

Article

Highly Accurate Tree Models Derived from Terrestrial Laser Scan Data: A Method Description

Jan Hackenberg ^{1,*}, Christopher Morhart ¹, Jonathan Sheppard ¹, Heinrich Spiecker ¹ and Mathias Disney ^{2,3}

¹ Chair of Forest Growth, University of Freiburg, Tennenbacher Str. 4, 79106 Freiburg, Germany; E-Mails: christopher.morhart@iww.uni-freiburg.de (C.M.);

jonathan.sheppard@iww.uni-freiburg.de (J.S.); instww@uni-freiburg.de (H.S.)

² Department of Geography, University College London, Gower Street, London WC1E 6BT, UK; E-Mail: mathias.disney@ucl.ac.uk

³ NERC National Centre for Earth Observation (NCEO), UK

* Author to whom correspondence should be addressed; E-Mail: jan.hackenberg@iww.uni-freiburg.de; Tel.: +49-761-203-8589; Fax: +49-761-203-3740.

Received: 4 March 2014; in revised form: 15 April 2014 / Accepted: 13 May 2014 /

Published: 22 May 2014

Abstract: This paper presents a method for fitting cylinders into a point cloud, derived from a terrestrial laser-scanned tree. Utilizing high scan quality data as the input, the resulting models describe the branching structure of the tree, capable of detecting branches with a diameter smaller than a centimeter. The cylinders are stored as a hierarchical tree-like data structure encapsulating parent-child neighbor relations and incorporating the tree's direction of growth. This structure enables the efficient extraction of tree components, such as the stem or a single branch. The method was validated both by applying a comparison of the resulting cylinder models with ground truth data and by an analysis between the input point clouds and the models. Tree models were accomplished representing more than 99% of the input point cloud, with an average distance from the cylinder model to the point cloud within sub-millimeter accuracy. After validation, the method was applied to build two allometric models based on 24 tree point clouds as an example of the application. Computation terminated successfully within less than 30 min. For the model predicting the total above ground volume, the coefficient of determination was 0.965, showing the high potential of terrestrial laser-scanning for forest inventories.

Keywords: TLS; terrestrial laser scanning; LiDAR; forestry; tree; stem; branch; volume; cylinder fitting

1. Introduction

Estimation of total above ground volume within traditional forest inventory methods is often predicted from volume tables according to the diameter at 1.3 m, referred to as the diameter at breast height (DBH), and height (h) or based on empirical volume functions, known as allometric models, commonly using independent predictor variables, such as DBH and h . These allometric models typically take the form of a power, exponential or polynomial model [1] with one or more independent predictor variables. An example of the most commonly used power function is shown in Equation (1), where v is the volume, d the DBH and a and b are the regression coefficients.

$$v = ad^b \quad (1)$$

Individual tree volume can also be calculated by means of a more general formula as a function of DBH and h , as given in Equation (2) [2], with the inclusion of a form factor (f) describing the shape and taper of the trunk, which must be determined for every tree species.

$$v = \frac{1}{4}\pi d^2 h f \quad (2)$$

To date, various functions for the estimation of above ground volume have been collated and published [3,4]. However, species-specific allometric models are still not always available, especially if regional variation of growth patterns, species provenance, spacing and site variability are to be taken into account. In this case, the analyst must rely on the use of destructive methods to obtain the volume of a tree. One such method is through the use of a xylometer [2]. Although this method is accepted to be accurate, drawbacks include:

1. the destructive and time-consuming nature of the method;
2. logs must be transported to the xylometer;
3. logs might absorb water;
4. the only obtained parameter is volume

Other destructive methods are cheaper, but slightly less accurate, such as the subdivision of a stem into sections and the prediction of the volume of each section as a truncated cone. Similarly, some simple, but less accurate, formulas have been suggested to estimate the volume of trees [2], as demonstrated in Equation (3) with g_i determining the cross-sectional areas at relative lengths i and l being the total length of the tree:

$$v = \frac{g_{0.1} + g_{0.3} + g_{0.5} + g_{0.7} + g_{0.9}}{5} l \quad (3)$$

Despite these traditional methods, other approaches have been carried out in the field of remote sensing. A recent sensor technology, which has been used successfully to retrieve forest structural

parameters, is light detection and ranging (LiDAR) utilizing different technologies. LiDAR is nowadays accepted as a rapid and efficient tool for the implementation of forest inventories [5,6]. This technology allows the ranging of objects by measuring either the time of flight between emitted pulses and their received reflections [5] or by measuring the phase shift between the emitted and the received beam. In addition to the distance (d), the polar angle (θ) and the azimuthal angle (ϕ) of the emitted beam are known, resulting in a set of spherical coordinates (d, θ, ϕ). Cartesian coordinates (x, y, z) can be obtained by a transformation of d, θ and ϕ .

Three different common LiDAR technologies currently exist:

- terrestrial laser scanning (TLS); the scanning system is stationary mounted on a tripod
- airborne laser scanning (ALS); the scanning system is mounted under an aircraft
- vehicle-based laser scanning (VLS); the scanning system is mounted on a ground vehicle

While ALS and VLS can both cover large areas, the resulting point cloud density is much lower than that captured with TLS. High resolution point clouds covering tree surfaces are promising input data for algorithms that fit geometrical objects, such as cylinders, into the point cloud data.

Methods that can derive tree volume and branching structure non-destructively can also potentially provide accurate 3D structural data. This has a range of uses in understanding tree growth and form, scaling theories, competition and resource quality.

In this work, we present an algorithm capable of fitting more than 10,000 connected cylinders into a single tree (*Prunus avium*, DBH \sim 18–20 cm, $h \sim$ 10–12 m), giving a level of detail far exceeding traditional, manual volume measurement-procedures.

1.1. Related Work

Computational forestry using TLS point clouds as input data has become more relevant in the last decade. Early approaches for the estimation of forestry parameters from TLS point clouds have been carried out, such as by Simonse *et al.* [7]. The authors first determined a digital terrain model (DTM) by selecting the points with the lowest z-coordinates and extracting, according to the DTM, a layer corresponding to DBH. In this layer, Simonse *et al.* detected circular structures, *i.e.*, stem sections, with the application of a Hough transformation. A ground truth comparison was applied both to estimated tree positions, with a difference of less than 20 cm for most trees, and to DBH, resulting in a standard deviation of 2.8 cm.

Thies *et al.* [8] used registered TLS data derived from three scan positions per tree to model the stem of single trees, namely European beech (*Fagus sylvatica*) and wild cherry (*P. avium*), by fitting cylinders into the point cloud. The non-linear least squares fitting method (NLS method) for estimating the cylinder parameters minimized the square sum of residuals to the cylinder surface. A beech stem was modeled with 75 overlapping cylinders with a length of 0.5 m each, while a cherry stem was modeled with 59 cylinders in total. Both stems could be modeled to the height of the first fork. A root mean square error (RMSE) of 1.7 cm of the fitted cylinders was calculated and also a ground truth comparison to DBH measurements was applied, showing a deviation of -1.3 cm and 0.6 cm for both trees, respectively.

These publications represent two different approaches in retrieving forestry parameters from TLS data applied during the last decade. One approach is to focus on single individuals to gain detailed

information, the other approach is to determine less accurate parameters, often just DBH and height, but covering a larger area of interest, e.g., hectare-sized plots.

At the plot level, various parameter estimation methods have previously been reported [9–15].

At tree level, Pfeifer *et al.* [16] developed a method to fit cylinders into multiple scan mode point clouds. Single trees were extracted and cylinders fitted by five parameter determination using the non-linear least squares method. As proposed by Thies *et al.* [8], cylinders have been shifted backward and forward to get an approximate position of the next cylinder. Stem and branches were partially detected. The assessment of quality was performed through visual inspection and by the computation of an RMSE of 1.8 cm between the cylinders and their allocated points.

Buksch *et al.* [17–19] developed a method to determine a tree skeleton using point cloud data. The authors generated an octree [20] containing the TLS points. By using the neighborhood information of the octree cells, a graph was extracted, and the contained cycles in the graph were deleted. The resulting graph was found to represent the tree skeleton. The evaluation of the goodness of fit was carried out by calculating and depicting distances between the surface points and the predicted skeleton.

Xu *et al.* [21] applied the Dijkstra algorithm [22] for tree skeletonization. Here, every point was linked with points in a neighborhood of 0.2 m, resulting in a connected graph. Along this graph for every node, the shortest path from a preselected root point was calculated with the Dijkstra algorithm. The lengths of these paths were quantized and clustered into bins. Then, a skeleton was formed by connecting centroids of adjacent bins. This method was further applied by Livny *et al.* [23]. Both papers focused on the visualization of trees rather than on the estimation of forestry parameters.

Yan *et al.* [24] also focused on a computer visualization related calculation of a cylinder tree model. First, a kD tree [25] was constructed, then the point cloud was segmented into clusters using the flooding algorithm [26]. All clusters represented a cylindrical subdivision of the point cloud; a cylinder was fitted afterwards. If no cylinder representing the cluster could be found, the cluster was further subdivided into a fourth step. No evaluation of the goodness of fit was applied, although a visual inspection showed a gap-less representation of the input point cloud.

Bayer *et al.* [27] used a manual skeletonization method on extracted point clouds of single trees under leaf-on conditions. The tree skeleton served as a basis for further automatic computation of branch angles, branch length and branch bending. The cosine of the branch angle was determined as the scalar product of the z -axis and a vector between a point on the branch and the branch base. The length was determined as the distance between the branch base and the branch tip. Furthermore, by using α -shapes [28] on points allocated to a branch section, the authors could calculate the space requirement of the branches by summing the volumes of all tetrahedrons being part of the α -shape. A comparison with ground truth data was not applied.

Belton *et al.* [29] applied a volume calculation for trees under the leaf-on condition in three steps. After extracting the tree from the raw point cloud, they first applied a principal component analysis to each point's neighborhood. The features described by the eigenvalues were used as input for clustering through a Gaussian mixture model [30]. As a result, points representing leaves could be deleted from the input point cloud, leaving stem and branch points only. By fitting ellipses into horizontal slices of the remaining point cloud and by connecting the centers of overlapping ellipses, a skeleton was extracted. A cylinder fitting routine was applied on allocated points to a skeleton section. The authors provided

visual inspection of the cylinder model, including gaps, and compared the volume results to an allometric function. The function estimated 34 m³, while the sum of cylinders was determined to be 74 m³.

Dassot *et al.* [31] employed multiple scan modes for 42 preselected trees consisting of three or four scan positions each. With Polyworks software, the authors selected woody axes as polylines and, afterwards, formed 25-cm segments of the polyline. In an automatic step, a cylinder fitting procedure was applied. Cylinders that were visually incoherent were manually removed, and lastly, missing woody segments were linearly interpolated. Although the cylinder model (only cylinders larger than 7 cm in diameter were taken into account) had some gaps, destructive ground truth measurements showed a good correlation to the TLS volume results derived from the semi-automatic method. Relative differences between the TLS estimates and manual measurements were in a range of 10% for the stems and in a range of 30% for the branches.

Raumonen *et al.* [32] subdivided tree point clouds into small subsets, each subset representing the surface of a tree component. After performing a principal component analysis, the authors were able to extract the skeleton line of the underlying branch segment. Neighboring subsets were merged to segment the point cloud into components, which could be modeled by cylinder fitting. Gaps in the resulting cylinder model had been closed by adding cylinders between neighboring cylinders. The validation of the proposed method was performed by running tests with an artificial point cloud, derived from a modeled tree with known volume. According to the volume, the stem could be modeled completely. Dependent on the different thresholds used, between $\frac{2}{3}$ and $\frac{9}{10}$ of the branch volume was covered. In addition, a comparison between modeled and manually measured diameters was performed.

Jayaratna [33] compared two methods for modeling TLS-scanned trees using cylinders. The first method applied used the RANSAC algorithm proposed by Schnabel *et al.* [34]. This methodology detected many cylinders incorrectly, possibly due to the irregular shape of the stems and branches that were scanned. Furthermore, Jayaratna developed a method that cut sphere surfaces with the input point cloud. After cutting, patterns could be detected in the output, such as circular structures. Every center point of the detected pattern was used as a center point for a new sphere, resulting in a recursive method covering both stem and branches without gaps. Although the author did not apply validation with ground truth data, visual results promised an accurate algorithm for modeling trees.

Additionally to the given examples, a comprehensive overview of more than 100 articles concerning LiDAR-based computational forestry is given in [5]. Recently, Dassot *et al.* [35] gave another overview, including supplementary information about various TLS-functionalities, TLS-devices and existing software solutions.

2. Rationale

This paper presents the development of a fully automated method for fitting cylinders into a manually pre-processed point cloud obtained using TLS tree data under leaf-off conditions, in contrast to the method proposed by Dassot *et al.* [31] requiring manual user interaction. The objective of the methodology is the derivation of gapless tree models, including the remaining branching structure, stored in a hierarchical tree-like data structure. In contrast to other approaches focusing on stem parameters [8,16], this method allows for the extraction of various tree parameters, such as tree diameter,

wood volume and crown architecture. The accuracy of the model must be validated by both ground truth measurements and by a quantification of the the goodness of fit of the model to the input data. This has not been performed at all or only to a lesser extent by other works [17–19,21,23,24,27,29,33] or has been performed in works showing improvable results [16,31]. Since the NLS-based method [8] provides a high accuracy, but a low level of robustness, we also implemented a second method that is less accurate, but more robust. By comparing the results of two methods, cylinders fitted incorrectly by the NLS method are detected and replaced automatically with cylinders fitted using the robust method. Here, we focus on the reconstruction of individual trees that have already been manually extracted from merged point clouds, rather than on the process of extracting these trees initially. A specific aim of our work was to achieve fast computation times for reconstruction and to be able to process point clouds of up to 20 million points efficiently. TLS-based reconstruction methods vary in accuracy when using differing scan properties (range, point density, number of scans, *etc.*), as well as with different tree species and sizes [32]. We discuss the sensitivity of our method to these issues and suggest that reconstruction accuracy for new scan properties and species can be improved by running multiple reconstructions with varying reconstruction parameters (refer to Appendix Figure A1).

