

A CONCEPT FOR THE SEGMENTATION OF INDIVIDUAL URBAN TREES FROM DENSE MLS POINT CLOUDS

P.-R. Hirt*, L. Hoegner, U. Stilla

Photogrammetry and Remote Sensing, Technical University of Munich, Munich 80333, Germany –
(philipp.hirt, ludwig.hoegner, stilla)@tum.de

Commission II, WG II/3

KEY WORDS: mobile laser scanning, point cloud, instance segmentation, voxel, tree segmentation, urban areas

ABSTRACT:

In our daily lives, trees can be seen as the tallest and most noticeable representatives of the plant kingdom. Especially in urban areas, the individual tree is of high significance and responsible for a manifold of positive effects on the environment and residents. In the context of urban tree registers and thus monitoring of urban vegetation, we propose a general concept for the segmentation of trees from 3D point clouds. Mobile Laser Scanning (MLS) is introduced as the preferred sensor. Based on an analysis of earlier work in this field, we gather arguments and methods in order to involve segmentation in the bigger frame of a tree register workflow, including detailed modeling and change detection. Our concept for segmentation is based on a voxel-structure. In a first step, region growing approaches are used for ground removal and rough segmentation. Later, graph-based optimization will separate neighboring trees. For now, only the general concept can be introduced—quantitative analysis and optimization of the steps will follow in future work.

1. INTRODUCTION

In the view of increasing urbanization and enhanced environmental consciousness, green spots and large trees are gaining more and more importance in urban areas. Not only do they improve the quality of air and living, they also contribute to the well-being of all residents and serve as a habitat for animals. Thus, it should be a primary concern for municipalities to grow and maintain a healthy and diverse population of plants, especially trees, inside cities. On the other hand, natural objects are far less predictable than man-made ones and—as they are living beings—are changing significantly over time. To ensure general road safety and to avoid wild growth in populated areas, especially trees have to be inspected regularly. Tree registers can be used to maintain a reliable monitoring of urban trees (Tanhuanpää et al., 2014).

As manual and repeated inspection of all trees in one city is expensive both in cost and time, automated approaches for measurement and processing should be considered. In order to describe the three-dimensional complexity of trees, we choose an active sensor: LiDAR (Light Detection And Ranging) promises high structural detail and independence of illumination. Furthermore, it can precisely reach and look through densely branched regions. The street as main infrastructural element in urban areas combined with aspects of mobility, cost and measuring perspective speak for the use of Mobile Laser Scanning. This acquisition method can cover the area of a city while still retaining detailed measurements. More importantly, the viewing angle of a laser scanner mounted on a vehicle allows to acquire data from stem, branches and crown, while Airborne Laser Scanning is often limited to the surface of the crown and the ground below. Only in the case of full waveform LiDAR (Reitberger et al., 2009), a certain but—compared to MLS—lower amount of points on stem and branches can be captured. Moreover, largely built structures may lead to the obstruction

of trees in case of an airborne acquisition. Mobile Laser Scanning is limited by objects directly on the side of the street, like parked vans or big hedges, but should still be able to see at least the important upper part of urban trees.

The potential of the acquired data has to be exploited as thoroughly as possible. For administrative purposes, position and certain parameters like the Diameter at Breast Height (DBH) are mandatory to be identified. Therefore, a mere classification of tree points is not sufficient. We need dedicated tree instances; hence, we have to identify which point belongs to which individual tree. But the dense and detailed point information from MLS should not only be utilized for these rough measures—even a detailed reconstruction for 3D city models is possible and should be kept in mind. Having dug into the advantages of spatial resolution, we have to consider temporal resolution, too. At least yearly acquisitions via MLS are desirable. Our process should hence also aim for temporal comparability, and provide the basic capabilities for change detection. These requirements will influence our decision on the methods to be used.

In this paper, we introduce a general concept for the segmentation of individual trees from dense 3D point clouds. We assume data acquisition by Mobile Laser Scanning due to its salient characteristics in urban areas. This can be seen as the first step for the establishment of a city-wide automated tree register featuring regular update intervals and detection of significant changes in the urban flora. After providing an overview of existing segmentation techniques, we will present our proposed method in the view of the challenges provided by our requirements (Section 3). However, this concept still needs to be tested and optimized in a few points. Quantitative analysis will follow in future work. In section 4, we will give an insight on the benefits of MLS by introducing our benchmark dataset. Correspondingly, the behavior of trees in view of naive distance clustering is analyzed (Section 4.2) and discussed with respect to our proposed concept (Section 5). An outlook on further work in this context will be presented in Section 6.

* Corresponding author

2. RELATED WORK

The challenge of tree segmentation, especially instantiation of single trees, has been discussed considerably over time. Whereas early approaches mostly rely on sparse data from Airborne Laser Scanning (ALS), a shift to Mobile Laser Scanning can be observed in the context of urban trees. However, some of the concepts from ALS data can still be adopted for MLS measurements. In a first glance on methods in forestry, we will present some of those. Sections 2.2 and 2.3 will then distinct the general approaches of point- and voxel-based methods. After an analysis of common concepts, we will draw the conclusion regarding our requirements in section 2.4.

