Learning Objects Parts through Grasping

Safoura Rezapour Lakani, Mirela Popa, Antonio J. Rodríguez-Sánchez, Justus Piater

Abstract— **This paper presents a novel method for decomposing objects into semantically meaningful parts based on robotic grasping experiments. We propose a bottom-up compositional method for forming meaningful object parts based on their sub-parts. The composition is guided by grasping experience. The compositional nature of the method results in finding novel object parts in unseen objects where the detected sub-parts are similar to those of the previously-encountered objects. We obtained promising results on part decomposition of novel objects.**

I. INTRODUCTION

Part-based object recognition has been studied in the computer vision domain for decades [1], [2], [3]. Representing an object by a configuration of its 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 has an important impact on object recognition and classification performance.

In the state-of-the-art methods for part-based object recognition, parts are manually labeled in training examples, and conventional feature descriptors are extracted from them. For recognition, a sliding-window method at different scales matches the area of interest to the trained models. In other work [4], [5], [6], parts are compositions of sub-parts, where the sub-parts are small patches of specific sizes and at different resolutions. The parts are also labeled manually for obtaining a better parametrization of the object structure. Recognition is performed mostly in two ways. Recognition is performed mostly in two ways. First, a bag-of-words method on the patches based on different parts is employed [7]. Then an *SVM*-based classification is performed on the obtained histograms. Many methods make use of a graphical model based on the patches that compose a labeled part during training [8]. In the work discussed in [9], object parts are formed in a compositional representation of sub-parts that constitute them. The sub-parts are learned from their co-occurrence statistics in training data.

There are also methods that make use of geometrical information for object part segmentation. In the work discussed in [10], geometrical structure is estimated from depth cues and surface normals. Then a learning algorithm classifies major object and scene surfaces and use them for a recognition task. Looking into convexity or concavity of object patches for finding an object part is discussed in [11]. Initially, an object is segmented to supervoxels. The part segmentation works on the supervoxels. The object is segmented into parts if the degree of convexity between adjacent supervoxels is less than a predefined threshold. In this work the semantic meaning of a part is assumed to be based on the local convexity or concavity.

Considering the aformentioned approaches, two main questions arise. First, what are the meaningful parts? Second, how distinctive must the part representation be? We start with the second question. As it can be noted in the aforementioned methods, parts are just a composition of same-size patches. These patches by themselves are not distinctive enough. In other words, these patches can be found in multiple parts in an object. Because we consider a specific scale for the patches, we might not be able to recognize parts distinctively across the object, as locally the patches are similar. Thus, a scale invariant representation of patches needs to be addressed. In addition, the relation between the patches that compose one part should be also distinctive enough for capturing intra-parts variability. Considering 3D objects, this relation is based on surfaces and curves that compose a specific object part. For the first question, the semantic meaning of a part needs to be defined. Most often, parts are defined based on performing a certain task such as manipulation or recognition. Also, part decomposition can be based on grasping experiments. In other words, segments of objects that can be grasped can form a semantically meaningful part. One source of inspiration to learn parts from grasping is humans, where infants learn to grasp objects before those objects are visually familiar [12].

Motivated by the open problems mentioned above, we propose an object segmentation method on RGB-D data. We address the problem of object decomposition into semantically meaningful parts. We consider a compositional bottom-up approach for forming object parts through a configuration of patches. We focus on a scale-invariant and distinctive patch representation. As noted above, the composition is guided by grasping experience. It should be noted that this compositional learning of meaningful parts for grasping would benefit both recognition and grasping. More precisely, from the learned object segments that form a parts, we can segment objects that seem visually different but share the same parts. The part generalization and compositional perspective of the proposed framework would then facilitate the generalization of the grasping paradigm to novel objects. Though, grasping generalization is not the focus of this work and it is regarded as future work.

