

aTrunk

An ALS-based Trunk Detection Algorithm

Schriftliche Hausarbeit
zur Erlangung des Grades eines
Master of Science

an der Universität Trier
Fachbereich VI

vorgelegt von
Sebastian Lamprecht
Avelsbacher Str. 60a
54295, Trier

Trier, im März 2015

1. Gutachter: Prof. Dr. Thomas Udelhoven
2. Gutachter: Dr. Johannes Stoffels

Contents

1	Abstract	1
2	Introduction	1
2.1	Relevance and Objective	1
2.2	Current Approaches	2
3	Materials and Methods	3
3.1	Study Area	3
3.2	Data	3
3.2.1	ALS Data	3
3.2.2	Validation Data	4
3.2.2.1	Diameter at Brest Height	4
3.2.2.2	GNSS Measurements	4
3.2.2.3	Terrestrial LiDAR	5
3.3	Methods	5
3.3.1	Assumptions on Trunk Representation	5
3.4	Trunk Detection Algorithm	6
3.4.1	Basic Data and Data Structure	7
3.4.1.1	Digital Terrain Model	7
3.4.2	Divide & Conquer	9
3.4.2.1	Splitting Implementation	9
3.4.3	Separation of the Trunk Section	10
3.4.4	Clustering	11
3.4.4.1	Cluster Definition	11
3.4.4.2	Clustering Implementation	12
3.4.5	Trunk Model	12
3.4.5.1	Basic PCA-Model	13
3.4.5.2	Idea of Best Model Selection	14
3.4.5.3	Model Quality	14
3.4.6	Merge Duplicated Trunks	16
3.5	Trunk Model Properties	16
3.5.1	Principal Component Model	16
3.5.2	Trunk Orientation	17
3.5.3	Position	17
3.5.4	Trunk Height	18
3.5.5	Quality Criteria	18
3.6	Methods of Evaluation	18
3.6.1	Affine Point Pair Registration	20
3.6.2	Rigid Point Set Registration	21
3.7	Sensitivity Analysis	21
4	Results	24
4.1	Evaluation	24
4.2	Modelling Results	26

5	Discussion	28
5.1	Potential Technical Improvements	29
5.2	Outlook	30
6	Author Contributions	30
7	Conflicts of Interest	30

List of Figures

1	Study area.	4
2	Sketches of the major steps of the <i>aTrunk</i> approach.	6
3	Flowchart of the <i>aTrunk</i> algorithm.	8
4	Sketch of an ALS forest cross section.	10
5	Two-dimensional clustering approach using distance parameter δ and a randomly chosen initial point p_i . Grey dots stand for the original points, while identified cluster regions are highlighted in different colours.	11
6	Three-dimensional one component PC trunk modelling concept with P_{Ct} points (black dots) and centre point (red dot).	13
7	RANSAC-based model fitting approach. Two possible raw trunk models (figures 7(a) and 7(b)) using two randomly chosen points (blue dots), inlier threshold (red dashed lines), identified inliers (green dots) and outliers (red dots). Grey dots correspond to not considered points. Final model (figure 7(c)) using inlier points t_P (black dots), centre point t_c (red dot) and trunk position t_p (green dot).	15
8	Comparison of the GNSS positions and the TLS reference positions. The left image illustrates the assignment of the positions using absolute coordinates. Here the notes about the detection rate should be neglected because the observation areas did not match completely. The right image illustrates the relative positional differences between the reference positions and the corresponding GNSS positions.	20
9	Rigid point set registration between detected positions (yellow dots) and reference positions (blue crosses) using the algorithm of Jian and Vemuri [16]. Before registration (left image) and after registration (right image).	21
10	Visualisation of the sensitivity analysis using the detection rate as the objective function. A global maximum is considered the optimal value of the corresponding parameter.	23
11	Visualisation of the sensitivity analysis using equation 18 as the objective function. A global minimum is considered the optimal value of the corresponding parameter.	23
12	Evaluation of the detected <i>aTrunk</i> positions using the TLS positions as reference. The left image illustrates the assignment of the positions using absolute coordinates. The right image illustrates the relative positional differences between the reference positions and the corresponding detected positions.	24
13	Evaluation of the detected <i>watershed</i> positions using the TLS positions as reference. The left image illustrates the assignment of the positions using absolute coordinates. The right image illustrates the relative positional differences between the reference positions and the corresponding detected positions.	25
14	Evaluation of the detected <i>combined</i> approach positions using the TLS positions as reference. The left image illustrates the assignment of the positions using absolute coordinates. The right image illustrates the relative positional differences between the reference positions and the corresponding detected positions.	25

15	Boxplots of the positioning residuals of the <i>aTrunk</i> , <i>watershed</i> and <i>combined</i> approach.	26
16	Modelling results of the study area.	27
17	Modelling results of the 1 km ² sample.	27
18	3D-view of the 1 km ² sample with detected trunks. Blue dots stand for trunk model supporting points, red dots for outliers.	28
19	3D-view of the 1 km ² sample with detected trunks and vegetation cover . .	28

List of Tables

1	Trunk Model Properties	16
2	Model Parameters	19
3	Parameter Groups	22

Preface

This masters thesis is designed in the form of a scientific article. Additional information is added in blue highlighted frames in the running text. The *aTrunk* implementation and the complete article in the submit version of the *Remote Sensing* journal [20] can be found in the digital appendix. It should be noted that the section and figure numbering of the article differs from this extended version.

1 Abstract

This paper presents a novel tree trunk detection approach for high resolution multi pulse airborne LiDAR (Light Detection And Ranging). The multi-core *Divide & Conquer* algorithm uses a 3D-clustering approach to isolate points associated with single trunks directly out of the raw point cloud. For each trunk, a principal-component-based linear model is fitted, while a modification of *LO-RANSAC* is used to identify an optimal model. The algorithm returns a vector-based model for each identified trunk while parameters like the ground position, zenith orientation, azimuth orientation and length of the trunk are additionally provided. The algorithm performed well for a study area of 109 trees (about 2/3 spruce and 1/3 beech), with a point density of around 7.6 points per m², while a detection rate of about 70% with an average difference in positioning of 0.32 m and an RMSE of 0.42 m is reached.

2 Introduction

2.1 Relevance and Objective

A sustainable multifunctional use of forest area—for forestry, timber production, energetic uses, carbon dioxide sequestration—needs an exact knowledge of the available number, distribution and species of trees, as well as the wood volume or the leaf area index (LAI). Not only is this knowledge necessary for forest management, but also for the quantification of forest ecosystem functions and services. Seen from a large-scale perspective—like for forestry offices—these desired parameters are currently often estimated by simple terrestrial estimation procedures, like angle-count sampling approaches. These approaches are also used for nationwide data ascertainment, like the *Bundeswaldinventur* in Germany, where samples on a 2 km grid are taken to assess nationwide forest conditions (cf. [3, 4]).

These random sample approaches are supported by statistical procedures to estimate the desired parameters. The interpolation of the derived information on a local scale represents a major task, which could be solved by *small area estimation* approaches of e.g. Schmid and Münnich [30].

Nevertheless, a higher spatial density of information could increase the accuracy of the estimated parameters. Especially automated procedures are needed, which derive some of these parameters using airborne LiDAR. In addition, there is a lack of knowledge

concerning the competition between different tree species and its effect on the spatial distribution of the trees.

The aim of the *ATrunk* approach is to identify single trunks based on data gained from high resolution multi-pulse airborne LiDAR. The novel point-based detection algorithm promises a reduction of information loss compared to raster-based approaches due to a direct usage of the raw point cloud.

Focusing on the trunks should ensure an accurate positioning, which can easily be used for validation because of a good comparability to ground-measured positions. In addition to these standard parameters, information like the zenith and azimuth orientation of each trunk shall be generated, which can again be used for further analysis. Especially the identification of trunks promises more information about the usable wood volume and the competitive behaviour of different tree species.

2.2 Current Approaches

In the past, many different approaches have been developed aiming at the detection of single trees using airborne LiDAR by analysing crown shapes. Nearly all approaches are using raster-based digital terrain models (DTM) or canopy models (DCM) to identify single trees. Often rudimentary detection approaches like local maximum identification (as described by Persson et al. [24]) are used, because it is assumed that the tops of the tree crowns are clearly separable from each other. After this first identification, the tree crowns are separated using different segmentation approaches, for instance algorithms inspired by watersheds (e.g. developed by Chen et al. [5]) or a segmentation of “[...] single trees in LIDAR raw data using cluster analysis, choosing local maxima as starting positions [...]” Morsdorf et al. [21]. Some fully GIS-based (Geographic Information System) approaches (e.g. Tiede et al. [31]) are also popular. The tree position and crown radii are rarely estimated simultaneously, except for Zhou et al. [35] who use a *marked point process model* to identify single trees. More up-to-date approaches, which generally use the same ideas, are discussed in Kaartinen et al. [17].

The results of this first crown identification are quite similar, while some of the desired parameters—like the crown positions, tree height, trunk volume, crown areas or crown radii—are estimated. Nearly all approaches use the raw LiDAR point cloud or voxel models and additional multi-spectral images as basic data for an additional more complex crown analysis. A focus is set on species identification (cf. Holmgren et al. [13], Ørka et al. [36] or Yu et al. [34]), a general analysis of the tree crowns (cf. Duncanson et al. [7]) or even on an identification of the trunks (cf. Reitberger et al. [29]).

It should be mentioned that the accuracy in positioning of pure raster-based tree detection approaches depends on the cell size. There is an obvious loss of information when rastering a point cloud. Even a higher grid resolution does not guarantee an increased accuracy because of a necessary interpolation of missing values. This effect is intensified by the often used smoothing of the DTM or DCM (done e.g. by Chen et al. [5] or Reitberger et al. [29]). In particular, the trunk positions cannot be derived accurately, because the presumption that the centre of the crown is directly located above the trunk root is not tenable.

The Study of Kaartinen et al. [17] has shown an average positional difference of different raster-based detection approaches of about 0.5 m up to above 1.5 m or RMSE values of 0.7 m up to 2 m. In comparison, Edson and Wing [9] notice average positional differ-

ences of about 2 m for the software programs *FUSION*, *TreeVaW* and a watershed based approach.

Nevertheless, most of these papers deal with an additional point-based analysis of the trees because of the mentioned lower information content of rasters. In preparation for this paper, no approach aiming at the direct detection of trees using the raw LiDAR point cloud which renders a raster-based pre-treatment of the DTM or DCM unnecessary could be found. One argument for using raster-based procedures is probably the smaller amount of data connected with an assumed reduced analysis effort. However, for a detailed analysis of the crowns, using the point cloud is mandatory. In addition, approaches for a direct detection of trunks can rarely be found in literature, although Reitberger et al. [29] used a RANSAC-based trunk modelling approach to detect trunks after the raster-based crown detection. Thus they reached an accuracy in positioning increased by up to 25 %.

By aiming at the detection of the trunks, previously performed crown detections seem to be unnecessary. In particular, a direct trunk detection using the raw point cloud could increase the detection rate and the accuracy in positioning while minimizing the analysis overhead.

3 Materials and Methods

3.1 Study Area

The study area—illustrated in figure 1—is located near the city Hermeskeil in Rhineland-Palatinate, Germany, at the lat-lon-coordinates 7.1676, 49.8136 with an expanse of about $100\text{ m} \times 80\text{ m}$ and an area of about 5100 m^2 . The examined open forest compartment with tree heights of up to 35 m was dominated by spruces (59 trees recorded), but edged by beeches (16 trees) at the western part of the study area. A measurement campaign on Tuesday, 19th August 2014 derived different types of validation data. This study site was selected, because the forest compartment was to be cleared by a harvester. With this information an analysis of the wood volume can be performed in a further study.

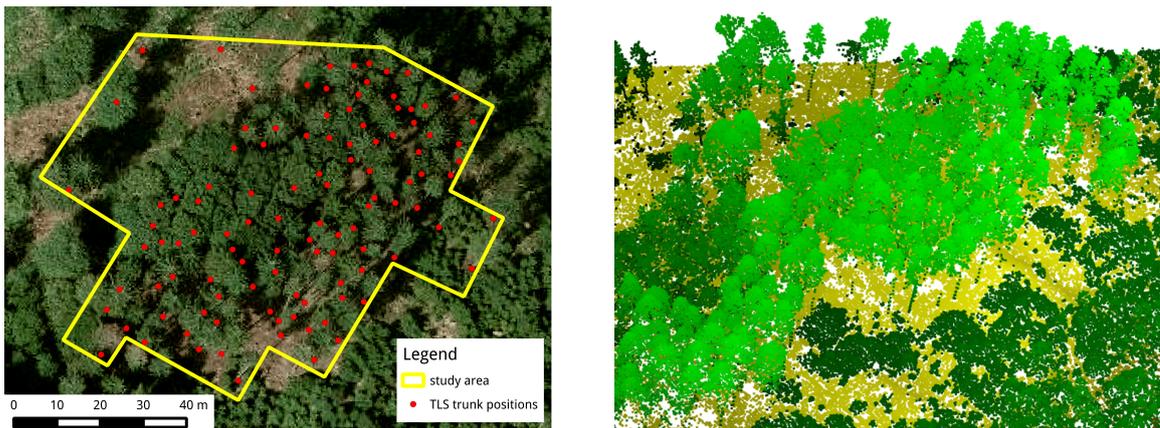
3.2 Data

High resolution multi-pulse airborne laser scanner (ALS) data served as base data for modelling while terrestrial data of a forest compartment was used for validation.

3.2.1 ALS Data

The ALS data used in this study was collected by the *Landesamt für Vermessung und Geobasisinformation* of Rhineland-Palatinate for the whole federal state with a generally high quality. The data was available in form of csv-files with a spatial expanse of $1\text{ km} \times 1\text{ km}$ for each dataset.

For the evaluation of the *aTrunk* approach, a subset of about $130\text{ m} \times 90\text{ m}$ with a Gauss-Krüger system extent of [368101, 5519482, 368232, 5519574] containing the study area was chosen. The data quality of this subset was excellent because of a high point density of about 7.6 points per m^2 and a sparse forest structure, which guarantees a good recording of the trunks. Figure 1(b) illustrates the three-dimensional structure of the



(a) Study area near Hermeskeil with trunk positions predicted by a TLS-based (Terrestrial Laser Scan) trunk detection approach, with background WMS-Service [18].