3. Materials and Methods

3.1. Terrestrial Laser Scanner

The TLS measurements were performed with the phase shift scanner, Z+F IMAGER 5010 [36]; for technical data, refer to Table 1.

Table 1. Technical data of the Z+F IMAGER 5010.

Parameter	Value
Beam divergence	<0.3 mrad (fullangle)
Beam diameter (at a 0.1-m distance)	~3.5 mm
Range	187.3 m
Resolution range	0.1 mm
Vertical field of view	320°
Horizontal field of view	360°
Vertical resolution	0.0004°
Horizontal resolution	0.0002°
Vertical accuracy	0.007°rms
Horizontal accuracy	0.007°rms
Scan mode “high”	10,000 pixel/360°
Scan mode “superhigh”	20,000 pixel/360°
Scan mode “ultrahigh”	40,000 pixel/360°

3.2. Description of the Scanning Campaign

Scans were made in November, 2012, in Southern Germany within an agroforestry system established on abandoned arable land in 1997. The plot is situated in the upper Rhine valley, west of the Kaiserstuhl and northeast of the city of Breisach on the former alluvial fan of the Rhine (48°4'24" N; 7°35'26" E), 182 m above sea level. The mean annual precipitation sum amounts to 705 mm with a mean annual air temperature of 10.1 °C, ranging between a monthly average minimum air temperature of 0.2 °C and a maximum of 19.6 °C [37]. Soils are a derivative of windblown Rhine valley sediments, such as periglacial loess. These have a high water permeability and low water storage capacity. Consequently, the site tends to be dry, which presents a potential to limit tree growth in hot dry summers [38].

Three different groups of trees have been scanned, and an overview of sample and scan parameters is shown in Table 2. For software development purposes, 14 leafless branches (Group I) of between 2 m and 3 m in length from all tree species planted on the site were collected and scanned inside a building under controlled conditions with four scan positions each. Two solitary trees (Group II) were selected and scanned with a total of six scans to minimize occlusion. Following scanning operations, both trees were felled below a height of 0.05 m to provide a ground truth comparison of manual measurement methods with TLS estimations. Twenty four trees (Group III) were sourced from the same site. Natural regeneration between and within the rows was controlled to maintain a plot suitable for unhindered scanning; scan positions in relation to the plot layout are depicted in Figure 1.

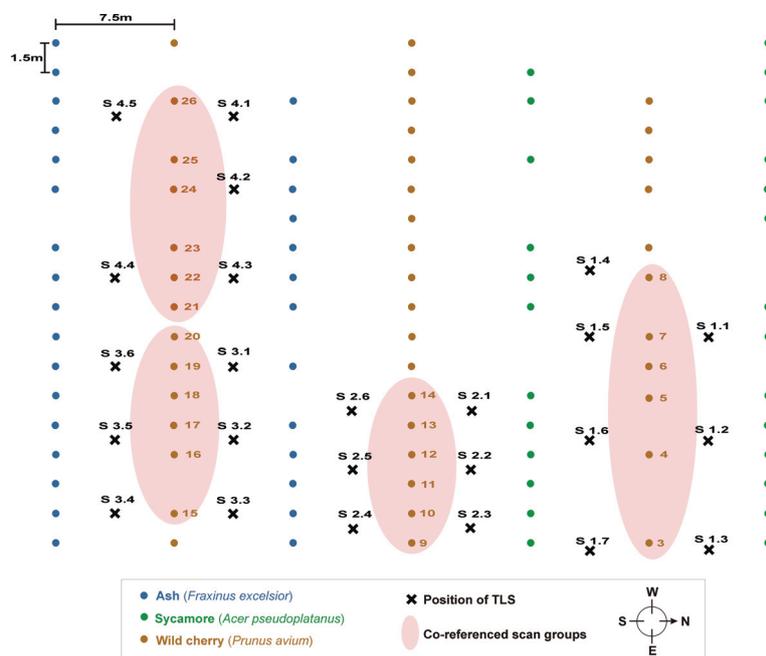
3.3. Artificial Point Cloud

An artificial point cloud was also used to test the reconstruction method, based on TLS data simulated from a 3D model of a Scots pine (*Pinus sylvestris*) [39]. The point cloud is that used by Raunonen *et al.* [32] in testing their method and is described in detail there. The major advantage of using this approach is that the tree size and volume is known exactly. In this case, the reconstruction process can be assessed independently of any possible measurement errors and, so, is an ideal test of the reconstruction. Briefly, the 3D model tree was generated from an empirical growth model, and TLS points were simulated from three locations, using the 3D Monte Carlo ray tracing model library [40]. This resulted in a point cloud of ~3.5 million points, which is used to reconstruct the known branch and trunk volumes.

Table 2. Overview of the applied sample and scan parameters used for data collection..

	Group I	Group II	Group III
Species	<i>Prunus avium</i> <i>Acer pseudoplatanus</i> <i>Quercus robur</i> <i>Fraxinus excelsior</i>	<i>Prunus avium</i>	<i>Prunus avium</i>
Number of trees/branches	14 branches	2 trees	24 trees
Spacing, within and between rows	—	7.5 m × 15 m	1.5 m × 7.5 m
Scan mode	superhigh	ultrahigh	high
Average scans per tree/branch	4	3	1

Figure 1. Group III: map of tree rows in the agroforestry system with marked scan positions. TLS, terrestrial laser scanning.



3.4. Software

The preprocessing of scan data was performed with the Z+F LaserControl software [41]. Multiple scans were co-registered with the help of calibration targets into one common coordinate system, and a mixed pixel filter was applied to delete scan points that provided multiple returns. In the final step, the individual trees were extracted manually from the surrounding point cloud, and then, these points were exported in an ASCII file format.

All algorithms were implemented with the Java Standard Edition Development Kit Version 7 [42]. The external libraries imported into the software project were the Java3d API [43] for visualization purposes and the linear algebra package, Jama [44], used for eigenvalue decomposition and least squares equation solution. Lastly, the CyberVRML97 package [45] was imported for the creation of virtual reality modeling language (VRML) objects from the cylinder models.

MeshLab [46] was used to convert the VRML objects to ply file format, which can be imported into many CAD systems and also into CloudCompare [47], used to render visual output.

3.5. Computer Specification

All computations were run on a machine with an Intel(r) Core(TM) i7-2600K CPU @ 3.40 GHz processor (Intel GmbH Braunschweig, Braunschweig, Germany) with 16 GB RAM (Corsair Components, Inc., Fremont, CA, USA) on a Windows 7 platform (Microsoft Corporation, Redmond, USA). As Java is platform independent, the developed software is expected to run on other operating systems. It must be noted that the processor’s full computation power was never used, but memory usage exceeded 8 GB while processing clouds scanned in “ultra-high” resolution.

3.6. Ground Truth Measurements

Group I samples were measured in detail. At 5-cm intervals, two caliper measurements were performed in perpendicular planes, resulting in four diameters per section. The averaged diameter measurement calculated from the four measurements plus the length of the segment were used as input variables within the volume formula for the creation of circular cylinders. The total sample volume was calculated by the summation of all segment volumes.

The parameters of Group II trees have been measured using standard techniques. Pre-felling DBH and the north direction were noted, in addition to the crown projection area. All branches were measured *in situ* for the cardinal direction and the perpendicular diameter of each branch at the branch collar; individual branch height from the ground was also recorded. The post-felling total tree length was measured. Additionally, the branches were weighed.

For each tree within Group III, DBH was measured.

4. Method Description

In this section, an overview of the utilized method is given followed by a detailed description of individual steps. After the preprocessing steps described in Section 3.4, the input data consists of:

- filtered scan points representing the surface of the target tree;
- noise points, which mainly represent ground artifacts and branches of other trees;
- height above the ground of the lowest z -coordinate, manually measured with Z+F LaserControl [41];
- optional point coordinates of two points, p_{r1} and p_{r2} , visually identified as non-ambiguous in natural environment and scan data, and the bearing between p_{r1} and p_{r2} , measured with a compass.

4.1. De-Noising

Data points are clustered spatially (refer to Appendix A), and all clusters are visualized in different colors. All clusters containing less than a user-defined number of points are deleted. The threshold selection is implemented with a slider; moving the slider changes the visibility of all affected clusters in real time. After accepting a threshold value, which itself does not remove points on the target tree, the deletion of points is executed. Through cursor selection, remaining undesired clusters are removed in the next step. False deletions can be undone before confirmation.

If noise points are spatially connected to target trees points, an occurrence that can prevent the separation of these points by clustering, artificial gaps can be introduced to the data by deleting points that are contained in a user-defined box, and the previously described de-noising procedure has to be repeated.

4.2. Referencing

First, a translation along the z -axis is performed, so that the z -coordinate of each point equals the height above ground in the natural environment.

Optionally, a rotation around the z -axis is performed, so that the vector, $(1, 0, 0)^T$, represents the north direction in the data. Let p_{r1} and p_{r2} be two identified points in the input point cloud and α the bearing between p_{r1} and p_{r2} ; Equations (4)–(6) can then be applied:

$$p_{r3} = p_{r1} \begin{pmatrix} \cos \alpha \\ -\sin \alpha \\ 0 \end{pmatrix} \quad (4)$$

$$\beta = \text{atan2}(p_{r1y} - p_{r2y}, p_{r1x} - p_{r2x}) + \text{atan2}(p_{r1y} - p_{r3y}, p_{r1x} - p_{r3x}) \quad (5)$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} := \begin{pmatrix} x \\ y \\ z \end{pmatrix} \begin{pmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

4.3. Cylinder Model Creation

4.3.1. Cylinder Detection with the Usage of a Sphere (Refer to Figures 2(a)–2(c))

The principal idea of cutting sphere surfaces with tree point clouds [33] is utilized. Let s be a sphere with a sphere center, sc , located approximately on the tree skeleton and with a radius, r , greater than the diameter of the cross-sectional area around sc .

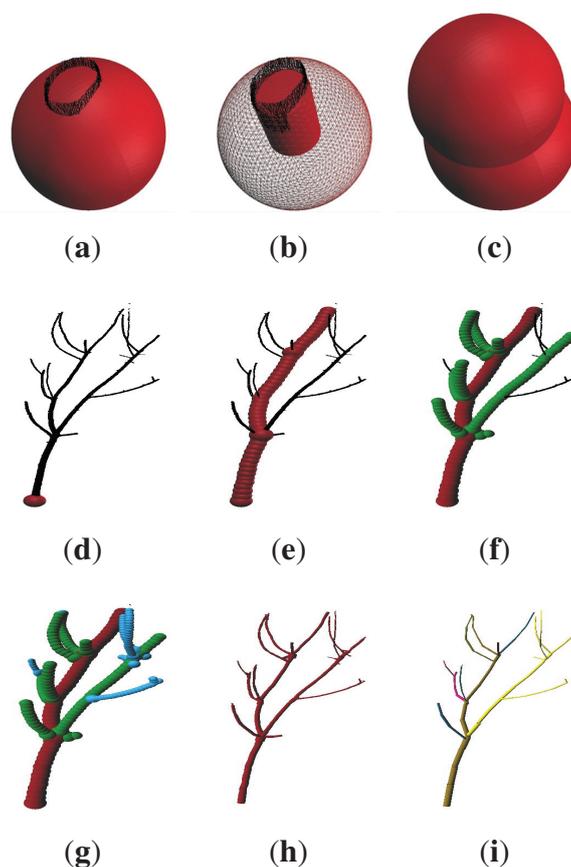
The point set, P , with P being all points included in the ϵ -neighborhood of the sphere surface, represents all cross-sectional areas in a distance $\sim r$ of sc . If s is sufficiently large, even within branch junctions, the cross-sectional areas will be spatially unconnected. Due to such unconnectivity, P can be spatially subdivided into subsets P_i , with P_i , representing the i -th cross-sectional area (refer to Figure 2(a)). As cross-sectional areas of stem and branches can be approximated by circles, into each P_i , a circle, c_i , with radius cr_i and center point cc_i is fitted (refer to Section 4.6). New spheres, s_i , are generated, with cc_i as the center point and $cr_i \times x$ ($x \in \mathbb{R} \wedge x > 1$) as the radius, in accordance with a lower limit for the radius. Additionally, cylinders with start point sc , endpoint cc_i and radius cr_i are stored in a list, with an optional threshold set for the radius (refer to Figure 2(b)). For all spheres, s_i , the described procedure can be repeated (refer to Figure 2(c)).

To prevent the algorithm from jumping back and forth, all points inside s are deleted from the input point cloud after child-spheres s_i are detected.

4.3.2. Initialization and Termination of the Method; the Usage of FIFO-Queues F_1 and F_2 to Follow the Stem and Branches According to Their Order (Refer to Figures 2(d)–2(g))

Since a recursive implementation of this method leads to undesired connections between neighboring branches, two First In - First Out (FIFO)-queues are used to store spheres yet to be processed. The FIFO queue F_1 , is initialized with a sphere located at the stem base. Since the de-noising procedure ensures that no compartments of other trees or ground artifacts are included in the point cloud, a thin slice at the height of the minimum z -coordinate contains only a cross-sectional area of the target tree stem. A circle is fitted into the slice and transformed into a sphere, which is then stored in F_1 (refer to Figure 2(d)).

