag{3.18}$$

The least square function computes the difference between the actual output t_{kn} and the predicted output $y(x_n; \mathbf{w})$. The gradient descent algorithm terminates when there is no substantial change in the weights.

Chapter 4

Affordances for Parts

Segmenting objects into parts is a fundamental contribution of this thesis. The goal here is to decompose objects into parts which afford certain functionalities thus providing the robot with the ability to perform a functional analysis of the scene. The robot should be able to generalize affordances to novel objects. Therefore, object decomposition should be also generalizable among unseen objects.

In order to have a generalizable approach, we follow a compositional bottom-up method. As mentioned in Chapter 1, compositional approaches have shown high generalization to novel objects (Fidler and Leonardis, 2007; Wang and Yuille, 2015; Ommer and Buhmann, 2010; Wu et al., 2010; Si and Zhu, 2013). In these methods, objects are composed of basic features. These basic features are then grouped to represent objects. These methods can detect objects in clutter and occluded scenes due to their compositional nature.

Since affordances are associated with the shape and geometry of objects, we apply a compositional approach on RGB-D pointclouds of objects considering only their depth information. In our approach, parts are composed of a combination of locally flat surfaces henceforth called *patches*. The patches are then linked together to form object parts.

The notion of object parts comes from affordances. Parts are the continuation of patches with the same affordances. Our decomposition approach is not combined with affordance detection (Nguyen et al., 2016, 2017). The reason is that the number of parts in our work is not limited to the number of affordances. Parts having same affordances might vary in shape among different objects. For example, container parts in tea pots, mugs, pitchers, or ladles afford fillability, but their shapes are different among these objects. We need to capture this variability in shapes of parts for object decomposition.

In the following, the papers related to our work on object part segmentation presented at the *2015 Conference on Computer and Robot Vision* and the *2017 IEEE Winter Conference on Applications of Computer Vision* are included. We first used *grasping* affordance for guiding object segmentation (Rezapour Lakani et al., 2015). In this work, a set of our own collected IKEA kitchen objects are used [1]. We manually labeled graspable and non-graspable regions of our IKEA objects. From the training data, we learn co-occurrence frequencies of the patches forming graspable or non-graspable regions. We then pursued a Bayesian probabilistic approach for inferring parts in novel objects. The paper related to this work was presented at the *2015 Conference on Computer and Robot Vision*.

Since grasping is associated with small regions of objects, we then extended our work by using affordances which are associated with functional parts of objects, such as handles, container parts, and pounding parts (Rezapour Lakani et al., 2017b). We used seven different affordances in this work, namely: handle-grasping, scooping, containing, pounding, supporting, cutting,

[1] `https://iis.uibk.ac.at/public/IkeaPartsObjectDataset/`

and wrap-grasping. We obtained training data from a publicly available dataset (Myers et al., 2015). This dataset contains RGB-D data of 105 different tools and kitchen utensils at multiple views. All pixels of the objects are manually labeled with their affordances. We define a part as a continuation of pixels with the same affordances. As mentioned earlier, parts are composed also of a set of patches. We then learn geometrical relationships between patches whether they belong to the same part or not. In addition, from the training data, we also learn the probability of co-occurrence between patches and parts. We use these two pieces of information in a pairwise Markov Random Field (MRF) to learn and infer parts. This work was presented at the *2017 IEEE Winter Conference on Applications of Computer Vision.*

CPS: 3D Compositional Part Segmentation through Grasping

Safoura Rezapour Lakani
University of Innsbruck
Innsbruck, Austria
safoura.rezapour-lakani@uibk.ac.at

Mirela Popa
University of Innsbruck
Innsbruck, Austria
mirela.popa@uibk.ac.at

Antonio J. Rodríguez-Sánchez
University of Innsbruck
Innsbruck, Austria
antonio.rodriguez-sanchez@uibk.ac.at

Justus Piater
University of Innsbruck
Innsbruck, Austria
justus.piater@uibk.ac.at

***Abstract*—Most objects are composed of parts which have a semantic meaning. A handle can have many different shapes and can be present in quite different objects, but there is only one semantic meaning to a handle, which is "a part that is designed especially to be grasped by the hand". We introduce here a novel 3D algorithm named CPS for the decomposition of objects into their semantically meaningful parts. These meaningful parts are learned from experiments where a robot grasps different objects. Objects are represented in a compositional graph hierarchy where their parts are represented as the relationship between subparts, which are in turn represented based on the relationships between small adjacent regions. Unlike other compositional approaches, our method relies on learning semantically meaningful parts which are learned from grasping experience. This compositional part representation provides generalization for part segmentation. We evaluated our method in this respect, by training it on one dataset and evaluating it on another. We achieved on average 78% part overlap accuracy for segmentation of novel part instances.**

***Keywords*-Compositional model, 3D object representation, object part segmentation, graspability**

I. Introduction

Computer vision deals with the understanding of the environment that surrounds us, enabling computers or/and robotic systems to acquire, process and understand the world based on visual information. In the case of a robot, given an image or the point cloud of an object, it should be able to assign a label to it but also to know how to interact with it. Learning how to interact with an object can be based on human knowledge, but is also directly linked to the structure of the object, which can be represented as a configuration of its parts. Interaction with the objects is influenced by their functionality, such as the way in which an object is grasped. Moreover, object representation can be structured according to their functionality.

For example, the pitcher depicted in Figure 1 will be grasped in different ways according to the goal of the required action. If the purpose is to pour something from it, the handle will be grasped. For holding an empty pitcher, a grasp on the body is also possible. This small example highlights the relation between object parts and the intended functionality. Our objective is to use this relation, in order to structure semantically meaningful object parts where the semantic aspect comes from the functionality such as grasping.

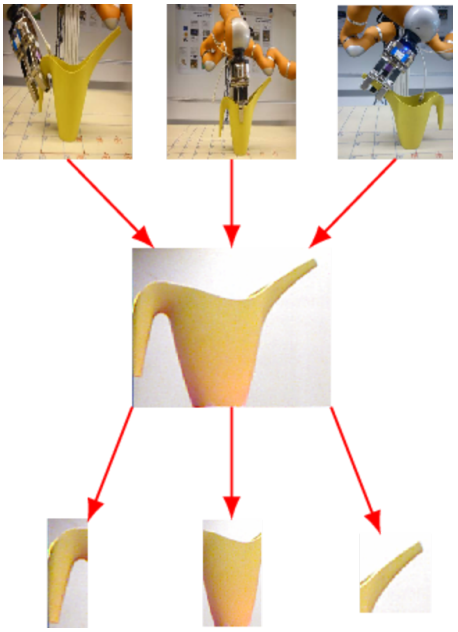


Figure 1. Object parts have semantics or functionality. Object parts can be formed based on a function such as graspablity.

Part-based object recognition has been studied in the computer vision domain for decades, for example in the work discussed in [1], [2], [3]. Representing an object by the configuration of its constituent parts is the key concept in these methods. In addition, representing a part itself is also critical. Parts should be represented distinctively in an object, and they should be semantically meaningful. Hence, decomposing an object into meaningful parts can have an important impact on object recognition and classification performance.

We present in this paper an approach towards object segmentation into semantically meaningful parts, which are formed from object regions obtained from robotic grasps. To this end, we developed a compositional, bottom-up approach starting from object points and culminating at object parts. Parts are described by the relationship between adjacent

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

patches. We focus on a scale-invariant and distinctive patch representation, which is especially useful for forming distinctive parts. We chose to employ a compositional representation instead of a flat model, for efficient capture of the large variability present in visual data.

The novelty of our approach is two-fold. First, the proposed algorithm forms semantically meaningful object parts in a hierarchical manner, by exploiting the object graspability. This approach, to the best of our knowledge, has not been followed before. Next, it provides a generalization mechanism for segmenting novel object part instances, by exploiting the relations between adjacent surface patches. Furthermore, our approach can facilitate visual reasoning by enabling the parsing of a scene into a set of semantically meaningful object parts.

In Section II we provide an overview of related work. Next we describe our bottom-up compositional method in Section III. In Section IV we introduce our probabilistic method for transferring the learned statistics from grasping to form parts in a novel object. We report the evaluation results in Section V, and finally we present our conclusions and future work in Section VI.

II. Related Work

Part-based object recognition based on RGB or RGB-D data has been thoroughly investigated in the literature. These approaches are classified into flat models and hierarchical models, and whether they do or do not use supervision.

In supervised methods based on RGB data, object parts are manually labeled in training examples [4]. The parts are then represented by extracting different types of features mainly in two ways, *globally* by extracting conventional feature descriptors from the parts [5], or *locally* by representing a part by its decomposition into small patches of specific sizes and in different resolutions where the patches are represented by different types of features [6], [7]. These approaches are then followed by classification, often using Support Vector Machines (SVM) [4], graphical [8] or other probabilistic models [9]. The main issue for these approaches is their generalization to novel objects which comes from the part representation. The global representations are not generalizable for novel object instances. However, local representations are more scalable, but they are not scale invariant. More precisely, one needs to perform an exhaustive search over different scales for low-level patches. Moreover, these low-level patches are not necessarily discriminative and can be found in different object parts. These two issues make these approaches difficult to generalize.

Hierarchical approaches as described in [10], [11], [12] tried to solve the generalization issue by learning object representations in a bottom-up, compositional manner. These approaches rely on the co-occurrence statistics of low-level features such as edges or contours extracted from training data. The advantage of this type of representation is the exploitation of the huge variability present in visual data in an efficient and general manner. However, these approaches are not guaranteed to produce meaningful object parts, which is an important aspect in part-based object recognition. Furthermore, these methods are based on 2D appearance data and are faced with various challenges posed by changes in illumination, color or texture of the objects.

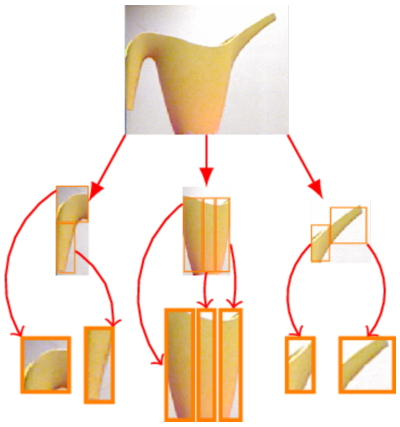


Figure 2. An object consists of a set of parts. Each part consists of a set of patches. The patches might not be discriminative on their own, but the relation between them can be distinctive.

One solution to this problem is to combine 3D cues with the RGB data. This idea is exploited for example in the work discussed in [13] for human body part segmentation and estimation. Even though the method is of the great use, it only addresses one category, the human body, while more emphasis needs to be given to object part representation across multiple categories. In the 3D space one important visual cue for object part segmentation is its geometrical structure, estimated from depth cues and surface normals, which is described in [14]. Moreover, the notion of convexity or concavity of object patches for unsupervised object segmentation into parts is discussed in [15]. In this work the semantic meaning of a part is assumed to be based on the local convexity or concavity.

We present here an approach to object part segmentation in 3D, designed to overcome the limitation of appearance-based representations as noted earlier in this section. Moreover, we exploit the hierarchical representation approach to obtain a generalizable part segmentation method.

III. Training a Compositional Part Model

The training process for learning our compositional part model (Figure 2) can be summarized as follows. The input to our system is an RGB-D point cloud, from which we use only its depth data. The representation of an object corresponds to its parts configuration (middle level of Fig 2). Parts are subsequently composed of regions that correspond to different local surface areas at the lowest level, which we call *patches*. Patches by themselves do not have the

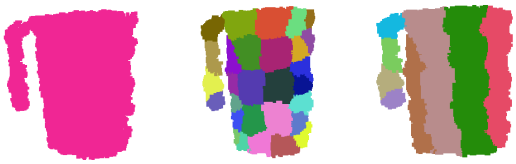
This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

representational power to segment object parts. However, those object parts can be obtained when we consider the *relationship* among their constituent patches. These relations among patches must be learned. This is one of the main contributions of this work, contributing to object part generalization. We explain the process next in more detail.

A. Obtaining patches from depth data

As already mentioned, patches form the lowest level of our compositional model. They are defined as locally flat surfaces, and their surface boundaries are defined based on relevant changes of the normal vectors. Thus, a patch by itself contains no discriminative information, while the relationship among neighboring patches contains sufficient discriminative information and can be used to represent object parts.

The starting point for creating a patch is given by supervoxels, since 3D point depth data is intrinsically very noisy. Thus, considering depth values directly would lead to unreliable patch approximations (local flat surfaces). We solve this problem by obtaining a more robust estimation of surface normals through the supervoxel algorithm presented in [16] (and available from the Point Cloud Library[1]). This method starts with evenly-distributed seeds, leading to a supervoxel representation by making use of k-means clustering. We then add an extra step, and merge the adjacent supervoxels whose mean normal vectors are close to parallel based on a pre-defined threshold. This merging step provides us with a set of locally flat patches as shown in Figure 3.



(a) (b) (c)

Figure 3. Patch representation of the pitcher object. (a) Original RGB-D point cloud; (b) Supervoxels; (c) Flat patches, which are the result of incremental supervoxel merging while their mean normal vector are close to parallel under a pre-defined threshold.

We would like to characterize a patch by the surface shape in its immediate surroundings, which is much more distinctive than the mostly-flat patch by itself. To this end, we represent a patch, henceforth called the *reference* patch, by a descriptor encoding the curvatures it forms in relation to each of its neighbors (Figure 4). For each patch adjacent to our reference patch, we compute the curvature formed by the pair of patches, as well as its angular location with respect to the reference patch. This angle is expressed with respect to the main axis of symmetry of the reference patch, which we obtain by computing its Extended Gaussian Image [17]. The

[1]http://pointclouds.org/

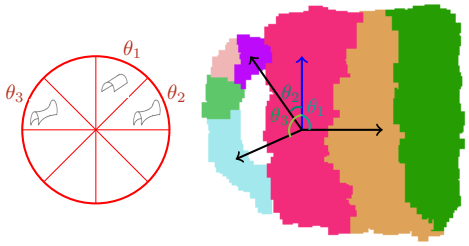


Figure 4. Patch representation. The patch descriptor is computed based on the relation of each reference patch with its neighbors. Its main axis of symmetry (blue arrow) defines the local coordinate system. Together with the reference patch, neighboring patches in different spatial locations may form different surface shapes, e.g. convex (θ_1) or hyperbolic (θ_2 and θ_3). The descriptor encodes surface curvature for each relative spatial location. The descriptor (red circle) is indexed by the quantized angle between the main axis of each reference patch and the centroid of each adjacent patch $(\theta_1, \theta_2, \theta_3)$, and contains, in each bin, the corresponding curvature.

patch descriptor is formed by quantizing each neighboring patch's location angles, and writing its associated curvature value into the corresponding orientation bin, as illustrated in Figure 4.

At most one curvature value is written into each bin of the descriptor. Bins not associated with a neighboring patch are set to zero. If more than one neighboring patch maps to a given bin, that bin's curvature value is computed from all those patches.

