Learning Part-Based 3D Compositional Object Representations

Anonymous CVPR submission

Paper ID 2416

Abstract

This paper presents a novel approach to parts-based object representation from depth images. We propose a bottom-up compositional model for representing object classes. Our model uses a probabilistic approach to build object representations starting from small patches that are successively built up into regions, and finally meaningful (possibly graspable) parts . Parts represented this way are the main representatives of the identity of an object class. We have evaluated our method at parts recognition and object categorization, outperforming competing methods.

1. Introduction

Object recognition is of interest for applications in robotics, scene understanding and tracking, just to name a few. In these applications, the generalization of recognition to novel previously unseen objects at the category or instance levels are of the great importance. In the same level of importance for these applications is robustness in the presence of clutter and occlusion. Parts-based object representation methods can help overcome some problems object recognition has to face [5]. In these methods, objects are represented by a configuration of parts where certain parts are to be shared between different objects or even object classes. This shareability allows for generalization of recognition for certain classes of objects. Representing objects by the configuration of its parts allow us to propose a probabilistic object representation which can be useful for overcoming occlusion scenarios

The most important challenge parts-based methods have to face is characterizing object parts in a learning stage and establishing the relation between those parts. Parts are either labeled from training examples [14, 7, 18] or are obtained in an unsupervised manner by making the assumption that certain specific geometrical constraints - such as specific amounts of local convexity or concavity [3] - hold. The parts are mostly characterized by the sub-parts that compose them in terms of either fixed size patches [8], supervoxels [3] or a distribution of visual features [17]. For classification, these methods use either Graphical models, such as belief propagation (imposing a relationship on the parts and their constituting sub-parts [15]) or bag-of-words approaches applied on the extracted parts [11, 17].

Compositional representations methods have shown to be of interest on 2D contours and edges based systems [6, 20, 12]. We propose that compositional representations can also be useful when dealing with 3D object representations. We will present here a novel 3D parts-based representation that 1) follows a compositional representation rather than a feature-based representation and 2) uses parts as the main distinctive characteristics that differentiates objects. For the former, this compositionality gives us the generalization capabilities of recognizing parts and their components. Our compositional model allows us not only to build a model of an object in a bottom-up fashion starting from patches, but also - since the representations are there - provides the possibility of extracting more abstract properties of the objects than its features, such as its component parts. For the latter, parts are composed of patches, which by themselves have little discriminative power, but they will be combined into a representation that constitutes a discriminative object part. Furthermore, we propose a scale-invariant representation of patches that allows us to form parts with different geometrical shapes and scales.

In Section 2 we give an overview of our compositional bottom-up system. Then, in Section 3 we explain our learning pipeline. Section 4 describes the probabilistic inference based on the compositional method to recognize object parts and categories. We report on the experiments in Section 5. Finally, we summarize the discussion with a brief conclusion and an overview of possible future work in Section 6.

2. Compositional Representation of Objects

We introduce here a bottom-up compositional model to represent objects. The input data is an RBG-D point cloud where we only use depth information (thus, we do not consider color information). Objects in our representation are composed of parts which subsequently contain regions that are obtained from the input data points through their compo-

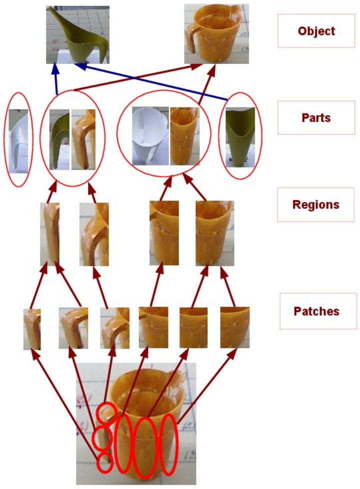

Figure 1. A general overview of the compositional method.

nent patches. The reason why we are considering a bottom-up approach is because it provides us with a more robust estimation in the presence of noise and occlusion rather than, otherwise considering a top-down approach.

As mentioned earlier, our objects are represented based on their parts configuration. We then need a representation for the parts. Our parts configuration is a configuration of the regions that constitute them. In order to differentiate between object classes, we must use parts information since at the region level, regions are not discriminative enough. Regions can be grouped into concave, convex, hyperbolic and flat. We define the regions as being smoothed surfaces where the surface normals change smoothly. Regions are further composed of patches which are defined as small local flat areas, where the changes of surface normals is negligible. On the other hand, there exists a variation of surface normals between neighboring patches.