2.1 Tree Segmentation in Forestry

The general idea of individual tree segmentation can be met in the context of forestry, too. An overview of segmentation methods from ALS data is given in (Lu et al., 2014). In their listing of earlier approaches, especially watershed, region growing and normalized cut are often used. Considering the top-down view of ALS data, watershed approaches are the most sensor specific ones, as they mainly aim for the separation of tree crowns in densely populated forests where trees are the only and highest objects. Region growing and normalized cut, on the other hand, could also be used in the context of MLS data. Lu et al. (2014) themselves start with a clustering based on horizontal and vertical distances. The complete tree is then derived in a growing process based on the 3D distance. Due to a density of approximately 10 pts/m², Lu et al. stay on point level for all processing steps.

Different sensors can change the characteristics of the data significantly: Full waveform LiDAR leads to more points along the tree stem compared to first/last pulse LiDAR (Reitberger et al., 2009). Rough initial tree locations are derived by a watershed algorithm. An efficient, grid-based Canopy Height Model (CHM) is used in that context. The stems are found by a clustering using horizontal distances; the individual trees are identified by applying a normalized cut method on the voxelized data. In that context, Reitberger et al. (2009) propose several similarity features: weighted horizontal and vertical distance, distance to the assumed stem position, mean intensity and mean pulse width for each voxel.

2.2 Point-Based Segmentation of Urban Trees

Concentrating on urban trees, ALS-oriented approaches like adaptive mean shift with a tree shaped 3D kernel (Xiao et al., 2016) may be feasible at lower point density, but not for dense MLS data. Other researches perform change detection on ALS data in urban areas, including trees, but do not have dedicated segmentation steps. These are Xu et al. (2015) with an octree-based change detection on buildings and trees in general, and Tran et al. (2018) who apply a Random Forests classification with a dedicated inter-epoch feature to achieve classification and change detection in one step. Both these methods rely on low point densities between 10 and 50 pts/m².

Combining our requirements of an urban environment and dense point clouds, we concentrate on segmentation approaches using MLS data. Due to the large amount of data, the processing of every single point comes expensive in storage and computational requirements. Assuming that the loss of detail through voxelization is unacceptable, Weinmann et al. (2017) present a

point-based method that uses Random Forests (RF) and automatically selected geometric features. The semantic segmentation only identifies the class *tree*. Thus, it is followed by a mean shift segmentation on a downsampled 2D projection. In that way, the individual trees are detected. These segments are then refined by rule-based shape analysis. In another point-wise classification method, Yao et al. (2017) underline the importance of local topological relations. Thus, they propose a workflow that augments point-wise semantic segmentation (Random Forests) by Conditional Random Fields (CRF). The authors concentrate on the efficient calculation of the contextual optimization. However, the original data has been downsampled in this approach by a voxel-grid.

It is equally feasible to first extract pole-like objects by slicing the data into horizontal layers and then to classify these objects into trees and others by local features (Fan et al., 2020). They use Support Vector Machines (SVM) as classifier. In a preprocessing step, data is reduced by filtering single points and large planes. Mapping on horizontal layers is performed in (Monnier et al., 2012), too. Dimensionality features and a cylindrical descriptor are combined with a probabilistic relaxation model in order to homogenize clusters. Monnier et al. then detect the trunks by projecting the features on a horizontal accumulator space while relying on a priori knowledge on the properties of trees compared to other objects. Similar to that, Husain and Chandra Vaishya (2020) use projection of their points on a 2D grid at different height layers. Assuming flat terrain, this leads to characteristic clusters in the projections on ground level, trunk and crown height.

In conclusion, point-based segmentation of dense MLS point clouds is possible, as long as efficient strategies are used. Preprocessing steps—especially if the only goal is to segment trees—and dimensionality reduction by 2D projection are essential (Yao et al., 2017; Fan et al., 2020; Monnier et al., 2012; Husain and Chandra Vaishya, 2020). For (semantic) segmentation, local features as presented in (Weinmann et al., 2015) are useful, but need a defined neighborhood. In the case of dense MLS data, a spherical neighborhood is to be preferred (Weinmann et al., 2017). To avoid isolated points, the context must be taken into consideration, either by probabilistic modeling (Yao et al., 2017; Monnier et al., 2012), or by further clustering algorithms (Weinmann et al., 2017).

2.3 Voxel-Based Segmentation of Urban Trees

Dense data can be handled by inspecting cubes containing several of the original points. In this context, we understand voxel-based methods as approaches relying on the voxel as semantic element with dedicated features based on its points, not as a mere filtering concept to reduce data size. This regular structure of semantically enriched elements is then used as a basis for a graph model (Yao and Fan, 2013; Guan et al., 2019; Chen et al., 2019; Xu et al., 2018).

Before transitioning into voxel space, Yao and Fan (2013) exclude man-made objects, especially facades, exploiting their respective properties when projected to horizontal accumulator spaces at different heights above ground. This idea is similar to (Husain and Chandra Vaishya, 2020), but only used as a pre-filtering step here to remove man-made objects. Inspired by the normalized cut approach of Reitberger et al. (2009) with ALS data, Yao and Fan apply the same idea to the 3D voxel space of MLS data. The weights are influenced by vertical and horizontal Euclidean distance, and the distance in intensity. In

that way, the graph optimization will ensure separability of adjacent crowns. However, the authors state that additional steps are needed to identify pole-like objects by shape analysis.

