

Unsupervised Discovery of Object Classes in 3D Outdoor Scenarios

Frank Moosmann
Institute of Measurement and Control
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
frank.moosmann@kit.edu

Miro Sauerland
Institute for Anthropomatics
Vision and Fusion Laboratory
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
miro.sauerland@kit.edu

Abstract

Designing object models for a robot's detection-system can be very time-consuming since many object classes exist. This paper presents an approach that automatically infers object classes from recorded 3D data and collects training examples. A special focus is put on difficult unstructured outdoor scenarios with object classes ranging from cars over trees to buildings. In contrast to many existing works, it is not assumed that perfect segmentation of the scene is possible. Instead, a novel hierarchical segmentation method is proposed that works together with a novel inference strategy to infer object classes.

1. Introduction

Robots need a certain amount of knowledge in order to be able to interact with the environment in a safe manner. The type and amount of knowledge thereby depends on the kind of environment. The more complex the environment gets the more complex the knowledge-base grows. For controlled factory settings on the one hand predefined trajectories are often sufficient. Uncontrolled outdoor scenarios on the other hand usually require knowledge about objects, their physical and semantic relationships, a road-map, and even more. Whereas predefined trajectories can be specified manually, the vast amount of knowledge needed for highly unstructured environments is impossible to completely design by hand.

This work targets at the problem of generating knowledge about object classes. The term "object class" thereby refers to an object category (*e.g. cars, trees*), whereas "object" is seen as one specific instance (*e.g. the pine tree in front of my house*). Input to the knowledge generation algorithm is in this work a raw 3D point cloud. In contrast to images, point clouds provide full 3D geometry where scale ambiguity is not a problem. This allows to identify objects

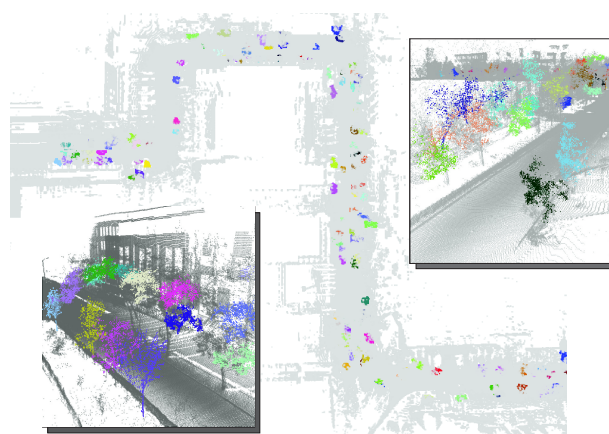


Figure 1: Bird eye's view on the dataset with marked instances of one discovered object class (*trees*).

based on their geometry alone and to ignore highly varying intensity information. We focus on unlabeled data, which is easy to obtain, hence no a-priori knowledge exists over possible object classes. Thus, the algorithm is only capable of discovering geometric structures that occur frequently. All similar structures then represent instances of the same object class. This discovery and grouping of structures is the output of the algorithm. In the ideal case, a human could afterwards simply attach semantic labels (as *e.g. cars, trees*) to the discovered groups. This knowledge can be used for many applications (like online object detection, scene understanding, object grasping *etc.*) although this is not within the focus of this work.

The main problem during object class discovery is that numerous classes exist and that the geometry of object instances might vary heavily within the respective object class. Some objects might even change appearance over time, like *e.g. walking pedestrians*. Furthermore, objects are usually not fully visible. All this makes it hard to find a

suitable representation and to automatically learn these representations from data.

Despite these challenges, some researchers already tackled this problem: Ruhnke *et al.* [8] take single range images from indoor scans and subtract the floor, walls, and the ceiling. The remaining point clouds are then grouped by comparing all candidates, and finally verified by registering them with ICP against a virtually merged range image. This method was improved in [3] by replacing the grouping and verification with a probabilistic technique also used in text analysis. The main disadvantage of these two methods is their restriction to scenes where objects can be easily segmented by removing walls *etc.* Shin *et al.* [11] do not rely on such simple segmentation but build a neighborhood-graph on segmented planes instead. Subsequently, they search segment-combinations that occur multiple times and verify them by ICP. However, the segmentation is still assumed to be perfect. Moreover, the method works only well for objects consisting of planes ([13]). A different approach was presented in [10], where a grammar of cuboids is used. But again, the problem of a required perfect segmentation remains. This is not the case for [9], where a hierarchical segmentation is used. However, this approach is restricted to completely observed objects.