The structure of the descriptor is quite similar to the shape context [18]. The differences are that we consider only one reference point, and our descriptor is just invariant to in-plane rotation. Due to this similarity, in order to obtain the similarity between two descriptors, we make use of the distance measure used in shape context. Given two descriptor vectors P and Q which are composed of bins p and q, the distance $D(P,Q)$ between them is computed as the Euclidean distance between their constituent bins $C(p,q)$. In order to make the descriptor orientation invariant, we rotate it along the angular bins $T(q)$ and we compute the distance between the transformed descriptor $T(q)$ and q. The final distance is the minimum distance among them,

$$D(P,Q) = \sum_{p \in P} \underset{q \in Q}{\operatorname{argmin}} C(p,T(q)) + \sum_{q \in Q} \underset{p \in P}{\operatorname{argmin}} C(p,T(q)). \tag{1}$$

The final step is to construct a *patch dictionary* in order to assign patch types to test data. The patch features extracted as explained before are clustered using a hierarchical agglomerative clustering approach. The reason for using this type of clustering is because we do not have any knowledge about the number of clusters, nor the data distribution beforehand. First, each patch feature forms a cluster. Clusters are then merged incrementally if their

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

average distance is below a specific threshold (whose value is obtained as explained in Section III-B). Then, a patch codebook is constructed, where the mean cluster values are the codewords. The threshold used for clustering will be used as the distance threshold for each codeword.

B. Part representation from patches and grasp information

In our approach, parts are associated with a semantic meaning. This semantic meaning is obtained from the functionality of those object parts through grasping experience. Figure 5 shows an example where a robotic grasp is performed on the object regions which belong to one object part, such as its handle or body part. We extract information about the grasped object regions for forming object parts on novel objects.

We first decompose an object into patches as shown in Figure 6(d). Next, we teach the robot to grasp the object by manually moving its gripper to grasp the object (Fig. 6(a)). We consider the patches touched by the robot for collecting the information for forming object parts. To this end, we collect statistics about co-occurrence of the adjacent patches that can form an object part. Furthermore, we consider the patches which are adjacent but do not belong to the same object parts (based on grasping) and compute the distance between their descriptors as described in Section III-A. The minimum distance obtained from multiple object grasp examples is the **threshold** for patch clustering.

For object part segmentation, we obtain the probability that two adjacent patches form an object part. An object part is denoted by Y, a non-object part by $\bar{Y}$ and the patches by $X = \{x_1, \ldots, x_n\}$. Hence, we are interested in computing $p(Y|x_1,x_2)$ where x_1 and x_2 are adjacent, which can be written as

$$p(Y|x_1,x_2) = \frac{p(x_1,x_2|Y)p(Y)}{p(x_1,x_2)} \tag{2}$$

$$= \frac{p(x_1,x_2|Y)p(Y)}{p(x_1,x_2|Y)p(Y) + p(x_1,x_2|\bar{Y})p(\bar{Y})}, \tag{3}$$

where we consider a uniform prior probability distribution for Y and $\bar{Y}$. Therefore, we need to learn two probability distributions $p(x_1,x_2|Y)$ and $p(x_1,x_2|\bar{Y})$ for each two adjacent patches x_1 and x_2, which we collect from positive and negative examples.

To obtain the probability $p(x_1,x_2|Y)$, we already computed the patch clusters and the codebook as discussed in Section III-A. Next we consider pairs of adjacent patches x_1,x_2 which belonged to one object region during grasping and we match them to the learned patch codebook. We obtain all the possible codebook identifiers c_{x_i} to which a patch x_i can be matched. From multiple examples we obtain the probability $p(c_1,c_2|Y)$ of each co-occurring pair of codewords c_1 and c_2 forming a part.

To compute the probability $p(x_1,x_2|\bar{Y})$, we consider the adjacent patches which belong to the different object parts.

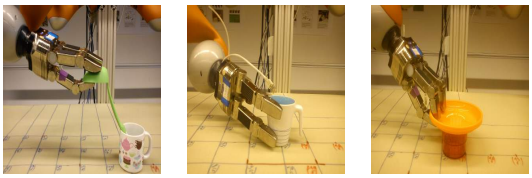


Figure 5. Kinesthetic grasp teaching for collecting patches that form a region and hence a part.

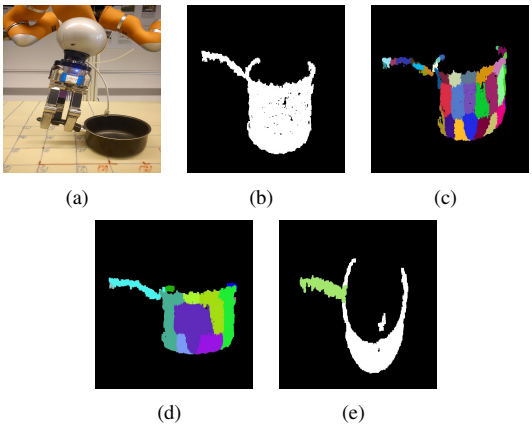


Figure 6. Figure 6(a) shows the kinesthetic grasp teaching on a *pot*. The original RGB-D data is depicted in Figure 6(b). The supervoxels are shown in Figure 6(c). Object decomposition into patches is shown in Figure 6(d). The patches contacted by grasping are shown in Figure 6(e).

In the same way, we match them to our codewords, and we obtain the probability $p(c_1,c_2|\bar{Y})$ of two co-occurring clusters which do not form a part. These two probability distributions will constitute our training data, and we employ them for inferring the semantically meaningful object parts.

IV. Part Inference in Novel Objects

We want to form parts in novel objects based on the learned co-occurrence statistics. As a first step, we decompose the novel object into patches. Starting from one patch, we estimate the co-occurrence probabilities between each patch and its neighbors. We then decide on merging the patch with its most probable neighbor as explained later in this section. We perform this procedure iteratively. At the first iteration, patches are merged to form regions. Next, regions are merged. We stop when no more merges are possible. At the end of the procedure, parts based on learned statistics have been identified.

A. From Patches to Regions

As mentioned earlier, we want to estimate the co-occurrences between each patch and its neighbors, based on the learned patch codebook. This information is then used to grow a region incrementally to form a part.

Let $Y(x_1,x_2)$ denote the predicate asserting that patches x_1 and x_2 belong to the same region. Then, $p(Y(x_1,x_2)|x_1,x_2)$,

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

or simply $p(Y|x_1,x_2)$ for short, denotes the probability that x_1 and x_2 belong to the same region.

Given the object patches, we start from a random patch x_1 and merge it with the patch x_e from its neighborhood $N(x_1)$ that is most probable to form a region with x_1, based on the learned codebook:

$$x_e = \underset{x \in N(x_1)}{\operatorname{argmax}}\, p(Y|x_1,x) \tag{4}$$

$p(Y|x_1,x)$ can be factorized as

$$x_e = \underset{x \in N(x_1)}{\operatorname{argmax}} \frac{p(x,x_1|Y)p(Y)}{p(x,x_1|Y)p(Y)+p(x,x_1|\bar{Y})p(\bar{Y})}. \tag{5}$$

Assuming identical, uniform priors for Y and $\bar{Y}$, (i.e. $p(Y) = p(\bar{Y})$), we can simplify Eqn. 5 as

$$x_e = \underset{x \in N(x_1)}{\operatorname{argmax}} \frac{p(x,x_1|Y)}{p(x,x_1|Y)+p(x,x_1|\bar{Y})}. \tag{6}$$

In order to compute the numerator in Eqn. 6, we marginalize over our patch codebooks C,

$$p(x,x_1|Y) = \sum_{c \in C} p(x,x_1|Y,c)p(c|Y). \tag{7}$$

The first term in Eqn. 7 can be further factorized as

$$p(x,x_1|Y) = \sum_{c \in C} p(x|x_1,Y,c)p(x_1|c,Y)p(c|Y). \tag{8}$$

The observation likelihood of a patch x_1 being matched to the codebook c is computed independently of Y; therefore $p(x_1|c,Y)$ can be written as $p(x_1|c)$. To compute $p(x|x_1,Y,c)$, we make use of the part co-occurrence table and marginalize over all codewords in the patch codebook $H = \{h_1,\ldots,h_n\}$ which can co-occur with c. Then we check whether x can be matched to them:

$$p(x|x_1,Y,c) = \sum_{h \in H} p(x,h|x_1,Y,c) \tag{9}$$

$$= \sum_{h \in H} \frac{p(x|h,x_1,Y,c)p(x_1|h,Y,c)p(h,c|Y)p(Y)}{p(x_1|Y,c)p(c|Y)p(Y)}. \tag{10}$$

Furthermore, we match the patch x to h independently of x_1,c,Y; the same holds for x_1. After substituting Eqn. 9 into Eqn. 7, we obtain

$$p(x,x_1|Y) = \sum_{c \in C} \sum_{h \in H} p(x|c)p(x_1|h)p(h,c|Y). \tag{11}$$

After calculating potential matches x for all the neighboring patches of x_1 in this fashion, we merge those that maximize the probability $p(Y|x_1,x)$ of forming a part.

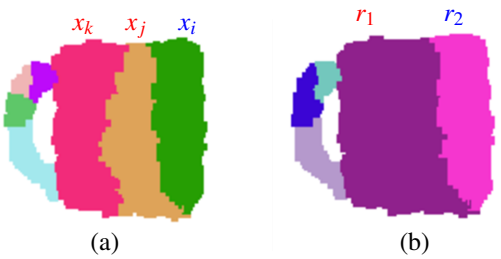


Figure 7. Merging regions based on their constituent patches. (a) object patches, (b) merged regions from patches. Starting from region r_1, x_j denotes its boundary patches. Patches adjacent to x_j inside r_1 are denoted x_k, and x_i inside r_2.

B. From Regions to Parts

From the above procedure, we obtain a collection of regions. To merge regions incrementally to compose parts, we follow a similar procedure as above, starting from a random region and merging neighboring regions. We start with a region r_1 as depicted in Figure 7.

We want to merge region r_1 to a region which is most probable to form a part with r_1, i.e.:

$$r_l = \underset{r \in N(r_1)}{\operatorname{argmax}}\, p(Y|r_1,r) \tag{12}$$

$$= \underset{r \in N(r_1)}{\operatorname{argmax}} \frac{p(r_1,r|Y)p(Y)}{p(r_1,r|Y)p(Y)+p(r_1,r|\bar{Y})p(\bar{Y})}, \tag{13}$$

where $p(Y|r_1,r)$ denotes the probability that r_1 and r belong to the same part. Assuming identical, uniform prior probability distributions for Y and $\bar{Y}$, we compute instead

$$r_l = \underset{r \in N(r_1)}{\operatorname{argmax}} \frac{p(r_1,r|Y)}{p(r_1,r|Y)+p(r_1,r|\bar{Y})}. \tag{14}$$

Any two adjacent regions contain adjacent component patches along their common boundary. We marginalize over these boundary patches to calculate $p(r_1,r|Y)$. As depicted in Figure 7, region r_1 is composed of patches x_j that are adjacent to region r. We would like to form a contiguous region by enforcing the co-occurrence of boundary patches with their neighbors in r_1 and r:

$$p(r_1,r|Y) = \sum_{x_j \in r_1} p(r_1,r,x_j|Y) \tag{15}$$

$$= \sum_{x_j \in r_1} p(r|r_1,x_j,Y)p(r_1|x_j,Y)p(x_j|Y). \tag{16}$$

We consider $p(r_1|x_j,Y)$, the conditional probability of a region given a boundary patch x_j, to be the conditional probability of the individual patches in that region that are

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

adjacent to x_j:

$$p(r_1|x_j,Y) = \prod_{\{x|x\in r_1 \wedge x\in N(x_j)\}} p(x|x_j,Y) \quad (17)$$

$$= \prod_{\{x|x\in r_1 \wedge x\in N(x_j)\}} \frac{p(x,x_j|Y)}{p(x_j|Y)}. \quad (18)$$

We compute $p(x,x_j|Y)$ in the same way as in Eqn. 11. Moreover, we consider a uniform probability distribution for $p(x_j|Y)$ based on the number of the patches in the region r_1, that is, $p(x_j|Y) = \frac{1}{N_C}$ where N_C indicates the total number of codewords.

In the same way, we calculate the conditional probability for region r based on those patches in r that are adjacent to patch x_j:

$$p(r|r_1,x_j,Y) = \prod_{\{x_i|x_i\in r \wedge x_i\in N(x_j)\}} p(x_i|r_1,x_j,Y). \quad (19)$$

Furthermore, we consider that $p(x_k|r_1,x_j)$ is independent of r_1 when its adjacent patches in r_1 are given:

$$p(r|r_1,x_j) = \prod_{\{x_i|x_k\in r \wedge x\in N(x_j)\}} p(x_i|x_j,Y) \quad (20)$$

$$= \prod_{\{x_i|x_i\in r \wedge x_i\in N(x_j)\}} \frac{p(x_i,x_j|Y)}{p(x_j|Y)}. \quad (21)$$

We compute the terms in Eqn. 20 analogously to Eqn. 17. After substituting them into Eqn. 15, we obtain

$$p(r_1,r|Y) = \sum_{x_j\in r_1} \prod_{\{x|x\in r_1 \wedge x\in N(x_j)\}} \sum_{c_1\in C, h_1\in H} p(x|c_1)p(x_j|h_1)p(h_1,c_1|Y) \prod_{\{x_i|x_i\in r \wedge x\in N(x_j)\}} \sum_{c_2\in C, h_2\in H} p(x_i|c_2) p(x_j|h_2)p(h_2,c_2|Y)N_C.$$

V. Experimental Evaluation

A. Experimental setup

We evaluated our part compositional method using two datasets: our own collected IKEA kitchen objects as well as a sample set of objects from the publicly available RGB-D Washington object database [19]. Our IKEA dataset as well as the part annotations we have made it available on our website (IKEA RGB-D object part database[2]).

The experimental setup for recording the IKEA 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 mounted in front of the robot for capturing the RGB-D data. It should be noted that our part segmentation method is independent of a specific robot setup. A different setup with different hands would yield similar segmentation results since we collect information about co-occurring patches involved in grasping.

[2]https://iis.uibk.ac.at/public/IkeaPartsObjectDataset/

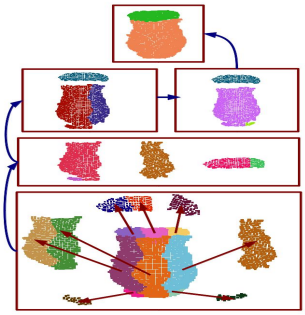


Figure 8. Compositional representation for forming object parts.

We recorded 18 kitchen objects, each at 39 different views (three different elevations and 12 different azimuths spaced 30 degree apart). We made annotations of object parts performed by kinesthetic grasp teaching. These annotation values were used for training as well as for the ground truth of object parts. We considered semantically meaningful grasps that are associated with one and only one part of an object. For the Washington dataset, we manually labeled the graspable object parts which we used as ground truth.

B. Experimental evaluation

The compositional and probabilistic framework for object segmentation allows us to generalize the segmentation to novel object parts where only some low-level patches are shared. The supervoxel merge threshold was kept at 10 degrees in our experiment. Figure 8 shows the compositional capability of our method in a real scenario. In order to show the applicability of our method for this object part generalization, we evaluated part segmentation on novel objects. To this end, we selected a random set of object classes from the IKEA object dataset. We used 70% of the objects from this set for training, from which 30% are used as the labeled part examples to structure and guide the clustering and collect statistics. Next, we applied the learned model to novel object instances and we evaluated the segmentation performance.