A similar workflow of pre-filtering, clustering and graph cut is adapted by Guan et al. (2019). In contrast to the aforementioned approach, the octree-based voxel structure is the key element in every step: Growing upwards from the bottom-most voxels to their respective 9 upper neighbors is used to filter ground voxels. The remaining data is first clustered by means of Euclidean distance. These clusters are refined by a voxel-based normalized cut. The graph weights in that case are horizontal, vertical and shortest Euclidean distance between nodes. Likewise, upward-growing for terrain filtering can be seen in (Qin et al., 2018) and (Chen et al., 2019). Chen et al. augment the process of clustering and normalized cut by a refinement process using further properties like reflectance and point distribution.

The general idea of octree-based region growing is presented in (Vo et al., 2015) for building segmentation. The features considered in growing are the normal vectors and the mean residual value to an approximated plane. This leads to segmentation of rather planar areas. These features are computed for each voxel, so computationally expensive neighborhood search can be avoided. In the same context, Xu et al. (2017) introduce graph-based segmentation using voxels and supervoxels. Their applied grouping laws hold true for any 3D scene, but seem to be optimized for man-made structures due to the emphasis on cutting at sharp edges and discontinuities. This might contradict the natural, partly rough structure of trees. Hence, in the context of natural objects, Xu et al. (2018) first apply a supervoxel-based semantic segmentation and then use graph-based regularization and segmentation to achieve homogeneous and distinctive segments. However, the exact division of overlapping tree crowns remains critical.

The aforementioned methods use an octree-like structure to generate a voxel grid or to manage their data and search procedures. An efficient generation of such a structure leading to distinctly addressable voxels is presented by Huang et al. (2019).

From the introduced approaches, we can clearly identify graph-based optimization as one of the mostly used methods. The primary element voxel or supervoxel leads to a significant reduction of nodes and already defines the neighborhood for the computation of features for each voxel. However, it becomes clear that in any case certain preprocessing steps are needed. Planar elements, especially the ground (Qin et al., 2018; Chen et al., 2019) or man-made objects in general (Yao and Fan, 2013) should be removed in a first step. This is followed by a simple clustering method, in case of (Guan et al., 2015; Chen et al., 2019) Euclidean clustering. Optional to these two steps, a general semantic segmentation can be performed in a first step, which is then optimized by graph-based methods (Xu et al., 2018).

2.4 Assessment of Existing Methods

Table 1 gives an overview of the presented methods. We see clear advantages in using voxels: The immediate neighborhood for feature computation can be defined by the voxel, and these neighboring points can easily be addressed by their parent voxel. In this process we assume that the voxel still contains a list of the original points inside of it. Efficient structuring and addressing can be realized by octrees (Huang et al., 2019). One drawback is the in-homogeneous character of MLS data: the further

away on object is from the sensor, the sparser are its points. This is contradicted by a fixed voxel size—in the worst case, dimensionality features cannot be computed for a voxel as it contains only one point, although the larger neighborhood of this point still describes an identifiable object. Yet, voxels open the way for any kind of graph-based optimization. Graph cut on voxels (Reitberger et al., 2009; Yao and Fan, 2013; Guan et al., 2015; Chen et al., 2019; Xu et al., 2018) has been used extensively in the recent years. One key element will be to find appropriate weights that accomplish correct cuts along the boundaries of a tree's foliage.

We can identify approaches for pre-processing and actual segmentation. The first step—data reduction—removes at least the planar elements and the ground from the point cloud. It is desirable to use the same basic data structure as in the follow-on steps (Guan et al., 2019), which again is leading to an octree. Voxel-based upward-growing generally is used in the phase of ground removal (Guan et al., 2019; Qin et al., 2018; Chen et al., 2019). This can be extended to using an upward-growing approach for the tree clustering, too. Starting from literal seed points, a region growing in general upward direction can lead to initial tree clusters. A fine segmentation at areas where two segments would grow into each other can then be optimized by normalized cut.

In a final step, the derived segments have to be identified as trees. Weinmann et al. (2015) propose shape analysis in that context. Likewise, a priori knowledge (Monnier et al., 2012) on e.g. the difference between trees and man-made objects can already be included in the growing phase.

3. PROPOSED METHOD

The analysis of existing literature and our own requirements lead us to a concept for tree segmentation. Having defined our assumptions on the data (Section 3.1), we will outline the individual steps of our approach. An overview of this process is depicted in Figure 1. Finally, we will give a recommendation for handling spatially in-homogeneous MLS data (Section 3.4).

3.1 Assumptions and Preconditions

First, we have to define certain assumptions on our scene. In general, most methods are optimized for trees growing in more or less vertical direction, especially if an upward-growing method is used. Some old trees or certain species might not fulfill this criterion. The direction of growth is also influenced by the surrounding landscape—on steep cliffs or embankments, partly vertical growth especially in the lower part of a trunk can be observed. On the other hand, the urban areas we focus on will most likely contain little of such rough terrain. In that context, we also have to assume that there is a considerable volume filled with points above the ground position of the stem. That means, the main part of the crown must be approximately vertically above the trunk. Another requirement for some of the existing methods is flat terrain (Husain and Chandra Vaishya, 2020), meaning a ground surface lying in the XY-plane. Our approach aims to be more flexible. This can be achieved by using height values relative to an existing or computed Digital Terrain Model (DTM). However, a certain continuity of the terrain should improve the behavior of any ground extraction method in general.

