

Article

# Separating Tree Photosynthetic and Non-Photosynthetic Components from Point Cloud Data Using Dynamic Segment Merging

Di Wang <sup>1,\*</sup>, Jasmin Brunner <sup>1</sup>, Zhenyu Ma <sup>2</sup>, Hao Lu <sup>3</sup>, Markus Hollaus <sup>1</sup>, Yong Pang <sup>2</sup>, and Norbert Pfeifer <sup>1</sup>

- <sup>1</sup> Department of Geodesy and Geoinformation, TU Wien, 1040 Vienna, Austria; brunner.jasmin@a1.net (J.B.); markus.hollaus@geo.tuwien.ac.at (M.H.); norbert.pfeifer@geo.tuwien.ac.at (N.P.)
- <sup>2</sup> Institute of Forest Resource Information Techniques, Chinese Academy of Forestry, 100091 Beijing, China; mazhenyu22@163.com (Z.M.); pangy@ifrit.ac.cn (Y.P.)
- <sup>3</sup> School of Information Science and Technology, Beijing Forestry University, 100083 Beijing, China; luhao@bjfu.edu.cn
- \* Correspondence: di.wang@geo.tuwien.ac.at; Tel.: +43-1-58801-12280

Received: 28 February 2018; Accepted: 2 May 2018; Published: 5 May 2018



**Abstract:** Many biophysical forest properties such as wood volume and leaf area index (LAI) require prior knowledge on either photosynthetic or non-photosynthetic components. Laser scanning appears to be a helpful technique in nondestructively quantifying forest structures, as it can acquire an accurate three-dimensional point cloud of objects. In this study, we propose an unsupervised geometry-based method named Dynamic Segment Merging (DSM) to identify non-photosynthetic components of trees by semantically segmenting tree point clouds, and examining the linear shape prior of each resulting segment. We tested our method using one single tree dataset and four plot-level datasets, and compared our results to a supervised machine learning method. We further demonstrated that by using an optimal neighborhood selection method that involves multi-scale analysis, the results were improved. Our results showed that the overall accuracy ranged from 81.8% to 92.0% with an average value of 87.7%. The supervised machine learning method had an average overall accuracy of 86.4% for all datasets, on account of a collection of manually delineated representative training data. Our study indicates that separating tree photosynthetic and non-photosynthetic components from laser scanning data can be achieved in a fully unsupervised manner without the need of training data and user intervention.

**Keywords:** Laser scanning; dynamic segmentation; point classification; pattern recognition; wood-leaf classification

# 1. Introduction

Forest foliage profile affects the photosynthesis and evapotranspiration processes [1], species competition, wood production, and ecosystem and agro-ecosystem dynamics [2]. Many biophysical forest properties such as wood volume [3] and leaf area index (LAI) [4] require prior knowledge on either photosynthetic (e.g., leaf, grass, and flower) or non-photosynthetic (e.g., stem and branch wood) components. For example, understanding the spatial distribution of wood volume contributes to the monitoring of carbon stocks in forested ecosystems [3]. Above ground biomass estimation typically requires wood-only parts [5]. On the other hand, LAI ( $m^2/m^2$ ), which is defined as one-half of the total green leaf area per unit ground surface area [6], is a key descriptor of vegetation condition in numerous physiological and biogeochemical studies [7]. For a better estimation of leaf area density (LAD) ( $m^2/m^3$ ), which is defined as the total one-sided leaf area per unit volume, the wood and leaf



parts of each tree should be separated [8]. Otherwise, the LAI/LAD will be overestimated if woody components were not eliminated. Therefore, it is vital to quantitatively describe forest structures.

Laser scanning, also known as light detection and ranging (lidar), emerges as an innovative technique for nondestructive quantification of forest structures [9–12]. From single tree level to plot level forest studies, terrestrial laser scanning (TLS) is often utilized, while airborne laser scanning (ALS) is usually applicable for region scale studies. TLS is a type of ground-based scanning strategy. It acquires three-dimensional (3D) coordinates in combination with radiometric information of objects. The acquired high-density point clouds enable detailed tree quantification in a nondestructive way [9]. TLS has widespread applications in estimating tree attributes, such as stem location [13], diameter at breast height (DBH) [14], basal area and volume [15], above ground biomass [16] and LAD [8]. Some efforts have been made to separate photosynthetic and non-photosynthetic components [1,2,8,17–19]. However, separating these components in TLS data is still challenging [20].

Overall, existing methods for separating photosynthetic and non-photosynthetic components from point cloud data can be categorized into two groups: intensity based and geometry based. A different third approach uses the combination of radiometric and geometric features [21]. Intensity-based methods [2,19] use radiometric information of objects captured by a laser scanner. For forest analysis, the assumption is that wood and leaf components have different optical properties at the operating wavelength of the laser scanner [17]. However, these optical properties are influenced by the distance, partial hit, and laser incident angle [22]. Therefore, a key challenge of using intensity values is that they have to go through instrument-specific radiometrical calibration before being included in further processing [23–25]. Calders et al. [23] have shown that the radiometrical calibration carried out for a specific scanner cannot be transferred to another scanner, which greatly limits the applicability of using intensity information for downstream processing. Various authors have made attempts to construct and use multi-wavelength laser scanning to exploit different material reflectance at different wavelengths [26–28].

On the other hand, geometry-based methods only use the three-dimensional coordinates of the point cloud captured by a laser scanner, thus giving more potential and usability as point coordinates are the most fundamental information acquired by a laser scanner [17]. For separating photosynthetic and non-photosynthetic components, supervised machine learning classification has shown promise and is potentially applicable to any tree point clouds [20]. Geometrical features are extracted for each point. Machine learning classifiers such as Support Vector Machine (SVM) [18], Gaussian Mixture Model (GMM) [1,29], and Random Forest (RF) [30] have been employed to classify wood and leaf points. A major drawback of supervised machine learning classification is the requirement of training data. First of all, manual delineation of various components from point cloud data can be extremely tedious. Rendering and manipulating the high density TLS data is also an intensive task for hardware. Furthermore, manual selection of training points is impractical for processing large numbers of trees [20]. Second, the spatial distribution of training data greatly impacts the overall performance of machine learning methods. Points on main stems, small branches, leaves, bushes, and so on have to be carefully covered, which impedes the feasibility of manual manipulation. A group of other unsupervised geometry methods [8,17] look at specific geometric properties of certain components. For example, Li et al. [8] separated magnolia leaves by assuming that those leaves were basically flat in surfaces. Tao et al. [17] observed that tree trunk and branch boundaries appear as circles or circle-like shapes. However, these methods were only applied to single trees with multi-scan TLS data coverage. It is unclear if they can be adapted to nature forest scenes and to point cloud data acquired from strategies other than multi-scan TLS such as single-scan TLS and Simultaneous Localization and Mapping (SLAM).

The current study focuses on developing a fully unsupervised approach that is free of user intervention and manual training data, for separating tree photosynthetic and non-photosynthetic components from various point cloud data with varied acquisition sources. The core observation applied in our method is that non-photosynthetic components such as stems and branches appear to be

linear at various scales. Therefore, we firstly propose a robust and dynamic point cloud segmentation routine, namely Dynamic Segment Merging (DSM), to partition a point cloud into homogeneous parts. Then, the linear segments are identified by examining segments' feature saliency [1]. We test the effectiveness of the proposed DSM method using one single tree dataset, and four plot-level datasets. These datasets cover varied data acquisition strategies, scene complexities, and scanning instruments. One plot additionally features calibrated intensity information. We also compare our results to a supervised random forest model. Another incidental aim of our work is to efficiently process large amounts of points such as forest level data. We show examples of how point cloud structuring can accelerate the process.

In the following, our test data are described in Section 2. In Section 3, the proposed DSM method is explained in detail, together with a brief description of the compared RF method. We give visual and quantitative results in Section 4. In Section 5, we discuss the performances and potential improvements of our method. A general application extension example is also given. Finally, the major findings and conclusions of our work are summarized in Section 6.