Figure 2. Visualization of the method in detailed steps; a branch of Group I is used: (a) The cross-sectional area, P_i , derived from the ϵ -neighborhood of the visualized sphere with sphere center sc and radius r ; (b) A circle, c_i , is fitted into P_i ; circle parameters cr_i (radius) and center point cc_i are used as the radius and endpoint of a new cylinder with its start point being the sphere center, sc ; (c) A new sphere, si , is generated with cc_i as the center point and a radius larger than cr_i ; (d) Initialization at the base of the tree/branch; (e) Terminus of branch with order zero (stem for a real tree) is reached (FIFO queue F_2 is transferred to FIFO queue F_1 the first time); (f) All branches of order one have been processed (F_2 is transferred to F_1 the second time); (g) The cylinder following terminates; F_1 and F_2 are both empty; (h) All detected cylinders after termination of the cylinder following; (i) The final cylinder model after post-processing steps; branches of order one are colored differently; the fit quality has been improved.



This sphere is processed as follows. If during processing, more than one child-sphere is detected, the largest child-sphere is always processed first. The $n - 1$ smaller child-spheres are stored in another FIFO-queue, F_2 , until no more cross-sectional areas along the followed branch or stem are detected (*i.e.*, the main branch or stem has been processed to its tip). This procedure is repeated for all spheres stored in F_1 sequentially. When F_1 is empty, all spheres contained in F_2 are transferred to F_1 (refer to Figures 2(e) and 2(f)). Up to five thresholds can be adjusted at this time, namely two clustering parameters (refer to Appendix A), the minimum sphere size, the minimum cylinder radius and the size

of the ϵ -neighborhood of the sphere surface. The reason for the adjustment is an expected change in the order of processed branches by one. Branches of order $n + 1$ have a diameter smaller than their n -th-order predecessors. At the same time, the frequency of branches becomes greater after the order change, leading to a denser neighborhood.

The loop in algorithm 1 terminates, when the currently processed sphere reaches a terminus, and F_1 and F_2 are empty (refer to Figures 2(g) and 2(h)). Termination is assured, as during every generation of child-spheres, points are deleted from the input point cloud, and the number of points is finite.

Algorithm 1: Cylinder model creation.

Data: A list of denoised and referenced TLS points located on a single tree surface

Result: A hierarchical tree-like data structure containing cylinders representing the tree

Create two empty FIFO-queues, F_1 and F_2 , to store spheres;

Create an empty list, L_{Cyl} , to store the cylinders;

Compute a starting sphere on base of the tree and enqueue to F_2 ;

while F_2 not empty **do**

 move all spheres from F_2 to F_1 ;

 adjust thresholds;

while F_1 not empty **do**

 dequeue a sphere s of F_1 ;

while $s \neq null$ **do**

 extract a subset, P , from the input list; P contains all points located in the ϵ neighborhood of s ;

 cluster P into a list, $L_{cluster}$, of clusters P_i ;

if $L_{cluster}$ not empty **then**

 fit a circle, c_i , into each P_i ;

 generate a cylinder for each c_i , using the center point of s , the center point of c_i and the radius of c_i ;

 store the cylinders in L_{Cyl} ;

 generate a sphere, s_i , for each c_i ;

$s =$ largest s_i ;

 all other s_i are enqueued in F_2 ;

else

$s = null$;

end

end

end

end

start from first cylinder to build parent child relation in L_{Cyl} to gain an hierarchical tree-like data structure;

allocate points to each cylinder ;

improve the fit of each cylinder by proposed method;

If the scan quality is high enough, namely scans taken with a very high resolution and with sufficient scan positions preventing occlusion effects, the output will include a list of cylinders covering the complete tree. Otherwise, the traced route of a branch might end in an occlusion gap or the number of points representing a cross-sectional area might be too low to fit a circle, which subsequently will prevent the tracking of the affected branch to its tip (refer to Section 4.5).

4.3.3. Post-Processing Steps (Refer to Figure 2(i))

4.3.3.1. Hierarchical Tree-Like Data Structure

In the resulting list, one constraint remains: the start point of every cylinder (with the exception of the first cylinder in the list), constitutes the end point of the preceding cylinder. This constraint allows for the construction of a simple tree data structure.

The first cylinder in the list is added to the tree structure and set as the root-node. After this, a recursive search seeks cylinders that possess a start-point that is equal to the end-point of the previously added cylinder. Such cylinders are added to the tree, with a link between parent-node and child-node and *vice versa*. After recursion stops, all cylinders have been added to the tree structure.

For some of the following operations, the allocation of cylinders between two neighboring branch junctions to a common segment is needed. The tree structure allows for the efficient detection of these segments. The first segment starts at the root node. Moving up the tree, all nodes, which have one or zero child-nodes, are allocated to the segment, until a node is found with at least two child nodes. This is the last node allocated to the segment. All child-nodes are the starting nodes of new segments.

All segments containing only one cylinder, which is not related to any child-cylinders, are removed from the tree structure, as these are most likely to be incorrectly detected cylinders.

4.3.3.2. Improvement of Branch Junctions

Visualization of the results showed that the first cylinder of each segment is often not in accordance with its surrounding points. While radius and end points are well approximated, the start point can be misallocated. This might be attributed to the fact that in Algorithm 1, the sphere centers are located along the tree skeleton, but are not necessarily located at the points where one skeleton line splits into two or more skeleton lines. A new start point, sp , for these cylinders is computed according to Equation (7), with ep being the end point of the cylinder, l its length and \vec{d} the normalized direction vector of the next cylinder in the segment:

$$sp = ep - l\vec{d} \quad (7)$$

The length of a cylinder is proportional to the diameter of the stem or branch section. For this reason, a merging operation is performed to elongate and reduce the number of cylinders, so more points can be allocated to a single cylinder. Only cylinders belonging to the same segment are merged. Up to three cylinders are merged into one cylinder; the radius of the merged cylinder is calculated by the computation of the original cylinders' median radius.

4.3.3.3. Improvement of Fit Quality

In the next step, we allocate TLS point data with the updated cylinders. Cylinder radii are temporarily enlarged, and points contained in the enlarged cylinders are allocated to the cylinder. Allocated points and the non-enlarged cylinder are used as input data for the improvement of the cylinder fit (refer to Section 4.7).

4.3.3.4. Extraction of Tree Parameters

Post-processing procedures are performed, with an aim to enrich the amount of information contained within the tree model. First, stem cylinders are determined with the help of the tree structure. For all leaf nodes in the structure, the length of the path to the root node is calculated. Cylinders along the longest path are considered stem cylinders. By a visualization check, a failure of this automation can be detected, and the leaf cylinder can be selected manually.

All child nodes of stem cylinders, which do not contain other stem cylinders, act as start nodes for branches. All branches belonging to one cylinder can be extracted from a subtree with the start node acting as the root node.

For the determination of the crown base, a search along the stem starting at the base of the tree is performed. The search terminates when reaching the first branch with predefined minimum thresholds for the diameter at branch collar and length. Since this predicate varies with tree age and species, generic thresholds cannot be found. Additionally, other attributes of the crown base might be desired (e.g., the first whorl). By visual inspection, a failure of this automation can be detected, and the user can determine the branch base manually.

All cylinders contained in the subtree with the crown base cylinder as a root node are determined as part of the crown. Cylinder endpoints are extracted to be used as input data for the determination of a convex hull for crown modeling. This input data has two advantages over using the TLS point cloud for convex hull determination. First, the data size is reduced, and secondly, the data contains neither noise points nor points below the crown base on the stem. The “gift-wrapping” algorithm proposed by Chan *et al.* [48] was implemented for 3D convex hull determination. Despite the fact that faster algorithms exist, this algorithm was employed due to its simplicity. With input sizes of $\sim 10,000$ input points, the algorithm, which assures a runtime complexity of $\mathcal{O}(n\#h)$ ($\#h$ being the number of hull faces), terminated in less than a second.

The input points of the “gift-wrapping” algorithm were projected to the x, y -plane, and a two-dimensional incremental convex hull algorithm [49] was used to compute the crown projection area of the target tree.

The final tree model (refer to Figure 2(h)) allows the extraction of the parameters shown in Table 3.

4.4. Implemented Search Structure

All implementations of the method can be run by so-called brute force searches in theory. With an input data size of up to 20 million points, a runtime complexity of $\mathcal{O}(n^2)$ or more, though, is unpractical. Often, search-operations in 3D data either rely on the usage of either kD-trees [25] or octrees [20].

The implemented search structure needs to provide two time-efficient operations: the return of all points contained inside a sphere (often referred to as a range search) and the removal of all points contained inside a sphere. Although binary trees, such as the kD-tree, guarantee basic operations, such as finding the nearest neighbor or removing a point, in $\mathcal{O}(\log(n))$ complexity, the complexity for a range search in three-dimensional data is $\mathcal{O}(n^{2/3})$ for a balanced tree and $\mathcal{O}(n)$ for the unbalanced version. To assure $\mathcal{O}(n^{2/3})$ at all times, deletion operations require time costly re-balancing. While in an octree worst case, the time complexity for a range search is always $\mathcal{O}(n)$, the expected runtime is sublinear; although, being less efficient regarding runtime complexity the octree provides other advantages over the kD-tree. It can contain other geometric objects than points (*i.e.*, cylinders), and the deletion of objects never requires re-balancing. In an octree, the test for an octree cell intersecting with a sphere is straightforward, enabling the efficient search for points inside a cylinder, as these are approximately represented by their bounding spheres.

Table 3. Output parameters derived from the final tree model.

Parameter	Method
Tree length	Sum of all stem cylinder lengths
Tree height	Maximum z -coordinate of all cylinder end points
Total above ground volume	Sum of all cylinder volumes
Stem volume	Sum of all stem cylinder volumes
Solid volume	Sum of all cylinder volumes with cylinder diameter larger than 7 cm
DBH	Diameter of the cylinder at 1.3 m
Crown space occupation	Volume of the convex hull of crown cylinders' endpoints
Crown projection area	Area of the 2D convex hull of projected crown cylinders' endpoints
Branch azimuth	Angle between north vector and direction vector of a branch segment
Branch length	Length along the longest path in a branch
Branch volume	Sum of all branch cylinders volume
Branch height	Z -coordinate of first branch cylinders start point

4.5. Imperfect Point Clouds

If points have not been used for detecting cylinders in Algorithm 1, these points are still contained in the octree structure, while all used points have been deleted. All points left in the octree are clustered (refer to Appendix A); very small clusters are deleted. For each cluster, the described method is applied once more, resulting in a cylinder list for each cluster.

Let c be the first cylinder in such a list. A search is performed to find a cylinder, c' , in the primary list to connect to the start point of c . Distances d_i between the start point, sp , of c and all cylinder endpoints,

ep_i , in the primary list are determined. Let ep' be the endpoint with the minimum distance; then, the cylinder with endpoint ep' is chosen as c' .

A new cylinder, supposed to fill the volume of the occlusion gap, is generated with ep' as the start point, sp as the endpoint and a radius of c as the radius. This procedure must be performed before the tree structure is built to include newly-detected cylinders.

4.6. Circle Fitting

Three different methods (least squares fit, plane fit and median fit) have been implemented to fit a circle, $c(x_0, y_0, r)$, into a subset, P , of the TLS points, representing a cross-sectional area of a branch or stem.

Circle c can be fitted into a set of $n \geq 3$ points, $p_i(x_i, y_i) \in \mathbb{R}^2$, using least squares [50–52]. After a transition of P into \mathbb{R}^2 , the Gauss–Newton algorithm [53] is applied. For implementation details, refer to Appendix B.2.

The second method was proposed by Jayaranata [33]. A plane, Pl , is fitted into P (refer to Appendix C), and we calculate the intersection, *i.e.*, the desired circle, c , between Pl and the sphere, s , which was used in Algorithm 1. Center point $cc(x_0, y_0)$ of c is the orthogonal projection of the sphere center, sc , onto the plane. With distance d between sc and cc and sr being the sphere radius, we can apply Equation (8) to compute the circle radius, cr :

$$cr = \sqrt{sr^2 - d^2} \quad (8)$$

The third method is straightforward, using the center of mass of P as center point $cc(x_0, y_0)$ and the median distance of all allocated points to cc as radius cr .

4.7. Cylinder Fitting

Into a subset, P , of $n \geq 5$ TLS point clouds, representing a branch or stem section, a cylinder $c(x_0, y_0, z_0, a, b, c, r)$ can be fitted with least squares [8]; here, $p_0(x_0, y_0, z_0)$ being a point on the axis, ax , $\vec{d}_a = (a, b, c)^T$ the direction vector of ax and r the cylinder radius. The implementation details of the applied Gauss–Newton algorithm [53] are given in Appendix B.1.

Cylinder fitting with least squares sometimes results in incorrectly fitted cylinders. The visualization of fitted cylinders showed:

- incorrectly fitted cylinders are often located in components representing very small branches or in regions with low point cover;
- in each iteration of the fit, the diameter of the cylinder grows rapidly.

Whenever the initial cylinder radius is smaller than a certain threshold or when the number of allocated points is too low, another cylinder fitting method is applied. This is also the case, when the least squares fitted cylinder diameter has become much larger than the initial parameter.

The second method (the quantile method) is straightforward. Under the assumption that the cylinder axis is already well fitted, the only parameter left to be improved is the radius. Distance d_i to the cylinder

axis is computed for each allocated point. The cylinder radius is updated according to a quantile of the set of all d_i (refer to Section 5.1 for the quantile adjustment procedure).