Authors	Data acquisition	Algorithm
Point-based		
Lu et al. (2014)	ALS	Region growing (based on distance)
Xiao et al. (2016)	ALS	Adaptive mean shift
Weinmann et al. (2017)	MLS	RF classification, mean shift and shape analysis
Yao et al. (2017)	MLS	RF + CRF
Fan et al. (2020)	MLS	Pole-like objects extraction, tree classification by SVM
Monnier et al. (2012)	MLS	Dimensionality features, horizontal accumulator planes
Husain and Chandra Vaishya (2020)	MLS	2D projection at different heights, circular growing
Voxel-based		
Reitberger et al. (2009)	ALS	Clustering followed by normalized cut
Yao and Fan (2013)	MLS	Horizontal accumulator planes, voxel-based normalized cut
Guan et al. (2019)	MLS	Voxel-based upward-growing, Euclidean clustering, normalized cut
Qin et al. (2018)	MLS	Ground removal by voxel-based upward-growing
Chen et al. (2019)	MLS	Voxel-based upward-growing, Euclidean clustering, normalized cut
Xu et al. (2018)	MLS	Graph-based segmentation using pre-classified supervoxels
Vo et al. (2015)	general	Octree-based region growing

Table 1. Existing algorithms on segmentation of individual trees.

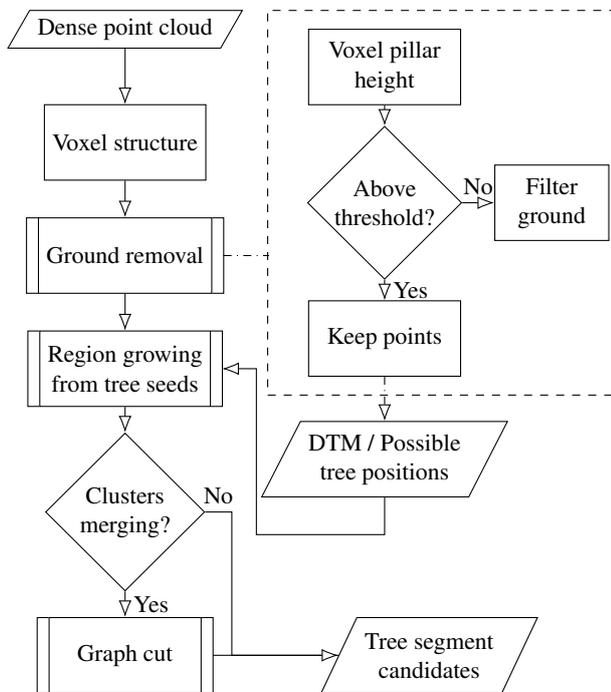


Figure 1. Overview of the processing from input point cloud to segments of possible trees.

3.2 Preprocessing and Ground Removal

At the beginning of the process, data reduction is the key element. This can be reached by data filtering and removal of non-tree objects, especially points from streets and natural ground.

3.2.1 Efficient Octree. For computational efficiency in the following processes, a noise filtering should be applied in the beginning to remove single blunder points. This can be done by radius filtering. In the next step, the point cloud is voxelized using the octree structure proposed by Huang et al. (2019). A binary address can be computed from the coordinates of each point and the desired voxel size. By this address, every point is assigned to a voxel whilst the address represents the position in an underlying octree. Adjacent voxels can be found easily by

increasing or decreasing certain positions in the binary address by 1.

3.2.2 DTM and Continuous Pillar Model. With this data structure as basis, growing approaches can easily be implemented. To start an upward-growing process for ground removal, seeds are needed. We start at our lowermost voxel layer and check for every voxel whether it is populated. If not, we step up one layer until a populated voxel is reached. Its lowest point height can be stored as pixel value in a corresponding DTM. Due to possible obstructions or blunders, empty cells in the DTM have to be filled by an adaptive median filtering only effective on data gaps. Outliers can be removed by smoothing.

Starting with this first populated voxel in one stack or *pillar* of voxels, we proceed to check the voxels above. This process is outlined in the dashed sub-process box of Figure 1 and further illustrated in Figure 2 for one slice of voxel space. As soon as the voxels above contain no more points, growing stops. The Z-coordinate of the highest point compared to the terrain height leads to the height of this voxel pillar. In that way, we automatically include non-planar terrain by measuring the actual elevation model. At the end of that process, we have derived the DTM including the lowest point heights per voxel and a pillar-top model including the maximum z-coordinates of each continuous pillar. The difference between these layers can be designated as *continuous pillar height* and stored as 2D raster image, too. Each raster value can then be used to assign a voxel pillar to *ground* or *non-ground*. This computation of actual pillar height can be implemented by a 2D-image difference (Figure 2b), the filtering process as simple threshold application.

In case of an object in a certain height above ground, e.g. a tree crown, this approach will only identify the lowest point layer as ground and—as long as the height difference is bigger than one voxel—ignore the overlapping crown points. This raster-based approach can easily be parallelized by multiple threads.

