Unsupervised 3D Object Discovery and Categorization for Mobile Robots

Jiwon Shin Rudolph Triebel Roland Siegwart

Abstract We present a method for mobile robots to learn the concept of objects and categorize them without supervision using 3D point clouds from a laser scanner as input. In particular, we address the challenges of categorizing objects discovered in different scans without knowing the number of categories. The underlying object discovery algorithm finds objects per scan and gives them locally-consistent labels. To associate these object labels across all scans, we introduce *class graph* which encodes the relationship among local object class labels. Our algorithm finds the mapping from local class labels to global category labels by inferring on this graph and uses this mapping to assign the final category label to the discovered objects. We demonstrate on real data our alogrithm's ability to discover and categorize objects without supervision.

1 Introduction

A mobile robot that is capable of discovering and categorizing objects without human supervision has two major benefits. First, it can operate without a hand-labeled training data set, eliminating the laborious labeling process. Second, if a human-understandable labeling of objects is necessary, automatic discovery and categorization leaves the user with the far less tedious task of labeling categories rather than raw data points. Unsupervised discovery and categorization, however, require the robot to understand what an object constitutes. In this work, we address the challenges of unsupervised object discovery and categorization using 3D scans from a

Jiwon Shin
Autonomous Systems Lab, ETH Zurich e-mail: jiwon.shin@mavt.ethz.ch

Rudolph Triebel
The Oxford Mobile Robotics Group, University of Oxford e-mail: rudi@robots.ox.ac.uk

Roland Siegwart
Autonomous Systems Lab, ETH Zurich e-mail: rsiegwart@ethz.ch

laser as input. Unlike other object discovery algorithms, our approach does not assume presegmentation of background, one-to-one mapping between input scan and label, nor a particular object symmetry. Instead, we simply assume that an entity is an object if it is composed of two or more parts and occurs more than once.

We propose a method for robots to discover and categorize objects without supervision. This work especially focuses on categorization of the discovered objects. The proposed algorithm is composed of three steps: detection of potential object parts, object discovery, and object categorization. After segmenting the input 3D point cloud, we extract salient segments to detect regions which are likely to belong to objects. After detecting these potential object parts, we cluster them in feature and geometric space to acquire parts labels and object labels. Reasoning on the relationship between object parts and object labels provides a locally-consistent object class label for each discovered object. Processing a series of scans results in a set of discovered objects, all labeled according to their local class labels. To associate these local class labels, we build a *class graph*. Class graph encodes the dependency among local class labels of similar appearance, and smoothing the graph results in a distribution of the global category labels for each local class label. Marginalizing out the local class labels gives the most likely final category label for each discovered object. We demonstrate on real data the feasibility of unsupervised discovery and categorization of objects.

Contributions of this work are two-folds. First, we improve the object discovery process by extracting potential foreground objects using saliency. Instead of relying entirely on perfect foreground extraction, our algorithm takes the foreground segments only as potential object parts and performs further processing on them before accepting them as object parts. It can thus handle imperfect foreground extraction by removing those potential object parts deemed less fit to be actual object parts. Second, we propose a novel categorization method to associate the locally-consistent object class labels to the global category labels without knowing the number of categories. Our algorithm improves the results of categorization over pure clustering and provides a basis for on-line learning. To our knowledge, no other work has addressed the problem of unsupervised object categorization from discovered objects.

The organization of the paper is as follows. After discussing related work in Sec. 2, we introduce a saliency-based foreground extraction algorithm and explain the single-scan object discovery algorithm in Sec. 3. In Sec. 4, we propose a method for associating the discovered objects for object categorization. After the experimental results in Sec. 5, the paper concludes with Sec. 6.

2 Related Work

Most previous work on unsupervised object discovery assume either a presegmentation of the objects, one object class per image, or a known number of objects and their classes [5, 14, 2]. In contrast, [17] proposed an unsupervised discovery algorithm that does not require such assumptions but instead utilizes regularity of

patterns in which the objects appear. This is very useful for man-made structures such as facades of buildings. [3] developed a method to detect and segment similar objects from a single image by growing and merging feature matches.

Our work builds on our previous work [18], which gives nice results for single scenes but does not address the data association problem across different scenes. Thus, the above algorithm cannot identify instances of the same object class that appear in different scenes. In contrast, this approach solves the data association problem and introduces a reasoning on the object level, instead of only assigning class labels to object parts.

An important step in our algorithm is the clustering of feature vectors extracted from image segments. Many different kinds of clustering algorithms have been proposed and their use strongly depends on the application. Some classic methods such as the Expectation-Maximization (EM) algorithm and *k*-means clustering assume that data can be modeled by a simple distribution, while other methods such as agglomerative clustering are sensitive to noise and outliers. To overcome these problems, alternative approaches have been proposed. [12] presented a *spectral clustering* algorithm, which uses the eigenvectors of the data matrix to group points together, with impressive results even for challenging data. Another recent clustering approach is named *affinity propagation*, proposed by [6]. It clusters data by finding a set of exemplar points, which serve as cluster centers and explain the data points assigned to it. This method avoids the pitfalls of a bad initialization and does not require the number of clusters to be prespecified. In this work, we use affinity propagation to cluster image segments in feature space.

Our object categorization method is inspired by the *bag of words* approach [4]. Outside of document analysis, the bag of words method has been applied in computer vision, e.g., for texture analysis or object categorization [11, 16]. Our work uses it to bridge the gap between reasoning on object parts and object instances.