We give first an overview of our bottom-up compositional method in Section II. In Section III, we show how grasping experience can be used in collecting statistics for forming meaningful object parts. Section IV explains the probabilistic method for transferring the learned statistics from grasping to

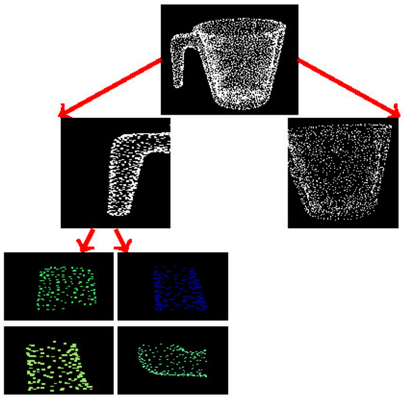


Fig. 1. Object parts and patches

form parts in a novel object. We report the evaluation results in Section V, and finally we close the discussion with a brief conclusion and an overview on future work in Section VI.

II. BOTTOM-UP OBJECT COMPOSITION

In this section, we explain our bottom-up compositional model for forming parts in an object. We consider robotics grasping experience for forming the parts.

The input to the system is an RGB-D point cloud where we only make use of the depth data. We want to form object parts from the input object points. As shown in Figure 1, an object is considered a configuration of parts such as a *cylindrical body* and a *handle* for the presented *mug*. Unlike representing a part by a combination of voxels at a specific scale, we represent it as a configuration of sub-parts that are more distinctive than voxels. As can be seen in Figure 1, a object part consists of some regions which have different surface shapes such as concave, convex, flat, hyperbolic. In each specific region, the surface normal changes smoothly. Hence, a region consists of a patches that seem locally flat but considering pairs of adjacent patches, substantial change in normal vectors can be observed.

Hence, we want to have a compositional representation of object in terms of its *parts*, *regions* that compose a part and *patches* that compose a region. We want to have distinctive parts in an object. Hence, we enforce that to the regions and patches that compose a part. We did not considered composition of object points to patches in our compositional representation because the points that form a patch are not distinctive by themselves. The patches are in the lowest level of compositional model. The patches by themselves are not distinctive enough but considering pairs of them, we can compose a *region* that can be found distinctively in a part. It should be noted that, in this work, we did not consider surface estimation in terms of concavity, convexity or flatness in a specific region, and we only considered the relation between patches to compose a region and inherently a part.

A. From Points to Patches

As mentioned in Section II, in the lowest level of our object compositional representation, we have locally flat patches. So, our first goal is to compose object input points into locally flat patches such that the surface shape considering each two adjacent patches should be non-flat. An obvious way to form these patches is to merge points that have parallel normal vectors. However, estimation of normals based on small neighborhoods around points is very noisy in typical depth data. To overcome this problem, we should consider larger regions. To this end, we form supervoxels on the input RGB-D point cloud. We used the supervoxel algorithm [13] (available in the Point Cloud Library[1]). This method uses K-Means, starting with evenly-distributed cluster seeds over an object. Then we merge the supervoxels if their mean normal vectors are close to parallel. In Figure 2, the supervoxels and merged supervoxels for some objects are shown. We then use the obtained flat patches considering their relation between adjacent patches and grasp experience to form object parts.

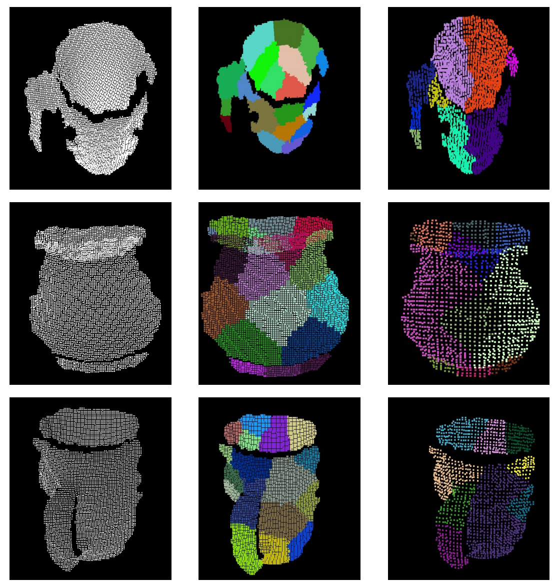


Fig. 2. From left to right, original RGB-D pointcloud, Supervoxels and flat patches

