University of Zurich^UZH

Morphological Traits in a Temperate Forest: Comparing UAV-LS and TLS Point Clouds

GEO 610: Master's Thesis

Author
Manuel Andreas Luck
08 920 050

Supervised by
Dr. Felix Morsdorf
Daniel Kükenbrink

Faculty representative:
Prof. Dr. Michael Schaepman

30.10.2018
Department of Geography, University of Zurich

Abstract

The impact of global change on biodiversity is of unprecedented scale. In regard of ongoing alteration of biodiversity researchers come up with multiple approaches aiming at decelerating its decrease. Therefore, revealing and monitoring locations of this alteration is fundamental. Monitoring forest areas is achieved by describing the functional diversity of plants which is embodied in their structural traits. An approach for estimating these structural traits is the application of quantitative structure models (QSM). These models describe the morphology of individual trees by quantitatively describing its topological, volumetric and geometric properties. Building a QSM requires a three-dimensional point cloud which can be acquired with Laser Scanning sensors. These sensors can be mounted on different platforms. To describe the three-dimensional structure of a forest, airborne (ALS) or terrestrial laser scanning (TLS) is common to be used. While TLS data stand out with very high density which enables us to create high-detail models of individual trees, ALS offers the mobility to acquire less detailed data over a much larger area as well as another perspective. Recent progress in unmanned aerial vehicle (UAV) technology allows for the installation of high-resolution laser scanners on board of small drones. UAV-LS surveys benefit from higher mobility in comparison to a terrestrial field study and higher point density than ALS missions.

In this study we try to implement a semi-automatic workflow to extract essential structural traits from LiDAR point clouds in a temperate mixed forest. The performance of a UAV laser scanning system is assessed by comparing the extracted variables with the ones retrieved from TLS point clouds as well as from conventional field survey data.

The approach includes four main steps. First the generation of a high-resolution digital terrain model (DTM) to separate the vegetation points from the ground points. Second the detection of trunk in the vegetation points. This was achieved by filtering the cloud and subsequently applying a “Nearest Neighbours Clustering”. The tree separation was obtained by the creation of a “Nearest Neighbours Graph” and the comparison of the shortest paths from each point in the Graph to each trunk. For the model creation we used the QSM approach.

The extracted diameters at breast height (DBH) from the TLS dataset mainly matched the field measurements (RMSE 0.1167 m). The modelling based on the UAV data was only working to a limited extent. This is due to the different point distribution based on the different view angles in the two datasets. Nevertheless, we were able to determine morphological features such as DBH (RMSE 0.3014 m) and tree height (2 m difference between UAV-LS and TLS) with reasonable accuracy. For the extraction of more complex morphological traits our QSM based on UAV-LS data were not sufficiently accurate.

To put it in a nutshell, we can conclude that there is high potential in the data derived from UAV-LS surveys, nonetheless a QSM generated based on those data does not reach the same accuracy as the one generated from TLS data. The progress in sensor and UAV technology as well as improvements in the implemented approach will certainly enable the extraction of more complex morphological traits from UAV-LS data in the near future.

Zusammenfassung

Globaler Wandel beeinflusst die Biodiversität in beispielslosem Ausmasse. Um der weltweiten Änderung der Biodiversität entgegenzuwirken, arbeiten Wissenschaftler an verschiedenen Methoden. Grundlage für all diese ist die Identifizierung und Untersuchung von Gebieten, die einer solchen Veränderung der Biodiversität unterliegen wie unter anderem auch Wälder. Um deren Biodiversität zu beschreiben, betrachtet man unter anderem die funktionelle Diversität, welche in strukturellen Merkmalen der Bäume widergespiegelt wird. Eine Herangehensweise an die Waldkartierung ist die Erstellung sogenannter „Quantitative Structure Models" (QSM) aus lasergenerierten Daten. Diese Modelle stellen die Morphologie einzelner Bäume dar, indem sie ihre topologischen, volumetrischen und geometrischen Eigenschaften quantitativ beschreiben. Die Erstellung solcher Modelle wird ermöglicht durch den Einsatz von sowohl terrestrischen (engl. TLS) als auch an Flugobjekte gebunden (engl. ALS) Laserscannern. Diese liefern Daten, welche für die Darstellung eines Waldes in seiner drei-dimensionalen Struktur verwendet werden können.

Einerseits können durch die hohe Punktdichte der TLS Daten Modelle einzelner Bäume mit hoher Detailgenauigkeit erstellt werden. Dahingegen ermöglicht die Mobilität der ALS die Gewinnung von Daten weitaus grösserer Gebiete, wenn auch weniger detailliert. Dank der jüngsten technologischen Fortschritte ist es möglich, hochauflösende Laserscanner auf unbemannten Flugobjekten (engl. UAV) zu montieren. Studien mit diesen profitieren von der Kombination erhöhter Mobilität im Vergleich zu terrestrischen Studien mit erhöhter Punktdichte im Vergleich zu luftgebundenen Missionen. Damit einher geht die Erweiterung von Forschungsgebieten zu geringen Kosten.

Das Ziel dieser Studie ist, grundlegende strukturelle Merkmale aus den zuvor vorgestellten Punktwolken am Beispiel eines temperierten Mischwaldes semi-automatisiert zu extrahieren. Die Leistung des UAV-Lasersystems soll dann durch den Vergleich der gewonnen Variablen mit den Äquivalenten von TLS und konventionellen Feldstudien beurteilt werden.