3 Object Discovery

This section describes the algorithm for discovering objects from a single scan. Fig. 1 depicts the overall process of the object discovery. Our single-scan object discovery algorithm is based on our previous work [18], which treats every segment as a potential object part and accepts them as objects if after inference any nearby segment has the same class label as itself. This algorithm, however, has several disadvantages. First, because the original algorithm considers all segments as potential object parts, it makes many false neighborhood connections between foreground and background segments. This results in object candidates composed of real object parts and background parts. Second, it has relatively high false-positive rate because it cannot differentiate clutter objects from real objects. Third, it wastes computation by extracting feature descriptors on background segments. In this paper, we introduce saliency-based foreground extraction algorithm to overcome these problems.

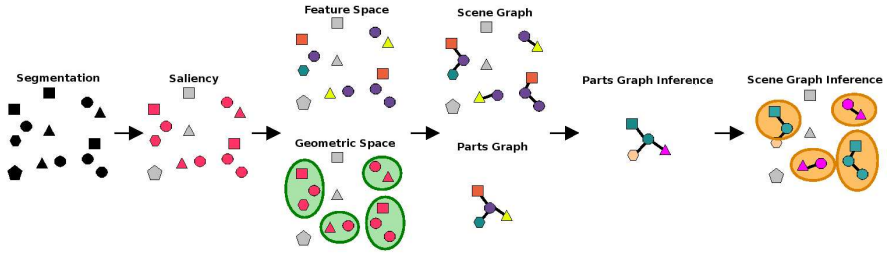


Fig. 1: Overview of the discovery process (best seen in color). After performing segmentation on input data and extracting salient segments, the algorithm clusters the salient segments in feature and geometric space. The clusters are then used to create scene graph and parts graph, which encode the relationship between object parts and objects. Running inference on the graphs result in the discovery of four objects as shown on the right.

3.1 Extraction of Potential Object Parts

A simple way to seperate foreground from background is to fit planes into the data and remove all points that correspond to the planes. This removes all wall, ceiling, and floor parts as in, e.g., [5], but can cause at least two problems. First, it may also remove planar segments close to a wall or floor that are actually object parts and thus should not be removed. Second, it is often insufficient to define background as planar because background may be truly curved or non-planar due to sensor noise.

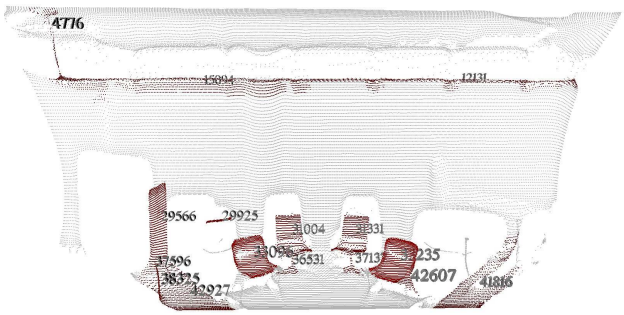


Fig. 2: An example image after saliency computation. Colored segments are considered salient and thus treated as potential object parts. Numbers indicate segment ID.

Inspired by computer vision [8], we suggest a different approach for foreground extraction using *saliency*. The idea is to classify certain parts of an image as visu-

ally more interesting or salient than others. This classifications determines saliency based on difference in entropy of a region to its nearby regions. Most work on saliency has been on 3D images, but [7] uses saliency for object recognition in 3D range scans. Their technique, however, remaps depth and reflectance images as greyscale images and applies 2D saliency techniques to find salient points. This work detects salient segments in true 3D by processing depth values of range data directly.

Our saliency algorithm computes saliency at point level and segment level. Point saliency provides saliency of a point while segment saliency represents saliency of a segment. A *point saliency* s_p is composed of a *local* saliency s_l and a *global* saliency s_g. Local saliency s_l is defined as

$$s_l(\mathbf{p}) = \frac{1}{s_l^{max}} \sum_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} \mathbf{n} \cdot (\mathbf{p} - \mathbf{p}'), \tag{1}$$

where $\mathbf{n}$ is the normal vector at a point $\mathbf{p}$, and $\mathcal{N}(\mathbf{p})$ defines the set of all points in the neighborhood of $\mathbf{p}$. To obtain a value between 0 and 1, the local saliency is normalized by the maximum local saliency value s_l^{max}. Intuitively, local saliency measures how much the point $\mathbf{p}$ sticks out of a plane that best fits into the local surrounding $\mathcal{N}(\mathbf{p})$. This resembles the plane extraction techniquementioned earlier.

Points that are closer to the sensor are more likely to belong to foreground and thus globally more salient than points that are far away from the sensor. We capture this property in global saliency. Global saliency s_g is defined as

$$s_g(\mathbf{p}) = \frac{1}{s_g^{max}} \|\mathbf{p}^{max} - \mathbf{p}\|, \tag{2}$$

where $\mathbf{p}^{max}$ denotes the point that is farthest away from the sensor origin. As in local saliency, global saliency is normalized to range between 0 and 1.