## 2. Study Data

Five datasets consisting of one single tree set and four plot-level sets are used in this study. These datasets are intended to cover various data sources such as single-scan TLS, multi-scan TLS, and hand-held laser scanning; various tree species including both coniferous and deciduous trees; various terrain conditions such as urban roadside, flat terrains, and steep mountains. Terrain points were removed in advance using the method proposed in Zhang et al. [31], which is also freely available as a plugin in the open-source software CloudCompare (CloudCompare 2.9.1, 2018). One of the plot-level datasets was also radiometrically calibrated, so that we can derive results from calibrated intensity information as well.

## 2.1. Single Tree Data—SD-1

The first dataset represents TLS data of an evergreen sub tropical tree, *Erythrophleum fordii*, which is provided by Hackenberg et al. [32]. The data were acquired in October 2013 from eight scan positions. For our single tree analysis purpose, the acquired point cloud was further manually cleaned, as the tree crown interacts with other trees. Therefore, points from adjacent trees' foliage need to be removed. The cleaned point cloud contains  $\sim$ 3.9 million points. In this study, we cut out of a section of crown in order to analyze in depth the capability of our method for separating photosynthetic and non-photosynthetic components inside the canopy (Figure 1a).



**Figure 1.** Five datasets used in this study. (a) SD-1. (b) PD-1. (c) PD-2. (d) PD-3. (e) PD-4. The left column shows the original point clouds. Manually classified point clouds are shown in the middle column (non-photosynthetic components) and right column (photosynthetic components). These manually classified points served as validation sets in this study.

## 2.2. Plot-Level Data—PD-1

In May 2017, a plot with 50 m radius inside a forest in Großgöttfritz in the federal state of Lower Austria (Austria) was scanned with a TLS Riegl VZ-2000 scanner (RIEGL Laser Measurement Systems, Horn, Austria) (Figure 1b). The measurement was carried from two scanning positions. These two measured point clouds were co-registered later using Riegl's RiSCAN PRO software (RIEGL Laser Measurement Systems, Horn, Austria). The forest consists of mainly conifers (*Pinus sylvestris* L.) and a couple of silver birches (*Betula pendula*). A radiometric calibration was performed prior to the present study with a Spectralon of a known reflectivity of 99%. Twenty-five measurements were made with the lidar at distances ranging from 1 to 50 m in order to calibrate the intensity measured by the lidar [22]. Approximately 600,000 wood and leaf points were manually selected in order to examine their corresponding intensity values. Consequently, the corrected intensity information is shown in Figure 2. By setting a threshold of 0.78, we separate photosynthetic and non-photosynthetic components for this dataset, besides using the proposed geometry method.



**Figure 2.** Distribution of reflectances of photosynthetic and non-photosynthetic components after radiometrical calibration.

# 2.3. Plot-Level Data—PD-2

This plot is one of a large set of plots provided by the EuroSDR International TLS benchmark project: Benchmarking of Terrestrial Laser Scanning for Forestry Applications [33]. These plots were located in a southern Boreal forest in Finland, and were scanned by the Finnish Geodetic Institute (FGI) during the summer of 2014, with a Leica HDS6100 scanner (Leica Geosystems AG, Heerbrugg, Switzerland). Twenty-four sample plots were provided by this project. Each plot features two TLS datasets; a single-scan set from the plot center, and a multi-scan set from the center and four corners. Each plot has a size of 32 m  $\times$  32 m. In this study, we randomly select one plot from the single-scan datasets (Figure 1c).

## 2.4. Plot-Level Data—PD-3

TLS data of a plot with 15 m radius were acquired by the Chinese Academy of Forestry (CAF) in April 2017. The plot is located in the Baihuashan National Natural Reserve (Beijing, China). The tree species consists of mainly Dahurian Larch (*Larix gmelinii*) and white birch (*Betula platyphylla*). The measurement was carried out with a Trimble TX8 scanner (Trimble Inc., Sunnyvale, CA, USA). Multi-scans were performed from five locations. The resulting point clouds were further co-registered by using the Trimble RealWorks software (Trimble Inc., Sunnyvale, CA, USA) (Figure 1d).

This plot is a section of a test dataset from a roadside forest in Wolfsgraben in the federal state of Lower Austria (Austria) (Figure 1e). The main tree species in this plot is European beech (*Fagus sylvatica*) with a coniferous tree and broad-leaved weeds. The dataset was acquired by a hand-held scanner GeoSlam Zeb-1 (GeoSLAM Ltd., Nottinghamshire, UK). Zeb-1 is a lightweight and hand-held laser scanner which records more than 40,000 measurement points per second. Due to the limitation of its measurement range, the canopy level was barely covered. The purpose of acquiring this dataset was to test the lightweight scanner in forest-related studies, such as location detection, wood-leaf separation, diameter estimation, and stem curve retrieval.

#### 2.6. Validation Data

Validation data are needed to test the separation accuracy in this study. We manually classified whole point clouds (Figure 1) except the PD-3 dataset, due to its extremely complex subcanopy and understory structure. Validation data for the PD-3 set were spatially evenly distributed, and covered all class types such as leaf, small branch, main branch, and trunk (Figure 1d). Manual classification was carefully performed in the open-source software CloudCompare (CloudCompare 2.9.1, 2018). In addition, we downsampled the validation points to equal numbers for both non-photosynthetic and photosynthetic components, so that the separation accuracy will not be biased by the category that has more points than the other.

# 3. Methods

Our method builds upon the conventional region growing algorithm mentality [34]. We intend to semantically segment a point cloud into meaningful parts, so that each segment will only contain points belonging to a unique class. In this way, tree stems and branches are isolated from leaf points. Subsequently, all segments representing stem and branch can be identified by examining the linearity feature saliency of each segment. However, the challenge lies in the fact that forests have extremely complex structures. A segmentation routine has to be robust against irregular point cloud structures, varied point cloud densities, and indistinct boundaries between object classes. The conventional region growing method has difficulties on gradual change regions, which causes under-segmentation [35]. This deficiency is also evident in our objective, because tree branches and leaves often will not have distinct boundaries in point clouds. To mitigate such a disadvantage, we develop an over-segmentation-based dynamic merging strategy to segment a tree point cloud semantically and robustly.

Our proposed DSM method initially segments the point cloud into small parts, resulting in over-segmentation. Each small part is defined as a *Segment*. If these small segments are represented by a single point individually, the resulting representative is also known as a collection of *Superpoints* (e.g., [36]). The following merging process is a dynamic approach, which is similar to the region merging idea applied in the image process (e.g., [35,37]). We defined a similarity metric which was estimated for each segment and used it for merging similar neighboring segments step by step. Meanwhile, the similarities of neighboring segments are updated after each merging event occurs. Therefore, the merging procedure is dynamic, while the conventional region growing method uses static strategies to test and merge neighboring regions, thus not preserving global properties.

Our DSM algorithm was implemented in Matlab 2017a (The MathWorks, Inc., Natick, MA, USA).

## 3.1. Dynamic Segment Merging

In a point cloud region growing routine (e.g., [38]), a constraint or a combination of constraints has to be defined for deciding whether neighboring points can be merged with the current segment. Such constraints are typically on features such as normal vector variation [38], amplitude density [39], and coordinates of the origin's normal projection on the best fitted plane [40]. In this study, we use a

simple constraint that is the deviation between the z-component of normal vectors. The z-component of normal vectors represents its spatial orientation relative to the vertical direction (i.e., verticality). In other words, it stands for the angular disparity between a normal vector and the z-axis. For example,

the z-component of a normal vector will vary—in absolute value—from 0 for a point located on a vertical surface to 1 for a point on a horizontal surface. This simple criterion is useful to segment tree stem and branches, as they will grow in the similar orientation locally in space.