4.8. Distance Analysis between TLS Cloud and Cylinder Model

The distance, d_{ij} , between a point, p_i , and a cylinder, c_j , is defined according to distance d_i in Equation (A4) within Appendix B.1, if the projection, p_i^* , of p_i onto the cylinder axis is between start point c_s and end point c_e of the cylinder. Else, we define d_i^* in Equation (9):

$$d_i^* = \min(\text{dist}(p_i^*, c_s), \text{dist}(p_i^*, c_e)) \quad (9)$$

and apply Equation (10):

$$d_{ij} = \sqrt{d_i^{*2} + d_i^2} \quad (10)$$

with d_i the distance defined in Equation (A4). The minimum of all possible distances between p_i and every cylinder in the model is the distance between p_i and the cylinder model. With n being the number of input points and n^* being the number of points with a distance of less than 3 cm to the model, we define the cover of our model in Equation (11) as a measure for the quantity of the completeness of the fit.

$$\text{cover} = \frac{n^*}{n} \quad (11)$$

Furthermore, we apply a normal distribution fitting to the distance histogram of the n^* nearest points to the cylinder model to obtain different measures for the quality of the fit, *i.e.*, the standard deviation, sd , and the mean, \bar{x} , of the original distance data and the standard deviation, σ , and the mean, μ , of the fitted normal distribution.

The usage of 3 cm as a threshold for *cover* acceptance is justified by the large deviation of stem forms of both Group II trees from a circle. Although this rather high value improves the value for the *cover* falsely in a positive way, all four quality fit parameters, *i.e.*, sd , \bar{x} , σ and μ , are effected with a negative, increasing effect. The average standard deviation of 3.82 mm for the Group III models (refer to Figure 3) indicates that 99.73% of the point data, which is accounted as being fit by cylinders, are included in the range of 1.14 cm.

Figure 3. Visualization of eight neighboring Group III tree models.



5. Results and Discussion

We first give an overview on the Group I results, as the algorithm was developed and adjusted utilizing these branches. Afterwards, a deeper analysis of the Group II results is presented, including a comparison to ground truth measurements, as well as an examination between the input point cloud and the resulting cylinder model. An analysis of the artificial point cloud results, where all structure parameters are known *a priori*, is performed. Two allometric models derived from 24 Group III trees will conclude the results.

5.1. Group I Results

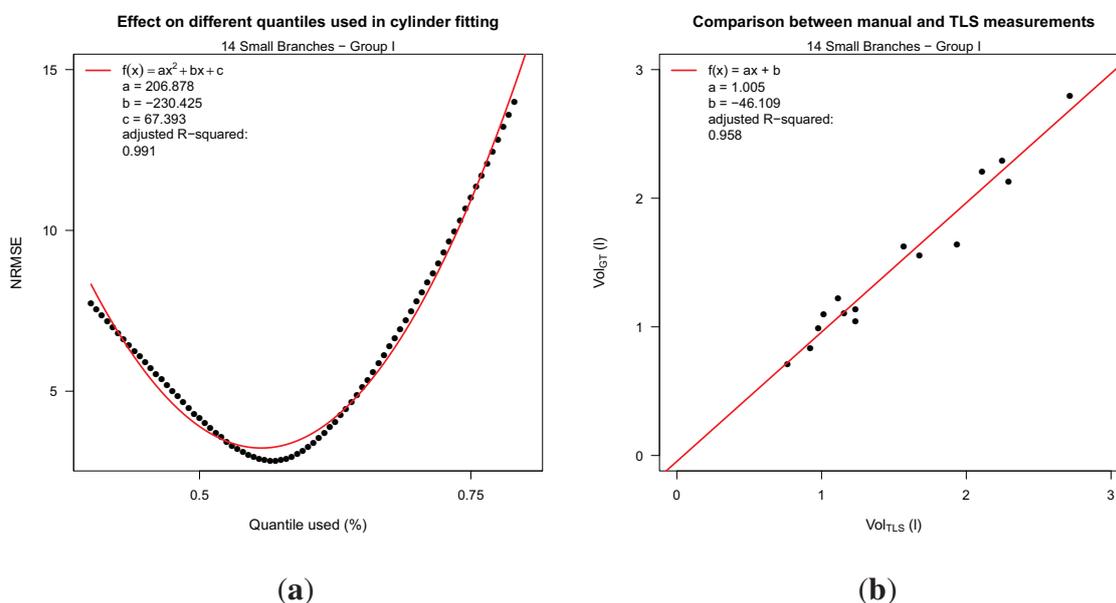
A comparison was carried out between manual ground truth volume measurements VOL_{GT} and TLS results VOL_{TLS} for this group. First, we chose to use the median as the quantile for the cylinder fitting method, as outlined in Section 4.7, which is performed whenever a least squares fit is not applied. The linear model in Equation (12) (adjusted coefficient of determination (r_{adj}^2) = 0.95) points to a slight underestimation of the volume within the TLS results.

$$VOL_{GT} = 0.99VOL_{TLS} + 129 \tag{12}$$

Using 81 quantiles, 81×14 volume calculations (ranging from the 0.4% to the 0.8% quantile) were carried out for all 14 branches, and the normalized root mean square error (NRMSE) for each quantile was computed. A minimum NRMSE of 3.16 (see Figure 4(a)) was achieved by using the 0.56% quantile. The use of this quantile leads to the linear model in Equation (13) ($r_{adj}^2 = 0.96$, as depicted in Figure 4(b)).

$$VOL_{GT} = 1.01VOL_{TLS} - 44 \tag{13}$$

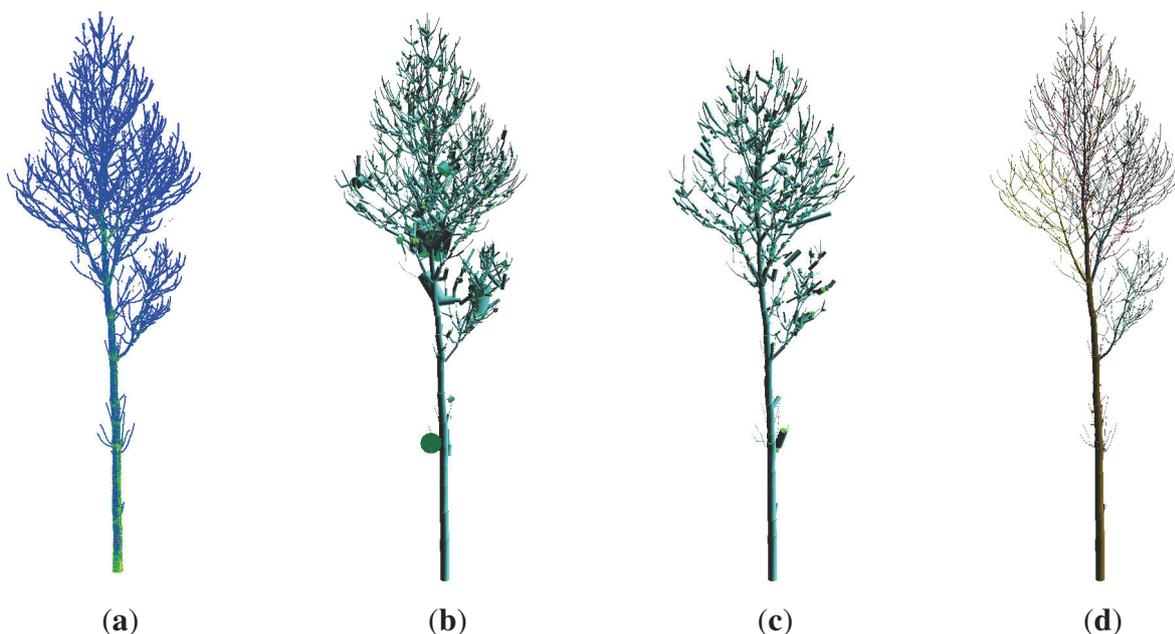
Figure 4. Group I results: (a) normalized root mean square error (NRMSE) in dependency of the quantile used for the quantile cylinder fitting method; (b) Linear regression between TLS results and manual volume measurements.



5.2. Group II Results

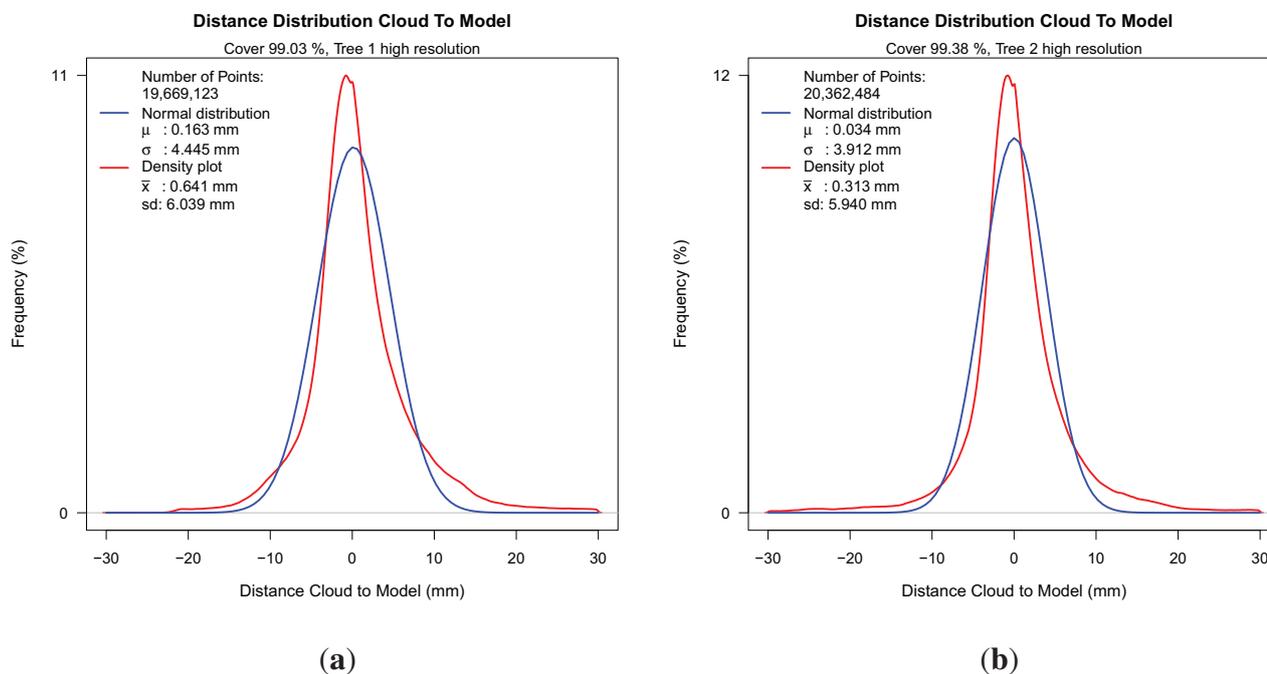
First, a discussion on the impact of the selection of one of the three different circle fitting methods (refer to Section 4.6) is applied. Visual inspection, as depicted in Figure 5(b), indicates that the least squares method alone is capable of detecting nearly every branch within the tree; nevertheless, the method still leads to occasional, obviously too large incorrect cylinders. In Figure 5(c), representing the plane fitting method, the errors are smaller, but the method is not capable of detecting every branch on the tree. The circle fitting method used throughout the results in this paper is the quantile method, which is capable of following every branch without the erroneous results provided by the least squares method. It should be noted that the errors produced in the first (NLS fitting) and the second (plane fitting) method could be minimized with manual threshold adjustment, but never completely eradicated. In public releases of the software, it is intended to include all three circle fitting methods, with an option for the user to choose between them.

Figure 5. Impact on the selection of different circle fitting methods of tree #1 (height ≈ 12 m): (a) the input point cloud of Tree 1; colors represent (unused) intensity values; (b) a model derived by the NLS method with large failure cylinders; (c) a model derived by the plane fitting method with failure cylinders; (d) a model derived by the quantile method covering the complete input point cloud without errors, with successfully detected stem and branches colored differently.



Distance analysis (refer to Section 4.8) derived plots of the histograms and fitted distributions are depicted in Figure 6 for both tree models.

Figure 6. Distance histogram with fitted normal distribution for both Group II point clouds: (a) distance histogram for Tree 1; (b) distance histogram for Tree 2.