We define segment saliency s_s for a segment $\mathbf{s}$ as a weighted average of the local and global saliency for all points which belong to the segment and multiply it by a size penalty α, i.e.,

$$s_s(\mathbf{s}) = \alpha \left(\frac{1}{|\mathbf{s}|} \sum_{\mathbf{p} \in \mathbf{s}} w s_l(\mathbf{p}) + (1 - w) s_g(\mathbf{p}) \right), \tag{3}$$

where $\alpha = \exp(-(|\mathbf{s}| - |\mathbf{s}_{mean}|)^2)$ penalizes segments that are too big or too small as they are likely to originate from a wall or sensor noise; $|\mathbf{s}|$ denotes the size (number of points) of the segment $\mathbf{s}$; and w weighs between local and global saliency. The weight w depends on the amount of information contained in local and global saliency, measured by entropy of the corresponding distributions. Interpreting s_l and s_g as probability distributions, we can determine entropy h_l and h_g for local and global saliency by

$$h_l = -\sum_{i=1}^{N} s_l(\mathbf{p}_i) \log s_l(\mathbf{p}_i) \tag{4}$$

$$h_g = -\sum_{i=1}^{N} s_g(\mathbf{p}_i) \log s_g(\mathbf{p}_i), \tag{5}$$

where $N = 20$ in this work. As a saliency distribution with lower entropy is more informative, we set the weight w as $w = \frac{h_g}{h_g + h_l}$, which is high when local saliency has low entropy and low when it has high entropy. The weight ensures that more informative entropy distribution contributes more to the final saliency.

Segment saliency $s_s(\mathbf{s})$ ranges between 0 and 1. We consider a segment salient if its saliency is higher than 0.5 and accept it as a potential object part. Only these potential object parts $\mathcal{S}$ are further processed for object discovery. Fig. 2 shows a scene after salient segments are extracted.

3.2 Object Discovery for a Single Scan

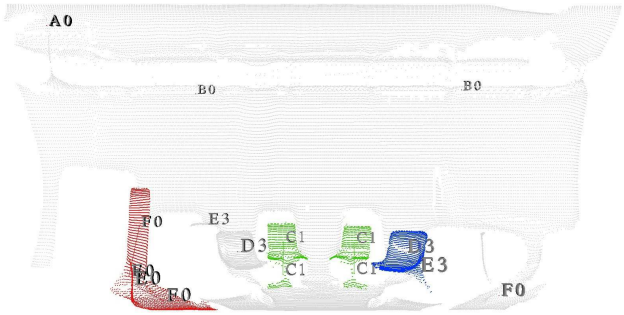


Fig. 3: Result of object discovery of the scene shown in Fig. 2. Discovered objects are colored according to their class labels. Letters indicate the parts types and numbers indicate object classes. Notice that not all potential object parts are accepted as object parts.

Once we extract potential object parts $\mathcal{S}$, next step is to reason on them to discover objects. The object discovery step on single scan is based on our previous work [18]. The underlying idea behind our object discovery algorithm is that object parts which belong to the same object are frequently observed together, and hence by observing which parts occur together frequently, we can deduce object class label for these parts. Using this idea, a brief summary of the algorithm is as follows. Given the potential object parts $\mathcal{S}$, we extract a feature vector $\mathbf{f}_i$ for each

potential object part $\mathbf{s}_i$. The feature vector $\mathbf{f}_i$ is composed of spin images [9], shape distributions [13], and shape factors [19]. To determine which set of potential object parts originate from the same *parts type* $\mathcal{F}_i$, we cluster these parts in feature space using affinity propagation [6]. Affinity propagation implicitly estimates the number of clusters C, resulting in clusters $\mathcal{F}_1,\ldots,\mathcal{F}_C$. These clusters define the discovered object parts types.

Clustering in feature space provides parts types, but it does not define which parts belong to the same object *instance*. To obtain the object instances, we perform another clustering on the potential object parts $\mathcal{S}$ but this time in geometric space. As object parts for the same object instance are physically close, clustering in geometric space enables us to group together potential object parts which belong to the same object instance. The geometric clustering algorithm connects every pair of potential objects whose centers are closer than a threshold ϑ_g, and this results in a collection of connected components. The number of connected components K define the maximum number of object classes present in the scene, and each cluster $\mathcal{G}_i$ of the resulting clusters $\mathcal{G}_1,\ldots,\mathcal{G}_K$ correspond to an object instance.

Given parts types $\mathcal{F}_1,\ldots,\mathcal{F}_C$ and object classes $\mathcal{G}_1,\ldots\mathcal{G}_K$, next step is to assign a class label $\mathcal{G}_i$ to each potential object part $\mathbf{s}_i$. We determine the assignments by reasoning on the labels at two levels. First, on a more abstract level, the statistical dependency of class labels $\mathcal{G}_1,\ldots,\mathcal{G}_K$ across different parts types $\mathcal{F}_1,\ldots,\mathcal{F}_C$ is encoded in a Conditional Random Field (CRF) [10] named *parts graph*. Parts graph exploits the fact that object parts that co-occur frequently in the same object instance are more likely to belong to the same object class. For example, back rest and seat, both of which belong to a chair, are frequently found together while seat and shelf, which belong to different objects, are not. The second level of reasoning propagates parts types to object class relationship onto a finer level by combining the class labels obtained from the parts graph with the local contexual information from actual scenes. This is encoded using another CRF called *scene graph*. Performing inference on the parts graph provides the most likely object class label $\mathcal{G}_i$ per parts type $\mathcal{F}_i$ while inference on the scene graph leads to the object class label $\mathcal{G}_i$ per object part $\mathbf{s}_i$. Once for all object instances, all their parts are labeled with the most likely object class label, we accept those object instances which contains at least two parts with the same class label as discovered objects $O_1,\ldots,O_N$. Fig. 3 shows an example of the outcome of the discovery algorithm.