All these approaches work on 3D indoor-data which usually simplifies segmentation and the type of object classes. One of the first works on 3D outdoor data is [6]. But since a user must click on objects (which then initiates segmentation, feature calculation and storage of the model) this approach cannot be seen as fully unsupervised.

A different, but related research area focuses upon images instead of 3D data. As an image is not a metric representation of the world scale ambiguity must be taken care of. Additional challenges as varying lighting conditions and viewing perspectives complicate the task even further. A good overview of unsupervised learning approaches for images can be found in [14]. Interesting are in particular the works of Todorovic *et al.* [12, 1]. They do not rely on a single error-free segmentation, but employ a segmentation tree which is evaluated by the learning algorithm. Rabinovich *et al.* [7] on the contrary consider several "stable" segmentations in parallel.

This work focuses on 3D data from outdoor scenarios. To the best of our knowledge, this is the first fully-unsupervised approach for learning object classes from this kind of data. To overcome imperfect segmentation, a novel hierarchical segmentation method is introduced. A specific combination of features combined with two novel inference strategies thereafter conduct the final object class discovery.

The work is organized as follows: The following section describes the proposed approach in detail. Sec. 3 shows preliminary results on collected data. Sec. 4 concludes and gives an outlook to future research.

2. Proposed Method

As mentioned in Sec. 1, the proposed method is based on the assumption that multiple observed objects, characterized by a similar geometric structure, can be interpreted as instances of an object class. The final goal is to identify these classes from unlabelled data. Critical steps, putting this assumption into action, are isolating objects from the complete scene and defining a similarity measurement, able to handle arbitrary objects. We propose to solve these challenges with the algorithm illustrated in Fig. 2.

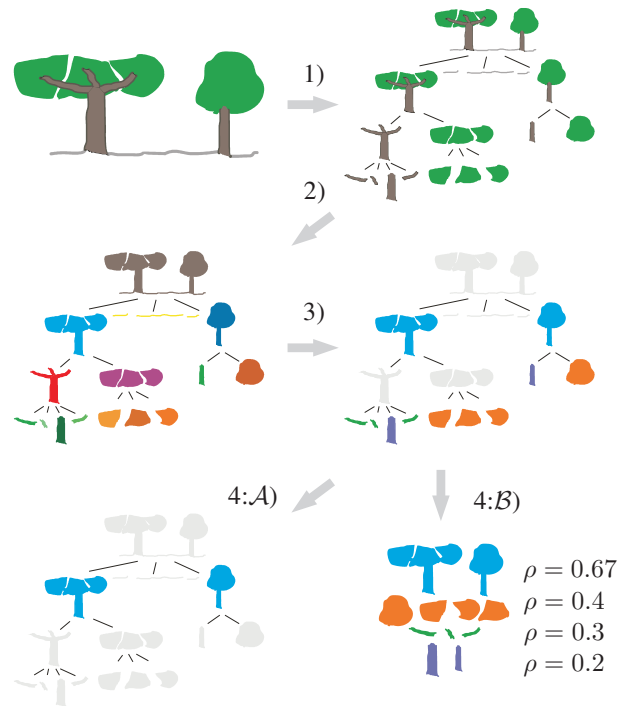


Figure 2: Sketch of the proposed algorithm: 1) decomposing the scene hierarchically. 2) calculating features for each segment. 3) finding similar segments by clustering. 4:A/B) selecting clusters as object classes.

Input is a 3D point cloud (usually obtained with a laser scanner) containing several instances of various object classes. In the first step (Sec. 2.1), objects are segmented by a Region Growing algorithm. A specific property of the proposed approach is that not a single *perfect* segmentation is considered, but a set of segmentation parameters is chosen and the results are organized in a hierarchical segmentation tree (ST). The corresponding 3D points of each node are thereby contained in the set of 3D points of the parent node. In the second step (Sec. 2.2), one feature vector is calculated for each segment node which enables comparison of segments (depicted by different colors in Fig. 2). These features are used within the third step (Sec. 2.3), where simi-

lar feature vectors are clustered. This forms groups of segments with similar geometric properties. In the final step (Sec. 2.4), these clusters are analyzed in order to define object classes. Two variants, strategy \mathcal{A} and \mathcal{B} , were developed to perform this inference. All of these steps are detailed in the following.