(b) 3D ALS view of the study area. Yellow-coloured pixels correspond to ground pixels and green pixels are associated with vegetation. The brightness of the pixels corresponds to their elevation value.

Figure 1: Study area.

study area. In addition, a full 1 km² dataset with an extend of [368000, 5519000, 369000, 5520000] and a point density of 7.7 points per m² was analysed.

3.2.2 Validation Data

The collection of validation data aimed especially at the position of the trunks because this information can be easily used to estimate the detection rate of the algorithm and the accuracy in modelling the trunk positions. Therefore differential GNSS (Global Navigation Satellite System) measurements of the trunk positions were taken. In addition, terrestrial laser scans (TLS) were taken, because it is possible to get more accurate information about the topology of the trunks, trunk diameter and trunk orientation by using a tree modelling approach (e.g. Bienert et al. [2]). Moreover, ground measurements of the diameter at breast height were taken for each trunk.

3.2.2.1 Diameter at Brest Height To get an idea about the wood volume of each trunk and to get information about the error margin of the modelled trunks, the diameter at breast height (DBH) was measured at a level of about 130 m above ground. In addition, this information can be used to compare this ‘ground truth’ to the TLS measurements. The DBH was determined by measuring the trunk circumference with a measuring tape.

3.2.2.2 GNSS Measurements For a measurement of the trunk positions, a differential GNSS of the type *Topcon HiPer V* (cf. Topcon Corporation [32]) was used. Because of a poor signal, the total accuracies of the measurements were quite low, with location differences clearly above 0.5 m.

To determine the ground positions, the rover element of the differential GNSS was

held directly next to the trunk, while it was attempted to take advantage of inclined trunks by placing the antenna directly above the trunk root. As it is obvious that the accuracy of these measurements is not ideal, these problems are dealt with in section 3.6.

However, the focus was set on trunk topology while neglecting the absolute position, because this information is needed to evaluate the detection rate and the relative accuracy.

3.2.2.3 Terrestrial LiDAR Eight terrestrial laser scans were taken with a *Laser Scanner Photon 120* of the manufacturer FARO[®], which measures at a wavelength of 758 nm and reaches a ranging accuracy of ± 2 mm at a distance of 25 m (cf. FARO Europe GmbH [11]). Each of the all-around scans had a scan size of 8044×3446 pixels with the scanner-specific parameters: 1/5 resolution and $3\times$ quality. The positions of the scans were chosen in such a manner that the study area was mapped completely. The alignment of the single scans to each other was prepared by placing reference spheres which can be identified in the post-processing software.

These TLS datasets were used to estimate the trunk position and diameter by applying a slicing approach (like Bienert et al. [2]). The slices were 5 cm thick, beginning at a level of 1.3 m up to the crowning height with a vertical distance of 1 m. To compare these models to the *aTrunk* positions, the positions at ground level were estimated by fitting a linear model to the slice centres of each trunk.

Because of some rain showers during the measurement campaign, the strived for number of scans could not be reached. In addition, the distances between the reference spheres were chosen too large, which led to problems in the alignment of the datasets. Therefore the slicing approach was applied to each scan. The estimated trunk positions were used to assign the datasets to the central scan (see section 3.6). Therefore an affine point set registration approach was used.

3.3 Methods

The ideas and concept of the *aTrunk* approach are clarified in this section, while the equations and parameters of the implemented algorithm are presented step by step in section 3.4. In addition, the validation of the estimated trunk positions is described in section 3.6. In section 4 the modelling results of the study area are presented and discussed. Finally an overall discussion is given and an outlook of possible improvements given by this approach is proposed in section 5.

3.3.1 Assumptions on Trunk Representation

Firstly, different multi-pulse ALS point clouds of forested areas were visually inspected. The *one of many pulses* primarily display the vegetation layer. Therefore the *first and only* and *last of many* pulses can be neglected. Hence the vegetation layer of the study area (green highlighted dots in figure 1(b)) was analysed. The 3D view of the vegetation layer shows the complicated crown structure, which allows for a visual distinguishing between different species. It is possible to identify single trunks because of the linear structure of the corresponding points.

This visual inspection suggests the following characteristics of a trunk mapped by ALS, which forms the basis of the trunk model concept outlined in sections 3.4.4 and 3.4.5.

The LiDAR points associated with a trunk are ...

- spatially separable from the crown portion and the ground covering vegetation.
- moderately surrounded by points associated with branches, foliage or other objects.
- arranged in a straight line, which is oriented along the growth direction of the trunk. The maximum deviation from this line depends on the length of the trunk, e.g. caused by irregular growth or branching.
- largely uniformly distributed in growth direction of the trunk, which is substantiated in the spatial resolution of the scanner.

3.4 Trunk Detection Algorithm

Figure 2 illustrates the major steps of the presented *aTrunk* approach, which are described in detail below. The actual implementation was done in *python* [26].

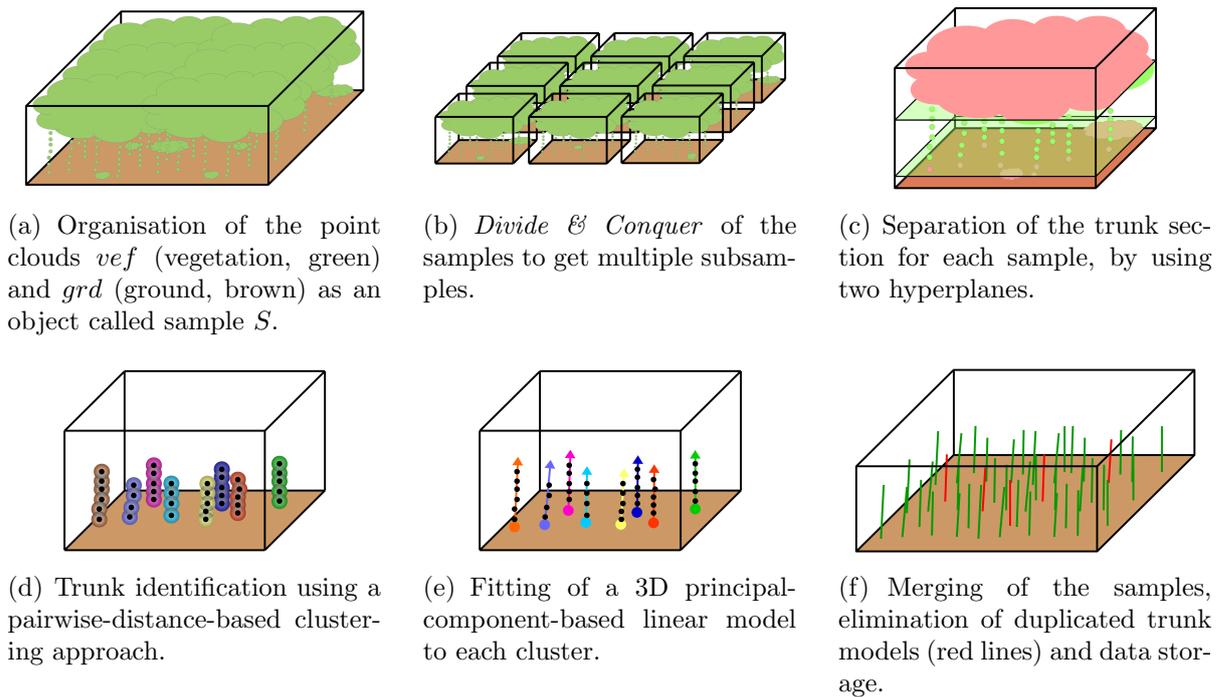


Figure 2: Sketches of the major steps of the *aTrunk* approach.

For the implementation, a focus was set on the universality, correctness and usability of the program. Attention was payed on a clear file structure, class structure and readability of the code. In addition, all parameters are summarised in a configuration file.

The *Divide & Conquer* approach (see figure 2(b)) enabled a multi core imple-

mentation, which reduces the computation time significantly. After each split of a sample, a new job is created, which can either generate a result or new (easier to handle) jobs. The list of jobs is handled by the CPU cores, while each cores takes one job and executes it. When new jobs are generated, they are appended to the list of jobs, otherwise the results are collected.

To enable an optimal up-to-dateness and to ensure the availability of the modelling results for remote clients a database is proposed as storage. Therefore a *PostgreSQL* database [25] with the *PostGIS* [22] extension was used. So, next to the pure data storage and data retrieval, a spatial analysis and 2D visualisation using *QGIS* [23] or 3D visualisation using *R* [27], is enabled.

The flowchart in figure 3 illustrates the logical steps of the implemented algorithm in detail. Here all necessary parameters and decisions are linked.

3.4.1 Basic Data and Data Structure

The algorithm uses the *first and only* and *last of many* product (hereafter referred to as *grd*) and the *one of many* product (hereafter referred to as *vef*) of the ALS basic data, which were available as csv files. The *grd* data corresponds to the terrain or the ground, while the *vef* represents the surface or vegetation cover. Both datasets have to be represented in the same coordinate system and extent. It is necessary that the *grd* point density is significantly higher than the stocking density of the forest. This is justified by the usage of the *grd* dataset as basic data for fitting terrain models.

The *grd* and *vef* datasets are brought together in one object—hereafter referred to as sample S . Such a sample uses its ground points $S_{grd} \subseteq \mathbb{R}^3$ to derive a DTM which can be used to estimate the point heights of its vegetation points $S_{vef} \subseteq \mathbb{R}^3$ above ground. For the purpose of this paper a linear plane is fitted to these points.

The three-dimensional extent of the sample is mainly defined by the S_{vef} points, in which the minimum z-value is taken from the S_{grd} point cloud because it can be assumed that the S_{grd} points define the lower limit. So the extent is given by equation 1.

$$extent(S) := [\min(S_{vef_x}), \min(S_{vef_y}), \min(S_{grd_z}), \max(S_{vef_x}), \max(S_{vef_y}), \max(S_{vef_z})] \quad (1)$$

The *vef* and *grd* datasets are sketched in figure 2(a) as green or brown areas or dots, while the extent is sketched as a box.

A special property of a sample is that it can be separated into two different subsets of the same class (see section 3.4.2). In this case the point clouds S_{grd} and S_{vef} are split simultaneously in some xy-direction. For this reason it has to be mentioned that the DTM is not calculated for each sample because the heights of the points above ground are not needed until the trunk models are fitted (see section 3.4.5).

3.4.1.1 Digital Terrain Model To receive a DTM some kind of a function $f_{S_{grd}} : \mathbb{R}^2 \rightarrow \mathbb{R}^3 : (x, y) \mapsto (x, y, z)$ has to be derived. This function returns an estimated height $z \in \mathbb{R}$ above ground for any point with the coordinates x and y . There is a great number of possibilities to find such a function, while attention should be paid to the accuracy of this terrain model.

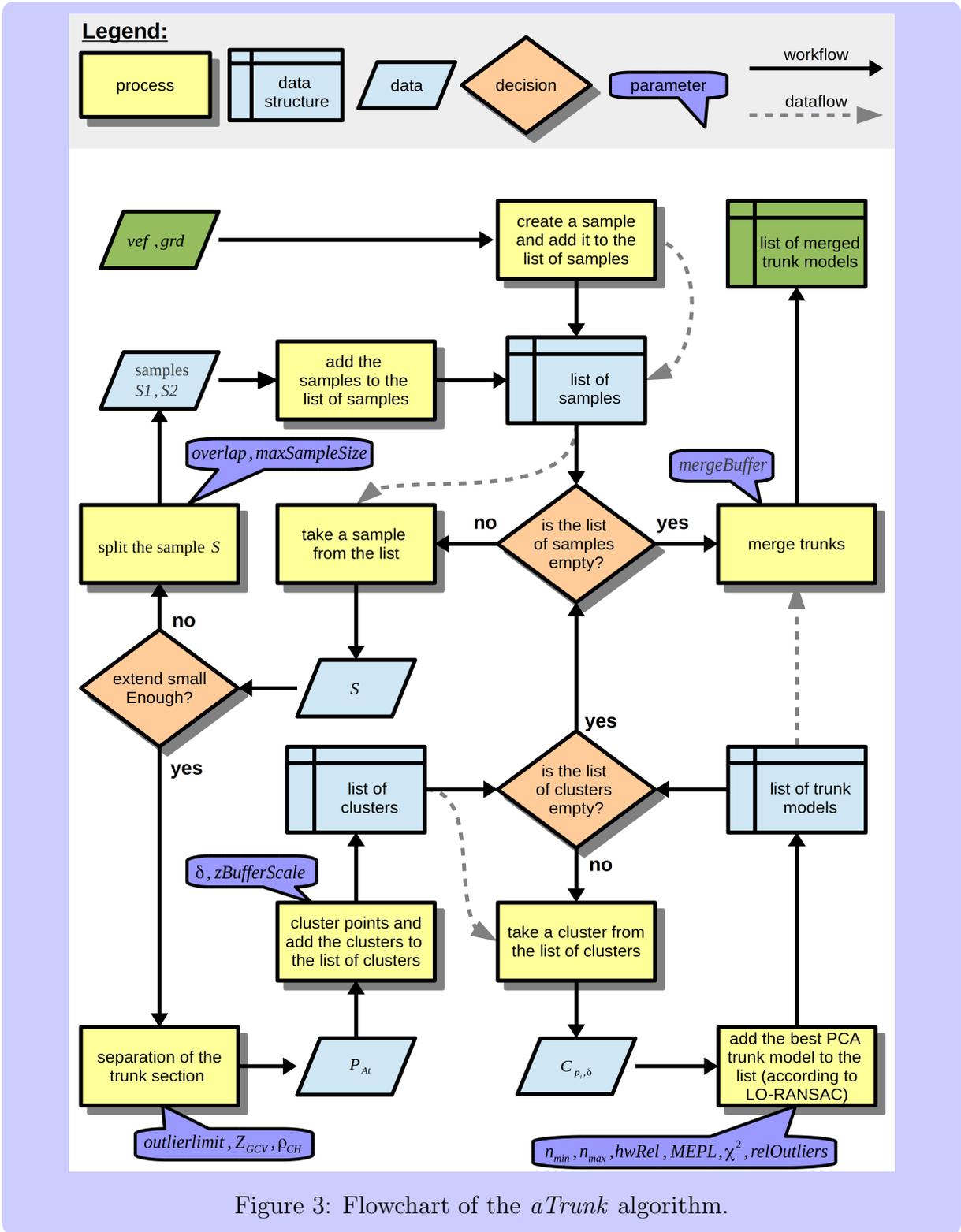


Figure 3: Flowchart of the *aTrunk* algorithm.

