



A Classification-Segmentation Framework for the Detection of Individual Trees in Dense MMS Point Cloud Data Acquired in Urban Areas

Martin Weinmann, Michael Weinmann, Clément Mallet, Mathieu Brédif

► To cite this version:

Martin Weinmann, Michael Weinmann, Clément Mallet, Mathieu Brédif. A Classification-Segmentation Framework for the Detection of Individual Trees in Dense MMS Point Cloud Data Acquired in Urban Areas. *Remote Sensing*, 2017, 9 (3), pp.277. 10.3390/rs9030277 . hal-01557698

HAL Id: hal-01557698

<https://hal.science/hal-01557698>

Submitted on 6 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Article

A Classification-Segmentation Framework for the Detection of Individual Trees in Dense MMS Point Cloud Data Acquired in Urban Areas

Martin Weinmann ^{1,2,*}, Michael Weinmann ³, Clément Mallet ² and Mathieu Brédif ²

¹ Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology (KIT), Englerstr. 7, D-76131 Karlsruhe, Germany

² Univ. Paris-Est, LASTIG MATIS, IGN, ENSG, 73 avenue de Paris, F-94160 Saint-Mandé, France; clement.mallet@ign.fr (C.M.); mathieu.bredif@ign.fr (M.B.)

³ Institute of Computer Science II, University of Bonn, Friedrich-Ebert-Allee 144, D-53113 Bonn, Germany; mw@cs.uni-bonn.de

* Correspondence: martin.weinmann@kit.edu; Tel.: +49-721-608-47302

Academic Editors: Norman Kerle, Markus Gerke, Sébastien Lefèvre, Clement Atzberger and Prasad S. Thenkabail

Received: 29 December 2016; Accepted: 9 March 2017; Published: 16 March 2017

Abstract: In this paper, we present a novel framework for detecting individual trees in densely sampled 3D point cloud data acquired in urban areas. Given a 3D point cloud, the objective is to assign point-wise labels that are both class-aware and instance-aware, a task that is known as instance-level segmentation. To achieve this, our framework addresses two successive steps. The first step of our framework is given by the use of geometric features for a binary point-wise semantic classification with the objective of assigning semantic class labels to irregularly distributed 3D points, whereby the labels are defined as “tree points” and “other points”. The second step of our framework is given by a semantic segmentation with the objective of separating individual trees within the “tree points”. This is achieved by applying an efficient adaptation of the mean shift algorithm and a subsequent segment-based shape analysis relying on semantic rules to only retain plausible tree segments. We demonstrate the performance of our framework on a publicly available benchmark dataset, which has been acquired with a mobile mapping system in the city of Delft in the Netherlands. This dataset contains 10.13 M labeled 3D points among which 17.6% are labeled as “tree points”. The derived results clearly reveal a semantic classification of high accuracy (up to 90.77%) and an instance-level segmentation of high plausibility, while the simplicity, applicability and efficiency of the involved methods even allow applying the complete framework on a standard laptop computer with a reasonable processing time (less than 2.5 h).

Keywords: mobile mapping systems; 3D point cloud; feature extraction; feature selection; semantic classification; semantic segmentation; instance-level segmentation; tree-like objects

1. Introduction

The automated analysis of data acquired in urban areas has become a topic of major interest in the fields of remote sensing, photogrammetry, computer vision and robotics. In recent years, particular attention has been paid to the analysis of data in the form of densely sampled 3D point clouds representing the measured counterpart of object surfaces in the local surrounding of the acquisition system. Due to the technological advancements, such densely sampled 3D point clouds can meanwhile be acquired directly by using mobile mapping systems (MMSs), which allow one to efficiently acquire large amounts of densely sampled 3D point cloud data, e.g., corresponding to street sections or city districts. Once respective 3D data have been acquired, different tasks may be addressed, such as a

semantic point cloud labeling with respect to different class labels [1–7] or the extraction of specific objects (e.g., building structures [8], roads or road inventory [9–12] or individual trees [13–15]) in the scene.

Among a diversity of objects, particularly trees play an important role in urban areas since they can provide measurable economic, environmental, social and health benefits [16,17]. As a prerequisite for urban planning, numerous municipalities and governmental agencies meanwhile focus on acquiring tree cadasters, which allow statements about the number of trees, the tree species and the physical and environmental effects of respective trees. Such tree cadasters can be derived from publicly available aerial and street view images from Google maps [18], but in many cases, they are derived from acquired MMS point cloud data. To foster research on the extraction of individual trees from MMS point cloud data as, e.g., shown in Figure 1, a special track within the recent IQmulus Processing Contest IQPC'15 has been initiated [13]. This special track focuses on two subtasks facing the challenges of an irregular point sampling, a high complexity of real-world scenes and a huge amount of data resulting from the acquisition of larger scenes. The first subtask is given by the binary classification of the 3D points of an MMS point cloud into “tree points” and “other points”, whereas the second subtask is given by the separation of the “tree points” into clusters corresponding to individual trees. The joint consideration of both subtasks represents an instance-level segmentation inferring both class labels and instance labels [19,20]. To evaluate the performance of respective approaches for both subtasks, a labeled benchmark dataset acquired with a mobile mapping system in the city of Delft in the Netherlands has been released. This dataset comprises about 10.13 M labeled 3D points, whereby the reference labels are given at point-level with respect to a binary classification distinguishing “tree points” and “other points”. The results of a segmentation of individual trees can easily be verified by visual inspection. Regarding the IQPC'15, a diversity of proposed approaches for both subtasks relies on a voxelization of 3D space. For reasons of accuracy, however, it would be desirable to have an efficient end-to-end processing pipeline for individual tree extraction from MMS point cloud data that is scalable towards the processing of large datasets without involving a voxelization of 3D space.

In this paper, we focus on the extraction of individual trees from densely sampled 3D point cloud data. We present a novel two-step framework that addresses (1) semantic classification by assigning semantic class labels to irregularly distributed 3D points and (2) semantic segmentation by separating individual objects within the labeled 3D points. More specifically, the first step of our framework is given by the detection of tree-like objects, which is achieved via a binary classification distinguishing 3D points belonging to tree-like objects (“tree points”) from 3D points belonging to non-tree-like objects (“other points”). The second step only relies on those 3D points belonging to tree-like objects, and it is given by a segmentation of individual trees within these 3D points. To improve efficiency, this step involves a 2D projection and a mean shift segmentation, which are applied to a downsampled version of the 3D points belonging to tree-like objects. Since misclassifications resulting from the first step should not significantly affect the second step, the latter also involves a segment-based shape analysis to only retain plausible tree segments. We demonstrate the performance of our framework by presenting the respective results obtained for the IQPC'15 benchmark dataset.

This paper represents an extended version of [21], whereby the extension is given by a more comprehensive analysis comprising:

- feature subsets that are selected manually,
- feature subsets that are derived automatically via feature selection techniques and
- an improved segment-based shape analysis relying on semantic rules.

In addition to drawing conclusions about which features are most relevant for the given classification task, we focus on increased efficiency via a parallelized memory-efficient implementation, which can be run on a standard laptop computer.

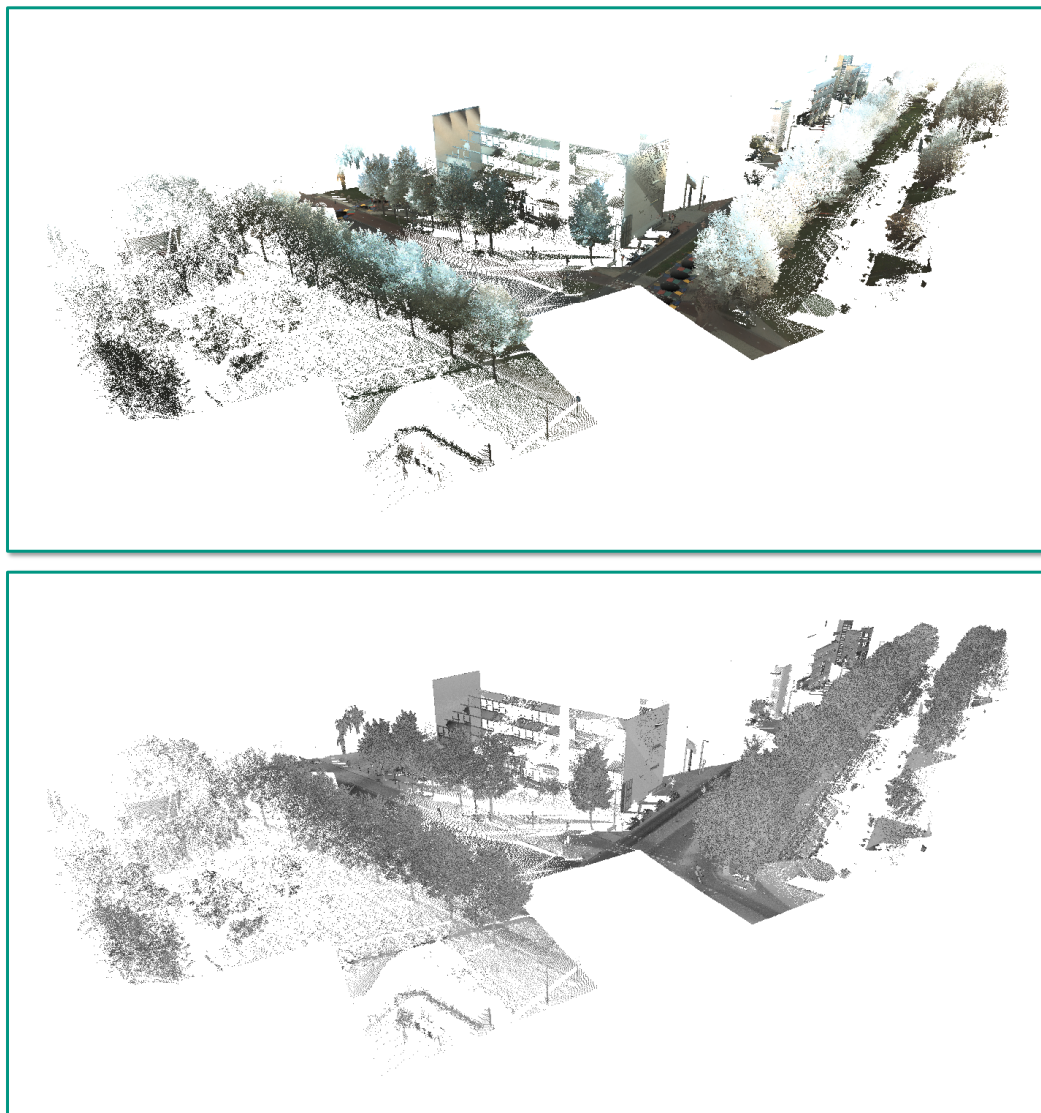


Figure 1. The MMS point cloud dataset released in the scope of the IQmulus Processing Contest IQPC'15: 3D point cloud colored with respect to the provided RGB values (**top**) and 3D point cloud colored with respect to the provided reflectance information (**bottom**).

After briefly summarizing related work (Section 2), we describe our novel two-step framework for extracting individual trees from densely sampled 3D point cloud data (Section 3). Subsequently, we demonstrate the performance of our framework on a publicly-available benchmark dataset (Section 4). This is followed by a discussion of the derived results with respect to accuracy, robustness and efficiency (Section 5) and a summary of the strengths and limitations of our framework and the involved methods. Finally, we provide concluding remarks and suggestions for future work (Section 6).

2. Related Work

In recent years, the semantic interpretation of 3D point cloud data has been addressed by many investigations. Among a variety of research directions, particular interest has been paid to (1) a semantic classification, which aims at assigning a semantic class label to each point of a given 3D point cloud [6,22], and (2) a semantic segmentation, which aims at providing a meaningful partitioning of a given 3D point cloud into smaller, connected subsets corresponding to objects of interest or to parts of these [23,24].

2.1. Semantic Classification

The classical processing pipeline for the semantic classification of 3D point cloud data may be decomposed into different components addressing neighborhood selection, feature extraction (and optionally feature selection) and classification [22]. In the following, we briefly summarize related work with respect to these components.

The local neighborhood of a considered 3D point X_0 is often defined by considering all 3D points within a spherical neighborhood [25,26] or within a cylindrical neighborhood [27]. In this regard, the neighborhood size is commonly defined by involving prior knowledge about the scene and/or the data, and the same value for the scale parameter is typically used for all points of the 3D point cloud. However, as demonstrated in recent investigations [6,22], different structures in the scene may favor a different neighborhood size. Accordingly, data-driven approaches for optimal neighborhood size selection have been presented, which focus on a local adaptation of the neighborhood size with respect to the local 3D structure. Among these approaches, the most promising ones are represented by an approach focusing on the local surface variation [28], an approach relying on curvature, point density and noise of normal estimation [29,30], dimensionality-based scale selection [31] and eigenentropy-based scale selection [6,22].

In contrast to optimizing the neighborhood size in order to derive features with improved distinctiveness, it has been proposed to describe the local 3D structure at different scales and thus also how the local 3D structure changes across these scales. Respective approaches typically focus on extracting geometric features from multiple spherical neighborhoods of different scales [32] or from multiple cylindrical neighborhoods of different scales [33]. However, some approaches also focus on extracting geometric features from multiple neighborhoods of different scales and types, e.g., from a combination of cylindrical and spherical neighborhoods [34], or from different entities in the form of voxels, blocks and pillars [3] or in the form of spatial bins, planar segments and local neighborhoods [35].