2.1. Segmentation

Segmentation starts from a set of 3D point measurements $\{\mathbf{p}_i\}$ derived by sampling from the surfaces of an outdoor scene. Assuming that the Nyquist-Shannon sampling theorem holds, local surface geometry at a point \mathbf{p}_i can be reconstructed by looking at the distribution of neighboring point measurements N_i . A local plane is fitted to $N_i \cup \mathbf{p}_i$ using principal component analysis (PCA) as described in [4]. The plane is represented by its normal vector \mathbf{n}_i .

The same neighborhood is then used for segmentation. For each point $\mathbf{p}_j \in N_i$ a cost function $c(i, j)$ is evaluated. \mathbf{p}_i and \mathbf{p}_j belong to the same segment iff $c(i, j) < \theta$. Basing segmentation on pair-wise connections enables to implement it as efficient Region-Growing (see *e.g.* [5]).

In this work the cost function $c(i, j) \mapsto [0..1] := \frac{c_1 + c_2 + c_3}{3}$ is a linear combination of three cost functions which are defined on the basis of the 3D coordinates $\mathbf{p}_{i/j}$, the distance vector $\mathbf{d}_{ij} = \mathbf{p}_i - \mathbf{p}_j = -\mathbf{d}_{ji}$, and the normal vectors $\mathbf{n}_{i/j}$:

$$c_1(i, j) = \max \left[0, \frac{\mathbf{n}_i^\top \mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|}, \frac{\mathbf{n}_j^\top \mathbf{d}_{ji}}{\|\mathbf{d}_{ji}\|} \right] \quad (1)$$

$$c_2(i, j) = \min \left[1, 1 - \mathbf{n}_i^\top \mathbf{n}_j \right] \quad (2)$$

$$c_3(i, j) = \min \left[1, \frac{1}{\hat{d}} |\delta_i - \delta_j| \right] \quad (3)$$

$c_1(i, j)$ rates the convexity and penalizes concave normal arrangement between \mathbf{n}_i and \mathbf{n}_j (for further information refer to [5]). $c_2(i, j)$ rates the relative normal direction, so similar normal vectors tend to be grouped together. $c_3(i, j)$ is sensitive to changes in point densities, where $\delta_i = \frac{1}{|N_i|} \sum_{\mathbf{p}_k \in N_i} \|\mathbf{d}_{ik}\|$ and \hat{d} is the average δ_i across the complete scene.

Unfortunately, in 3D point clouds there often exist regions where the formerly stated assumption about the sampling theorem does not hold. Thus, there are two types of regions:

1. Over-sampled regions (*e.g.* building facades) : a subset of 3D points, describing an object that can be entirely reconstructed according to the sampling theorem.
2. Under-sampled regions (*e.g.* vegetation) : the density of 3D points can't resolve the details of the corresponding original object, spatial information is lost.

Segmentation using the previously introduced $c(i, j)$ only works well in regions of over-sampling. To adapt the segmentation to both cases, we start by determining for each region in the data, if the underlying real-world object was over- or under-sampled. In principle this is impossible, because no knowledge about the real-world object is available. One way out is to look at the arrangement of the points in N_i . A locally planar arrangement indicates low spatial frequencies which leads to over-sampling. For this reason the planarity factor $s_i = \frac{|\lambda_2 - \lambda_3|}{|\lambda_1|}$ is determined by re-using the eigenvalues $\lambda_{1,2,3}$ of the PCA from the normal calculation on N_i . A plane is characterized by high $\lambda_{1,2}$ and a low λ_3 which leads to $s_i \approx 1$. The less plane-like the region around \mathbf{p}_i is, the lower the value of s_i gets.

Having s_i , it is possible to develop an additional segmentation method for under-sampled regions. Surprisingly, the simple *negation* of $c(i, j)$ works well: $1 - c(i, j)$.

