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.

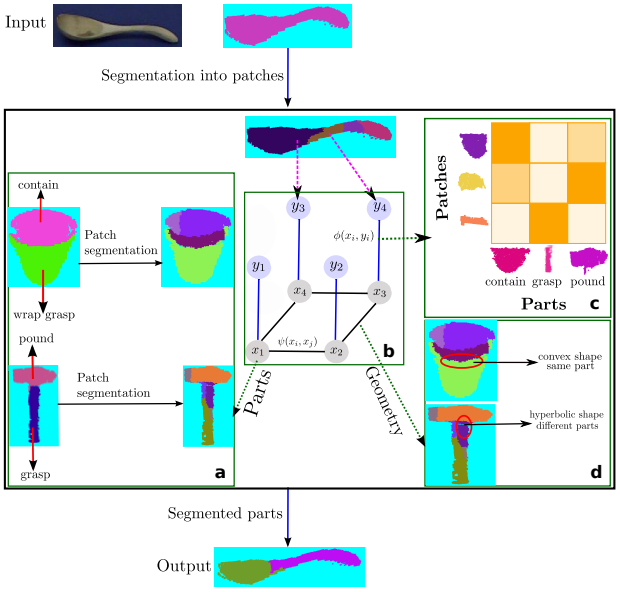

Figure 1. Object part segmentation based on affordances. Object parts in our model have semantic meaning based on their affordances such as pounding, grasping and containing. We learn a graphical model for part segmentation from locally-flat object patches based on two sources of information: 1) the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

But what is an object part? Object segmentation into

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-

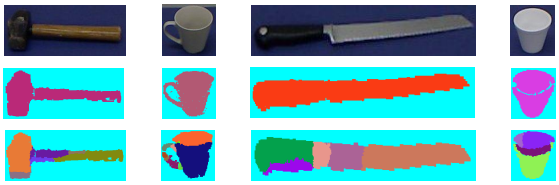

Figure 2. Patch segmentation from pointcloud. Top row: RGB image, middle row: pointclouds and bottom row: segmented patches; colors indicate different patches.

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/

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].

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.

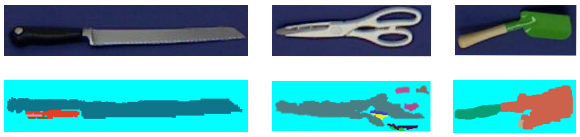

Figure 5. Segmentation error for patches. Top row: RGB image of objects; bottom row: segmented patches based on Region Growing Segmentation. Each patch is shown in a different color. Patch segmentation uses local information based on normals of adjacent points. This results in over-segmentation in areas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

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.

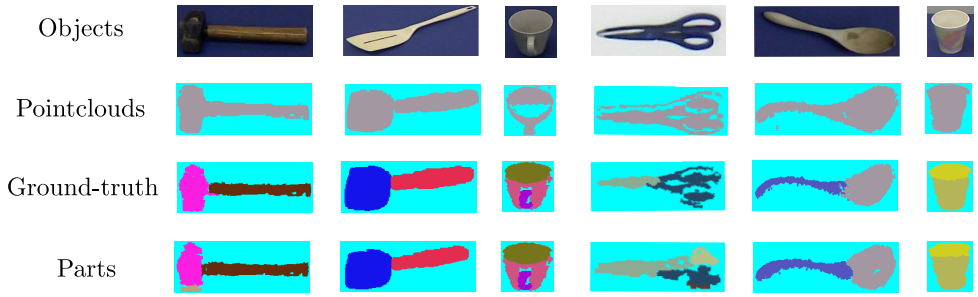

Figure 3. Part decomposition of novel object instances. First row: objects; second row: corresponding pointclouds; third row: ground-truth parts illustrated by different colors; fourth row: segmented objects using our method. Colors are assigned randomly, each color representing one object part. Segmented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

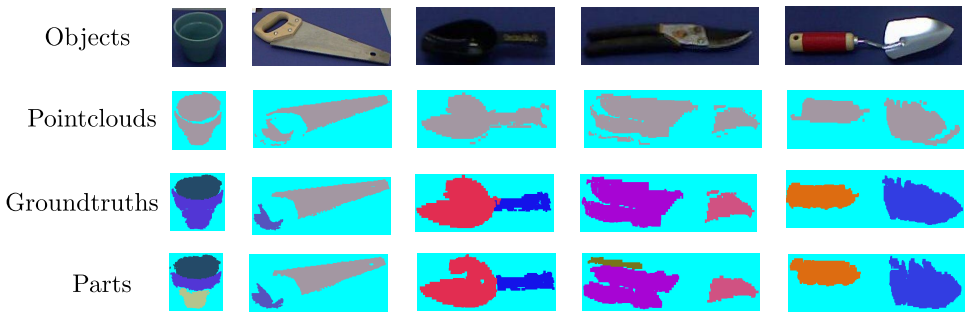

Figure 4. Part decomposition on novel categories. First row: objects; second row: corresponding pointclouds; third row: ground-truth parts illustrated by different colors; fourth row: segmented objects using our method. Colors are assigned randomly, each color representing one object part. Segmented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

[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