# 3.1.1. Normal Vector

Consistent with many other studies (e.g., [38]), our proposed DSM method heavily relies on point cloud normal vectors. In fact, estimation of surface normal vectors is one of the fundamental problems for point cloud analysis [41]. Challenges arise from outlier points, non-uniform distributions, and missing points [42]. When estimating normal vectors, a group of neighboring points has to be defined. A point cloud can be defined as  $P = \{x_i, y_i, z_i, || i = 1..n\}; P \subset \mathbb{R}, x_i, y_i, z_i$  are the coordinates of a point  $p_i$  in P. The covariance measures the variation of each dimension from the mean with respect to each other. The eigenvalues of the covariance matrix represent the variation along the direction of the eigenvectors [43]. Therefore, the normal vector NV at a point  $p_i \in P$  can be estimated as the eigenvector to the smallest eigenvalue  $EV_i$  of the covariance matrix [44] given by

$$Cov_{p_i} = \frac{\sum_{i=1}^{K} (p_i - \overline{p})(p_i - \overline{p})^T}{K},$$
(1)

where  $\overline{p}$  is the barycenter of the *K* nearest neighbors of the point  $p_i$ . The z-component of the normal vector is denoted as  $N_z$ .

The selection of *K* neighborhoods is often determined by empirical or prior knowledge on the scene, and usually with a fixed size [45]. However, many scenes contain structures with diverse sizes. For example, in forests, tree trunks and different levels of branches have varied structure scales. This indicates that multi-scale analysis or adaptive analysis can be useful in better sensing the scene (e.g., [46–48]). To examine the effects raised from *K*, we test two strategies with a fixed *K* at 10 neighborhoods and an adaptive one. To select adaptive optimal neighborhoods, we follow the approach proposed in Weinmann et al. [45]. This approach minimizes an eigenentropy given by

$$E_{EV} = -\frac{EV_1}{\sum_{n=1}^3 EV} ln(\frac{EV_1}{\sum_{n=1}^3 EV}) - \frac{EV_2}{\sum_{n=1}^3 EV} ln(\frac{EV_2}{\sum_{n=1}^3 EV}) - \frac{EV_3}{\sum_{n=1}^3 EV} ln(\frac{EV_3}{\sum_{n=1}^3 EV})$$
(2)

across different scales ranging from 9 to 99 nearest neighbors with an increment of 9 to find the optimal neighborhood size.

# 3.1.2. Initial Segmentation

We firstly partition the point cloud into small segments. Each segment should satisfy the homogeneity requirement and be geometrically simple. Several methods were deployed to overly segment a point cloud, such as spectral clustering [49] and global energy model [36]. There are several advantages of overly segmenting the point cloud into small simple and homogeneous segments. First, a segment that contains a group of points carries on more information in describing the geometric characteristics of objects than a single point [37,50]. Second, the number of initial segments (e.g., *Superpoints*) is much fewer than the number of original points in a point cloud, thus accelerating the processing speed [37]. Lastly, the initial segments help to identify isolated points. These isolated points are not grouped with other points, meaning that they are local outliers that cannot be merged with surrounding points. Therefore, they can be preliminarily removed as they will also participate in the dynamic similarity tests, which will greatly slow down the processing. However, they cannot be simply removed exhaustively, because locally isolated points may be grouped to a large structure.

In this study, we simply re-allocated each isolated point to the final results by majority voting in a sphere neighborhood.

In this study, we develop a computationally efficient strategy to overly partition the point cloud into geometrically simple segments. For a single point, a sphere is created around the point. The radius of the sphere is relevant to the study data. It should be large enough to contain meaningful geometry properties, but retain computational efficiency as a point cloud can easily contain millions of points. The range query can be exhausted with a large search radius [51]. For forest scenes, we use a radius of 25 cm for all datasets in this study. The 25 cm radius is loosely defined, and other numbers should work equally as long as they reflect local geometry and retain the computation efficiency. For example, a sensitivity test with values ranging from 15 cm to 35 cm with an increment of 5 cm showed that the standard deviation of accuracy is only 1.2%. All neighboring points within the sphere are ranked based on the distance to the center point in ascending order. The ranked points are tested orderly if the difference between its  $N_z$  and that of the center point is smaller than a threshold T (see Section 3.1.3). This procedure is terminated if the current tested point is unqualified. Consequently, all previously qualified points are merged with the center point, thus forming a small segment. All other points are returned to the point cloud. This process is repeated to a new ungrouped point, and continued until all points have a segment ID. The resulting segments are a group of cover sets (Figure 3), similar to that of Raumonen et al. [52]. We further note that the outcome of the initial segmentation only serves as the underlying architecture of the DSM algorithm, and does not significantly affect the merging result. Different degrees of over-segmentation can be carried out depending on the dataset and available computation power.



**Figure 3.** Initial segments by over-segmentation. Each segment is randomly colored. The stem surface has large size segments because it is geometrically homogeneous, while bushes have smaller segments as they are more irregular in shape.

## 3.1.3. Dynamic Merging

The segment merging processing requires the knowledge of neighboring relations [52]. In image processing, identifying neighbor pixels is more straightforward. Nevertheless, it can be cumbersome for a point cloud because a point cloud is unstructured. Usually, a search radius has to be defined to find adjacent points and segments [53,54]. In this study, we define the search radius as the 99th quantile of point spacing. Point spacing for a single point is defined as the distance to its nearest point. Consequently, the search radius is adapted to the point cloud itself, without the need of user setup.

Forests 2018, 9, 252

For each segment, its adjacent segments can be identified (Figure 4). For efficient computation, we can build an adjacency matrix for all segments. The adjacency relationship of the target segment in Figure 4 is shown in Figure 5a. Initially, the adjacency graph is unweighted. Our DSM method finds the most similar adjacent segment and dynamically updates the adjacency relation with similarity metrics of adjacent regions if the target segment is changed. For this purpose, we define a similarity measure (*SL*) as

$$SL(s,s') = \begin{cases} 1 - D_{n_z} D_q D_d & \text{if } D_{n_z} <= T \text{ and } std(N'_{z\{z=1,\dots,n'\}} \cup N_{z\{z=1,\dots,n\}}) <= 0.8 * T \\ 0 & \text{else} \end{cases}$$
(3)

where *s* and *s'* are adjacent segments, and *T* is a threshold value.  $D_{n_z}$ ,  $D_q$ , and  $D_d$  are three dissimilarity metrics. All three metrics are individually normalized to their max values, so that the similarity metric *SL* is within the range from 0 to 1.  $D_{n_z}$  stands for the  $N_z$  dissimilarity with  $D_{n_z} = |\overline{N'_z} - \overline{N_z}|$ .  $\overline{N_z}$  denotes the average value in the segment.  $D_q$  is the size dissimilarity. It is calculated as the difference in number of points between two segments.  $D_d$  represents the distance dissimilarity. To accelerate the computation, we calculate the  $D_d$  as the closest distance among distances between any points in an adjacent region *s'* to the center of the target region *s* (Equation (4)).

$$\arg\min_{i} D(i) = \{i \mid D(i) = \sqrt{\left(x_{s_{i}'} - \overline{x}_{s}\right)^{2} + \left(y_{s_{i}'} - \overline{y}_{s}\right)^{2} + \left(z_{s_{i}'} - \overline{z}_{s}\right)^{2}}\},\tag{4}$$

where  $\overline{x}$ ,  $\overline{y}$ , and  $\overline{z}$  stand for the mean values. As described in Section 3.1.2, the constraint of merging used in this study is the  $D_{n_z}$ . Therefore, we define the threshold *T* as 0.1 for dataset PD-1, PD-2, and PD-3, as TLS has higher range accuracy. For dataset SD-1 and PD-4, we set the *T* as 0.2 to tolerate the noisy data inside the canopy and data generated from the hand-held scanner.



**Figure 4.** Identifying adjacency relations. (**a**) Original point cloud. (**b**) Overly partitioned segments. The middle segment with black color is the target segment. Its adjacent segments are founded by the adaptive search radius, and are randomly colored. (**c**) *Superpoints* representation of (**b**).

