DETECTION, SEGMENTATION AND LOCALIZATION OF INDIVIDUAL TREES FROM MMS POINT CLOUD DATA

M. Weinmann[a,b], C. Mallet[a], M. Brédif[a]

[a] Université Paris-Est, IGN, LaSTIG, MATIS,
73 avenue de Paris, 94160 Saint-Mandé, France - (martin.weinmann, clement.mallet, mathieu.bredif)@ign.fr
[b] Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology (KIT),
Englerstr. 7, 76131 Karlsruhe, Germany - martin.weinmann@kit.edu

KEY WORDS: Mobile Mapping Systems, Point Cloud, Feature Extraction, Classification, Segmentation

ABSTRACT:

In this paper, we address the extraction of objects from 3D point clouds acquired with mobile mapping systems. More specifically, we focus on the detection of tree-like objects, a subsequent segmentation of individual trees and a localization of the respective trees. Thereby, the detection of tree-like objects is achieved via a binary point-wise classification based on geometric features, which categorizes each point of the 3D point cloud into either tree-like objects or non-tree-like objects. The subsequent segmentation and localization of individual trees is carried out by applying a 2D projection and a mean shift segmentation on a downsampled version of that part of the original 3D point cloud which represents all tree-like objects, and it also involves a segment-based shape analysis to only retain plausible tree segments. We demonstrate the performance of our framework on a benchmark dataset which contains 10.13M 3D points and has been acquired with a mobile mapping system in the city of Delft in the Netherlands.

1. INTRODUCTION

Due to the technological advancements in 3D data acquisition, there has been an increasing interest for the acquisition and automated analysis of large scenes in recent years. Particularly mobile mapping systems are meanwhile widely used since they allow to capture data in the form of 3D point clouds representing densely sampled object surfaces. The automated analysis of such 3D point cloud data has been addressed with respect to different avenues of research such as 3D point cloud classification, the segmentation of pole-like objects, the detection of roads and/or road inventory, or the identification of individual trees.

In this paper, we present a novel two-stage framework for tree extraction from 3D point clouds acquired with mobile mapping systems. The first stage of this framework focuses on the detection of tree-like objects via a binary point-wise classification based on geometric features, which categorizes each point of the 3D point cloud into either tree-like objects or non-tree-like objects. The second stage exploits the derived classification results and performs a segmentation and localization of individual trees, whereby a 2D projection and a mean shift segmentation are applied on a downsampled version of that part of the original 3D point cloud which represents all tree-like objects. To only retain plausible tree segments, this stage also involves a segment-based shape analysis.

We demonstrate the performance of our framework on a benchmark dataset acquired with a mobile mapping system in the city of Delft in the Netherlands. For our experiments, we consider the provided subset comprising 26 tiles with a total number of about 10.13M points for which reference labels with respect to a binary classification are available and for which the results of a segmentation and localization of individual trees may easily be verified by visual inspection. This evaluation strategy allows a comparison to those results achieved in the scope of the recent IQmulus Processing Contest IQPC'15, where the first step consists in a classification of 3D points corresponding to trees and other 3D points and the second step consists in segmenting the 3D points corresponding to the *tree* class into clusters referring to the respective individual trees. Based on the derived results, we discuss the strengths and limitations of both the framework and the involved methods in detail.

For our framework, appropriate classification results represent an important prerequisite for the segmentation and localization of individual trees. In the evaluation, particular attention is therefore paid to the impact of using different feature sets on the classification results. With a detailed evaluation in this regard, we draw conclusions about which features to use, about the computational efficiency of feature extraction and classification, and about how to further increase the quality of the derived results.

After briefly discussing related work in Section 2, we explain our novel framework and its components in detail in Section 3. Subsequently, we demonstrate the performance of this framework on a publicly available benchmark dataset in Section 4 and discuss the derived results. Finally, in Section 5, we provide concluding remarks and suggestions for future work.

2. RELATED WORK

To describe related work relying on the use of mobile laser scanning data, we focus on recent progress in two research directions represented by (*i*) a *semantic classification* which aims to assign a semantic class label to each 3D point (Weinmann et al., 2015a; Weinmann, 2016) and (*ii*) a *semantic segmentation* which aims to provide a meaningful partitioning of a set of 3D points into smaller, connected subsets which correspond to objects of interest or to parts of these (Melzer, 2007; Vosselman, 2013).

2.1 Point Cloud Classification

An automatic point-wise semantic labeling of 3D point clouds typically relies on the use of meaningful features. In this regard, a variety of geometric features describing the spatial arrangement of 3D points within the local neighborhood of a considered 3D point has been proposed (West et al., 2004; Munoz et al., 2008; Weinmann et al., 2013; Hackel et al., 2016). Since an appropriate neighborhood is a crucial prerequisite for the extraction of distinctive features, further investigations focus on an automatic selection of an appropriate neighborhood size for each individual 3D point (Lalonde et al., 2005; Demantké et al., 2011; Weinmann et al., 2015a; Weinmann, 2016) and on the use of multi-scale neighborhoods relying on one neighborhood type with varying scale parameter (Brodu and Lague, 2012) or on different entities such as voxels, blocks and pillars (Hu et al., 2013).

The derived features are typically concatenated to a feature vector serving as input for classification, whereby a standard classification based on the derived feature vectors may be used (Weinmann et al., 2015a; Weinmann, 2016) as well as a contextual classification also taking into account relationships among 3D points within the local neighborhood in order to assign the class label (Munoz et al., 2008; Munoz et al., 2009a; Munoz et al., 2009b; Xiong et al., 2011).

Among the classes which are considered in the task of multi-class classification based on MLS data, several benchmark datasets contain at least one class referring to vegetation (Munoz et al., 2009a; Serna et al., 2014; Brédif et al., 2014). However, a particular focus on the binary classification of MLS point clouds with respect to *tree* and *non-tree* classes has been set in recent investigations presented in (Sirmacek and Lindenbergh, 2015), where a 2D probability matrix is defined on a horizontal plane and each entry of the matrix represents a probability value calculated by checking the respective point density. Assuming that tree trunks correspond to high values in the derived 2D probability matrix, local maxima are selected as tree trunks and further points are assigned to these tree trunks if they appear in the close proximity.