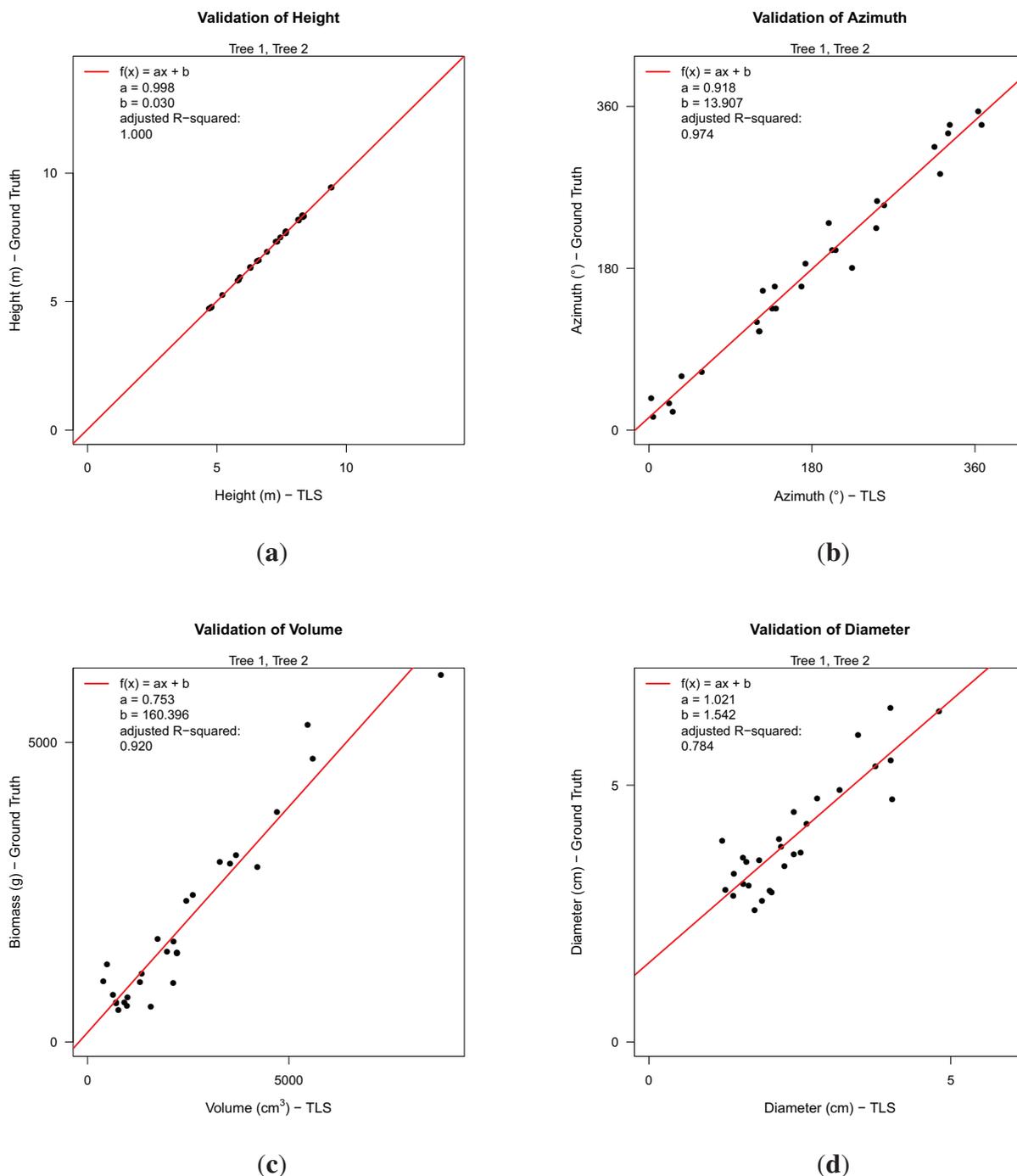
In both density plots, two peaks are visible. We attributed the left peak to the least squares fitted parts of the tree and the smaller right peak to the quantile cylinder fitting method. Resulting values for both trees are given for two resolutions; lower resolutions were produced artificially with Laser Control [41] and noted in Table 4. For the two high resolution results, we achieved a cover of more than 99%. By reducing the point density by a factor of ten, we had in both results a cover loss of approximately 2%. The average fit quality represented by \bar{x} and μ in every case presented sub-millimeter accuracy. The variance of distances, presented by sd and σ , is high compared to the average fit accuracy, this may be attributed to the fact that natural branch segments differ greatly from a mathematical cylinder. Variance also increased with reduced point cloud density. The data preparation with Laser Control was time consuming for the two trees scanned in ultra-high resolution, with total scan file sizes of 36 GB. The software runtime changed from 491 seconds to 41 seconds for Tree 1 by reducing the resolution.

Table 4. The impact of lowering the point cloud density on the goodness of fit.

Tree ID	Number of Points	sd (mm)	\bar{x} (mm)	σ (mm)	μ (mm)	Cover (%)
1	19,669,123	6.04	0.64	4.45	0.16	99.03
1	1,688,509	6.09	0.79	4.47	0.28	97.04
2	20,362,484	5.94	0.31	3.91	0.03	99.38
2	2,618,278	6.10	0.68	3.99	0.22	97.74

Additionally, comparisons to ground truth measurements were applied, which can be seen in Figure 7.

Figure 7. Analysis between TLS-derived data and ground truth data for 28 branches with respect to: (a) branch height; (b) azimuthal direction of branches; (c) TLS-volume against biomass (fresh weight); (d) diameter at branch collar.

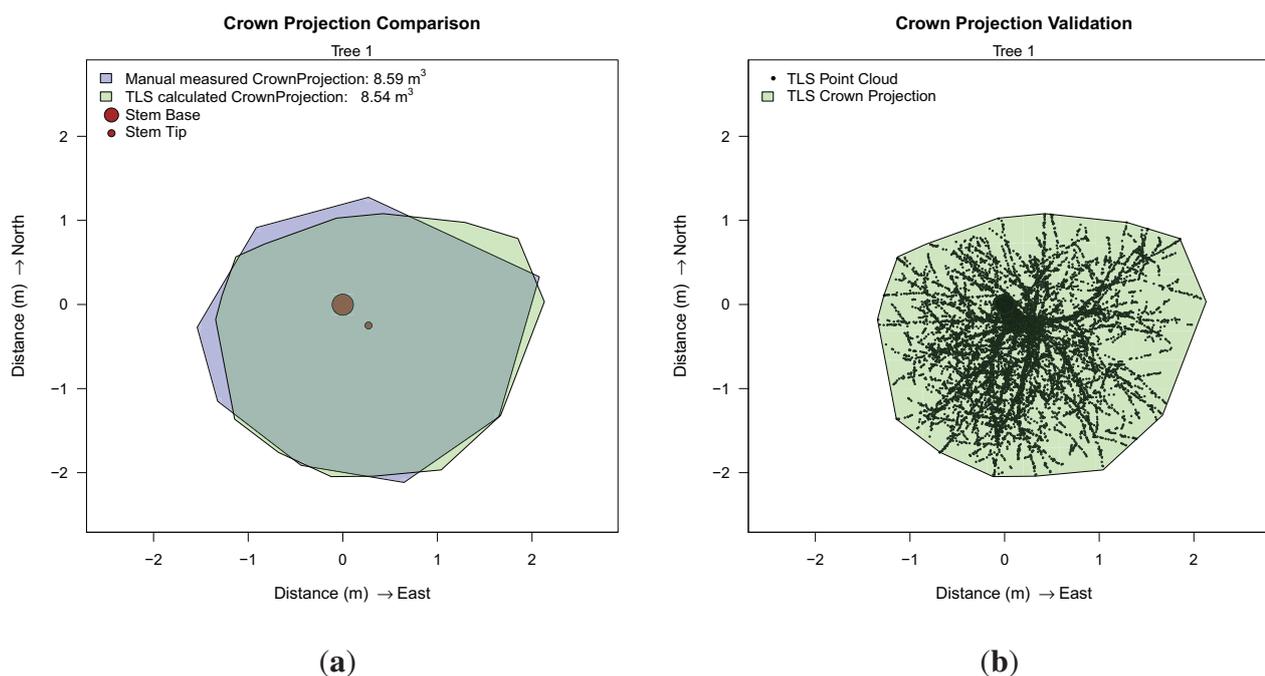


From a total of 29 larger branches, 28 could be successfully detected in our resulting data according to height and azimuth. For these branches, we analyzed the correlation between manually-measured and TLS-derived height, the correlation of both azimuth measurements, the correlation between TLS-derived volume and fresh weight and the correlation of diameters at the branch collar. Branch height showed a r^2_{adj} equaling 1.000, suggesting that both ground truth and TLS data seem to be measured accurately.

For the azimuthal direction, the acquisition of ground truth data was reported to be rather difficult on a standing tree. The coefficient, a , in the linear model ($r_{adj}^2 = 0.974$) with a value of 0.918 might be an indicator of an error in the manual measurements, as visual inspection of TLS results showed no errors regarding the azimuth. As there was no ground truth volume data collected for the branches, it was only possible to model the relation between volume and weight. Density is considered to vary within and between trees [54]; the resulting r_{adj}^2 might be expected to be slightly lower. Linear modeling resulted with a r_{adj}^2 value of 0.92. The largest discrepancy between ground truth data and TLS results was detected in the diameter analysis with a r_{adj}^2 of 0.784. This discrepancy can be partially explained by two different measurement methods. Ground truth diameters have been measured at the outer extremity of the branch collar. In the TLS method, the diameter of the third cylinder of every branch was taken, as cylinders near branch junctions showed larger errors in diameter than others by visual inspection.

Additionally, a comparison between the manual and TLS-derived crown-projection area was performed for both trees, depicted for Tree 1 in Figure 8.

Figure 8. Crown projection analysis for Tree 1: (a) comparison to manual results; (b) in the x,y -plane projected point cloud as visual validation.



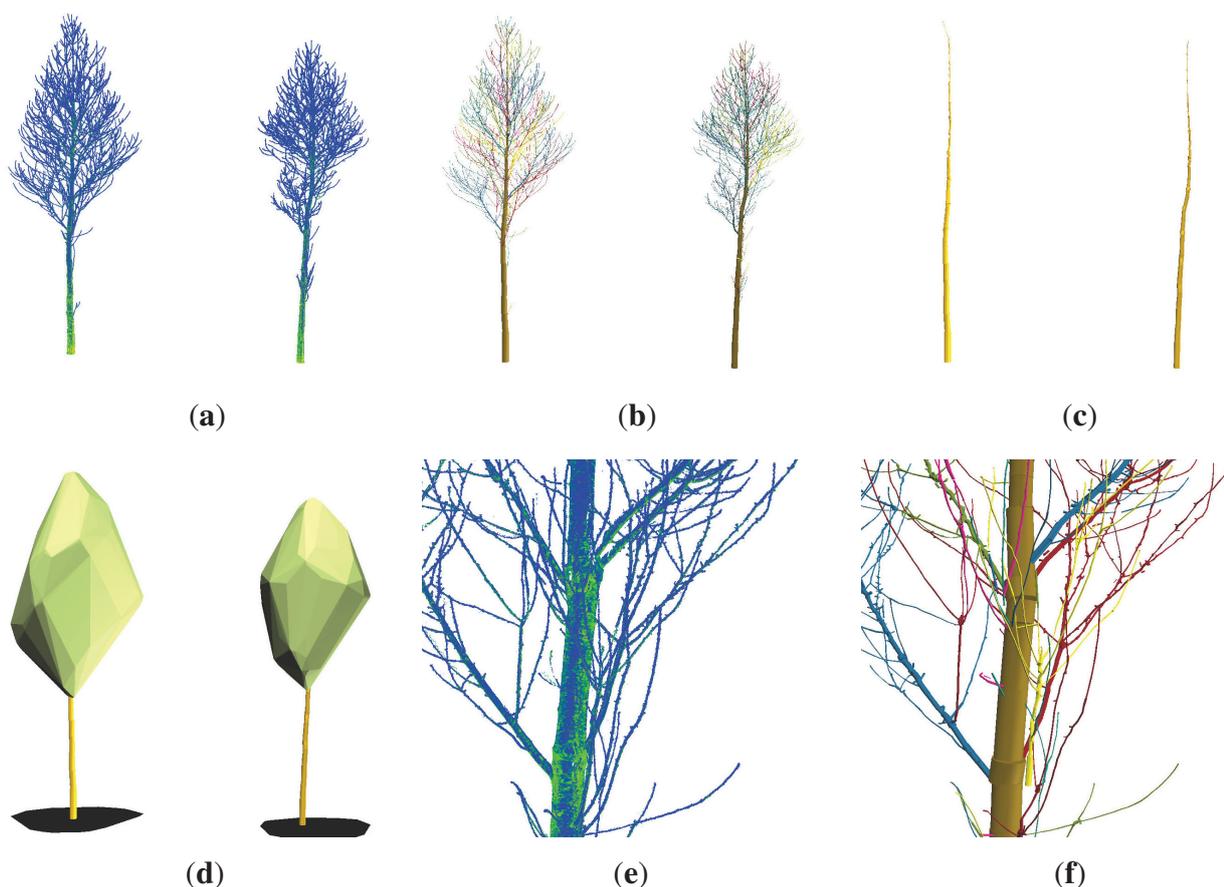
Results of the crown projection area, DBH and length measurements are included in Table 5, showing high accuracy in length measurements. DBH and crown projection area correlate well for Tree 1 with a larger deviation for Tree 2. Visualization of TLS derived results did not reveal an explanation for the deviation, but this might be dictated by the fact that the TLS-derived diameter is taken from a cylinder with a length larger than 20 cm, which still covers the DBH height, but also covers sections below and above 1.3 m. An error in ground truth data or an error caused by the deviation of the stem form from a circular cylinder can also not be suspended.

Table 5. Comparison of manual ground truth data with the TLS-derived results.

Tree ID	Method	DBH (cm)	Length (m)	Crown projection area (m ²)
1	manual	18.6	11.52	8.59
1	TLS	18.1	11.57	8.54
2	manual	20.0	12.60	10.70
2	TLS	21.4	12.62	12.00

Visualization of the cylinder models of both trees are depicted in Figure 9.

Figure 9. Visualization of both Group II trees (height \approx 12 m): (a) TLS point clouds; (b) cylinder models with branches colored differently; (c) extracted stems of the cylinder models; (d) model of the crowns, stems and crown projections; (e) enlarged view into the lower crown point cloud of 2; (f) enlarged view into the lower crown cylinder model of 2.