Considering this compositional model, we can then infer that an object is composed of patches at the lowest level, which subsequently form regions, which are the components of parts. Figure 2 shows an overview of the architecture of our approach. How each of these representational levels are constructed is explained in detail in the following sections.

3. Learning the Compositional Object Model

We explain here the training procedure used for learning our compositional object models. Based on our bottom-up composition, we have three levels in the model which form *patches*, *regions*, *parts* and *objects* (Figure 2). Details of this procedure are given in section 4. We summarize next how we perform the training and representation at each of these levels.

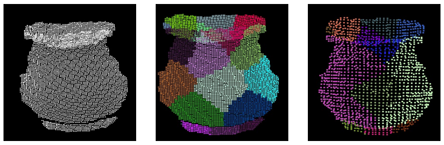

Figure 2. Decomposition of an object into patches. From left to right, original object pointcloud, its supervoxels and patches.

3.1. Training Patch Models

As shown in Figure 2, patches form the lowest level of our compositional model. Patches are combined into regions that in turn form parts. We define patches as local surfaces that can be approximated by a plane, but have a relationship to neighboring patches where there is a substantial change in surface normals. When considered individually, patches carry barely any information. The importance of patches lies here in the relationships to neighboring patches. We now explain how patches are obtained from depth images and how they are represented such that they build up to discriminative parts.

From Points to Patches We first transform the 3D point clouds into local planar surfaces. We do this by merging adjacent points whose normal vectors are parallel. Since depth data is quite noisy, just performing this operation yields quite unreliable approximations. To overcome this problem, we require larger regions that allow robust surface normals to be estimated. We use RGB-D supervoxels, computed with 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 pointcloud. We then consider only supervoxels for forming patches. We merge adjacent supervoxels if their mean normal vectors are close to parallel. Figure 2 shows decomposition of an object into patches starting from its supervoxels. It should be noted that even though the supervoxels compose patches, they are not considered in our compositional model.

Patch Representation We represent a patch in terms of its relation to its neighbors. For the representation we take two criteria into account, the spatial relation between adjacent patches and the surface curvature. For the first criterion, we consider a reference coordinate system on the patch of interest and represent the spatial relation to its neighbors with respect to that coordinate system. The reference system is the axis of symmetry corresponding to the patch. It should be noted that, since we only consider a smooth object region, we have robust estimation of the symmetrical axis. To calculate this axis, we compute the inertia matrix [19]. We then place each patch's neighbor in a 3D histogram indexed by polar coordinates based on the calculated axis. For the second criterion, we consider the curvature on

[1] http://pointclouds.org/

the separation between neighboring patches and set it into the relative polar location calculated as before.

3.2. Training Part Models

In our compositional object representation, object parts are composed of regions of adjacent patches. In order to form parts, the objective is to find out which patch types co-occur together to form an object part. With this idea in mind, we collect statistical co-occurrences of certain patch types that compose a part. The input to the system is an object dataset, where only about 10% of them are labeled. The idea is to use the unlabeled ones to train the different patch types and utilize the labeled objects to collect the statistical data about the co-occurrence of certain patch types in their labeled parts.

For finding the different patch types, we decompose our training data into patches which are then grouped by hierarchical agglomerative clustering [4]. First, each patch is considered a cluster by itself. Clusters are merged if their average distance is less than a certain threshold. This threshold is obtained from the parts corresponding to the labeled data. We obtain the minimum distance between the patches which are adjacent as well as belonging to different parts.

Then, we create a codebook from the clusters and use that for collecting the patch co-occurrence statistics. In order to do that, we first decompose our labeled data into patches. We then consider each two adjacent patches that compose a part and match each of them to the codebook. This gives us the hypotheses regarding which clusters match each of the patches. After that, we increment the frequency of co-occurrence of each of the two matching clusters. Hence our statistical data is based on the co-occurrence of the clusters which are implicitly related to object patches. Next, we make a codebook of parts based on the obtained clusters. Finally, we follow the probabilistic method described in Section 4.1 to compose parts based on the learned patch codebooks.