2.2 Point Cloud Segmentation

There are different approaches which may be applied for 3D point cloud segmentation (Vosselman, 2013). While the segmentation may generally address a variety of objects, we focus on detecting single trees from 3D point cloud data, i.e. the derived segments should correspond to individual trees. In order to achieve a respective segmentation, many approaches rely on a voxelization of 3D space. In (Yao and Fan, 2013), an approach is presented which derives a 2D accumulation map on a horizontally oriented plane and – based on respective features – allows to separate natural objects such as trees from man-made objects. Those 3D points corresponding to natural objects are transferred to a voxel space and, subsequently, a normalized cut segmentation based on the voxel structure is carried out (Reitberger et al., 2009). A different strategy consists in performing a voxelization of 3D space, deriving connected components and separating the components further if they contain multiple clusters (Gorte et al., 2015). Alternatively, tree individualization may be achieved by a downsampling and retiling of the original 3D point cloud data via voxelization, where a subsequent 2D gridding allows to find local maxima in point density and thus potential tree locations (Lindenbergh et al., 2015). Based on these tree locations, individual trees are finally segmented via octree-based region growing and thresholding techniques.

Besides approaches involving a voxelization of 3D space, there are also approaches which perform tree individualization by considering the original data on point-level. An exemplary approach focuses 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 obtain individual tree clusters (Monnier et al., 2012). Furthermore, an approach for tree individualization on point-level has been proposed which relies on deriving connected components for those 3D points categorized into a *tree* class, and the connected components are further split via an upward and downward growing algorithm if there are multiple seeds at a height between 0.5m and 1m (Gorte et al., 2015; Oude Elberink and Kemboi, 2014). Alternatively, a direct consideration of the original data on point-level may be achieved by applying a standard clustering technique such as k-means clustering or hierarchical clustering (Gupta et al., 2010), or the mean shift algorithm presented in (Fukunaga and Hostetler, 1975). The latter has for instance been applied on 3D point cloud data in (Ferraz et al., 2012; Schmitt et al., 2013; Yao et al., 2013; Shahzad et al., 2015). However, such an approach can be computationally demanding, particularly for a large number of considered 3D points. To improve computational efficiency, it seems desirable to apply the mean shift algorithm on a 2D projection of a 3D point cloud as e.g. described in (Schmitt et al., 2015) for 3D point cloud data acquired via tomographic SAR processing. However, the point density will be significantly higher for a ground-based acquisition of 3D point cloud data as e.g. given when using mobile mapping systems.

3. METHODOLOGY

In this paper, we present a novel framework for detecting, segmenting and localizing individual trees from MMS point cloud data. Our framework first addresses the detection of tree-like structures in a considered 3D point cloud via classification (Section 3.1), which is followed by a segmentation and localization of individual trees (Section 3.2).

3.1 Detection of Tree-Like Structures via Classification

In the scope of this work, we mainly focus on the use of geometric features to obtain a point-wise semantic labeling of a considered 3D point cloud. Accordingly, a respective local neighborhood has to be recovered for each 3D point $\mathbf{X}$ in order to appropriately describe the local 3D structure at $\mathbf{X}$ (Section 3.1.1). Subsequently, a variety of geometric features may be extracted for $\mathbf{X}$ based on those 3D points within its local neighborhood (Section 3.1.2). If available, intensity and color information may also be considered to define features. Depending on the features of interest, we may define different feature sets (Section 3.1.3) which allows us to conclude about their absolute and relative performance with respect to the classification task (Section 3.1.4).

3.1.1 Neighborhood Selection: To appropriately describe the local 3D structure at a considered 3D point $\mathbf{X}$, we may generally consider different neighborhood types (Weinmann et al., 2015a). Since we focus on a processing of MMS point cloud data for which the point density is rather high, we favor the use of a spherical neighborhood. Furthermore, we intend to avoid including prior knowledge about the scene and/or the data for specifying the neighborhood. Accordingly, we involve a generic solution which automatically selects a suitable neighborhood size for each individual 3D point $\mathbf{X}$. Respective approaches typically rely on a neighborhood formed by the k nearest neighbors of $\mathbf{X}$, and we determine the optimal scale parameter k_{opt} for each individual 3D point via eigenentropy-based scale selection (Weinmann et al., 2015a; Weinmann, 2016) which has proven to be favorable in comparison to other approaches.

3.1.2 Feature Extraction: Based on the defined neighborhood, we extract a set of 14 geometric 3D features (Weinmann et al., 2013; Weinmann et al., 2015a). These features comprise eight local 3D shape features (West et al., 2004; Pauly et al., 2003):

- 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: $O_\lambda = \sqrt[3]{\lambda_1\,\lambda_2\,\lambda_3}$
- Anisotropy: $A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1}$
- Eigenentropy: $E_\lambda = -\lambda_1 \ln(\lambda_1) - \lambda_2 \ln(\lambda_2) - \lambda_3 \ln(\lambda_3)$
- Sum of eigenvalues: $\Sigma_\lambda = \lambda_1 + \lambda_2 + \lambda_3$
- Local surface variation: $C_\lambda = \frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3}$

Here, the λ_i with $i \in \{1,2,3\}$, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ and $\lambda_1+\lambda_2+\lambda_3 = 1$ represent the normalized eigenvalues of the 3D structure tensor calculated based on all 3D points within the neighborhood of $\mathbf{X}$. Furthermore, the set of geometric 3D features comprises six basic geometric 3D properties of the considered 3D point $\mathbf{X}$ and its neighborhood:

- Height $H = Z$ of the considered 3D point
- Radius $R_{k\text{-NN}}$ of the neighborhood
- Local point density: $D = \frac{\text{\# 3D points within the local neighborhood}}{\text{volume of the local neighborhood}}$
- Verticality: $V = 1 - \mathbf{n}_Z$ where $\mathbf{n}$ is the normal vector
- Maximum height difference $\Delta H_{k\text{-NN}}$ within the neighborhood
- Standard deviation of height values $\sigma_{H,k\text{-NN}}$ within the neighborhood

Since urban environments contain many man-made objects with almost perfectly vertical structures (e.g. building façades, walls, poles or traffic signs), we also consider features relying on a projection of $\mathbf{X}$ and its nearest neighbors onto a horizontally oriented plane. For the resulting 2D space, we define local 2D shape features in analogy to the 3D case:

- Sum of eigenvalues: $\Sigma_{\lambda,\text{2D}} = \lambda_{1,\text{2D}} + \lambda_{2,\text{2D}}$
- Ratio of eigenvalues: $R_{\lambda,\text{2D}} = \lambda_{2,\text{2D}} / \lambda_{1,\text{2D}}$

where $\lambda_{1,\text{2D}}$ and $\lambda_{2,\text{2D}}$ are the eigenvalues of the 2D structure tensor. Furthermore, we use two geometric 2D properties:

- Radius $R_{k\text{-NN,2D}}$
- Local point density D_{2D}

To also account for the vertical behavior of 3D scene points around $\mathbf{X}$, we introduce a further neighborhood definition in the form of a spatial binning resulting from the discretization of the horizontally oriented plane into quadratic bins with a side length of 0.25m and, for each bin, we derive features for $\mathbf{X}$ from the statistics of those 3D points assigned to the respective bin:

- Number N_{bin} of 3D points falling into the respective bin
- Maximum height difference ΔH_{bin} within the bin
- Standard deviation of height values $\sigma_{H,\text{bin}}$ within the bin

If available, reflectance and color information may be considered to define radiometric features which can be used in addition to all these geometric features.

3.1.3 Feature Selection:

For our experiments, we test different feature sets which are defined as follows:

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

$$\mathcal{S}_{\text{dim}} = \{L_\lambda, P_\lambda, S_\lambda\}$$

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

$$\mathcal{S}_{\text{EV,3D}} = \{L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda\}$$

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

$$\begin{aligned}\mathcal{S}_{\text{3D}} = \{&L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda,\\ &H, R_{k\text{-NN}}, D, V, \Delta H_{k\text{-NN}}, \sigma_{H,k\text{-NN}}\}\end{aligned}$$

- The feature set $\mathcal{S}_{\text{3D+2D*}}$ 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:

$$\begin{aligned}\mathcal{S}_{\text{3D+2D*}} = \{&L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda,\\ &H, R_{k\text{-NN}}, D, V, \Delta H_{k\text{-NN}}, \sigma_{H,k\text{-NN}},\\ &\Sigma_{\lambda,\text{2D}}, R_{\lambda,\text{2D}}, R_{k\text{-NN,2D}}, D_{\text{2D}}\}\end{aligned}$$

- The feature set $\mathcal{S}_{\text{3D+2D}}$ contains all 3D and 2D features, i.e. the local 3D shape features, the geometric 3D properties, the local 2D shape features, the geometric 2D properties and the features based on the 2D accumulation map:

$$\begin{aligned}\mathcal{S}_{\text{3D+2D}} = \{&L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda,\\ &H, R_{k\text{-NN}}, D, V, \Delta H_{k\text{-NN}}, \sigma_{H,k\text{-NN}},\\ &\Sigma_{\lambda,\text{2D}}, R_{\lambda,\text{2D}}, R_{k\text{-NN,2D}}, D_{\text{2D}},\\ &N_{\text{bin}}, \Delta H_{\text{bin}}, \sigma_{H,\text{bin}}\}\end{aligned}$$

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

$$\begin{aligned}\mathcal{S}_{\text{3D+2D+I}} = \{&L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda,\\ &H, R_{k\text{-NN}}, D, V, \Delta H_{k\text{-NN}}, \sigma_{H,k\text{-NN}},\\ &\Sigma_{\lambda,\text{2D}}, R_{\lambda,\text{2D}}, R_{k\text{-NN,2D}}, D_{\text{2D}},\\ &N_{\text{bin}}, \Delta H_{\text{bin}}, \sigma_{H,\text{bin}},\\ &I\}\end{aligned}$$

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

$$\begin{aligned}\mathcal{S}_{\text{3D+2D+I+RGB}} = \{&L_\lambda, P_\lambda, S_\lambda, O_\lambda, A_\lambda, E_\lambda, \Sigma_\lambda, C_\lambda,\\ &H, R_{k\text{-NN}}, D, V, \Delta H_{k\text{-NN}}, \sigma_{H,k\text{-NN}},\\ &\Sigma_{\lambda,\text{2D}}, R_{\lambda,\text{2D}}, R_{k\text{-NN,2D}}, D_{\text{2D}},\\ &N_{\text{bin}}, \Delta H_{\text{bin}}, \sigma_{H,\text{bin}},\\ &I, R, G, B\}\end{aligned}$$

For each feature subset, the respective features are concatenated to a feature vector and a subsequent normalization is carried out so that the values of each dimension are mapped to the interval $[0, 1]$. Thereby, the normalization is defined based on the minimum and maximum values of each dimension for the training examples. The test data is mapped accordingly and values outside of $[0, 1]$ are mapped to the closest border of the interval.

3.1.4 Classification: The normalized feature vectors serve as input for classification for which we use a Random Forest (RF) classifier (Breiman, 2001) as representative of modern discriminative methods. Generally, the RF classifier consists of an ensemble of decision trees as weak learners, where each decision tree is trained for a subset of the training data which is randomly drawn with replacement. Due to this random selection, it may be expected that the weak learners are all randomly different from each other, so that taking the majority vote across the hypotheses of all weak learners results in a generalized and robust hypothesis of a single strong learner.

3.2 Segmentation and Localization of Individual Trees

The procedure to get from 3D points categorized into the *tree* class to 3D segments corresponding to trees mainly relies on our previous work (Weinmann et al., 2016) which involves a downsampling of the original data, a projection of the downsampled data onto a horizontally oriented plane, a mean-shift-based segmentation of the projected points, a transfer of the segmentation results to the original data, a refinement of the segmentation results via segment-based shape analysis, and a localization of respective tree trunks. However, we also take into account that misclassifications might occur for 3D points corresponding to flat surfaces (Gorte et al., 2015), and we therefore introduce an initial filtering based on the feature of verticality. Normalized to the interval $[0, 1]$, this feature characterizes horizontal surfaces in case of low values (e.g. in $[0, T_1]$) and high values (e.g. in $[0, 1 - T_1]$), while a vertical structure is indicated by a value of ≈ 0.5 (e.g. in $[0.5 - T_2, 0.5 + T_2]$). Accordingly, we apply an initial thresholding whereby the thresholds are selected based on the histogram of values for the feature of verticality ($T_1 = 0.1, T_2 = 0.2$).