Unsere Herangehensweise lässt sich in vier Hauptschritte gliedern. Zuerst wurden hochauflösende Digital Terrain Modelle (DTM) generiert, wobei Vegetations- von

Bodenpunkte getrennt werden. Nachfolgend wurden aus ersteren die Stämme detektiert. Hierfür wurde die Punktwolke gefiltert und anschliessend ein „Nearest Neighbour Clustering“ angewandt. Die Baumtrennung konnte durch das Erstellen eines “Nearest Neighbour Graph“ und den Vergleich der kürzesten Distanzen eines jeden Punktes im Graphen zu jedem Stamm erreicht werden. Für die Modell-Generierung wurde dann die zuvor vorgestellte QSM-Methode angewandt.

Die aus den TLS-Daten gewonnenen Durchmesser auf Brusthöhe (engl. DBH) stimmten überwiegend mit den Messungen der Feldstudie überein (RMSE 0.1167). Dahingegen gelang die Modellierung anhand der UAV-Daten nur eingeschränkt. Dies lässt sich auf die verschiedenen Punktverteilungen aufgrund der unterschiedlichen Blickwinkel in den beiden Datensätzen zurückführen. Nichtsdestotrotz war es uns möglich, morphologische Eigenschaften wie DBH (RMSE 0.3014) und Baumhöhe (2 Meter Unterschied zwischen UAV-LS und TLS) mit angemessener Genauigkeit zu bestimmen. Für die Extrahierung komplexerer morphologischer Eigenschaften war die Detailgenauigkeit unserer anhand von UAV-Daten erstellten Modelle jedoch nicht ausreichend.

Zusammenfasend lässt sich festhalten, dass die Daten aus UAV-LS Studien grosses Potential bergen. Jedoch wird mit einem QSM aus diesen Daten nicht dieselbe Detailgenauigkeit erreicht wie mit einem QSM aus TLS-Daten. Sowohl die Fortschritte in Sensor- und UAV-Technologie als auch weitere Verbesserungen der hier angewandten Methode bieten aber Grund zur Annahme, dass die Extrahierung komplexer morphologischer Eigenschaften aus UAV-Daten in naher Zukunft möglich sein wird.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

LiDAR	Light Detection and Ranging
TLS	Terrestrial Laser Scanning
UAV-LS	Unmanned Aerial Vehicle Laser Scanning
UAV	Unmanned Aerial Vehicle
DTM	Digital Terrain Model
DSM	Digital Surface Model
CHM	Canopy Height Model
PC	Point Cloud
GP	Ground Point
VP	Vegetation Point
QSM	Quantitative Structure Model

1 Introduction

Biodiversity refers to the abundance and distributions of and interactions between genotypes, species, communities, ecosystems and biomes (Leadley et al., 2010). Several studies have shown an alarming decrease in biodiversity (Bellard et al., 2012; Butchart et al., 2010; Cardinale et al., 2012). Malanson and Alftine (2016) state that "biodiversity is impacted by humans in many ways, including habitat degradation and loss, fragmentation, introduction of invasive species and pollution." In order to counteract this decrease, revealing locations of change is crucial. Therefore, various methods have been developed to map biodiversity and its changes (Bowker, 2000; Pereira et al., 2013; Williams et al., 1997). Some of these methods use remote sensing approaches, which essentially means to detect changes in data derived from different sensors (e.g. Image Spectroscopy, Laser Scanning and Radar). The surveys are carried out on different spatial resolutions, correlating to the three major levels of biodiversity defined by Jaisankar et al. (2018): diversity of Ecosystems, diversity of species and the genetic diversity within a species. Yamasaki et al. (2017) describe the connection between the genetical approach and the remote sensing approach on biodiversity. The genetic approach describes the effect of genetic variance within a species and its impact on the morphological appearance of single individuals. This morphology can be derived on different levels of complexity from remote sensing data. Numerous studies were conducted from either spaceborne missions (Bergen et al., 2009), airborne surveys (Mura et al., 2015; Sverdrup-Thygeson et al., 2016; Vihervaara et al., 2015; Vosselman et al., 2011) or terrestrial surveys (Calders et al., 2015; Ehbrecht et al., 2017, 2016; Hilker et al., 2012; Liang et al., 2016; Liang and Hyyppä, 2013; Zhao et al., 2015).

This thesis focuses on a survey technology called Light Detection and Ranging (LiDAR). LiDAR is a laser-based technology to measure distances to targets in a three-dimensional space by measuring the time a laser pulse travels till it is reflected by a target and travels back to the source. Scanning an area with such a method produces high resolution three-dimensional point clouds. Those are applied in research areas such as earth system sciences (geography, geology, meteorology etc.), biology, architecture and navigation. In this thesis we study the benefits, LiDAR surveys offer for forest inventory. Those benefits largely depend on the platform the laser scanner is mounted on. The more the distance between platform and area of interest increases, the more the complexity of the gained point cloud decreases. However, airborne and spaceborne missions cover a much larger area and do not suffer from accessibility issues. The combination of those benefits and the higher accuracy of terrestrial laser scanning surveys can be achieved by using small unmanned aerial vehicles (UAV) as a platform. With regard to this study we aim at investigating the difference between UAV-LS and TLS (Terrestrial Laser Scanning) data. Doing so, we acquire two different LiDAR point clouds. One is obtained by mounting the sensor on a tripod, which has to be carried manually from each scan position to the next (TLS). By contrast, the other point cloud derives from mounting the sensor on an unmanned aerial vehicle (UAV-LS). Both approaches hold specific advantages and disadvantages shown in the table.

Table 1-1: Advantages & Disadvantages of the two LiDAR systems.