So given two points, \mathbf{p}_i and \mathbf{p}_j , four cases are possible: both are over-sampled, both are under-sampled, i is over- and j is under-sampled, and vice versa. We linearly blend these cases using the mean planarity factor $\bar{s}_i = \sum_{N_i} s_i$.

$$\begin{aligned} c_{\text{blend}}(i, j) = & c(i, j) \cdot (\bar{s}_i + \bar{s}_j) \\ & + (1 - c(i, j)) \cdot (2 - \bar{s}_i - \bar{s}_j) \\ & + 1 \cdot (1 - \bar{s}_i + \bar{s}_j) \\ & + 1 \cdot (1 - \bar{s}_j + \bar{s}_i) \end{aligned} \quad (4)$$

The last two situations are thereby penalized with high costs ($= 1$) as a border between over- and under-sampled regions is a desired border for segmentation. Fig. 4 shows a result of the proposed segmentation method for various thresholds.

As mentioned in Sec. 1, many works in the field of unsupervised discovery of object classes assume segmentation to be perfect. This limitation can be overcome by performing the segmentation several times with varying cost thresholds. In our case, a set of thresholds $\Theta = \{\theta_1, \dots, \theta_k\}$ can be used. If $\theta_i > \theta_{i+1}$ is valid for all i , this results in recursive splits of the data set, which allows to arrange the results in a segmentation tree (ST), see Fig. 3. The depth of the ST is determined by the number of segmentation thresholds (in our case 13 equidistant thresholds were chosen). A ST not only returns alternative segmentations, but even allows to extract additional information about the assembly of objects (see [1]). However, this additional information is only partly used within this work.

2.2. Feature space

For similarity calculations, all nodes of the ST are transformed into feature space. Each feature vector $\mathbf{f}_i = [f_1, \dots, f_{21}]^\top$ describes the geometry of the corresponding set of 3D points, represented by a node in the ST. Except for

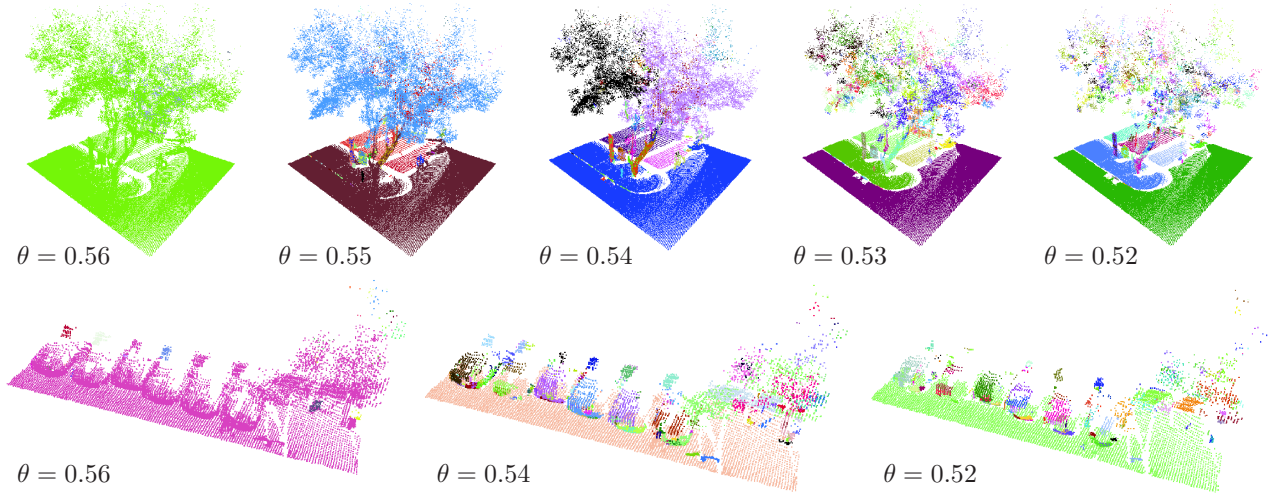


Figure 4: Result of threshold variation on segmentation (colors indicate segments). *Upper row:* big tree. *Lower row:* parked cars and trees.

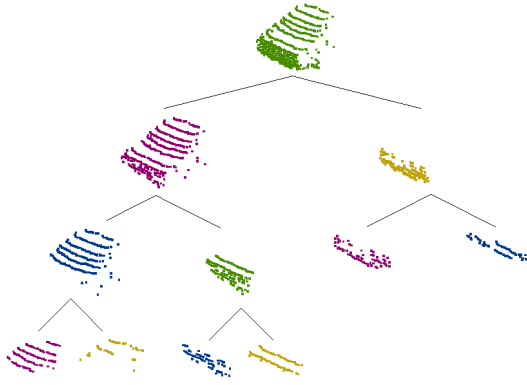


Figure 3: Resulting segmentation (sub-)tree of a car.

f_1 , each feature is defined scale invariant, rotation invariant around the vertical axis and is standardized to $[0..1]$.