5.3. Artificial Point Cloud Results

Thresholds used for Groups II and III could not be applied to the artificial point cloud. It took about 20 software runs to adjust the thresholds to a point where satisfactory results were achieved (refer to Figure A1). Comparisons between the values of the original cylinder model and that derived by our software are included in Table 6. An underestimation of the crown projection area by our method

indicates that not every branch was followed to its tip. With an overestimation of 2.5% of total volume, either some incorrect cylinders were included in our model or the diameter of cylinders was slightly overestimated. Visualization showed for this point cloud that, due to the very dense neighborhood of small branches, sometimes artificial connections between two branches are included in our model. Distance analysis with values given in Table 7 indicates a good cover of our model: missing only 0.5% of the point cloud. This suggests that the algorithm is suitable for a wider range of tree forms than just *P. avium* [55]. Better variance parameters than those found for the ground truth trees are possible due to the mathematical origin of the point cloud data, while \bar{x} results in poorer values than for the same parameter for the Group II trees.

Table 6. Comparison between original data and the proposed method results for the artificial point cloud.

Data Set	DBH (cm)	height (m)	Volume (L)	Crown Projection Area (m ²)
original	22.4	17.73	666.5	30.81
program results	22.4	17.74	683.6	29.70

Table 7. Distance analysis between the point cloud and the derived model for the artificial tree.

Tree ID	Number of Points	<i>sd</i> (mm)	\bar{x} (mm)	σ (mm)	μ (mm)	Cover (%)
artificial	3,442,699	5.00	2.134	2.13	0.07	99.48

Visualization of the results of the artificial point cloud (Figure 10(g)) is depicted in Figure 10(h),10(i), showing a histogram of the volume distribution according to the diameter of the cylinders. While most diameter classes are well represented in this histogram, the 180–200 mm class is largely underestimated. We suppose, as this class is located in part of the stem, where many branch junctions are growing, that some cylinders' diameters in the area of branch junctions are overestimated, due to the allocation of branch points to the stem cylinder before a least squares fit is applied. These cylinders are then classified into the next two larger diameter classes, leading to the depicted overestimation of total volume in these classes.

Both our method and the one of Raunonen *et al.* [32] are able to reconstruct height and volume very accurately (*i.e.*, the error is within a few percent) [56].

5.4. Group III Results

Using a loop, the total runtime for all 24 extracted trees was less than 30 min in total. Results for distance analysis, DBH and volume are given in Table 8. This table also includes ground truth DBH measurements, and a comparison of the two different DBH values is performed in Figure 11. The derived linear model shows a r_{adj}^2 value of 0.979.

Table 8. Partial results for 24 Group III trees.

ID	Points	DBH _{TLS(GT)} ¹ (cm)	Vol(l)	sd (mm)	\bar{x} (mm)	σ (mm)	μ (mm)	Cover (%)
3	503,794	10.69 (11.3)	64.24	5.39	1.11	3.61	0.30	81.01
4	1,750,783	16.25 (16.5)	164.30	5.80	1.35	4.51	0.57	94.24
5	1,033,913	8.96 (9.0)	54.40	5.24	1.31	3.45	0.41	97.99
6	1,901,388	14.38 (14.5)	139.96	5.79	1.69	4.34	0.78	86.84
7	1,091,585	14.35 (13.3)	122.11	7.36	1.96	5.77	1.01	78.03
8	722,546	10.41 (10.5)	65.21	6.27	1.77	4.52	0.78	86.16
9	1,289,349	15.08 (15.1)	140.82	5.39	1.19	3.65	0.31	78.82
10	604,092	7.83 (7.9)	34.53	4.90	1.07	2.97	0.23	88.05
11	2,414,504	15.70 (14.9)	159.42	5.93	1.73	3.99	0.67	82.12
12	1,332,867	13.80 (14.3)	131.83	6.41	1.77	4.47	0.73	84.05
13	1,241,908	12.84 (12.9)	109.74	6.04	1.57	4.08	0.48	90.00
14	645,269	10.09 (9.9)	62.17	5.84	1.44	3.64	0.38	88.04
15	1,289,349	10.11 (9.8)	49.28	6.58	1.95	3.78	0.62	71.80
16	523,777	7.36 (8.1)	32.74	5.58	1.44	3.55	0.48	97.38
17	1,743,865	12.18 (12.2)	102.83	5.35	1.47	3.69	0.57	92.16
18	1,316,364	12.71 (12.7)	96.06	5.51	1.64	3.73	0.68	79.26
19	649,161	10.95 (11.4)	58.93	5.38	1.06	3.71	0.31	93.10
20	933,400	13.21 (12.8)	91.17	5.40	1.48	3.63	0.54	88.62
21	266,787	7.30 (7.4)	22.85	5.30	1.39	2.85	0.34	84.19
22	355,830	7.99 (7.8)	27.65	5.27	1.29	2.84	0.26	77.52
23	1,074,614	12.34 (12.5)	80.72	5.92	1.81	3.85	0.72	89.03
24	674,759	11.90 (11.2)	74.51	5.77	1.27	4.24	0.40	96.51
25	373,704	7.39 (7.4)	28.16	5.21	1.36	3.33	0.49	92.30
26	410,720	9.42 (9.8)	40.33	5.61	1.62	3.46	0.50	77.60
mean	1,006,255	11.36 (11.43)	81.41	5.72	1.49	3.82	0.52	86.45

GT = ground-truth.

Although the average cover for the 24 models is only 86.45%, which naturally enlarges the parameter, \bar{x} , parameters of the fitted normal distribution, σ and μ , were comparable to those of Group II models.

We were able to build two allometric models using DBH as a predictor variable, as an example of the more general use of the reconstructed 3D tree information. The number of samples (24) is still too low to use these example allometric models in forestry applications.

Total above ground volume is derived from Equation (1); the volume of woody segments with diameter larger than 7 cm (solid volume) utilized a modified equation, as given in Figure 12(b). Modification of the equation was performed to prevent the prediction of solid volume for trees from being too small to contain solid segments. Models showed values for r^2 equaling 0.965 and 0.968, respectively. A visualization of a group of eight neighboring trees can be seen in Figure 3, depicting overlapping crowns. Low scan resolution together with occlusion effects causes a low density of data points, leading to incorrectly modeled branches located in the upper crowns.

Figure 10. Visualization of the artificial tree results (height \approx 18 m): (g) artificial point cloud; (h) resulting cylinder model with branches colored differently; (i) volume distribution in both the original data and program results.

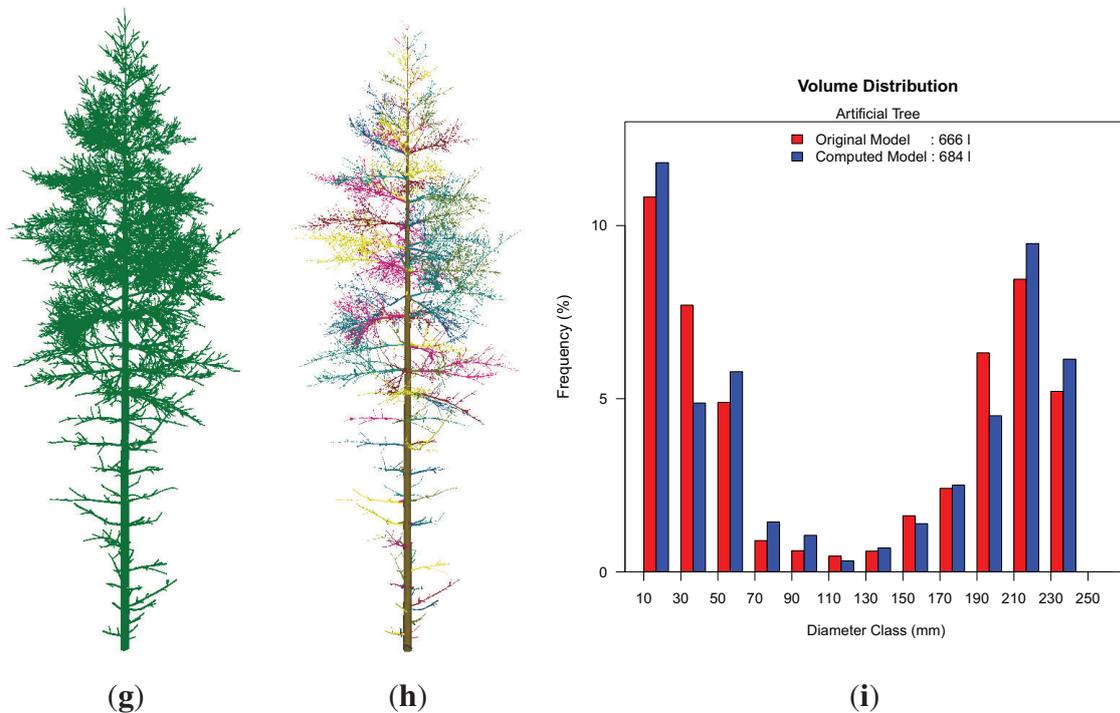


Figure 11. Validation of TLS-derived and manually-measured DBH measurements.

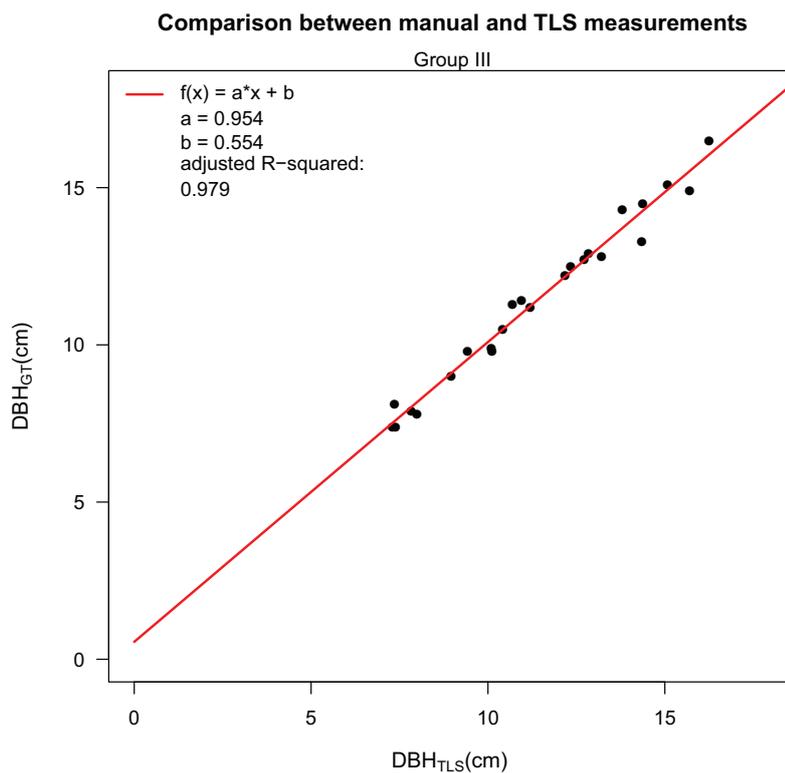
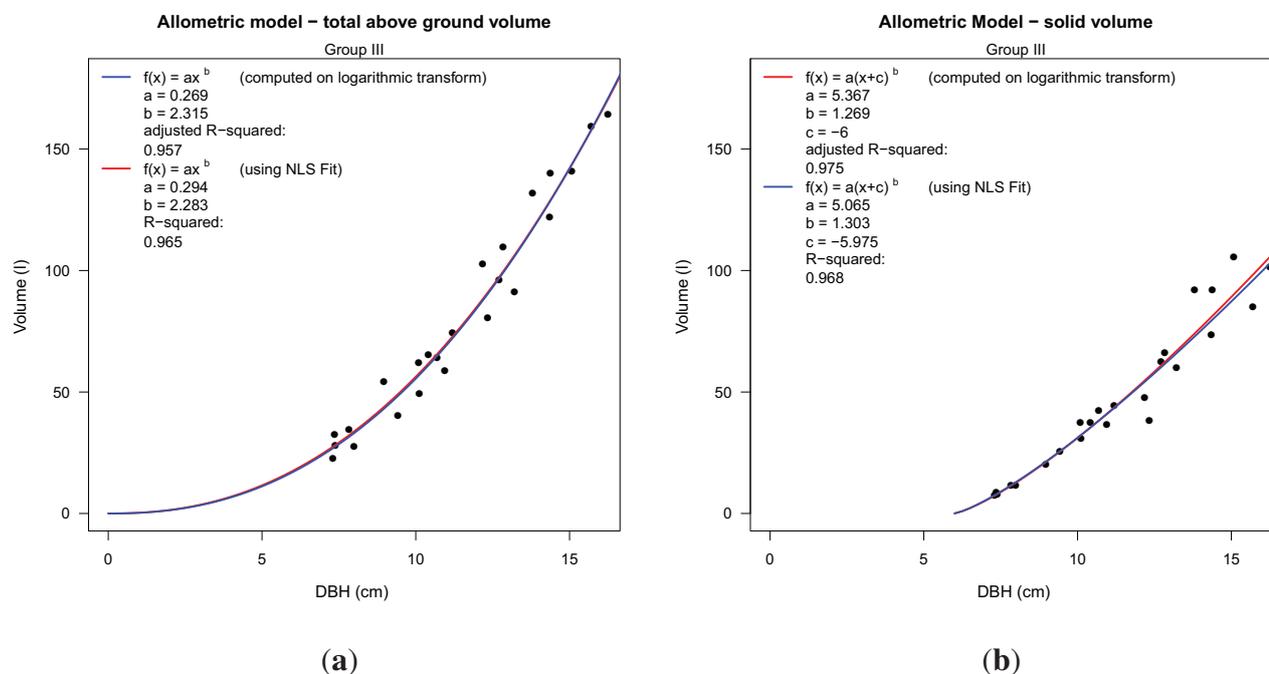


Figure 12. Two allometric models for volume estimation derived from 24 Group III models using TLS-derived DBH as the input variable: (a) total above-ground volume modeled as a function of DBH; (b) solid volume modeled as a function of DBH.