III. TRAINING A PART MODEL

Given the flat patches described in Section II, we want to form parts based on grasping experience. To this end, we decompose an object into patches. We then perform kinesthetic grasp teaching on the object as shown in Figure 3. From the grasping contact points, we can identify the patches that are grasped. In Figure 4 an example of extracting patches based on kinesthetic grasp teaching is shown. This gives us information about the patches that can be combined into a part. Therefore, we collect statistics about co-occurrence of two adjacent patches, and we use these statistics for finding parts in a novel object.

A. Patch Representation

We need to have a descriptor to characterize a patch. As mentioned in Section II, patches are assumed to form

[1]`http://pointclouds.org/`

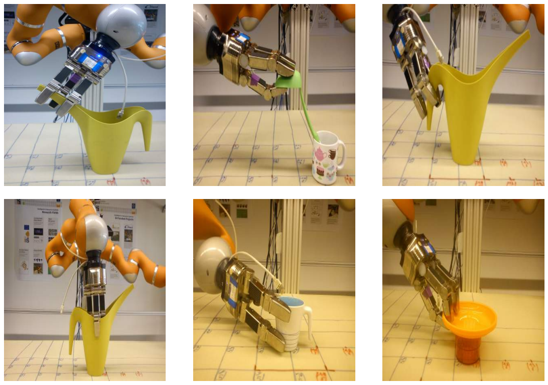


Fig. 3. Kinesthetic grasp teaching for collecting patches that form a region and hence a part.

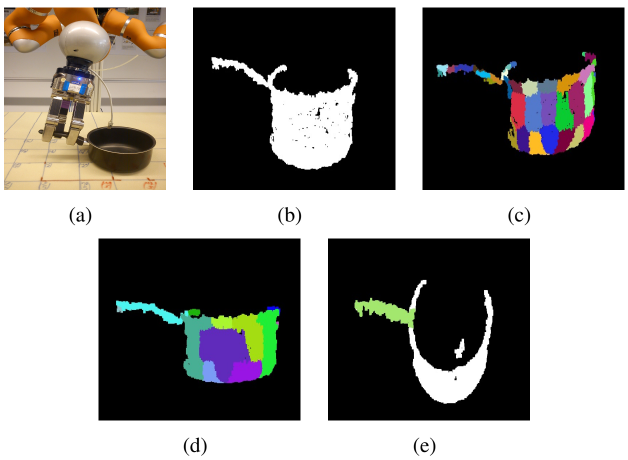


Fig. 4. Figure 4(a) shows the kinesthetic grasp teaching on a *pot*. Object decomposition into patches is shown in Figure 4(b). The patches contacted by grasping are shown in Figure 4(e).

regions based on the relation with neighbors. We exploit this assumption to construct a descriptor for a patch. In this way, we encode the relation between each patch and either of its neighbors by the angular relation between their normal vectors. Since the patches are locally flat, we only consider their mean normal vector as the representative normal vector. We then quantize these angular relations into a histogram, which becomes our patch descriptor.

B. Clustering Patches

For part identification, we want to find the co-occurrences of the adjacent patches from a dataset of objects. For efficiency and compactness, we form a codebook from the extracted patches by clustering the patches based on their aformentioned descriptors. We use hierarchical agglomerative clustering for this purpose. In this procedure, each patch initially constitutes its own cluster; we then merge clusters incrementally if the measured distance between them is lower than a certain threshold.

There are two choices to be made here, which are the distance measure and the merging threshold. As we represent the patches by a histogram, we use the *Chi-Square* distance as the distance measure. For the distance merging threshold, we make use of the training data as follows. We decomposed training objects into patches and observed which patches form a part together. We consider all pairs of adjacent parts in an object. For each pair, we then measure the distance between the patches along the boundary between the two adjacent parts. Finally, we pick the minimum distance among all the objects' parts as the merging threshold. We then form a codebook of the obtained clusters based on their centers.