Absolute scale is a very descriptive feature, but there are various ways to measure the scale of a point cloud, especially when it comes to the two cases of over- and under-sampling. In the first case, the 3D points describe a smooth surface. Under-sampling instead can cause volume-like point distributions. Counting occupied volume elements V (cells of a 3D grid) is a suitable measurement for both cases. With the Manhattan-Distance as similarity measurement in mind, the scale feature is designed as $f_1 = f_{scale} := \log_a(|V|)$. Hence, scaling an object by factor a results in distance 1 in feature space, independent of the absolute size.

The other features characterize enlargement (4 features),

normal distribution (8 features) and cost distribution (8 features) of the corresponding point cloud. To achieve rotation invariance around the vertical axis z , PCA is carried out leading to three principal directions 1, 2, 3.

Enlargement characterizes the extent of the object $\Delta_{dir} = \|\mathbf{p}_{max} - \mathbf{p}_{min}\|_2$, measured in the three PCA main directions and in the vertical direction.

$$[f_2, \dots, f_5]^\top = \left[\frac{\Delta_2}{\Delta_1}, \frac{\Delta_3}{\Delta_1}, \frac{\Delta_3}{\Delta_2}, \frac{\Delta_z}{\Delta_1} \right]^\top \quad (5)$$

The normal distribution is based on a 2D histogram of the angles of the normal vectors $N = \{\mathbf{n}_i\}$. To achieve rotation invariance, the histogram is characterized by counting the minimally necessary bins b to contain at least a certain percentage p of the normals.

$$[f_6, \dots, f_9]^\top = [\xi_{25\%}, \xi_{50\%}, \xi_{75\%}, \xi_{90\%}]^\top, \text{ with} \quad (6)$$

$$\xi_p = \arg \min_i \sum_{k=1}^i |b_k| \geq p \cdot |N|$$

For the three most occupied bins the exact percentage is calculated.

$$[f_{10}, f_{11}, f_{12}]^\top = \left[\frac{|b_1|}{|N|}, \frac{|b_2|}{|N|}, \frac{|b_3|}{|N|} \right]^\top \quad (7)$$

The last normal feature determines the percentage of vertical components in the normals, thus describing the horizontal orientation of the underlying point cloud.

$$f_{13} = \frac{1}{|N|} \sum_{i=1}^{|N|} n_z \quad (8)$$

The geometry is further described by the segmentation costs c_1, c_2, c_3 together with the planarity factor s . The distribution of the costs and planarity factor within the segment are described by the mean μ and the standard deviation σ .

$$[f_{14}, \dots, f_{21}]^\top = [\mu_{c_1}, \mu_{c_2}, \mu_{c_3}, \sigma_{c_1}, \sigma_{c_2}, \sigma_{c_3}, \mu_s, \sigma_s]^\top \quad (9)$$

The result is a feature vector $\mathbf{f}_i = [f_1, \dots, f_{21}]^\top$ describing a segment in 21 dimensions.

2.3. Clustering

Each feature vector describes the geometric characteristic of a point cloud by a simple point in feature space. Grouping close points together means to group similar geometric structures. It was already mentioned that the Manhattan-Distance induces a meaningful similarity measure in feature space. With this metric a simplification of the iterative clustering algorithm Mean-Shift [2] is used. Our Mean-Shift algorithm is controlled by one parameter d , describing the diameter of a local hyper sphere, in which the mean is calculated. Every iteration step, the sphere center is moved to the mean of all feature vectors inside the sphere. In the case of convergence, all feature vectors are removed from the solution set and the mean is stored as center for cluster \mathbf{c}_i . By doing so, the diameter d can be interpreted as the maximum allowed dissimilarity between two feature vectors in one cluster. Note that every cluster represents a potential object class, containing point clouds with similar geometric structures. According to the main idea, clusters with only one member can be ignored.

2.4. Inference strategy

Inference from repetitive objects to an object class is sometimes ambiguous. In Fig. 2 you see *e.g.* a ST of a scene containing two instances of the object class "tree". However, possible object classes are also "foliage", "branches" and "stems". Thus, defining all clusters from Sec. 2.3 as an object class is not the ideal solution.