The features themselves are commonly derived by describing geometric characteristics of the defined local neighborhood. Many investigations rely on the use of local 3D shape features derived from the 3D structure tensor [28,36], as such features are relatively simple and allow rather intuitive descriptions (e.g., with respect to linear, planar or volumetric structures in the scene). To address further geometric characteristics of the local neighborhood, the feature vector is typically extended by adding complementary features, which are for instance given by angular statistics [1], height and plane characteristics [37,38], low-level 3D and 2D features [6] or moments and height features [7].

Among the extracted features, some may be more and others less relevant to the classification task. However, the less relevant features as well as the redundant features may have a detrimental effect on the predictive accuracy of the involved classifier, which is commonly referred to as the Hughes phenomenon [39]. Furthermore, it should be taken into account that the use of many features typically increases the computational burden with respect to processing time and memory consumption and, consequently, it seems desirable to involve methods for feature selection. Regarding the classification of 3D point cloud data, feature selection has for instance been addressed by filter-based feature selection methods evaluating (1) feature-class relations to reason about relevant features and (2) partly also feature-feature relations to remove redundancy [22,40]. Furthermore, wrapper-based feature selection methods interacting with a classifier have been applied [37,41].

Finally, the extracted features are provided as input for classification. In the simplest form, the classification of a considered 3D point X_0 only relies on the respective feature vector describing the local 3D structure at X_0 . For that purpose, a variety of classifiers relying on different learning principles may be used [6,22]. Respective classifiers are meanwhile available in a variety of software tools, and they can easily be applied by non-expert users. Due to the separate consideration of each 3D point, however, the derived labeling typically reveals a noisy behavior. To derive a rather smooth labeling, smooth labeling techniques [42] or contextual classification approaches may be applied.

The latter classify a 3D point X_0 based on the respective feature vector as well as those feature vectors and labels of neighboring points. Accordingly, interactions among 3D points within the local neighborhood of X_0 have to be modeled, which can be done by following different strategies, e.g., focusing on the use of Associative Markov Networks (AMNs) [1], Non-Associative Markov Networks (N-AMNs) [43], Conditional Random Fields (CRFs) [33,44–46] or more sophisticated inference procedures [2,47]. However, inferring interactions among neighboring points typically corresponds to an increased computational burden and to the requirement of a larger amount of training data to train the respective classifier.

Besides the classical processing pipeline, recent effort also involves Convolutional Neural Networks (CNNs) adapted to 3D data. Respective 3D-CNNs for instance focus on predicting an object class label given a 3D point cloud segment containing a single object [48,49]. To classify a 3D point cloud, a network architecture has been proposed that comprises an encoder and a decoder part [50]. Furthermore, a 3D-CNN has been presented to classify each 3D point of a point cloud by considering a voxel-occupancy grid corresponding to the respective local neighborhood [51]. A similar approach relying on a voxel-occupancy grid for representing a local neighborhood has been proposed in [52]. While the 3D-CNNs tend to outperform conventional approaches [53], they typically require a large amount of training data. Furthermore, the network architecture and its internal settings need to be defined, which is typically performed heuristically.

In the scope of this paper, we focus on a classical processing pipeline for the semantic classification of 3D point cloud data. Thereby, we take into account that recent benchmark datasets for multi-class classification of densely sampled 3D point cloud data contain at least one class referring to vegetation [1,4,53,54] and that the extraction of individual trees from densely sampled 3D point cloud data has become a topic of major interest [13]. The latter has recently been addressed by an approach for point-wise classification with respect to “tree points” and “other points” [14]. This approach focuses on defining a 2D probability matrix on a horizontally oriented plane, where each entry represents a probability value derived from the local point density. Tree trunks are expected to correspond to high probability values, and consequently, tree trunks are indicated by local maxima of the 2D probability matrix. Finally, further points are assigned to the tree trunks if they appear in the close proximity. In contrast to this approach, we intend to test the capability of a classical processing pipeline for the semantic classification of densely sampled 3D point cloud data with respect to “tree points” and “other points”. Thereby, we also focus on the simplicity and applicability of the involved methods so that non-expert end-users can apply the complete framework on a standard laptop computer without requiring expert knowledge about single methods. For this reason, we intend to avoid a heuristic specification of local neighborhoods and therefore apply the data-driven approach of eigenentropy-based scale selection [6,22]. Based on the derived local neighborhoods, we calculate rather intuitive, low-level geometric 3D and 2D features that are expected to be sufficient to derive reasonable classification results. Thereby, we define different feature sets selected either manually or automatically as input for the classification task. The latter is kept relatively simple by using a standard Random Forest (RF) classifier [55], since we expect that contextual classification will neither be possible for the given amount of training data nor allow a processing on a standard laptop computer.

2.2. Semantic Segmentation

For point cloud segmentation, different approaches may be applied [24]. Some of these approaches first perform an oversegmentation of the given 3D point cloud and subsequently merge neighboring segments with similar characteristics. In contrast, other approaches start with seed points and perform a region growing. After applying any of the segmentation approaches, the derived segments should correspond to meaningful objects or object parts. In [56], for example, it is proposed to apply a segmentation approach that relies on surface growing so that the derived segments can be described via a variety of geometric, radiometric and topological features. These features, in turn, allow a segment-wise classification, e.g., by applying a Support Vector Machine (SVM) classifier. Since

a generic segmentation approach operating at point-level typically results in a high computational burden, a voxelization of 3D space is often introduced. One of these approaches has been presented in [57] where the main idea is to use a voxelization of the considered 3D point cloud, apply a subsequent supervoxel segmentation and classify the derived segments by exploiting a set of geometric and radiometric features.

Instead of focusing on the segmentation of a variety of objects in 3D point cloud data, we are particularly interested in the segmentation of individual trees where each derived segment should correspond to an individual tree (i.e., each segment should cover both the foliage and the trunk of an individual tree). In this regard, many investigations also rely on a voxelization of 3D space. Based on such a voxelization, the properties of neighboring voxels may be evaluated to derive voxel groups corresponding to potential trees in the scene [58]. Alternatively, the voxelization of 3D space can be followed by deriving connected components and separating these connected components further if they contain multiple clusters [13]. Furthermore, it has been proposed to perform a downsampling and retiling of a given 3D point cloud via voxelization so that a subsequent 2D gridding allows finding local maxima in point density, which indicate potential tree locations [15]. For these potential tree locations, octree-based region growing and thresholding techniques may be applied to derive segments corresponding to individual trees. A different strategy has been followed by first deriving a 2D accumulation map corresponding to a horizontally oriented plane and then extracting respective features allowing the separation of natural objects, such as trees, from man-made objects [59]. Those 3D points corresponding to natural objects are subsequently transferred to a voxel space, where individual trees are segmented by applying a normalized cut segmentation based on the voxel structure [60].

In contrast to those approaches involving a voxelization of 3D space, there are also approaches that focus on segmenting individual trees from the original data at point-level. In this regard, it has for instance been proposed to use the 3D Hough transform and a surface growing algorithm in order to segment a given 3D point cloud into planar regions [61,62]. As larger planes typically correspond to man-made objects, the respective 3D points can be removed so that the remaining small segments as well as those 3D points that were not segmented can be merged via connected component analysis. Based on geometric features, further non-vegetation objects can be detected and removed. A different approach relies on the calculation of geometric descriptors for each 3D point, the projection of these descriptors onto a horizontally oriented 2D accumulation map and the consideration of a spatial filtering to derive individual tree segments [63].

In some cases, a pre-classification with respect to “tree points” and “other points” is already available. Then, individual trees can directly be segmented at point-level by deriving connected components for those 3D points categorized as “tree points”, whereby connected components are further split via an upward and downward growing algorithm if there are multiple seeds at a height between 0.5 m and 1 m [13,64]. Furthermore, it is possible to directly apply a standard clustering technique, such as a *k*-means clustering or hierarchical clustering [65], or the mean shift algorithm [66]. Particularly the mean shift algorithm has been applied for point cloud segmentation [67–70], since it operates in a data-driven manner without the need for defining a specific geometric model or involving prior knowledge about the number of expected modes. Such a data-driven point cloud segmentation can however be computationally demanding, particularly for the consideration of larger 3D point clouds. Accordingly, it is often desirable to improve computational efficiency, which can for instance be achieved by applying the mean shift algorithm on a 2D projection of the considered 3D point cloud as proposed in the context of tomographic SAR data processing [71]. For a ground-based acquisition of 3D point cloud data as, e.g., given when using mobile mapping systems, a significantly higher point density can be expected so that further strategies for improving computational efficiency are required.

In the scope of this paper, we assume that the first step of our framework provides an appropriate pre-classification with respect to “tree points” and “other points”. Accordingly, we can directly focus on a separation of “tree points” with respect to individual trees. For the sake of simplicity and applicability, we focus on the use of a data-driven point cloud segmentation at point-level via

the mean shift algorithm, whereby we introduce an adaptation towards data-intensive processing. To account for misclassifications resulting from the first step of our framework, we furthermore consider plausibility checks on the basis of segment-wise features and segment-based shape analysis.

3. Methodology

In this paper, we present a two-step framework for the detection of individual trees in dense 3D point cloud data acquired in urban areas. The first step of the framework is given by a semantic classification in terms of assigning semantic class labels to irregularly distributed 3D points (Section 3.1), whereas the second step is given by a semantic segmentation in terms of separating individual objects within the labeled 3D points (Section 3.2). For both steps, we focus on a data processing at point-level to achieve accurate results.

3.1. Detection of Tree-Like Structures via Semantic Classification

The first step of our framework is given by a point-wise, binary semantic labeling of a given 3D point cloud, whereby the labels are represented by “tree points” and “other points”. In general, such a semantic classification relies on an appropriate description of each considered 3D point X_0 , for which we focus on the use of geometric features. Accordingly, we first focus on the recovery of a suitable local neighborhood for each 3D point and then use those 3D points within the recovered local neighborhood for the extraction of low-level geometric 2D and 3D features (Section 3.1.1). Among the extracted features, there might be more and less relevant ones, so that it may be desirable to test different feature sets, which are selected either manually or automatically (Section 3.1.2), and draw conclusions about their absolute and relative performance with respect to the classification task (Section 3.1.3). As shown in Figure 2, we intend to apply the first step of our framework in order to retain “tree points” and discard “other points”.

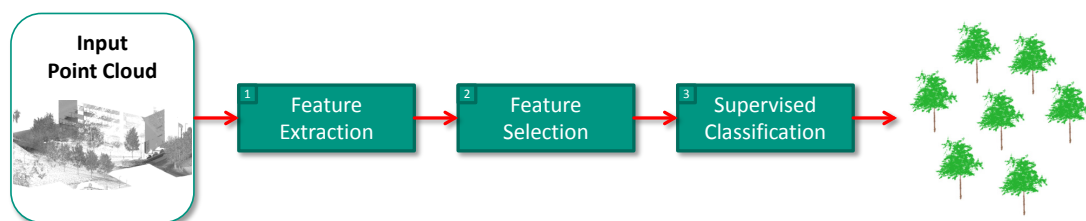


Figure 2. Illustration of the first step of our framework given by a semantic classification of “tree points” and “other points” in dense 3D point cloud data acquired in urban areas.

3.1.1. Feature Extraction

To adequately describe the local 3D structure at a considered 3D point X_0 with geometric features, spatial relationships among those 3D points within a specific local neighborhood of X_0 are typically quantified via handcrafted features. Accordingly, a suitable local neighborhood has to be recovered first, and the respective 3D points within that local neighborhood are subsequently used to extract the geometric features.

Generally, a variety of neighborhood definitions may be used such as a spherical neighborhood [25,26] or a cylindrical neighborhood [27]. For densely sampled 3D point cloud data acquired with mobile mapping systems, a spherical neighborhood definition is to be preferred to a cylindrical neighborhood definition, since different objects can be expected at different heights. To parameterize a spherical neighborhood for a considered 3D point X_0 , we can rely on a radius [25] or on the k nearest neighbors of X_0 [26]. Since the latter allows for more flexibility with respect to the absolute size of the local neighborhood, we consider this option. However, it often remains

challenging to find a suitable scale parameter k without using prior knowledge about the scene and/or the data. Furthermore, recent investigations clearly revealed that using the same value of the scale parameter for all points of a densely sampled 3D point cloud is significantly outperformed by allowing for a local data-driven adaptation of the scale parameter for each individual 3D point [6,22]. In these investigations, a well-suited generic approach for automatically selecting an optimal scale parameter k_{opt} for each 3D point \mathbf{X}_0 individually has been presented with eigenentropy-based scale selection and been proven to outperform a variety of other approaches. Thereby, for different scales k , the 3D covariance matrix also known as the 3D structure tensor and its normalized eigenvalues λ_j with $j \in \{1, 2, 3\}$, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$ are derived from the 3D coordinates of all 3D points within the neighborhood of \mathbf{X}_0 . Using the normalized eigenvalues λ_j , we can define the measure of eigenentropy E_λ via the Shannon entropy and as a function of the scale parameter k according to:

$$E_\lambda(k) = - \sum_{j=1}^3 \lambda_j(k) \ln \{ \lambda_j(k) \} \quad (1)$$

and this measure $E_\lambda(k)$ indicates the order/disorder of 3D points within the local neighborhood comprising the 3D point \mathbf{X}_0 and its k nearest neighbors. Since we favor a robust geometric description of the local 3D structure, we are interested in locally adapting the neighborhood size so that a minimum disorder of 3D points within the local neighborhood is achieved. This can be done by minimizing the eigenentropy E_λ across different scales k and selecting the scale k_{opt} corresponding to the minimal E_λ as the optimal neighborhood size for the considered 3D point \mathbf{X}_0 . Following our previous investigations [6,22], we consider all possible scale parameters between $k_{\text{min}} = 10$ and $k_{\text{max}} = 100$ with $\Delta k = 1$.