4 Object Categorization

Object discovery algorithm of the previous section is able to find object classes for which at least two instances occur in a given scene. It uses appearance and geometry, i.e., similarity of features and structures, to find several instances of objects that are most likely to define a class in one given scene. In this paper, we go one step further and try to find object *categories*, i.e., object classes that are consistent across a sequence of input scenes. This, however, is not straightforward. As the ob-

Fig. 4: Objects found in two different scenes. Segments of the same local object label have the same color locally.

ject discovery process is entirely unsupervised, the resulting local class labels are not unique over a given number of input scans. This means that an object class might be associated with a class label $\mathcal{G}_1$ when one scene is observed, but the same object class might have a different class label $\mathcal{G}_2$ if observed in a different scene. An example of this is shown in Fig. 4. To identify object instances of the same class from different scenes, we need to solve the *data association problem*. Unfortunately, this problem is intractable in general as it involves a correspondence check between every pair of object classes which are found in different scenes. One simple way to address this correspondence problem is to join all scenes into one big scene and run the discovery algorithm on the big scene. This approach, however, has two major drawbacks: first, the number of connected components K in this big scene would be very large. This heavily increases the computation time of the algorithm and decreases its detection performance because it fails to sufficiently restrict the number of potential object classes. And second, it limits the possibility of running the object discovery in an online framework, which is one major goal of this work. The reason here is that the parts graph would need to be re-built every time a new scene is observed, which decreases the efficiency of the algorithm.

This work addresses the data association problem by introducing a third level of reasoning named *class graph*. The key idea behind the class graph is to find a mapping from local class labels to global category labels. Unlike the parts graph and the scene graph, the class graph models the statistical dependencies between labels of object class instances rather than object parts. Details of the class graph is explained in Sec. 4.2. Next section describes object feature vector for representation of object instances, which are the building blocks of class graph.

4.1 Object Representation

Object feature vector enables a compact representation of object instances. This work employs object feature vector **o** which captures object instance's appearance and shape. The object feature vector **o** is composed of a histogram **h** of visual word

occurrences and a shape vector **v**. The histogram **h** captures object appearance while the shape vector **v** captures object volume. To compute the histograms, we take the *bag of words* approach and represent an object as a collection of visual words. Bag of words requires visual vocabulary to be defined, and we determine the visual vocabulary by clustering the object parts feature vector **f** of all discovered objects. Each cluster $\mathcal{F}_i^*$ is a word in the visual vocabulary $\mathcal{F}_1^*,\ldots,\mathcal{F}_{C^*}^*$, and the total number of words in the vocabulary C^* is equal to the number of clusters C^*. With the visual vocabulary, representing an object as a histogram simplifies to counting the number of occurrences of each visual word in the object. In traditional bag of words approaches, every feature makes a contribution to the bin corresponding to the visual word that best represents the feature. Such approaches, however, do not take into account the uncertainty inherent in the assignment process. Hence, in our work, each object part feature vector **f** contributes to all bins of the corresponding histogram **h**, where the contribution to a bin is determined by the probability $p(\mathbf{w}_i|\mathbf{f})$ of the feature vector **f** belonging to the visual word $\mathbf{w}_i$. We compute this probability by nearest-neighbor.

In addition to a histogram **h**, object feature vector **o** contains a shape vector **v**, which represents object's physical properties. The shape vector **v** is composed of three elements – size in horizontal direction, size in vertical direction, and object's location in vertical direction. The horizontal and vertical spans provide the bounding volume in which the object resides. The vertical location gives an estimate on where the object is likely to be found.

4.2 Class Graph

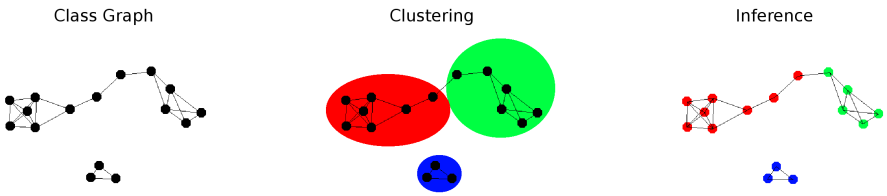


Fig. 5: Categorization by class graph. Local class labels, represented as mean histograms, are the nodes of the graph, and the links between two similar nodes form the edges. Clustering the local class labels provides the initial mapping from local class labels to global category labels. Running inference on the class graph provides a distribution of category labels for each local label. These distributions are then used to determine the category label for each discovered object.

Once the object feature vectors $\mathbf{o}_1,\ldots,\mathbf{o}_{N^*}$ are computed for all discovered objects $O_1,\ldots,O_{N^*}$, we determine the mapping from local class labels $\mathcal{G}_1,\ldots,\mathcal{G}_M$ to global category labels $\mathcal{G}^*_1,\ldots,\mathcal{G}^*_{K^*}$ using *class graph* $\mathfrak{C}$. Class graph $\mathfrak{C}$ consists of the node set $\mathcal{V}_{\bar{\mathbf{o}}} = \{\bar{\mathbf{o}}_1,\ldots,\bar{\mathbf{o}}_M\}$ and the edge set $\mathcal{E}_{\bar{\mathbf{o}}} = \{(\bar{\mathbf{o}}_i,\bar{\mathbf{o}}_j) \mid D(\bar{\mathbf{o}}_i,\bar{\mathbf{o}}_j) < \vartheta_{\bar{\mathbf{o}}}\}$. The nodes are the local class labels $\mathcal{G}_1,\ldots,\mathcal{G}_M$ represented as mean object feature vectors $\bar{\mathbf{o}}_1,\ldots,\bar{\mathbf{o}}_M$, and the edges connect similar local class labels, where the similarity between two local labels is the distance between their mean object feature vectors. The threshold for object similarity $\vartheta_{\bar{\mathbf{o}}}$ is set to 0.5.