We used the maximum overlap [15], [14] of the segmented parts with respect to the ground truth as evaluation metric. For a pointcloud, we have a set $G = \{G_1,\ldots,G_M\}$ of human-annotated ground-truth parts and a set $S = \{S_1,\ldots,S_N\}$ of segments produced by the part segmentation method. Then for each ground-truth part, a segment with the greatest overlap is considered as the best estimator. The overlap between a pair of ground truth and part segment is computed as $\text{overlap}_i = \frac{|G_i \cap S_j|}{|G_i \cup S_j|}$. The overall score is computed as the weighted average based on the size of each ground-truth object part,

$$Wov = \frac{1}{\sum_i |G_i|} \sum_i |G_i| \cdot \underset{i}{\text{overlap}} \quad (22)$$

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

C. Results

We report on the following three evaluation conditions: 1. correlation of parts across different categories, 2. finding parts on new objects across different datasets, and 3. a comparison with the state of the art.

We evaluated the part segmentation based on novel, previously-unseen object instances in the IKEA dataset. Furthermore, to show the applicability of CPS across datasets, we used the parts learned from the IKEA dataset and evaluated them on the Washington RGB-D dataset. Although the Washington dataset is a very rich object dataset, many objects do not have a complex structure, composed of multiple parts. Therefore, we only considered objects composed of at least two parts such as mugs, caps and staplers. Some examples from our part segmentation method are shown in Figure 9. As can be seen, CPS decomposes the objects into semantically meaningful parts such as handles, bodies, etc. that have functions.

There are mainly two sources of errors for our method. The first is the view-based representation: As mentioned in Section IV, CPS examines adjacent connected object regions to form a part. However, due to the view-based representation as well as reflection and transparency of some object regions, some object regions are observed as disconnected regions, which is shown for the mug body in Figure 9(a) and the cap in Figure 9(g). The second source of errors is noise propagated from the low-level supervoxels: When the supervoxels are not accurate as demonstrated in Figure 11, object patches and hence the part segmentation will be affected by them. This problem can be seen in Figure 9(e). These issues and the possible solutions for them are discussed in Section V-D.

In order to show the applicability of our approach in complex scenes, we also evaluated CPS in 14 different scenes composed of IKEA objects, where the scenes contained novel objects as well. An example of our part segmentation method for a scene is shown in Figure 10.

Finally, we compared CPS with the recently-proposed Locally Convex Connected Patches (*LCCP*) algorithm [15]. We included two scenarios, one considering negative part examples (CPS) which is $p(x_1, x_2|\bar{Y})$ and the other without considering those (CPS-). Quantitative results of this experiment are shown in Table I. These results are computed based on overlap accuracy 22. As can be seen, CPS obtains promising results for part segmentation.

The method is implemented on an Intel Core I5 2.6 GHz processor. The overall part segmentation takes on average 5.6s for an object. The decomposition into low-level patches and feature extraction takes on average 5.4s, and the segmentation procedure takes on average 20ms.

D. Discussion

We have shown that our approach contributes to extracting semantically meaningful object parts. Furthermore, the

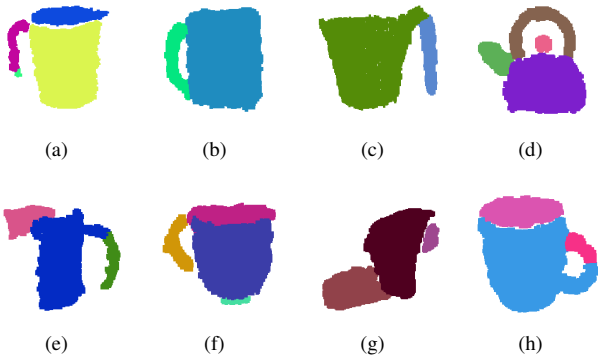


Figure 9. Examples of segmented objects in IKEA and Washington datasets.

(a) Original scene (b) Scene segmented into parts

Figure 10. Example of a scene and its segmentation into parts. The scene is first parsed into objects using PCL plane segmentation methods. The parsed objects were then segmented into parts using CPS.

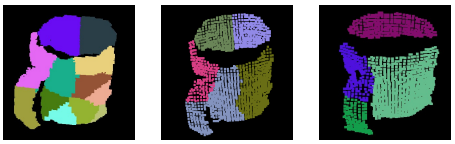


Figure 11. Example of a poorly-estimated part due to inaccurate supervoxel segmentation.

Method	IKEA objects	RGB-D mugs	RGB-D caps	RGB-D staples	IKEA scenes
CPS	89%	78%	68%	76%	84%
CPS-	89%	57%	64%	76%	83%
LCCP [15]	82%	81%	69%	79%	73%

Table I
OVERLAP ACCURACY FOR OBJECT PART SEGMENTATION. LCCP IS COMPARED WITH CPS, WITH AND WITHOUT USING THE TRUE NEGATIVE EXAMPLES (CPS-).

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

compositional nature of our method allows for a high degree of generalization for object parts since they can be learned on one dataset and transferred and evaluated on another one. We have also compared CPS with another state-of-the-art approach and obtained promising part segmentation overlapping accuracy.

Still, there are a number of issues which could lead to an improvement of our compositional model. First, even though our method is scale invariant, it is view-dependent. Thus, at this point our part representation is not 3D rotation invariant. View invariance may in the future be incorporated into the compositional model (section IV). Secondly, our model depends on correct supervoxel extraction. When the low-level supervoxel segmentation is poor, it affects the part segmentation. This problem could be overcome by either designing our model to be less dependent on the supervoxels, or by considering a different, more robust low-level segmentation method. In the long term we aim to build a part-based object model representation based on the proposed part segmentation method, by first addressing these issues.

VI. Conclusion

The contribution of our work consists of a novel compositional model, named CPS, that works with three-dimensional objects whose main characteristics are twofold: (1) *Features* at the lowest level of our model are not based on a combination of isolated points, but represent the relationship among neighboring patches. (2) Parts are semantically meaningful and are learned from grasping experience. We presented a statistical approach for segmenting object parts which is based on the grasped segments of an object. These two characteristics allow us to segment previously unseen objects into meaningful parts.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2013-2016 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 600918, PaCMan.

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, 2005.

[2] M. C. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," in *European Conference of Computer Vision (ECCV)*, 1998.

[3] M. Burl, T. K. Leung, and P. Perona, "Face localization via shape statistics," in *Workshop on Automatic Face and Gesture Recognition*, 1995.

[4] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, 2010.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005.

[6] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3D geometry to deformable part models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] X. Ren and J. Malik, "Learning a classification model for segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[8] S. M. A. Eslami and C. Williams, "A generative model for parts-based object segmentation," in *Advances in Neural Information Processing Systems*, 2012.

[9] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Workshop on Statistical Learning in Computer Vision at ECCV*, 2004.

[10] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[11] L. Zhu and A. L. Yuille, "A hierarchical compositional system for rapid object detection." in *Advances in Neural Information Processing Systems*, 2005.

[12] B. Ommer and J. Buhmann, "Learning the Compositional Nature of Visual Object Categories for Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, 2010.

[13] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[14] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *European Conference of Computer Vision (ECCV)*, 2012.

[15] C. S. Stein, M. Schoeler, J. Papon, and F. Wörgötter, "Object partitioning using local convexity," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[16] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2013.

[17] C. Sun and J. Sherrah, "3-D symmetry detection using the extended gaussian image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, 1997.

[18] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, 2002.

[19] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE International Conference on on Robotics and Automation*, 2011.

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2015 Conference on Computer and Robot Vision

Can Affordances Guide Object Decomposition Into Semantically Meaningful Parts?

Safoura Rezapour Lakani
University of Innsbruck
safoura.rezapour-lakani@uibk.ac.at

Antonio J. Rodríguez-Sánchez
University of Innsbruck
antonio.rodriguez-sanchez@uibk.ac.at

Justus Piater
University of Innsbruck
justus.piater@uibk.ac.at

Abstract

Most objects are designed for certain functionalities. For example, a knife is designed for cutting, and a hammer for pounding. Indeed, functionalities are not related to the objects themselves but to certain object parts; e.g., the blade of a knife affords cutting. A part can have different shapes and can exist in different objects such as a scraper or a peeler, but it carries the same functional meaning. There is a strong correlation between object parts and affordances. In this paper, we exploit this correlation to decompose objects into semantically meaningful parts. The semantics are limited here to object affordances. We evaluate our method on a part decomposition task, and obtained 77% weighted overlap with ground-truth object parts.

1. Introduction

Learning object representations and the affordances related to them is an important aspect in robotics. In robot vision, we are not only interested in recognizing and giving names to the objects but also in how to use them. The term affordance refers to the perceived and actual properties of the object [8, 17] that determine how the object might possibly be used. Affordances provide strong cues to the operation of objects. For example, a knife affords cutting another object, and a hammer affords pounding another object. An important aspect about affordances is that we know effortlessly what to do with the object; there is no need for a label or instruction.

As can be seen in Figure 1, affordances are not just related to the objects per se, but mostly to their parts. For example, the head of a hammer affords pounding, the handle of a hammer grasping and the inside of a mug containing. The latter is very useful in the robotics domain especially for generalization among novel objects. Robots must be able to deal with unfamiliar objects. Recognizing affordances can provide them with effective strategies for the interaction with objects. For example, every sharp object part can be used for cutting. It can be the blade of a knife or a scraper. Therefore, a notion of an object part is needed for efficient and generalizable affordance detection.

But what is an object part? Object segmentation into

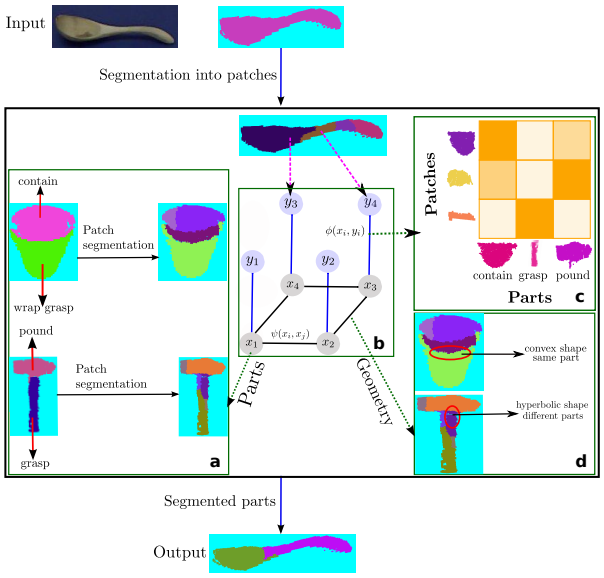


Figure 1. Object part segmentation based on affordances. Object parts in our model have semantic meaning based on their affordances such as pounding, grasping and containing. We learn a graphical model for part segmentation from locally-flat object patches based on two sources of information: 1) the potential of a patch 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.

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

parts has been widely used in computer vision [25, 6]. Although such methods have shown high performance in object recognition scenarios, the parts they produce are not necessarily applicable to robotic manipulation scenarios. There is a gap between object part decomposition and robotic affordances in that part decomposition and affordances are not linked together. As mentioned earlier, affordances provide a strong cue about the semantics of parts; making use of this cue can benefit object representation in terms of semantically meaningful parts. Here, we limit the notion of semantics to affordances.

This paper addresses the problem of segmenting an object in an RGB-D pointcloud into its parts. Affordances such as pounding, containing, or grasping are related to certain object parts (Fig. 1a). We propose a bottom-up segmentation approach that allows parts to be generalized among novel objects. Parts in our model are composed of locally flat patches. A Markov Random Field (Fig. 1b) relates those patches among themselves (Fig. 1d) as well as with affordances (Fig. 1c). This approach, which leads toward semantically meaningful parts, is the main contribution of our work.

In Section 2, we discuss related work for object part segmentation and affordance detection. We describe our affordance-based part segmentation method in Section 3. In Section 4, we explain inference and affordance detection based on our method. We report on the experimental evaluation for part decomposition and affordance detection in comparison to other state-of-the-art methods in Section 5.

2. Related Work

Learning object affordances based on visual features has been long investigated in the computer vision and robotics communities. Affordance detection has been either performed at the *global* object level, or through *local* object segments. At the object level, affordances can be assigned as object attributes [12]. Object level affordance is then inferred based on attributes derived from appearance features. Object affordances can also be detected based on the relationship of objects and the scene. In the work discussed by Katz *et al*. [15], the affordances are inferred from the orientations of object surfaces with respect to their underlying surface. The main drawback of object-based affordance detection is its limited generalization to novel objects.

At the local level, affordances have been related to geometric shapes in an object [24, 20], such as primitives derived from CAD models or superquadrics [1]. These methods make strong assumptions on geometric shapes and primitives and mostly operate on CAD models. This limits the applicability of these methods, and makes them difficult to apply in real domains. In order to make the local level affordance detection applicable in practice, pixel-wise affordance detection has been proposed [16]. In these methods object affordances are inferred from the aggregation of appearance features from fixed-size object segments. Although these approaches have shown good generalization performance, they are limited to fixed-size object segments. Moreover, the affordance prediction is error-prone since it is obtained from object pixels which do not carry any semantic meaning.

A compromise between fine and coarse affordance detection can be made using *object parts*. Decomposing objects into parts has been performed using only visual features [6, 7, 10, 5, 4, 28, 9] or geometric properties [25, 22, 19, 14, 27]. Although these approaches provide fairly good decomposition, the parts produced by these methods do not necessarily carry any semantic meaning or can be used for affordance detection. To overcome this problem, non-visual cues are used to guide the decomposition. In other work [26, 11, 13], decomposition was guided by non-visual cues such as actions. While these methods aim to bring semantics into the object representation, their focus is on decomposing a scene into objects rather than decomposing objects into parts.

To the best of our knowledge, most object part decomposition studies to date lack semantic and functional meaning of parts. In this work, we employ affordances to guide object part decomposition. We argue that in this way, decomposition will result in functional parts useful in a robotic manipulation scenario.

3. Learning a Part Model from Affordances

The input to our system is an RGB-D pointcloud. Since we are interested in capturing shape information, we use only its depth data. We consider an object to be represented by a configuration of its functional parts. We obtain functional parts based on labeled training data. We aim for generalization of object parts among different objects; hence we consider a *compositional* representation for object parts. A part in our model is represented by a configuration of locally flat surfaces, which we call *patches*. We then learn a pairwise relationship between patches which can form object parts. In the following we explain the learning procedure for our method in more detail.

3.1. Patch Segmentation

Patches are at the lowest level of our part representation. Patches are defined as locally flat surfaces but, if we consider the neighborhood of any of them, there exist noticeable changes in surface normals. Therefore, a patch itself does not carry any discriminative information, but its relationship with neighboring patches carries useful information.

In order to obtain the patches, we used the Region Growing Segmentation [18] (available in the Point Cloud Li-

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

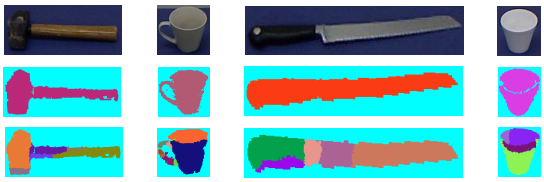


Figure 2. Patch segmentation from pointcloud. Top row: RGB image, middle row: pointclouds and bottom row: segmented patches; colors indicate different patches.

brary [21][1]). This algorithm segments a pointcloud into surfaces based on the angles between normals of adjacent points. The neighboring points get assigned to the same surface if the computed angle is less than a pre-determined threshold. As can be seen in Figure 2, this segmentation ensures the flatness of patches up to the specified threshold value.

We would like to represent a patch based on the geometry and surface shape. As mentioned earlier, a patch is represented based on its surface shape relationship among its neighbors. To represent this relationship for a patch, we consider points which are on the boundary of a patch with its neighbors. Since we are interested in encoding surface shape information, we compute surface normals of the boundary patches. Therefore a patch is represented based on surface normals of boundary points. Since patches do not have a fixed size, the number of boundary points and hence their normals vary among patches. To have a fixed representation for all patches, we made a histogram of surface normals.