3.3 Segmentation of Tree Candidates

After having removed all voxels marked as ground, we can start the tree segmentation at the DTM cells beneath high voxel pillars. Our general approach is to first cluster the objects by region growing from the lowermost points. These clusters grow

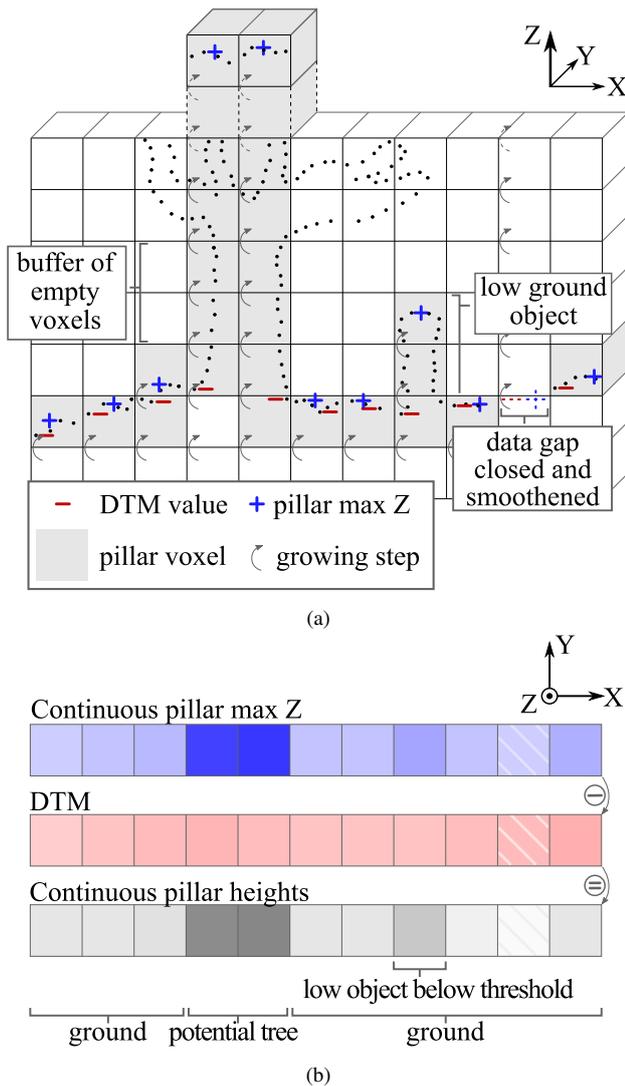


Figure 2. Computation of the *continuous pillar heights* for ground filtering. Only one slice of the voxel space (a) and the corresponding rows in the raster images (b) are shown. The color intensity represents the relative value per raster cell.

upwards and will—in case of isolated objects—already describe the single tree or some other compact, high-rising object.

If trees are very close to each other and their crowns are overlapping, points on the border can be claimed by two clusters from two original trees. In that case, these clusters will merge into one object, which then will be flagged. As a correct division cannot be found by simple clustering, an optimization based on normalized cut will be used to separate them at an ideal border (Section 3.3.3).

3.3.1 Feature Computation. For any graph-based or growing approach, we need to assign certain features to each voxel on which we can base our decisions. In the context of trees, other features are descriptive than e.g. in planar segmentation (Huang et al., 2019; Vo et al., 2015). The normal vector computed from the included points can be useful to ensure smoothness, however the influence on complex tree structures has to be examined. Likewise, the center of gravity (centroid) of the points of each voxel is computed. In order to avoid the strong influence of few foreign points at the edge of the crown, medoid computation should be considered, too. The medoid of a voxel

is the exact data point with the minimal average distance to all the other points.

Due to the trees' complex geometry, it cannot yet be decided whether eigenvalue-based features as in (Weinmann et al., 2017) are of significant use in clustering, as the different parts of a tree have different characteristics in dimensionality and point distribution. Further investigations shall identify the most promising candidates for the specific application of region growing and graph cut.

3.3.2 Potential Tree Clusters by Region Growing. To get initial clusters of potential high-rising objects, we start at the voxels on DTM level that were not filtered in the step before. The most robust way is to initiate region growing to the direct 26 neighbors of each voxel, as we cannot be sure that all parts of the tree will always grow into vertical direction. Because of the same reason, we compute 3D Euclidean distance as well as the distance in XY-direction between two neighboring centroids. The lesser of both has to be below a defined threshold in order for the two voxels to be clustered together. In that way, we favor exact vertical growing as it will happen at the main stem, but also consider other more general growing directions. The normal vector is not used at the moment as it might lead to problems at locations where branches leave the main stem—sharp angles are to be expected there.

Another approach would be to perform exact Euclidean clustering point by point. In that way, we might derive more continuous clusters by the cost of higher computational effort. In addition, we have to get back on voxel level for the normalized cut approach. First investigations in that direction will be presented and analyzed in Section 4.2.

In any case, if two clusters meet, they will be merged into one and flagged for further processing with graph cut.

3.3.3 Separation of Merged Clusters by Normalized Cut. The simple clustering approach presented before is based on a hard thresholding decision and thus cannot account for the overall optimization of one certain segment. Hence, we apply graph-based optimization to clusters that most likely include two separate objects, but have grown together in the clustering step. A graph is built with the voxels as nodes, and edges between two adjacent voxels. Each voxel is hence connected to its 26 neighbors, if they exist. These edges will need weights.