We start with the initial segment with the largest size (target segment). Its adjacent segments are identified and ranked by their SL metrics in ascending order by the above-mentioned strategy. If there is at least one adjacent segment with a SL value unequal to 0, meaning that at least one segment is qualified, the adjacent segment with the largest SL is merged with the target segment. Afterwards, the adjacency relation has to be updated since the target segment is grown. The adjacent segments

of the original target segment of the merged segment are combined. The similarity metrics SL are re-calculated and ranked against the new enlarged target segment. Furthermore, we constrain that the standard deviation of  $N_z$  within a segment cannot exceed 80% of the threshold T (Equation (3)). The 80% is set empirically. In this way, we ensure the global geometrical homogeneity of each segment. If no adjacent segments can be merged, the procedure is terminated and the segment with the largest size among the rest of the segments again serves as the target segment. The process is traversed over all segments. As a consequence, our proposed DSM method tries to merge the most similar qualified adjacent segment, and adjusts the testing during the whole procedure.



**Figure 5.** Adjacency graph of segments in Figure 4. The identified graph is initially unweighted (**left**), while it is further weighted by *SL* values defined in Equation (3) (**right**). Segment 13 is unqualified as in Equation (3), thus it has 0 weight to the target segment 1. For this example, segment 10 has the largest weight, meaning that it is the most similar qualified segment. Therefore, it will merge with the target segment 1 in this step.

## 3.2. Post Processing

In this study, we use the absolute deviation of  $N_z$  as a unique constraint for merging the segments of a point cloud. In some cases, several branches can grow closely and with the same vertical orientation (Figure 6a). As a consequence, they will be grouped as a single segment. Although the segment only contains most points from stems and branches, the linearity of the segment will be impeded as multiple branches are merged as one segment. To mitigate such a situation, we further develop a top-shift algorithm to separate multiple branches into individual ones. The top-shift method works similarly to the mean-shift algorithm [55]. Instead of iteratively projecting points to their local mean, we project points to their local highest points. The search radius (bandwidth) is the same as the one used in the DSM method, which is the 99th quantile of point spacing. This simple idea will separate points from various branches, because they grow into different directions at some point following the branch structures (Figure 6b). Meanwhile, this top-shift processing also helps to identify wrongly merged points from the DSM procedure (Figure 6b), as they will converge to the local highest, while branch points converge to the top of each branch, respectively.



**Figure 6.** An example of tree branches grouped into one segment with the Dynamic Segment Merging (DSM) method (**a**) and then separated with the top-shift procedure (**b**).

## 3.3. Segment Feature

The DSM method robustly partitions the point cloud into meaningful segments. For each segment, we estimate its linearity feature saliency to examine if the segment is part of tree stems or branches. Unlike point-wise features (e.g., [30]), segment features are calculated based on all points of the segment. Linearity is defined as  $(EV_1 - EV_2)/EV_1$ .  $EV_1$ ,  $EV_2$ , and  $EV_3$  are three eigenvalues from the covariance matrix (Equation (1)), and are sorted in descending order. For segment features, the covariance matrix is formed by all points in a segment. In this study, we recognize segments with linearity values above a threshold  $TH_l$  and number of points more than  $TH_n$  as tree stems and branches (non-photosynthetic components of trees). The thresholds  $TH_l$  and  $TH_n$  are determined by trial-and-error (Figure 7).



**Figure 7.** An example of a Heat map of acquired accuracy for the PD-2 dataset to locate proper  $TH_l$  and  $TH_n$  values.

#### 3.4. Tile Processing

Point cloud manipulation can be computationally intensive. Data measured from TLS can contain millions to billions of points [11], thus efficient manipulation is important. As used in other software and studies [56,57], we partition the whole point cloud into tiles, and process each tile individually and in parallel. We develop three routines to partition a point cloud; uniform voxels, octree, and 2D tiles (Figure 8). Point cloud voxelization is a commonly used approach to partition a point cloud into local spaces (e.g., [58]). It is fast to generate, and works with satisfaction for a point cloud with homogeneous density distribution. For a point cloud with varied point density, such as data from single-scan TLS measurement, Octree is better at balancing the needed computational resources in each tile. Two-dimensional tiles (Figure 8c) are often deployed for large-scale datasets, such as ALS data [56]. By tiling the point cloud, the processing can be run in parallel, and the memory consumption is distributed to each computer worker, avoiding memory overcommitment for a single worker. In each tile, a *k*-*d* tree structure is constructed to enable fast neighborhood and range searching. Besides, individually processed tiles are further merged afterwards [57]. Figure 9 shows an example of point cloud processing on tiles and merging afterwards. The tile processing greatly facilitates and accelerates point cloud manipulation.



Figure 8. Point cloud tiling. (a) Uniform voxels. (b) Octree partition. (c) 2D tiles on XY plane.



Figure 9. Point cloud tile processing and merging. (a) DSM results on tiles. (b) Results after tile merging.

#### 3.5. Random Forest Classification

In this study, we additionally apply supervised machine learning classification as a method comparison [1,18]. Only point-wise geometric features are used, and the random forest (RF) classifier is explored [59]. Previous studies showed that the RF model is particularly effective for separating tree photosynthetic and non-photosynthetic components from point cloud data [21,30]. RF is a decision

tree-based ensemble learning method that was proposed by Breiman et al. [59]. The learned model is a collection of weak models. Multiple decision trees are grown on random subsets of training data. The class determination is based on a majority votes approach. Two necessary foundational parameters need to be specified for the RF model; the number of classification trees  $n_{trees}$  and the number of features  $m_{ft}$  used at each node. We set the number of trees  $n_{trees}$  as 100 [30]. Another parameter  $m_{ft}$  is determined as  $\sqrt{p}$ , where p denotes the number of input features, as suggested by [59]. Thirty-two 3D and 2D geometrical features are calculated for each point (Table 1). Two-dimensional features are calculated from projected grids on the XY plane. Many of these features are defined and used in other point cloud classification studies [60]. Training data are manually selected for each dataset. The spatial distribution of training data follows a strategy used in Ma et al. [1]. Points that belong to different classes at the 25%, 50%, and 75% height of the bounding box of the forest plot are manually identified in the open source software CloudCompare (CloudCompare 2.9.1, 2018). Training points near the ground are additionally selected if low vegetation is present in the dataset. The RF classification was performed using the Statistics and Machine Learning Toolbox in Matlab 2017a (The MathWorks, Inc., Natick, Massachusetts, United States).

No.	Feature	Description				
1	linearity <sub>3D</sub>	linear saliency $(EV_1 - EV_2)/EV_1$ .				
2	planarity <sub>3D</sub>	planar saliency $(EV_2 - EV_3)/EV_1$ .				
3	scattering <sub>3D</sub>	volumetric saliency $EV_3/EV_1$ .				
4	omnivariance <sub>3D</sub>	volume of the neighborhood $(EV_1 * EV_2 * EV_3)/3$ .				
5	anisotropy <sub>3D</sub>	$(\breve{E}V_1 - EV_3)/EV_1$ .				
6	eigenentropy3D	$-\sum_{n=1}^{3} EV_n * log(Ev_n)$				
7	$sum_EV_{3D}$	$\sum_{n=1}^{3} EV_n$ .				
8	surface_variation <sub>3D</sub>	change of curvature $EV_{min} / (\sum EV)$ .				
9	X value $X_{3D}$	X coordinate of the point.				
10	Y value $Y_{3D}$	Y coordinate of the point.				
11	Z value $Z_{3D}$	height of the point.				
12	density <sub>3D</sub>	local point density.				
13	verticality <sub>3D</sub>	$1 - NV_z$ .				
14	$\Delta Z_{knn_{3D}}$	height difference of local neighborhood.				
15	$\sigma Z_{knn_{3D}}$	standard deviation of heights of local neighborhood.				
16	$N_{z3D}$	z-component of the normal vector NV.				
17	radius <sub>2D</sub>	radius of local neighborhood.				
18	density <sub>2D</sub>	local point density.				
19	$sum_EV_{2D}$	$\sum_{n=1}^{2} EV_n.$				
20	EV_ratio <sub>2D</sub>	$EV_2/EV_1.$				
21	cell_density <sub>2D</sub>	density of projected 2D cells.				
22	skewness <sub>2D</sub>	skewness of point heights in each cell.				
23	kurtosis <sub>2D</sub>	kurtosis of point heights in each cell.				
24	Max_z <sub>2D</sub>	maximum of heights of points in each cell.				
25	Min_z <sub>2D</sub>	minimum of heights of points in each cell.				
26	Mean_z <sub>2D</sub>	average height of points in each cell.				
27	Median_z <sub>2D</sub>	median height of points in each cell.				
28	$EV_{13D}$	first eigenvalue of 3D covariance matrix.				
29	$EV_{23D}$	second eigenvalue of 3D covariance matrix.				
30	$EV_{3D}$	third eigenvalue of 3D covariance matrix.				
31	$EV_{12D}$	first eigenvalue of 2D covariance matrix.				
32	$EV_{2D}$	second eigenvalue of 2D covariance matrix.				