In order to assign patch types to test data, we construct a dictionary of patches. To this end, we cluster patches based on histograms of surface normals. We use the K-Means algorithm to obtain the clusters. Then, a patch codebook $C = \{c_1, \ldots, c_n\}$ is constructed, where the mean cluster values are the codewords.

3.2. Part Representation from Patches and Affordances

A part in our model is formed based on patches and their affordance cues. The ultimate goal in our approach is to decompose a scene into parts. In this context, we must determine 1) the number of parts present in a scene and 2) the probability of a patch (or combination of patches) to be assigned to a part in the scene.

For the first issue, we need to learn the maximum number of part classes in a scene. One can argue that this number is limited to the maximum number of affordances for object parts. But the parts related to the same affordance might differ in shape; e.g. container parts have shapes that differ between, say, mugs and bowls. We thus learn a dictionary of object parts based on visual information as shown in Figure 1. As can be seen in the figure, the training data consist in labeled object part affordances such as container parts, scooping parts, etc. A part is composed of locally flat patches which can be seen for the example parts in Fig. 1. Let's consider a part z consisting of patches $\{y_1, \ldots, y_n\}$. We will assign patch types by matching $\{y_1, \ldots, y_n\}$ to the patch codebook $C = \{c_1, \ldots, c_n\}$, where c_{y_i} would correspond to the patch type c_i that best matches the patch y_i. Since object parts in our model are scale invariant, they might consist of different numbers of patches. Therefore, we represent a part by a histogram of its constituent patch types $\{c_{y_1}, \ldots, c_{y_n}\}$. In order to recognize part classes $X = \{x_1, \ldots, x_m\}$ in test data, we construct a dictionary of parts as follows. First, we cluster the parts represented based on histograms of patch types using the K-Means algorithm. Then, a part codebook is constructed, where the mean cluster values are the codewords.

After obtaining the part classes $X = \{x_1, \ldots, x_m\}$, the next step is to establish a relationship between patches $Y = \{y_1, \ldots, y_n\}$ and those part classes. To model this relationship, we will explore two sources of information. The first source of information relevant to modeling the relationship between part classes and patches is the probability of a patch type being a constituent of a part type $p(c|x), x \in X, c \in C$. Considering the type x of training part z and the types $\{c_1, \ldots, c_n\}$ of the constituent patches of z, we collect statistics of co-occurrences of the patch types $\{c_1, \ldots, c_n\}$ and part classes x from each training part z. We then learn the probability $p(c|x)$ based on the aforementioned statistics. We obtain the second source of information based on the pairwise geometric relation between patches in order to decide whether they can be assigned to the same part types. Considering two adjacent patches y_i and y_j, this geometric relationship is defined based on the curvature γ_{y_i,y_j} between them. As indicated in Fig. 1, the pairwise curvature value gives information on surface shape on the boundary of the patches. As an example, for the hammer in the Fig. 1, a hyperbolic surface shape determines that the patches belong to different parts (i.e. the handle and the head of the hammer). In contrast, a convex shape for the mug in Fig. 1 indicates that the patches belong to the same part.

To determine whether two patches can have the same part label, we train a classifier based on the pairwise curvature values as follows. We, first collect curvatures between pairs of adjacent patches which belong to the same part and those which do not belong to the same part. We then train a Support Vector Machine (*SVM*) classifier with an Radial Basis Function (*RBF*) kernel based on those curvature values. We use the score of the classifier to determine the probability of neighboring patches y_i and y_j belonging to the same part. Figure 1 illustrates the information flow start-

[1] http://pointclouds.org/

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

ing from locally flat patches to representing parts in novel objects.

3.3. MRF for Object Part Segmentation

For decomposing objects into parts, we use a pairwise Markov Random Field (*MRF*). We decompose objects into patches as discussed in Section 3.1. The nodes are the patches and their connections are edges in the graph.

The graph of an MRF consists of a set of cliques C (fully connected sets of nodes). The joint distribution of all nodes is represented based on the cliques in the graph,

$$p(X) = \frac{1}{Z} \prod_c \psi_c(x_c), \tag{1}$$

where $\psi_c(x_c)$ is called the potential function and Z is the normalization constant, integrated over all the states x,

$$Z = \sum_x \prod_c \psi_c(x_c). \tag{2}$$

It is common to represent the potential function as an energy function to simplify the problem from a product of potentials to a sum of energies. We restrict ourselves to pairwise MRF, i.e., we consider only cliques of size two. In this case, the energy function

$$E(x, y) = \sum_i \phi(x_i, y_i) + \sum_{i,j} \psi(x_i, x_j) \tag{3}$$

is composed of two terms, a unary potential ϕ and a pairwise potential ψ. The unary potential determines how likely an observation y_i belongs to a certain state/label x_i. The pairwise potential ψ encodes neighborhood information, i.e. how different the label of one variable is from that of its neighbor.

In our case, object patches are the observations and states are the finite discrete set of object parts in a scene. The number of states in our model is the maximum number of part classes in a scene. The unary term ϕ determines the likelihood that a patch y_i belongs to a part of class x_i as shown in Figure 1. This likelihood is obtained from the probability $p(c|x)$ that a patch type c belongs to a part of certain class x (see Section 3.2). We then compute the patch type c_{y_i} of y_i by matching it to the codebook C of patch types (Section 3.2). Finally, we obtain the unary term as

$$\phi(x_i, y_i; \theta_\phi) = \exp(-\theta_\phi * p(c_{y_i}|x_i)), \tag{4}$$

where θ_ϕ are the parameters of the unary energy function ϕ and $p(c_{y_i}|x_i)$ is obtained during training. As can be seen, the energy is minimized as the probability gets higher.

The pairwise term defines the pairwise neighborhood likelihood of patches y_i and y_j belonging to part classes x_i and x_j. Learning this pairwise neighborhood likelihood for each part class combination is computationally very expensive. Instead, we define a potential that reflects whether two patches belong to the same part class or not (Figure 1). We learn this potential based on the classifier trained from curvatures between adjacent patches γ_{y_i,y_j} as described in Section 3.2. We represent the score of the classifier as $\text{score}(\gamma_{y_i,y_j})$. If the patches have the same label, the score is non-negative, and negative if they are not. Since we use the *SVM* classifier, the score corresponds to the distance to the margin. We define the pairwise energy term based on this score. We penalize neighboring patches having different labels except for the cases determined by the curvature classifier as explained below:

$$\psi(x_i, x_j) = \begin{cases} 0, & x_i = x_j \\ t, & x_i \neq x_j, \text{score}(\gamma_{y_i,y_j}) < 0 \\ \exp(-\theta_\psi \cdot \text{score}(\gamma_{y_i,y_j})), & x_i \neq x_j, \text{score}(\gamma_{y_i,y_j}) \geq 0 \end{cases} \tag{5}$$

When the patches have the same label $x_i = x_j$, the energy is set to zero. Otherwise when labels are different ($x_i \neq x_j$), we consider their pairwise curvature value γ_{y_i,y_j}. The pairwise energy is set to a maximum value t subject to $\text{score}(\gamma_{y_i,y_j}) < 0$. The reason is to discourage adjacent patches from having different labels. And when $\text{score}(\gamma_{y_i,y_j}) \geq 0$, the energy is determined as a function of the score. The energy is reduced as the score gets higher. θ_ψ is the parameter/weights of the function. We learn the parameters of the model with stochastic gradient descent.

4. Inferring Parts in Novel Objects

In order to infer object parts in novel objects, we use the learned *MRF* model described in Section 3 as follows. We first segment the objects into patches $Y = \{y_1, y_2, \ldots, y_n\}$ and extract features from them following the procedure described in Section 3.1. We then compute the Euclidean distance between features extracted from the patches and the patch types $\{c_1, c_2, \ldots, c_m\}$ in the patch codebook C. The patches are assigned to the patch types $\{c_{y_1}, c_{y_2}, \ldots, c_{y_n}\}$ to which have the smallest distance. We use these patch types to compute the unary potential $\phi(x_i, y_i)$ between patches and part classes. Next, we compute the curvature between pairs of adjacent patches which is used for the pairwise potential $\psi(x_i, x_j)$ in our model. We consider the adjacency of the patches and represent them as a graph. Each patch is one vertex in the graph and the edges are determined by the patch adjacency. We compute unary and pairwise potentials as described in Section 3.3.

After constructing the graph of patches, we perform inference to find the best configuration of parts for an object/scene. We compute the configuration which minimizes the energy. The minimization of the energy in an undirected graph is a NP-hard problem, and exact inference is thus not possible. For this reason, we use a standard implementation of Loopy Belief Propagation (*LBP*) [23].

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

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 [16].

5. Experimental Results

We evaluated our method on the RGB-D part affordance dataset [16]. The dataset contains RGB-D images and ground-truth affordance labels for 105 objects. Each object pixel is labeled with the most likely affordance as well as all the possible affordances with their ranks. There are seven labeled affordances: grasp, cut, contain, pound, scoop, support and wrap-grasp as shown in Table 1. In our experiments, we only used the top-ranked affordance labels.

For training, we used the labeled data from the RGB-D part affordance dataset. We consider a part as the continuation of adjacent pixels with the same top-ranked affordances. We segment each part into patches based on the method discussed in Section 3.1. We set the threshold for the Region Growing Segmentation as mentioned in Section 3.1 to three degrees. We learned 50 patch clusters and 20 part clusters with the *K-Means* algorithm and used them in the MRF model as explained in Section 3.3. Finally, we used the Undirected Graphical Model package (*UGM*) [23] for inference and sampling. We used the learned MRF model for inferring parts in novel objects.

We computed the decomposition performance using two standard measures, Weighted Overlap (Wov) [2, 25] and Rand Index (RI) [3, 14, 27].

Wov measures the parts' maximum overlap with the ground-truth parts. For a point cloud, we have a set $G = \{G_1, \ldots, G_M\}$ of human-annotated ground-truth parts and a set $S = \{S_1, \ldots, S_N\}$ of segments produced by the part segmentation method. Then, for each ground-truth part, the segment with the greatest overlap is considered the best estimator. The overlap between a pair of a ground truth and a part segment is computed as $\text{overlap}_i = \frac{|G_i \cap S_j|}{|G_i \cup S_j|}$. The overall score is computed as the weighted average based on the size of each ground-truth object part,

$$\text{Wov} = \frac{1}{\sum_i |G_i|} \sum_i |G_i| \cdot \text{overlap}_i \,. \tag{6}$$

RI has been used for measuring the segmentation performance on Mesh models [3, 14, 27]. In this work, we adapted it for pointclouds. It measures the likelihood that a pair of points are either in the same part in two segmentations, or in different parts in both segmentations. Considering the ground-truth and segmented parts G and S as before, g_i and s_i indicate the part ids of point i in G and S. We then construct two matrices C and P of co-occurrences of part labels between pairs of points in each segmentation. When a pair of points i and j have the same part id in the ground-truth parts, i.e. $g_i = g_j$, then $C_{ij} = 1$. Likewise, when the points have the same part id in the segmented parts, i.e. $s_i = s_j$, then $P_{ij} = 1$. The RI is then defined as

$$\text{RI} = \binom{2}{N}^{-1} \sum_{i,j,i<j} C_{ij}P_{ij} + (1 - C_{ij})(1 - P_{ij}). \tag{7}$$

Since segmentation dissimilarity is a more common measure than similarity, we report $1 - \text{RI}$. The lower the number, the better the segmentation result.

Our evaluation is three-fold. First, we evaluated our part decomposition method on novel object instances and compared it with the Locally Convex Connected Patches (*LCCP*) method [25] and object patches from Richtsfeld *et al.*'s object segmentation method [19]. Next, in Section 5.2, we report on part decomposition performance on novel object categories. Finally, in Section 5.3, we show qualitative results of applying our method on the cluttered scenes from [16].

5.1. Part Decomposition on Novel Object Instances

For this experiment, we divided the data into training and test sets based on the category split following [16]. We used the data from the first category split for training and the other for testing. Results are shown in Figure 3 where our method is compared with the ground-truth parts. For a better illustration, object parts are colored randomly. Segmented parts which have the maximum overlap with the ground-truth parts, are assigned to the same colors. Otherwise, they are colored with a different random color. We observe that our method decomposes objects into meaningful and nameable parts such as the inside, the outside and the handle of the mug, the curved surface mouth and the handle of the scoop, etc. Due to the disconnectivity between patches, we observe an under-segmentation of object parts e.g. in some cases the scissors.

The decomposition performance of our method is given in Table 2. We also compared it with two other state-of-the-art segmentation methods, the *LCCP* [25] method and Richtsfeld *et al.*'s [19] object segmentation method. *LCCP* segments objects based on local convexity of adjacent supervoxels into parts. Richtsfeld *et al.* [19] provide an object segmentation method from pre-segmented *patches* and geometrical models. The patches are formed considering geometrical information of surfaces and planes. Since our task is object part segmentation, we used their segmented patches for comparison.

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

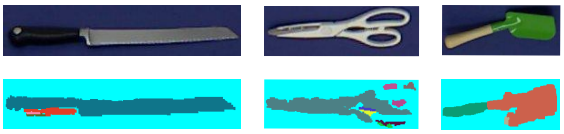


Figure 5. Segmentation error for patches. Top row: RGB image of objects; bottom row: segmented patches based on Region Growing Segmentation. Each patch is shown in a different color. Patch segmentation uses local information based on normals of adjacent points. This results in over-segmentation in areas with too few points.

We can see in Table 2 that our method achieves on average a higher Wov and a lower $1 - \mathrm{RI}$ than the other two methods. This shows the importance of including semantics in addition to geometrical information.

5.2. Part Decomposition on Novel Object Categories

To prove the generalization capabilities of our method, we applied it to novel object categories (categories not seen during training). To this end, we used the novel category split provided by Myers *et al*. [16]. The data are divided into two sets with different categories. We followed the same procedure for training as described in Section 5.1 on the training split. Figure 4 shows how our method segments objects into meaningful parts, producing segmentations consistent among different objects. Since we have a compositional representation of object parts from *patches*, we are able to segment objects into meaningful parts even though these categories have not being seen during training. Object parts are colored for a clear illustration. The segmented parts which have the maximum overlap with the ground-truth parts are colored the same. Otherwise, they are colored with a different random color.

The quantitative results from this experiment are shown in Table 3. We achieved on average better segmentation performance than the other two state-of-the-art methods. However, for some objects in Table 3, we obtained slightly lower performance. Those objects are mostly locally flat in their part connectivities. Since we consider the same flatness threshold for patch segmentation, we obtain over-segmented patches in some cases for those objects as we see in Figure 5. Unfortunately, this error propagates to the part segmentation as well. Making the patch segmentation adaptive is left for future work.

5.3. Qualitative Results on Scenes

To show the applicability of our decomposition method, we went one step further and evaluated it on cluttered scenes. We used the scenes provided in the part affordance dataset [16]. This dataset contains three different scenes as shown in the first column of Figure 6. Each scene is captured in different views. We used the model trained on the objects in Section 5.1 and applied it on the cluttered scenes. We show here the qualitative results of applying our method to each scene at four different views. Since the ground-truth labels in cluttered scenes are not provided, we cannot provide a quantitative analysis. Even so, our results demonstrate that our decomposition does not change much between different views which proves the robustness of our method to viewpoint changes. In addition, we are able to segment a scene into meaningful object parts such as handles, containing parts, blades, etc. Due to our *compositional* part representation from locally flat *patches*, we are able to perform segmentation where object parts are not fully visible. This segmentation with cluttered scenes in this way is very useful for robotic manipulation tasks.