The most obvious inference strategy to resolve the ambiguity takes always the class with the largest instances (strategy \mathcal{A}). Therefore, a cluster gets assigned the averaged scale value f_{scale} of its members. After sorting the clusters in descending order, the following iterative process is carried out. Define the largest cluster as object class and delete the subtrees of every instance. Take the next largest cluster until every cluster is deleted or defined as object class. The upside of this strategy is that ambiguity is fully resolved in a way that a 3D point can occur only in one object class. The downside is that large classes dominate the results and information about subclasses isn't used.

Another approach, denoted as strategy \mathcal{B} , calculates a plausibility value $\rho(\mathbf{c}_i)$ as a function of the number of cluster members and their distribution in the ST tree. A 3D

point will often occur in several clusters, but these clusters can be put in order according to their plausibility value (*c.f.* Fig. 5). Two steps are necessary to calculate the plausibility value. First, the node positions in the ST are set in relation to their longest paths (root to leave), which yields the "relative position" $\eta = \frac{\text{node level}}{\text{longest path}}$. Second, the position of a cluster \mathbf{c}_i in the ST can be specified by the relative positions of all cluster members. Thus every cluster has a mean relative position $\bar{\eta}_{\mathbf{c}_i}$ with respective standard deviation $\sigma_{\eta_{\mathbf{c}_i}}$. The plausibility value is finally defined as

$$\rho(\mathbf{c}_i) = \frac{|\mathbf{c}_i|}{|\{\text{nodes } j \in \text{ST} | \eta_j \in [\bar{\eta}_{\mathbf{c}_i} - \sigma_{\eta_{\mathbf{c}_i}}, \bar{\eta}_{\mathbf{c}_i} + \sigma_{\eta_{\mathbf{c}_i}}]\}|} \quad (10)$$

which puts the number of cluster members in relation to all nodes of the ST located in the interval $[\bar{\eta}_{\mathbf{c}_i} \pm \sigma_{\eta_{\mathbf{c}_i}}]$. The distribution of the plausibility values for the dataset used in the experiments is shown in Fig. 5.

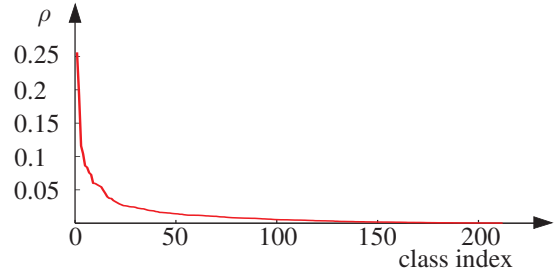


Figure 5: Plausability value ρ of the object classes

3. Experiments

We carried out experiments on a publicly available dataset recorded in the city of Beijing [15], see Fig. 1. The scene comprises objects like building facades, trees, bushes, cars, and street lamps. Unfortunately, as with other outdoor datasets, there is no ground-truth labeling available and obtaining such a labeling would be very hard (especially as compared to indoor datasets or images). This leaves us to stick to a qualitative evaluation.

The first focus of evaluation lies on the proposed segmentation. As mentioned in Sec. 2, it is difficult to design a method capable of segmenting both man-made structures and vegetation. Additionally, parameters must often be adapted for different kind of objects. The method proposed in this work reduces all adaption to one single segmentation threshold θ . Some segmentation outcomes for various choices of θ are depicted in Fig. 4. It can be seen that the proposed method is able to deliver the requested outcome – although the optimal value of θ depends on the local scene. Hence, segmentation is carried out for various values of θ and the segments are arranged within a segmen-

tation tree, as in Fig. 3. This delays the choice of an appropriate value for θ to the inference stage and allows for different values at different local scenes.

After segmentation, feature extraction and clustering are performed to discover similar segments. The final inference algorithm then defines object classes based on the clusters. The two proposed inference algorithm versions are analyzed in the following. Please note that the algorithms have no notion of semantics, so the specified meanings of the object classes were manually attached to the results.