Fig. 5. One sample cluster in IKEA kitchen objects. From left to right, three different instances of the cluster, shown in the context of the whole object. For better visualization, the images are shown.

Figure 5 shows one cluster of our own collected set of IKEA objects. The patches inside each cluster are projected to their corresponding objects. For better visualization, the corresponding patch and its respective object are shown. As it can be seen, the clusters are formed based on the surface shape relation with local neighborhood.

C. Co-occurrence Statistics

We want to collect statistics of co-occurrences of adjacent patches that form the same part. We have clusters of patches from an object database described in Section III-B. Based on these clusters, we can estimate the co-occurrences of adjacent patches. Hence, for each two adjacent patches that belong to the same part (x_1, x_2), we match them to the codebook and find all possible matching clusters for either of them, (h_1, h_2), where h_1 contains all the matched clusters for patch x_1 and likewise h_2 for x_2. We then increment the table entry for each two pairs in h_1 and h_2. After this, we normalize the table entries to get the probabilities. Therefore, we have a co-occurrence table based on clusters where each cluster represents a patch type and we use that for forming parts in novel objects.

IV. Part Inference in Novel Objects

We want to form parts in novel objects based on 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 to merge the patch with its neighbor of maximum probability. We perform this procedure in multiple iterations. At the first iteration, patches merge together and form regions. Next, regions merge together. We stop when no more merges are possible. At the end, parts based on learned statistics have been identified.

A. From Patches to Regions

As mentioned earlier, we want to estimate the co-occurrence between each patch and its neighbors based on the learned codebook. Then, we grow the 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_l from its neighborhood $N(x_1)$ that is most probable to form a region with x_1, based on the learned codebook:

$$x_l = \underset{x \in N(x_1)}{\operatorname{argmax}}\, p(Y|x_1,x) \tag{1}$$

Assuming an uninformative prior $p(Y)$, we instead maximize $p(x_1,x|Y) \propto p(Y|x_1,x)$. We make use of our codebook C to compute the probability of x_1 and x forming a region:

$$\begin{aligned} p(x_1,x|Y) &= \sum_{c \in C} p(x,x_1|c,Y)p(c|Y) && (2)\\ &= \sum_{c \in C} p(x,x_1|c)p(c|Y) && (3) \end{aligned}$$

Equation 3 follows from the conditional independence of neighboring patches forming a region given their common codebook vector, since we encoded the co-occurrences of neighboring patches that form a region in the codebook. Inside the sum, the first factor factorizes as

$$p(x,x_1|c) = p(x|x_1,c)p(x_1|c). \tag{4}$$

The second factor of Eqn. 4 indicates whether a patch x_1 can be matched to cluster c, while the first factor indicates whether a patch x can then be merged with x_1. Based on the learned co-occurrence table, we extract a set of patches that can co-occur with cluster c, and check whether patch x can be matched to them, that is,

$$\begin{aligned} p(x|x_1,c) &= \sum_{h \in H} p(x|x_1,c,h)p(h|x_1,c) && (5)\\ &= \sum_{h \in H} p(x|h)p(h|c) && (6) \end{aligned}$$

where H is a set of clusters that can co-occur with cluster c. Equation 6 follows from the facts that x is conditionally independent of c and x_1 given h, and that h is conditionally independent of x_1 given c.

We compute $p(h|c) = \frac{p(h,c)}{p(c)}$ by taking $p(h,c)$ from our co-occurrence table and assuming $p(c)$ is uniformly distributed. Finally, we assume that the conditional probability of a cluster c given Y is uniform which substitutes the last term in Eqn. 3.

After calculating potential matches x for all the neighboring patches to x_1 in this fashion, we merge those that maximize the probability $p(Y|x_1,x)$ of forming a part (1).