To assign global category labels $\mathcal{G}^*_1,\ldots,\mathcal{G}^*_{K^*}$ to local class labels $\mathcal{G}_1,\ldots,\mathcal{G}_M$, we need to find the number of global categories K^*. As mentioned earlier, Affinity Propagation (AP) implicitly determines the number of clusters, and therefore, we cluster the mean object feature vectors $\bar{\mathbf{o}}_1,\ldots,\bar{\mathbf{o}}_M$ by AP clustering. The number of clusters K^* resulting from AP clustering is the maximum number of global categories, and the clusters $\mathcal{G}^*_1,\ldots,\mathcal{G}^*_{K^*}$ are the initial global category labels for the local class labels $\mathcal{G}_1,\ldots,\mathcal{G}_M$. Smoothing this initial mapping determines the final mapping from local class labels to global category labels. Fig. 5 shows the overall steps of categorization by class graph.

4.3 Smoothing

Class graph $\mathfrak{C}$ captures the dependency among the local class labels $\mathcal{G}_1,\ldots,\mathcal{G}_M$, but it does not assign a category label $\mathcal{G}^*_i$ to each local label $\mathcal{G}_i$. To determine the category labels, we apply probabilistic reasoning. We treat the nodes of the graph as random variables and the edges between adjacent nodes as conditionally dependent. That is, the global category label $\mathcal{G}^*_i$ of a local class label $\mathcal{G}_i$ depends not only on the local evidence $\bar{\mathbf{o}}_i$ but also on the class labels $\mathcal{G}^*_j$ of all neighboring labels $\mathcal{G}_j$. For example, if the local class label $\mathcal{G}_i$ is strongly of category $\mathcal{G}^*_i$,, based on its evidence $\bar{\mathbf{o}}_i$, then it can propagate its category label $\mathcal{G}^*_i$ to its neighbors $\mathcal{G}_j$. On the other hand, if its category label is weak, then its category label $\mathcal{G}^*_i$ can be flipped to the category label $\mathcal{G}^*_j$ of its neighbors. This process penalizes sudden changes of category labels, producing a smoothed graph. We perform the smoothing again using a Conditional Random Field (CRF).

Our CRF models the conditional distribution

$$p(g \mid \bar{\mathbf{o}}) = \frac{1}{Z(\bar{\mathbf{o}})} \prod_{i \in \mathcal{V}_{\bar{o}}} \varphi(\bar{\mathbf{o}}_i, g_i) \prod_{(i,j) \in \mathcal{E}_{\bar{o}}} \psi(\bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j, g_i, g_j), \tag{6}$$

where $Z(\bar{\mathbf{o}}) = \sum_{g'} \prod_{i \in \mathcal{V}_{\bar{o}}} \varphi(\bar{\mathbf{o}}_i, g'_i) \prod_{(i,j) \in \mathcal{E}_{\bar{o}}} \psi(\bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j, g'_i, g'_j)$ is the *partition function*; $\mathcal{V}_{\bar{o}}$ are the local classes; and $\mathcal{E}_{\bar{o}}$ are the edges between the local classes. Our formulation of the CRF is slightly different from the conventional approaches in that our feature similarity function f_n of the node potential $\log \varphi(\bar{\mathbf{o}}_i, g_i) = w_n \cdot f_n(\bar{\mathbf{o}}_i, g_i)$ is the conditional probability $p(g_i \mid \bar{\mathbf{o}}_i)$. Likewise, the feature similarity function f_e of the edge potential $\log \psi(\bar{\mathbf{o}}_i, g_i) = w_e \cdot f_e(\bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j, g_i, g_j)$ is also defined as a conditional

probability $p(g_i, g_j \mid \bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j)$. The feature functions f_n and f_e hence range between 0 and 1, simplifying the weighting between node and edge potentials to scalars. In supervised learning with CRFs, node weight w_n and edge weight w_e are learned from training data. In this unsupervised work, however, we cannot learn these values as there is no training data available. We therefore determine node weight w_n and edge weight w_e manually using an appropriate evaluation measure on a validation set. Fig. 8 in Sec. 5 shows the effect of setting different combinations of node weight w_n and edge weight w_e.