The surface plane is derived by a linear regression model of degree one (illustrated by equation 3). The scalars $c_0, c_1, c_2 \in \mathbb{R}$ correspond to the regression coefficients of the solved regression equation 2.

$$S_{grd_z} = c_0 + c_1 \cdot S_{grd_x} + c_2 \cdot S_{grd_y} \quad (2)$$

$$f_{S_{grd}}(x, y) := c_0 + c_1 \cdot x + c_2 \cdot y \quad (3)$$

This plane DTM is modelled in the close region of single trunks. The accuracy of the terrain model varies with the extent of the sample, the variability of the surface and the number of available ground points. Therefore more advanced terrain models are imaginable, like polynomials of a higher degree or splines. Due to the complexity of the modelling function, the minimum number of points to derive a DTM varies, so this number is ensured by the parameter *minGrdPoints*.

3.4.2 Divide & Conquer

The large amount of points makes a direct analysis of the point cloud almost unmanageable. For this reason the idea of introducing a splitting step is to divide the point cloud into subsets that are easier to handle, which is sketched in figure 2(b). In addition, this procedure facilitates the fitting of local DTMs, which allows a separation of points associated with trunks from other points (done in section 3.4.3).

Towards this, a sample S is split into multiple smaller samples, while a split is always done in a xy-direction. The size of a sample needs to be selected in such a way that it is small enough to model the ground accurately by the simple (e.g. linear) DTM, but large enough to include at least one complete tree. The splitting of the sample S can be done gradually. Each sample is separated into two new samples until the extent of the sample falls below a predefined threshold *maxSampleSize*. This successive splitting allows a multi-core implementation of the algorithm in which it is possible to add a new CPU core to the program after each split.

Nevertheless, the splitting of a sample results in some disadvantages, which have to be dealt with. It is possible that points associated with a single trunk are separated from each other at random. This would result in an under-detection of trees. Therefore an overlapping area along the cutting edge of width *overlap* is suggested. The implementation of an overlap inevitably results in an additional analysis effort and the possibility of a multiple detection of a single trunk. These multiple detected trunks are merged, as described in section 3.4.6.

3.4.2.1 Splitting Implementation A splitting of a sample S can be done in different ways. For example a grid-based or principal-component-based splitting of the sample is imaginable. In this paper, a grid-based splitting of the samples is proposed in which each sample is split either in x-direction or y-direction. The selection of the splitting direction is based on comparing the S_{vef_x} -range to the S_{vef_y} -range, while the split is performed in the direction of the minimum range. The centre of the dividing line is set to half of the corresponding range. To ensure the above mentioned overlapping of the subsamples, the *overlap* parameter is considered by shifting the dividing line by half of this parameter for each subsample.

3.4.3 Separation of the Trunk Section

The separation of the trunk section is based on the assumption that the points associated with trunks lie between points which are associated with ground-covering vegetation and those associated with tree crowns. A simple approach to distinguish these point layers is to use two threshold planes parallel to the DTM.

The first lower plane at a height of $Z_{GCV} \in \mathbb{R}^+$ shall detach the ground-covering vegetation. The parameter Z_{GCV} corresponds to the expected height of the low-growing vegetation. It has to be mentioned that a value selected too high reduces the number of available points for the trunk model fitting. The height $Z_{CH} \in \mathbb{R}^+$ of the second plane should comply with the expected crowning height. Therefore this threshold is selected dynamically by estimating the crowning height Z_{CH} by equation 4 while $\rho_{CH} \in]0, 1]$ conforms to some proportion of the tree height. It should be noted that Z_{CH} is strongly affected by outliers in z -direction—e.g. by noise, flying birds or power supply lines. Therefore a pre-cleaning of the data is performed by removing all those points whose z -value is above the quantile $1 - outlierlimit$.

$$Z_{CH} = \rho_{CH} \cdot \max(S_{vef_z}) \quad (4)$$

The parameter ρ_{CH} is set regardless of the tree species. This is justified by the assumption that especially the points below the crowning height are arranged in a line, so the trunk models should be stable even if the models reach to the lower parts of the crowns. The result of this separation step is a set of points associated with the trunk section $P_{TS} \subseteq S_{vef}$. Figure 2(c) illustrates the separation of the points associated with trunks by the two threshold planes.

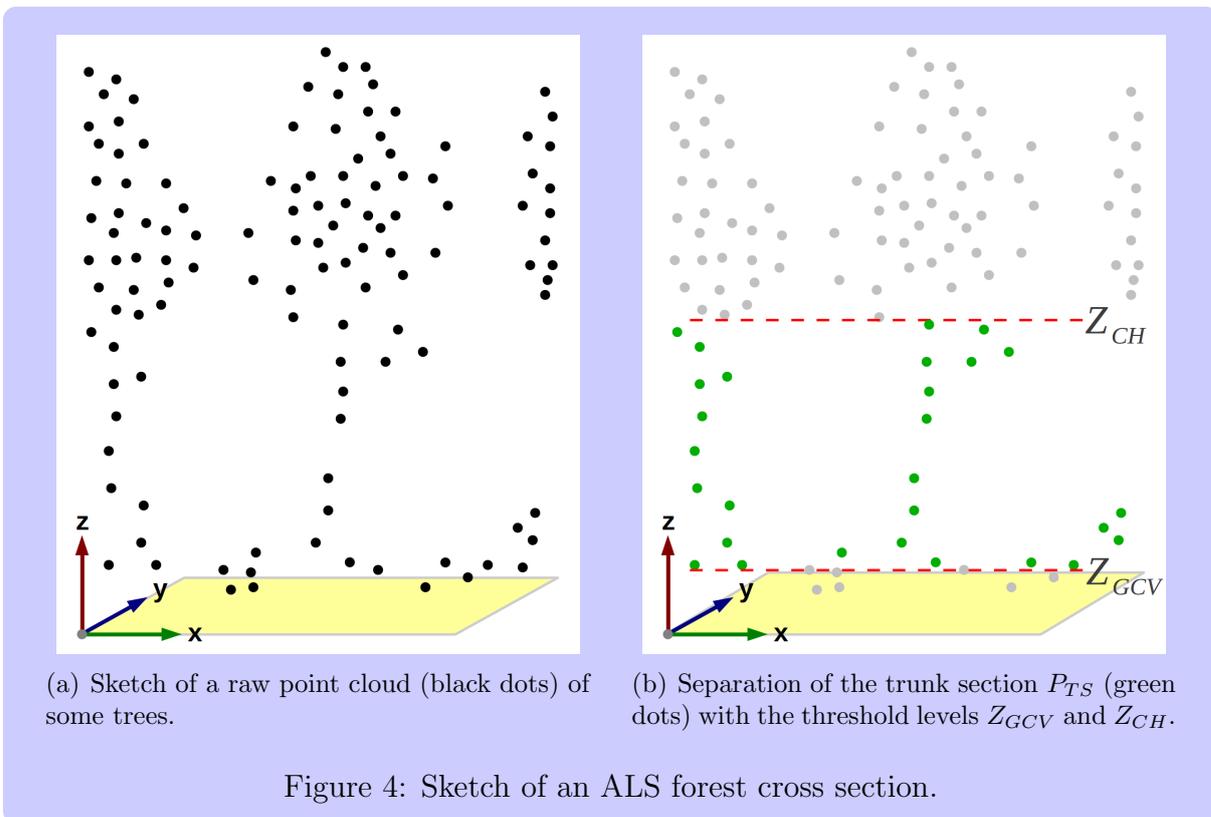


Figure 4 gives a more detailed illustration of the separation of these points.

3.4.4 Clustering

The clustering step is used to identify points potentially associated with trunks by using the P_{TS} point set of section 3.4.3. It is assumed that several points which are spatially close together will form a trunk. Thus, isolated points without spatial neighbours are assumed to be noise or spare vegetation. Therefore the major problem is to identify clusters of points, without knowing the number of clusters a priori (in contrast to e.g. a *k-Means*-based clustering approach).

In addition, it is not primarily desired to minimize the inter-group variance and maximise the intra-group variance (e.g. done by decision tree approaches like *Random Forest*), but to identify close points which form continuous objects.

The following cluster definition is used to isolate points associated with a single trunk because of the mentioned reasons.

3.4.4.1 Cluster Definition The cluster definition is inspired by the clustering approach used by Reitberger et al. [29] who take advantage of the pair-wise spatial neighbourhood of points to identify clusters. To omit the additional estimation of the number of clusters, the clusters are identified by the spatial proximity of their points only. The universal definition of this clustering can be performed independently from the dimension $N \in \mathbb{N}^+$ of a point cloud or vector space $P \subseteq \mathbb{R}^N$. Therefore a cluster $C_{p_0, \delta} \subseteq P$, as it is defined by equation 5, can be developed around a point $p_0 \in P$ using a threshold distance $\delta \in \mathbb{R}^+$.

$$C_{p_0, \delta} := \{ p \in P : \|p - p_0\|_N < \delta \vee \exists p_c \in C_{p_0, \delta} : \|p - p_c\|_N < \delta \} \quad (5)$$

For the purpose of this paper this cluster definition allows for a separation of spatially close points without knowing the number of clusters respectively trunks a priori. Additionally, the clustering can be done for a three-dimensional point cloud with $N = 3$ just

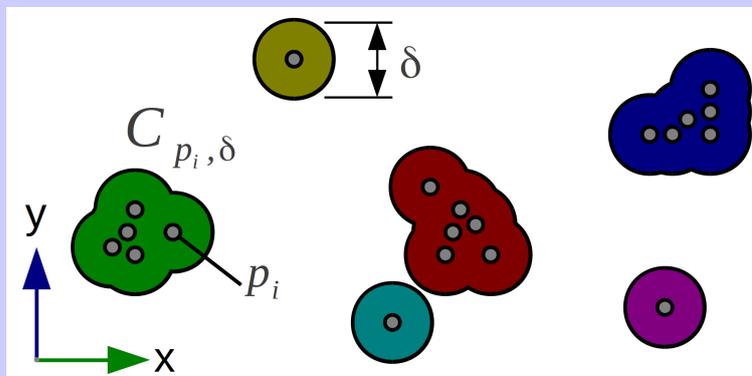


Figure 5: Two-dimensional clustering approach using distance parameter δ and a randomly chosen initial point p_i . Grey dots stand for the original points, while identified cluster regions are highlighted in different colours.

as well as for a two-dimensional point cloud with $N = 2$. So this type of clustering is used to identify points associated with a single trunk P_{Ct} out of P_{TS} (illustrated in figure 2(d)).

Figure 5 illustrates the two-dimensional version of the clustering approach in detail. The resulting clusters are used to model the trunks (see section 3.4.5).

3.4.4.2 Clustering Implementation The given cluster definition points out that the implementation can be done by defining one (randomly chosen) point of a point cloud P as the development point p_0 of the cluster $C_{p_0, \delta}$.

Cluster development using a distance threshold $\delta \in \mathbb{R}^+$ and a point cloud $P \subseteq \mathbb{R}^N$:

```

while  $|P| > 0$  do
   $p_i \leftarrow$  random point  $\in P$ 
   $C_{p_i, \delta} \leftarrow \{p_i\}$ 
  for all  $p_c \in C_{p_i, \delta}$  do
    for all  $p_j \in P$  do
      if  $\|p_c - p_j\|_2 \leq \delta$  then
         $C_{p_i, \delta} \leftarrow C_{p_i, \delta} \cup \{p_j\}$ 
         $P \leftarrow P \setminus \{p_j\}$ 
      end if
    end for
  end for
end while

```

This clustering implementation implies a computation time complexity of order $O(n^2)$ with $n := |P|$, caused by pair-wise distance comparisons between all points of P . Therefore remarkable run times are generated when the number of points of P is high. In a two-dimensional space the *DBSCAN*-approach (cf. Ester et al. [10]) allows for an average run time complexity of $O(n \cdot \log(n))$, in which the *Eps*-parameter corresponds to δ and the *MinPts*-parameter is set to zero.

The mentioned parameters Z_{GCV} (see section 3.4.3) and *maxSampleSize* (see section 3.4.2) help to reduce the number of points to be considered. The identification of clusters associated with trunks uses a three-dimensional clustering ($N = 3$), while the modelling parameter $zBufferScale \in \mathbb{R}_0^+$ provides the opportunity to scale the z-coordinates by multiplying them with this parameter. An optional 2D-clustering is possible by setting *zBufferScale* to zero.

3.4.5 Trunk Model

To get the desired information about the trunks, a linear model is fitted to the points associated with the trunk P_{Ct} because the points are assumed to be arranged in a line. To receive the three-dimensional vector-based regression model, a principal component analysis (PCA) is used.

3.4.5.1 Basic PCA-Model A PCA of a point cloud $P \subseteq \mathbb{R}^N$ with $N \in \mathbb{N}$ dimensions provides N pair-wise orthogonal vectors \overrightarrow{PC}_i (with $i \in 1 \dots N$) which are called principal component (PC) vectors. The PCs define a projected coordinate system, in which the PC scores ($scores(P) \subseteq \mathbb{R}^N$) of the points P correspond to the projected coordinates of these points in the new coordinate system (cf. Wold et al. [33]).

The PCA model can be seen as a vector-based linear regression model, because “[...] the fitting of a principal component line to a number of data points is a least squares process” (Wold et al. [33], p.41). The principal component “[...] vectors are usually written in order of descending eigenvalues” (Wold et al. [33], p.42). So, the first principal should be oriented in the growing direction of the corresponding trunk, while the second and third PC are perpendicular to the first component and characterise the residuals.

The first principal component $t_{\overrightarrow{PC}_1}$ of the point cloud P_{Ct} is oriented in the direction of the highest variance. For a nearly perfect linear alignment of the points associated with the trunk, the $t_{\overrightarrow{PC}_1}$ vector should be oriented in the direction of the trunk. As a data pre-treatment, a mean centring of the P_{Ct} points is applied. So the trunk model corresponds to the $t_{\overrightarrow{PC}_1}$ vector which is translated to the original centre point of the defining P_{Ct} points.