We use some of the features presented in Section 3.3.1. We define weight components between the nodes i and j as

$$\omega_{f_{i,j}} = \frac{\|f_i - f_j\|^2}{\sigma_f^2}, \quad (1)$$

where the numerator is the squared Euclidean distance in the respective feature between both voxels. This holds true independently of the features being vectors or scalars. σ_f^2 can be used to assign different individual weightings to the single components. The overall weight is defined by the 3D and 2D distances, either between centroids or, if significant improvement can be observed, between medoids. Moreover, we add a condition to focus on a vertical cylinder with radius r_{XY} (inspired by Reitberger et al. (2009)) around the respective centroid. In that way, we favor vertical growth and allow horizontal dissection in case two points are horizontally too far away from each other.

Thus, the edge weight between two nodes is

$$w_{ij} = \begin{cases} \exp(-\omega_{D_{i,j}^{3D}}) \cdot \exp(-\omega_{D_{i,j}^{XY}}) & \text{if } D_{i,j}^{XY} < r_{XY} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Further features can be appended by adding the respective version of Equation 1 to the product of exponentials. The ideal combination and individual weighting will be subject to future work.

Normalized cut will then split the current segment into two by minimizing the cost function

$$\text{NCut}(A, B) = \frac{\text{Cut}(A, B)}{\text{Assoc}(A, V)} + \frac{\text{Cut}(A, B)}{\text{Assoc}(B, V)} \quad (3)$$

between the target segments A and B.

$\text{Cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$ is the sum of weights between these segments and $\text{Assoc}(A, V) = \sum_{i \in A, j \in V} w_{ij}$ is the association of one segment; that is the sum of weights of all edges which end in segment A— analogously for $\text{Assoc}(B, V)$. The initial tree positions and hence estimation of the number of trees will also contribute to tree separation.

3.4 Approaching In-homogeneous Data

One major flaw of MLS data is the in-homogeneous distribution of points. The distance to the sensor can be seen as the main criterion for average neighboring point distance. If the position of the vehicle is known, as e.g. its trajectory has been recorded and included in the point cloud, we can adapt the voxel size to the distance to the sensor. To keep this feasible, the voxel side length could be doubled as soon as the average point density is expected to be halved. Of course, this is only a rough approximation. The implementation, on the other hand, can be realized by stepping up one layer in the octree—this gives the parent node containing eight of the original voxels, and thus an increased volume adapted to the alleged point density.

4. EXPERIMENTS

4.1 Dataset

First tests of the presented methods have been conducted on the benchmark dataset TUM-MLS-2018 (Fraunhofer IOSB, 2018). In connection with the earlier TUM-MLS-2016 dataset (Gehring et al., 2017; Zhu et al., 2020) and potential future measurement campaigns, experiments on change detection will be possible, too. Point clouds were captured by two Velodyne HDL-64E LiDAR sensors oriented oblique to each other. They are mounted on the right and left front corner of the vehicle roof and rotated by 25° to the horizontal and 45° outwards. This leads to a considerable overlap especially in front of the vehicle. As each sensor scans with 64 scan lines, point density is varying strongly depending on the motion of the vehicle and the location relative to it. The data of 2018 was captured in December under leaf-off conditions.

The sensors also record the echo amplitude (intensity) of the returned laser pulse. However, this value is of limited use in connection with trees as it varies strongly along the tree stem and branches. Thus, our approach restricts itself to pure geometric information and analysis. In that way, we gain independence from the specific sensor as long as a point cloud is produced. Figure 3 shows a patch of the observed area.

4.2 Euclidean Clustering: Insights on Data Behavior

In order to analyze the data properties and clustering behavior of trees, some first experiments have been conducted with a straightforward approach. First, outlier points were removed by radius filtering (so that one point must have at least 50 neighbors in a radius of 0.5 m). The ground removal from section 3.2 has been applied as described. Then, we used point-wise Euclidean clustering to derive initial segments. We took the seed points from the ground removal step as in 3.2. The maximum point distance was set to 15 cm.

5. INTERMEDIATE RESULTS

From the first experiments on general clustering behavior we derived the side-views presented in Figure 4. We can observe some well delineated trees. At other trees, the trunk and foliage have been assigned to different clusters (Figure 4b). Of course, there are still some objects like facades and poles left. These will have to be removed beforehand or at the end of the process, likely by the analysis of their shape compared to what can be expected from trees.

All gray points have not been assigned to any cluster because they were isolated: more than 15 cm away from any cluster starting at a seed point. These non-clustered points can be found mainly at the outermost areas of tree crowns. This is due to the comparatively lower point density and the more scattered positioning of the fine branches there. Details like that will be missed as long as clustering with a hard threshold in maximum distance is applied. On the other hand, Figure 5 shows large segments containing multiple crowns. The view from atop clearly illustrates that phenomenon. Again, this happens due to the strict threshold—in some regions, two tree crowns are so close that their respective points have a distance below clustering threshold, thus merging both trees. These are the major effects and drawbacks generated by naive Euclidean clustering in our case.

6. CONCLUSION AND OUTLOOK

From the preliminary investigations taken in Section 4.2, we can draw the main conclusion that clustering in principle does work well on tree point clouds. But, as missed detailed structures show as well as merging crowns, it has to be refined by a more flexible optimization approach. This leads to the ideas outlined in this paper.