6. Discussion

We have shown that affordances can guide the segmentation of objects into their semantically meaningful and functional parts. Object parts are associated with certain functionalities, which in this work we exploit in order to guide segmentation. Our decomposition relies on shape and geometrical information derived from surface normals and curvatures. Our experimental results show the validity of our approach outperforming the state of the art in the first two tasks and providing a segmentation very close to the ground truth in cluttered scenes. Even so, there is still room for improvement, like the cases of disconnected object areas (e.g. the scissors in Figure 3) or others where we obtain an over-segmentation (Figure 5).

7. Conclusions

We explained here a novel method for 3D object part decomposition using affordances. Our method is compositional starting from locally flat object *patches* to form semantically meaningful object parts. The main contribution of our method is guiding compositional model with affordances. To formulate this, we used a pairwise *MRF*. The results show that our method decomposes objects into semantically meaningful parts. We obtained on average higher overlap with respect to the ground-truth object parts in comparison to other state-of-the-art methods [25, 19]. We showed the value of compositional part representation for segmenting novel object categories (Section 5.2) and cluttered scenes (Section 5.3).

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 610878, 3rd HAND.

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

Method	bowl	cup	hammer	knife	ladle	mallet	mug	pot	saw	scissors	scoop	shears	shovel	spoon	tenderizer	trowel	turner	Average
Weighted Overlap																		
Our Method	95.8	**87.9**	**72.7**	**75.2**	**77.3**	**79.5**	**84.2**	**76.3**	**93.6**	44.4	**81.2**	**64.8**	72.6	**79**	59.9	77.4	82.6	**76.7**
LCCP	**97.9**	81.9	58.3	53.5	72.8	65.1	56.4	58	59.6	46.6	51.7	57.3	48.8	64.5	**67.3**	60	66.7	62.7
Patches in [19]	44.1	61.7	64.7	67.4	69.7	69.7	58.8	50.4	93	**49.6**	52.5	61.4	**73.3**	78.3	61.9	**78.1**	**87.9**	66
1 − Rand Index																		
Our Method	**1.1**	**12.1**	**23.9**	**18.3**	**13.8**	**15.2**	**13.4**	**17.3**	**1**	45.1	**13.1**	22.9	28.8	**14.9**	**21.9**	20.8	10.5	**17.3**
LCCP	3.3	16.5	31.8	35.7	18.9	25.8	22.3	29.8	14.8	**28.9**	15.9	**18.4**	**25.4**	24.1	26.2	23	17.6	22.3
Patches in [19]	68	26.6	28	31.5	24.3	24.7	25.6	31.2	4.4	51.3	45.2	31.4	27.4	18	27.8	**18.4**	**8.3**	29

Table 2. Part decomposition performance based on overlap Wov and 1 − RI scores on novel object instances: our method versus LCCP and segmented patches in [19]. Results are given as percentages.

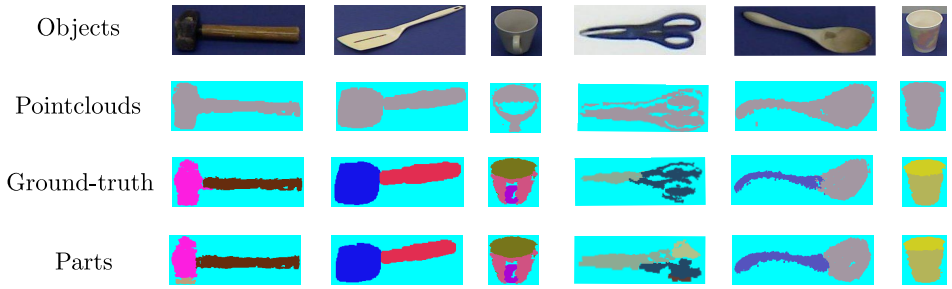


Figure 3. Part decomposition of novel object instances. First row: objects; second row: corresponding pointclouds; third row: ground-truth parts illustrated by different colors; fourth row: segmented objects using our method. Colors are assigned randomly, each color representing one object part. Segmented parts with the maximum overlap score with the ground-truth parts are colored the same; otherwise, they are colored randomly.

Method	cup	ladle	pot	saw	scoop	shears	shovel	tenderizer	trowel	Average
Weighted Overlap										
Our Method	**87.8**	**80**	**76.3**	87.8	**85.1**	55.2	73.1	50.3	**77.9**	**74.8**
LCCP	82.5	71.6	60.9	61.1	51.4	51.4	49.5	**51.7**	60.6	60.1
Patches in [19]	63.1	66.2	50.5	**88.3**	51.5	**60.6**	**73.9**	51.5	76.3	64.6
1 - Rand Index										
Our Method	**11.7**	**12.3**	**16.1**	**5.5**	**8.8**	29.1	28.6	33.1	**18**	**18.1**
LCCP	16.6	19.1	27.3	12.3	16.3	**23.4**	**24.7**	**30.1**	22.5	21.4
Patches in [19]	26.1	26.8	31.3	8.8	45.6	30.7	27.1	37.3	20.4	28.2

Table 3. Part decomposition performance based on overlap Wov and 1 − RI scores on novel object categories: our method versus *LCCP* and segmented patches in [19]. Results are given as percentages.

References

[1] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(1):11–23, 1981. 2

[2] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34, 2012. 5

[3] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. In *ACM Transactions on Graphics (TOG)*, volume 28, page 73. ACM, 2009. 5

[4] Y. Chen, L. L. Zhu, and A. Yuille. Active mask hierarchies for object detection. In *European Conference on Computer Vision*, pages 43–56. Springer, 2010. 2

[5] D. J. Crandall and D. P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *European conference on computer vision*, pages 16–29. Springer, 2006. 2

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, Sept. 2010. 2

[7] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

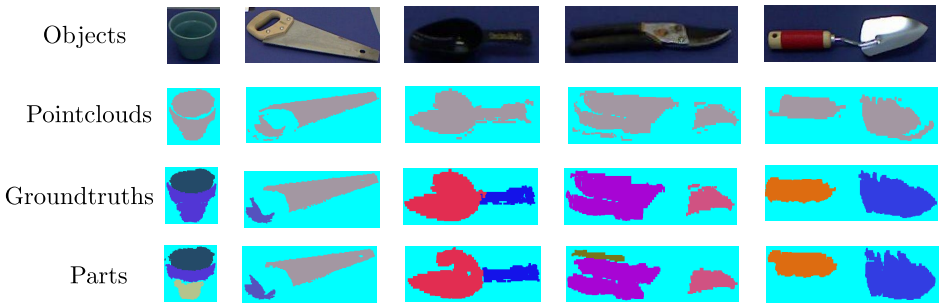


Figure 4. Part decomposition on novel categories. First row: objects; second row: corresponding pointclouds; third row: ground-truth parts illustrated by different colors; fourth row: segmented objects using our method. Colors are assigned randomly, each color representing one object part. Segmented parts with the maximum overlap score with the ground-truth parts are colored the same; otherwise, they are colored randomly.

Figure 6. Detection in cluttered scenes. First column: objects; second through fifth columns: decomposition results of our method in different views. Colors are assigned randomly, each color representing one object part.

IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. 2

[8] J. Gibson. The theory of affordances. In *Perceiving, Acting, and Knowing: Toward and Ecological Psychology*, pages 62–82. Erlbaum, 1977. 1

[9] R. B. Girshick, P. F. Felzenszwalb, and D. A. Mcallester. Object detection with grammar models. In *Advances in Neural Information Processing Systems*, pages 442–450, 2011. 2

[10] C. Gu, P. Arbeláez, Y. Lin, K. Yu, and J. Malik. Multi-component models for object detection. In *European Conference on Computer Vision*, pages 445–458. Springer, 2012. 2

[11] K. Hausman, C. Bersch, D. Pangercic, S. Osentoski, Z.-C. Marton, and M. Beetz. Segmentation of cluttered scenes through interactive perception. In *International Conference on Robotics and Automation: Workshop on Semantic Perception and Mapping for Knowledge-enabled Service Robotics*, May 14–18 2012. 2

[12] T. Hermans, J. M. Rehg, and A. Bobick. Affordance prediction via learned object attributes. In *International Conference on Robotics and Automation: Workshop on Semantic Perception, Mapping, and Exploration*, 2011. 2

[13] T. Hermans, J. M. Rehg, and A. F. Bobick. Guided pushing for object singulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4783–4790. IEEE, 2012. 2

[14] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *ACM Transactions on Graphics (TOG)*, 29:102, 2010. 2, 5

[15] D. Katz, A. Venkatraman , M. Kazemi, J. A. D. Bagnell, and A. T. Stentz . Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. In *Robotics: Science and Systems Conference (RSS)*, June 2013. 2

[16] 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. 2, 5, 6

[17] D. A. Norman. Affordance, conventions, and design. *interactions*, 6(3):38–43, 1999. 1

[18] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):248–253, 2006. 2

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

[19] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze. Segmentation of unknown objects in indoor environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4791–4796. IEEE, 2012. 2, 5, 6, 7

[20] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. Object recognition by functional parts. In *Image Understanding Workshop, pp. II*, pages 1531–1539, 1994. 2

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

[22] R. B. Rusu, A. Holzbach, G. Bradski, and M. Beetz. Detecting and segmenting objects for mobile manipulation. In *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV)*, September 2009. 2

[23] M. Schmidt. UGM: A matlab toolbox for probabilistic undirected graphical models. `http://www.cs.ubc.ca/~schmidtm/Software/UGM.html`, 2007. 4, 5

[24] L. Stark and K. W. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(10):1097–1104, 1991. 2

[25] C. S. Stein, M. Schoeler, J. Papon, and F. Wörgötter. Object partitioning using local convexity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2, 5, 6

[26] H. van Hoof, O. Kroemer, and J. Peters. Probabilistic Segmentation and Targeted Exploration of Objects in Cluttered Environments. *IEEE Transactions on Robotics*, 30(5):1198–1209, 2014. 2

[27] Y. Zheng, C.-L. Tai, E. Zhang, and P. Xu. Pairwise harmonics for shape analysis. *IEEE transactions on visualization and computer graphics*, 19(7):1172–1184, 2013. 2, 5

[28] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1062–1069. IEEE, 2010. 2

This article is part of Chapter 4: Affordances for Parts. It appeared in: 2017 IEEE Winter Conference on Applications of Computer Vision

Chapter 5

Parts for Affordances

In Chapter 4, we presented a part segementation approach using affordances. In this chapter, we focus on detecting the affordances using object parts. Even though there have been many works for affordance detection using objects (Aldoma et al., 2012; Katz et al., 2014) or parts driven only from visual properties (Laga et al., 2013; Fu et al., 2008; Varadarajan and Vincze, 2011), our goal here is to apply detection on functional object parts. Using functional object parts not only ensures that parts can be directly used for affordance detection but it also ensures generalizability of affordances among novel objects.

As mentioned in Chapter 4, parts are represented based on patches. We experimentally discovered that such a representation is suitable for part segmentation but it does not give us a high affordance detection performance. Therefore we extracted features from parts for learning affordances. Many works in affordance detection use hand-designed geometrical features (Myers et al., 2015; Laga et al., 2013; Fu et al., 2008; Varadarajan and Vincze, 2011; Rivlin et al., 1995). Even though affordances are associated with shape and geometry of objects, relevant affordance features should be obtained in an unsupervised manner to be more generalizable. There exists a number of unsupervised approaches for feature discovery such as Principal component analysis (PCA), Independent component analysis (ICA), sparse coding, neural networks, and autoencoders (Murphy, 2012). In our work, we employ a multilayer non-linear feature learning approach using autoencoders. In our work, an autoencoder is trained on surface normal images of parts. The parts are then represented based on their corresponding codes obtained from the autoencoder. We use these codes for learning affordances from parts.

The typical scenarios considered in this thesis are kitchen scenarios and tool use. For example, cleaning a cluttered table by grasping objects, scooping coffee beans, and pouring coffee beans. In these scenarios detecting affordances of object parts is important for a robot. Let us consider the pouring coffee beans scenario. In this task, beans can be poured only in container object parts which afford fillability. Thus detecting fillability of parts is important. Input data is obtained from a kinect sensor which might be also mounted on a robot. The input pointcloud might contain multiple objects. Our objective in these scenarios is to detect affordances of object parts in the presence of partial occlusion or clutter.

The paper included in the following pages describes our work in part-based affordance detection and has been conditionally accepted for *Autonomous Robots* (Rezapour Lakani et al., 2017a). Our part segmentation approach is trained as described in Chapter 4. Parts' features are subsequently learned using an autoencoder. Since parts can have multiple affordances, we trained discriminative binary affordance classifiers. Our work is evaluated on an RGB-D part-affordance dataset (Myers et al., 2015) on single objects as well as on cluttered scenes. We obtained higher affordance detection performance compared to other state-of-the-art approaches on this dataset. We also evaluated our approach on real robotic grasping scenarios. In this ex-

periment, the robot is asked to grasp objects with a particular affordance from a table. Thus a successful grasp depends on a correct affordance detection.

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 article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

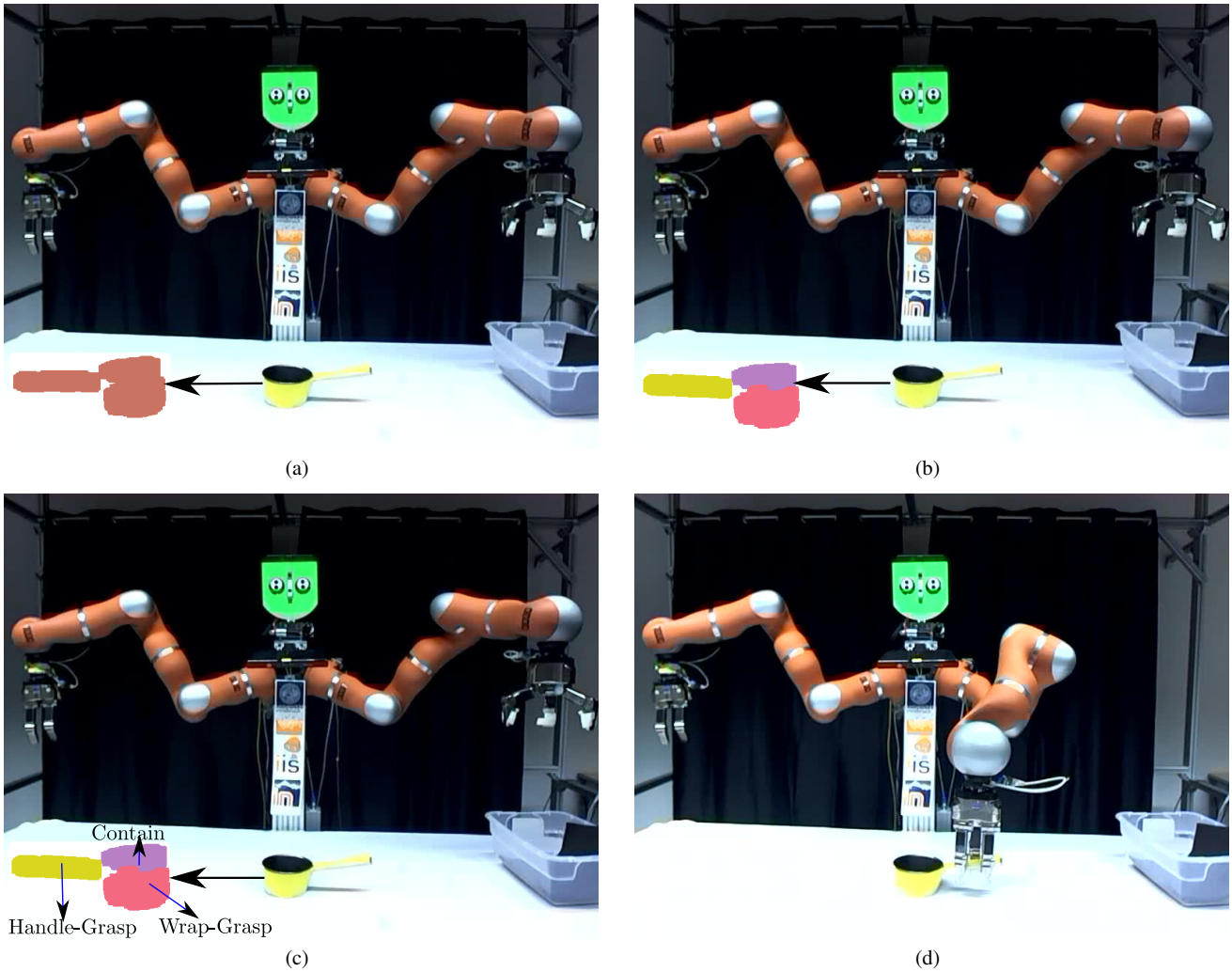


Fig. 1 The robot is asked to grasp the object on the table using the handle-grasp affordance. It gets an input from the kinect on its chest (Fig. 1(a)), segments the object into parts (Fig. 1(b)) and detects its affordances (Fig. 1(c)). It then uses the handle-grasp affordance for grasping the object (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.