TLS	UAV-LS
+ Precision	+ Area coverage
+ Point Density	+ Acquisition time
- longer Survey	- Precision
- Terrain accessibility	- Occlusion of trunk area
- Occlusion of Canopy	- max. flight duration

Comparing the data and investigating its differences, biodiversity of forests serves as a framework. Putting a forests richness in figures is achieved on multiple spatial scales. While one can observe the species richness and distribution, one can also describe the intraspecies variability representing the genetic diversity hence regarding biodiversity on a more complex level. Surveys covering larger areas using satellite or airborne data are rarely able to describe individuals (Yamasaki et al., 2017). Investigating biodiversity, Cianciaruso et al. (2009) highlight the importance of intraspecific variability. Studying the phenotype allows for measuring this variability without analysing DNA sequences on a molecular level. The phenotype is displayed in the morphology and hence the structure of the individual. There are various studies about changes in morphology within a species due to adaptation to its particular environment (Barbeito et al., 2017; Pretzsch, 2014; Rötheli et al., 2012).

In order to describe the morphology of individuals with remote sensing data, high density point clouds are required. Those are obtained by using the previously introduced combination of LiDAR and UAV technology mounting a small footprint sensor on a UAV (Morsdorf et al., 2017).

The gained point clouds can then be used to extract morphological traits of individual trees. The complexity of a trait is variable. An example of a basic trait is the diameter at breast height (DBH) which describes the trunk diameter at a height of 1.3 meters above ground. More complex measures are the covered area of the crown or the wood volume of branch compartments. To describe such complex traits, recent studies make use of a method called quantitative structure model.

A Quantitative Structure Model (QSM) is a model of an object that quantitatively describes its topological, volumetric and geometric properties. The model is built from multiple primitive blocks fitted on the objects structure (Raumonen et al., 2013). This thesis applies the QSM implemented by Raumonen et al. (2013). They use cylinders as building blocks for the QSM due to the robust results for estimating diameters, lengths, directions, angles and volumes. Various studies (Calders et al., 2015; Hackenberg et al., 2015b; Kaasalainen et al., 2014; Raumonen et al., 2013) on this topic are based on this approach.

Calders et al. (2015) managed to reconstruct a one-hectare point cloud, acquired with a TLS, with an altered version of the QSM to calculate a radiative transfer model of the forest. Bienert et al. (2018) used a vehicle-based mobile laser scanning system for the data acquisition and then discussed its advantages and disadvantages. The main advantage is the higher mobility during the survey of corridor-like areas, but its biggest handicap is the requirements due to the vehicle. In their study they used a van which required forest paths with a clear height of 2.8 meters. They further propose to mount the sensor on different ground vehicles to enhance the mobility in a forest.

Increasing the mobility is also possible by going aerial. In our knowledge no studies were conducted using UAV-LS data as input for a QSM, especially not for areas covered with dense vegetation. This thesis introduces a method to detect single trees in a forest using Light Detection and Ranging data acquired by an unmanned aerial vehicle and tests the potential of the data for the QSM approaches. The results are to be compared to those using the data from a Terrestrial Laser Scanning survey and from conventional field surveys. Therefore, we work on the following research questions:

“Do the point cloud distributions of UAV-LS fit the requirement of tree detection as well as TLS point clouds?”

“Which traits can be derived from the UAV-LS point cloud?”

The research questions will be discussed in the 6th chapter.

2 Materials

2.1 Study Area

The study is applied on a test site located on the southern slope of the Laegern (2'669'779 m E, 1'259'019 m N, CH1903+ / LV95), a small hill approximately 15 km north-west of Zurich, Switzerland. The ridge has an elevation of 866 m, which corresponds to an approximate prominence of 425 m and is covered by a temperate mixed forest with European beech and Norway spruce as dominant species (Eugster et al., 2007; Schneider et al., 2014).

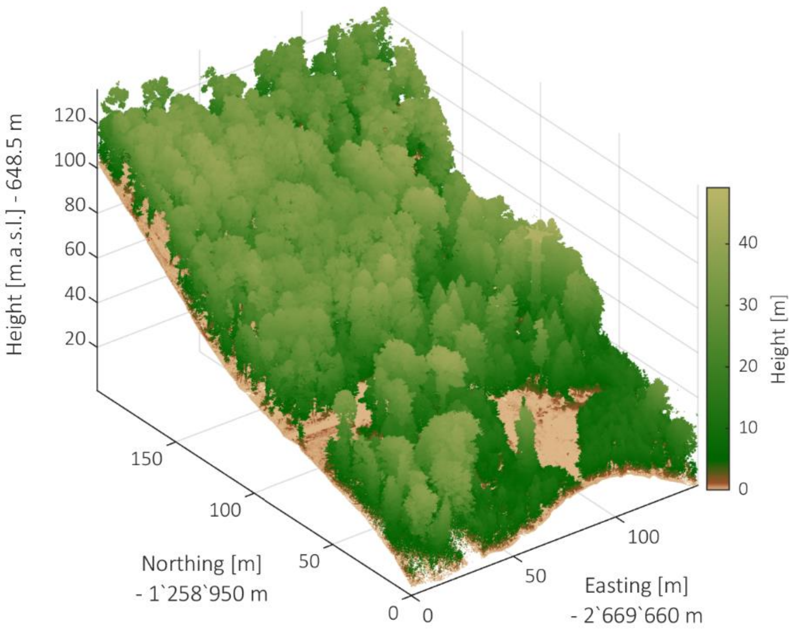


Figure 2-1: UAV-LS point cloud of the study area.

2.2 Data

For this study there are three datasets available. Two datasets contain Light Detection and Ranging data and one is a field survey conducted by A. Rudow from the ETH Zurich. The UAV-LiDAR data serve as test data and the other two sets are for validation purposes.