As mentioned in Sec. 4.2, the object feature vector clustering provides the total number of global object categories C^* and the initial mapping from local class labels $\mathcal{G}_1, \ldots, \mathcal{G}_M$ to global category labels $\mathcal{G}^*_1, \ldots, \mathcal{G}^*_{K^*}$. Using the clusters, we can model the feature similarity function $f_n = p(g_i \mid \bar{\mathbf{o}}_i)$ of node potential $\varphi(\bar{\mathbf{o}}_i, g_i)$ as

$$p(g_i \mid \bar{\mathbf{o}}_i) = \frac{p(\bar{\mathbf{o}}_i \mid g_i) p(g_i)}{\sum_{g'} p(\bar{\mathbf{o}}_i \mid g') p(g')} \tag{7}$$

where $p(\bar{\mathbf{o}}_i \mid g_i) = p(\bar{\mathbf{h}}_i \mid g_i^{\bar{\mathbf{h}}}) p(\bar{\mathbf{v}}_i \mid g_i^{\bar{\mathbf{v}}}) = \exp(- \parallel \bar{\mathbf{h}}_i - \bar{\mathbf{h}}^{g_i} \parallel) \exp(- \parallel \bar{\mathbf{v}}_i - \bar{\mathbf{v}}^{g_i} \parallel)$ and $p(g_i) = 1 - \frac{1}{|g_i|+1}$. $p(\bar{\mathbf{o}}_i \mid g_i)$ measures how well $\bar{\mathbf{o}}_i$ fits to the cluster center g_i, and the global category prior $p(g_i)$ reflects how likely the category exists. A cluster with more members are more likely to be a true object category than a cluster with fewer members, and hence $p(g_i)$ is proportional to the size $\mid g_i \mid$ of the category.

We define the edge feature as

$$p(g_i, g_j \mid \bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j) = p(g_i \mid \bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j) p(g_j \mid \bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j), \tag{8}$$

where $p(g_i \mid \bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j) = p(g_i \mid \bar{\mathbf{o}}_{ij})$ and $p(g_j \mid \bar{\mathbf{o}}_i, \bar{\mathbf{o}}_j) = p(g_j \mid \bar{\mathbf{o}}_{ij})$ are estimated by a mean object feauter vector $\bar{\mathbf{o}}_{ij}$. The probabilities $p(g_i \mid \bar{\mathbf{o}}_{ij})$ and $p(g_j \mid \bar{\mathbf{o}}_{ij})$ are computed by the nearest-neighbor.

To infer the most likely labels for the nodes of the class graph $\mathfrak{C}$, we use max-product loopy belief propagation. This approximate alogrithm returns the labels $\mathcal{G}^*_i$ which maximizes the conditional probability of Eq. 6. For the message passing, we take the generalized Potts model approach as commonly done and incorporate the edges in the inference only when g_i and g_j are equal. This results in the propagation of the belief only between equally-labeled nodes. The inference step continues until convergence and provides the distribution of global category labels $\mathcal{G}^*_1, \ldots, \mathcal{G}^*_{K^*}$ for every local class label $\mathcal{G}_i$.

To find the category label $\mathcal{G}^*$ for each discovered object O, we compute the category which maximizes the assignment probability

$$p(g \mid \mathbf{o}) = \sum_{\bar{\mathbf{o}}'} p(g \mid \bar{\mathbf{o}}') p(\bar{\mathbf{o}}' \mid \mathbf{o}). \tag{9}$$

The probability of the category for a given local label $p(g \mid \bar{\mathbf{o}}')$ can be read directly from the class graph $\mathfrak{C}$, and the probability of the local object class given an object $p(\bar{\mathbf{o}}' \mid \mathbf{o}) = \exp(- \parallel \bar{\mathbf{o}} - \mathbf{o} \parallel)$ is computed as the object's similarity to the class mean. Discovered objects are accepted as objects when the probability of its most likely

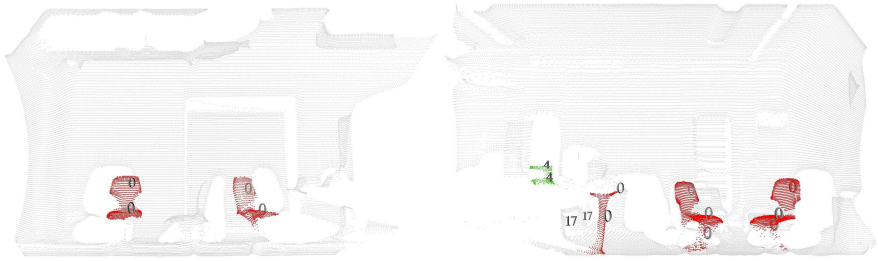


Fig. 6: Objects found in two different scenes. Segments of the same object label have the same color.

category label is greater than 0.5. Fig. 6 shows the results of categorization of the two scenes shown in Fig. 4.

5 Results

In this section, we present the results of running the algorithm on scans from real world scenes. The data set was collected using a nodding SICK laser with a width of 100 degrees and a height of 60 degrees. Each set was captured at the horizontal resolution of 0.25 degrees and the vertical resolution of 15 degrees a second. All scenes were static. The test set was a set of 60 scans from four offices. In total, these data sets contained 208 objects, including chairs, couches, poster boards, trash bins, and room dividers.

Fig. 7: The results of object discovery with (left) and without (right) saliency computation. All connected segments are considered objects for categorization. Objects are colored by their local class label.

We first tested the effect of including saliency in the discovery step. Fig. 7 qualitatively shows the difference in object discovery with and without saliency compu-

tation. Including saliency improves the precision [1] of discovery from 44% to 84% while decreasing recall from 83% to 74%. That is, while including the saliency step does eliminate some true objects, it is much more effective at eliminating none objects than the same algorithm without the saliency step.