The model residuals correspond to the PC scores of the second and third principal component, so these are calculated by equation 6 on page 17. Figure 6 illustrates both, the basic straight line principal component model and the corresponding PC-based trunk model.

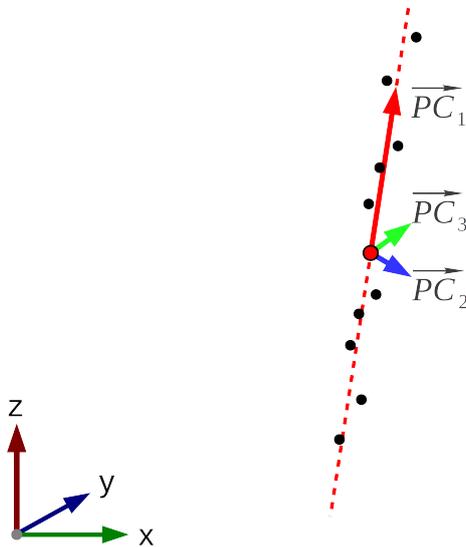


Figure 6: Three-dimensional one component PC trunk modelling concept with P_{Ct} points (black dots) and centre point (red dot).

Unfortunately, the assumption of perfectly linearly arranged points with no kind of outliers cannot be maintained. Therefore an attempt was made to find one PC model which fits most of the points by excluding outliers.

3.4.5.2 Idea of Best Model Selection One way to solve the problem of fitting a regression model to highly outlier-affected data is the *RANSAC* approach by Fischler and Bolles [12], which was also used by Reitberger et al. [29]. The *RANSAC* approach is based on the assumption that outlying points can be identified by fitting multiple models which each rely on the minimum number of necessary points.

A basic assumption is that the points are randomly chosen and all models are independent from each other. In the case of a PC model, exactly two points are needed for each *RANSAC* sample. After this initialisation, all those points which have a residuum to the model below a specific threshold are assumed to support the model, while the others are assumed to be outliers. The model with the smallest proportion of outliers is accepted as the best model.

In this study the idea of the *LO-RANSAC* approach [6] was used, which optimizes the outlier identification by running an additional model fitting for each *RANSAC* sample, using the model-supporting points. This technique relies on $k \approx \frac{\log(\eta)}{\log(1-\epsilon^m)}$ (cf. Chum et al. [6], p. 2 & 3) random samples to find—with a probability of $P(O)$ % a maximum proportion of O % outliers, with $\epsilon := P(O) \div 100 + 1$ and $\eta := O \div 100 + 1$ —a set of at least m inliers. This equation clarifies that the number of points associated with a trunk is insufficient in order to find enough independent points for the common *LO-RANSAC* approach. This is caused by the low number of points associated with a trunk (usually clearly below twenty) and an extremely high proportion of outliers (often caused by ramifications in the upper trunk section or by low-growing vegetation). To derive a deterministic model, every point pair permutation of P_{TS} is defined as a *RANSAC*-sample, whether or not the models are independent. The selection of the best model is done according to the next paragraph.

3.4.5.3 Model Quality The quality analysis of a trunk model is divided into two steps. In a first step, the validity of the model is tested, where already one fail results in rejecting the model. In a second step, the best valid model is selected by comparing the mean squared error (MSE) of each model, where an MSE close to zero can be assumed to be optimal. For the validity check, some properties of each model are evaluated according to the corresponding modelling parameters which are summarised in table 2 on page 19.

A model is assumed to be valid if ...

- the model contains enough points to ensure an accurate adaptation and unlikely false detections.
 $\Rightarrow |t_P| \geq n_{min} \in \mathbb{N}_{>1}$
- the model contains only some points, because it is assumed that a high number of neighbored points is probably caused by leaves or branches.
 $\Rightarrow |t_P| \leq n_{max} \in \mathbb{N}_{\geq n_{min}}$
- the range of z is large enough to contain a trunk.
 $\Rightarrow range(t_{P_z}) \geq minZRange \in \mathbb{R}_0^+$
- the ratio between the z -range (height) and xy -range (width) is comprehensible.
 $\Rightarrow \frac{range(t_{P_z})}{\max(range(t_{P_x}), range(t_{P_y}))} \geq hwRel \in \mathbb{R}^+$
- the zenith angle of the trunk is imaginable.
 $\Rightarrow t_\zeta \leq maxZenith \in [0, 90[$

- the model has a favourable ratio between model-supporting points and outliers.
 $\Rightarrow \frac{|t_P|+|t_O|}{|t_O|} \leq relOutliers \in [0, 1[$
- the points associated with the trunk are largely uniformly distributed in t_{PC_1} direction.
 $\Rightarrow t_{\chi^2} < uniformProb \in [0, 1]$

Due to the assumption of increasing residuals with the length of the trunk, an adaptable threshold is needed to identify outliers. Therefore a length-dependent quality criterion—called *MEPL* (Maximum Error Per Length)—is proposed, which shall privilege large trunks in residual weighting. Equation 14 (page 18) illustrates the calculation of this criterion, while the still unknown length of the trunk is approximated by the z-values of the points.

Some of the validity tests can already be evaluated before PCA modelling or only using the two initial points to avoid unnecessary evaluations. For example, some samples which contain too few points or have a zenith angle t_ζ larger than expected can be excluded.

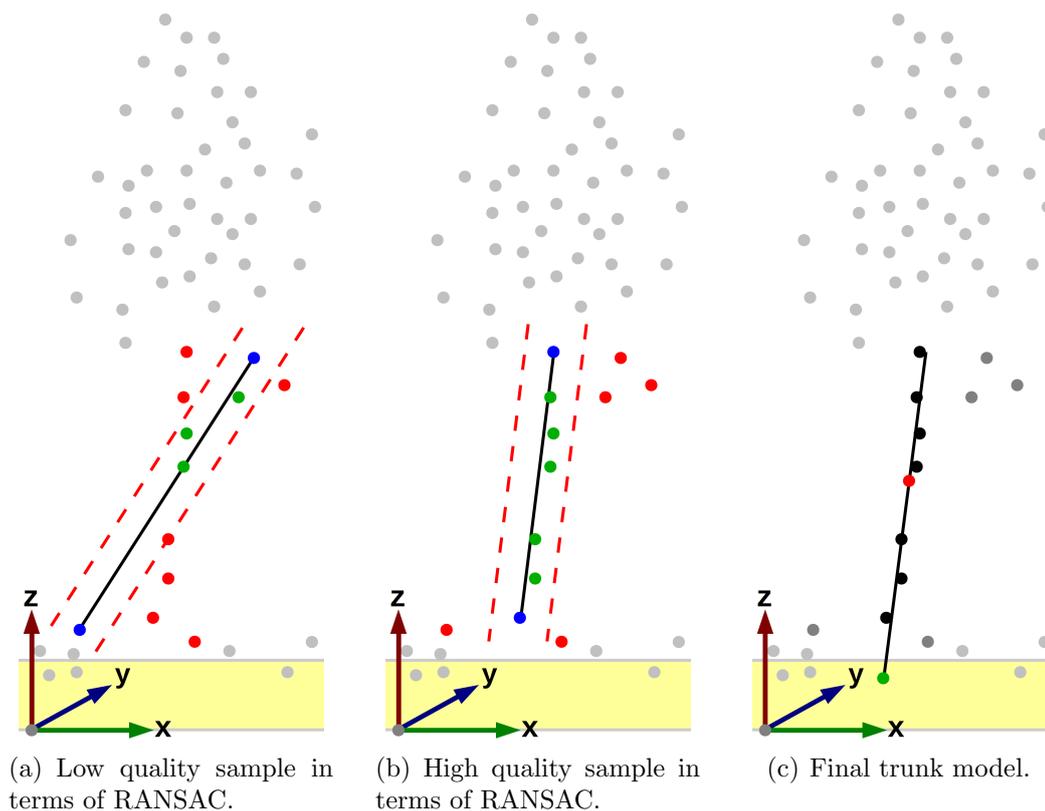


Figure 7: RANSAC-based model fitting approach. Two possible raw trunk models (figures 7(a) and 7(b)) using two randomly chosen points (blue dots), inlier threshold (red dashed lines), identified inliers (green dots) and outliers (red dots). Grey dots correspond to not considered points. Final model (figure 7(c)) using inlier points t_P (black dots), centre point t_c (red dot) and trunk position t_p (green dot).

Figure 7 illustrates the modified *LO-RANSAC* concept for the PC-based trunk modelling approach, while figure 2(e) sketches the modelling results in a sample.

3.4.6 Merge Duplicated Trunks

The split of the samples—done in step 3.4.2—with an intended overlap area can result in a multiple detection of single trunks in the overlapping separation section. This undesired effect is compensated for by an identification and merging of duplicated trunks whose centres are close together. Because it cannot be ruled out that the models differ from each other (caused by a different point basis), all raw points are joined, while duplicates are eliminated. After the merging, a new trunk model can be fitted just as explained in section 3.4.5. All the derived trunk models are combined again in a result dataset as sketched in figure 2(f).

To analyse the spatial proximity of the trunk centre points, the clustering approach as presented in section 3.4.4 with $N = 2$ is used. The parameter $mergeBuffer \in \mathbb{R}_0^+$ corresponds to the distance threshold (see table 2 on page 19). The value of this parameter should be less than the minimum expected distance between two trunks but large enough to identify slightly different trunk models. In some cases, the new model is not valid, so it is proposed to keep the model with the best quality value (presented in section 3.5.5) while discarding the others.

3.5 Trunk Model Properties

The fitted PC model provides multiple variables whose calculation is explained afterwards. Table 1 summarizes the properties of a trunk model.

Table 1: Trunk Model Properties

Property	Values Range	Unit	Description	Reference Section
t_c	\mathbb{R}^3	m	Spatial centre of the points used to fit the model	3.5.3
t_p	\mathbb{R}^3	m	Estimated position of the trunk at ground level	3.5.3
t_{top}	\mathbb{R}^3	m	Estimated position of the trunk at crowning level	3.5.3
$t_{\vec{PC}_1}$	\mathbb{R}^3	m	First principal component—models the growth direction	3.5.1
$t_{\vec{PC}_2}$	\mathbb{R}^3	m	Second principal component—associated with residuals	3.5.1
$t_{\vec{PC}_3}$	\mathbb{R}^3	m	Third principal component—associated with residuals	3.5.1
t_P	\mathbb{R}^3	m	Points associated with the trunk—used to fit the model	3.5.1
t_O	\mathbb{R}^3	m	Points which are assumed to be outliers	3.5.1
t_h	\mathbb{R}^+	m	Total height of the trunk	3.5.4
t_l	\mathbb{R}^+	m	Total length of the trunk	3.5.4
t_ζ	[0,90]	°	Zenith angle of the trunk	3.5.2
t_α	[0,360[°	Azimuth angle of the trunk	3.5.2
t_{sMEPL}	\mathbb{R}^+	m ²	Squared Maximum Error Per Length of the trunk—a quality criterion	3.5.5
t_{MSE}	\mathbb{R}^+	m ²	Mean Squared Error—a quality criterion	3.5.5
t_{χ^2}	[0,1]	–	Quality criterion—ensures a uniform distribution of the points t_P	3.5.5

3.5.1 Principal Component Model

The PC model is based on the points associated with the trunk $t_P \subseteq \mathbb{R}^3$, whose selection was explained in 3.4.5.2. In addition, the points assumed to be outliers $t_O \subseteq \mathbb{R}^3$, which surround the trunk, are provided by the model.

The three principal components $t_{\overrightarrow{PC_1}}, t_{\overrightarrow{PC_2}}$ and $t_{\overrightarrow{PC_3}} \in \mathbb{R}^3$ depend on the inliers t_P only. The characteristics of the PCA led to an orientation of the first principal component $t_{\overrightarrow{PC_1}}$ in the direction of the largest proportion of variance. The $t_{\overrightarrow{PC_1}}$ corresponds to a linear regression model of the trunk, while the residuals of this regression vector are defined by the PC scores of the second and third component. So the residuals $\epsilon(t_P) \in \mathbb{R}^{|t_P|}$ are calculated by equation 6:

$$\epsilon(t_P) = \sqrt{\text{scores}(t_P)_2^2 + \text{scores}(t_P)_3^2} \quad (6)$$

In the case of a negative orientation of the first PC in z-direction the vector is inverted by multiplying it with -1 to allow a simplified interpretation.

3.5.2 Trunk Orientation

The zenith angle $t_\zeta \in [0, 90]$ describes the angle deviation of $t_{\overrightarrow{PC_1}}$ (the trunk) to a vertical line and is calculated by equation 7:

$$t_\zeta = \frac{\text{acos}(t_{\overrightarrow{PC_{13}}})}{\sqrt{t_{\overrightarrow{PC_{11}}}^2 + t_{\overrightarrow{PC_{12}}}^2 + t_{\overrightarrow{PC_{13}}}^2}} \quad (7)$$

The azimuth angle $t_\alpha \in [0, 360[$ describes the deviation of $t_{\overrightarrow{PC_1}}$ from the northern direction which indicates the leaning direction of the trunk. So the azimuth angle t_α is calculated by equation 8:

$$t_\alpha = \text{atan}\left(\frac{t_{\overrightarrow{PC_{11}}}}{t_{\overrightarrow{PC_{12}}}}\right) \quad (8)$$

For the implementation, the python function `atan2` was used, because it gives information about the sign of t_α . So the azimuth angle is calculated by equation 9, where the azimuth angle reaches clockwise from north ($t_\alpha = 0$) over east ($t_\alpha = 90$) and south ($t_\alpha = 180$) to west ($t_\alpha = 270$).

$$t_\alpha = 180 - \text{atan2}(t_{\overrightarrow{PC_{11}}}, -t_{\overrightarrow{PC_{12}}}) \quad (9)$$

3.5.3 Position

The centre point $t_c \in \mathbb{R}^3$ of the model corresponds to the spatial centre of the points t_P associated with the trunk which were used to fit the model. So it is calculated by equation 10:

$$t_c = (\overline{t_{P_x}}, \overline{t_{P_y}}, \overline{t_{P_z}}) \quad (10)$$

A special advantage of the principal component line fitting approach is that the coordinates of every point of the trunk can be calculated as a linear combination of the first principal component $t_{\overrightarrow{PC_1}}$ and the centre point t_c . Therefore a point $p_k \in \mathbb{R}^3$ on this regression line can be calculated by equation 11, in which the scalar $k \in \mathbb{R}$ corresponds

to the desired height along the growth direction of the trunk relative to the centre point t_c :

$$p_k = k \cdot t_{\overrightarrow{PC_1}} + t_c \quad (11)$$

This feature is used to estimate the position of the trunk $t_p \in \mathbb{R}^3$, while the z-component of the t_p coordinate should be zero. Therefore the parameter k is estimated by equation 12, in which t_ζ corresponds to the zenith angle of the trunk and $h \in \mathbb{R}^+$ to the height of the centre point t_c above ground level (see section 3.4.1). It should be noted that this equation relies on the assumption of a flat ground. This approximation suffices for the goal of this application.

$$k = \frac{h}{\cos(t_\zeta)} \quad (12)$$

The trunk top position $t_{top} \in \mathbb{R}^3$ corresponds to the modelled centre of the trunk at the assumed crowning height. It can be calculated corresponding to the ground position by using equation 11, while the parameter k is replaced by the assumed height of the trunk (see section 3.5.4).