3.3. Training Object Models

Objects are at the highest level of our compositional model. They are composed of a configuration of parts whose starting point were locally planar surfaces, or patches. At this point, we need to represent which object parts co-occur together to form a certain class of objects. Thus, we will decompose objects into parts and recognize part types. We then follow the idea presented in section 3.2 to collect statistical data, but this time regarding adjacent parts.

The common way to recognize different part types is to cluster them. Since, parts are composed of patches, we represent them as a histogram of patches. As described in section 3.2, we obtained a codebook of different patch types. Hence, the parts are encoded as a histogram consisting of how frequent they appear on certain codebook entries. We then do a clustering on the aforementioned histogram to obtain specific part types. A codebook is then build from the clusters of parts. This second codebook gives us only information about the occurrence of a certain part in an object. In addition to this information, we will also need the statistics on the co-occurrence of parts. To achieve this, we will consider the composing patches that connects two adjacent parts. Just considering these selected part patches provides a higher degree of flexibility that allows us to achieve a better generalization capabilities (than otherwise considering all patches inside both related parts). We will collect on the statistical co-occurrences of these selected patches corresponding to neighboring parts as described in detail in section 4.4.

4. Probabilistic Part and Object Recognition

In this section, we describe details regarding the inference in our compositional object model that we briefly explained in Section 3.

4.1. Part Recognition

As described in Section 3.2, object parts are composed of patches. We collected the statistics on adjacent patches that compose a part. In order to form a part in a novel object, we consider adjacent patches and we merge them based on the learned patch co-occurrence statistics. This is an iterative procedure, in which we first merge adjacent patches to obtain regions. Then, these regions can be further merged to form an object part. We explain next the probabilistic formulation for composing regions from patches and composing parts from regions.

4.2. From Patches to Regions

In this section, we provide the mathematical formulation behind how a part region is represented from patches that are adjacent. Let z_1 and z_2 indicate two adjacent patches then $p(Y|z_1, z_2)$ is the probability that asserts whether the two patches can form a part region.

Given the object patches, we start from a random patch z_1 and we merge it with its neighbor z_l that is more likely to form a part region.

$$z_l = \operatorname*{argmax}_{z \in N(z_1)} p(Y|z_1, z) \quad (1)$$

Assuming an uninformative prior $p(Y)$, we instead maximize $p(z_1, z|Y) \propto p(Y|z_1, z)$. We make use of our patch codebook C to compute the probability of z_1 and z forming a region:

$$p(z_1, z|Y) = \sum_{c_1 \in C} p(z, z_1|c_1, Y)p(c_1|Y) \quad (2)$$

$$= \sum_{c_1 \in C} p(z, z_1|c_1)p(c_1|Y) \quad (3)$$

Equation 3 follows from the conditional independence of neighboring patches forming a region given their common codebook vector c_1, since we previously encoded the co-occurrences of neighboring patches that form a region in the codebook. Inside the sum, the first factor factorizes as

$$p(z, z_1|c_1) = p(z|z_1, c_1)p(z_1|c_1). \quad (4)$$

The second factor of Eqn. 4 indicates whether a patch z_1 can be matched to cluster c_1, while the first factor indicates whether a patch z can then be merged with z_1. Based on the learned co-occurrence table, we extract a set of patches that can co-occur with cluster c_1, and check whether patch z can be matched to them, that is,

$$p(z|z_1, c_1) = \sum_{c_2 \in C_2} p(z|z_1, c_1, c_2)p(c_2|z_1, c_1) \quad (5)$$

$$= \sum_{c_2 \in C_2} p(z|c_2)p(c_2|c_1) \quad (6)$$

where C_2 is a set of clusters that can co-occur with cluster c_1. Equation 6 follows from the facts that x is conditionally independent of c_1 and z_1 given c_2, and that c_2 is conditionally independent of z_1 given c_1.

We compute $p(c_2|c_1) = \frac{p(c_2,c_1)}{p(c_1)}$ by taking $p(c_2, c_1)$ from our co-occurrence table and assuming $p(c_1)$ is uniformly distributed. Finally, we assume that the conditional probability of a cluster c_1 given Y is uniform which substitutes the last term in Eqn. 3.