We presented a concept for tree segmentation from dense point clouds. Starting with voxel-based ground removal by introducing the idea of voxel pillars, we proposed the combination of region growing from the very bottoms of the trees with optimization by normalized cut. We aim for clearly separable tree crowns especially in dense settings while preserving the detailed structures of even the outermost branches. In this context, we introduced an easily addressable voxel structure while keeping the original data points. This will be the basis for future research: Instance segmentation of urban trees can be seen as the first step in implementing a tree register. Further work will concentrate on the extraction of important parameters like Diameter at Breast Height, detailed tree modeling for 3D city models and change detection between several epochs of data acquisition. Quantitative analysis of the presented concept and beyond will follow. We aim at a methodological framework which will allow the realization of all these three steps on a mutual basis in a highly automatized workflow.

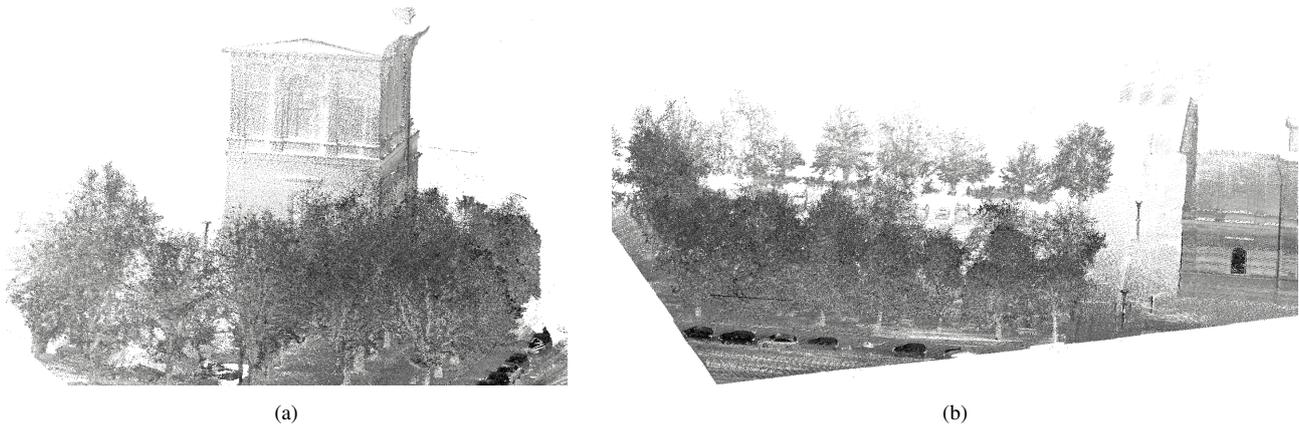


Figure 3. Test area in TUM-MLS-2018 (north of the Alte Pinakothek, Munich) including tightly planted deciduous trees, viewed from the North (a) and West (b), respectively. Gray values correspond to measured intensity.

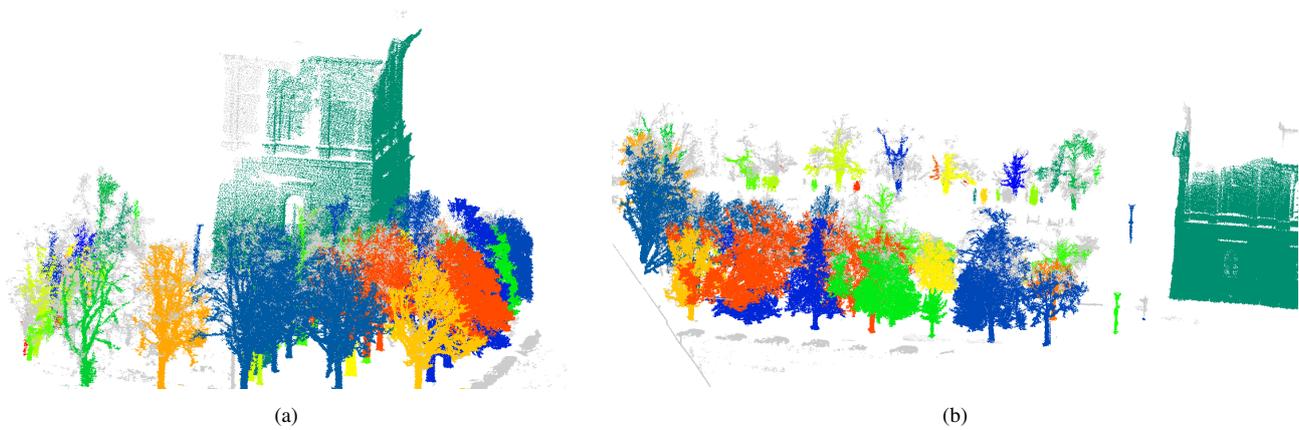


Figure 4. Results of Euclidean clustering with a maximum point distance of 15 cm. The ground has been removed according to section 3.2. Gray points are not clustered. Views from the North (a) and West (b).