3.5.4 Trunk Height

The height of a trunk $t_h \in \mathbb{R}^+$ is implicitly given by the expected crowning height Z_{CH} (see section 3.4.3). The corresponding length of a trunk $t_l \in \mathbb{R}^+$ is estimated with equation 12, in which the parameter h is replaced by the assumed tree height t_h .

3.5.5 Quality Criteria

The quality of a trunk model can be evaluated by different criteria. Apart from the classical MSE, the (squared) MEPL (mentioned in 3.4.5.3) or the uniform distribution criterion (t_{χ^2}) of the points associated with the trunk can be used. The t_{MSE} is based on the residuals $\epsilon(t_P)$ of the regression line, defined by $t_{\overrightarrow{PC_1}}$ (equation 13). The t_{MEPL} is defined by equation 14, while the corresponding squared version t_{sMEPL} is defined by equation 15. The uniform distribution criterion of the inliers is calculated by applying a *Chi-Square Test* on the pair-wise distance according to their sorted $t_{\overrightarrow{PC_1}}$ scores.

$$t_{MSE} = \overline{\epsilon(t_P)^2} \quad (13)$$

$$t_{MEPL} := \frac{\max(\epsilon(t_P))}{\max(t_{P_z}) - \min(t_{P_z})} \quad (14)$$

$$t_{sMEPL} = (t_{MEPL})^2 \quad (15)$$

3.6 Methods of Evaluation

To carry out an evaluation of the *aTrunk* approach, it was applied to the ALS point cloud of the study area. Table 2 contains the selected parametrisation optimized by a sensitivity analysis. A local maxima-based *watershed* segmentation approach served as a benchmark for the *aTrunk* approach. Hence, the *watershed* approach was applied to a 0.5 m grid derived from the ALS point cloud.

Parameter Name	Values' Range	Unit	Description	Value in this study	Reference Section
<i>minPoints</i> (n_{min})	$\mathbb{N}_{>1}$	–	Minimum number of points assumed to form a trunk	4	3.4.5.3
<i>minGrdPoints</i>	$\mathbb{N}_{>2}$	–	Minimum number of points needed to build a DTM	3	3.4.1.1
<i>maxPoints</i> (n_{max})	$\mathbb{N}_{\geq n_{min}}$	–	Maximum number of points assumed to form a trunk	40	3.4.5.3
<i>overlap</i>	\mathbb{R}_0^+	m	Width of the overlapping area	5	3.4.2
<i>maxSampleSize</i>	\mathbb{R}^+	m	Maximum xy-size of a sample before clustering	10	3.4.2
<i>hwRel</i>	\mathbb{R}^+	–	Minimum ratio between z- and xy-range of a trunk	5.0/1.0	3.4.5.3
<i>minZRange</i>	\mathbb{R}_0^+	m	Minimum range of a trunk in z direction (height)	3.5	3.4.5.3
<i>groundCoverLevel</i> (Z_{GCV})	\mathbb{R}	m	Maximum height of ground-covering vegetation	1.0	3.4.3
<i>relCrowningHeight</i> (ρ_{CH})	$]0, 1[$	–	Assumed relative crowning height of the trees	0.45	3.4.3
<i>delta</i> (δ)	\mathbb{R}^+	m	Buffer radius of clustering algorithm.	0.85	3.4.4
<i>zBufferScale</i>	\mathbb{R}_0^+	–	Scale factor of z-axis for 3D clustering	0.05	3.4.4
<i>outlierlimit</i>	$]0, 1[$	–	Z-values above this quartile are assumed to be outliers	0.005	3.4.3
<i>MEPL</i>	\mathbb{R}_0^+	m	Expected maximum error per length of trunk	0.065	3.4.5.3
<i>maxZenith</i>	$]0, 90[$	°	Maximum assumed zenith angle of a trunk	8	3.4.5.3
<i>relOutliers</i>	$]0, 1[$	–	Expected maximum ratio of t_P and t_O vs. t_O	0.75	3.4.5.3
<i>uniformProb</i> (χ^2)	$]0, 1[$	–	Assumed minimum unique distribution of the z-values	0.002	3.5.5
<i>mergeBuffer</i>	\mathbb{R}_0^+	m	Assumed minimum distance between two trunks	1.5	3.4.6

Table 2: Model Parameters

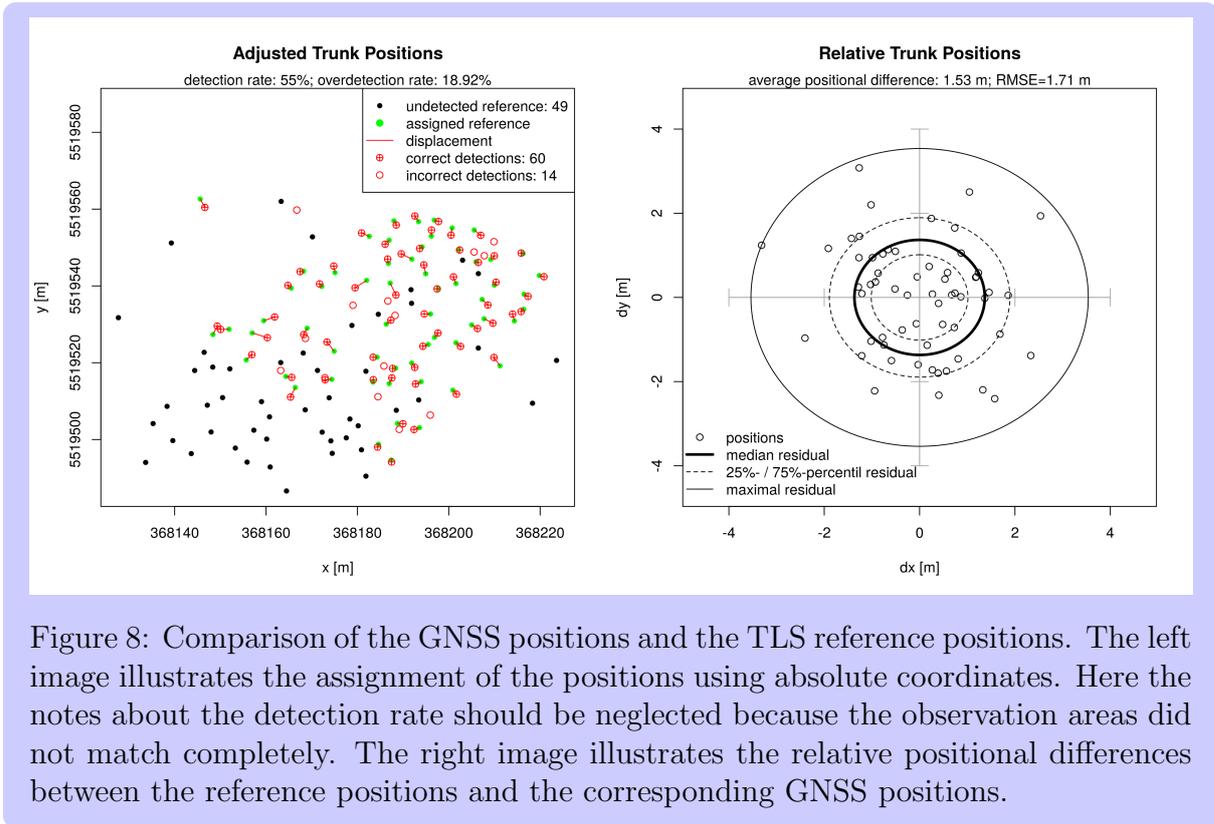
The measured GNSS positions and estimated TLS trunk positions were used as reference datasets. To minimize systematic errors both the GNSS and the TLS positions were adjusted to the detected positions using an affine point set registration.

A data adjustment was necessary, because it had to be assumed that the GNSS positions were inaccurate and shifted compared to the ALS data. In addition, the terrestrial measurements had to be adjusted to the ALS coordinates because they were not georeferenced.

To apply an affine point set registration of both reference datasets to the detected positions, some trunk positions were assigned manually using the *CloudCompare* [8] software. After this coarse registration a finer registration was done using a *python* script which is based on the theory presented in section 3.6.1. The optimisation was done iteratively by assigning each point to its pair-wise closest reference point and applying an affine transformation minimizing the residuals of these point pairs. This procedure was repeated until the RMSE changed only scarcely after one iteration. After the referencing of the point sets, the accuracy in positioning and the detection rate was evaluated.

A detected trunk position was assumed to correspond to its reference position when its distance was below 2 m for the *aTrunk* approach and—justified by the different positioning accuracies—4 m for the *watershed* approach. In addition, the selection was limited to the extent of the reference datasets. It should be noted that both reference datasets did not include all trunks because some trees were not measured or not detected by the slicing approach. Due to the large location residuals the GNSS reference dataset was rejected for further evaluation.

Figure 8 illustrates the position accuracy of the GNSS measurements compared to the slicing-based TLS trunk positions. The adjustment was done as described above while a maximum pair-wise distance of 4 m was assumed. An average positional difference of 1.53 m and an RMSE of 1.71 m occurred.



The *aTrunk* and *watershed* positions were combined to a merged dataset (called *combined* approach) to get information about the potential accuracy improvement by the *aTrunk* approach. In case of corresponding positions (distance below 4 m), the *aTrunk* positions were preferred.

3.6.1 Affine Point Pair Registration

The affine point pair registration uses already known point pairs to minimise the coordinate differences by applying an affine coordinate transformation. To transform points $\in \mathbb{R}^2$ to these new coordinates, a translation of $t_x \in \mathbb{R}$ in x-direction and $t_y \in \mathbb{R}$ in y-direction has to be performed. In addition, a rotation is applied using the angle $\varphi \in [0, \pi[$. So this affine coordinate transformation is done by equation 16. Here the matrix $\mathbf{P}^{n \times 2}$ corresponds to the $n \in \mathbb{N}$ coordinates and the matrix $\mathbf{A}^{n \times 2}$ corresponds to the transformed coordinates. The matrices $\mathbf{R}_\varphi^{2 \times 2}$ and $\mathbf{T}_{t_x t_y}^{n \times 2}$ apply the rotation and translation. To derive the required parameters t_x , t_y and φ , the equation system 17 has to be solved. The variables ϵ_x and $\epsilon_y \in \mathbb{R}$ correspond to the residuals. After the derivation of the matrices, \mathbf{R}_φ and $\mathbf{T}_{t_x t_y}$ can be used to apply this transformation to several datasets using equation 16.

$$\mathbf{A} = (\mathbf{R}_\varphi \cdot \mathbf{P}^T)^T + \mathbf{T}_{t_x t_y} \quad (16)$$

$$\begin{aligned}
 \begin{bmatrix} A_{x_1} & A_{y_1} \\ \vdots & \vdots \\ A_{x_n} & A_{y_n} \end{bmatrix} &= \left(\begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \cdot \begin{bmatrix} P_{x_1} & P_{y_1} \\ \vdots & \vdots \\ P_{x_n} & P_{y_n} \end{bmatrix}^T \right)^T + \begin{bmatrix} t_x & t_y \\ \vdots & \vdots \\ t_x & t_y \end{bmatrix} \\
 \begin{bmatrix} A_{x_1} & \dots & A_{x_n} \\ A_{y_1} & \dots & A_{y_n} \\ 1 & \dots & 1 \end{bmatrix} &= \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & \epsilon_x \\ -\sin(\varphi) & \cos(\varphi) & \epsilon_y \\ t_x & t_y & 1 \end{bmatrix} \cdot \begin{bmatrix} P_{x_1} & \dots & P_{x_n} \\ P_{y_1} & \dots & P_{y_n} \\ 1 & \dots & 1 \end{bmatrix} \quad (17)
 \end{aligned}$$

3.6.2 Rigid Point Set Registration

An automated *Rigid Point Set Registration* (cf. Jian and Vemuri [15] and Jian and Vemuri [16]) was rejected as an alternative to the manual pre-adjustment and affine point pair registration because this approach failed several times. This effect could be explained by the unfavourable spatial distribution of the trunk positions, which leads to a rare characteristic alignment of neighboured trunks. Nevertheless, figure 9 illustrates some results of the rigid point set registration approach.