Exploiting the derived locally optimized neighborhoods, the characteristics of neighboring 3D points can be described via geometric features. In this regard, we consider a set of 18 low-level geometric features proposed in [6,22]. These features can be categorized into 3D features and 2D features.

Among the 3D features, there are eight local 3D shape features represented by linearity L_λ , planarity P_λ , sphericity S_λ , omnivariance O_λ , anisotropy A_λ , eigenentropy E_λ , sum of eigenvalues Σ_λ and local surface variation C_λ , which according to previous work [28,36] are defined as indicated in the upper part of Table 1. Furthermore, the 3D features comprise six geometric 3D properties of the considered local neighborhood, which are given by the height H of the considered 3D point \mathbf{X}_0 , the radius R_{3D} of the local neighborhood, the local point density ρ_{3D} represented by the ratio of 3D points within the local neighborhood and the volume of the local neighborhood, the verticality V relying on the local normal vector and the maximum difference ΔH as well as the standard deviation σ_H of the height values corresponding to those 3D points within the local neighborhood. The respective formulae are provided in the lower part of Table 1.

The use of 2D features is motivated by the fact that urban environments are characterized by an aggregation of man-made objects, which typically exhibit almost perfectly vertical structures (e.g., building facades, walls, poles or traffic signs). To describe such characteristics, we consider features relying on a projection of the considered 3D point \mathbf{X}_0 and its k_{opt} nearest neighbors onto a horizontally oriented plane. For the resulting 2D projections, the 2D structure tensor and its eigenvalues ξ_j with $j \in \{1, 2\}$ and $\xi_1 \geq \xi_2 \geq 0$ can be derived in analogy to the 3D case. Accordingly, we can define two local 2D shape features represented by the sum Σ_ξ and the ratio $R_\xi = \xi_2/\xi_1$ of eigenvalues. Furthermore, we may exploit the 2D projections to define two geometric 2D properties represented by the radius R_{2D} of the local 2D neighborhood and the local point density ρ_{2D} , respectively.

If the involved acquisition system also allows capturing additional information (e.g., reflectance information I or color information in the form of RGB values) which could be relevant for the classification task, that information may be used to define further features.

Table 1. The involved local 3D shape features (upper part) and the involved geometric 3D properties (lower part).

Feature	Formulae
Linearity	$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planarity	$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1}$
Sphericity	$S_\lambda = \frac{\lambda_3}{\lambda_1}$
Omnivariance	$O_\lambda = \sqrt[3]{\prod_{j=1}^3 \lambda_j}$
Anisotropy	$A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1}$
Eigenentropy	$E_\lambda = - \sum_{j=1}^3 \lambda_j \ln(\lambda_j)$
Sum of eigenvalues	$\Sigma_\lambda = \sum_{j=1}^3 \lambda_j$
Local surface variation	$C_\lambda = \frac{\lambda_3}{\Sigma_\lambda}$
Height	$H = Z_0$
Radius	$R_{3D} = \ \mathbf{X}_{k_{opt}} - \mathbf{X}_0\ $
Local point density	$\rho_{3D} = \frac{k_{opt} + 1}{\frac{4}{3}\pi R_{3D}^3}$
Verticality	$V = 1 - \mathbf{n}_Z$
Height difference	$\Delta H = \max_{k=0..k_{opt}} Z_k - \min_{k=0..k_{opt}} Z_k$
Standard deviation of height values	$\sigma_H = \sqrt{\frac{1}{k_{opt}} \sum_{k=0}^{k_{opt}} Z_k - \bar{Z}}$

3.1.2. Feature Selection

To allow a statement about the suitability of the defined features, we select different feature sets as input for the subsequent classification task. First, we manually select feature sets, which are defined as follows:

- The feature set \mathcal{S}_{dim} contains the dimensionality features:

$$\mathcal{S}_{dim} = \{L_\lambda, P_\lambda, S_\lambda\} \quad (2)$$

- The feature set $\mathcal{S}_{EV,3D}$ contains the eight local 3D shape features:

$$\mathcal{S}_\lambda = \{L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda\} \quad (3)$$

- The feature set \mathcal{S}_{3D} contains all defined 3D features, i.e., the local 3D shape features and the geometric 3D properties:

$$\mathcal{S}_{3D} = \{L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda, H, R_{3D}, \rho_{3D}, V, \Delta H, \sigma_H\} \quad (4)$$

- The feature set \mathcal{S}_{3D+2D} contains all 3D and 2D features relying on the k -NN neighborhood, i.e., the local 3D shape features, the geometric 3D properties, the local 2D shape features and the geometric 2D properties:

$$\mathcal{S}_{3D+2D} = \{L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda, H, R_{3D}, \rho_{3D}, V, \Delta H, \sigma_H, \Sigma_{\tilde{\zeta}}, R_{\tilde{\zeta}}, R_{2D}, \rho_{2D}\} \quad (5)$$

- The feature set $\mathcal{S}_{3D+2D+I}$ contains all 3D and 2D features as well as the given reflectance information:

$$\mathcal{S}_{3D+2D+I} = \{L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda, H, R_{3D}, \rho_{3D}, V, \Delta H, \sigma_H, \Sigma_\xi, R_\xi, R_{2D}, \rho_{2D}, I\} \quad (6)$$

- The feature set $\mathcal{S}_{3D+2D+I+RGB}$ contains all defined 3D and 2D features as well as reflectance and color information:

$$\mathcal{S}_{3D+2D+I+RGB} = \{L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda, H, R_{3D}, \rho_{3D}, V, \Delta H, \sigma_H, \Sigma_\xi, R_\xi, R_{2D}, \rho_{2D}, I, R, G, B\} \quad (7)$$

Subsequently, we automatically select feature sets by applying commonly used methods for filter-based feature selection [72]. One of these feature sets is derived via Correlation-based Feature Selection (CFS) [73], where the main idea is given by evaluating relations between features and classes as well as relations among features in order to discriminate between relevant, irrelevant and redundant features. This is done by defining random variables X_i for the features and C for the class labels, so that the relevance R of a feature subset comprising n features can be expressed as:

$$R(X_{1...n}, C) = \frac{n\bar{\rho}_{XC}}{\sqrt{n + n(n-1)\bar{\rho}_{XX}}} \quad (8)$$

where $\bar{\rho}_{XC}$ represents the average correlation between features and classes and $\bar{\rho}_{XX}$ represents the average correlation between different features. Thereby, the correlation metric is determined via the measure of symmetrical uncertainty SU [74], which is defined as:

$$SU(X, Y) = 2 \frac{E(X) + E(Y) - E(X|Y)}{E(X) + E(Y)} = 2 \frac{MI(X, Y)}{E(X) + E(Y)} \quad (9)$$

for two random variables X and Y , where $E(\cdot)$ represents the entropy and $MI(\cdot, \cdot)$ represents the mutual information. The most suitable feature subset maximizes the relevance R . Accordingly, a search in the feature subset space has to be carried out, which is performed by iteratively adding or removing a feature from the feature subset until the relevance R converges to a stable maximum.

As a second approach for automatic feature subset selection, we use the Fast Correlation-Based Filter (FCBF) [75], where the SU is used to rank features with respect to their correlation with the respective label vector. Thereby, relevant features are indicated by an SU value above a certain threshold. Subsequently, the SU among features is compared to the SU between features and classes in order to detect and remove redundant features, so that only the predominant features are kept.

Finally, for each of the considered feature sets, the respective features are concatenated to a feature vector. To account for the fact that the features may address different quantities with possibly different units and a different range of values, a subsequent normalization across all feature vectors is carried out. This normalization maps the values of each dimension to the interval $[0, 1]$. For that purpose, the minimum and maximum values of each dimension are selected based on the examples in the training data. The examples in the test data are mapped accordingly, and those values outside the interval $[0, 1]$ are mapped to the closest interval border.

3.1.3. Supervised Classification

In the scope of our work, the derived normalized feature vectors serve as input to a binary point-wise classification distinguishing “tree points” and “other points”. For such a classification task, a good trade-off between accuracy and efficiency can be obtained by using a Random Forest (RF) classifier [55], which is a representative of modern discriminative methods and easy-to-use (and easy-to-tune) for non-expert users. The RF classifier consists of an ensemble of decision trees created via bootstrap aggregating (“bagging”) [76], i.e., numerous subsets of the training data are randomly

drawn with replacement, and an individual decision tree is trained for each subset. Thus, the trained decision trees are randomly different from each other so that, for a new feature vector, each decision tree casts an individual vote, and the majority vote hence represents a prediction with improved generalization and robustness [77].

3.2. Separation of Individual Trees via Semantic Segmentation

After the binary point-wise classification, we use the “tree points” and focus on a respective separation of these points into segments corresponding to individual trees. As a generic segmentation typically results in a high computational burden, we consider adaptations to retain an efficient approach for individual tree extraction, which is also scalable towards the processing of very large datasets, while still operating at point-level without requiring a voxelization of 3D space. More specifically, our approach mainly relies on insights gained in our previous investigations [21,78], and it is given by successively applying a downsampling of the “tree points” (Section 3.2.1), a 2D projection of the downsampled “tree points” (Section 3.2.2), a mean shift segmentation on the derived 2D projections (Section 3.2.3), an upsampling of the segmentation results to all “tree points” (Section 3.2.4), a refinement of the segmentation results via segment-based shape analysis (Section 3.2.5) and a localization of the individual trees in the considered scene (Section 3.2.6). As shown in Figure 3, we apply the second step of our framework only on the “tree points” and intend to derive plausible segments corresponding to individual trees.

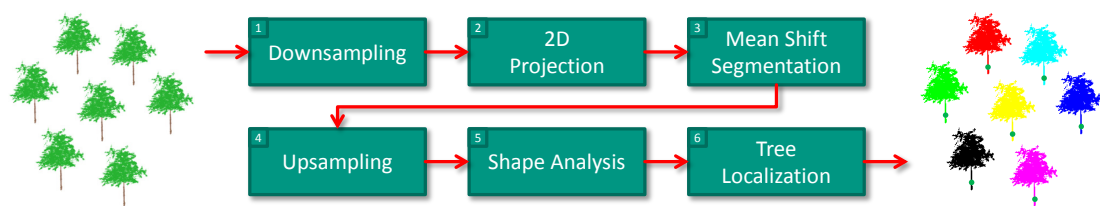


Figure 3. Illustration of the second step of our framework given by a semantic segmentation of “tree points” into segments corresponding to individual trees.

To improve computational efficiency for the subsequent parts, we initially discard those 3D points that are characterized by either a local vertical structure or a local horizontal structure, as we expect that the local structure should rather be cluttered for vegetation. Hence, we introduce a filtering that relies on the already calculated feature of verticality V (Section 3.1.1). Since this feature is normalized to the interval $[0, 1]$, low values (e.g., $V \in [0, T_1]$) as well as high values (e.g., $V \in [1 - T_1, 1]$) indicate almost horizontally oriented surfaces, whereas a value of $V \approx 0.5$ (e.g., $V \in [0.5 - T_2, 0.5 + T_2]$) indicates an almost vertical structure. Consequently, we apply a thresholding, whereby thresholds of $T_1 = 0.1$ and $T_2 = 0.2$ are selected heuristically based on the histogram of values for the feature of verticality V .

3.2.1. Downsampling

Since a generic segmentation of 3D point cloud data may be desirable, but computationally demanding, we focus on improving efficiency with respect to both processing time and memory consumption. In this regard, we focus on the expected characteristics of the acquired MMS point cloud data, where we can assume densely sampled object surfaces near the acquisition system. Accordingly, not all 3D points are required to appropriately describe respective objects in the scene, and the point density may therefore be decreased significantly while still being able to detect the objects of interest (in our case, represented by individual trees) in the respective 3D point cloud data.

In the scope of our work, we use a straightforward downsampling of the “tree points” by only keeping every ν -th point, whereby we heuristically select a parameter of $\nu = 10$ as proposed in [78].

Instead of such a manual selection of the parameter ν , a parameter tuning based on the local point density could be introduced [79].