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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/

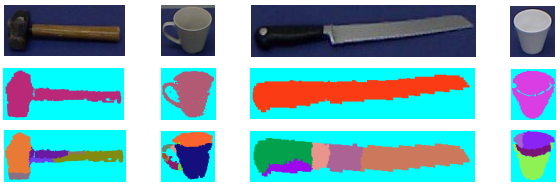


Fig. 2 Patch segmentation of pointclouds. Top row: RGB images; middle row: pointclouds; bottom row: segmented patches. Colors indicate distinct patches.

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

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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

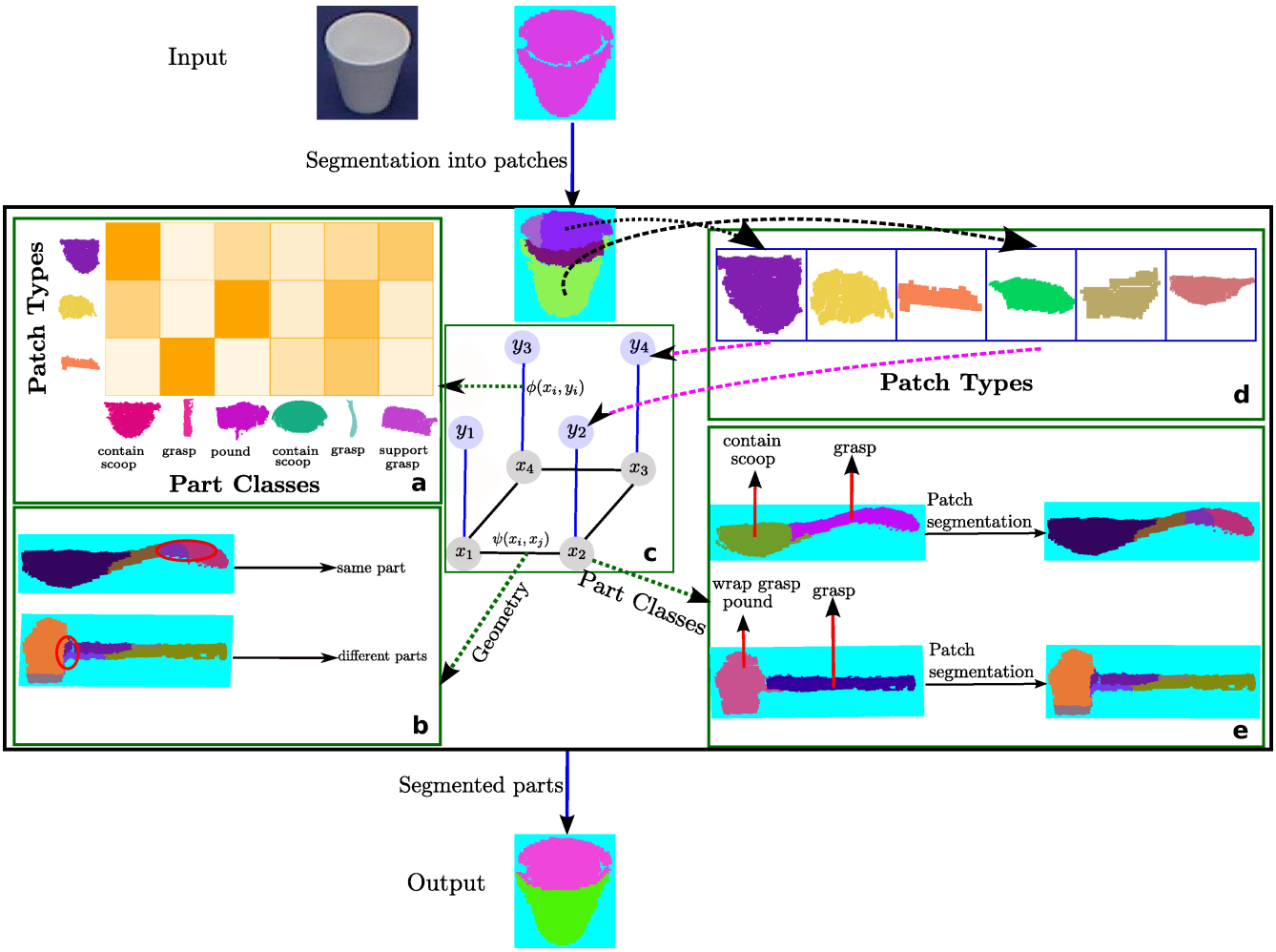


Fig. 3 Object part segmentation based on affordances. Object parts in our model are driven based on their affordances such as pounding, grasping and containing. We learn a graphical model for part segmentation from locally-flat object patches based on two sources of information: 1) the potential of a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

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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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.

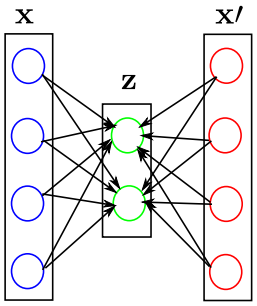


Fig. 4 A schematic diagram of an autoencoder with one hidden layer. It has an input layer $\mathbf{x}$ and an output layer $\mathbf{x}'$ and one hidden layer $\mathbf{z}$. The network attempts to reconstruct the input data. The number of neurons in the input and output layers are the same. The hidden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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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.

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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.

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.

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

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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

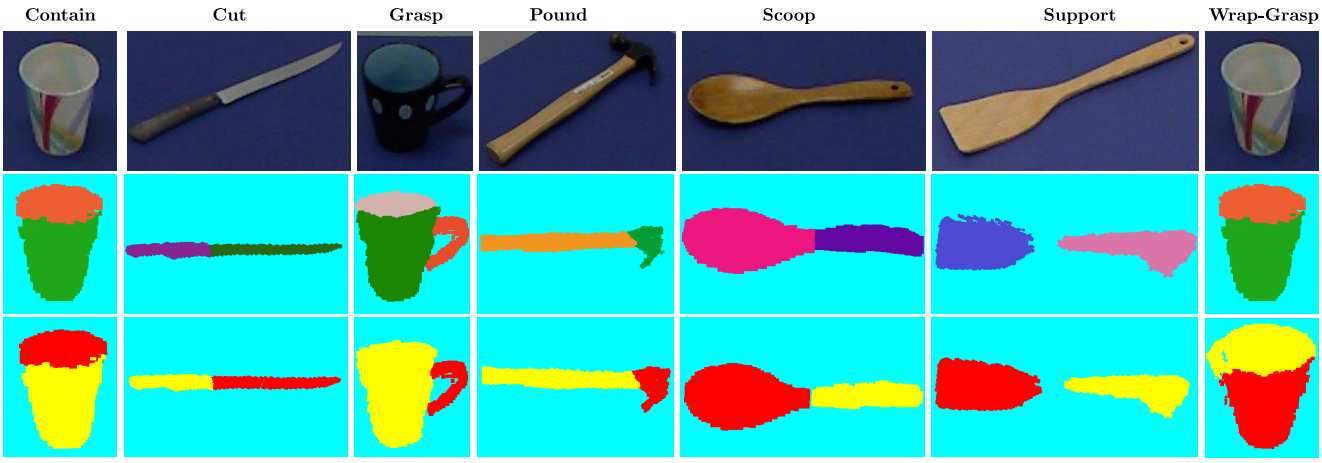


Fig. 5 Qualitative results on novel object instances of the RGB-D part affordance dataset. Labels: affordances to be detected for the objects in the first row. First row: RGB images of the objects. Second row: segmented object parts. Third row: results of the affordance detection on the object parts.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

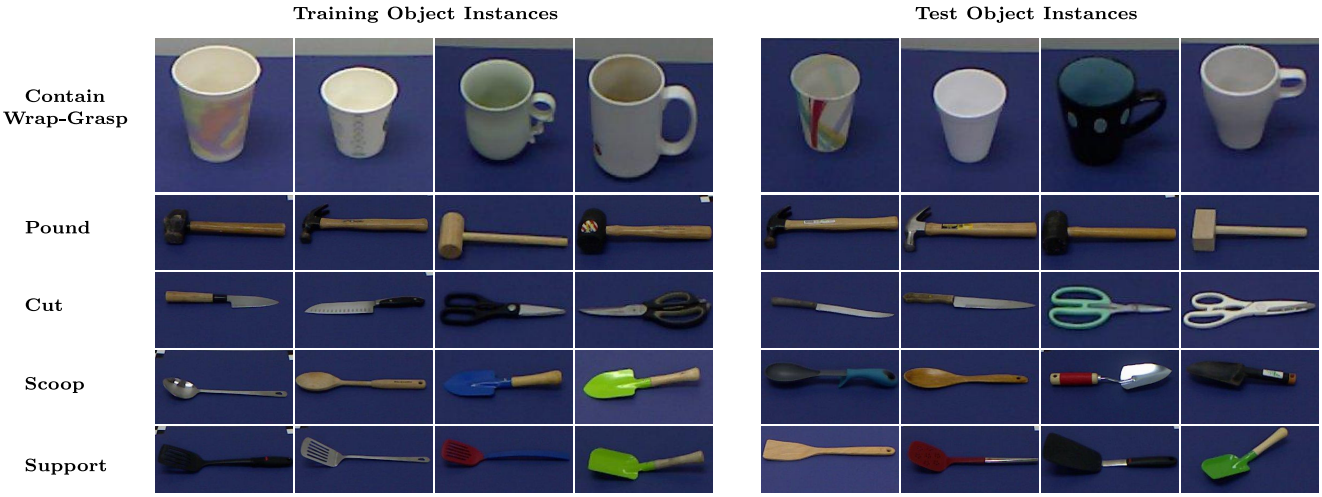


Fig. 6 Examples of training and test objects used for novel object instances in Fig 5. For simplicity only two instances of each category are shown.

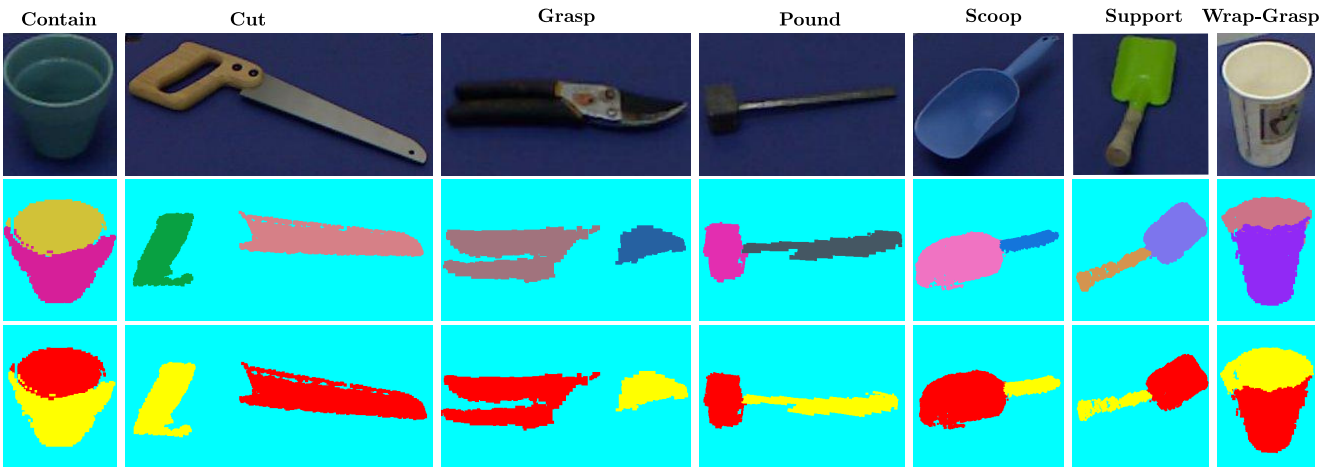


Fig. 7 Qualitative results on novel object categories of the RGB-D part affordance dataset. Labels: affordances to be detected for the objects in the first row. First row: RGB images of the objects. Second row: segmented object parts. Third row: results of the affordance detection on the object parts.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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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.

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.

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 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.

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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.

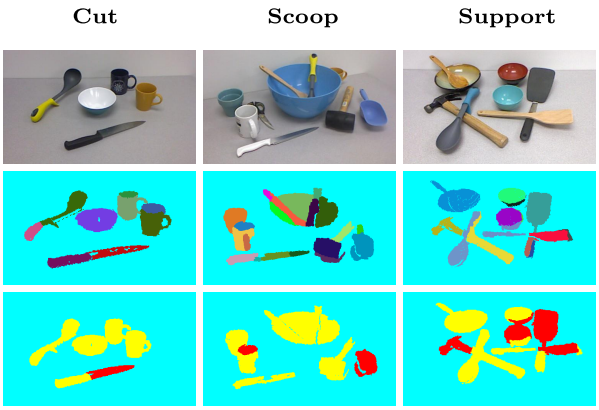


Fig. 9 Qualitative results on cluttered scenes of the RGB-D part affordance dataset. Labels: affordances to be detected for the objects in the first row. First row: RGB images of the objects. Second row: segmented object parts. Third row: results of the affordance detection on the object parts. The red-highlighted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

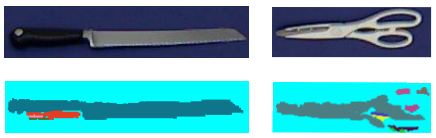


Fig. 10 Segmentation error for patches. Top row: RGB image of objects; bottom row: segmented patches based on Region Growing Segmentation. Each patch is shown in a different color. Patch segmentation uses surface normals of adjacent points. This results in a false segmentation in disconnected areas or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

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 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.