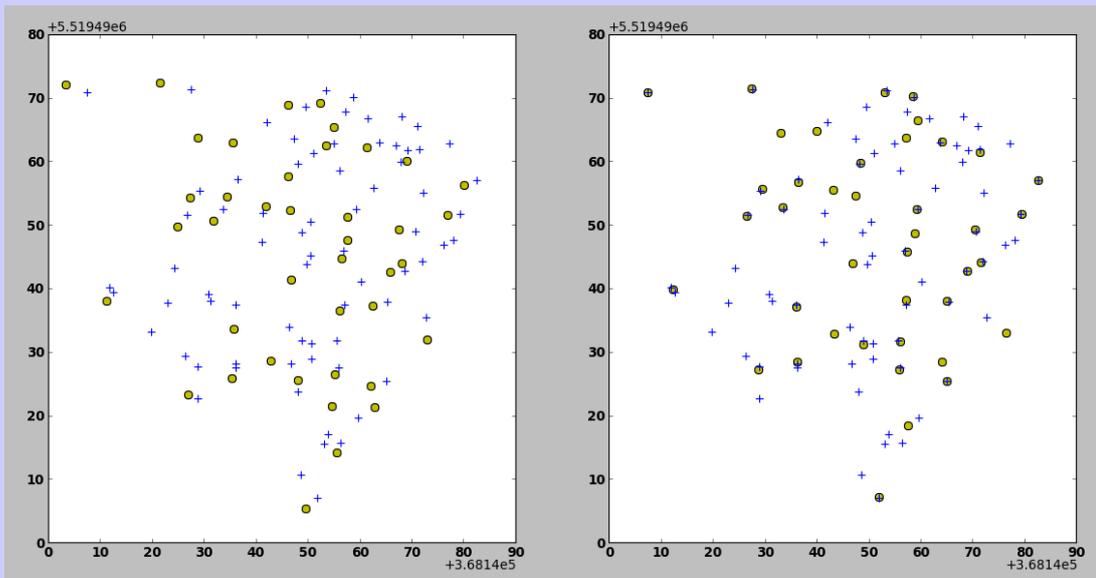


Figure 9: Rigid point set registration between detected positions (yellow dots) and reference positions (blue crosses) using the algorithm of Jian and Vemuri [16]. Before registration (left image) and after registration (right image).

3.7 Sensitivity Analysis

A sensitivity analysis was performed to derive an optimal parametrisation of the model. In doing so, a valid and conceivable parameter range was defined for each parameter. The approach was applied a few hundred times to the reference dataset

using a randomly selected parameter combination at each run. For that purpose the optimisation software of Bergstra et al. [1] was used. The TLS dataset served as reference for the evaluation of each modelling result. To reduce the number of necessary model runs the parameter ranges should be as narrow as possible. The first tests led to a conceptual categorisation of the parameters in three groups, which are summarised in table 3.

Table 3: Parameter Groups

Group 1	Group 2	Group 3
<i>minPoints</i>	<i>maxPoints</i>	<i>relCrowningHeight</i>
<i>minGrdPoints</i>	<i>overlap</i>	<i>delta</i>
<i>minZRange</i>	<i>maxSampleSize</i>	<i>MEPL</i>
<i>mergeBuffer</i>	<i>zBufferScale</i>	<i>uniformProb</i>
<i>maxZenith</i>	<i>hwRel</i>	<i>relOutliers</i>
<i>outlierlimit</i>	<i>groundCoverLevel</i>	

The first group contains all parameters which are defined by the quality requirements of the user. These values depend either on the point density of the input data or on general assumptions on the characteristics of the study area. The second group is mainly characterized by the effect on the overall validity of the model or by the effect of the parameters on the computation time. The third group contains parameters which are strongly affected by side effects between these parameters. These parameters are very sensitive to changes compared to the others. So the values of the parameters of Groups 1 and 2 were set to a constant value based on expert knowledge, while the parameters of Group 3 were used for the sensitivity analysis.

Figures 10 and 11 illustrate the modelling results of selected parameters of Group 3 using the detection rate d , and $OF(d, o, r)$ of equation 18 respectively as an objective function. The model was applied a few hundred times to the study area to derive these scatterplots. The rate of false detections o and the RMSE r were used.

$$OF(d, o, r) := \sqrt{(1 - d)^2 + (1 - o)^2 + r^2} \quad (18)$$

The point distribution of the scatterplots was used to identify optimal parameter ranges. A well-defined local maximum respectively minimum implies an optimal value for the corresponding parameter. To derive an optimal value semi-automatically the best 5% of the objective values were selected (green dots). The median value was assumed the optimal value. A parameter range was defined in a close range around the median (black dots), while red dots signify those points which are located in the optimal ranges for all parameters.

Such figures were used to limit the parameter ranges and to remove those parameters from the consideration which have a minor influence on the results. The final parametrisation was selected manually while whole-numbered values were preferred.

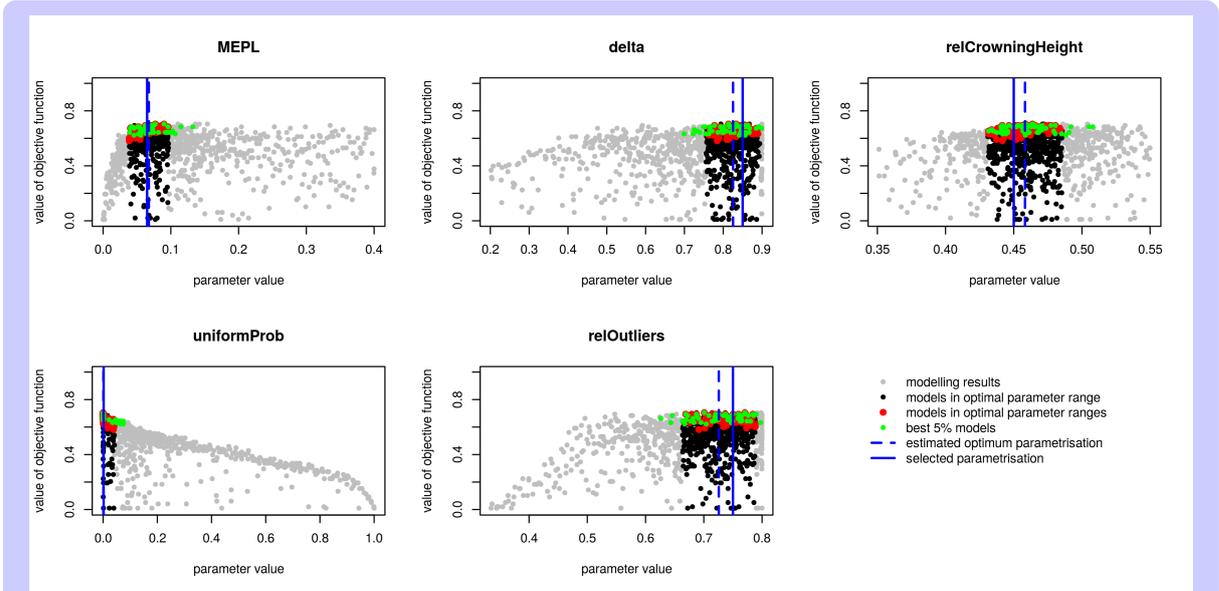


Figure 10: Visualisation of the sensitivity analysis using the detection rate as the objective function. A global maximum is considered the optimal value of the corresponding parameter.

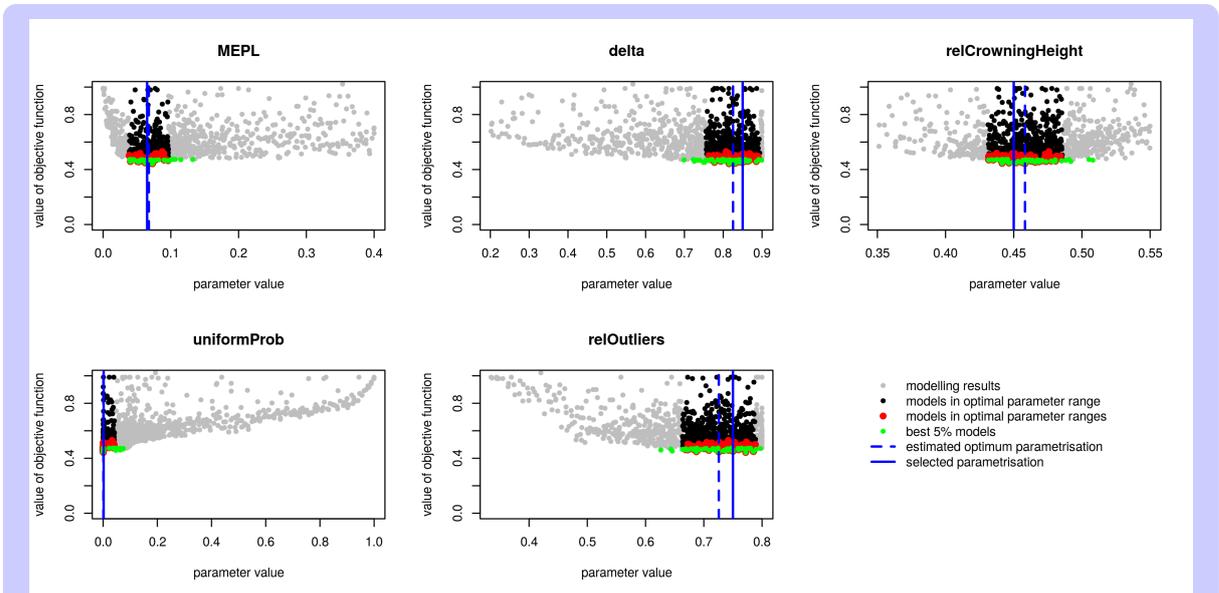


Figure 11: Visualisation of the sensitivity analysis using equation 18 as the objective function. A global minimum is considered the optimal value of the corresponding parameter.

4 Results

4.1 Evaluation

The comparison between the detected *aTrunk* models with the TLS positions resulted in an average positional difference of 0.32 m and an RMSE of 0.42 m (see figure 12). In total, a detection rate of about 69.7% was achieved, while about 8.4% of the detections were assumed to be false detections. In contrast, the local maxima-based *watershed* segmentation approach reached a detection rate of about 92.7%, while 12.4% of the detections were assumed to be incorrect. Since a clearly higher average positional difference of 1.09 m and an RMSE of 1.30 m occurred with the *watershed* approach, the *aTrunk* approach reached a remarkably high accuracy in positioning.

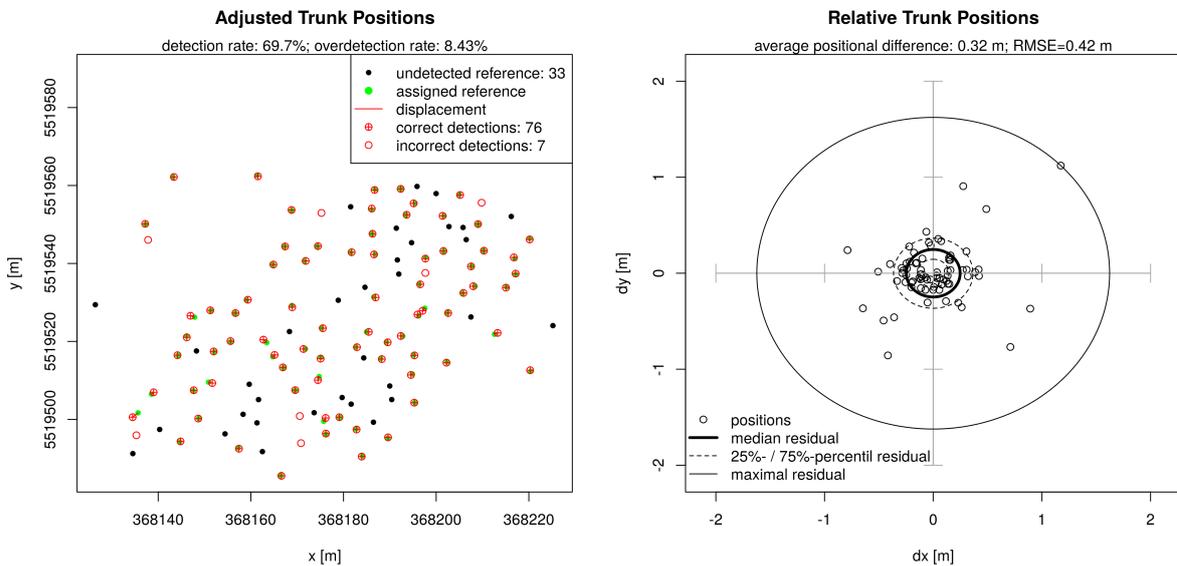


Figure 12: Evaluation of the detected *aTrunk* positions using the TLS positions as reference. The left image illustrates the assignment of the positions using absolute coordinates. The right image illustrates the relative positional differences between the reference positions and the corresponding detected positions.

Figures 13 and 14 illustrate the corresponding evaluation results of the *watershed* approach respectively the *combined* approach compared to the TLS positions.

The *combined* approach reached a significantly increased detection rate of about 97.2%, while about 15.9% of the positions were assumed to be false detections. An average positioning accuracy of 0.60 m and an RMSE of 0.83 m were reached.

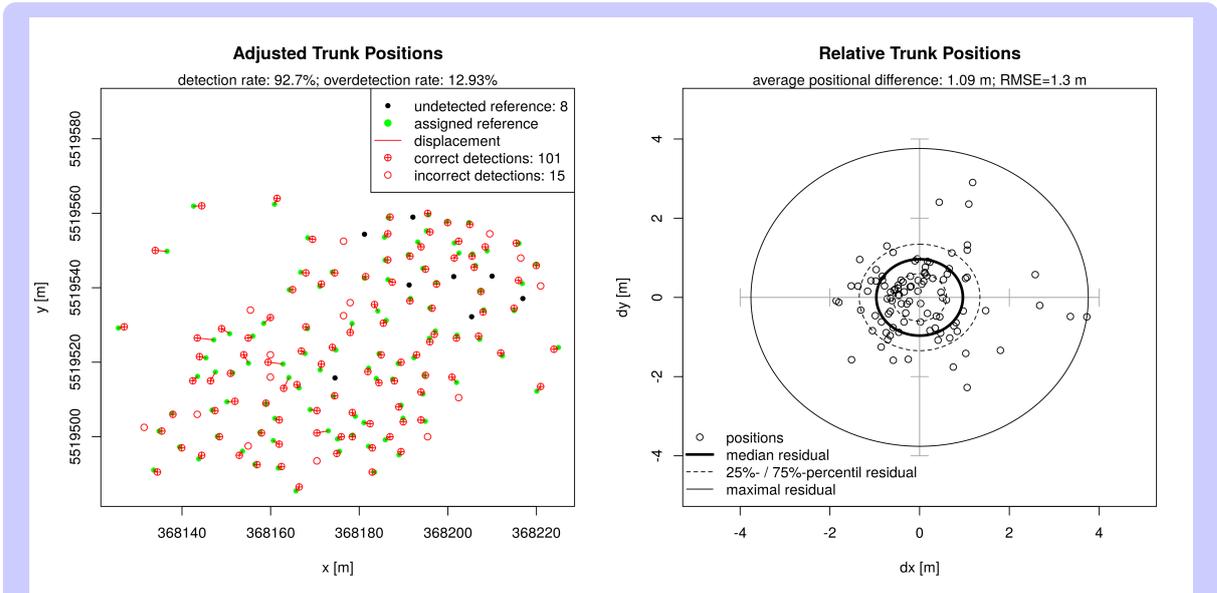


Figure 13: Evaluation of the detected *watershed* positions using the TLS positions as reference. The left image illustrates the assignment of the positions using absolute coordinates. The right image illustrates the relative positional differences between the reference positions and the corresponding detected positions.