3.2.2. 2D Projection

In addition to a downsampling of the 3D point cloud corresponding to the “tree points”, we take into account that, due to planning processes as well as human intervention in nature, the respective point distribution in urban areas is not completely random, but follows certain rules that can be considered as prior knowledge about the scene and the data. Considering individual trees as objects of interest, it is reasonable to assume a larger spacing and less overlap for individual trees in urban areas than for individual trees in forested areas. Furthermore, we may neglect the occurrence of dominant, co-dominant or dominated trees in urban environments, whereas this may be important in forested areas. Based on such prior knowledge about the scene and the data, we may assume that individual trees can still sufficiently be delineated if only a 2D projection of the downsampled “tree points” onto a horizontally oriented plane is considered.

3.2.3. Mean Shift Segmentation

For segmentation, we use the 2D projections of the downsampled “tree points” onto a horizontally oriented plane and apply the mean shift algorithm [66,80,81] to derive a meaningful partitioning with respect to individual trees. In general, the mean shift algorithm is an iterative statistical technique for locating the maxima/modes of a Probability Density Function (PDF) by only considering discrete data sampled from that PDF, i.e., there is no need to recover the PDF itself. A further advantage of the mean shift algorithm is that neither assumptions on a specific geometric model nor prior knowledge about the number of expected modes are required.

More specifically, we treat the 2D projections of the downsampled “tree points” as discrete 2D data points sampled from an empirical PDF. For each data point \mathbf{x}_0 , the mean shift algorithm performs the following steps until convergence (up to numerical accuracy):

1. calculation of the weighted mean μ of all data points within a window centered at \mathbf{x}_0 and defined via a kernel \mathcal{K} (whereby the kernel is typically represented by an isotropic kernel such as a Gaussian kernel or an Epanechnikov kernel [81]),
2. definition of the mean shift vector \mathbf{m} from the difference between \mathbf{x}_0 and μ ,
3. movement of the data point \mathbf{x}_0 along the mean shift vector \mathbf{m} and
4. consideration of the resulting point as an update of the point \mathbf{x}_0 .

Accordingly, an iterative adaptive gradient ascent is performed, whereby the magnitude of the mean shift vector \mathbf{m} will be larger in those areas of low point density and smaller in those areas of high point density. The movement converges in stationary points, which correspond to regions of high point density and represent the modes of the underlying PDF. Each mode corresponds to a segment, and all data points leading to the same mode form the respective segment. Due to the consideration of 2D projections of the downsampled “tree points” in the scope of our work, the derived segments are expected to correspond to the individual trees in the considered scene.

It becomes obvious that the number of detected modes and thus the derived segmentation results strongly depend on the involved kernel. In our experiments, we intend to detect individual trees in the 2D projection of densely sampled 3D point cloud data, and we may therefore involve prior knowledge about the expected shape and size of these trees. For this reason, we select an isotropic Gaussian kernel, which is parameterized via the bandwidth parameter h indicating the kernel width. In the context of our work, the bandwidth parameter h has a physical meaning with respect to the expected size of trees in the scene, and it has a limited sensitivity, as slight variations of h will typically not change the segmentation results too much. Based on heuristic tests, we selected a value of $h = 3.8$ m for our experiments [21,78].

3.2.4. Upsampling

When applying the generic mean shift segmentation to the 2D projection of the downsampled “tree points”, the segmentation results only refer to a subspace of the 3D points classified as “tree points”, whereas a respective upsampling of the derived segmentation results to those 3D points obtained after classification would be desirable. To achieve such an upsampling, we apply a rather intuitive, simple and straightforward approach. To each 3D point classified as a “tree point”, this approach assigns the segment label of the closest 3D point of the downsampled “tree points”.

3.2.5. Shape Analysis

Among the derived segments, there might be some that do not correspond to individual trees. To remove such irrelevant segments, we focus on a segment-based shape analysis relying on semantic rules. These semantic rules, in turn, rely on either segment statistics or on a segment-based extraction of geometric features (e.g., the low-level geometric features presented in Section 3.1.1), whereby the features are now extracted on the basis of a segment as the respective neighborhood.

1. The first semantic rule focuses on discarding smaller segments comprising only relatively few 3D points. This is motivated by the fact that, due to the data acquisition with a mobile mapping system, the meaningful segments corresponding to individual trees should comprise many densely sampled 3D points, whereas small segments are not likely to correspond to the objects of interest, i.e., individual trees. Using a superscript ^s to indicate segment-wise features, we apply this semantic rule to discard segments that comprise less than $N^s = 1000$ points.
2. The second semantic rule focuses on failure cases observed in recent investigations, e.g., in the form of misclassifications of 3D points corresponding to building facades, which for instance becomes visible in the classification results for one of the approaches presented in [13]. In this regard, we take into account that building facades are characterized by an almost line-like structure in their 2D projection onto a horizontally oriented plane. Accordingly, we may consider the ratio R_{ξ}^s of the eigenvalues of the 2D structure tensor, and we discard those segments that are rather elongated in the 2D projection by checking if R_{ξ}^s is below a certain threshold $T_{R_{\xi}^s}$. Thereby, we select the threshold heuristically to $T_{R_{\xi}^s} = 0.2$, which means that, for a segment corresponding to a tree, the smaller eigenvalue ξ_2^s has to be equal to or even above a value of 20% of the larger eigenvalue ξ_1^s , i.e., $R_{\xi}^s \geq 0.2$.
3. The third semantic rule focuses on discarding segments that exhibit a structure with almost no extent in the horizontal direction. This can be done by thresholding the products of the eigenvalues ξ_j^s of the 2D structure tensor and their sum Σ_{ξ}^s , where we assume that segments corresponding to individual trees are characterized by $\xi_1^s \Sigma_{\xi}^s \geq 1$ m and $\xi_2^s \Sigma_{\xi}^s \geq 1$ m.
4. The fourth semantic rule focuses on discarding segments that exhibit a low curvature C_{λ} where $C_{\lambda} < 0.07$, since such segments typically reveal planar structures.

All involved threshold parameters have been selected via a heuristic search and worked for all of our experiments.

3.2.6. Tree Localization

Finally, for all plausible tree segments, we define the location of the respective tree via the corresponding mode derived during the mean shift segmentation relying on 2D projections of the downsampled “tree points”.

4. Results

In the following, we first describe the involved benchmark dataset (Section 4.1). Subsequently, we present the derived results for semantic classification (Section 4.2) and semantic segmentation (Section 4.3).

4.1. Dataset

To evaluate the performance of our framework, we use the IQPC'15 benchmark dataset [13], which has been acquired in the vicinity of the campus of TU Delft in the Netherlands with the Fugro DRIVE-MAP system. This system allows acquiring spatial 3D data in the form of a 3D point cloud as well as the corresponding reflectance and color information (see Figure 1). In total, the dataset corresponds to 26 tiles (each covering an area of $25\text{ m} \times 25\text{ m}$) and comprises 10,126,500 labeled 3D points, whereby the labeling distinguishes “tree points” from “other points” (see Figure 4). About 1.78 M 3D points (17.6%) are labeled as “tree points”, and all remaining 3D points are labeled as “other points”. The provided reference labeling allows performance evaluation for a binary semantic classification of 3D points with respect to “tree points” and “other points”, whereas a separation of the “tree points” into clusters corresponding to individual trees has to be verified by visual inspection.

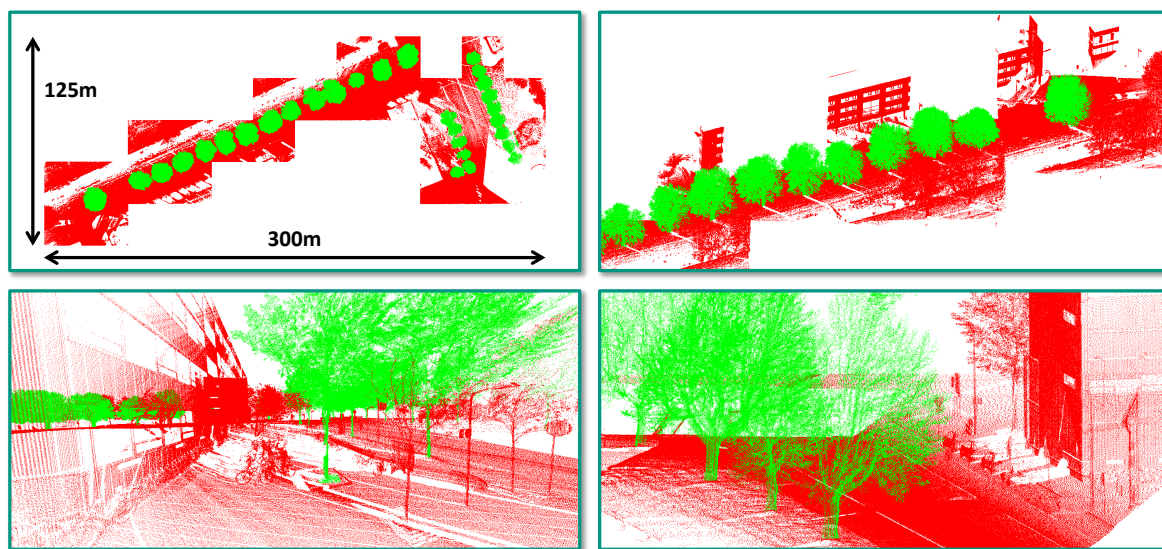


Figure 4. Visualization of the IQPC'15 benchmark dataset with about 10.13 M labeled 3D points (**top**: nadir view and side view; **bottom**: more detailed views): those 3D points categorized as “tree points” are colored in green, and those 3D points categorized as “other points” are colored in red [21,78].

4.2. Task 1: Semantic Classification

The first step of our framework focuses on the point-wise semantic classification of a given 3D point cloud with respect to “tree points” and “other points”. For this purpose, each 3D point is (1) assigned a local neighborhood of optimal size via eigenentropy-based scale selection, (2) described with a feature vector by concatenating the defined low-level geometric (and partially also radiometric) features, and (3) classified by the Random Forest (RF) classifier.

To train the RF classifier, we take into account that a sufficient number of representative training examples is required and that an unbalanced distribution of training examples per class might have a detrimental effect on the training process [77,82]. Hence, we randomly select 1000 training examples per class, i.e., the training set \mathcal{X} comprises 2000 labeled 3D points and their corresponding feature vectors. The remaining labeled 3D points of the dataset and their corresponding feature vectors are used as test set \mathcal{Y} . Furthermore, we perform a heuristic grid search to define the settings of the RF classifier. Thereby, the most important parameter is represented by the number N_T of involved decision trees, and this parameter is respectively selected to $N_T = 100$ for all considered feature sets.

For performance evaluation, we consider the global evaluation metrics of overall accuracy (OA) and Cohen's kappa coefficient (κ). Furthermore, we consider the class-wise evaluation metrics of recall (R) and precision (P). To derive objective results allowing one to compare the suitability of each feature set, we perform 20 repetitions of the corresponding classification task and report the mean value as

well as the standard deviation for each of the defined classification metrics. The derived classification results are provided in Table 2. A visualization of the classification results derived with the feature sets S_{3D+2D} and S_{FCBF} is shown in Figures 5 and 6.

Table 2. Mean value and standard deviation for the different evaluation metrics (averaged across 20 runs): the class C_1 comprises “tree points”, whereas the class C_2 comprises “other points”.

Feature Set	# Features	OA (%)	κ (%)	$R(C_1)$ (%)	$R(C_2)$ (%)	$P(C_1)$ (%)	$P(C_2)$ (%)
S_{dim}	3	74.33 ± 2.51	35.20 ± 2.37	73.24 ± 3.92	74.57 ± 3.82	38.32 ± 2.36	92.92 ± 0.66
S_λ	8	84.04 ± 1.51	56.53 ± 2.54	88.18 ± 2.16	83.15 ± 2.20	52.96 ± 2.78	97.07 ± 0.46
S_{3D}	14	90.08 ± 1.16	71.36 ± 2.43	96.21 ± 2.02	88.77 ± 1.76	64.85 ± 3.18	99.11 ± 0.46
S_{3D+2D}	18	90.02 ± 1.22	71.39 ± 2.68	97.08 ± 1.64	88.51 ± 1.69	64.51 ± 3.06	99.31 ± 0.38
$S_{3D+2D+I}$	19	88.03 ± 1.88	67.20 ± 3.89	98.15 ± 1.35	85.87 ± 2.50	60.03 ± 4.08	99.55 ± 0.32
$S_{3D+2D+I+RGB}$	22	84.74 ± 1.58	60.73 ± 2.88	99.59 ± 0.11	81.57 ± 1.94	53.67 ± 2.40	99.89 ± 0.03
S_{CFS}	2–4	90.72 ± 0.57	71.52 ± 1.28	89.30 ± 3.61	91.03 ± 1.24	68.12 ± 2.05	97.57 ± 0.79
S_{FCBF}	2–4	90.77 ± 0.50	71.66 ± 1.43	89.53 ± 4.36	91.03 ± 1.19	68.19 ± 2.06	97.62 ± 0.94

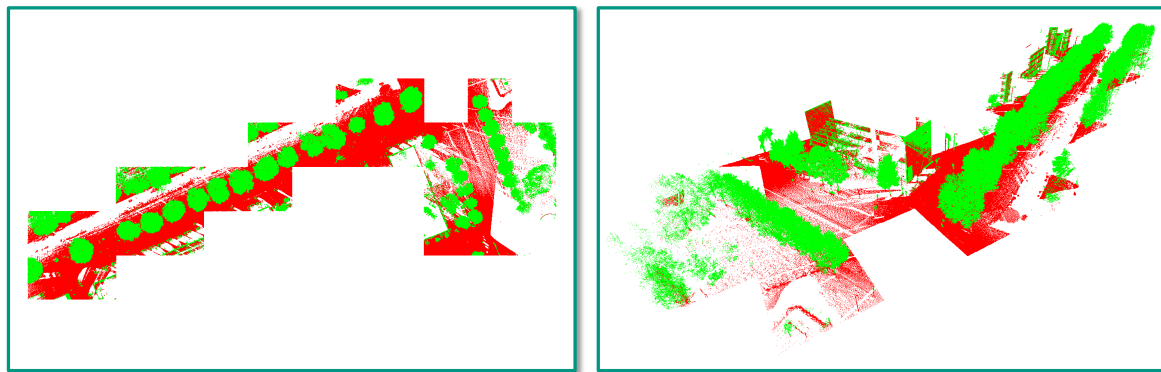


Figure 5. Visualization of a classification result derived with the feature set S_{3D+2D} comprising 18 low-level geometric 3D and 2D features (**left:** nadir view; **right:** side view): those 3D points categorized as “tree points” are colored in green, whereas those 3D points categorized as “other points” are colored in red (the classification result corresponds to an overall accuracy of 90.07% and a kappa value of 71.54%).