3.2.1 Downsampling: To improve efficiency with respect to processing time and memory consumption, we take into account that MMS point cloud data provides a dense representation of object surfaces near the acquisition system and that the point density may significantly be decreased while still being able to detect individual trees in the respective 3D point cloud data. The reduced 3D point cloud in turn might facilitate time-consuming tasks such as a generic segmentation. For this reason, we introduce a downsampling of the 3D points classified as *tree* by only keeping every k-th point, whereby we heuristically select a parameter of $k = 10$ as done in (Weinmann et al., 2016). To avoid such a manual selection, a pruning of this parameter could be conducted based on the local point density (Caraffa et al., 2015).

3.2.2 2D Projection: Since – due to human intervention in nature and due to planning processes – urban areas typically provide a larger spacing and less overlap between individual trees in comparison to forested areas, we neglect the occurrence of dominant, co-dominant or dominated trees in urban environments and assume that individual trees may still sufficiently be delineated when only considering a 2D projection of the downsampled 3D point cloud data corresponding to the *tree* class onto a horizontally oriented plane.

3.2.3 Mean Shift Segmentation: To derive a meaningful partitioning, the 2D projections of the downsampled 3D point cloud data corresponding to the *tree* class onto a horizontally oriented plane are provided as input for the mean shift algorithm (Fukunaga and Hostetler, 1975; Cheng, 1995; Comaniciu and Meer, 2002) which represents an iterative technique for locating the maxima / modes of a probability density function by only considering discrete data sampled from that probability density function. Thereby, the probability density function does not have to be determined explicitly, and there is no need to make assumptions on a specific geometric model or the number of modes. Considering the derived 2D projections as discrete data points sampled from an empirical 2D probability density function, the mean shift algorithm takes each data point and iteratively (*i*) calculates the weighted mean of data points within a window defined by a kernel $\mathcal{K}$ (typically an isotropic kernel such as a Gaussian kernel or an Epanechnikov kernel (Comaniciu and Meer, 2002)), (*ii*) defines the mean shift vector $\mathbf{m}$ as the difference between the data point and the weighted mean of data points within the considered window, and (*iii*) moves the data point along the mean shift vector. Thereby, the magnitude of the mean shift vector will be large in areas of low point density, whereas it will be low in areas of high point density. Accordingly, the mean shift algorithm iteratively performs an adaptive gradient ascent until convergence (up to numerical accuracy). The stationary points correspond to regions of high point density and represent the modes of the underlying distribution of data points. Finally, all data points leading to the same mode are considered as cluster or segment. In the scope of our work, the single clusters / segments are expected to represent the individual trees in the considered scene.

The number of detected modes depends on the specification of the involved kernel for which we select an isotropic Gaussian kernel with the same bandwidth h in all directions. Such a choice is intuitively justified since we want to detect individual trees and it may rely on prior knowledge about the shape and size of the trees in the considered scene. Accordingly, we carried out different tests and heuristically selected a value of $h = 3.8$m for our experiments (Weinmann et al., 2016).

3.2.4 Data Transfer: Since the segmentation results are derived for the downsampled 3D point cloud data corresponding to the *tree* class, a transfer of these results to the original 3D point cloud data is required. For this purpose, we focus on an intuitive, simple and straightforward approach which assigns each 3D point of the respective part of the original 3D point cloud data the segment label of the closest 3D point in the downsampled version of the original data. Thereby, we conduct a nearest neighbor search based on Euclidean distances.

3.2.5 Shape Analysis: In contrast to our previous work (Weinmann et al., 2016), we focus on a segment-based shape analysis which relies on a feature extraction on the basis of a segment as respective neigborhood, i.e. those geometric features presented in Section 3.1.2 may also be derived for each segment. Defining the derived segments as neighborhood, we first discard small segments which are not likely to correspond to the objects of interest. Accordingly, those segments comprising less than 500 points are removed since, for 3D point clouds acquired with mobile mapping systems, a much larger number of 3D points may be expected for meaningful segments corresponding to trees. For the remaining segments, we take into account that misclassifications may occur for 3D points corresponding to building façades which e.g. becomes visible in the results for one of the approaches presented in (Gorte et al., 2015). To address this issue, we take into account that the ratio $R_{\lambda,\mathrm{2D}}$ of the eigenvalues of the 2D structure tensor reveals line-like structures as e.g. given for the 2D projection of a building façade onto the horizontally oriented plane. Since such line-like structures in the 2D projection are rather elongated, we may simply discard all segments for which $R_{\lambda,\mathrm{2D}}$ is below a certain threshold t_R. Thereby, we heuristically select a value of $t_R = 0.3$ indicating that the smaller eigenvalue has to be equal to or even above a value of 30% of the larger eigenvalue for a segment corresponding to a tree.

3.2.6 Localization: For all plausible tree segments, we define their location via the respective mode determined during the mean shift segmentation based on the 2D projections of the downsampled 3D point cloud data.