After calculating potential matches z for all the neighboring patches to z_1 in this fashion, we merge those that maximize the probability $p(Y|z_1, z)$ of forming a part (1).

4.3. From Regions to Parts

We have now 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 random region r_1 and we merge it to an adjacent region r which is most probable to form a part, i.e.,

$$r_l = \operatorname*{argmax}_{r \in N(r_1)} p(Y|r_1, r), \quad (7)$$

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 $p(r_1, r|Y)$. Region r_1 has patches z_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:

$$p(r_1, r|Y) = \sum_{z_j} p(r_1, r|z_j, Y)p(z_j|Y) \quad (8)$$

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

$$p(r_1, r|Y) = \sum_{z_j} p(r_1, r|z_j)p(z_j|Y) \quad (9)$$

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

$$p(r_1, r|z_j) = \sum_{z_j} p(r|r_1, z_j)p(r_1|z_j)p(z_j|Y) \quad (10)$$

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

$$p(r_1|z_j) = \prod_{\{z|z \in r_1 \wedge z \in N(z_j)\}} p(z|z_j) \quad (11)$$

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

$$p(z|z_j) = \sum_{c \in C} p(z|z_j, c)p(c|z_j) \quad (12)$$

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

$$p(c|z_j) = \frac{p(z_j|c)p(c)}{p(z_j)} \quad (13)$$

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

$$p(r|r_1, z_j) = \prod_{\{z|z \in r \wedge z \in N(z_j)\}} p(z|r_1, z_j) \quad (14)$$

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

$$p(r|r_1, z_j) = \prod_{\{z|z \in r \wedge z \in N(z_j)\}} p(z|z_j) \quad (15)$$

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

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

We then substitute Equations 11 and 14 into Equation 10. Furthermore, we assume that the conditional probability of a patch z_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.

4.4. Object Recognition

Obtaining object parts as described in Section 4.1, we can further recognize its category based on our compositional model. To this aim, we make use of the patch codebook C and the parts codebook H. For the recognition task we are given a set of parts X and their configuration S in terms of the adjacency matrix. We want to recognize the category of the novel object t_X,

$$t_X = \operatorname*{argmax}_t p(t|X, S) \tag{17}$$

Due to considering a uniform prior for each object class t, we maximize then the following equation,

$$t_X = \operatorname*{argmax}_t p(X, S|t) \tag{18}$$

We assume conditional independence for each object part and its configuration. Hence, we can re-write the conditional probability in Eqn 18 as following,

$$p(X, S|t) = \prod_{x_i \in X, s_i \in S} p(x_i, s_i|t) \tag{19}$$

Further we can marginalize over all parts that vector s_i gives us. In fact, those are the adjacent parts to x_i,

$$p(x_i, s_i|t) = \sum_{x \in s_i} p(x_i, s_i|t, x)p(x|t) \tag{20}$$

We further factorize the first term in the summation,

$$p(x_i, s_i|t, x) = \frac{p(x_i, x|t)p(t|s_i)p(s_i)}{p(x|t)p(t)} \tag{21}$$

We assume that a configuration s_i is uniformly distributed for all the object classes, hence $p(t|s_i) = \frac{1}{N_T}$. Also, as we presumed earlier, the prior probability of each object class is also uniform. We further assume a uniform distribution for a specific configuration s_i over all the configuration N_S. These assumptions leave us with the following formula for Eqn 20

$$p(x_i, s_i|t) = \sum_{x \in s_i} \frac{p(x_i, x|t)}{N_S} \tag{22}$$

In order to compute the conditional probability of the adjacent patches x_i and x, we make use of their border patches. Patches z_j connects x_i to x, hence we marginalize over these patches,

$$p(x_i, x|t) = \sum_{z_j \in x_i, z_j \in N_x} p(x_i, x|t, z_j)p(z_j|t) \tag{23}$$

We can then factorize the first term in the summation of Eqn 23,

$$p(x_i, x|t) = \sum_{z_j \in x_i, z_j \in N_x} p(x|t, z_j, x_i)p(x_i|t, z_j)p(z_j|t) \tag{24}$$