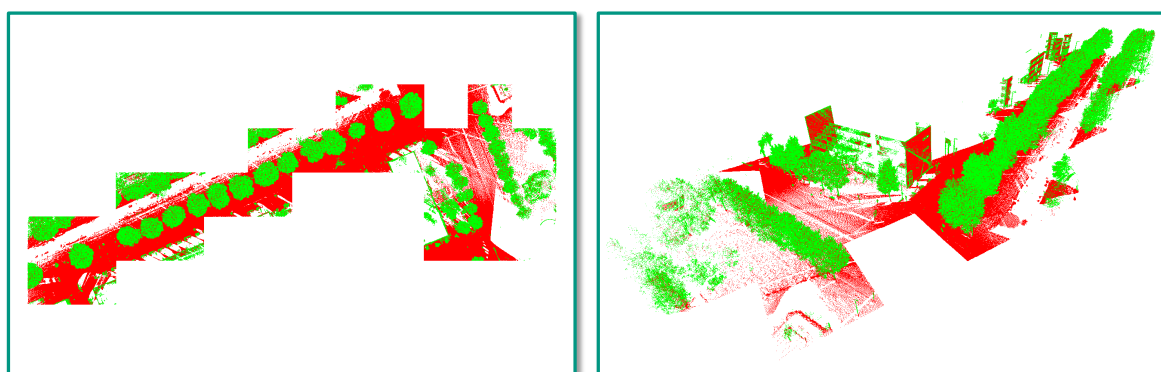


Figure 6. Visualization of a classification result derived with the feature set S_{FCBF} (**left:** nadir view; **right:** side view): those 3D points categorized as “tree points” are colored in green, whereas those 3D points categorized as “other points” are colored in red (the classification result corresponds to an overall accuracy of 90.71% and a kappa value of 72.14%).

4.3. Task 2: Semantic Segmentation

The second step of our framework focuses on the separation of the “tree points” with respect to individual trees. For this purpose, we involve: (1) a downsampling of the “tree points”; (2) a 2D projection onto a horizontally oriented plane; (3) a mean shift segmentation in 2D space; (4) an upsampling of the segmentation results to all “tree points”; (5) a segment-based shape analysis relying on semantic rules; and (6) a tree localization.

To obtain an impression about the quality of the segmentation results derived with the mean shift algorithm, a visualization of the derived segments is provided in Figures 7 and 8. Here, Figure 7 shows a segmentation relying on a classification result derived for the feature set \mathcal{S}_{3D+2D} (Figure 5), whereas Figure 8 shows a segmentation relying on a classification result derived for the feature set \mathcal{S}_{FCBF} (Figure 6). The effect of applying the different semantic rules during segment-based shape analysis is shown in Figures 9 and 10, respectively. Finally, a visualization highlighting the individual trees detected in the considered scene is provided in Figure 11. The estimated location of the respective trees is given with the corresponding modes derived during the mean shift segmentation (Figures 7 and 8).

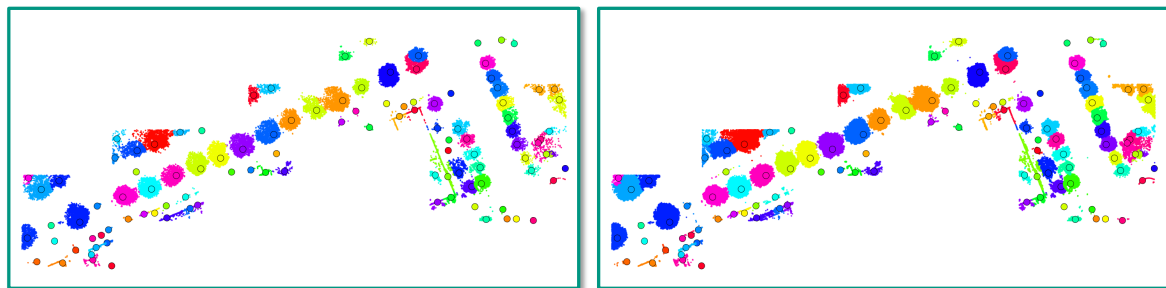


Figure 7. Visualization of a segmentation result relying on a classification based on the feature set \mathcal{S}_{3D+2D} (**left**: mean shift segmentation result; **right**: segmentation result after the upsampling to the “tree points”): the segment modes are indicated with a circle in the respective color (the segmentation result corresponds to the classification result depicted in Figure 5).

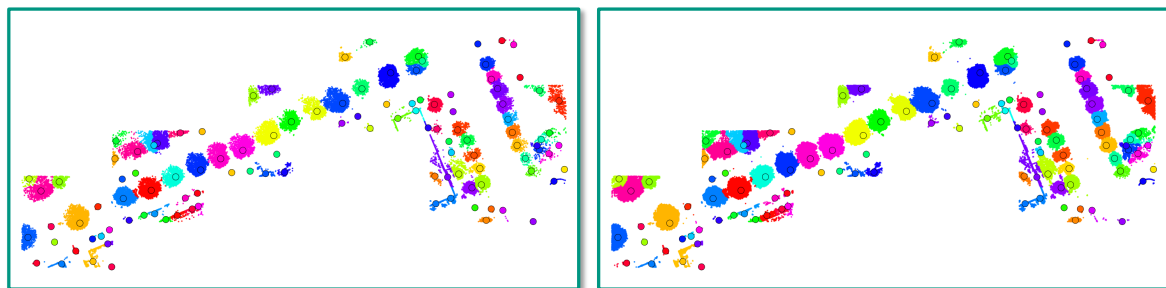


Figure 8. Visualization of a segmentation result relying on a classification based on the feature set \mathcal{S}_{FCBF} (**left**: mean shift segmentation result; **right**: segmentation result after the upsampling to the “tree points”): the segment modes are indicated with a circle in the respective color (the segmentation result corresponds to the classification result depicted in Figure 6).

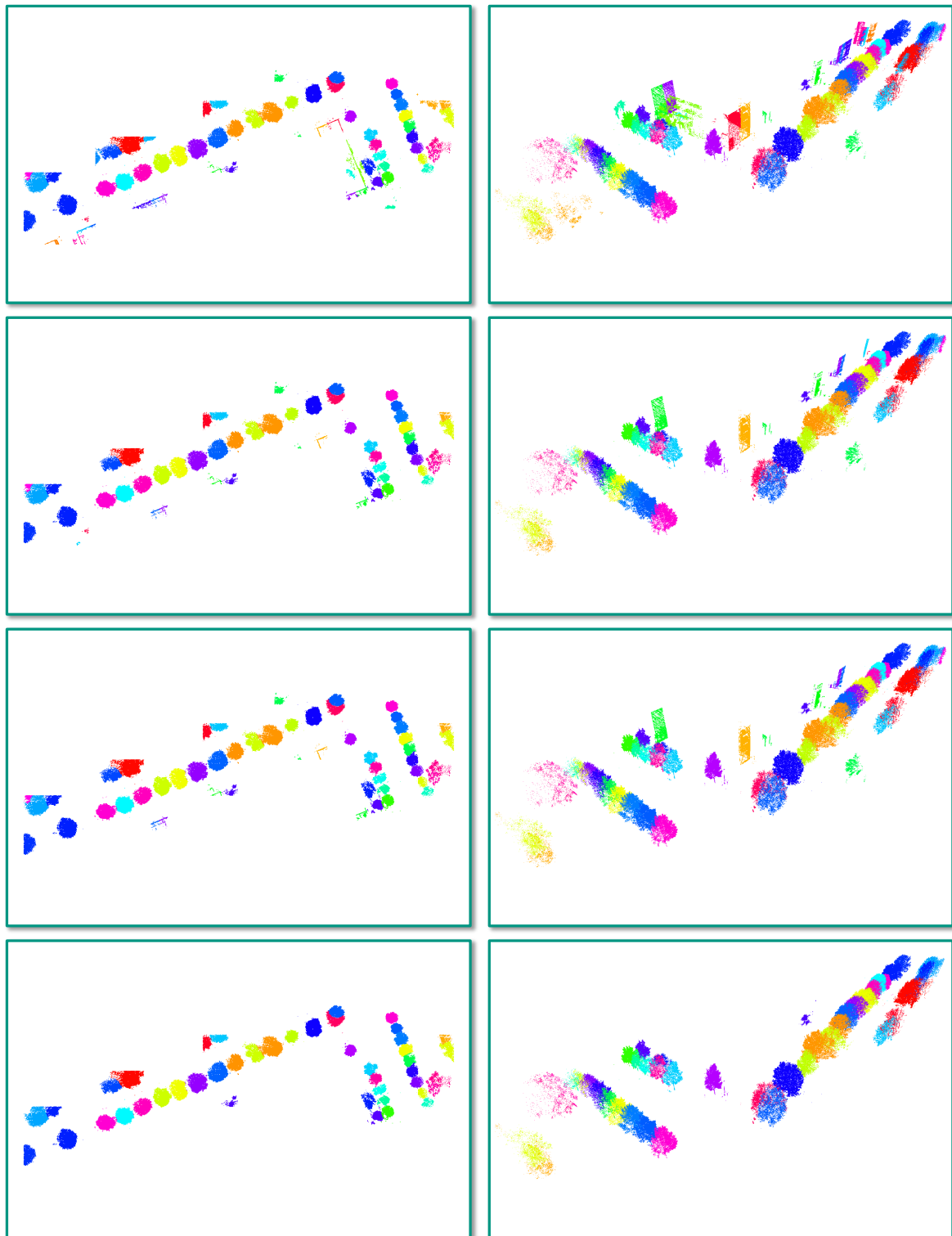


Figure 9. Segment-based shape analysis corresponding to the segmentation depicted in Figure 5 and relying on a classification based on the feature set \mathcal{S}_{3D+2D} (**left**: nadir view; **right**: side view): segments with $N^s \geq 1000$ points (first row) and segments that additionally satisfy the constraints $R_\xi^s \geq 0.2$ (second row), $\xi_j^s \Sigma_\xi^s \geq 1$ m for $j = 1, 2$ (third row) and $C_\lambda \geq 0.07$ (fourth row).