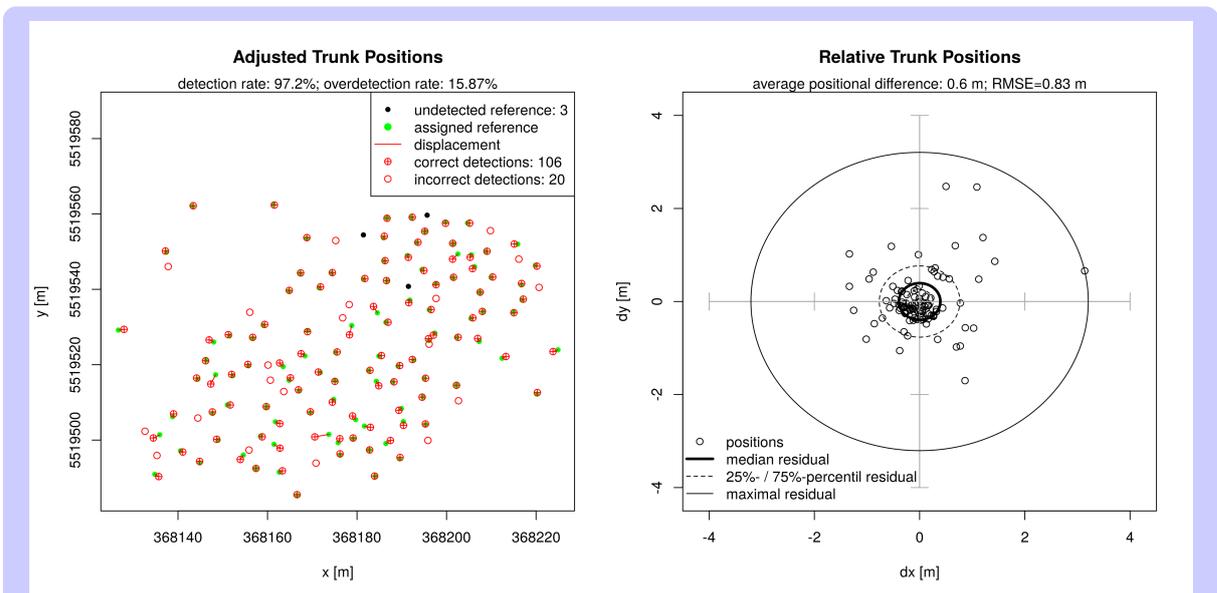


Figure 14: Evaluation of the detected *combined* approach positions using the TLS positions as reference. The left image illustrates the assignment of the positions using absolute coordinates. The right image illustrates the relative positional differences between the reference positions and the corresponding detected positions.

The boxplots of figure 15 illustrate the residual distribution of the *aTrunk*, *watershed* and *combined approach*. Clearly visible is the optimized positioning accuracy using both approaches. A *Mann-Whitney U-test* was applied to compare accuracies, because of the right-skewed distribution of the data. The p-values clearly close to zero imply that the positioning accuracies of both the *aTrunk* and the *combined* approach are significantly superior to the *watershed* approach.

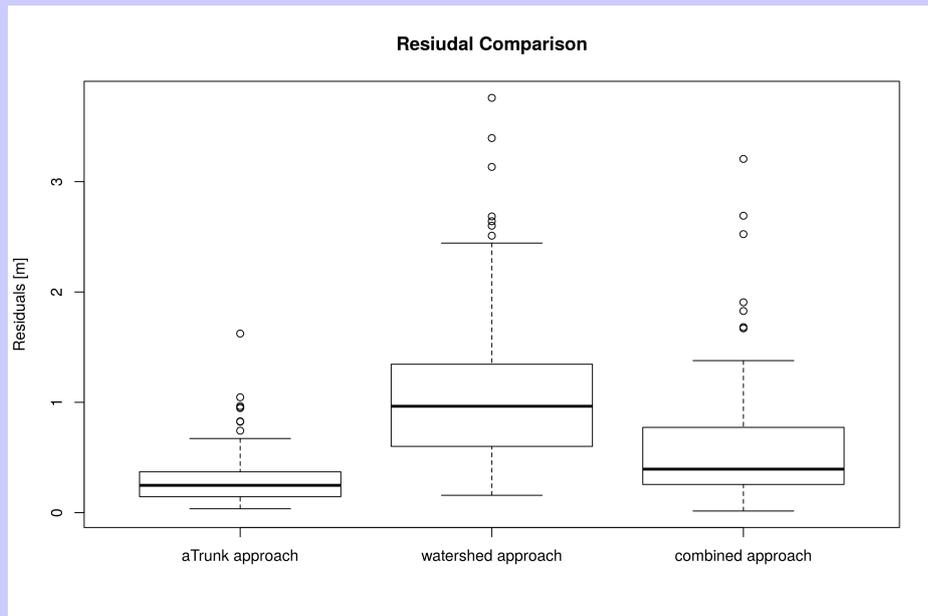


Figure 15: Boxplots of the positioning residuals of the *aTrunk*, *watershed* and *combined* approach.

For both reference datasets the relatively high overdetection rate is caused by the fragmented measurements across the study area. Moreover, many of the trunks at the edge of the study area were not recorded caused by missing TLS trunk detections (e.g. because of shade effects). By neglecting the false detections near the edges of the study area, a negligible proportion of over-detections remains.

It should be noted that the total accuracy in positioning can be overestimated, caused by the referencing step explained in section 3.6. Due to this strategy, the effect of a potential systematic shift in trunk positions, e.g. caused by uniform oblique trunks, is removed. Nevertheless the desired evaluation of the topology between the trunk positions remains. The relatively high detection rate of the *aTrunk* approach is favoured by the open forest structure of the study area.

4.2 Modelling Results

On a *Lenovo U310* with an *Intel Core i7-3517U* processor and 4 GB RAM, the algorithm required a computation time of about 12 seconds (0.4 seconds data loading and 11.1 seconds analysis) using the subset of the study area presented in section 3.2.1. For the

larger 1 km² sample the approach detected 4604 trunks in below 9 minutes (20 seconds data loading and 8 minutes analysis) with the same parametrisation. Figures 16 and 17 illustrate the modelling results of the study area and the larger 1 km² sample.

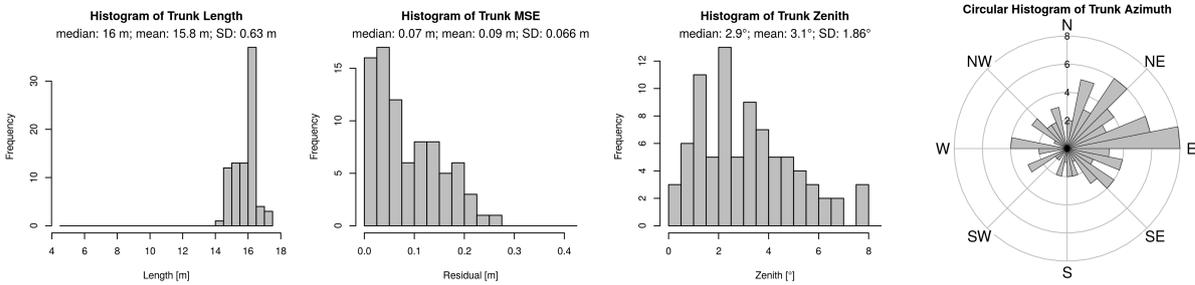


Figure 16: Modelling results of the study area.

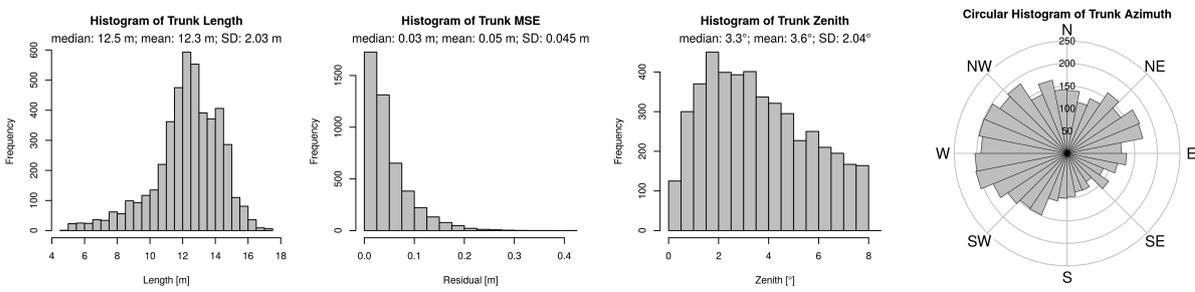


Figure 17: Modelling results of the 1 km² sample.

The study area is characterised by trees of a homogeneous great high, which results in long detected trunks, with an mean value of 16 m and a low standard deviation (SD) of 0.6 m. In comparison, the 1 km² sample consists of different areas with inhomogeneous tree heights. This results in a high standard deviation of about 2 m. The MSE distribution of the points associated with the trunk is right-skewed for both datasets. A median of below 7 cm is reached for the study area and of about 3 cm for the 1 km² sample.

The distribution of the trunk zenith implies that the *maxZenith*-parameter was chosen too small. Nevertheless, a right-skewed distribution is recognisable. In addition, a median of about 3° occurs, while most of the trunks clearly differ from a vertical orientation. The circular histogram of the trunk azimuth implies a preferred east-west or west-east orientation of the trunks. This effect could be caused by an actual process or by systematic measurement errors (e.g. caused by the flight direction). Under the assumption of an actually preferred orientation, this effect could be explained by a preferred growth in the dominant wind direction with a noticeable deviation from a vertical orientation.

Figures 18 and 19 illustrate the *aTrunk* detection results of the 1 km² sample.

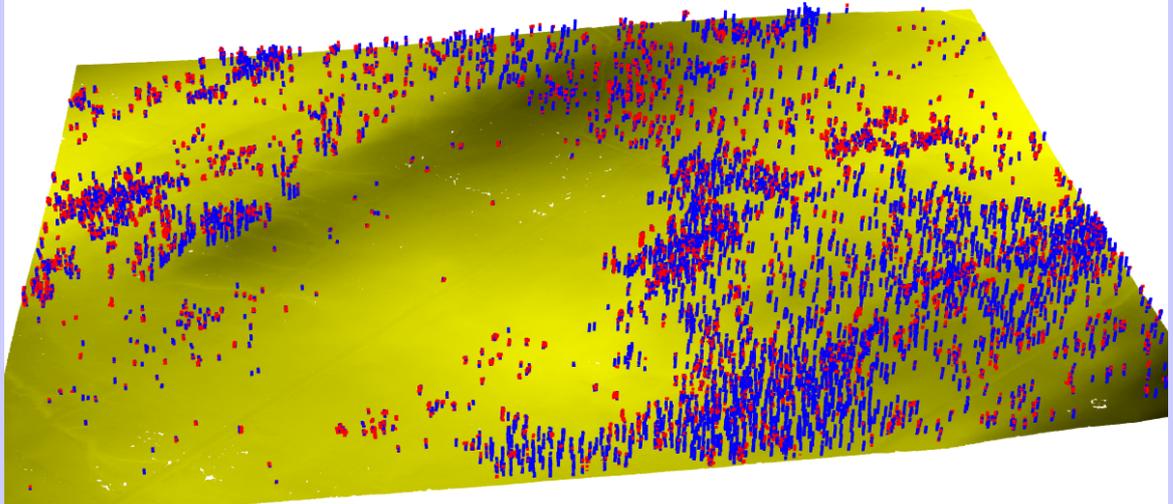


Figure 18: 3D-view of the 1 km² sample with detected trunks. Blue dots stand for trunk model supporting points, red dots for outliers.

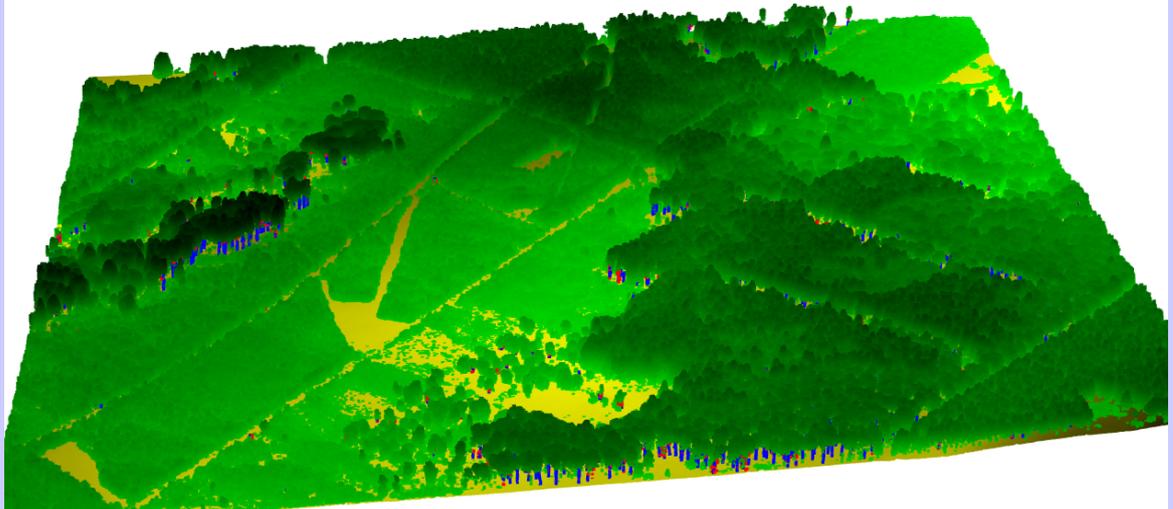


Figure 19: 3D-view of the 1 km² sample with detected trunks and vegetation cover

5 Discussion

The high positional accuracy of the *aTrunk* approach (with an average positional difference of 0.32 m and a RMSE of 0.42 m) reaches almost the magnitude of the ALS horizontal accuracy of about 5 to 15 cm (assumed by May and Toth [19]). The trunk detection

approach has shown clearly increased positioning results compared to the local-maxima-based *watershed* approach with a grid size of 0.5 m. The combination of the *aTrunk* and the *watershed* approach has shown an improvement of the average positional difference of 44.7% and an improvement of the RMSE of 36.5%. These observations support the results of Reitberger et al. [29], in which an improvement of up to 25% was mentioned. The inaccurate GNSS measurements imply that the trunk detection approach can be superior to ground-based GNSS positioning. Other raster-based approaches usually reach positioning accuracies above 0.5 m (cf. Kaartinen et al. [17]).