2.2.1 LiDAR

2.2.1.1 Unmanned Aerial Vehicle Laser Scanner Data

The UAV-LS data spread over an 140x200m area located between 2’669’660-2’669’800 m Easting and 1’258’950-1’259’150 m Northing (CH1903+ / LV95). It was acquired by the aeroscout GmbH using a Riegl VUX-1UAV during the leaf off season. The sensor has an accuracy and precision of 25mm with a beam divergence of 0.5 mrad and operates in the near infrared. The flight acquisition was done on an 80m nominal altitude above ground level at a cruising speed of 4m/s. This setting leaded to a point density of approximately 230pts/m^2 for each flight strip (Morsdorf et al., 2017). The resulting point cloud consists of 107’016’349 points. For each point there is an Easting and Northing coordinate in the swiss coordinate system (CH1903+ / LV95) as well as a height above sea and laser return intensity. The height varies between approximately 650 and 700 m and the intensity. In order to be able to compare the intensity, we normalized the pulse intensity between 0 and 1 for TLS as well as UAV-LS data. There is no decrease in point density towards the edges of the study area. Yet, one can recognize a slight structure from the overflight (figure 2-2). The stems as well as areas with dense branches or needles show an increase in intensity.

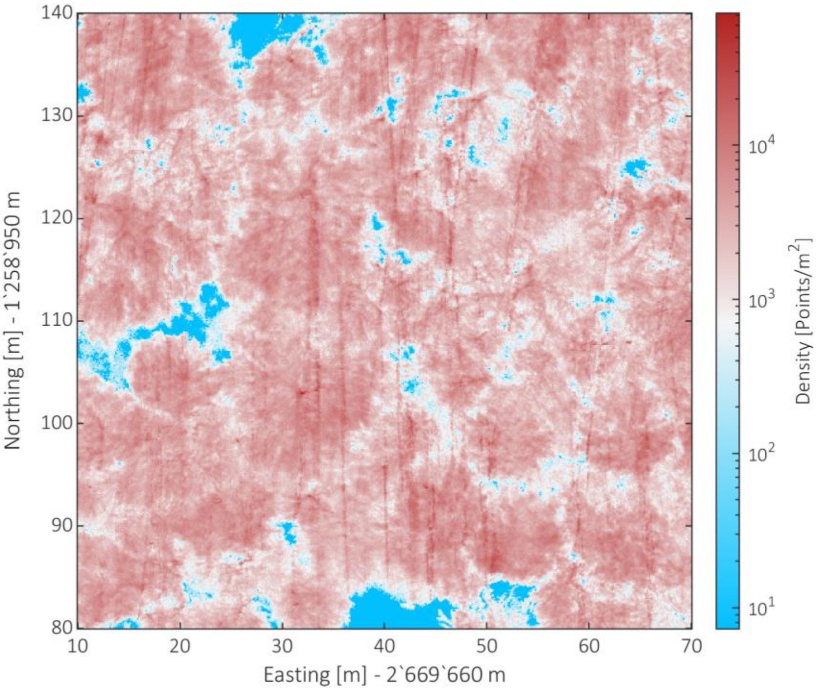


Figure 2-2: Intensity within a transect of the UAV-LS point cloud.

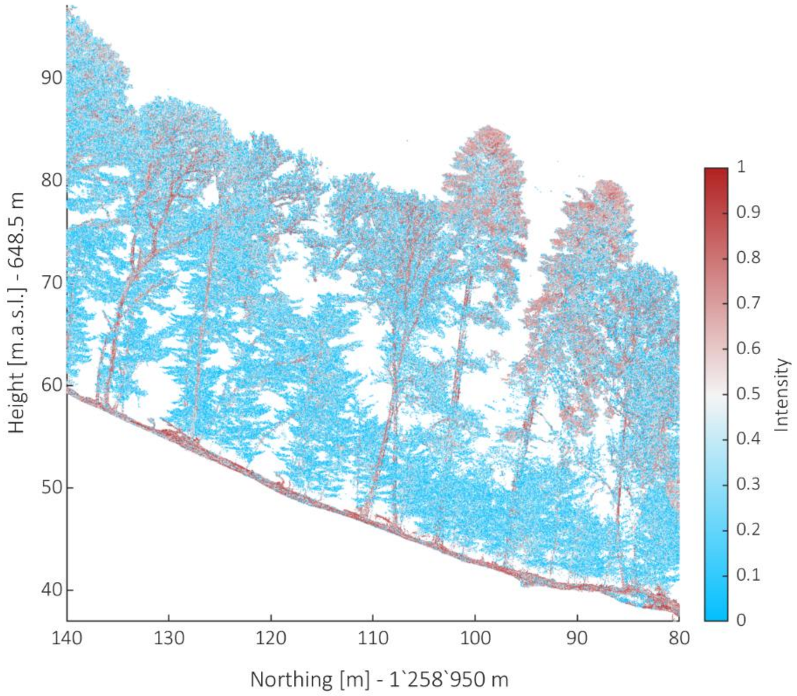


Figure 2-3: Point density of the UAV-LS point cloud. Points in each pillar extrapolated on a square meter.

2.2.1.2 Terrestrial Laser Scanner Data

The TLS data spreads over an 80x80m area located in the mid-west of the UAV-LS coverage, 2’669’660-2’669’740 m Easting and 1’259’020-1’259’100 m Northing. A Riegl VZ-1000 was used for the survey. On a total of 20 scan locations 40 scans were taken. As the TLS Riegl instrument has a horizontal field of view of 360° and a vertical field of view of 100°, two scans for each location were acquired in order to cover the whole hemisphere. For the corregistration of the scans approximately 50 reflecting targets were distributed in the area. The corregistration was done using the RiSCAN PRO software. The resulting point cloud consists of 105’587’740 points, each with local coordinates (x,y,z) and intensity. The point density strongly decreases towards the edges of the research area (figure 2-5) due to the positions of the scans.