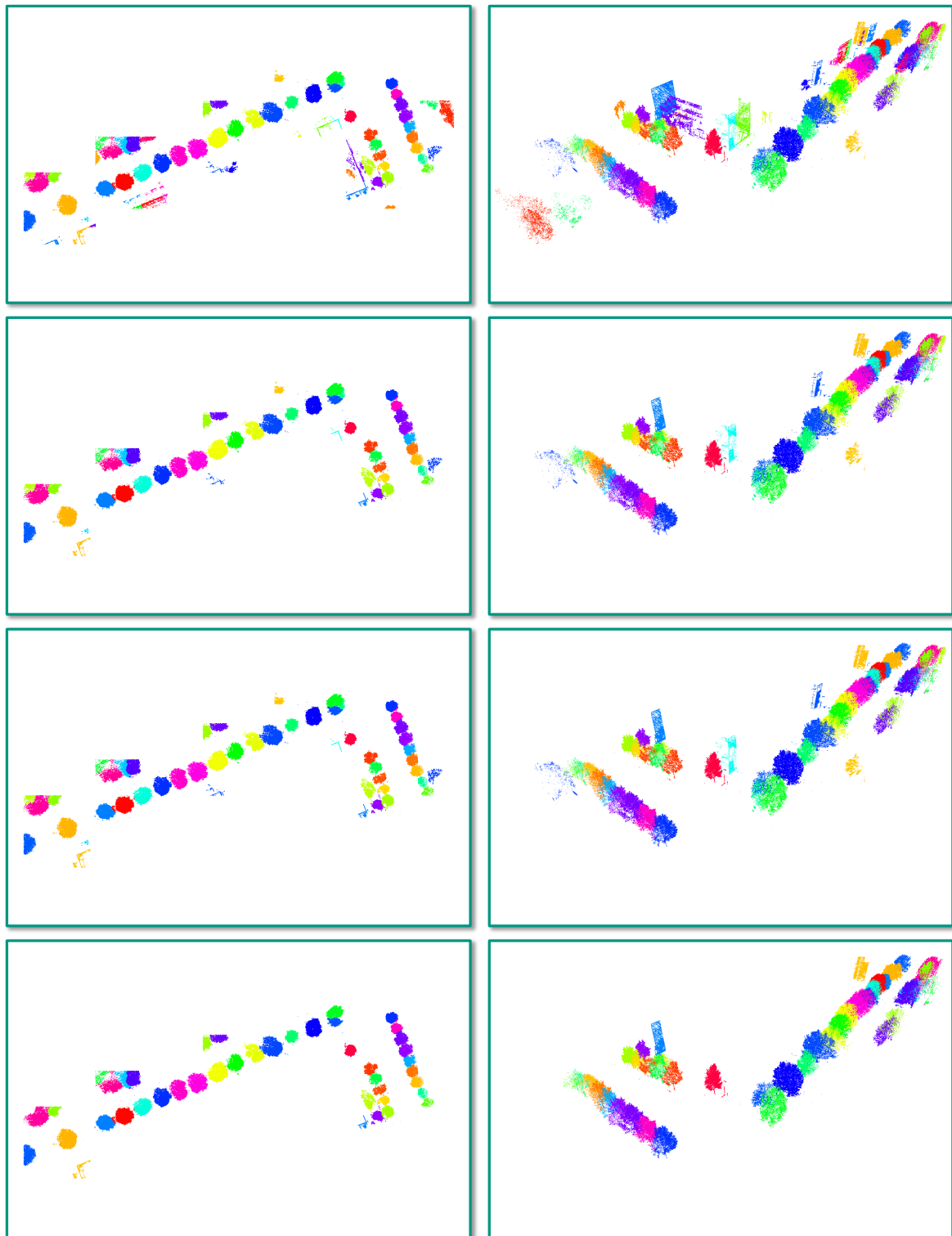


Figure 10. Segment-based shape analysis corresponding to the segmentation depicted in Figure 6 and relying on a classification based on the feature set $\mathcal{S}_{\text{FCBF}}$ (**left**: nadir view; **right**: side view): segments with $N^s \geq 1000$ points (first row) and segments that additionally satisfy the constraints $R_\xi^s \geq 0.2$ (second row), $\xi_j^s \Sigma_\xi^s \geq 1$ m for $j = 1, 2$ (third row) and $C_\lambda \geq 0.07$ (fourth row).

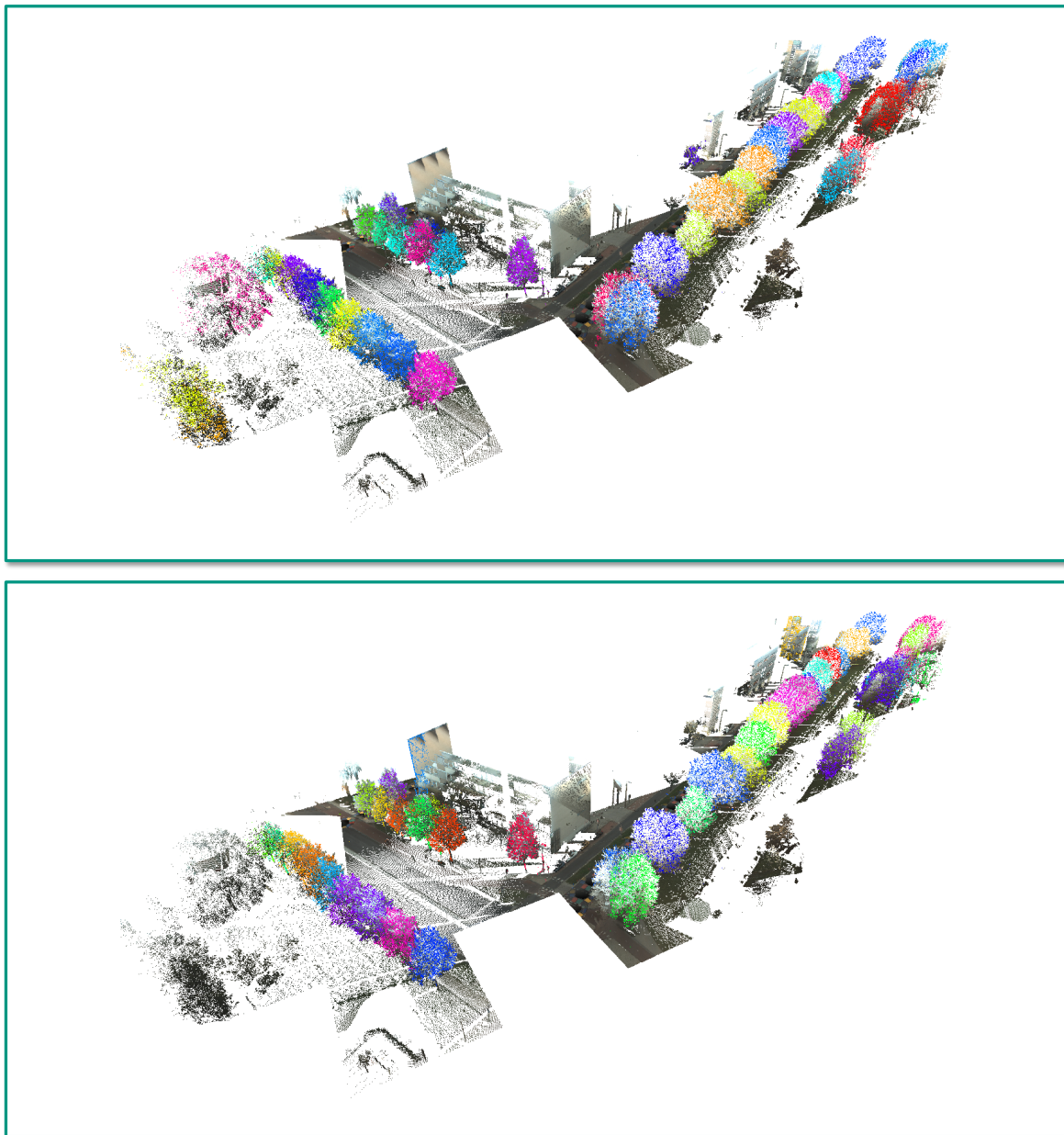


Figure 11. Visualization of the individual trees detected in the considered scene when relying on a classification based on the feature set \mathcal{S}_{3D+2D} (**top**) and when relying on a classification based on the feature set \mathcal{S}_{FCBF} (**bottom**).

5. Discussion

The derived results provide important insights with respect to semantic classification (Section 5.1), semantic segmentation (Section 5.2) and the computational effort corresponding to both tasks (Section 5.3).

5.1. Task 1: Semantic Classification

For the first step of our framework, we can conclude that it is relatively simple and easy-to-use for non-expert end-users. Furthermore, the consideration of different sets of geometric features as input for semantic classification allows one to reason about their suitability with respect to the classification task. In this regard, the derived results clearly reveal that using the feature set \mathcal{S}_{dim} comprising only

the three dimensionality features of linearity L_λ , planarity P_λ and sphericity S_λ for the classification task does not lead to accurate results (Table 2). The respective mean values of $OA = 74.33\%$ and $\kappa = 35.20\%$ across all 20 runs are relatively low, whereas the standard deviation $\sigma_{OA} = 2.51\%$ is relatively high. The consideration of the feature set S_λ comprising the three dimensionality features as well as five further local 3D shape features as input for classification yields a significant improvement of the derived classification results and thereby decreases the standard deviation σ_{OA} by about 1%, but the mean values of $OA = 84.04\%$ and $\kappa = 56.53\%$ across all 20 runs are still not sufficient to claim appropriate classification results. When further extending the set of considered features to the set S_{3D} comprising all defined 3D features, mean values of $OA = 90.08\%$ and $\kappa = 71.36\%$ are achieved, and the standard deviation σ_{OA} is further reduced to $\sigma_{OA} = 1.16\%$. The additional use of 2D features, i.e., the consideration of the feature set S_{3D+2D} comprising 18 low-level geometric 3D and 2D features [6,22], leads to classification results of approximately the same quality ($OA = 90.02\%$, $\kappa = 71.39\%$, $\sigma_{OA} = 1.22\%$). In comparison to the feature set S_{dim} , the feature set S_{3D+2D} delivers a significant gain of about $\Delta_{OA} = 15.69\%$ in overall accuracy and $\Delta_\kappa = 36.19\%$ in kappa, while the standard deviation σ_{OA} is reduced by about $\Delta_{\sigma_{OA}} = 1.29\%$. As pointed out in earlier investigations [5], the computational effort for calculating the low-level geometric 3D and 2D features in the feature set S_{3D+2D} reveals a linear behavior, i.e., a linear increase of the processing time can be observed for an increasing number of points in the considered 3D point cloud.

Interestingly, the additional use of reflectance information in the feature set $S_{3D+2D+I}$ decreases the quality of the derived classification results to values of $OA = 88.03\%$ and $\kappa = 67.20\%$, i.e., by a difference of $\Delta_{OA} = 1.99\%$ and $\Delta_\kappa = 4.19\%$. The further consideration of color information in the feature set $S_{3D+2D+I+RGB}$ also reveals a negative effect on the derived classification results, which is even stronger. The respective values of $OA = 84.74\%$ and $\kappa = 60.73\%$ are almost at the level corresponding to the classification results derived for the feature set S_λ and do not indicate a reliable classification. The low relevance of the given reflectance information and *RGB* color information for the classification task can be motivated by the fact that this kind of information is not really meaningful to separate “tree points” and “other points”. Considering the colored 3D point clouds in Figure 1, even for us humans, it remains impossible to separate both classes by only focusing on reflectance information and color information.

In contrast to manually selecting feature sets based on the respective feature design, the involved feature selection methods represented by Correlation-based Feature Selection (CFS) and the Fast Correlation-Based Filter (FCBF) focus on automatically selecting suitable features based on the given training data. As can be seen in the derived classification results (Table 2), both feature selection methods perform comparably well. They even improve the predictive accuracy of the involved RF classifier, while the classification itself is only based on feature sets comprising between two and four out of the 22 defined features. Since the features are selected based on the training data, there is no need to calculate and store the less suitable features for the test data. Hence, regarding the test data, the memory consumption for data storage can be reduced significantly to only about 9.09% and 18.18%, respectively. Consequently, the main motivation of applying feature selection methods, which is given by improving predictive accuracy, while simultaneously reducing the computational burden with respect to processing time and memory consumption [83], also holds for our experiments. A closer look at the corresponding feature sets S_{CFS} and S_{FCBF} reveals that, across the 20 runs (whereby the training data are randomly selected and therefore different for each run), the local surface variation C_λ and the height H are selected in all 20 runs, while the feature of linearity L_λ is selected in nine runs, and the remaining features seem to be less relevant according to the respective selection criteria of CFS and FCBF. Intuitively, the feature C_λ should allow distinguishing surfaces of low curvature (as, e.g., given for walls or ground) from cluttered surfaces (as, e.g., given for vegetation), whereas the height should allow distinguishing vegetation at different height levels (e.g., tree foliage and low vegetation) for the case of an almost flat environment without significant slope.

To provide a qualitative analysis of the derived classification results, we focus on a visual inspection of the classified 3D point cloud and on a detailed consideration of failure cases (Figures 12 and 13):

- **Incorrect reference labels:** A closer look at Figure 4 already reveals that some mislabeling obviously occurred during the annotation process. Some of the trees in the scene are completely labeled as “other points”, while the respective label should have been “tree points” instead. Due to the random sampling of training examples, some incorrectly labeled 3D points might have been selected for training the involved RF classifier, and hence, its generalization capability might be reduced. Furthermore, the incorrect labeling has a negative impact on the derived classification results as a significant number of correctly classified 3D points is considered as classification errors (Figures 12 and 13).
- **Registration errors:** The visualization of the classified 3D point clouds in Figures 5 and 6 as well as the visualization of failure cases in Figures 12 and 13 reveal that certain 3D points corresponding to building facades are likely to be classified as “tree points”, although they should be labeled as “other points”. Such misclassifications might be caused by the fact that the local neighborhood of respective 3D points is characterized by a volumetric behavior instead of a planar behavior. The volumetric behavior in turn might result from a slight misalignment of different MMS point clouds or from a degraded positioning accuracy of the involved MMS system due to GNSS multipath errors, which are more significant in urban canyons. Furthermore, the volumetric behavior could result from noise effects resulting from limitations of the used sensor in terms of beam divergence or measurement accuracy, but also from specific characteristics of the observed scene in terms of object materials, surface reflectivity and surface roughness [22]. Besides these influencing factors, the scanning geometry in terms of the distance and orientation of object surfaces with respect to the used sensor might have to be considered as well [84,85].
- **Edge effects:** For some feature sets, misclassifications might occur at the boundaries of single tiles, which is due to the separate processing of different tiles [21]. This can easily be solved by considering a small padding region around each tile so that those 3D points within the padding region are also used if they are within the local neighborhood of a 3D point within the considered tile [5].