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

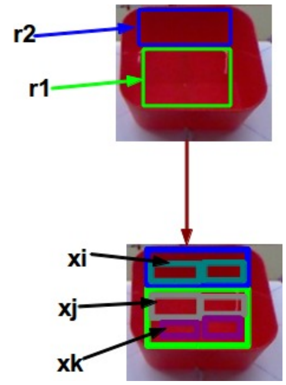


Fig. 6. Two adjacent regions in the *Red Container* object. Starting from region r_1, x_j denotes its boundary patches. Patches adjacent to x_j inside r_1 are denoted x_k, and x_i inside r_2. For visualization purposes a region has rectangular shape, on general a region can have any shape

$$r_l = \underset{r \in N(r_1)}{\operatorname{argmax}}\, p(Y|r_1,r), \tag{7}$$

where $p(Y|r_1,r)$ denotes the probability that r_1 and r belong to the same part. Assuming an uninformative prior $p(Y)$, we instead maximize $p(r_1,r|Y) \propto p(Y|r_1,r)$.

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 6, region r_1 has 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:

$$p(r_1,r|Y) = \sum_{x_j} p(r_1,r|x_j,Y)p(x_j|Y) \tag{8}$$

The first term in the summation is assumed to be independent of Y given x_j

$$p(r_1,r|Y) = \sum_{x_j} p(r_1,r|x_j)p(x_j|Y) \tag{9}$$

We then factorize the first term in Eqn .9 as follows,

$$p(r_1,r|x_j) = \sum_{x_j} p(r|r_1,x_j)p(r_1|x_j)p(x_j|Y) \tag{10}$$

We consider $p(r_1|x_j)$, 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:

$$p(r_1|x_j) = \prod_{\{x|x \in r_1 \wedge x \in N(x_j)\}} p(x|x_j) \tag{11}$$

where $N(x_j)$ stands for the neighbors of patch x_j. To estimate the conditional probability of adjacent patches, we marginalize over codebook clusters c, similarly to Eqn. 2:

$$p(x|x_j) = \sum_{c \in C} p(x|x_j,c)p(c|x_j) \tag{12}$$

The conditional probability of a cluster c given patch x_j is not given, hence we calculate it as,

$$p(c|x_j) = \frac{p(x_j|c)p(c)}{p(x_j)} \tag{13}$$

where $p(x_j)$ is considered uniform based on all the patches in an object, and $p(c)$ is also uniform based on the number of clusters. Therefore, $p(x_j) = \frac{1}{N_P}$ where N_P stands for the number of the patches and $p(c) = \frac{1}{N_C}$ where N_C stands for the number of clusters. 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:

$$p(r|r_1,x_j) = \prod_{\{x|x\in r \wedge x\in N(x_j)\}} p(x|r_1,x_j) \tag{14}$$

We consider that $p(x|r_1,x_j)$, the conditional probability of a patch x in region r given region r_1 and its neighboring patch in r_1, x_j to be independent of r_1,

$$p(r|r_1,x_j) = \prod_{\{x|x\in r \wedge x\in N(x_j)\}} p(x|x_j) \tag{15}$$

To calculate the conditional probability of adjacent patches, we marginalize over the codebook clusters c,

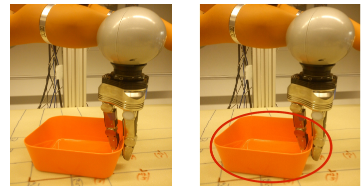
$$p(x|x_j) = \sum_{c\in C} p(x|x_j,c)p(c|x_j) \tag{16}$$

We then substitute Equations 11 and 16 into Equation 10. Furthermore, we assume that the conditional probability of a patch x_j given Y is uniform, eliminating the last factor inside the sum of Equation 10. From this, we obtain the potential of merging two adjacent regions r_1, r to form a part. We merge those that maximize the probability $p(Y|r_1,r)$ of forming a part.

V. EXPERIMENTS

We evaluated our part compositional method on our own collected IKEA kitchen objects as well as the RGB-D Washington object database [14]. For the IKEA dataset, we performed kinesthetic grasp teaching to obtain the parts. We considered semantically meaningful grasps that are associated with one and only one part of an object. Also, we tried to cover the entire area of a specific object part. For example, a single rim grasp as shown in Figure 7 cannot cover the entire area of the grasped part. Hence, we did multiple rim grasps around the object rim to cover the part completely. Our robot is a two KUKA 7-DoF Light-Weight Robot 4+ 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.