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

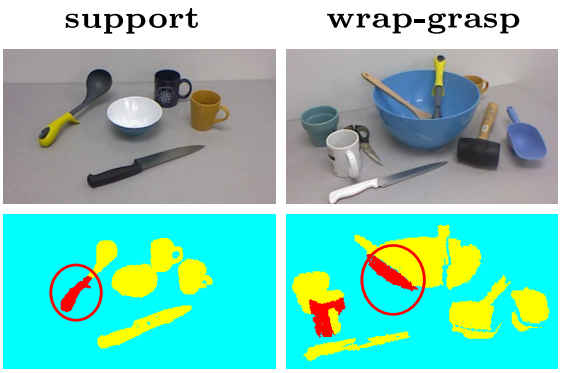


Fig. 11 False positives for support and wrap-grasp affordances. Affordances are detected on single parts. This may result in false positives especially in occluded scenes as indicated by the red circles. Using affordances of neighboring parts can reduce the false positive rate.

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.

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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

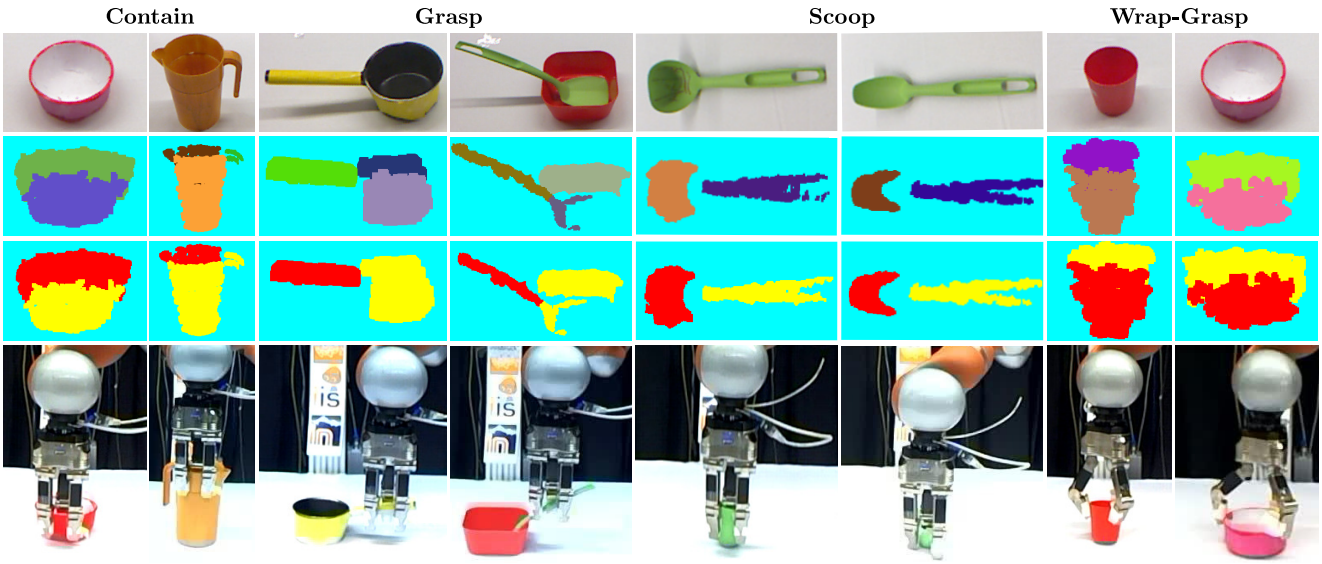


Fig. 13 Robotic grasping experiment on single objects. The robot is asked to grasp objects based on the given affordances shown above the objects. First row: RGB images of the objects, Second row: segmented object parts, Third row: results of the affordance detection on the object parts. Parts that afford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

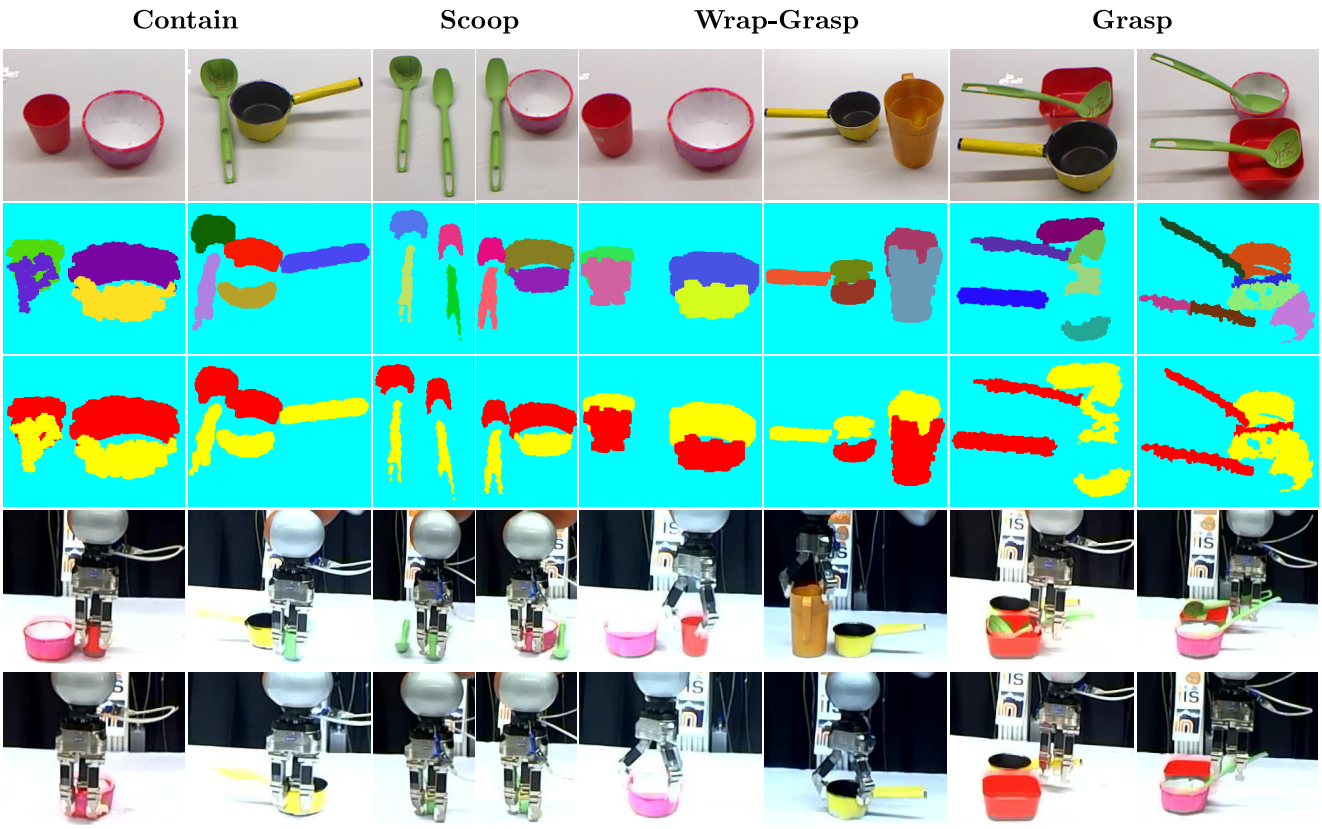


Fig. 15 Robotic grasping experiment on scenes. The robot is asked to grasp objects based on the given affordances on top of the scenes. First row: RGB images of the scenes, Second row: segmented object parts, Third row: results of the affordance detection on the object parts. Parts that afford the given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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

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)

This article is part of Chapter 5: Parts for Affordances. It is conditionally accepted for: Autonomous Robots

Chapter 6

Affordances for Tasks

In Chapter 5, we described our part-based affordance detection framework. In this chapter, we focus on exercising the detected affordances by performing robotic tasks. There are many approaches in robotics for performing tasks based on affordances using visual information (Hjelm et al., 2014; Song et al., 2015, 2010; Aleotti and Caselli, 2011; Abelha Ferreira and Guerin, 2017; Tenorth et al., 2013). These methods have shown a high task detection and execution performance mostly in simulated environments. Furthermore, they are often applied on CAD models of objects. Using CAD models and being applied on simulated environments limits the applicability of these approaches in real robotic scenarios. In real applications, we do not necessarily have precise 3D models of objects. In our work, we use view-based pointclouds obtained from a Kinect sensor. Thus our method is robust to partial views of objects in real tasks. In addition to the input data, most of the state-of-the-art methods have either linked tasks to the entire object (Song et al., 2010, 2015) or to geometrical regions driven from CAD object models (Aleotti and Caselli, 2011; Tenorth et al., 2013). Linking tasks to objects restricts the generalizability of performing tasks to novel objects. Moreover, associating tasks to single object parts is not enough for performing many robotic tasks. For example, in order to pound a nail with a hammer, the hammer should be grasped from its handle to strike the nail from its head. Therefore, not only the head of the hammer (pounding part) but also its handle (graspable part) is important to pound the nail successfully. In our work, we use relations among parts for performing tasks.

The main idea in this work is to link affordances and their parts to different robotic tasks. We considered tasks in indoor scenarios such as cutting a cake, placing sponge on an object, and cleaning a table by removing objects. The robot is asked to perform a task given the visual percepts of its environment. The visual data is obtained from a Kinect sensor mounted on a robot. The input pointcloud is segmented into parts and affordances are detected subsequently. During training, the robot learns relevant affordances for executing tasks. These affordances are linked to the segmented object parts. Thus the robot learns how to manipulate parts for performing tasks.

Rezapour Lakani et al. (2017a) presents the association of affordances with grasping. In this work, we used four different affordances namely: containing, scooping, handle-grasping, and wrap-grasping. The affordances are then associated with different grasp types. The wrap-grasp affordance is associated with the spherical grasp and the other affordances with the rim grasp. The robot is asked to detect the affordances and grasp the parts associated with them using the corresponding grasp type.

In the following we present an extension of this work with six different tasks (Rezapour Lakani et al., 2018). These tasks are: striking a ball, scooping coffee beans, pouring coffee, placing a sponge on an object, cutting a cake, and removing objects from a table. In order to perform

these tasks, relevant affordances should be detected. For example, for cutting a cake with a knife, cutting and grasping affordances are relevant. The part associated with the grasping affordance is used for manipulation and the part linked to the cutting affordance for execution of the task. We computed frequency of co-occurrence of affordances to perform the tasks. We then used these frequencies to compute the manipulative affordance for a particular task. Since affordances are connected to the parts, manipulative parts can be detected subsequently. Our experiments showed that using a part-based approach for performing tasks results in high task detection and execution performance. The paper included in the following pages addresses our work in performing tasks based on affordances and has been accepted for publication for the *IEEE Robotics and Automation Letters* and the 2018 *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Exercising Affordances of Objects: A Part-Based Approach

Safoura Rezapour Lakani Antonio J. Rodríguez-Sánchez Justus Piater

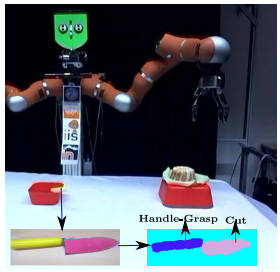
***Abstract*— This study shows how learning relations between affordances facilitates performing robotic tasks. Tasks usually involve multiple affordances. For example, for pounding a nail with a hammer, grasp-ability and pound-ability of the hammer are important for performing the pounding task successfully. 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.**

I. INTRODUCTION

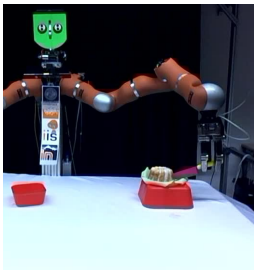
Learning affordances of objects and the tasks that can be performed given them are important capabilities of robots. Let us consider the robot in Figure 1. For cutting the cake in Fig. 1(b), the robot should detect an object which affords *cutting* like the knife in Fig. 1(a). It should also reason that 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.

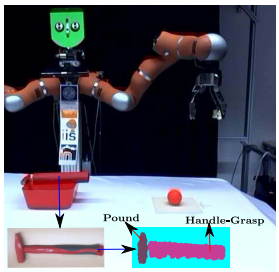
Learning relations between affordances plays an important role for performing tasks. Some affordances are related to manipulation of objects to perform a task and others are related to execution of the tasks. For example, in the task of cutting the cake, the *handle-grasp* affordance is important for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



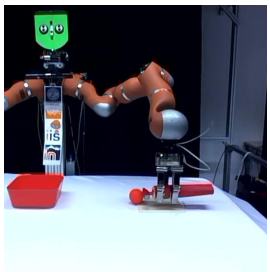
(a) For cutting the cake, an object should afford *cutting* and be graspable from its handle.



(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.

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 of part-based affordance detection on the basis of shape from RGB-D data was introduced in [19]. 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-

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

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], [8], [6], [7]. Most studies have been done on grasping affordance. In the work discussed in [5], [8], 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 [9], [10], [11]. In the work discussed in [11], 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 [12], a CNN-based approach is used to segment and classify objects. Objects have also been segmented into parts based on local convexity [13]. In the work discussed in [13], 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 [14] uses Reeb graph [15], [16] 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.

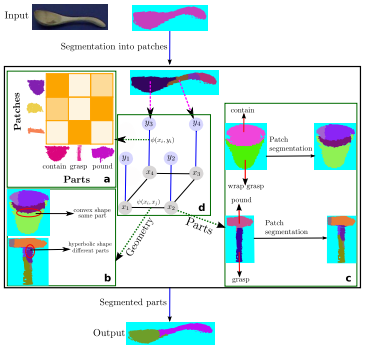


Fig. 2. Object part segmentation based on affordances [17]. Object parts have functional meaning such as pounding, grasping and containing. A graphical model for part segmentation from locally-flat object patches are learned based on two sources of information: 1) the potential of a patch y_i to belong 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.

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 [17]. 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 ϕ and a sum of pairwise potentials ψ. The

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

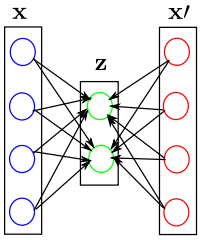


Fig. 3. A schematic diagram of an autoencoder with one hidden layer. It has an input layer $\mathbf{x}$ and an output layer $\mathbf{x}'$ and one hidden layer $\mathbf{z}$. The network attempts to reconstruct the input data. The number of neurons in the input and output layers are the same. The hidden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

unary potential ϕ determines the probability of a patch belonging to a part (Fig. 2c). The pairwise potential ψ indicates the probability of two adjacent patches belonging to the same part (Fig. 2d). The parameters of the potential functions θ and Θ 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 [17].

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

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

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

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 [19].

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.

The manipulative affordance is computed as

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

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 [19] 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 [19] 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 [19].

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 [19] 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.

B. Task Detection Performance

In this section, we report on the evaluation results for the perception part of our system. As mentioned earlier, affor-

[1] http://pointclouds.org/

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

(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.

dance classifiers are trained on the RGB-D part affordance dataset [19]. 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 [19] 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 [19] and SRF [19] 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 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.