**Table 1.** Thirty-two features extracted from the point cloud. EV denotes the eigenvalue and NV is the normal vector. Three EVs are sorted in descending order.  $EV_1$  defines the first eigenvalue, and so on.

We note that our aim is not to achieve the best performance from the supervised machine learning method. Some optimization strategies such as feature selection, hyper-parameters fine-tuning, and model pruning may further improve the classification performance. However, the deployed RF classier, in combination with our crafted features, has shown promising results in a machine learning benchmark study for wood-leaf separation [30].

## 3.6. Evaluation

The performance of our DSM method and the supervised RF method is evaluated based on three statistical indexes; sensitivity, specificity, and accuracy. Sensitivity measures the correctly classified positive samples (true positive rate, *TPR*). In this study, it represents the correct rate for non-photosynthetic components such as wood and other points. Specificity gives the true negative rate (*TNR*), thus it measures the correct rate for photosynthetic components (i.e., leaf points). Accuracy (*ACC*) gives the overall correctness by

Sensitivity = 
$$\frac{TP}{TP + FN}$$
 (5)

Specificity = 
$$\frac{TN}{TN + FP}$$
 (6)

$$Accuracy = \frac{TP + TN}{P + N},$$
(7)

where *P* and *N* are the number of real positive (non-photosynthetic components) and negative (photosynthetic components) samples. *TP* and *TN* are the correctly identified positive and negative samples, respectively.

# 4. Results

#### 4.1. Single Tree Data

A visual inspection of the separation result from the proposed DSM method for the SD-1 dataset is shown in Figure 10a. We observe that some small branch sections at the bottom were misclassified as leaf points. Some leaf points on the canopy surface were also wrongly labeled as wood points. Nevertheless, the branches inside the canopy were generally separated from leaves successfully. Quantitatively, the overall classification accuracy of our DSM method reached 86.9% with fixed neighborbood sizes, and 88.5% when using adaptive neighborhood sizes, which were higher than that of 82.1% and 83.9% from the supervised RF method (Table 2). The achieved sensitivity from the DSM method were higher than the RF method, while the specificity was of the same class. This indicates that our method is particularly confident in detecting non-photosynthetic components.

**Table 2.** Results summary of various methods on all datasets. Bold numbers are the best results among methods. (F) denotes the results acquired with fixed neighborhood sizes, and (A) stands for adaptive neighborhood sizes. \* Training and validation points have an equal split of photosynthetic and non-photosynthetic components.

Evaluation	Method	Dataset				
		SD-1	PD-1	PD-2	PD-3	PD-4
Number of points		553,556	16,259,081	2,013,331	3,901,367	1,269,318
Number of RF training points *		86,618	117,534	53 <i>,</i> 872	59,114	31,776
Number of validation points *		279,580	15,492,926	1,648,514	77,462	945,262
	DSM(F)	94.7	97.9	92.2	97.5	95.5
	DSM(A)	93.7	96.4	90.5	96.7	95.2
Sensitivity (%)	RF(F)	81.5	73.1	88.4	86.2	95.7
	RF(A)	86.6	84.7	92.5	91.8	95.9
	Intensity	-	92.1	-	-	-
	DSM(F)	79.1	74.5	64.8	81.5	76.1
	DSM(A)	83.3	81.1	73.0	87.4	79.4
Specificity (%)	RF(F)	82.7	92.9	77.2	97.1	70.3
	RF(A)	81.1	87.9	74.0	95.8	73.5
	Intensity	-	83.1	-	-	-
	DSM(F)	86.9	86.2	78.5	89.5	85.8
	DSM(A)	88.5	88.7	81.8	92.0	87.3
Overall accuracy (%)	RF(F)	82.1	83.0	82.8	91.7	83.0
2 * *	RF(A)	83.9	86.3	83.2	93.8	84.7
	Intensity	-	87.6	-	-	-



**Figure 10.** Separation results with adaptive neighborhoods from the DSM method for (**a**) SD-1, (**b**) PD-1, (**c**) PD-2, (**d**) PD-3, and (**e**) PD-4. Photosynthetic and non-photosynthetic components are colored by green and brown, respectively.

## 4.2. Plot-Level Data

Figure 10b–e gives an overview of the separation results for our four plot-level datasets from PD-1 to PD-4. We observe that most non-photosynthetic components were resoundingly separated from photosynthetic components, regardless of their spatial orientations. The overall accuracy for the four plot-level datasets ranged from 78.5% to 89.5% in our DSM method with fixed neighborhood sizes, and from 81.8% to 92.0% with adaptive neighborhood sizes (Table 2). The PD-2 dataset had the worst overall accuracy. Its associated specificity was only 64.8%/73.0%, meaning that many

non-photosynthetic components were mislabeled as photosynthetic points. This deduction can also be observed in Figure 10c, in which some high stem points were mislabeled as leaf points, potentially due to the low point density on the top and severe occlusions in the single-scan TLS data. Our DSM method had a sensitivity from 92.2% to 97.9% with fixed neighborhood sizes, and from 90.5% to 96.7% with adaptive ones, which were overall higher than the RF method. However, the supervised RF had significantly higher specificity than the DSM method, except the PD-4 dataset. The average overall accuracy for the DSM and RF method was 85.0% (87.5% with adaptive neighbors) and 85.1% (87.0% with adaptive neighbors), respectively.

The PD-1 dataset contains useful intensity information after radiometrical calibration. The intensity calibration and separation thresholding is described in Section 2.2. As a result, the intensity method achieved an overall accuracy of 87.6%, with a sensitivity of 92.1% and a specificity of 83.1% (Table 2). The performance of the intensity method was similar to our DSM method.

## 5. Discussion

In this study, tree photosynthetic and non-photosynthetic components are automatically separated by the proposed dynamic segment merging method. The proposed method is fully unsupervised, thus not requiring any training data and user interventions. The core concept of the DSM method is a robust point cloud segmentation routine. We have shown in this study that the DSM method has been successfully used in separating tree photosynthetic and non-photosynthetic components from point cloud data. To some extent, we solve an ongoing challenge in an unsupervised manner, and overcome the bottlenecks in other methods such as calibrating laser intensities or manually selecting training data [20]. In the following subsections, we discuss some vital inputs of our algorithm, its performance, and future applications. At the same time, we address several challenges about the separation of tree photosynthetic and non-photosynthetic components from point cloud data.

## 5.1. Calculation of Normal Vectors

Calculation of normal vectors is crucial in this study. Figure 11 gives an example of normal vectors calculated by the adaptive method, compared to those with fixed neighborhoods for the PD-4 dataset. This dataset was acquired with a hand-held scanner, with a lower position accuracy compared to TLS. Points on the stem surfaces are not smooth due to low accuracy, thus giving a good example of the performance of normal vector estimation. Visually, the normal vectors are more homogeneous on the stems after using the optimal neighborhood selection method.