To show the applicability of the proposed part segmentation method, we evaluated it on the Washington RGB-D dataset. However this dataset is very rich for RGB-D data, but the objects in this dataset do not have a complex structure. Therefore, we considered the objects which have more complex structure. For that dataset, we labeled the graspable parts for three categories of mugs. We sampled 30% of the data from each instance category for training and 10% of them as the models for collecting statistics.



(a) Grasp teaching (b) Semantic part related to the grasp

Fig. 7. Example of kinesthetic grasp teaching 7(a). As shown, the grasp covers only one portion of the semantic object part 7(b).

Category	Proposed method	LCCP
IKEA kitchen objects	83%	52%
Washington RGB-D mugs	76%	56%

TABLE I

OVERLAP ACCURACY COMPARISON

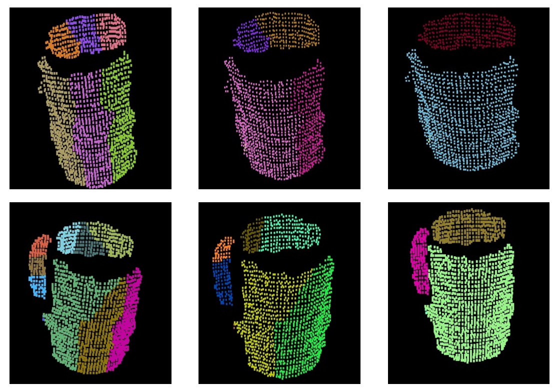


Fig. 8. Example of detected object parts in multiple iterations. From left to right, patches to regions to parts. Rows show different objects from Washington RGB-D mugs.

We divided the dataset into training and test sets. We used a random sample from the training set as the models to collect statistics for co-occurrences of patches, and the rest for clustering. For testing, we have the ground truth of object parts. We use the proposed compositional method to obtain the parts. Then, we calculate the overlap between the parts obtained by the method and the ground truth. We estimated the overlap between each ground-truth part and each estimated part, and we took the average overlap for each object. The overlap is calculated as

$$\text{overlap}(x,y) = \frac{x\cap y}{x\cup y}, \tag{17}$$

where x is a part obtained by the method, and y is the ground truth part. Table I gives overlap accuracy comparison for our proposed method and for *LCCP* [11] on the collected IKEA objects, as well as for the three categories of the Washington RGB-D dataset. Figure 8 shows some of the detected object parts in multiple iterations of the algorithm. Also some examples of extracted parts in the IKEA dataset are shown in figure 9.

We also evaluated our method on novel objects that

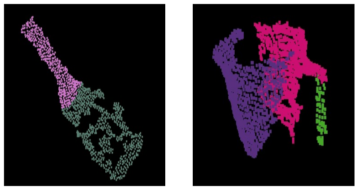
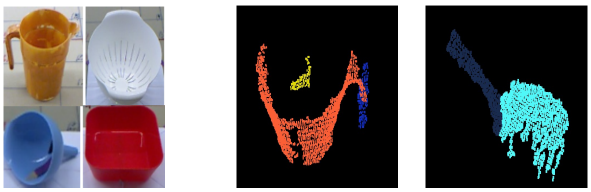


Fig. 9. Example of detected parts in IKEA kitchen dataset.