We also consider that the conditional probability of the part x_i is independent of its own patch z_j given that object class t. This assumption is made because we compute this conditional probability based on the composed patches of x_i. Therefore, we obtain the following formulation for computing the conditional probability of the two adjacent part,

$$p(x_i, x|t) = \sum_{z_j \in x_i, z_j \in N_x} p(x|t, z_j, x_i)p(x_i|t)p(z_j|t) \tag{25}$$

To compute the conditional probability of the part x which is adjacent to part x_i, we consider its patches that are neighbor to z_j and we compute them independently,

$$p(x|t, z_j, x_i) = \sum_{z_k \in x, z_k \in N(z_j)} p(x|t, z_j, x_i, z_k)p(z_k|t, z_j, x_i) \tag{26}$$

where z_k denotes the adjacent patches to z_j in x. We consider that the conditional probability of the patch z_k is independent of the part x_i when its adjacent patch is given. Furthermore, we also consider that when the conditional probability of patch z_k which belongs to x is given, conditional probability of part x is only dependent to the object class t,

$$p(x|t, z_j, x_i, z_k) = p(x|t) \tag{27}$$

We factorize the conditional probability of z_k in Eqn 26 as followings,

$$p(z_k|t, z_j) = \frac{p(z_k, z_j|t)}{p(z_j|t)} \tag{28}$$

To compute the conditional probability of two adjacent patches in Eqn 28, we marginalize over our patch codebook C,

$$\begin{aligned} p(z_k, z_j|t) &= \sum_{c_1 \in C} p(z_k, z_j|t, c_1)p(c_1|t) \\ &= \sum_{c_1 \in C} p(z_k|z_j, t, c_1)p(z_j|t, c_1)p(c_1|t) \\ &= \sum_{c_1 \in C} p(z_k|z_j, t, c_1)p(z_j|c_1)p(c_1|t) \end{aligned} \tag{29}$$

Further, let's consider that patch z_j matches cluster c_1, z_k must then be matched to the clusters that co-occur with c_1. Hence, we marginalize over the clusters that co-occur with c_1

$$p(z_k|z_j,t,c_1) = \sum_{c_2 \in C_2} p(z_k|z_j,t,c_1,c_2)p(c_2|z_j,t,c_1)$$

where C_2 stands for the set of clusters that co-occur with c_1. For the first term, we assume that when the co-occurrence codebook c_2 is given, conditional probability of z_k is only dependent to c_2, which yields us,

$$p(z_k|z_j,t,c_1) = \sum_{c_2 \in C_2} p(z_k|c_2)p(c_2|z_j,t,c_1) \tag{30}$$

where the conditional probability of z_k given cluster c_2 indicates if it can be matched to c_2. The second term can be factorized as followings,

$$p(c_2|z_j,t,c_1) = \frac{p(c_2,c_1|z_j,t)}{p(c_1|t,z_j)} \tag{31}$$

Since, we collected statistics on the co-occurrences of adjacent patches that belong to different part based on a certain class of objects, we consider that the nominator in Eqn 31 is independent of the patch z_j when the object class is given. That also holds for the denominator,

$$p(c_2|z_j,t,c_1) = \frac{p(c_2,c_1|t)}{p(c_1|t)} \tag{32}$$

Finally, we substitute the obtained formulas into Eqn 22,

$$p(x_i,s_i|t) = \frac{1}{N_S} p(x_i|t) \sum_{x} p(x|t) \sum_{z_j \in x_i, z_j \in N(x)} \sum_{z_k \in x, z_k \in N(z_j)} \sum_{c_1 \in C} \sum_{c_2 \in C_2} p(z_k|c_2)p(c_2,c_1|t)p(z_j|c_1)$$

To compute the conditional probability of a part given the object class, we marginalize over our part codebook H,

$$p(x_i|t) = \sum_{h \in H} p(x_i|h)p(h|t) \tag{33}$$

where the first term indicates whether part x_i can be matched to cluster h. We compute the second term based on our collected statistics of occurrence of a certain part codebook in the object classes.