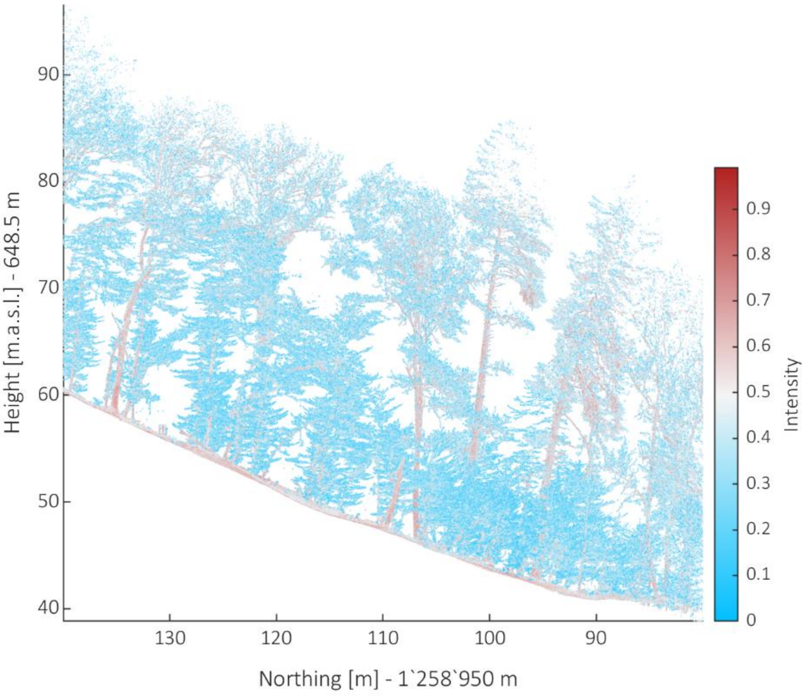


Figure 2-4: Intensity within a transect of the TLS point cloud.

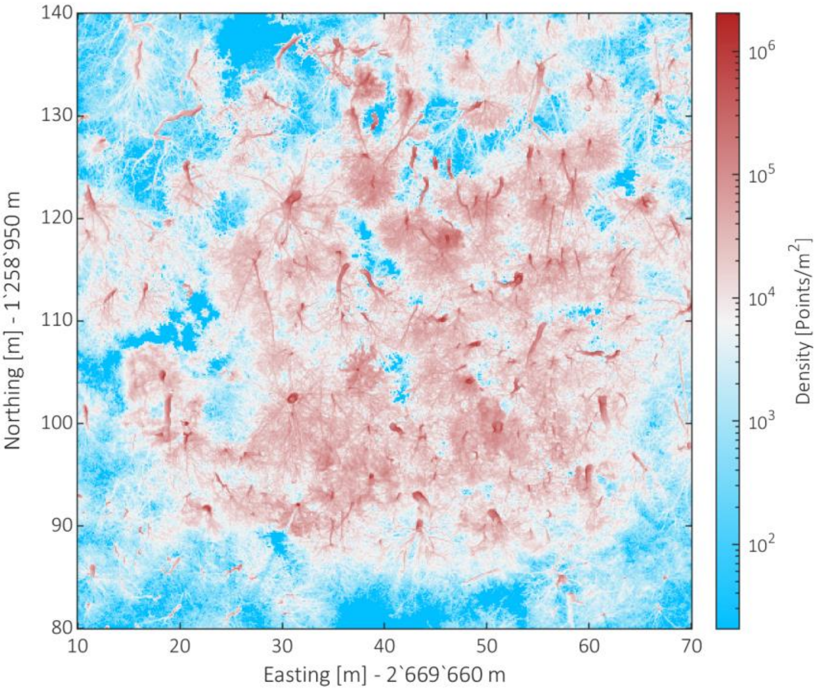


Figure 2-5: Point density of the TLS point cloud. Points in each pillar extrapolated on a square meter.

2.2.2 Field survey

The field survey was conducted during spring/summer 2018. It covers the same area, as the one of the UAV-LS dataset. A. Rudow and his team used a previously created stem position map, extracted from the UAV-LS data using a first version of the stem detection algorithm, for orientation. The positions contain estimated coordinates, diameter at breast height (DBH), leaning angle, ratio between vertical and horizontal standard deviation and an id.

The aim of the field survey was to map all trees with a DBH higher than 0.15 m, determine the species and measure the DBH. Thus, we received a table containing the following measures:

Table 2-1: Field survey measures.

ID	For identification. Tree detected which were not detected by the algorithm have an ID ≥ 1000
Easting	First Dimension (CH1903+ / LV95)
Northing	Second Dimension
Height	Third Dimension, only available in the extracted data.
DBH	Estimated from UAV-LS data, only available in the extracted data.
Angle	Estimated from UAV-LS data, only available in the extracted data.
Ratio	Estimated from UAV-LS data, only available in the extracted data.
DBH_real	Field measurement
Species	Field measurement
Memo	Additional comments

The collected dataset describes 724 trees consisting of 12 species. Main species are beech (fagus sylvatica and carpinus betulus) followed by different kinds of acer (acer platanoides, acer pseudoplatanus and acer campestre). The diameter at breast height varies between our threshold of 0.15 m and a maximum of 1.09 m. The histogram peaks at a DBH of 0.17 m as seen in figure 2-6. The distribution of the different species can be seen in figure2-7 and the trunk positions in figure 2-8.