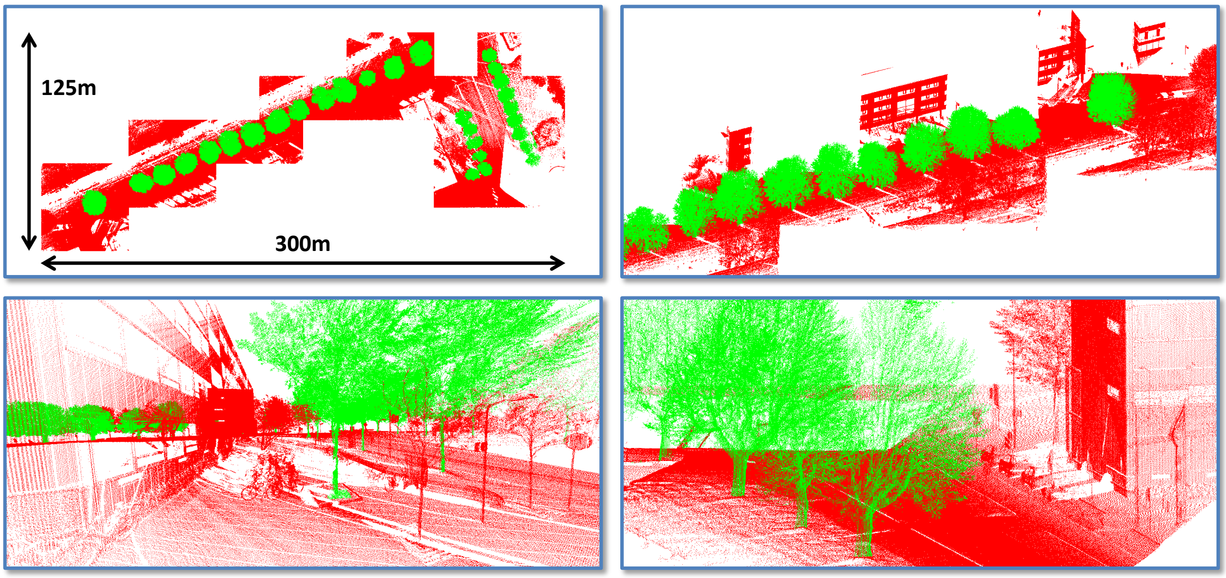

Figure 1. Visualization of the used benchmark dataset with about 10.13M labeled 3D points (top row: nadir view and side view; bottom row: more detailed views): 3D points categorized into the *tree* class are colored in green and all other 3D points are colored in red.

4. EXPERIMENTAL RESULTS

In the following, we provide details on the involved dataset (Section 4.1) and, subsequently, we present results obtained for the task of tree classification (Section 4.2) as well as results obtained for the task of tree segmentation and localization (Section 4.3).

4.1 Dataset

The considered benchmark dataset has been acquired in the vicinity of the campus of TU Delft in the Netherlands with the Fugro DRIVE-MAP system (Gorte et al., 2015). For our experiments, we use the provided subset consisting of 26 tiles with a total number of 10,126,500 labeled 3D points (see Figure 1), where the class labels refer to a binary classification to distinguish between (*i*) 3D points corresponding to trees and (*ii*) other 3D points. Thereby, the *tree* class comprises about 1.78M points (17.6%) and the *non-tree* class comprises the remaining 3D points.

4.2 Task 1: Tree Classification

To evaluate the performance of our framework with respect to tree classification, we consider the results after the binary classification categorizing 3D points with respect to the *tree* class and the *non-tree* class. Accordingly, the local neighborhood is first derived for each 3D point via eigenentropy-based scale selection. This neighborhood serves as the basis for deriving point-wise feature vectors which serve as input for the involved RF classifier. For training this classifier, we take into account that an unbalanced distribution of training examples per class might have a detrimental effect on the training process (Chen et al., 2004; Criminisi and Shotton, 2013). For this reason, we randomly select 1,000 training examples per class for training the classifier, i.e. we use 2,000 points as training set and the remaining 10,124,500 points as test set. The number N_T of decision trees used for the RF classifier has been selected heuristically via grid search and is given with $N_T = 100$ for all the feature sets introduced in Section 3.1.3. The respectively derived RF-based classification results (averaged across 10 runs) are provided in Table 1, where the following evaluation metrics are provided: overall accuracy (OA), Cohen's kappa coefficient (κ), precision for the *tree* class ($P(\textit{tree})$) and for the *non-tree* class ($P(\textit{non-tree})$), and recall for the *tree* class ($R(\textit{tree})$) and for the *non-tree* class ($R(\textit{non-tree})$). A visualization of the classification results derived for the feature set $\mathcal{S}_{\text{3D+2D*}}$ is provided in Figure 2.

These results clearly indicate that only using the three dimensionality features for classification does not lead to accurate results. The respective values for OA and κ are relatively low, while the respective standard deviation across the 10 runs is relatively high. By adding more features, the results are significantly improved and the standard deviation is reduced in most of the cases. When for instance extending the feature set $\mathcal{S}_{\text{dim}}$ comprising the three dimensionality features to the feature set $\mathcal{S}_{\text{3D+2D}}$ comprising 21 low-level geometric 3D and 2D features (Weinmann et al., 2013; Weinmann et al., 2015a), a gain of about 17.43% in OA and about 39.99% in κ may be observed, while the standard deviation σ_{OA} of the overall accuracy OA is reduced by 0.99%. When only using the feature set $\mathcal{S}_{\text{3D+2D*}}$ comprising all 3D and 2D features relying on the k-NN neighborhood, the gain is still about 15.69% in OA and about 35.83% in κ, while σ_{OA} is reduced by 0.26%. The use of $\mathcal{S}_{\text{3D+2D*}}$ can be motivated by the fact that the calculation of features relying on a 2D accumulation map is not required. This is meaningful since the computational effort for calculating respective features reveals a non-linear behavior, whereas the computational effort for calculating the remaining 18 geometric features – all depending on characteristics of the same 3D points within the local neighborhood determined via eigenentropy-based scale selection – reveals a linear behavior (Weinmann et al., 2015b). For both $\mathcal{S}_{\text{3D+2D}}$ and $\mathcal{S}_{\text{3D+2D*}}$, however, the improvement in comparison to $\mathcal{S}_{\text{dim}}$ can be considered as significant. Yet, additionally considering radiometric information (i.e. reflectance or color information) does not seem to lead to an improvement of the classification results for our application.

A more detailed consideration of failure cases (see Figure 3) reveals the following insights:

- *Incorrect labeling*: As shown in Figure 3, some trees are completely labeled as non-tree-like objects. If training examples corresponding to respective 3D points are selected, the generalization capability of the classifier might be reduced. Furthermore, the incorrect labeling causes that the evaluation on the test set considers a significant number of correctly classified 3D points as classification errors.

Feature Set	# Features	OA [%]	κ [%]	$P(\textit{tree})$ [%]	$P(\textit{non-tree})$ [%]	$R(\textit{tree})$ [%]	$R(\textit{non-tree})$ [%]
$\mathcal{S}_{\text{dim}}$	3	74.34 ± 1.47	35.62 ± 1.36	38.32 ± 1.35	93.20 ± 0.33	74.59 ± 2.07	74.29 ± 2.22
$\mathcal{S}_{\text{EV,3D}}$	8	84.42 ± 0.70	57.06 ± 1.33	53.52 ± 1.32	96.95 ± 0.21	87.65 ± 0.93	83.73 ± 0.94
$\mathcal{S}_{\text{3D}}$	14	90.12 ± 1.07	71.67 ± 2.42	64.60 ± 2.64	99.40 ± 0.13	97.49 ± 0.57	88.54 ± 1.38
$\mathcal{S}_{\text{3D+2D}^*}$	18	90.03 ± 1.21	71.45 ± 2.75	64.44 ± 2.96	99.37 ± 0.12	97.36 ± 0.53	88.46 ± 1.55
$\mathcal{S}_{\text{3D+2D}}$	21	91.77 ± 0.48	75.62 ± 1.18	68.80 ± 1.36	99.40 ± 0.11	97.46 ± 0.47	90.55 ± 0.61
$\mathcal{S}_{\text{3D+2D+I}}$	22	91.74 ± 0.60	75.61 ± 1.44	68.68 ± 1.71	99.47 ± 0.10	97.73 ± 0.45	90.46 ± 0.79
$\mathcal{S}_{\text{3D+2D+I+RGB}}$	25	91.34 ± 0.50	74.53 ± 1.19	67.68 ± 1.39	99.36 ± 0.09	97.28 ± 0.41	90.07 ± 0.65

Table 1. Mean values and standard deviation for the averaged classification results across 10 runs (2D*: only those 2D features relying on the k-NN neighborhood).