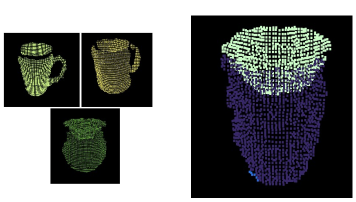


(a) (b) (c)

Fig. 10. Example of part segmentation on two object in *IKEA* objects (Figures 10(b), 10(c)) and the corresponding training models Figure 10(a). For better visualization the corresponding images for the models are shown.

share only some patches and we obtained promising results. Figure 10 shows two examples of the identified parts in the objects and the corresponding training models. We tested it with two completely different subset of objects in our own collected *IKEA* kitchen objects. The generalization is due to the compositionality nature of our approach that considers a part as a composition of regions and hence patches. It can be observed that in spite of the noise in RGB-D data, the method gives us promising results. We obtained overlapping accuracy of 72% in our IKEA objects. We also evaluated it on Washington dataset where we trained the models of three types of mugs and we evaluated it on another type of mug. The result for this experiment is also shown in Figure 11. We obtained 79% overlapping in this experiment. The evaluation on two different sets of mugs in Washington RGB-D database is noted in Table II. In the table, we can observe the overlapping accuracy of training on one set of mugs verse the other as well as evaluation on the same type of mugs.

As can be seen in Figures 12(a) and 12(b), the detected part for the *handle* is not precise. The main source of error lies in the poorly-estimated supervoxels, which propagate to imprecisely estimated patches. An example of this mis-



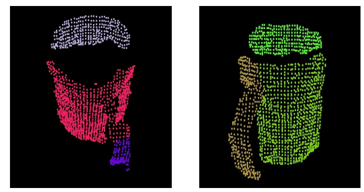
(a) (b)

Fig. 11. Example of part segmentation on a mug in Washington RGB-D dataset (Figure 11(b)). The corresponding training models are shown in Figure 11(a).

Object Category		
	72%	67%
	72%	69%

TABLE II

OVERLAP ACCURACY COMPARISON BETWEEN TWO SETS OF MUGS. HORIZONTAL AND VERTICAL AXIS STAND FOR TRAINING AND TEST CATEGORIES.



(a) (b)

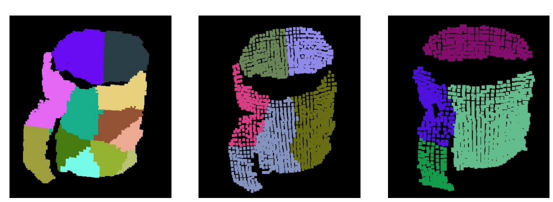
Fig. 12. Examples of poorly-estimated parts.

estimation is illustrated in Figure 13. Also, since we are using statistical learning, for parts which are not often seen, we cannot obtain a high probability and hence they will not be merged together.

VI. CONCLUSIONS

We presented a novel object-part segmentation approach. Our objective is to produce semantically-meaningful parts, focusing in this work on graspable parts. From grasping experiments, we collect statistics for forming parts based on grasped segments of an object. We utilize these segments as patches to compose object parts. We proposed a compositional method for forming parts based on scale-invariant though locally distinctive object patches. The evaluation results show promising results on novel objects.

As the patch descriptor, we considered semi-local relations between adjacent patches. This descriptor proved to fulfill our requirement for the moment. However, as future work, we will considering a stronger descriptor that jointly encodes local, relational information and surface shape.



(a) (b) (c)

Fig. 13. Example of a poorly-estimated patch due to inaccurate supervoxel segmentation.

APPENDIX

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," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, Jan. 2005.

[2] M. C. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," 1998.

[3] M. Burl, T. K. Leung, and P. Perona, "Face localization via shape statistics," 1995.

[4] 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)*, June 2012.

[5] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[6] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ser. ICCV '03. IEEE Computer Society, 2003, pp. 10–.

[7] R. Toldo, U. Castellani, and A. Fusiello, "A bag of words approach for 3d object categorization," in *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*, ser. MIRAGE '09. Springer-Verlag, 2009, pp. 116–127.

[8] S. M. A. Eslami and C. Williams, "A generative model for parts-based object segmentation." in *NIPS*, 2012, pp. 100–107.

[9] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *CVPR*, 2007.

[10] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[11] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter, "Object partitioning using local convexity," June 2014.

[12] R. K. Clifton, D. W. Muir, D. H. Ashmead, and M. G. Clarkson, "Is visually guided reaching in early infancy a myth?" *Child Development*, vol. 64, pp. 1099–1110, 1993.

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

[14] 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.