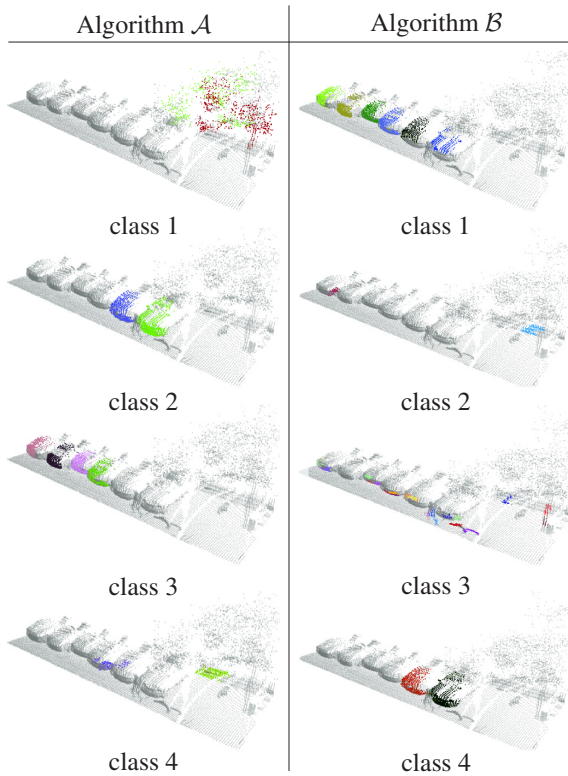


Figure 6: Discovered object classes of inference algorithm \mathcal{A} and \mathcal{B} on a sub-part of the scene (color indicates object instance).

Fig. 6 shows the first four classes discovered by both algorithms from a small subsection of the dataset. Using the complete data set, more classes are discovered. Statistics of the first twenty classes are listed in Tab. 1 which comprise the classes floor, building facade, tree-tops, trees, bushes, cars, and foliage. Street lamps are the only occurring objects that were not among the first 20 discovered classes. Some selected object classes are depicted in Fig. 7.

Both algorithms are able to discover relevant object classes. Because algorithm \mathcal{A} focuses on spatially bigger objects, periodic object arrangements like trees in an avenue cause merged instances (picture 1 in Fig. 7(a)). Algorithm \mathcal{B} seems to focus more on frequently occurring classes, which

Table 1: Statistics of the first 20 discovered object classes of the full dataset.

class	algorithm \mathcal{A}	algorithm \mathcal{B}	
	objects	objects	ρ
1	3	472	0.26
2	2	606	0.19
3	9	284	0.12
4	2	285	0.1
5	2	133	0.09
6	2	1342	0.08
7	2	57	0.08
8	8	128	0.07
9	2	12	0.06
10	3	1012	0.06
11	8	2	0.06
12	14	896	0.06
13	2	35	0.05
14	17	50	0.05
15	9	717	0.04
16	6	635	0.04
17	5	597	0.04
18	2	92	0.04
19	25	125	0.03
20	10	20	0.03

is why objects are preferred that fractionize a lot during segmentation (e.g. foliage, depicted in picture 1 in Fig. 7(b)). An advantage of the latter is that each object class has associated a plausibility value which might be used as an objective measure to limit the number of discovered classes.

Altogether, there is no clear preference for either of the two algorithms. Selection will depend on the application area.

4. Conclusion and Future Work

This work presented a first approach to discover object classes in outdoor scenarios in an unsupervised manner. The focus on pure 3D point cloud data thereby makes the approach independent from texture variations and enables the use of a wide range of sensors. A novel segmentation algorithm was developed that is able to deal with man-made structures as well as with vegetation. Two inference algorithms were presented that both deliver good but different results. Nevertheless, there remain areas to improve. Segmentation might be improved to yield the same sub-parts for similar objects. Additional features might be developed to describe the segments. Finally, the inference algorithms might be extended. Especially with an improved segmentation the inference algorithm can be altered to match sub-trees which might in the end be more robust.