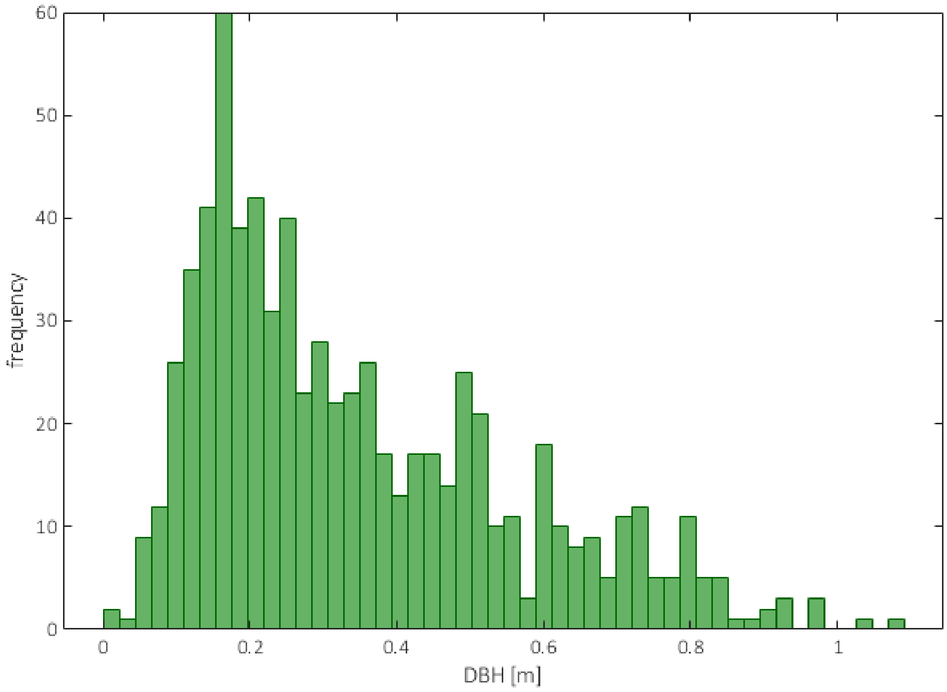


Figure 2-6: Histogram of the Diameter at Breast Height measured by A. Rudow during the field study.

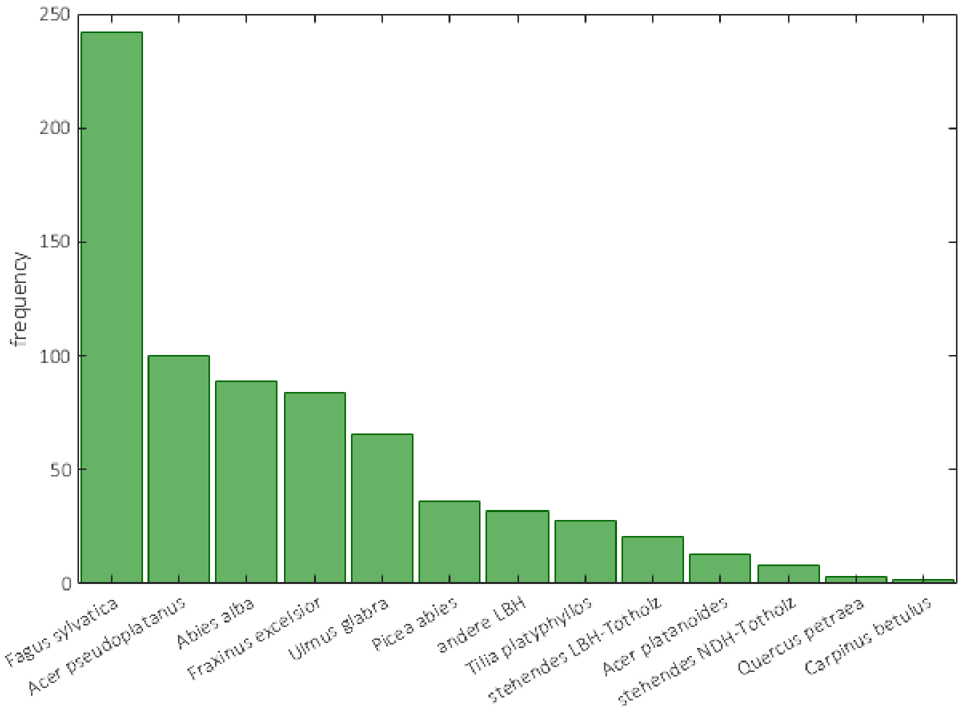


Figure 2-7: Histogram of Species described by A. Rudow during the field study.

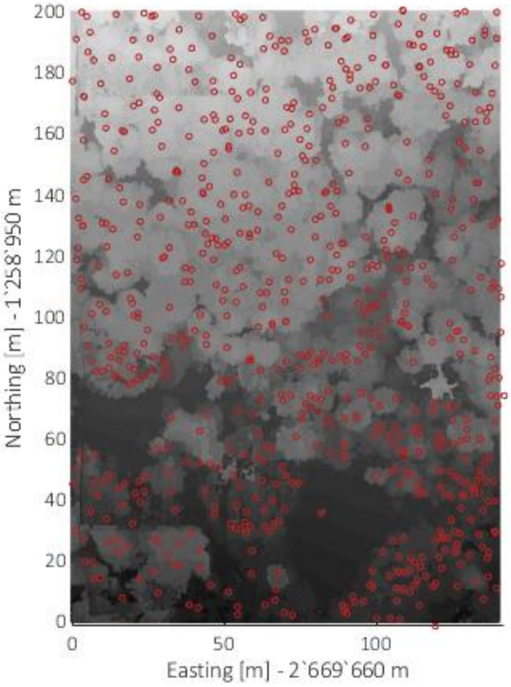


Figure 2-8: Tree Positions localized by A. Rudow during the field study (red circles) display on top of a Canopy Height Model.

