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.

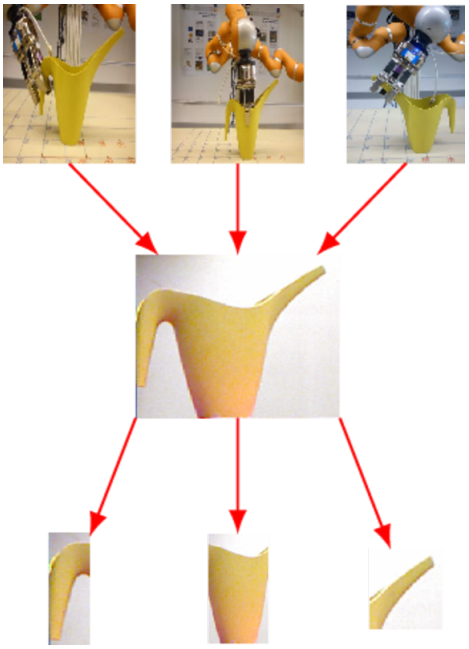


Figure 1. Object parts have semantics or functionality. Object parts can be formed based on a function such as graspablity.

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

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.

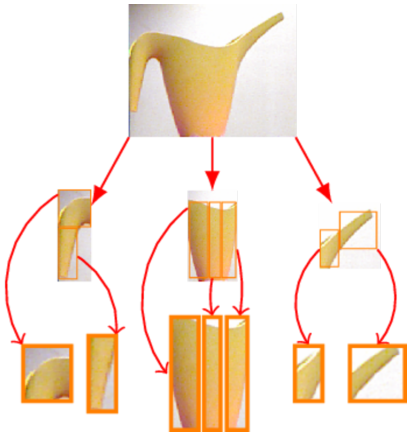


Figure 2. An object consists of a set of parts. Each part consists of a set of patches. The patches might not be discriminative on their own, but the relation between them can be distinctive.

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

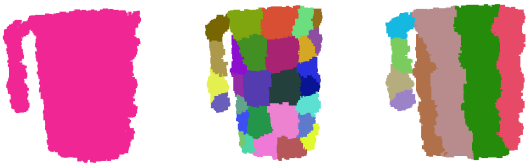
The training process for learning our compositional part model (Figure 2) can be summarized as follows. The input to our system is an RGB-D point cloud, from which we use only its depth data. The representation of an object corresponds to its parts configuration (middle level of Fig 2). Parts are subsequently 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

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/

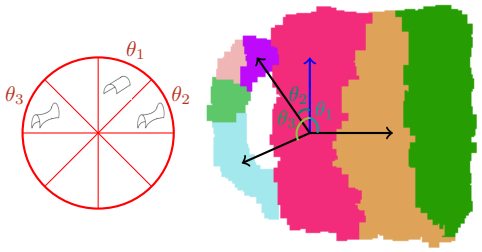


Figure 4. Patch representation. The patch descriptor is computed based on the relation of each reference patch with its neighbors. Its main axis of symmetry (blue arrow) defines the local coordinate system. Together with the reference patch, neighboring patches in different spatial locations may form 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

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.

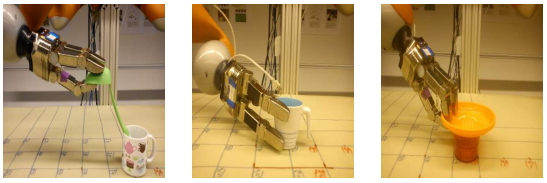


Figure 5. Kinesthetic grasp teaching for collecting patches that form a region and hence a part.

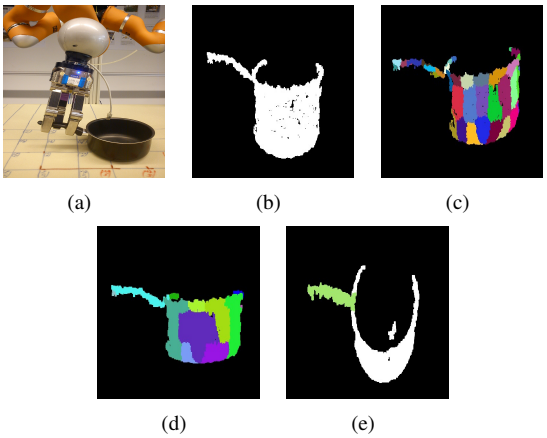


Figure 6. Figure 6(a) shows the kinesthetic grasp teaching on a *pot*. The original RGB-D data is depicted in Figure 6(b). The supervoxels are shown in Figure 6(c). Object decomposition into patches is shown in Figure 6(d). The patches contacted by grasping are shown in Figure 6(e).

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, $p(Y(x_1,x_2)|x_1,x_2)$,

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.

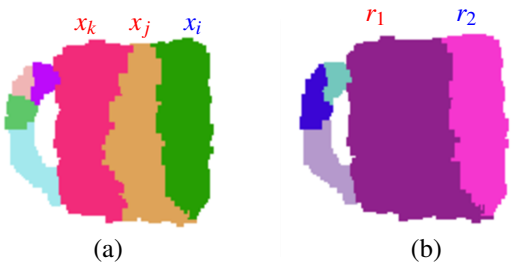


Figure 7. Merging regions based on their constituent patches. (a) object patches, (b) merged regions from patches. Starting from region r_1, x_j denotes its boundary patches. Patches adjacent to x_j inside r_1 are denoted x_k, and x_i inside r_2.

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

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/

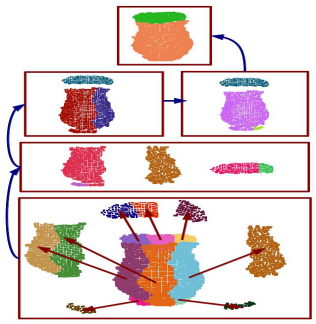


Figure 8. Compositional representation for forming object parts.

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,

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

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

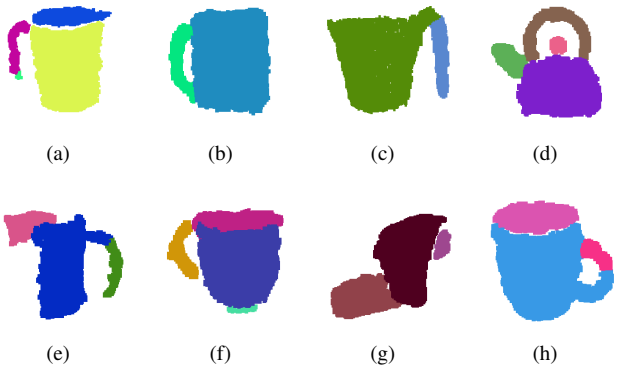


Figure 9. Examples of segmented objects in IKEA and Washington datasets.

(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.

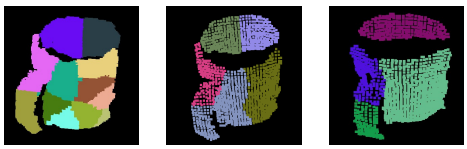


Figure 11. Example of a poorly-estimated part due to inaccurate supervoxel segmentation.

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-).

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.