In practice, we considered two matters concerns about clustering and computing the conditional probabilities. Because we used agglomerative clustering to group patches and parts, we have clusters which contain very few number of samples or even one sample. That is due to the nature of this type of clustering. For those clusters, we allowed for a small variation ϵ when making the codebook. For computing the conditional probability, since we made independent assumption between object parts and patches, we face the problem with the parts/patches that have zero probability. That propagates to the overall computation, and gives overall zero probability. To overcome this problem, we also considered a very small probability ϵ for those parts. However, this problem can also be solved by allowing missing parts/patches in the probabilistic model and using dynamic programming to solve it, we consider that for our future work.

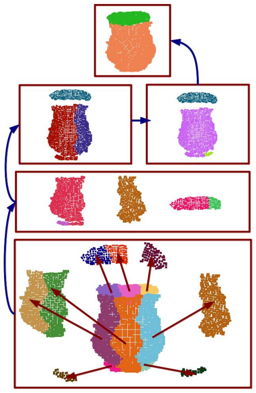

Figure 3. Compositional representation of forming object parts.

5. Experimental Results

We evaluated our compositional model mainly on two aspects: part recognition on novel objects, and object classification. We evaluated our method in the RGB-D Washington dataset [10].

5.1. Part recognition in novel objects

Parts Evaluation is computed based on the overlap of the parts recognized by our method in comparison with the ground truth. The overlap score is computed as [9, 2],

$$\text{overlap}(x,y) = \frac{|x \cap y|}{|x \cup y|}, \tag{34}$$

where x is a part obtained using our method, and y is the ground truth part.

We divide the data into training data, test data and training models. The training data are unlabeled and used for learning the codebook of patches and parts as explained in Section 3, whereas the training models correspond to the labeled data. For the test data, we labeled the parts as our ground truth for evaluation.

Figure 3 shows the compositional capability of our method in a real case scenario.

In order to show the applicability of our part recognition method on the Washington dataset, we selected the objects which have more than one part. From two categories of

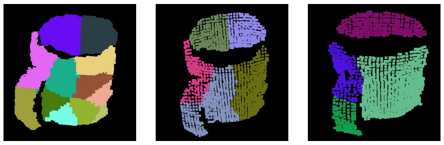

Figure 4. Example of a poorly-estimated part due to inaccurate supervoxel segmentation.

objects, we sampled a set of them for evaluation. We compared our method in this dataset with Locally Convex Connected Patches (*LCCP*) from [3]. The overlapping results as given in Table 1 which proves that our method outperforms the *LCCP* method.

Category	Proposed method	LCCP [3]
mugs	76%	58%
caps	75%	40%
staplers	88%	61%

Table 1. Overlap accuracy comparison on the Washington RGB-D dataset

In addition to the evaluating out method using the afformentioned overlap measure, we are also interested in showing the generalization capabilities of our method. With this goal in mind, we performed an experiment where our algorithm had to find parts that were already present in our part-based model, but in objects not seen previously. Table 2 shows the overlapping accuracy for novel part recognition in some objects of the RGB-D Washington dataset. Each row shows in its first column the object that was used to store the part representation in our compositional model. Then, we computed the part recognition through evaluating the overlapping accuracy for the objects in the rest of the columns.

Object Category		
	72%	67%
	72%	69%

Table 2. Overlap accuracy comparison between two sets of mugs. Horizontal and vertical axis stand for testing and training categories.

The main source of error lies in poorly-estimated supervoxels as shown in Figure 4, which then build up into imprecisely estimated patches and parts.

5.2. Object Categorization

For our second experiment, we evaluated our compositional framework on and object classification task. Again, we want to prove how our method generalizes to novel or unseen objects. In this case, we went even farther and tested our method on an object recognition task. Due to the compositionality nature of our object representation, we are able obtain a very high recognition rate as shown in Table 3 and 4. As before, each row shows in its first column the object that was used to store the part representation in our compositional model. Then, we computed the part recognition through evaluating the overlapping accuracy for the objects in the rest of the columns. Since, the staplers and caps have more intra-class overlapping, and due to space limitation, we only show a few examples. The accuracy is computed as,

$$\text{accuracy} = \frac{(\text{TP} + \text{TN})}{P + N}$$

where TP is the number of true positives and TN the number of true negatives. P and N stand for the number of positive and negative examples.

Object Instances		
	98%	97%
	96%	80%

Table 3. Classification accuracy on novel object instances of mugs in the RGB-D Washington dataset.