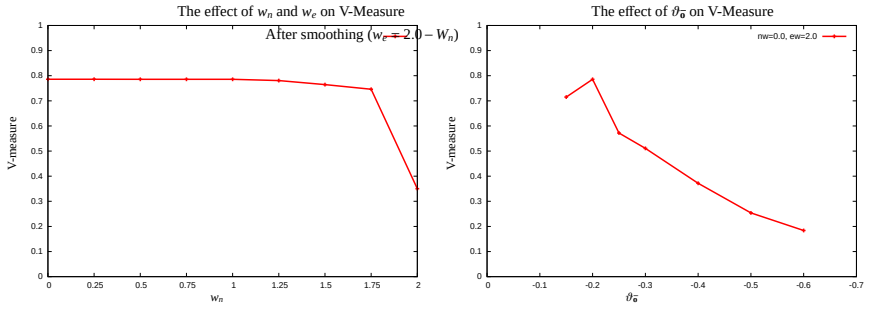


Fig. 8: Evaluation of our categorization step using V-measure. Left graph shows the effect of node and edge weights on v-measure. Right graph shows the effect of the object distance threshold on v-measure.

Quantitatively, we computed V-measure [15] of our algorithm. V-Measure is a conditional entropy-based external cluster evaluation measure which captures the cluster quality by homogeneity and completeness of clusters. It is defined as

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta * h) + c}, \tag{10}$$

where h captures homogeneity, c completeness, and β the weighting between homogeneity and completeness. A perfectly homogeneous solution has $h = 1$, and a perfectly complete solution has $c = 1$. Fig. 8 shows the quality of clustering with varying node and edge weights and the effect of object distance threshold on the quality of clustering. Left graph indicates that the results of our algorithm is robust to the change of node and edge weights, but smoothing improves the overall results over pure clustering. Right graph shows that the quality of clusters depends on the object distance threshold $\vartheta_{\bar{\mathbf{o}}}$, which indicates that the initial clustering result influences the final categorization quality.

Fig. 9 shows precision and recall[2] of the algorithm for varying object distance threshold $\vartheta_{\bar{\mathbf{o}}}$. Not suprisingly, precision drops and recall increases as the threshold increases. This is because higher threshold results in fewer categories, which in turn means more of the discovered objects are accepted as categorized objects.

[1] A discovered object is considered true positive if it originates from a real object and false positive if it is not a real object. False negative count is when a real object is not discovered.

[2] In computing precision and recall, we did not take into consideration the correctness of the category labels. Any real object that got categorized was considered true regardless of its label.

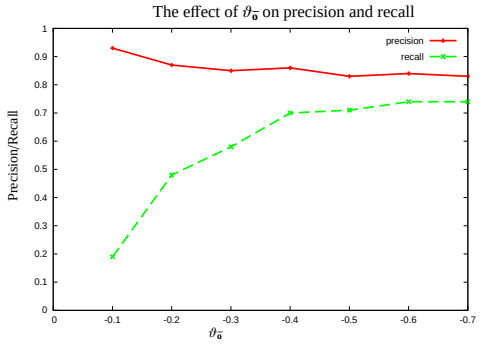


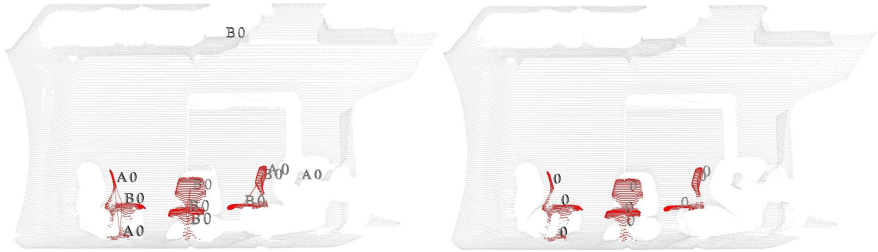
Fig. 9: Effect of the object distance threshold on precision and recall.

Fig. 10 shows qualitative results. Left images are the results of performing object discovery per each scan, and right images are the corresponding images after categorization. Discovered objects are colored according to their local class label, i.e., with respect to other objects within a single scan, while categorized objects are colored according to their global category label, i.e., with respect to all other objects of the data set. The categorization step is able to assign the same global category labels to objects with different local class labels as shown in Fig. 10b while assigning different global category labels to objects with the same local label as shown in Fig. 10d. In addition, the chairs found in different scene are correctly labeled to be the same type as shown in Fig. 10a, 10b, 10d.

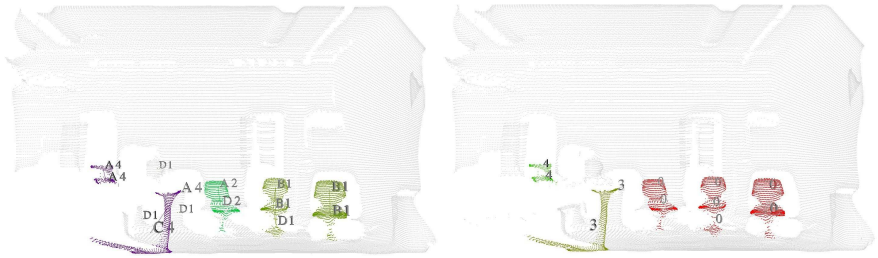
6 Conclusion and Outlook

We presented a seamless approach to discover and categorize objects in 3D environment without supervision. The key idea is to categorize the objects discovered in various scenes without requiring a presegmented image or the number of classes. Our approach considers objects to be composed of parts and reasons on each part's membership to an object class. After objects are discovered in each scan, we associate these local object labels by building a class graph and inferring on it. We demonstrated our capability of discovering and categorizing objects on real data and performance improvement class graph smoothing brings over pure clustering.