**Figure 11.** Normal vector estimation. (a) Normal vectors estimated with fixed neighborhoods, and colored by  $N_z$ . (b) Optimal neighborhood size for each point. (c) Normal vectors estimated with optimal neighborhoods, and colored by  $N_z$ .

The accuracy for the PD-4 dataset was improved from 85.8% to 87.3%, proving the positive effects in estimating normal vectors (Figure 11). A similar trend was also found for other datasets and for the performances of the supervised RF method. The average overall accuracy for all datasets was improved from 85.0% to 87.5% for the DSM method, and from 85.1% to 87.0% for the RF model. The improvements are more significant for our DSM method. Theoretically, this is expected as our DSM method is a kind of segmentation approach which heavily relies on the quality of normal vectors. The selection of optimal neighborhoods is inherently a multi-scale analysis. The local irregularity is smoothed in a large scale, resulting in more homogeneous normal vectors. Consequently, these branches can be identified, and the specificity was broadly improved. The supervised RF method is a machine learning classification technique that relies more on the feature distinguishability and the classifier mechanism.

However, robust and efficient estimation of normal vectors from complex scenes is still a challenging task. Although methods with adaptive neighborhood searching can be effective in quality, they are computationally intensive. The computation time and memory consumption are significantly higher for large datasets. For example, the computation time is about six times longer for the adaptive neighborbood than the fixed neighborbood in our SD-1 dataset, when using seven CPU threads. This is still a bottleneck for most applications [42]. A possible solution is to explore GPU processing using a parallel implementation [42].

## 5.2. Algorithm Performance

The DSM method achieved generally constant performances across different datasets. The standard deviation of accuracy was 4.1% with fixed neighborhood sizes and 3.7% with adaptive sizes. The accuracy was low for the PD-2 dataset (Table 2), mostly because it was acquired with single-scan TLS, which resulted in severe occlusions and shadows in the point cloud, and low point density at distances away from the scanner. Consequently, the branches were not well represented as linear segments in the point cloud, and our DSM method relies on segmenting and detecting linear objects. In addition, we observed that our proposed DSM method achieved higher sensitivity than specificity (Table 2) across all datasets. This indicates a systematic tendency of our method for having confidence in detecting non-photosynthetic components over photosynthetic components. In this study, a low specificity means that many wood points were mislabeled as leaf points, while in general high sensitivity states our method is good at detecting wood points robustly. We note that this can be an effect of the irregularity of branch structures. If a section of a branch is severely deformed in terms of growing orientation, it will not be grouped as one segment as its normal vectors are corrupted. This challenge was partly mitigated when using normal vectors with optimal neighborhoods. The specificity ranged from 73.0% to 87.4%, compared to those with fixed neighborhoods which ranged from 64.8% to 81.5%.

In this study, we also deployed the supervised machine learning method RF as a comparison of our DSM method to the state-of-the-art methods. Overall, the RF method achieved similar accuracy compared to the DSM method. However, it required tedious works to delineate training points. These training points must be representative, as they will greatly influence the outcomes of machine learning methods. For example, if we exclude training points on the bushes for the dataset PD-2, the RF results dropped dramatically. In contrast to our method, the RF achieved better model specificity than sensitivity. Ma et al. [1] used a supervised GMM model to separate photosynthetic and non-photosynthetic components from TLS data. Their overall accuracy for single tree data was 82.6%, and further improved to 99.6% after applying a series of post filters. Yun et al. [18] deployed a SVM model for four single tree dataset SD-1 was 88.5%. The RF method had an accuracy of 83.9%. For plot-level datasets, Ma et al. [1] reached a preliminary accuracy ranging from 65.8% to 75.2%, which was further improved to a range from 84.3% to 97.8%. Our DSM results for plot-level datasets were in the same standard (81.8% to 92.0%) compared to the improved results from Ma et al. [1]. We note

that our implemented RF achieved higher accuracy than the GMM model in Ma et al. [1]. A potential explanation is that we crafted more features, and the RF model may be more efficient than GMM for such a task [30].

The proposed DSM method effectively partitions forest point cloud into photosynthetic and non-photosynthetic components. It dynamically adjusts the merging process, while the conventional region growing method uses a static strategy. For complex scenes such as forests, the conventional region growing method has difficulties in distinguishing gradual changing regions. Therefore, it under-segmented trees, as illustrated in Figure 12b. For a fair comparison, we used the same normal vectors and searching radius to compare our DSM method. Figure 12a demonstrates that our DSM method can effectively separate leaf and wood points, where conventional region growing failed. We further show a detailed inspection of PD-3 in a magnified view (Figure 13). We chose to show fine-detail information of the PD-3 plot because its complexity in tree structures is instrumental in testing the performance of our DSM method. In addition, we used a simple constraint in this study-the deviation of z-component of normal vectors (verticality)-to group tree trunks and branches. We have shown that it is effective for irregular branches (Figure 13). It is further noted that the constraint in our DSM method only serves a role to check if an adjacent segment is qualified, while the merging is determined by the similarity ranking (Equation (3)). In a nutshell, our DSM method is less sensitive to the selected constraints, which is one other distinction compared to conventional region growing routines.



**Figure 12.** Comparison of segmentation for the SD-1 dataset. Each segment is randomly colored. (a) Segmentation from the proposed DSM method. (b) Segmentation from the conventional region growing method.



Figure 13. A close inspection of results for the PD-3 dataset.

#### 5.3. Challenges in Components Separation

Apart from the methodology shortage, there are still many challenges in separating tree photosynthetic and non-photosynthetic components from a point cloud. Forest structures are multifarious in nature. The acquired point cloud will be affected by severe occlusions. This results in data shallows and non-uniform density distributions. For intensity-based methods [2,19], this effect is less significant. However, geometry-based methods will suffer from these challenges. For machine learning-based methods, the crafted features should cope with non-uniform density, and be robust and distinctive to guide the model. However, this is still challenging for complex scenes. Recently developed deep networks on point clouds may further help to extract higher level features [61,62]. Nevertheless, these models are still in the very early development stage, and hardly work with high-density TLS data.

Another bottleneck of TLS data processing is to deal with the large amount of data. With millions and billions of data [11], point cloud manipulation can be intensive. Efficient point cloud management and processing are mainly targeted at ALS data [56]. In this study, we briefly implemented tile processing together with the k - d tree structure to accelerate TLS processing. Future optimization is still imperative.

## 5.4. Future Applications of the DSM Method

In this study, we developed an efficient method applied to a wood–leaf separation task. This processing is a crucial step in reducing uncertainties in TLS-derived estimates of above ground biomass [20]. Our method can greatly facilitate the automation in this processing. For example, Figure 14 gives an example on how our method was applied to the reconstruction and parameter retrieval of the entire *Erythrophleum fordii* tree (SD-1). Leaf points were detected and removed automatically by using our DSM method (Figure 14b). Subsequently, the entire tree was reconstructed by the Quantitative Structure Model (QSM) [52]. Some vital parameters can then be extracted from the QSM, such as total volume (581.6L), trunk volume (436.6L), and branch length (84.4m). On the other hand, our algorithm can be easily deployed for other applications. For example, the presented pipeline in this study can be adapted to detect tree stems by simply filtering linear segments based on their spatial orientations and sizes. Based on the detected tree stems, downstream processing such as diameter estimation [63] and stem curve [9] retrieval can be realized.



**Figure 14.** An example of our DSM method applied to detect wood points of the entire *Erythrophleum fordii* tree (**a**). Detected wood components are shown in (**b**), and the corresponding Quantitative Structure Model (QSM) is given in (**c**).