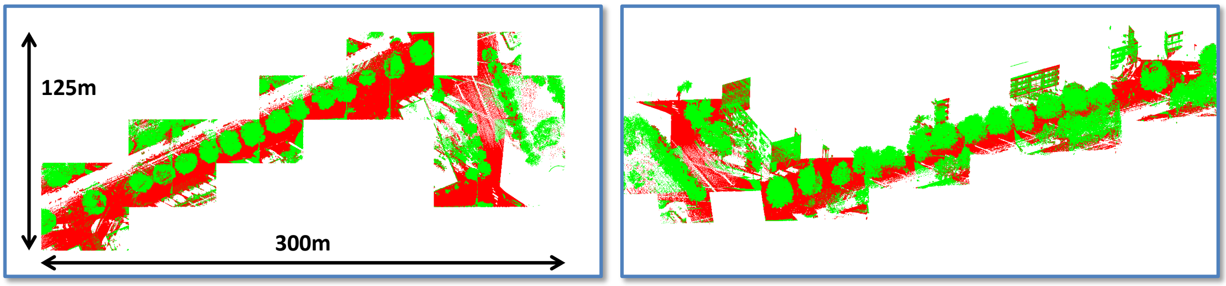

Figure 2. Visualization of exemplary classification results derived for the feature set $\mathcal{S}_{\text{3D+2D}^*}$ (left: nadir view; right: side view): 3D points classified as *tree* are colored in green and all other 3D points are colored in red.

- *Registration errors*: A closer look on the classified point cloud also reveals that there seems to be a slight misalignment of different MLS point clouds, resulting in the fact that 3D points on some building façades are characterized by a volumetric behavior when considering local neighborhoods derived via eigenentropy-based scale selection.

- *Significant variations in point density*: Some regions are not appropriately classified since the point density is extremely high (which results in an extremely small neighborhood tending to be rather meaningless) or extremely low (which results in an extremely large neighborhood tending to smooth details of the local 3D structure).

- *Edge effects*: There are some misclassifications which occur at the boundary of tiles. This might be solved by considering small padding regions at the borders of each tile, so that those 3D points within the small padding around each tile are also used if they are within the neighborhood of any 3D point within the considered tile (Weinmann et al., 2015b).

As a consequence, misclassifications might mainly depend on the considered dataset and less on the proposed methodology.

4.3 Task 2: Tree Segmentation and Localization

To evaluate the performance of our framework with respect to tree segmentation and localization, we use the results of Task 1 focusing on tree classification and delivering a classified 3D point cloud, where the single 3D points are categorized either into the *tree* class or into the *non-tree* class. All those 3D points belonging to the *non-tree* class are removed, and the remaining 3D points serve as input for the segmentation pipeline.

In the following, we consider the classification results derived when considering the feature set $\mathcal{S}_{\text{3D+2D}^*}$ containing all 3D and 2D features relying on the k-NN neighborhood. After a filtering based on the feature of verticality, the segmentation pipeline involves a mean shift segmentation on a suitable subspace of the data for reasons of efficiency. A visualization of the segmentation results derived from the classification results depicted in Figure 2 is provided in Figure 4 and also shows intermediate results after different subtasks.

A closer look on these segmentation results reveals that the derived segmentation results are sufficiently accurate for the benchmark dataset (i.e. almost all derived segments correspond to individual trees) and that only minor segmentation errors occur at segment borders if adjacent trees are relatively close to each other. However, the latter also becomes visible in the results presented in (Gorte et al., 2015). Besides, the proposed approach for individual tree segmentation and localization is rather simple and easy-to-use. It directly works on the given data without relying on a voxelization as e.g. presented in (Gorte et al., 2015; Lindenbergh et al., 2015), where the voxel size as well as the voxel orientation might strongly influence the respective segmentation results. The consideration on point-level remains efficient since time-consuming tasks such as the mean shift algorithm are applied on a subspace of the considered 3D point cloud and respective results are subsequently transferred back to the input data, whereby the subspace is defined via a downsampling and a 2D projection. The downsampling increases efficiency while still allowing to detect individual trees in the respective 3D point cloud data (Weinmann et al., 2016), and the 2D projection further reduces the computational effort since a mean shift segmentation in 2D can be conducted much faster than a mean shift segmentation in 3D (Ferraz et al., 2012; Schmitt et al., 2013).

The prototype of our framework has been implemented in Matlab and tested on a high-performance computer (Intel Core i7-3820, 3.6GHz, 64GB RAM). For the first task focusing on classification, the processing times are significant (8.34h for neighborhood selection, 10.84h for feature extraction, 0.34s for training, 23.81s for testing), while the second task focusing on a segmentation and localization of individual trees requires less than 1min in total. Yet, our implementation is not fully optimized and a significant speed-up of the first task may be achieved via parallelization.