6. Conclusions

The goal of our work was to find a fully automated method capable of building cylinder models of trees providing a high level of detail and accuracy utilizing manually extracted tree point clouds. The presented Group II tree models covered the point cloud almost completely, ensuring that every compartment of the tree is represented. The distance analysis between cloud and model still neglects the detection of incorrectly added cylinders. Such cylinders can be detected efficiently through visual inspection. Inspection showed that the Group II models did not contain such cylinders. The goodness of fit provided sub-millimeter accuracy with the variance equaling ~ 5 mm for all models. This shows a great potential of the proposed method, especially compared to traditional manual methods.

Execution of the method utilizing point cloud data from other tree species seems reasonable [55], as the total volume results of the artificial point cloud showed an overestimation of less than 3%, comparable to the results gained by Raunonen *et al.* [32], in which the authors used the same point cloud. The threshold adjustment, which was needed to gain acceptable results, is a non-automatic process. This proves to be more efficient, when the user has greater insight into the proposed method.

The implementation of our method did not provide an automatic extraction of the tree points from the combined point clouds; tree extraction and pre-processing was performed manually. Existing works, such as that by Simonse *et al.* [7], show the potential that this process can be automated in the future.

Our work did not handle scans taken under windy conditions. Undiscussed results of scans taken under such conditions showed poorer results than those discussed. Manual or semi-automatic methods, such as that proposed by Eysn *et al.* [10], seem to be more promising under these circumstances than

fully automatic methods. The downside of such manual methods is a reduced accuracy with a longer processing time.

The Group II tree scans showed a low degree of occlusion, as scans were taken with a very high resolution, while the trees had no competing neighbors. Group III scans, on the other hand, did not present these advantages; every tree had close neighbors, as depicted in Figure 3. Furthermore, the point cloud for many trees consisted of less than a million points. The attained cover of these models was by far the worst, but the resulting allometric models still showed a very good correlation with a very high r_{adj}^2 , giving the assumption of providing acceptable volume estimation results.

In summary, the method is capable of producing tree models with a very high level of detail and accuracy. Depending on individual circumstances, such as the level of occlusion by neighboring trees, occlusion by the tree itself, the size of the tree, wind conditions and scanning resolution, the implemented algorithm's output can provide a model representing the complete tree with every branch accounted for. Future testing and improvement of the algorithm on point clouds taken from other tree species and larger individuals, both under alternative growing conditions, is anticipated.

7. Future Work

It is envisaged to publish the software under an open source license after improvement of the implementation. For information, refer to [55], where Group I and Group II point clouds, as well as scans of other tree species, are available for development and comparison purposes.

Appendix

A. Clustering

Density-based spatial clustering of applications with noise (DBSCAN) is an efficient way to cluster spatial data. DBSCAN requires two parameters: a maximum distance, ϵ , and a minimum number of points, $minPts$. A point, p , is considered to be a core point of a cluster, if its ϵ -neighbourhood contains at least $minPts$ points. DBSCAN will return a list of clusters, consisting of core points and points whose ϵ -neighbourhood contains another core point, but less than $minPts$ points (density-reachable points). The algorithm is deterministic in calculating the number of clusters and also in the determination of core and noise points. The allocation of density-reachable points to a cluster is non-deterministic. Pseudo code and a detailed description are available [57].

The runtime of DBSCAN depends on the runtime of a region query, which is executed for every point. As we used an octree for this operation, the guaranteed runtime for our implementation is $\mathcal{O}(n^2)$ with an expected runtime of $\mathcal{O}(n \log(n))$.

B. Least Squares Fit for Cylinders and Circles

For both fits, we apply the Gauss–Newton algorithm [53], which linearizes the non-linear problem with an initial solution. By knowing distance d_i from a point, p_i , to the approximated circle or cylinder, one can solve Equation (A1), with \mathcal{J} being the Jacobian matrix.

$$\mathcal{J}p = d \tag{A1}$$

The open source guide [58] was referred to for implementation purposes presented in this section.

B.1. Cylinder Least Squares Fit

When the least squares fit is applied in our method, a preliminary cylinder with a start point, c_s , an end point, c_e , and a radius, r , is already known. The initial parameters of a cylinder, $c(x_0, y_0, z_0, a, b, c, r)$, can be derived from c_s, c_e and r .

These seven parameters to be estimated can be reduced to five (x_0, y_0, a, b, r) by:

- a normalization of d_a , so c can be computed from a and b
- a translation of p_0 to the origin of the coordinate system and a rotation of the input data, so d_a is along the z -axis; therefore, z_0 can be determined from a, b, x_0, y_0 .

After normalizing \vec{d}_a , a translation to the coordination center is applied to the point set and the initial cylinder parameters in Equation (A2):

$$\begin{pmatrix} x^t \\ y^t \\ z^t \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} cs_x \\ cs_y \\ cs_z \end{pmatrix} \tag{A2}$$

Next, a rotation is required, so that the cylinder axis aligns with the z -axis, as applied in Equation (A3):

$$\begin{pmatrix} x^r \\ y^r \\ z^r \end{pmatrix} = \mathbf{U} \begin{pmatrix} x^t \\ y^t \\ z^t \end{pmatrix} \tag{A3}$$

with rotation matrix \mathbf{U} defined as in Equation (A7), replacing \vec{n} with \vec{d}_a .

For the minimization of squared distances $\sum_{i=1}^n d_i^2$, we define distance d_i of a point, p_i^r , to the cylinder in Equation (A4):

$$d_i = r_i - r = \frac{|\vec{d}_a^r \times (p_i^r - p_0^r)|}{|\vec{d}_a^r|} - r \tag{A4}$$

thus changing global Equation (A1) to the least squares system in Equation (A5):

$$\begin{pmatrix} -\frac{x_1}{r_1} & -\frac{y_1}{r_1} & -\frac{x_1 z_1}{r_1} & -\frac{y_1 z_1}{r_1} & -1 \\ -\frac{x_2}{r_2} & -\frac{y_2}{r_2} & -\frac{x_2 z_2}{r_2} & -\frac{y_2 z_2}{r_2} & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\frac{x_n}{r_n} & -\frac{y_n}{r_n} & -\frac{x_n z_n}{r_n} & -\frac{y_n z_n}{r_n} & -1 \end{pmatrix} \begin{pmatrix} p_{x_0} \\ p_{y_0} \\ p_a \\ p_b \\ p_r \end{pmatrix} = \begin{pmatrix} -d_1 \\ -d_2 \\ \vdots \\ -d_n \end{pmatrix} \tag{A5}$$

Parameters must be updated in each iteration according to Equation (A6):

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} := \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \mathbf{U}^{-1} \begin{pmatrix} p_{x_0} \\ p_{y_0} \\ -p_{x_0}p_a - p_{y_0}p_b \end{pmatrix}$$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := \mathbf{U}^{-1} \begin{pmatrix} p_a \\ p_b \\ 1 \end{pmatrix} \tag{A6}$$

$$r := r + p_r$$

Utilization of steps from Equation (A2) to Equation (A6) are repeated, until one of the following circumstances becomes true:

- The standard deviation of the points to the fitted cylinder falls below a threshold of 3 mm.
- A maximum number of 30 iterations is reached.

Thresholds have been estimated by the observation of the standard deviation change for each iteration for various cylinders. Using more than 30 iterations results in longer run times without a noticeable improvement of the fit, while a standard deviation of less than 3 mm can already be considered as a near perfect fit of irregular branch segments, shown by the distance analysis provided in Section 5.2.

The fitted cylinder is infinite in length; however, a cylinder segment with a start and an end point is desired, so we project the initial start and end point to the fitted axis, ax .

B.2. Circle Least Squares Fit

For fitting a circle into a TLS point set, P , representing a cross-sectional cut with the least squares method, P must be translated into \mathbb{R}^2 . As a cross-sectional cut through the point cloud is coplanar, first, the best fit plane procedure (refer to Section C) is applied to calculate plane $Pl(\vec{n}, p)$. A rotation of P is applied, so that \vec{n} is parallel to the z -axis. Rotation-matrix \mathbf{U} is computed by a multiplication of a rotation, \mathbf{U}_1 , around the x -axis to bring \vec{n} into the x, z plane and a second rotation, \mathbf{U}_2 , around the y -axis directing \vec{n} along the z -axis. \mathbf{U} is described in Equation (A7), with α being the angle between \vec{n} and the x, z -plane and β the angle between $\mathbf{U}_1\vec{n}$ and the z -axis.

$$\mathbf{U} = \mathbf{U}_2\mathbf{U}_1 = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \tag{A7}$$

The rotated P^r is now translated into \mathbb{R}^2 by using only x, y -coordinates of $p_i^r \in P^r$. For an initial estimation of x_0, y_0 and r of a circle, $c(x_0, y_0, r)$, we first minimize the distance function given in Equation (A8), with r_i being the distance of p_1 to $p_0(x_0, y_0)$:

$$d_i = r_i^2 - r^2 \tag{A8}$$

leading to linear least squares system in Equation (A9):

$$\begin{pmatrix} 2x_1 & 2y_1 & -1 \\ 2x_2 & 2y_2 & -1 \\ \vdots & \vdots & \vdots \\ 2x_n & 2y_n & -1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ \rho \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \vdots \\ x_n^2 + y_n^2 \end{pmatrix} \tag{A9}$$

with ρ defined in Equation (A10):

$$\rho = x_0^2 + y_0^2 - r^2 \tag{A10}$$

With initial parameters x_0, y_0, r , we can minimize $\sum_{i=1}^n d_i^2$, with d_i being defined in Equation (A11):

$$d_i = \sqrt{((x_i - x_0)^2 + (y_i - y_0)^2)} - r \tag{A11}$$

Global Equation (A1) with defined d_i leads to the linear least squares system in Equation (A12):

$$\begin{pmatrix} \frac{\partial d_1}{\partial x_0} & \frac{\partial d_1}{\partial y_0} & \frac{\partial d_1}{\partial r} \\ \frac{\partial d_2}{\partial x_0} & \frac{\partial d_2}{\partial y_0} & \frac{\partial d_2}{\partial r} \\ \vdots & \vdots & \vdots \\ \frac{\partial d_i}{\partial x_0} & \frac{\partial d_i}{\partial y_0} & \frac{\partial d_i}{\partial r} \end{pmatrix} \begin{pmatrix} p_{x_0} \\ p_{y_0} \\ p_r \end{pmatrix} = \begin{pmatrix} -\frac{x_1-x_0}{r_1} & -\frac{y_1-y_0}{r_1} & -1 \\ -\frac{x_2-x_0}{r_2} & -\frac{y_2-y_0}{r_2} & -1 \\ \vdots & \vdots & \vdots \\ -\frac{x_n-x_0}{r_n} & -\frac{y_n-y_0}{r_n} & -1 \end{pmatrix} \begin{pmatrix} p_{x_0} \\ p_{y_0} \\ p_r \end{pmatrix} = \begin{pmatrix} -d_1 \\ -d_2 \\ \vdots \\ -d_n \end{pmatrix} \tag{A12}$$

which must be solved to update parameters according to Equation (A13):

$$\begin{aligned} x_0 &:= x_0 + p_{x_0} \\ y_0 &:= y_0 + p_{y_0} \\ r &:= r + p_r \end{aligned} \tag{A13}$$

Utilization of the steps in Equations (A12) and (A13) are repeated, until one of the following circumstances becomes true:

- The standard deviation of P^r to the fitted circle falls below a threshold of 3 mm.
- A maximum number of 30 iterations is reached.

C. Plane Fitting Using Principal Component Analysis

Let P be a subset of n points, as shown in Equation (A14):

$$P = \{p_i | p_i \in P \wedge i \in \{1, \dots, n\}\} \tag{A14}$$

with \bar{p} being the subset's center of mass. Then, its covariance matrix (**C**) is formed according to Equation (A15).

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^n (p_i - \bar{p})(p_i - \bar{p})^T = \begin{pmatrix} s_x^2 & s_{xy} & s_{xz} \\ s_{xy} & s_y^2 & s_{yz} \\ s_{xz} & s_{yz} & s_z^2 \end{pmatrix} \tag{A15}$$

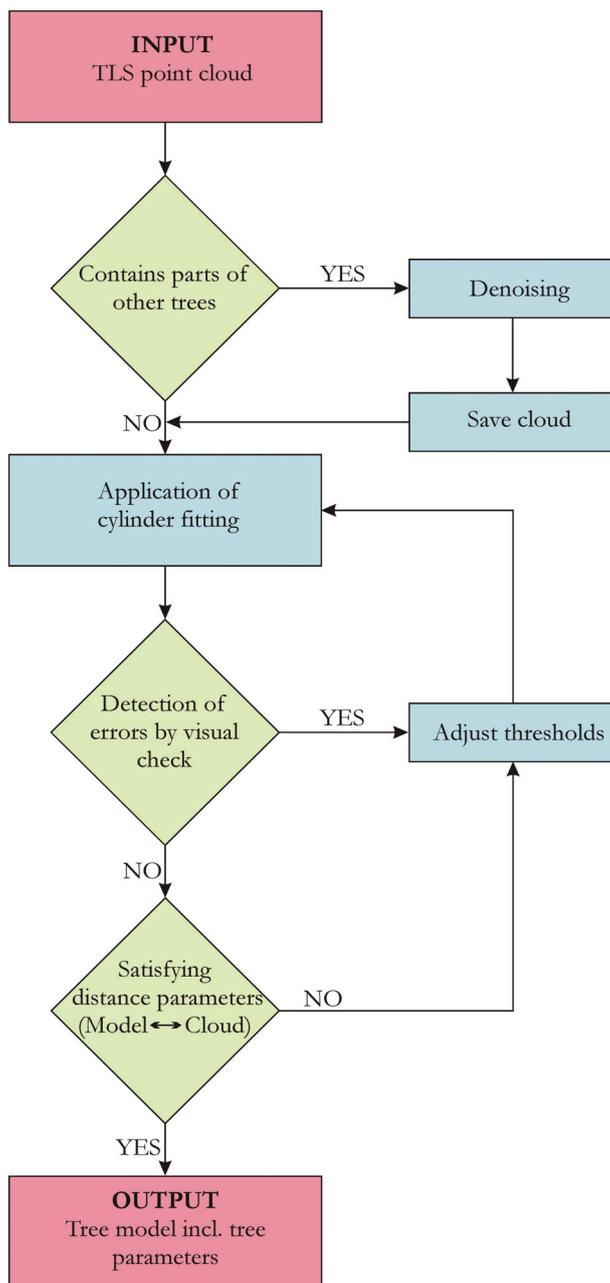
C is a 3×3 , real, positive, semi-definite matrix [59]. **C** can be decomposed into the eigenvectors, $\vec{v}_1, \vec{v}_2, \vec{v}_3$, satisfying the linear Equation (A16), where λ_i are the eigenvalues corresponding to each \vec{v}_i .

$$\mathbf{C}\vec{v}_i = \lambda_i\vec{v}_i \tag{A16}$$

This decomposition is called principal component analysis (PCA). The natural order of the tuple $(\vec{v}_1, \vec{v}_2, \vec{v}_3)$ also satisfies the expression $\lambda_1 > \lambda_2 > \lambda_3$. Then, \vec{v}_1 is defined as the first principal component of P , being the vector oriented along the direction of the greatest variance of P . All principal components are orthogonal to each other. The fitted plane in point-normal form is defined by \bar{p} and \vec{v}_1 .