## 6. Conclusions

In this study, we present a fully automatic and unsupervised approach to separate tree photosynthetic and non-photosynthetic components from point cloud data. This geometry-based method is free of user interventions and dispenses with manual delineation of training data, which is a tedious prerequisite for the supervised machine learning algorithms. Our method is based on a robust

dynamic point cloud segmentation routine. The point cloud is firstly partitioned into meaningful segments by using the proposed DSM method. Non-photosynthetic segments such as stems and branches are then identified by estimating their linear feature saliency. The approach is tested for a single tree dataset and four plot-level datasets. These datasets cover single-scan TLS, multi-scan TLS, hand-held laser scanning, varied terrain conditions, and various tree species. The achieved accuracy reached 88.5% for the single tree dataset, and ranged from 81.8% to 92.0% for plot-level datasets. We also compared our results to a supervised machine learning method. In addition, we show that point cloud structuring enables efficient point cloud manipulation even for large datasets. Furthermore, we have discussed the performances and some challenges of separating photosynthetic and non-photosynthetic components in nature forests.

The extensive experiments on various datasets suggest that the proposed DSM method is efficient, and can be equally as effective as supervised machine learning methods. Nevertheless, the distinct advantage of our method lies in that it is unsupervised and fully automatic. Our work highlights the potential of unsupervised separation of wood and leaf points even in plot-level analysis.

**Author Contributions:** D.W. designed the study, performed the data analyses and prepared the manuscript. J.B. and Z.M. acquired the datasets and manually classified point clouds. J.B. designed and performed radiometrical calibration. H.L. contributed in research discussions. M.H., Y.P. and N.P. supervised the study. All authors contributed to editing and reviewing the manuscript.

Acknowledgments: Special thanks are due to referees for their constructive reviews that improved the manuscript significantly. The study was supported by the project "The influence of Biomass and its change on landSLIDE activity" (BioSLIDE) within the research program Earth System Sciences (ESS) of the Austrian Academy of Science (Österreichische Akademie der Wissenschaften, ÖAW). Partial fund was jointly provided by the ÖAW, the Ministry of Education, Youth and Sports of the Czech Republic and the Slovak Research and Development Agency under the contract No. DS-2016-0040. The authors also thank the Austrian Research Promotion Agency (FFG) for providing financial support via the project No. 860021.

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

## References

- Ma, L.; Zheng, G.; Eitel, J.U.; Moskal, L.M.; He, W.; Huang, H. Improved salient feature-based approach for automatically separating photosynthetic and nonphotosynthetic components within terrestrial lidar point cloud data of forest canopies. *IEEE Trans. Geosci. Remote Sens.* 2016, 54, 679–696. [CrossRef]
- 2. Béland, M.; Baldocchi, D.D.; Widlowski, J.L.; Fournier, R.A.; Verstraete, M.M. On seeing the wood from the leaves and the role of voxel size in determining leaf area distribution of forests with terrestrial LiDAR. *Agricult. For. Meteorol.* **2014**, *184*, 82–97. [CrossRef]
- Levick, S.R.; Hessenmöller, D.; Schulze, E.D. Scaling wood volume estimates from inventory plots to landscapes with airborne LiDAR in temperate deciduous forest. *Carbon Balance Manag.* 2016, 11, 7. [CrossRef] [PubMed]
- 4. Zheng, G.; Moskal, L.M.; Kim, S.H. Retrieval of effective leaf area index in heterogeneous forests with terrestrial laser scanning. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 777–786. [CrossRef]
- Calders, K.; Newnham, G.; Burt, A.; Murphy, S.; Raumonen, P.; Herold, M.; Culvenor, D.; Avitabile, V.; Disney, M.; Armston, J.; et al. Nondestructive estimates of above-ground biomass using terrestrial laser scanning. *Methods Ecol. Evol.* 2015, *6*, 198–208. [CrossRef]
- 6. Chen, J.M.; Black, T. Defining leaf area index for non-flat leaves. *Plant Cell Environ.* **1992**, *15*, 421–429. [CrossRef]
- 7. Asner, G.P.; Scurlock, J.M.; A Hicke, J. Global synthesis of leaf area index observations: Implications for ecological and remote sensing studies. *Glob. Ecol. Biogeogr.* **2003**, *12*, 191–205. [CrossRef]
- Li, S.; Dai, L.; Wang, H.; Wang, Y.; He, Z.; Lin, S. Estimating leaf area density of individual trees using the point cloud segmentation of terrestrial LiDAR data and a voxel-based model. *Remote Sens.* 2017, *9*, 1202. [CrossRef]
- 9. Wang, D.; Hollaus, M.; Puttonen, E.; Pfeifer, N. Automatic and self-adaptive stem reconstruction in landslide-affected forests. *Remote Sens.* **2016**, *8*, 974. [CrossRef]

- 10. Liang, X.; Kankare, V.; Yu, X.; Hyyppä, J.; Holopainen, M. Automated stem curve measurement using terrestrial laser scanning. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 1739–1748. [CrossRef]
- Liang, X.; Kankare, V.; Hyyppä, J.; Wang, Y.; Kukko, A.; Haggrén, H.; Yu, X.; Kaartinen, H.; Jaakkola, A.; Guan, F.; et al. Terrestrial laser scanning in forest inventories. *ISPRS J. Photogramm. Remote Sens.* 2016, 115, 63–77. [CrossRef]
- 12. Maas, H.; Bienert, A.; Scheller, S.; Keane, E. Automatic forest inventory parameter determination from terrestrial laser scanner data. *Int. J. Remote Sens.* **2008**, *29*, 1579–1593. [CrossRef]
- 13. Liang, X.; Litkey, P.; Hyyppä, J.; Kaartinen, H.; Vastaranta, M.; Holopainen, M. Automatic stem mapping using single-scan terrestrial laser scanning. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 661–670. [CrossRef]
- You, L.; Tang, S.; Song, X.; Lei, Y.; Zang, H.; Lou, M.; Zhuang, C. Precise measurement of stem diameter by simulating the path of diameter tape from terrestrial laser scanning data. *Remote Sens.* 2016, *8*, 717. [CrossRef]
- 15. Chen, Q.; Gong, P.; Baldocchi, D.; Tian, Y.Q. Estimating basal area and stem volume for individual trees from lidar data. *Photogramm. Eng. Remote Sens.* **2007**, *73*, 1355–1365. [CrossRef]
- Feliciano, E.A.; Wdowinski, S.; Potts, M.D. Assessing mangrove above-ground biomass and structure using terrestrial laser scanning: A case study in the Everglades National Park. *Wetlands* 2014, 34, 955–968. [CrossRef]
- 17. Tao, S.; Guo, Q.; Su, Y.; Xu, S.; Li, Y.; Wu, F. A geometric method for wood-leaf separation using terrestrial and simulated lidar data. *Photogramm. Eng. Remote Sens.* **2015**, *81*, 767–776. [CrossRef]
- 18. Yun, T.; An, F.; Li, W.; Sun, Y.; Cao, L.; Xue, L. A novel approach for retrieving tree leaf area from ground-based LiDAR. *Remote Sens.* **2016**, *8*, 942. [CrossRef]
- 19. Pfennigbauer, M.; Ullrich, A. Improving quality of laser scanning data acquisition through calibrated amplitude and pulse deviation measurement. *SPIE Def. Secur. Sens. Int. Soc. Opt. Photonics* **2010**, 7684, 76841F.
- 20. Disney, M.I.; Boni Vicari, M.; Burt, A.; Calders, K.; Lewis, S.L.; Raumonen, P.; Wilkes, P. Weighing trees with lasers: Advances, challenges and opportunities. *Interface Focus* **2018**, *8*. [CrossRef] [PubMed]
- Zhu, X.; Skidmore, A.K.; Darvishzadeh, R.; Niemann, K.O.; Liu, J.; Shi, Y.; Wang, T. Foliar and woody materials discriminated using terrestrial LiDAR in a mixed natural forest. *Int. J. Appl. Earth Observ. Geoinf.* 2018, 64, 43–50. [CrossRef]
- 22. Kaasalainen, S.; Krooks, A.; Kukko, A.; Kaartinen, H. Radiometric calibration of terrestrial laser scanners with external reference targets. *Remote Sens.* **2009**, *1*, 144–158. [CrossRef]
- Calders, K.; Disney, M.I.; Armston, J.; Burt, A.; Brede, B.; Origo, N.; Muir, J.; Nightingale, J. Evaluation of the Range Accuracy and the Radiometric Calibration of Multiple Terrestrial Laser Scanning Instruments for Data Interoperability. *IEEE Trans. Geosci. Remote Sens.* 2017, 55, 2716–2724. [CrossRef]
- 24. Höfle, B.; Pfeifer, N. Correction of laser scanning intensity data: Data and model-driven approaches. *ISPRS J. Photogramm. Remote Sens.* **2007**, *62*, 415–433. [CrossRef]
- Kaasalainen, S.; Jaakkola, A.; Kaasalainen, M.; Krooks, A.; Kukko, A. Analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods. *Remote Sens.* 2011, 3, 2207–2221. [CrossRef]
- Li, Z.; Douglas, E.; Strahler, A.; Schaaf, C.; Yang, X.; Wang, Z.; Yao, T.; Zhao, F.; Saenz, E.J.; Paynter, I.; et al. Separating leaves from trunks and branches with dual-wavelength terrestrial LiDAR scanning. In Proceedings of the 2013 IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS), Melbourne, Australia, 21–26 July 2013; pp. 3383–3386.
- 27. Hakala, T.; Suomalainen, J.; Kaasalainen, S.; Chen, Y. Full waveform hyperspectral LiDAR for terrestrial laser scanning. *Opt. Express* **2012**, *20*, 7119–7127. [CrossRef] [PubMed]
- Vauhkonen, J.; Hakala, T.; Suomalainen, J.; Kaasalainen, S.; Nevalainen, O.; Vastaranta, M.; Holopainen, M.; Hyyppä, J. Classification of spruce and pine trees using active hyperspectral LiDAR. *IEEE Geosci. Remote Sens. Lett.* 2013, 10, 1138–1141. [CrossRef]
- Belton, D.; Moncrieff, S.; Chapman, J. Processing tree point clouds using Gaussian mixture models. In Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Antalya, Turkey, 11–13 November 2013; pp. 11–13.