5. CONCLUSIONS

In this paper, we have presented a framework for detecting, segmenting and localizing individual trees from MMS point cloud

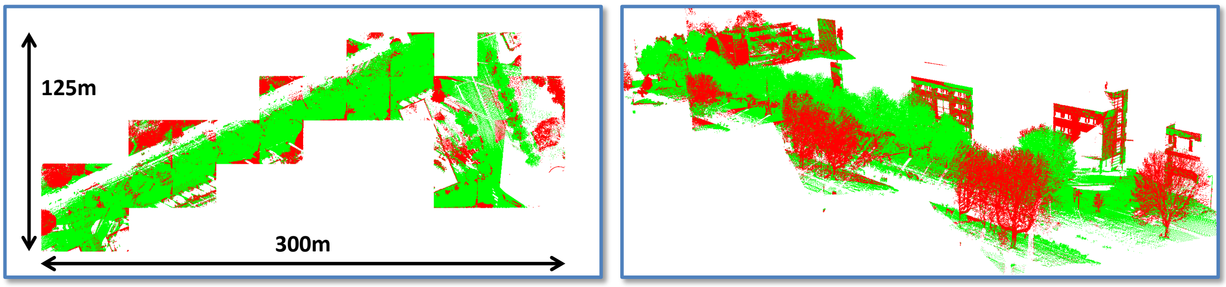

Figure 3. Visualization of the main failure cases in the form of misclassifications for trees and building façades (left: nadir view; right: side view): correctly classified 3D points are colored in green and all other 3D points are colored in red.

data. The main novelty of this framework consists in an end-to-end processing workflow from an acquired 3D point cloud to individual trees, whereby all steps are performed on point level. The derived results indicate that (*i*) classification results of high-quality may be achieved by only involving geometric features and (*ii*) appropriate segmentation results may be derived based on the classified 3D point cloud and the derived point-wise features.

For future work, we plan to integrate parts of the proposed framework into the IQmulus platform (Böhm et al., 2016) focusing on large-scale scene analysis, where one goal consists in the extraction of individual trees for a dataset which represents about 10km of streets and has been acquired in the city of Toulouse, France.

ACKNOWLEDGEMENTS

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*).

REFERENCES

Böhm, J., Brédif, M., Gierlinger, T., Krämer, M., Lindenbergh, R., Liu, K., Michel, F. and Sirmacek, B., 2016. The IQmulus urban showcase: automatic tree classification and identification in huge mobile mapping point clouds. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Prague, Czech Republic, Vol. XLI-B3, pp. 301–307.

Brédif, M., Vallet, B., Serna, A., Marcotegui, B. and Paparoditis, N., 2014. TerraMobilita/IQmulus urban point cloud classification benchmark. In: *Proceedings of the IQmulus Workshop on Processing Large Geospatial Data*, Cardiff, UK, pp. 1–6.

Breiman, L., 2001. Random forests. *Machine Learning* 45(1), pp. 5–32.

Brodu, N. and Lague, D., 2012. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing* 68, pp. 121–134.

Caraffa, L., Brédif, M. and Vallet, B., 2015. 3D octree based watertight mesh generation from ubiquitous data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, La Grande Motte, France, Vol. XL-3/W3, pp. 613–617.

Chen, C., Liaw, A. and Breiman, L., 2004. *Using random forest to learn imbalanced data*. Technical Report, University of California, Berkeley, USA.

Cheng, Y., 1995. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17(8), pp. 790–799.

Comaniciu, D. and Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), pp. 603–619.

Criminisi, A. and Shotton, J., 2013. *Decision forests for computer vision and medical image analysis*. Advances in Computer Vision and Pattern Recognition, Springer, London, UK.

Demantké, J., Mallet, C., David, N. and Vallet, B., 2011. Dimensionality based scale selection in 3D lidar point clouds. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Calgary, Canada, Vol. XXXVIII-5/W12, pp. 97–102.

Ferraz, A., Bretar, F., Jacquemoud, S., Gonçalves, G., Pereira, L., Tomé, M. and Soares, P., 2012. 3-D mapping of a multi-layered Mediterranean forest using ALS data. *Remote Sensing of Environment* 121, pp. 210–223.

Fukunaga, K. and Hostetler, L., 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory* 21(1), pp. 32–40.

Gorte, B., Oude Elberink, S., Sirmacek, B. and Wang, J., 2015. IQPC 2015 Track: Tree separation and classification in mobile mapping lidar data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, La Grande Motte, France, Vol. XL-3/W3, pp. 607–612.

Gupta, S., Weinacker, H. and Koch, B., 2010. Comparative analysis of clustering-based approaches for 3-D single tree detection using airborne fullwave lidar data. *Remote Sensing* 2(4), pp. 968–989.

Hackel, T., Wegner, J. D. and Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Prague, Czech Republic, Vol. III-3, pp. 177–184.

Hu, H., Munoz, D., Bagnell, J. A. and Hebert, M., 2013. Efficient 3-D scene analysis from streaming data. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, pp. 2297–2304.

Lalonde, J.-F., Unnikrishnan, R., Vandapel, N. and Hebert, M., 2005. Scale selection for classification of point-sampled 3D surfaces. In: *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, Ottawa, Canada, pp. 285–292.

Lindenbergh, R. C., Berthold, D., Sirmacek, B., Herrero-Huerta, M., Wang, J. and Ebersbach, D., 2015. Automated large scale parameter extraction of road-side trees sampled by a laser mobile mapping system. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, La Grande Motte, France, Vol. XL-3/W3, pp. 589–594.

Melzer, T., 2007. Non-parametric segmentation of ALS point clouds using mean shift. *Journal of Applied Geodesy* 1(3), pp. 159–170.

Monnier, F., Vallet, B. and Soheilian, B., 2012. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, Vol. I-3, pp. 245–250.

Munoz, D., Bagnell, J. A., Vandapel, N. and Hebert, M., 2009a. Contextual classification with functional max-margin Markov networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, pp. 975–982.

Munoz, D., Vandapel, N. and Hebert, M., 2008. Directional associative Markov network for 3-D point cloud classification. In: *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, Atlanta, USA, pp. 63–70.

Munoz, D., Vandapel, N. and Hebert, M., 2009b. Onboard contextual classification of 3-D point clouds with learned high-order Markov random fields. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 2009–2016.

Oude Elberink, S. and Kemboi, B., 2014. User-assisted object detection by segment based similarity measures in mobile laser scanner data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Zurich, Switzerland, Vol. XL-3, pp. 239–246.