D. Required Software Interaction

Figure A1. Data processing flowchart.



Acknowledgments

This research was funded by the Federal Ministry of Education and Research (BMBF) within the Lin2Value project (support code 033L049A) and the AGROCOP project (support code 033L051B). The article processing charge was funded by the Albert Ludwigs University of Freiburg in the funding program open access publishing. The authors would like to express their thanks to Zoller + Fröhlich GmbH for the support regarding laser scanning and Raphael Ehnis for preprocessing the utilized point cloud data.

Author Contributions

The contribution of the authors to this work is as stated:

Jan Hackenberg^{1,2,3,5,6,7,8}, Christopher Morhart^{1,2,3,7,8}, Jonathan Sheppard^{1,2,3,7,8}, Heinrich Spiecker^{1,8,9} and Mathias Disney^{4,8}.

1, Designing of the experiment; 2, data collection; 3, data processing; 4, development of artificial point cloud; 5, development of algorithm; 6, statistical analysis; 7, paper preparation; 8, review; 9, project coordination.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Picard, N.; Saint-André, L.; Henry, M. *Manual for Building Tree Volume and Biomass Allometric Equations: From Field Measurement to Prediction*; Food and Agricultural Organization of the United Nations: Rome, Italy; Centre de Cooperation Internationale en Recherche Agronomique pour le Developpement: Montpellier, France, 2012.
2. Van Laar, A.; Akça, A. *Forest Mensuration*; Springer London, Limited: London, UK, 2007; Volume 13.
3. GlobeAllomeTree. Available online: <http://www.globallometree.org/> (accessed on 20 November 2013).
4. Zianis, D.; Seura, S.M. *Biomass and Stem Volume Equations for Tree Species in Europe*; Finnish Society of Forest Science, Finnish Forest Research Institute: Vantaa, Finland, 2005; Volume 4.
5. Leeuwen, M.; Nieuwenhuis, M. Retrieval of forest structural parameters using LiDAR remote sensing. *Eur. J. For. Res.* **2010**, *129*, 749–770.
6. White, J.; Wulder, M.; Varhola, A.; Vastaranta, M.; Coops, N.; Cook, B.; Pitt, D.; Woods, M. *A Best Practices Guide for Generating Forest Inventory Attributes From Airborne Laser Scanning Data Using an Area-Based Approach*; Information Report FI-X-010; Natural Resources Canada, Canadian Forest Service, Canadian Wood Fibre Centre: Victoria, Canada, 2013.
7. Simonse, M.; Aschoff, T.; Spiecker, H.; Thies, M. Automatic determination of forest inventory parameters using terrestrial laser scanning. In Proceedings of the ScandLaser Scientific Workshop on Airborne Laser Scanning of Forests, Umea, Sweden, 19 September 2003; pp. 252–258.

8. Thies, M.; Pfeifer, N.; Winterhalder, D.; Gorte, B.G. Three-dimensional reconstruction of stems for assessment of taper, sweep and lean based on laser scanning of standing trees. *Scand. J. For. Res.* **2004**, *19*, 571–581.
9. Bienert, A.; Scheller, S.; Keane, E.; Mullooly, G.; Mohan, F. Application of terrestrial laser scanners for the determination of forest inventory parameters. In Proceedings of the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Dresden, Germany, 25–27 September 2006; Volume 36.
10. Eysn, L.; Pfeifer, N.; Ressler, C.; Hollaus, M.; Graf, A.; Morsdorf, F. A practical approach for extracting tree models in forest environments based on equirectangular projections of terrestrial laser scans. *Remote Sens.* **2013**, *5*, 5424–5448.
11. Moskal, L.M.; Zheng, G. Retrieving forest inventory variables with terrestrial laser scanning (TLS) in urban heterogeneous forest. *Remote Sens.* **2011**, *4*, 1–20.
12. Fritz, A.; Kattenborn, T.; Koch, B. UAV-Based Photogrammetric Point Clouds—Tree Stem Mapping in Open Stands in Comparison to Terrestrial Laser Scanner Point Clouds. In Proceedings of the ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Rostock, Germany, 4–6 September 2013; Volume XL-1/W2, pp. 141–146.
13. Litkey, P.; Liang, X.; Kaartinen, H.; Hyyppä, J.; Kukko, A.; Holopainen, M. Single-scan TLS methods for forest parameter retrieval. In Proceedings of SilviLaser 2008, 8th International Conference On LiDAR Applications in Forest Assessment and Inventory, Edinburgh, UK, 17–19 September 2008; pp. 295–304.
14. Liang, X.; Hyyppä, J.; Kaartinen, H.; Holopainen, M.; Melkas, T. Detecting Changes in Forest Structure over Time with Bi-Temporal Terrestrial Laser Scanning Data. *ISPRS Int. J. Geo-Inf.* **2012**, *1*, 242–255.
15. 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.
16. Pfeifer, N.; Gorte, B.; Winterhalder, D. Automatic reconstruction of single trees from terrestrial laser scanner data. In Proceedings of the 20th ISPRS Congress, Istanbul, Turkey, 12–23 July 2004; pp. 114–119.
17. Bucksch, A.; Lindenbergh, R. {CAMPINO} A skeletonization method for point cloud processing. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 115–127; Theme Issue: Terrestrial Laser Scanning.
18. Bucksch, A.; Lindenbergh, R.; Menenti, M. SkelTre. *Vis. Comput.* **2010**, *26*, 1283–1300.
19. Bucksch, A. Revealing the Skeleton from Imperfect Point Clouds. Ph.D. Thesis, Technische Universiteit Delft, Delft, The Netherlands, 2011.
20. Meagher, D. Geometric modeling using octree encoding. *Comput. Graph. Image Process.* **1982**, *19*, 129–147.
21. Xu, H.; Gossett, N.; Chen, B. Knowledge and Heuristic-based Modeling of Laser-scanned Trees. *ACM Trans. Graph.* **2007**, *26*, doi:10.1145/1289603.1289610.
22. Dijkstra, E. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271.
23. Livny, Y.; Yan, F.; Olson, M.; Chen, B.; Zhang, H.; El-Sana, J. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph.* **2010**, *29*, 151:1–151:8.

24. Yan, D.M.; Wintz, J.; Mourrain, B.; Wang, W.; Boudon, F.; Godin, C. Efficient and robust reconstruction of botanical branching structure from laser scanned points. In Proceedings of the 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, Huangshan, China, 19–21 August 2009; pp. 572–575.
25. Bentley, J.L. Multidimensional binary search trees used for associative searching. *Commun. ACM* **1975**, *18*, 509–517.
26. Cohen-Steiner, D.; Alliez, P.; Desbrun, M. Variational shape approximation. *ACM Trans. Graph.* **2004**, *23*, 905–914.
27. Bayer, D.; Seifert, S.; Pretzsch, H. Structural crown properties of Norway spruce (*Picea abies* [L.] Karst.) and European beech (*Fagus sylvatica* [L.]) in mixed versus pure stands revealed by terrestrial laser scanning. *Trees* **2013**, *27*, 1035–1047.
28. Edelsbrunner, H.; Mücke, E.P. Three-dimensional alpha shapes. *ACM Trans. Graph.* **1994**, *13*, 43–72.
29. 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; Volume II-5/W2, pp. 43–48.
30. Reynolds, D. Gaussian Mixture Models. *Encyclopedia of Biometrics*; Springer Science + Business Media: New York, USA, 2009; pp. 659–663.
31. Dassot, M.; Colin, A.; Santenoise, P.; Fournier, M.; Constant, T. Terrestrial laser scanning for measuring the solid wood volume, including branches, of adult standing trees in the forest environment. *Comput. Electron. Agric.* **2012**, *89*, 86–93.
32. Raumonon, 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.
33. Jayaratna, S. Baumrekonstruktion aus 3D-Punktwolken. Diploma Thesis, Institut für Informatik, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany, 2009.
34. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum* **2007**, *26*, 214–226.
35. Dassot, M.; Constant, T.; Fournier, M. The use of terrestrial LiDAR technology in forest science: Application fields, benefits and challenges. *Ann. For. Sci.* **2011**, *68*, 959–974.
36. Data Sheet Z + F IMAGER 5010. Available online: http://www.zf-laser.com/fileadmin/editor/Datenblaetter/Datasheet_Z.F_IMAGER_5010_E.kompr.01.pdf (accessed on 18 November 2013).
37. Web-based Weather Request and Distribution System (WebWerdis). Available online: <http://www.dwd.de/webwerdis> (accessed on 18 November 2013).
38. Morhart, C.; Sheppard, J.; Spiecker, H. Above Ground Leafless Woody Biomass and Nutrient Content within Different Compartments of a *P. maximowiczii* x *P. trichocarpa* Poplar Clone. *Forests* **2013**, *4*, 471–487, doi:10.3390/f4020471.
39. Disney, M.; Kalogirou, V.; Lewis, P.; Prieto-Blanco, A.; Hancock, S.; Pfeifer, M. Simulating the impact of discrete-return lidar system and survey characteristics over young conifer and broadleaf forests. *Remote Sens. Environ.* **2010**, *114*, 1546–1560.

40. Disney, M.; Lewis, P.; Gomez-Dans, J.; Roy, D.; Wooster, M.; Lajas, D. 3D radiative transfer modelling of fire impacts on a two-layer savanna system. *Remote Sens. Environ.* **2011**, *115*, 1866–1881.
41. Data Sheet Laser Control. Available online: http://www.zf-laser.com/fileadmin/editor/Broschueren/Z_F_LaserControl_kompr.pdf (accessed on 18 November 2013).
42. Java Standart Edition Development Kit. Available online: <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html> (accessed on 18 November 2013).
43. Java 3D. Available online: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138252.html> (accessed on 19 May 2014).
44. Jama 1.0.3. Available online: <http://math.nist.gov/javanumerics/jama/> (accessed on 18 November 2013).
45. CyberVRML97 for Java. Available online: <http://sourceforge.net/projects/cv97java/> (accessed on 20 November 2013).
46. MeshLab. Available online: <http://sourceforge.net/projects/meshlab/> (accessed on 20 November 2013).
47. CloudCompare. Available online: <http://www.danielgm.net/cc/> (accessed on 18 November 2013).
48. Chand, D.R.; Kapur, S.S. An Algorithm for convex polytopes. *J. ACM* **1970**, *17*, 78–86.
49. De Berg, M.; van Kreveld, M.; Overmars, M.; Schwarzkopf, O.C. *Computational Geometry*; Springer: Berlin/Heidelberg, Germany, 2000.
50. Gander, W.; Golub, G.; Strebler, R. Least-squares fitting of circles and ellipses. *BIT Numer. Math.* **1994**, *34*, 558–578.
51. Pratt, V. Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph.* **1987**, *21*, 145–152.
52. Ahn, S.J.; Rauh, W.; Warnecke, H.J. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognit.* **2001**, *34*, 2283–2303.
53. Marquardt, D. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441,
54. Zobel, B.J.; Buijtenen, J.V. *Wood Variation: Its Causes and Control.*; Springer-Verlag: Berlin/Heidelberg, Germany, 1989.
55. SimpleTree Project. Available online: <http://www.simpletree.uni-freiburg.de/> (accessed on 19 May 2014).
56. Disney, M.; Raunonen, P.; Lewis, P. Testing a new vegetation structure retrieval algorithm from terrestrial lidar scanner data using 3D models. In Proceeding of the SilviLaser 2012, Vancouver, BC, Canada, 16–19 September 2012.
57. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
58. SphereFitting. Available online: <http://www.caves.org/section/commelect/DUSI/openmag/pdf/SphereFitting.pdf> (accessed on 27 November 2013).

59. El-Halawany, S.; Lichti, D. Detection of Road Poles from Mobile Terrestrial Laser Scanner Point Cloud. In Proceedings of the 2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM), Xiamen, China, 10–12 January 2011; pp. 1–6.

© 2014 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 license (<http://creativecommons.org/licenses/by/3.0/>).