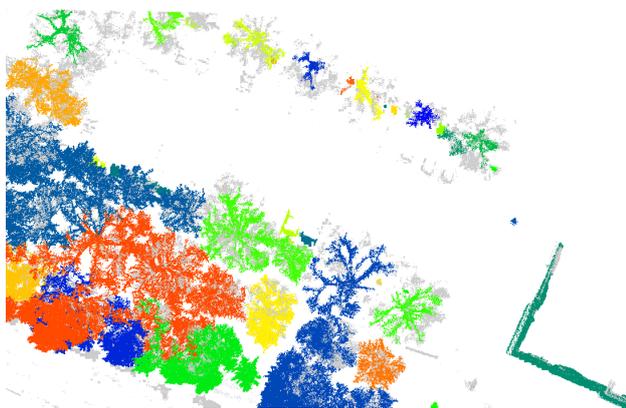


Figure 5. Results of Euclidean clustering, view from above. Connected crowns are clearly visible. Missed points of the outer foliage structure (gray) can be seen, too.

REFERENCES

- Chen, Y., Wang, S., Li, J., Ma, L., Wu, R., Luo, Z., Wang, C., 2019. Rapid urban roadside tree inventory using a Mobile Laser Scanning system. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9), 3690–3700.
- Fan, W., Yang, B., Liang, F., Dong, Z., 2020. Using Mobile Laser Scanning point clouds to extract urban roadside trees for ecological benefits estimation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020, 211–216.
- Fraunhofer IOSB, 2018. Mobile laser scanning - test data: "TUM-MLS-2018". Available online: iosb.fraunhofer.de/en/competences/image-exploitation/object-recognition/3d-data/datasets/tum-mls-2018.html (accessed on 1 Feb 2021).
- Gehring, J., Hebel, M., Arens, M., Stilla, U., 2017. An approach to extract moving objects from MLS data using a volumetric background representation. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1, 107–114.
- Guan, H., Yu, Y., Ji, Z., Li, J., Zhang, Q., 2015. Deep learning-based tree classification using mobile LiDAR data. *Remote Sensing Letters*, 6(11), 864–873.
- Guan, H., Yu, Y., Yan, W., Li, D., Li, J., 2019. 3D-CNN based tree species classification using mobile lidar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, 989–993.
- Huang, M., Wei, P., Liu, X., 2019. An efficient encoding voxel-based segmentation (EVBS) algorithm based on fast adjacent voxel search for point cloud plane segmentation. *Remote Sensing*, 11, 2727.
- Husain, A., Chandra Vaishya, R., 2020. An automated approach for street trees detection using mobile laser scanner data. *Remote Sensing Applications: Society and Environment*, 20, 100371.
- Lu, X., Guo, Q., Li, W., Flanagan, J., 2014. A bottom-up approach to segment individual deciduous trees using leaf-off lidar point cloud data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 94, 1–12.
- Monnier, F., Vallet, B., Soheilian, B., 2012. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3, 245–250.
- Qin, H., Guan, G., Yu, Y., Zhong, L., 2018. A voxel-based filtering algorithm for mobile LiDAR data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1433–1438.
- Reitberger, J., Schnörr, C., Krzystek, P., Stilla, U., 2009. 3D segmentation of single trees exploiting full waveform LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6), 561–574.
- Tanhuanpää, T., Vastaranta, M., Kankare, V., Holopainen, M., Hyypä, J., Hyypä, H., Alho, P., Raisio, J., 2014. Mapping of urban roadside trees – A case study in the tree register update process in Helsinki City. *Urban Forestry & Urban Greening*, 13(3), 562–570.
- Tran, G., Ressel, C., Pfeifer, N., 2018. Integrated change detection and classification in urban areas based on airborne laser scanning point clouds. *Sensors*, 18, 448.
- Vo, A.-V., Truong-Hong, L., Laefer, D. F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104, 88–100.
- Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105, 286–304.
- Weinmann, M., Weinmann, M., Mallet, C., Brédif, M., 2017. A classification-segmentation framework for the detection of individual trees in dense MMS point cloud data acquired in urban areas. *Remote Sensing*, 9(3), 277:1–277:28.
- Xiao, W., Xu, S., Elberink, S. O., Vosselman, G., 2016. Individual tree crown modeling and change detection from airborne LiDAR data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(8), 3467–3477.
- Xu, H., Cheng, L., Li, M., Chen, Y., Zhong, L., 2015. Using octrees to detect changes to buildings and trees in the urban environment from airborne LiDAR data. *Remote Sensing*, 7, 9682–9704.
- Xu, Y., Hoegner, L., Tuttas, S., Stilla, U., 2017. Voxel- and graph-based point cloud segmentation of 3D scenes using perceptual grouping laws. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1, 43–50.
- Xu, Y., Sun, Z., Hoegner, L., Stilla, U., Yao, W., 2018. Instance segmentation of trees in urban areas from MLS point clouds using supervoxel contexts and graph-based optimization. *2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*, 1–5.
- Yao, W., Fan, H., 2013. Automated detection of 3d individual trees along urban road corridors by mobile laser scanning systems. *International Symposium on Mobile Mapping Technology (MMT)*.
- Yao, W., Polewski, P., Krzystek, P., 2017. Semantic labelling of ultra dense mls point clouds in urban road corridors based on fusing crf with shape priors. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W7, 971–976.
- Zhu, J., Gehring, J., Huang, R., Borgmann, B., Sun, Z., Hoegner, L., Hebel, M., Xu, Y., Stilla, U., 2020. TUM-MLS-2016: An annotated mobile LiDAR dataset of the TUM city campus for semantic point cloud interpretation in urban areas. *Remote Sensing*, 12(11), 1875.