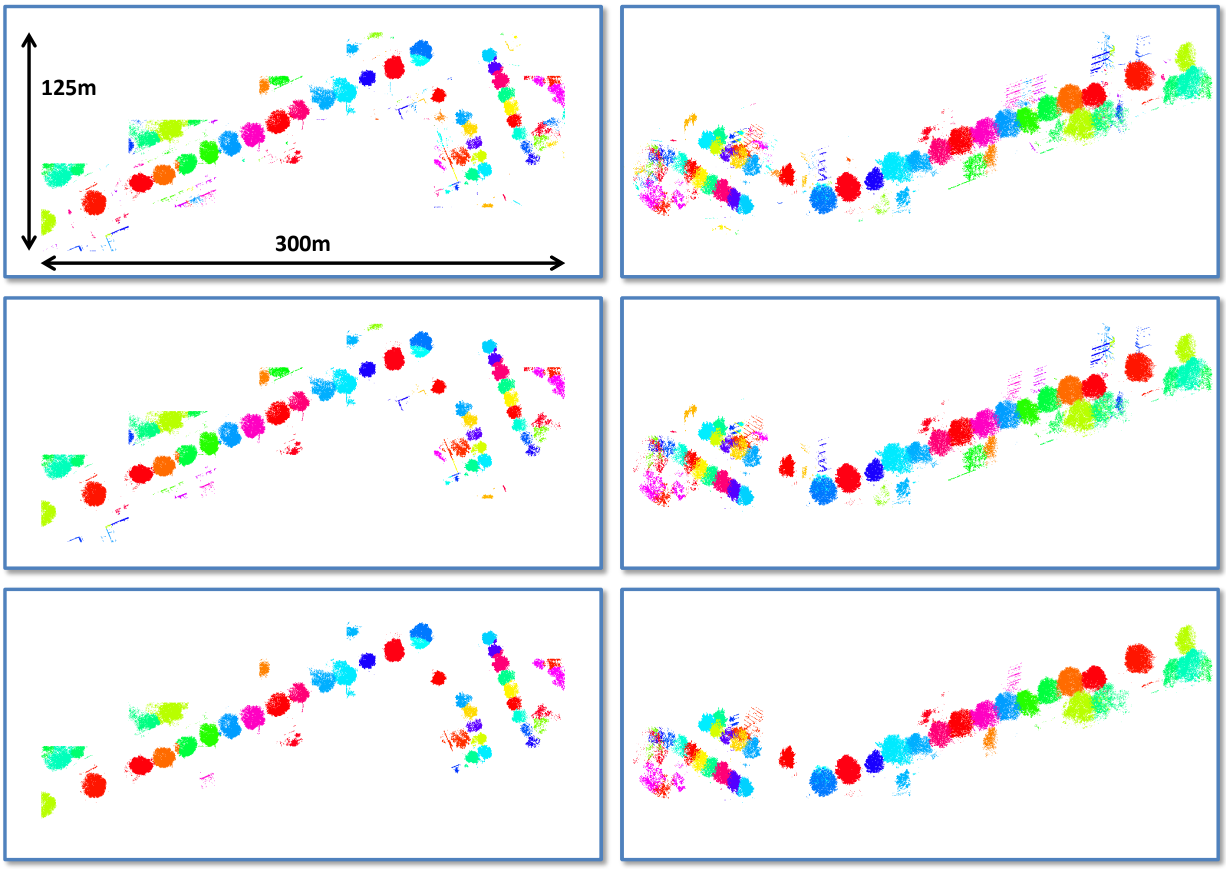

Figure 4. Visualization of the segmentation results derived from the classification results depicted in Figure 2 (left: nadir view; right: side view): single segments are indicated in different color. The figure illustrates the results after the transfer of the mean shift segmentation results to 3D space (top), after removing segments with less than 500 points (center) and after the refinement based on shape analysis (bottom).

Pauly, M., Keiser, R. and Gross, M., 2003. Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum* 22(3), pp. 81–89.

Reitberger, J., Schnörr, C., Krzystek, P. and Stilla, U., 2009. 3D segmentation of single trees exploiting full waveform lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(6), pp. 561–574.

Schmitt, M., Brück, A., Schönberger, J. and Stilla, U., 2013. Potential of airborne single-pass millimeterwave InSAR data for individual tree recognition. In: *Tagungsband der Dreiländertagung der DGPF, der OVG und der SGPF*, Freiburg, Germany, Vol. 22, pp. 427–436.

Schmitt, M., Shahzad, M. and Zhu, X. X., 2015. Reconstruction of individual trees from multi-aspect TomoSAR data. *Remote Sensing of Environment* 165, pp. 175–185.

Serna, A., Marcotegui, B., Goulette, F. and Deschaud, J.-E., 2014. 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, pp. 819–824.

Shahzad, M., Schmitt, M. and Zhu, X. X., 2015. Segmentation and crown parameter extraction of individual trees in an airborne TomoSAR point cloud. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Munich, Germany, Vol. XL-3/W2, pp. 205–209.

Sirmacek, B. and Lindenbergh, R., 2015. Automatic classification of trees from laser scanning point clouds. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, La Grande Motte, France, Vol. II-3/W5, pp. 137–144.

Vosselman, G., 2013. Point cloud segmentation for urban scene classification. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Antalya, Turkey, Vol. XL-7/W2, pp. 257–262.

Weinmann, M., 2016. *Reconstruction and analysis of 3D scenes – From irregularly distributed 3D points to object classes*. Springer, Cham, Switzerland.

Weinmann, M., Jutzi, B. and Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3D point cloud data. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Antalya, Turkey, Vol. II-5/W2, pp. 313–318.

Weinmann, M., Jutzi, B., Hinz, S. and Mallet, C., 2015a. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing* 105, pp. 286–304.

Weinmann, M., Mallet, C. and Brédif, M., 2016. Segmentation and localization of individual trees from MMS point cloud data acquired in urban areas. In: *Tagungsband der Dreiländertagung der DGPF, der OVG und der SGPF*, Bern, Switzerland, Vol. 25, pp. 351–360.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B. and Mallet, C., 2015b. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Computers & Graphics* 49, pp. 47–57.

West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M. and Iverson, A. E., 2004. Context-driven automated target detection in 3-D data. *Proceedings of SPIE* 5426, pp. 133–143.

Xiong, X., Munoz, D., Bagnell, J. A. and Hebert, M., 2011. 3-D scene analysis via sequenced predictions over points and regions. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 2609–2616.

Yao, W. and Fan, H., 2013. 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, pp. 1–6.

Yao, W., Krzystek, P. and Heurich, M., 2013. Enhanced detection of 3D individual trees in forested areas using airborne full-waveform lidar data by combining normalized cuts with spatial density clustering. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Antalya, Turkey, Vol. II-5/W2, pp. 349–354.