2.3 Software

For the coregistration of the LiDAR scans, the RIEGL RiScanPro software was used and all further computations were implemented in MATLAB 2018a.

3 Methods

The implemented approach can be divided into five main parts. First, the pre-processing of the single point clouds. This consists of normalization of the intensity values, adjustment of the extend of the area of interest and the generation of a digital terrain model and a voxel creation. Second part is the stem detection. In this part we try to detect the different stems in the point cloud only using filtering and clustering. In the third part we separate the single trees using the stems locations from the second part. This is done by building a nearest neighbours graph and calculating the shortest path from each point to the surrounding stem locations. Having the trees separated, we use the QSM algorithm implemented by Raumonen et al (2013) to construct the models of the trees. Last part is to run a tree measurement extraction for traits such as DBH or tree height. This part is meant as backup for the case, that the QSM approach does not work with the UAV-LS dataset.

After these steps, we are able to compare the quantitative structure models and tree measures gained from either terrestrial and airborne data. The DBH can be verified by the measurements of the field study.

3.1 Pre-Processing

3.1.1 Digital Terrain Model Creation

A crucial step for this study is the separation of the point cloud into vegetation points and ground points. A way to achieve such a separation is to create a Digital Terrain Model (DTM) and classify all points within a specific distance of the DTM as ground points. The more accurate the DTM, the more accurate the separation. To achieve a high accuracy, the method created in this study uses a quad tree approach. Quad trees are often used in storing and displaying image data.

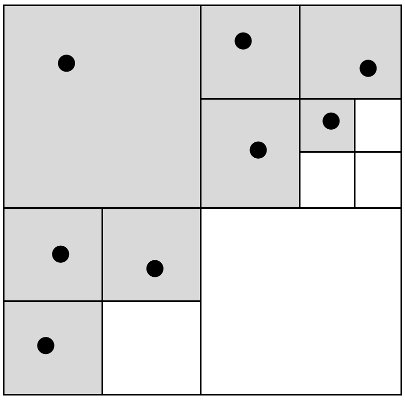


Figure 3-1: Quad tree storage approach. Simplified representation of the changing resolutions.

Broadly speaking the resolution in parts with a high amount of details is better than in parts with little information. Same idea can be used in the DTM generation. Being able to create a finer grid directly corresponds to the point density in the ground area. Meaning if the density of points representing ground is higher, we are able to use a finer grid. The point density and therefore the possibility to fill a finer grid is higher in regions with a vegetation density low enough to allow the LiDAR rays to reach the ground, than in regions where the ground is occluded by dense vegetation. Being able to vary the resolution spatially, enables us to reach the highest possible resolution for each region. A more detailed description of the approach can be found in the supplementary material.

3.1.2 Voxel Creation

To reduce the amount of points and therefore enable the computation to be executed on different kinds of workstations we implement a voxel grid. We use the cells generated by the DTM creation algorithm and apply an additional height binning of 0.1m. This is done by calculating a histogram for the z-values of all points contained in each cell. The final voxel points result from the means (x-, y-, z-mean) of the coordinates and the maximal intensity of all points in each height bin. Additionally, we add an indexing to each voxel- and vegetation point aiming at applying the tree classification based on the voxel points to the vegetation data.

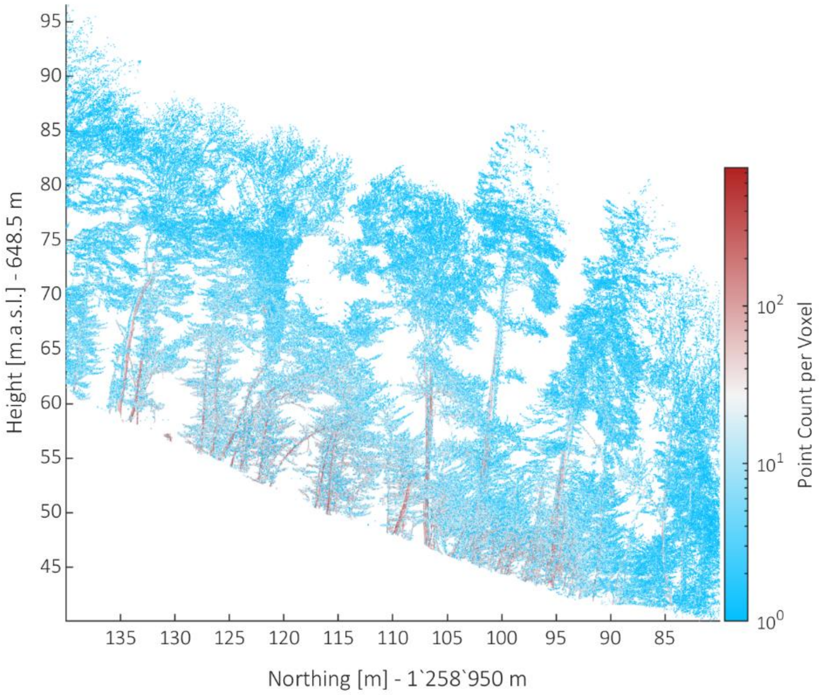


Figure 3-2: TLS voxel points in a 10mx60m transect. Voxel points standing for multiple vegetation points are coloured in red.

3.2 Stem Detection

3.2.1 Filtering

To separate the single trees in the point cloud we start by detecting their stems. On the one hand we need to reduce the amount of points to a size that makes it possible to differentiate the stems and on the other hand we don't want to reduce the point density within the stems itself too much. To achieve this task, the vegetation point cloud (VP) is filtered three times. Due to the different intensity and height distributions in the TLS and UAV-LS point clouds we need to adapt the parameters for each dataset separately.