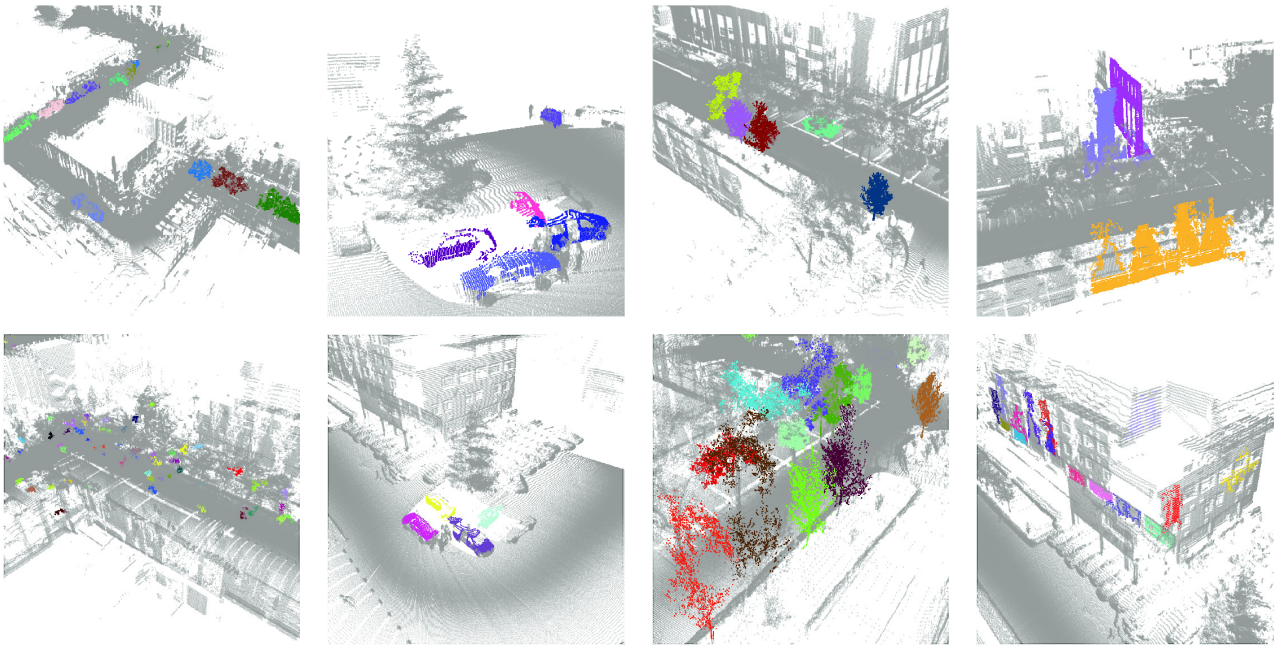


Figure 7: Discovered object classes (color indicates object instances) of (a) inference algorithm \mathcal{A} (top row) : trees, cars, trees, building facades. (b) inference algorithm \mathcal{B} . (bottom row): foliage, cars, trees, building facades

References

- [1] N. Ahuja and S. Todorovic. Connected segmentation tree - a joint representation of region layout and hierarchy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 2, 3
- [2] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. 5
- [3] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard. Unsupervised discovery of object classes from range data using latent dirichlet allocation. In *Proceedings of Robotics: Science and Systems*, 2009. 2
- [4] K. Klasing, D. Althoff, D. Wollherr, and M. Buss. Comparison of surface normal estimation methods for range sensing applications. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3206–3211, 2009. 3
- [5] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. In *IEEE Intelligent Vehicles Symposium*, pages 215–220, 2009. 3
- [6] A. Patterson, P. Mordohai, and K. Daniilidis. Object detection from large-scale 3D datasets using bottom-up and top-down descriptors. In *European Conference on Computer Vision (ECCV)*, pages 553–566, 2008. 2
- [7] A. Rabinovich, S. Belongie, T. Lange, and J. Buhmann. Model order selection and cue combination for image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1130–1137, 2006. 2
- [8] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard. Unsupervised learning of 3D object models from partial views. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 801–806, 2009. 2
- [9] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang. Contextual part analogies in 3D objects. *International Journal of Computer Vision*, 89:309–326, 2010. 2
- [10] J. Shin, S. Gachter, A. Harati, C. Pradalier, and R. Siegwart. Object classification based on a geometric grammar with a range camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2443–2448, 2009. 2
- [11] J. Shin, R. Triebel, and R. Siegwart. Unsupervised discovery of repetitive objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010. 2
- [12] S. Todorovic and N. Ahuja. Unsupervised category modeling, recognition, and segmentation in images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(12):2158–2174, 2008. 2
- [13] R. Triebel, J. Shin, and R. Siegwart. Segmentation and unsupervised part-based discovery of repetitive objects. In *Proceedings of Robotics: Science and Systems*, 2010. 2
- [14] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88:284–302, 2010. 2
- [15] H. Zhao, L. Xiong, Z. Jiao, J. Cui, H. Zha, and R. Shibasaki. Sensor alignment towards an omni-directional measurement using an intelligent vehicle. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 292–298, 2009. 5