- Wang, D.; Hollaus, M.; Pfeifer, N. Feasibility of machine learning methods for separating wood and leaf points from terrestrial laser scanning data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2017, *IV-2/W4*, 157–164. [CrossRef]
- 31. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* **2016**, *8*, 501. [CrossRef]
- 32. Hackenberg, J.; Wassenberg, M.; Spiecker, H.; Sun, D. Non destructive method for biomass prediction combining TLS derived tree volume and wood density. *Forests* **2015**, *6*, 1274–1300. [CrossRef]
- 33. Hyyppä, J.; Liang, X. Project Benchmarking on Terrestrial Laser Scanning for Forestry Applications. Available online: http://www.eurosdr.net/research/project/project-benchmarking-terrestrial-laserscanning-forestry-applications (accessed on 18 February 2018).
- 34. Adams, R.; Bischof, L. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 641–647. [CrossRef]
- 35. Huang, Z.; Zhang, J.; Li, X.; Zhang, H. Remote sensing image segmentation based on dynamic statistical region merging. *Optik Int. J. Light Electron Opt.* **2014**, *125*, 870–875. [CrossRef]
- 36. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. *arXiv* **2017**, arXiv:1711.09869.
- 37. Peng, B.; Zhang, L.; Zhang, D. Automatic image segmentation by dynamic region merging. *IEEE Trans. Image Process.* **2011**, *20*, 3592–3605. [CrossRef] [PubMed]
- 38. Rabbani, T.; Van Den Heuvel, F.; Vosselmann, G. Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2006**, *36*, 248–253.
- 39. Höfle, B. Radiometric correction of terrestrial LiDAR point cloud data for individual maize plant detection. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 94–98. [CrossRef]
- 40. Lari, Z.; Habib, A.; Kwak, E. An adaptive approach for segmentation of 3D laser point cloud. *ISPRS Workshop Laser Scanning* **2011**, *XXXVIII-5/W12*, 29–31. [CrossRef]
- Klasing, K.; Althoff, D.; Wollherr, D.; Buss, M. Comparison of surface normal estimation methods for range sensing applications. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'09), Kobe, Japan, 12–17 May 2009; pp. 3206–3211.
- Liu, M.; Pomerleau, F.; Colas, F.; Siegwart, R. Normal estimation for pointcloud using GPU based sparse tensor voting. In Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 11–14 December 2012; pp. 91–96.
- 43. Garland, M. *Quadric-Based Polygonal Surface Simplification;* Technical Report; School of Computer Science, Carnegie Mellon University: Pittsburgh, PA, USA, 1999.
- Bazazian, D.; Casas, J.R.; Ruiz-Hidalgo, J. Fast and robust edge extraction in unorganized point clouds. In Proceedings of the 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Adelaide, Australia, 23–25 November 2015; pp. 1–8.
- Weinmann, M.; Jutzi, B.; Mallet, C. Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2014, 2, 181. [CrossRef]
- 46. Chen, M.; Wan, Y.; Wang, M.; Xu, J. Automatic stem detection in terrestrial laser scanning data with distance-adaptive search radius. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2968–2979. [CrossRef]
- 47. Xia, S.; Wang, C.; Pan, F.; Xi, X.; Zeng, H.; Liu, H. Detecting stems in dense and homogeneous forest using single-scan TLS. *Forests* **2015**, *6*, 3923–3945. [CrossRef]
- 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. [CrossRef]
- 49. Kaick, O.V.; Fish, N.; Kleiman, Y.; Asafi, S.; Cohen-Or, D. Shape segmentation by approximate convexity analysis. *ACM Trans. Graph.* **2014**, *34*, 4. [CrossRef]
- 50. Vosselman, G.; Coenen, M.; Rottensteiner, F. Contextual segment-based classification of airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.* 2017, 128, 354–371. [CrossRef]
- 51. 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. [CrossRef]

- Raumonen, P.; Kaasalainen, M.; Åkerblom, M.; Kaasalainen, S.; Kaartinen, H.; Vastaranta, M.; Holopainen, M.; Disney, M.; Lewis, P. Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sens.* 2013, 5, 491–520. [CrossRef]
- Pöchtrager, M.; Styhler-Aydin, G.; Döring-Williams, M.; Pfeifer, N. Automated reconstruction of historic roof structures from point clouds-development and examples. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2017, *IV-2/W2*, 195–202.
- 54. Filin, S.; Pfeifer, N. Neighborhood systems for airborne laser data. *Photogramm. Eng. Remote Sens.* 2005, 71, 743–755. [CrossRef]
- 55. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, 24, 603–619. [CrossRef]
- 56. Pfeifer, N.; Mandlburger, G.; Otepka, J.; Karel, W. OPALS—A framework for Airborne Laser Scanning data analysis. *Comput. Environ. Urban Syst.* **2014**, *45*, 125–136. [CrossRef]
- 57. Vosselman, G. Point cloud segmentation for urban scene classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *1*, 257–262. [CrossRef]
- Papon, J.; Abramov, A.; Schoeler, M.; Wörgötter, F. Voxel cloud connectivity segmentation-supervoxels for point clouds. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2027–2034.
- 59. Breiman, L. Random forests. Machine learning 2001, 45, 5-32. [CrossRef]
- 60. 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. [CrossRef]
- 61. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Comput. Vis. Pattern Recognit.* **2017**, *1*, 4.
- 62. Li, Y.; Bu, R.; Sun, M.; Chen, B. PointCNN. arXiv 2018, arXiv:1801.07791.
- 63. Wang, D.; Kankare, V.; Puttonen, E.; Hollaus, M.; Pfeifer, N. Reconstructing stem cross section shapes from terrestrial laser scanning. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 272–276. [CrossRef]



© 2018 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/).