Object Instances				
	93%	97%	-	-
	91%	95%	-	-
	-	-	92%	85%
	-	-	84%	85%

Table 4. Classification accuracy on novel object instances of staplers and caps in the RGB-D Washington dataset.

We further evaluated the classification performance of the proposed approach using three classes of objects from the RGB-D Washington dataset. We performed binary classification between pairs of them. The precision (or positive predictive value *PPV*) and recall (or true positive rate *TRP*) for two pair of classes are shown in Table 5 and 6. We compared our results with viewpoint feature histogram (*VFH*) [16] as well as Oriented, Unique and Repeatable Clustered VFH (*OUR-CVFH*) [1] classified with K-nearest neighbor. In addition, we also learned the distance threshold for the K-nearest neighbor. Since, the most important aspect of our system is the generalization to novel object instances, we also evaluated our binary classifier on that. Hence, we trained our model with only one object instance of each class and we tested it on the other object instances. As denoted in the table, our system outperformed other method in this aspect which proves the applicability of our system.

The source of error in classification which can be observed in the stapler instances in Table 5 is again related to the poorly estimated supervoxels in the training models. Therefore, the statistics that we collect from those wrong models would be also incorrect. A common way to solve this problem is to consider an error for the training models and integrate it into our probabilistic model which we considered that as our future work.

Method				
	PPV	TPR	PPV	TPR
Our Method Novel Instances	26%	**85%**	**94%**	36%
VFH [16]	92%	98%	100%	45%
VFH [16] Novel Instances	80%	9%	76%	22%
OUR-CVFH [1]	100%	57%	100%	36%
OUR-CVFH [1] Novel Instances	-	-	100%	50%

Table 5. Precision and Recall for binary classification for object classes staplers and mugs.

Method				
	PPV	TPR	PPV	TPR
Our Method Novel Instances	**63%**	**91%**	**92%**	**93%**
VFH [16]	92%	98%	100%	45%
VFH [16] Novel Instances	50%	12%	25%	1%
OUR-CVFH [1]	100%	81%	100%	61%
OUR-CVFH [1] Novel Instances	-	-	-	-

Table 6. Precision and Recall for binary classification for object classes caps and mugs.

6. Conclusion

We have presented a novel probabilistic approach for representing 3D objects and applied it on an object recognition task. Our goal was to present a scalable and generalizable representation where object parts play a main role. We have presented a novel 3D compositional object representation where the object is composed of a configuration of parts which represent a composition of regions built from planar patches. This compositional probabilistic approach allows us to recognize certain object parts in novel, previously unseen objects as well as correctly recognizing them by using the discriminative power of parts our compositional model provides.

References

[1] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze. Our-cvfh - oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In *DAGM/OAGM Symposium*, volume 7476, pages 113–122. Springer, 2012. 7, 8

[2] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012. 6

[3] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter. Object partitioning using local convexity. In *CVPR*, June 2014. 1, 7

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000. 3

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, Jan. 2005. 1

[6] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007. 1

[7] W. Hu. Learning 3d object templates by hierarchical quantization of geometry and appearance spaces. In *CVPR*, pages 2336–2343. IEEE, 2012. 1

[8] M. Kiefel and P. Gehler. Human pose estimation with fields of parts. In *ECCV*, volume 8693, pages 331–346. Springer International Publishing, Sept. 2014. 1

[9] P. Krhenbhl and V. Koltun. Geodesic object proposals. In *ECCV 2014*, volume 8693, pages 725–739. Springer International Publishing, 2014. 6

[10] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011. 6

[11] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, June 2001. 1

[12] B. Ommer and J. Buhmann. Learning the Compositional Nature of Visual Object Categories for Recognition. *PAMI*, (99):1, Jan. 2009. 1

[13] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *CVPR*, June 2013. 2

[14] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, June 2012. 1

[15] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, Portland, OR, June 2013. 1

[16] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *IROS*, 10/2010 2010. 7, 8

[17] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *ICCV*, 2005. 1

[18] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, September 2010. 1

[19] C. Sun and J. Sherrah. 3-d symmetry detection using the extended gaussian image. *PAMI*, 19:164–168, 1997. 2

[20] L. Zhu and A. L. Yuille. A hierarchical compositional system for rapid object detection. In *NIPS*, 2005. 1