With a detection rate of about 70% the results of the *aTrunk* approach are outnumbered by the *watershed* approach with about 93%. Nevertheless, especially high reliability of the *aTrunk*'s detected trunks, can be seen as a benefit towards raster based approaches.

The derived detection rates of the *atrunk* approach seem to be comparable to other raster-based studies. Reitberger et al. [29], for instance mention a detection rate of about 60%, Chen et al. [5] 64%, Duncanson et al. [7] up to 70% and Hyyppä et al. [14] up to 83%. Depending on the tree neighbourhood, Kaartinen et al. [17] notice detection rates up to nearly 100% for isolated trees, while overlapping trees often cannot be identified.

It should be noted that the *atrunk* approach depends on a high point density in the trunk section. For dense vegetation, leafy trees or a lower point density, a decreased detection rate has to be expected, caused by the need of a registration of the trunk section and the assumption of a linear arrangement of points associated with the trunks. Therefore, it can be assumed that the detection rate also decreases for branched trees (like beech compared to spruce).

However, as mentioned above, the high reliability of trunks detected with a very high accuracy in positioning is an advantage over raster-based approaches. Especially the additional orientation information of the trunks increases the accuracy of detected trunk locations. The computation time of below 9 minutes for a 100 ha dataset on a commercial computer seems to be acceptable taking the additional benefit of the high quality information into consideration.

The computation time of the *aTrunk* seems to be clearly better than the computation time of the raster-based algorithm of Duncanson et al. [7], requiring a runtime of 1.5 h for a dataset of only 16.0 ha, with 32 CPUs and 2GB RAM per CPU. So, despite the higher amount of data of this point-based approach, the analysis can be very fast compared to raster-based approaches. This could be caused by the suitable data structure for mathematical analysis, compared to raster-based approaches which often rely on laborious window-moving techniques.

5.1 Potential Technical Improvements

The accurate prediction of the trunk position and orientation could be used to improve existing crown detection approaches, like Chen et al. [5] or Reitberger et al. [29]. Especially the approach of Zhou et al. [35] could be improved by choosing the trunk positions detected by this approach as initialization points. This strategy could optimise the detection rate and accuracy of tree crowns with a negligible additional

effort. In addition, it might be possible to achieve a better separability of overlapping trees in further studies.

The analysis effort of the *aTrunk* approach could be reduced by optimizing the clustering algorithm (see section 3.4.4.2) using a sweep-based approach, for instance, which could reduce the computation effort to $O(n \cdot \log_2(n))$. In addition, the plane terrain model of section 3.4.1.1 could be superseded by a polynomial of higher order to increase the accuracy. In section 3.5.3, it was mentioned that the estimated trunk position at ground level relies on the assumption of a flat ground. To optimize the trunk positioning accuracy, the intersection point between the trunk vector and the terrain model should be calculated rather than assuming a flat ground.

A first coarse sensitivity analysis was performed, which should be extended in further studies. To anticipate some results, the parameters *maxPoints*, *uniformProb*, *relCrowningHeight*, *delta* and *MEPL* seem to have the largest impact on the detection rate and positioning accuracy, as expected. In addition, the dependency of the point density on the modelling results should be analysed, because the ability to detect the trunks depends strongly on the number of recorded points associated with the trunks.

An additional validation of the derived trunk vectors will be performed in a further study, which will combine a larger number of terrestrial scans and tree models (for example using the approach of Raunonen et al. [28]) with ALS trunk detections.

5.2 Outlook

The derived information about the trunk orientation or distribution could be used for a large scale geostatistic analysis of forest characteristics. Here the underlying processes should be examined. Because of the high reliability of the *aTrunk* approach, the derived trunk positions can be used to optimize the tree identification of raster based approaches. In addition, the trunk lengths for example, could be used for an estimation of the available wood volume or as additional information for a species identification.

Kaartinen et al. [17] notice an overall dependency of the tree size on the accuracy of detected tree locations for different approaches, while “the taller a tree is, the better is the accuracy of location”. Under the assumption of a high point density in the trunk section, the point-based *atrunk* approach could reduce this effect.

6 Author Contributions

Sebastian Lamprecht as the main author developed and implemented the approach, led the data acquisition, analysed the data and is responsible for the content of the manuscript. Sandra Dotzler, Erik Haß and Johannes Stoffels took part in the validation data acquisition. Thomas Udelhoven and Johannes Stoffels initiated the study and cross-checked the manuscript.

7 Conflicts of Interest

The authors declare no conflict of interest.

References

- [1] James Bergstra, Dan Yamins, and David D Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *12th Python in Science Conference. (SCIPY 2013)*. SciPy, 2013.
- [2] Anne Bienert, Hans-Gerd Maas, and Steffen Scheller. Analysis of the information content of terrestrial laserscanner point clouds for the automatic determination of forest inventory parameters. In *Workshop on 3D Remote Sensing in Forestry*, pages 1–6, Viena, Feb 2006.
- [3] BMELV. *Aufnahmeanweisung für die dritte Bundeswaldinventur 2011-2012*, volume 2. Bundesministerium für Ernährung Landwirtschaft und Verbraucherschutz, Bonn, 2011.
- [4] BWaldG. Bundeswaldgesetz (bwaldg) vom 02.05.1975 i. d. f. vom 31.07.2010 bundeswaldgesetz vom 2. mai 1975 (bgbl. i s. 1037), das zuletzt durch artikel 1 des gesetzes vom 31. juli 2010 (bgbl. i s. 1050) geändert worden ist, §41a walderhebungen.
- [5] Qi Chen, Dennis Baldocchi, Peng Gong, and Maggl Kelly. Isolating individual trees in a savanna woodland using small footprint lidar data. *Photogrammetric Engineering and Remote Sensing*, 72(8):923–932, 2006.
- [6] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In Bernd Michaelis and Gerald Krell, editors, *Pattern Recognition*, number 2781 in Lecture Notes in Computer Science, pages 236–243. Springer Berlin Heidelberg, January 2003. ISBN 978-3-540-40861-1, 978-3-540-45243-0.
- [7] L. I. Duncanson, B. D. Cook, G. C. Hurtt, and R. O. Dubayah. An efficient, multi-layered crown delineation algorithm for mapping individual tree structure across multiple ecosystems. *Remote Sensing of Environment*, 2013. ISSN 0034-4257. doi: 10.1016/j.rse.2013.07.044.
- [8] EDF R&D. Cloudcompare – 3d point cloud and mesh processing software open source project. <http://www.cloudcompare.org/>, 2014. [Online; accessed 14-November-2014].
- [9] Curtis Edson and Michael G. Wing. Airborne light detection and ranging (LiDAR) for individual tree stem location, height, and biomass measurements. *Remote Sensing*, 3(11):2494–2528, November 2011. doi: 10.3390/rs3112494. URL <http://www.mdpi.com/2072-4292/3/11/2494>.
- [10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [11] FARO Europe GmbH. FARO® laser scanner photon 120/20. www.faroeurope.com/portal/htdocs/download.php?id=1794&type=DOC&SiteCatalyst=true, May 2010. [Online; accessed 13-September-2014].

- [12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692.
- [13] J. Holmgren, Å. Persson, and U. Söderman. Species identification of individual treesHolmgren by combining high resolution LiDAR data with multi-spectral images. *International Journal of Remote Sensing*, 29(5):1537–1552, 2008. ISSN 0143-1161. doi: 10.1080/01431160701736471.
- [14] Juha Hyyppä, Xiaowei Yu, Hannu Hyyppä, Mikko Vastaranta, Markus Holopainen, Antero Kukko, Harri Kaartinen, Anttoni Jaakkola, Matti Vaaja, Jarkko Koskinen, and Petteri Alho. Advances in forest inventory using airborne laser scanning. *Remote Sensing*, 4(5):1190–1207, May 2012. doi: 10.3390/rs4051190. URL <http://www.mdpi.com/2072-4292/4/5/1190>.
- [15] Bing Jian and Baba C. Vemuri. A robust algorithm for point set registration using mixture of Gaussians. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, pages 1246–1251, 2005.
- [16] Bing Jian and Baba C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1633–1645, 2011. URL <http://gmmreg.googlecode.com>.
- [17] Harri Kaartinen, Juha Hyyppä, Xiaowei Yu, Mikko Vastaranta, Hannu Hyyppä, Antero Kukko, Markus Holopainen, Christian Heipke, Manuela Hirschmugl, Felix Morsdorf, Erik Næsset, Juho Pitkänen, Sorin Popescu, Svein Solberg, Bernd Michael Wolf, and Jee-Cheng Wu. An international comparison of individual tree detection and extraction using airborne laser scanning. *Remote Sensing*, 4(4):950–974, March 2012. doi: 10.3390/rs4040950. URL <http://www.mdpi.com/2072-4292/4/4/950>.
- [18] Landesamt für Vermessung und Geobasisinformation Rheinland-Pfalz (LVer-mGeo). Luftbild rp basisdienst. http://www.geoportal.rlp.de/mapbender/php/wms.php?layer_id=30692&PHPSESSID=02dbaf5a20e411b1c46de1f8ef2a9cdd&REQUEST=GetCapabilities&VERSION=1.1.1&SERVICE=WMS, 2014. [Online; accessed 25-September-2014].
- [19] Nora Csanyi May and Charles K Toth. Point positioning accuracy of airborne lidar systems: A rigorous analysis. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36:107–111, 2007.
- [20] MDPI AG. Remote sensing – open access journal. <http://www.mdpi.com/journal/remotesensing>, 2015. [Online; accessed 10-February-2015].
- [21] Felix Morsdorf, Erich Meier, Benjamin Kötz, Klaus I. Itten, Matthias Dobbertin, and Britta Allgöwer. LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote Sensing of Environment*, 92(3):353–362, August 2004. ISSN 0034-4257. doi: 10.1016/j.rse.2004.05.013.

- [22] Open Source Geospatial Foundation OSGeo. Postgis. <http://postgis.net>, 2014. [Online; accessed 14-November-2014].
- [23] Open Source Geospatial Foundation OSGeo. Qgis – a free and open source geographic information system. <http://www.qgis.org/en/site/>, 2014. [Online; accessed 14-November-2014].
- [24] Asa Persson, Johan Holmgren, and Ulf Söderman. Detecting and measuring individual trees using an airborne laser scanner. *Photogrammetric engineering and remote sensing*, 68(9):925–932, 2002. ISSN 0099-1112.
- [25] The PostgreSQL Global Development Group PostgreSQL Group. Postgresql. <http://www.postgresql.org/>, 2014. [Online; accessed 14-November-2014].
- [26] Python Software Foundation. python. <https://www.python.org>, 2015. [Online; accessed 9-February-2015].
- [27] The R Foundation for Statistical Computing R Foundation. The r project for statistical computing. <http://www.r-project.org/>, 2014. [Online; accessed 14-November-2014].
- [28] Pasi Raumonon, Mikko Kaasalainen, Markku Åkerblom, Sanna Kaasalainen, Harri Kaartinen, Mikko Vastaranta, Markus Holopainen, Mathias Disney, and Philip Lewis. Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sensing*, 5(2):491–520, January 2013. doi: 10.3390/rs5020491. URL <http://www.mdpi.com/2072-4292/5/2/491>.
- [29] J. Reitberger, Cl. Schnörr, P. Krzystek, and U. Stilla. 3d segmentation of single trees exploiting full waveform LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):561–574, November 2009. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2009.04.002.
- [30] Timo Schmid and Ralf T. Münnich. Spatial robust small area estimation. *Statistical Papers*, 55(3):653–670, August 2014. ISSN 0932-5026, 1613-9798. doi: 10.1007/s00362-013-0517-y.
- [31] Dirk Tiede, Gregor Hochleitner, and Thomas Blaschke. A full gis-based workflow for tree identification and tree crown delineation using laser scanning. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36: 9–14, 2005.
- [32] Topcon Corporation. Hiper v – dual-frequency gnss receiver. http://www.topconpositioning.com/sites/default/files/HiPer_V_Broch_7010_2121_RevB_TF_sm.pdf, 2014. [Online; accessed 8-Dezember-2014].
- [33] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1–3):37–52, August 1987. ISSN 0169-7439. doi: 10.1016/0169-7439(87)80084-9.
- [34] Xiaowei Yu, Paula Litkey, Juha Hyypä, Markus Holopainen, and Mikko Vastaranta. Assessment of low density full-waveform airborne laser scanning for individual tree

- detection and tree species classification. *Forests*, 5(5):1011–1031, May 2014. doi: 10.3390/f5051011. URL <http://www.mdpi.com/1999-4907/5/5/1011>.
- [35] Jia Zhou, Christophe Proisy, Xavier Descombes, Ihssen Hedhli, Nicolas Barbier, Josiane Zerubia, J. P. Gastellu-Etchegorry, and Pierre Couteron. Tree crown detection in high resolution optical and lidar images of tropical forest. *Remote Sensing for Agriculture, Ecosystems, and Hydrology*, 7824(7824), 2010. doi: 10.1117/12.865068.
- [36] Hans Ole Ørka, Erik Næsset, and Ole Martin Bollandsås. Classifying species of individual trees by intensity and structure features derived from airborne laser scanner data. *Remote Sensing of Environment*, 113(6):1163–1174, June 2009. ISSN 0034-4257. doi: 10.1016/j.rse.2009.02.002.

Erklärung zur Masterarbeit

Hiermit erkläre ich, dass ich die Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe.

Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.

Trier, den 8. März 2015