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

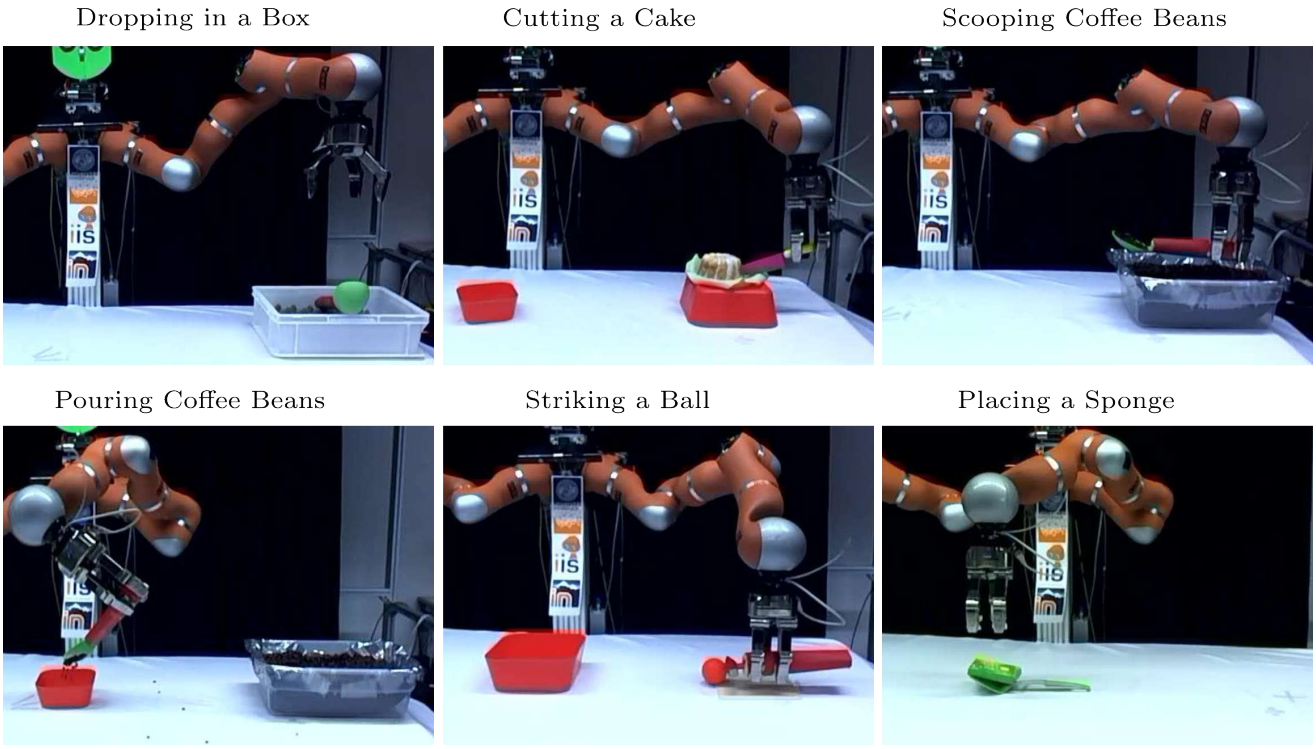


Fig. 6. Tasks performed based on affordances. Based on the definition of the affordances, each affordance is mapped to a particular task. To be used for manipulations, some thin object handles are covered.

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 [19]	0.15	0.04	0.05	0.17	0.04	0.03	0.10	0.08
SRF [19]	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 [19]	0.16	0.02	0.15	0.18	0.02	0.05	0.10	0.10
SRF [19]	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 [19]: RANK WEIGHTED F-MEASURES [22].

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.

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

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.

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.

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 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.

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.

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

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

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.

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].

(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 [19] 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," *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, pp. 67–82, 1977.

[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.

[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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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, p. 4, 2017.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

[19] 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.

[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.

[22] S. Rezapour Lakani, A. Rodríguez-Sánchez, and J. Piater, "Affordances for Parts, Parts for Affordances," 2017.

[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.

This article is part of Chapter 6: Affordances for Tasks. It is accepted for publication for: IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems

Chapter 7

Conclusions

In this thesis, we emphasized in relationships between objects, affordances, and tasks. Using these relationships, we proposed novel methods for object representation, affordance detection, and performing tasks. In this chapter, we review the contributions and discuss open questions which are related to the results that have been presented in the previous chapters.

7.1 Summery of Contributions

In this thesis, we contributed a number of novel methods and approaches to the field of robot vision. These contributions are summarized below:

- A novel RGB-D part segmentation approach guided by affordances of objects (Chapter 4).
- A novel affordance detection approach based on functional object parts (Chapter 5).
- An approach for performing tasks based on object parts and the relationships between them (Chapter 6).
- Extensive evaluations on real robotic scenarios for affordance detection and performing tasks (Chapter 5 and Chapter 6).

We proposed a novel approach for using affordances implicitly in object part segmentation. This affordance-driven part decomposition resulted in functional parts (Chapter 4). Our part-based approach showed a remarkable performance for affordance detection on a benchmark dataset compared to the other state-of-the-art approaches (Chapter 5). We performed quantitative analysis on novel object categories and instances for part segmentation and affordance detection. Using this part-based approach, we obtained better generalization for predicting affordances of novel object instances and categories compared to the other baseline approaches. Furthermore, we demonstrated the merit of our part-based method in learning associations between affordances and tasks that can be performed after successfully detecting them (Chapter 5 and Chapter 6). This way of linking visual and non-visual cues (i.e. affordances) is the hallmark of this thesis and to the best of our knowledge has not been studied extensively in robot vision.

7.2 Perspectives

In this thesis, we focused on overlapping area of visual object representation in computer vision and affordance detection in robotics. We focused on possibilities of linking object representation, affordances, and tasks together. We showed that making these connections has great impact on

affordance detection. Since affordances provide action possibilities to a robot, linking affordances to tasks is also important for executing tasks successfully. Our approach to these problems also opens up new research directions worth being investigated. In this section, we discuss these open research avenues.

In this work, we focused on one relationship at a time, i.e. affordances for object representation, or parts for detecting affordances. We assumed a fixed number of affordances and tasks. In the same line of work, a robot might deal with new affordances and tasks. Thus, making a recurrent connection between affordances, object representation, and tasks is important for life-long learning of robots.

Learning new tasks and affordances may also result in multiple object representations. For example, a bottle might be represented as two parts (inside and outside) for a pouring task whereas if the task is rolling, the entire bottle is considered as one part. These multiple object representations might also are related to each other. Learning relations between various object representations and linking them together have great impacts on affordance detection and the tasks performed by detecting the affordances.

In affordance learning, not only visual appearance of objects but also physical properties of objects play an important role. For example, for pounding a nail with a hammer, not only the shape of the hammer but also its weight is an important factor for reasoning about pounding affordance. Integrating physical properties such as force, weight, and friction enriches object representations. Such enhanced representations in turn improve affordance detection.

The discussed research problems go beyond our thesis, but investigating them may move the current research in affordance learning and robot vision a few steps forward.

Bibliography

Paulo Abelha Ferreira and Frank 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.

Aitor Aldoma, Federico Tombari, and Markus Vincze. 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)*, pages 1732–1739. IEEE, 2012.

Jacopo Aleotti and Stefano Caselli. Part-based robot grasp planning from human demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4554–4560. IEEE, 2011.

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for RGB-D based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013.

Chaitanya Desai and Deva Ramanan. Predicting functional regions on objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.

Sven J Dickinson. Object representation and recognition. *What is cognitive science*, 7:172–207, 1999.

Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

Hongbo Fu, Daniel Cohen-Or, Gideon Dror, and Alla Sheffer. Upright orientation of man-made objects. In *ACM transactions on graphics (TOG)*, volume 27, page 42. ACM, 2008.

James Jerome Gibson. The Theory of Affordances. *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, pages 67–82, 1977.

James Jerome Gibson. *The Ecological Approach to Visual Perception*. Psychology Press, 1979.

Stephen Hart, Paul Dinh, and Kim Hambuchen. Affordance templates for shared robot control. In *Artificial Intelligence and Human-Robot Interaction, AAAI Fall Symposium Series, Arlington, VA. USA*, 2014.

Stephen Hart, Paul Dinh, and Kimberly Hambuchen. The affordance template ros package for robot task programming. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6227–6234. IEEE, 2015.

Tucker Hermans, James M Rehg, and Aaron Bobick. Affordance prediction via learned object attributes. In *IEEE International Conference on Robotics and Automation (ICRA): Workshop on Semantic Perception, Mapping, and Exploration*, pages 181–184, 2011.

Martin Hjelm, Renaud Detry, Carl Henrik Ek, and Danica Kragic. Representations for cross-task, cross-object grasp transfer. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5699–5704. IEEE, 2014.

Yun Jiang, Changxi Zheng, Marcus Lim, and Ashutosh Saxena. Learning to place new objects. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3088–3095. IEEE, 2012.

Michael J Jones and James M Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3d mesh segmentation and labeling. In *ACM Transactions on Graphics (TOG)*, volume 29, page 102. ACM, 2010.

Dov Katz, Arun Venkatraman, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Autonomous Robots*, 37(4):369–382, 2014.

Vladislav Kreavoy, Dan Julius, and Alla Sheffer. Model composition from interchangeable components. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, pages 129–138. IEEE, 2007.

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

Manuel Lopes, Francisco S Melo, and Luis Montesano. Affordance-based imitation learning in robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1015–1021. IEEE, 2007.

Huaqing Min, ChangâĂŹan Yi, Ronghua Luo, Jinhui Zhu, and Sheng Bi. Affordance research in developmental robotics: a survey. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):237–255, 2016.

Bogdan Moldovan, Plinio Moreno, Martijn van Otterlo, José Santos-Victor, and Luc De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4373–4378. IEEE, 2012.

Luis Montesano, Manuel Lopes, Alexandre Bernardino, and Jose Santos-Victor. Modeling affordances using bayesian networks. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4102–4107. IEEE, 2007.

Luis Montesano, Manuel Lopes, Alexandre Bernardino, and José Santos-Victor. Learning object affordances: from sensory–motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26, 2008.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Austin Myers, Ching L. Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *International Conference on Robotics and Automation (ICRA)*, 2015.

Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Detecting object affordances with convolutional neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2765–2770. IEEE, 2016.

Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. IROS, 2017.

Donald A Norman. *The psychology of everyday things*. Basic books, 1988.

Bjorn Ommer and Joachim Buhmann. Learning the compositional nature of visual object categories for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (3):501–516, 2010.

Damir Omrčen, Christian Böge, Tamim Asfour, Aleš Ude, and Rüdiger Dillmann. Autonomous acquisition of pushing actions to support object grasping with a humanoid robot. In *9th IEEE-RAS International Conference on Humanoid Robots*, pages 277–283. IEEE, 2009.

Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3d geometry to deformable part models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3362–3369. IEEE, 2012.

Massimiliano Pontil and Alessandro Verri. Support vector machines for 3d object recognition. *IEEE transactions on pattern analysis and machine intelligence*, 20(6):637–646, 1998.

Safoura Rezapour Lakani, Mirela Popa, Antonio Rodríguez-Sánchez, and Justus Piater. CPS: 3D Compositional Part Segmentation through Grasping. In *12th Conference on Computer and Robot Vision*, pages 117–124. IEEE, 6 2015.

Safoura Rezapour Lakani, Antonio Rodríguez-Sánchez, and Justus Piater. Towards affordance detection for robot manipulation using affordance for parts and parts for affordance. In *Autonomous Robots*, 2017a. Submitted.

Safoura Rezapour Lakani, Antonio Rodríguez-Sánchez, and Justus Piater. Can Affordances Guide Object Decomposition Into Semantically Meaningful Parts? In *IEEE Winter Conference on Applications of Computer Vision*, 2017b.

Safoura Rezapour Lakani, Antonio Rodríguez-Sánchez, and Justus Piater. Exercising Affordances of Objects: A Part-Based Approach. In *IEEE Robotics and Automation Letters and IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018. Accepted.

Andreas Richtsfeld, Thomas Mörwald, Johann Prankl, Michael Zillich, and Markus Vincze. Segmentation of unknown objects in indoor environments. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4791–4796. IEEE, 2012.

Ehud Rivlin, Sven J Dickinson, and Azriel Rosenfeld. Recognition by functional parts. *Computer Vision and Image Understanding*, 62(2):164–176, 1995.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

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

Johann Sawatzky, Abhilash Srikantha, and Juergen Gall. Weakly supervised affordance detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Fabien Scalzo and Justus H Piater. Adaptive patch features for object class recognition with learned hierarchical models. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Bernt Schiele and James L Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

Paul Schnitzspan, Mario Fritz, Stefan Roth, and Bernt Schiele. Discriminative structure learning of hierarchical representations for object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2238–2245. IEEE, 2009.

Zhangzhang Si and Song-Chun Zhu. Learning hybrid image templates (hit) by information projection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1354–1367, 2012.

Zhangzhang Si and Song-Chun Zhu. Learning and-or templates for object recognition and detection. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2189–2205, 2013.

Dan Song, Kai Huebner, Ville Kyrki, and Danica Kragic. Learning task constraints for robot grasping using graphical models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1579–1585. IEEE, 2010.

Dan Song, Carl Henrik Ek, Kai Huebner, and Danica Kragic. Task-based robot grasp planning using probabilistic inference. *IEEE transactions on robotics*, 31(3):546–561, 2015.

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

Michael Stark, Philipp Lies, Michael Zillich, Jeremy Wyatt, and Bernt Schiele. Functional object class detection based on learned affordance cues. *Computer Vision Systems*, pages 435–444, 2008.

Christoph Simon Stein, Markus Schoeler, Jeremie Papon, and Florentin Wörgötter. Object partitioning using local convexity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014a.

Simon Christoph Stein, Florentin Wörgötter, Markus Schoeler, Jeremie Papon, and Tomas Kulvicius. Convexity based object partitioning for robot applications. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3213–3220. IEEE, 2014b.

Erik B Sudderth, Antonio Torralba, William T Freeman, and Alan S Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77(1): 291–330, 2008.

Jaeyong Sung, Seok Hyun Jin, and Ashutosh Saxena. Robobarista: Object part based transfer of manipulation trajectories from crowd-sourcing in 3D pointclouds. In *Robotics Research*, pages 701–720. Springer, 2018.

Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

Moritz Tenorth, Stefan Profanter, Ferenc Balint-Benczedi, and Michael 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*, pages 5943–5949. IEEE, 2013.

Sinisa Todorovic and Narendra Ahuja. Unsupervised category modeling, recognition, and segmentation in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 (12):2158–2174, 2008.

Karthik Mahesh Varadarajan and Markus Vincze. Affordance based part recognition for grasping and manipulation. In *Workshop on Autonomous Grasping, ICRA*, 2011.

Jianyu Wang and Alan L Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1788–1797, 2015.

Ying Nian Wu, Zhangzhang Si, Haifeng Gong, and Song-Chun Zhu. Learning active basis model for object detection and recognition. *International journal of computer vision*, 90(2):198–235, 2010.

Natsuki Yamanobe, Weiwei Wan, Ixchel G Ramirez-Alpizar, Damien Petit, Tokuo Tsuji, Shuichi Akizuki, Manabu Hashimoto, Kazuyuki Nagata, and Kensuke Harada. A brief review of affordance in robotic manipulation research. *Advanced Robotics*, 31(19-20):1086–1101, 2017.

Philipp Zech, Simon Haller, Safoura Rezapour Lakani, Barry Ridge, Emre Ugur, and Justus Piater. Computational models of affordance in robotics: a taxonomy and systematic classification. *Adaptive Behavior*, 25(5):235–271, 10 2017.

Youyi Zheng, Chiew-Lan Tai, Eugene Zhang, and Pengfei Xu. Pairwise harmonics for shape analysis. *IEEE transactions on visualization and computer graphics*, 19(7):1172–1184, 2013.

Long Zhu, Yuanhao Chen, and Alan Yuille. Unsupervised learning of probabilistic grammar-markov models for object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):114–128, 2009.