Due to these issues, most of the misclassifications might mainly be related to the characteristics of the considered dataset and less to the proposed methodology for a point-wise, binary semantic labeling of a given 3D point cloud with respect to “tree points” and “other points”.

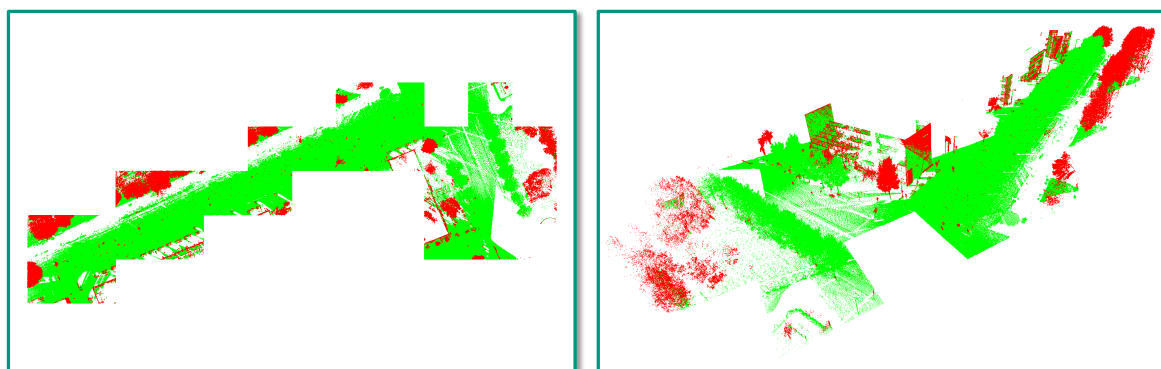


Figure 12. Visualization of the failure cases in the form of misclassifications that particularly occur for trees and building facades (**left:** nadir view; **right:** side view): those 3D points that are correctly classified with respect to the provided labeling are colored in green, whereas those 3D points that are not correctly classified with respect to the provided labeling are colored in red (the classification result corresponds to an overall accuracy of 90.07% and a kappa value of 71.54%; cf. Figure 5).

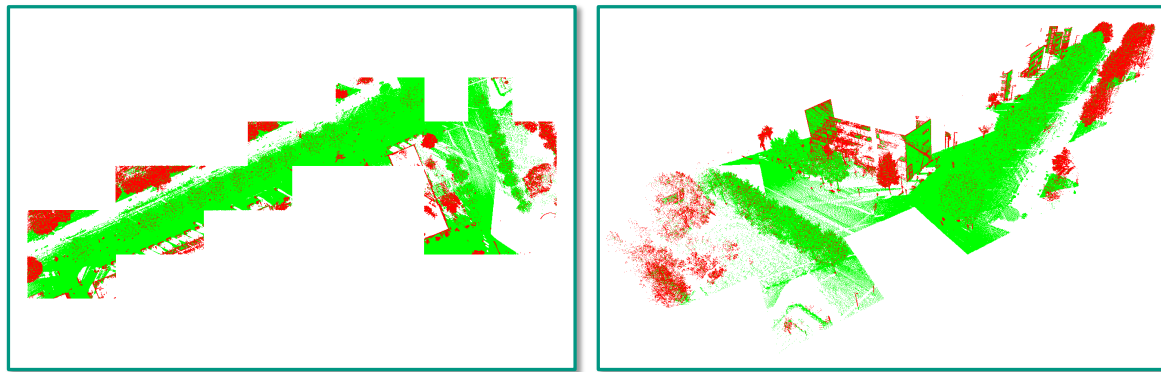


Figure 13. Visualization of the failure cases in the form of misclassifications that particularly occur for trees and building facades (**left:** nadir view; **right:** side view): those 3D points that are correctly classified with respect to the provided labeling are colored in green, whereas those 3D points that are not correctly classified with respect to the provided labeling are colored in red (the classification result corresponds to an overall accuracy of 90.71% and a kappa value of 72.14%; cf. Figure 6).

5.2. Task 2: Semantic Segmentation

For the second step of our framework, we can also conclude that it is relatively simple and easy-to-use for non-expert end-users. It directly operates at point-level and avoids using a voxelization, as for instance proposed in [13,15], where the respective segmentation results might be strongly influenced by the voxel size and the voxel orientation. The consideration at point-level remains efficient, since time-consuming tasks such as a mean shift segmentation are applied on only a small subspace of the given data and the respective results are subsequently upsampled. Thereby, the subspace is defined via a downsampling followed by a 2D projection, where the downsampling by a factor of 10 still allows detecting individual trees in the given 3D point cloud data [78] and the 2D projection improves the computational efficiency of a mean shift segmentation, which, in turn, can be performed much faster in 2D space than in 3D space [67,68].

Furthermore, we may conclude that the derived classification results as input are appropriate to achieve a meaningful segmentation with respect to individual trees (Figures 11 and 14), even if efficiency is significantly improved by applying crucial tasks only on a small subspace of the given data. A closer look at the derived segmentation results reveals that these are sufficiently accurate for the considered benchmark dataset, i.e., almost all derived segments correspond to an individual tree in the scene when relying on a classification based on the feature set \mathcal{S}_{3D+2D} , and there is only one of the detected segments that probably corresponds to a different object (Figure 9). When relying on a classification based on the feature set \mathcal{S}_{FCBF} , the derived segmentation results are still appropriate (Figure 10). However, for that case, there are a few trees in the scene that are not detected (particularly those trees that are at the boundaries of the considered scene and hence only partially acquired), and there are also some derived segments that correspond to a building edge. The segmentation results derived for both feature sets also reveal minor segmentation errors that occur at segment borders if adjacent trees are relatively close to each other and partially overlapping (see, e.g., Figure 14). Such minor segmentation errors however also become visible in the results presented in [13]. Furthermore, in the bottom left part of Figure 14, it can be observed that one tree is segmented into two parts indicated in blue and red. The blue segment contains foliage and the tree trunk, while the red segment only contains foliage. The latter segment however reveals a significant change in point density, and here, the mean shift segmentation favors the high density region, although it does not correspond to a tree trunk.

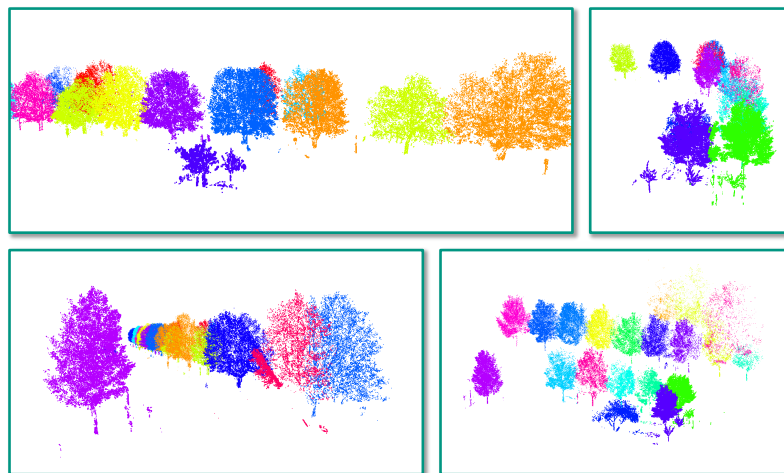


Figure 14. More detailed visualization of derived segments (the classification result corresponds to an underlying overall accuracy of 90.07% and a kappa value of 71.54%; cf. Figures 5 and 9).

5.3. Computational Effort

In contrast to our prototype implementation [21] that was designed to work on a high-performance computer with a lot of memory, we focus on improved efficiency in the scope of this paper so that the resulting implementation can also be run on a standard laptop computer (Intel Core i7-6820HK, 2.7 GHz, four cores, 16 GB RAM) with reasonable processing time. For this reason, we take into account that some of the features used in [21] can only be extracted with non-linear computational effort, whereas the computational effort for extracting local 3D shape features, geometric 3D properties, local 2D shape features and geometric 2D properties reveals a linear behavior [5]. The significant savings with respect to processing time and memory, in turn, allow for a parallel processing of crucial steps on multiple cores, which further reduces the processing time. As a consequence, we expect a significantly faster processing for the crucial steps of neighborhood selection (t_N) and feature extraction (t_{FEX}), which indeed can be verified in Table 3. Note that, for neighborhood selection, which exhibits a linear complexity, the speed-up roughly corresponds to the distributed processing on four cores. In contrast, the processing time t_{FEX} required for feature extraction is significantly less than about a fourth of the processing time reported in [21], since only those features are considered that can be extracted with linear complexity. Here, the speed-up corresponds to a factor of about 54.

Table 3. Specifications for our prototype framework presented in [21] and for our proposed framework.

Specifications	Prototype [21]	Proposed Framework
System	Intel Core i7-3820, 3.6 GHz, 64 GB RAM	Intel Core i7-6820HK, 2.7 GHz, 16 GB RAM
Implementation	MATLAB	MATLAB
Parallelization	—	4 cores
# Geometric Features	21	18
t_N	8.34 h	2.00 h
t_{FEX}	10.84 h	0.20 h
t_{train}	0.34 s	0.05 s
t_{test}	23.81 s	18.10 s

For the second step of our framework, the required processing time is significantly less than 1 min in total. Thereby, the mean shift segmentation is performed in 4.09 s and the segment-wise feature extraction in 0.41 s. Furthermore, during the segment-based shape analysis, each of the applied semantic rules takes between 0.29 s and 0.39 s. Consequently, the computational effort required for the second step of our framework is not significant in comparison to the computational effort required for the first step of our framework (Table 3).

6. Conclusions

In this paper, we have focused on an instance-level segmentation in the form of a detection of individual trees in dense 3D point cloud data acquired in urban areas. To solve this task, we have presented a novel two-step classification-segmentation framework operating at point-level and representing an end-to-end processing workflow from raw data to individual trees in the scene. The first step of our framework is given by a semantic classification in terms of assigning semantic class labels to the irregularly distributed 3D points, whereas the second step is given by a semantic segmentation in terms of separating individual objects within the labeled 3D points. For both steps, we have focused on the simplicity and applicability of the involved methods so that non-expert end-users can easily apply the complete framework on a standard laptop computer without requiring expert knowledge about single methods. The results derived for a benchmark dataset clearly indicate that a point-wise semantic classification relying on geometric features is able to deliver an appropriate semantic labeling with respect to “tree points” and “other points”, even for very few involved geometric features. Furthermore, the derived results indicate a reliable segmentation of individual trees from those 3D points that are classified as “tree points”. Thereby, the high quality of the segmentation results mainly depends on the use of semantic rules involving segment-based features, since these rules allow discarding implausible segments and thus retaining only those segments that are likely to correspond to an individual tree.

In future work, we plan to integrate parts of the proposed framework into the IQmulus platform [86] focusing on the processing of large geospatial data, where one of the defined showcases addresses large-scale scene analysis in the form of extracting individual trees from densely sampled MMS point cloud data. Furthermore, we intend to test our framework for the extension of the IQPC’15 benchmark dataset by 483 additional tiles that are unlabeled [13] as well as for a dataset corresponding to about 10 km of streets acquired in the city of Toulouse, France. Finally, it might be worth addressing planned benchmarks on tree detection and tree species classification relying on the TorontoCity dataset [87], for which airborne and ground-based acquisition systems have been used to collect different types of data for an area of about 712 km².

Acknowledgments: This work was partially supported by the European Commission’s Seventh Framework Programme under the grant agreement FP7-ICT-2011-318787 (IQmulus: A High-Volume Fusion and Analysis Platform for Geospatial Point Clouds, Coverages and Volumetric Data Sets).

Author Contributions: The authors jointly contributed to the concept of this paper, the implementation of the framework, the evaluation of the framework on a benchmark dataset, the discussion of derived results and the writing of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Munoz, D.; Bagnell, J.A.; Vandapel, N.; Hebert, M. Contextual classification with functional max-margin Markov networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 975–982.
2. Xiong, X.; Munoz, D.; Bagnell, J.A.; Hebert, M. 3-D scene analysis via sequenced predictions over points and regions. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2609–2616.
3. Hu, H.; Munoz, D.; Bagnell, J.A.; Hebert, M. Efficient 3-D scene analysis from streaming data. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 2297–2304.
4. Brédif, M.; Vallet, B.; Serna, A.; Marcotegui, B.; Paparoditis, N. TerraMobilita/IQmulus urban point cloud classification benchmark. In Proceedings of the IQmulus Workshop on Processing Large Geospatial Data, Cardiff, UK, 8 July 2014; pp. 1–6.
5. Weinmann, M.; Urban, S.; Hinz, S.; Jutzi, B.; Mallet, C. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Comput. Graph.* **2015**, *49*, 47–57.

6. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304.
7. Hackel, T.; Wegner, J.D.; Schindler, K. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *III-3*, 177–184.
8. Vanegas, C.A.; Aliaga, D.G.; Benes, B. Automatic extraction of Manhattan-world building masses from 3D laser range scans. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1627–1637.
9. Boyko, A.; Funkhouser, T. Extracting roads from dense point clouds in large scale urban environment. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, S02–S12.
10. Zhou, L.; Vosselman, G. Mapping curbstones in airborne and mobile laser scanning data. *Int. J. Appl. Earth Observ. Geoinf.* **2012**, *18*, 293–304.
11. Guan, H.; Li, J.; Yu, Y.; Wang, C.; Chapman, M.; Yang, B. Using mobile laser scanning data for automated extraction of road markings. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 93–107.
12. Pu, S.; Rutzinger, M.; Vosselman, G.; Oude Elberink, S. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, S28–S39.
13. Gorte, B.; Oude Elberink, S.; Sirmacek, B.; Wang, J. IQPC 2015 Track: Tree separation and classification in mobile mapping lidar data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-3/W3*, 607–612.
14. Sirmacek, B.; Lindenbergh, R. Automatic classification of trees from laser scanning point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *II-3/W5*, 137–144.
15. Lindenbergh, R.C.; Berthold, D.; Sirmacek, B.; Herrero-Huerta, M.; Wang, J.; Ebersbach, D. Automated large scale parameter extraction of road-side trees sampled by a laser mobile mapping system. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-3/W3*, 589–594.
16. Kelly, M. Urban trees and the green infrastructure agenda. In Proceedings of the Urban Trees Research Conference, Birmingham, UK, 13–14 April 2011; pp. 166–180.
17. Edmondson, J.L.; Stott, I.; Davies, Z.G.; Gaston, K.J.; Leake, J.R. Soil surface temperatures reveal moderation of the urban heat island effect by trees and shrubs. *Sci. Rep.* **2016**, *6*, 1–8.
18. Wegner, J.D.; Branson, S.; Hall, D.; Schindler, K.; Perona, P. Cataloging public objects using aerial and street-level images – Urban trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 6014–6023.
19. Zhang, Z.; Fidler, S.; Urtasun, R. Instance-level segmentation for autonomous driving with deep densely connected MRFs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 669–677.
20. Silberman, N.; Sontag, D.; Fergus, R. Instance segmentation of indoor scenes using a coverage loss. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 616–631.
21. Weinmann, M.; Mallet, C.; Brédif, M. Detection, segmentation and localization of individual trees from MMS point cloud data. In Proceedings of the International Conference on Geographic Object-Based Image Analysis, Enschede, The Netherlands, 14–16 September 2016; pp. 1–8.
22. Weinmann, M. *Reconstruction and Analysis of 3D Scenes—From Irregularly Distributed 3D Points to Object Classes*; Springer: Cham, Switzerland, 2016.
23. Melzer, T. Non-parametric segmentation of ALS point clouds using mean shift. *J. Appl. Geod.* **2007**, *1*, 159–170.
24. Vosselman, G. Point cloud segmentation for urban scene classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *XL-7/W2*, 257–262.
25. Lee, I.; Schenk, T. Perceptual organization of 3D surface points. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *XXXIV-3A*, 193–198.
26. Linsen, L.; Prautzsch, H. Local versus global triangulations. In Proceedings of Eurographics, Manchester, UK, 5–7 September 2001; pp. 257–263.
27. Filin, S.; Pfeifer, N. Neighborhood systems for airborne laser data. *Photogramm. Eng. Remote Sens.* **2005**, *71*, 743–755.
28. Pauly, M.; Keiser, R.; Gross, M. Multi-scale feature extraction on point-sampled surfaces. *Comput. Graph. Forum* **2003**, *22*, 81–89.

29. Mitra, N.J.; Nguyen, A. Estimating surface normals in noisy point cloud data. In Proceedings of the Annual Symposium on Computational Geometry, San Diego, CA, USA, 8–10 June 2003; pp. 322–328.
30. Lalonde, J.F.; Unnikrishnan, R.; Vandapel, N.; Hebert, M. Scale selection for classification of point-sampled 3D surfaces. In Proceedings of the International Conference on 3-D Digital Imaging and Modeling, Ottawa, ON, Canada, 13–16 June 2005; pp. 285–292.
31. Demantké, J.; Mallet, C.; David, N.; Vallet, B. Dimensionality based scale selection in 3D lidar point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, XXXVIII-5/W12, 97–102.
32. Brodu, N.; Lague, D. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS J. Photogramm. Remote Sens.* **2012**, 68, 121–134.
33. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, 87, 152–165.
34. Blomley, R.; Jutzi, B.; Weinmann, M. Classification of airborne laser scanning data using geometric multi-scale features and different neighbourhood types. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, III-3, 169–176.
35. Gevaert, C.M.; Persello, C.; Vosselman, G. Optimizing multiple kernel learning for the classification of UAV data. *Remote Sens.* **2016**, 8, 1025.
36. West, K.F.; Webb, B.N.; Lersch, J.R.; Pothier, S.; Triscari, J.M.; Iverson, A.E. Context-driven automated target detection in 3-D data. *Proc. SPIE* **2004**, 5426, 133–143.
37. Mallet, C.; Bretar, F.; Roux, M.; Soergel, U.; Heipke, C. Relevance assessment of full-waveform lidar data for urban area classification. *ISPRS J. Photogramm. Remote Sens.* **2011**, 66, S71–S84.
38. Guo, B.; Huang, X.; Zhang, F.; Sohn, G. Classification of airborne laser scanning data using JointBoost. *ISPRS J. Photogramm. Remote Sens.* **2015**, 100, 71–83.
39. Hughes, G.F. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Inf. Theory* **1968**, 14, 55–63.
40. Weinmann, M.; Jutzi, B.; Mallet, C. Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, II-5/W2, 313–318.
41. Khoshelham, K.; Oude Elberink, S.J. Role of dimensionality reduction in segment-based classification of damaged building roofs in airborne laser scanning data. In Proceedings of the International Conference on Geographic Object Based Image Analysis, Rio de Janeiro, Brazil, 7–9 May 2012; pp. 372–377.
42. Schindler, K. An overview and comparison of smooth labeling methods for land-cover classification. *IEEE Trans. Geosci. Remote Sens.* **2012**, 50, 4534–4545.
43. Shapovalov, R.; Velizhev, A.; Barinova, O. Non-associative Markov networks for 3D point cloud classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, XXXVIII-3A, 103–108.
44. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Conditional random fields for lidar point cloud classification in complex urban areas. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, I-3, 263–268.
45. Schmidt, A.; Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of full waveform lidar data in the Wadden Sea. *IEEE Geosci. Remote Sens. Lett.* **2014**, 11, 1614–1618.
46. Weinmann, M.; Schmidt, A.; Mallet, C.; Hinz, S.; Rottensteiner, F.; Jutzi, B. Contextual classification of point cloud data by exploiting individual 3D neighborhoods. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, II-3/W4, 271–278.
47. Shapovalov, R.; Vetrov, D.; Kohli, P. Spatial inference machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2985–2992.
48. Maturana, D.; Scherer, S. VoxNet: A 3D convolutional neural network for real-time object recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
49. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
50. Riegler, G.; Ulusoy, A.O.; Geiger, A. OctNet: Learning deep 3D representations at high resolutions. *arXiv* **2017**, arXiv:1611.05009v3.
51. Savinov, N. Point Cloud Semantic Segmentation via Deep 3D Convolutional Neural Network. 2017. Available online: <https://github.com/nsavinov/semantic3dnet> (accessed on 6 March 2017).

52. Huang, J.; You, S. Point cloud labeling using 3D convolutional neural network. In Proceedings of the International Conference on Pattern Recognition, Cancun, Mexico, 4–8 December 2016; pp. 1–6.
53. Large-Scale Point Cloud Classification Benchmark. 2016. Available online: <http://www.semantic3d.net> (accessed on 6 March 2017).
54. Serna, A.; Marcotegui, B.; Goulette, F.; Deschaud, J.E. Paris-rue-Madame database: A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In Proceedings of the International Conference on Pattern Recognition Applications and Methods, Angers, France, 6–8 March 2014; pp. 819–824.
55. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32.
56. Zhang, J.; Lin, X.; Ning, X. SVM-based classification of segmented airborne lidar point clouds in urban areas. *Remote Sens.* **2013**, *5*, 3749–3775.
57. Aijazi, A.K.; Checchin, P.; Trassoudaine, L. Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. *Remote Sens.* **2013**, *5*, 1624–1650.
58. Wu, B.; Yu, B.; Yue, W.; Shu, S.; Tan, W.; Hu, C.; Huang, Y.; Wu, J.; Liu, H. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sens.* **2013**, *5*, 584–611.
59. Yao, W.; Fan, H. Automated detection of 3D individual trees along urban road corridors by mobile laser scanning systems. In Proceedings of the International Symposium on Mobile Mapping Technology, Tainan, Taiwan, 1–3 May 2013; pp. 1–6.
60. Reitberger, J.; Schnörr, C.; Krzystek, P.; Stilla, U. 3D segmentation of single trees exploiting full waveform lidar data. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 561–574.
61. Rutzinger, M.; Pratihast, A.K.; Oude Elberink, S.; Vosselman, G. Detection and modelling of 3D trees from mobile laser scanning data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, XXXVIII-5, 520–525.
62. Rutzinger, M.; Pratihast, A.K.; Oude Elberink, S.J.; Vosselman, G. Tree modelling from mobile laser scanning data-sets. *Photogramm. Rec.* **2011**, *26*, 361–372.
63. Monnier, F.; Vallet, B.; Soheilian, B. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, I-3, 245–250.
64. Oude Elberink, S.; Kemboi, B. User-assisted object detection by segment based similarity measures in mobile laser scanner data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, XL-3, 239–246.
65. Gupta, S.; Weinacker, H.; Koch, B. Comparative analysis of clustering-based approaches for 3-D single tree detection using airborne fullwave lidar data. *Remote Sens.* **2010**, *2*, 968–989.
66. Fukunaga, K.; Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **1975**, *21*, 32–40.
67. Ferraz, A.; Bretar, F.; Jacquemoud, S.; Goncalves, G.; Pereira, L.; Tomé, M.; Soares, P. 3-D mapping of a multi-layered Mediterranean forest using ALS data. *Remote Sens. Environ.* **2012**, *121*, 210–223.
68. Schmitt, M.; Brück, A.; Schönberger, J.; Stilla, U. Potential of airborne single-pass millimeterwave InSAR data for individual tree recognition. In Proceedings of the Tagungsband der Dreiländertagung der DGPF, der OVG und der SGPF, Freiburg, Germany, 27 February–1 March 2013; Volume 22, pp. 427–436.
69. Yao, W.; Krzystek, P.; Heurich, M. Enhanced detection of 3D individual trees in forested areas using airborne full-waveform lidar data by combining normalized cuts with spatial density clustering. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, II-5/W2, 349–354.
70. Shahzad, M.; Schmitt, M.; Zhu, X.X. Segmentation and crown parameter extraction of individual trees in an airborne TomoSAR point cloud. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, XL-3/W2, 205–209.
71. Schmitt, M.; Shahzad, M.; Zhu, X.X. Reconstruction of individual trees from multi-aspect TomoSAR data. *Remote Sens. Environ.* **2015**, *165*, 175–185.
72. Zhao, Z.; Morstatter, F.; Sharma, S.; Alelyani, S.; Anand, A.; Liu, H. *Advancing Feature Selection Research — ASU Feature Selection Repository*; Technical Report; School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA, 2010.
73. Hall, M.A. Correlation-based feature subset selection for machine learning. Ph.D. thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand, 1999.
74. Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; Vetterling, W.T. *Numerical recipes in C*; Cambridge University Press: Cambridge, UK, 1988.

75. Yu, L.; Liu, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In Proceedings of the International Conference on Machine Learning, Washington, DC, USA, 21–24 August 2003; pp. 856–863.
76. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140.
77. Criminisi, A.; Shotton, J. *Decision Forests for Computer Vision and Medical Image Analysis*; Springer: London, UK, 2013.
78. Weinmann, M.; Mallet, C.; Brédif, M. Segmentation and localization of individual trees from MMS point cloud data acquired in urban areas. In Proceedings of the Tagungsband der Dreiländertagung der DGPF, der OVG und der SGPF, Bern, Switzerland, 7–9 June 2016; Volume 25, pp. 351–360.
79. Caraffa, L.; Brédif, M.; Vallet, B. 3D octree based watertight mesh generation from ubiquitous data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-3/W3*, 613–617.
80. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799.
81. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619.
82. Chen, C.; Liaw, A.; Breiman, L. *Using Random Forest to Learn Imbalanced Data*; Technical Report; University of California, Berkeley, CA, USA, 2004.
83. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
84. Soudarissanane, S.; Lindenbergh, R.; Menenti, M.; Teunissen, P. Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 389–399.
85. Weinmann, M.; Jutzi, B. Geometric point quality assessment for the automated, markerless and robust registration of unordered TLS point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *II-3/W5*, 89–96.
86. Böhm, J.; Brédif, M.; Gierlinger, T.; Krämer, M.; Lindenbergh, R.; Liu, K.; Michel, F.; Sirmacek, B. The IQmulus urban showcase: Automatic tree classification and identification in huge mobile mapping point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B3*, 301–307.
87. Wang, S.; Bai, M.; Mattyus, G.; Chu, H.; Luo, W.; Yang, B.; Liang, J.; Cheverie, J.; Fidler, S.; Urtasun, R. TorontoCity: Seeing the world with a million eyes. *arXiv* **2016**, arXiv:1612.00423.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).