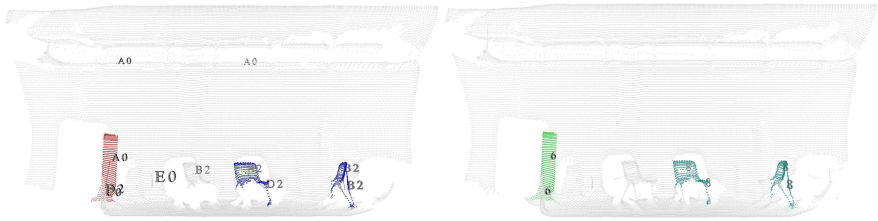
Our approach has several avenues for future work. First, we can use the results of categorization for object recognition. Once the robot has discovered enough instances of an object category, it can use the knowledge to detect and recognize objects, much the same way many supervised algorithms work. Our algorithm simplifies creating training data to converting robotic class representation to human representation. Another direction for future work is on-line learning. While the proposed approach allows the robot to reason on knowledge gained over time, the knowledge



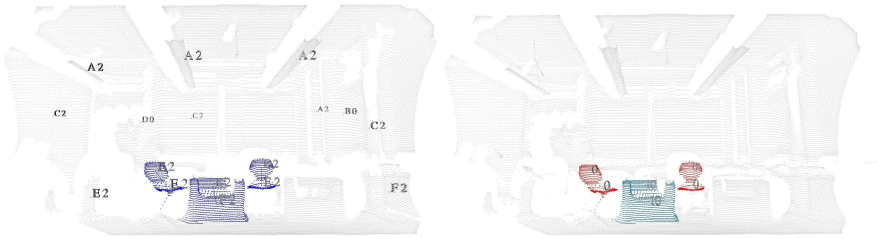
(a) Room 1



(b) Room 2



(c) Room 3



(d) Room 4

Fig. 10: Results of category discovery. Left images contain objects discovered through the object discovery process, and right images are the same objects after categorization. Objects in the left images are colored according to their local class labels while objects in the right images are colored by their global category labels. Notice that the categorization step can correct incorrect classifications of the discovery step.

is updated in batch. This limits the availability of new information until enough data is collected for the batch processing. A robot, which can process incoming data and update its knowledge on-line, can utilize the new information immediately and adapt to changing environment. Extending our work to handle categorization on-line will thus make unsupervised discovery and categorization more useful for robotics.

References

1. Bokeloh M, Berner A, Wand M, Seidel HP, and Schilling A (2009) Symmetry Detection Using Feature Lines. Computer Graphics Forum (Eurographics) 28.2:697–706(10)
2. Bagon S, Brostovski O, Galun M, and Irani M (2010) Detecting and Sketching the Common. In: IEEE Computer Vision and Pattern Recognition
3. Cho M, Shin Y, and Lee K (2010) Unsupervised Detection and Segmentation of Identical Objects. In: IEEE Computer Vision and Pattern Recognition
4. Csurka G, Bray C, Dance C, and Fan L (2004) Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV
5. Endres F, Plagemann C, Stachniss C, and Burgard W (2009) Unsupervised discovery of object classes from range data using latent Dirichlet allocation. In: Proc. of Robotics: Science and Systems
6. Frey BJ and Dueck D (2007) Clustering by passing messages between data points. Science 315.5814:972–976
7. Frintrop S, Nuechter A, Surmann H, and Hertzberg J (2004) Saliency-based Object Recognition in 3D data. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems
8. Itti L, Kock C, and Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on Pattern Analysis and Machine Learning 20.11:1254–1259
9. Johnson A E and Hebert M (1999) Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. IEEE Trans. on Pattern Analysis and Machine Learning 21.5:433-449
10. Lafferty J, McCallum A, and Pereira F (2001) Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proc. of Int. Conf. on Machine Learning
11. Leung T and Malik J (1999) Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. In. Int. Conf. on Computer Vision
12. Ng A, Jordan M, and Weiss Y (2002) On Spectral Clustering: Analysis and an Algorithm. In: Adv. in Neural Information Processing Systems
13. Osada R, Funkhouser T, Chazelle B, and Dobkin D (2002) Shape Distributions. ACM Trans. on Graphics 21.4:807–832
14. Ruhnke M, Steder B, Grisetti G, and Burgard W (2009) Unsupervised Learning of 3D Object Models from Partial Views. In: IEEE Int. Conf. Robotics and Automation, Kobe, Japan
15. Rosenberg A and Hirschberg J (2007) V-Measure: A Conditional Entropy-based External Cluster Evaluation Measure. In: Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning
16. Sivic J, Russell B, Efros A, Zisserman A, and Freeman W (2005) Discovering Object Categories in Image Collections. In: Proc. of the Int. Conf. on Computer Vision
17. Spinello L, Triebel R, Vasquez D, Arras K, and Siegwart R (2010) Exploiting Repetitive Object Patterns for Model Compression and Completion. In: European Conf. on Computer Vision
18. Triebel R, Shin J, and Siegwart R (2010) Segmentation and Unsupervised Part-based Discovery of Repetitive Objects. In: Proc. of Robotics: Science and Systems
19. Westin C, Peled S, Gudbjartsson H, Kikinis R, and Jolesz F (1997) Geometrical Diffusion Measures for MRI from Tensor Basis Analysis. In: ISMRM '97