Equation 1: Filtering of vegetation voxel points.

$$\mathrm{VP}_{filtered} = VP(VP_z > min_z \;\&\; VP_z < max_z \;\&\; VP_{int} > min_{int} \;\&\; VP_{DB} > min_{DB})$$

First, we only use points in between a height interval ($\min_z$ & $\max_z$). Thus, we exclude small understory vegetation and the ramified branches of the tree crowns. In case of the UAV-LS data we use a rather large area (1-9m) as the point density decreases towards the ground due to higher distance to the sensor and stronger occlusion effects. The density of the TLS point cloud is above average in the height of the stub of the tree and we therefore are able to work with a smaller interval (1-3m) without missing any trunk.

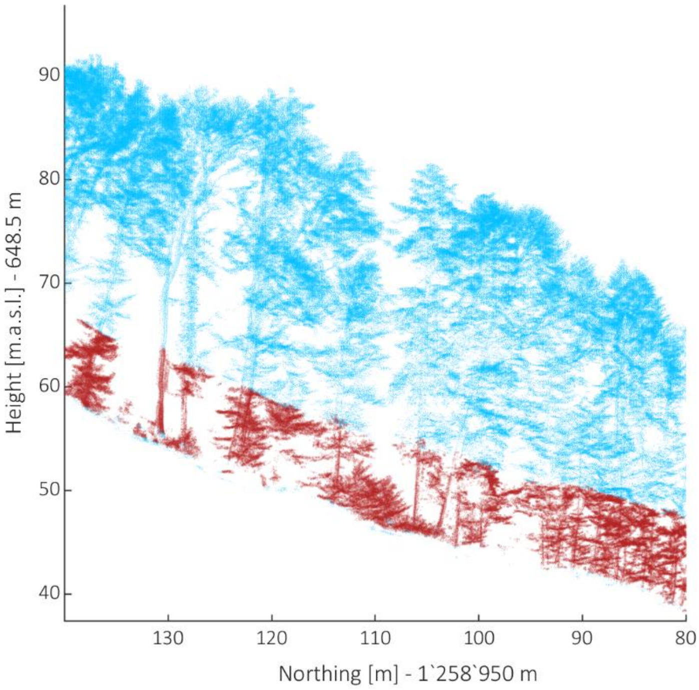


Figure 3-3: Height filter. Cutting a height interval between two thresholds (red).

Second, all points with an intensity below a fraction (min_{int}) of the maximal intensity are excluded. This step reduces the amount of points representing targets reflecting the beam with a low intensity, such as small objects or objects with a lower density than wood. We use a slightly higher threshold for the UAV-LS data (0.6) than for the TLS data (0.5). This is due to the taller height interval which leads to more branches connecting individual trunks. Having a higher intensity threshold reduce this problem.

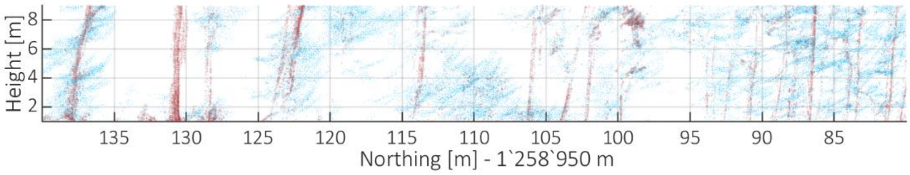


Figure 3-4: Intensity filter. Removing points with intensity below a specific threshold (blue).

And third, we run a density filter over the remaining points. With this filter we remove noise and small targets, with a number of neighbours lower than the threshold (min_{db}), from the point cloud. As we don't want to affect structural elements, we used a very low threshold (10 points) for both datasets.

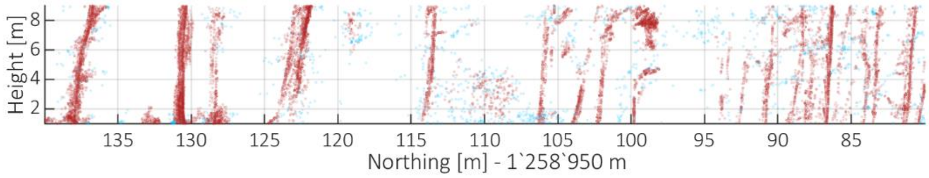


Figure 3-5: Density filter. Filters all points with too little neighbours in a specific distance (blue).

With these three filters applied, we end up with points mainly representing the tree trunks in this specific height interval.

3.2.2 Clustering

In the next step we identify clusters connected by a maximal distance. This is done by building a graph with all the points of the filtered point cloud. To be able to build a graph we created an adjacency list, which consist of all neighbours within the accepted distance (0.2m). As we are looking for the main stems of the trees, we assume that we are looking for vertical clusters in the point cloud. Aiming at giving more weight to vertical clusters, the z value is divided by a factor of 4. The searched clusters are then the connected compartments of the graph. As not every cluster represent stems, we filter the clusters by their size, their lowest point, their overall height and a ratio of their vertical and horizontal standard deviation.

The minimal cluster size is set to a low value (20 points) to remove noise and reduce the workload required for the subsequent step. The maximal value for the lowest point depends on the lower threshold from the height filtering. Only looking for clusters representing stems, we assume that the lowest point in a stem-cluster matches the lower height threshold of the previous height filtering. The minimal cluster height depends on the height filtering as well. In this case we assume that the expand of a tree cluster ranges over at least a third of the height interval. The remaining clusters are than filtered by the ratio between vertical and horizontal standard deviation (≥1).

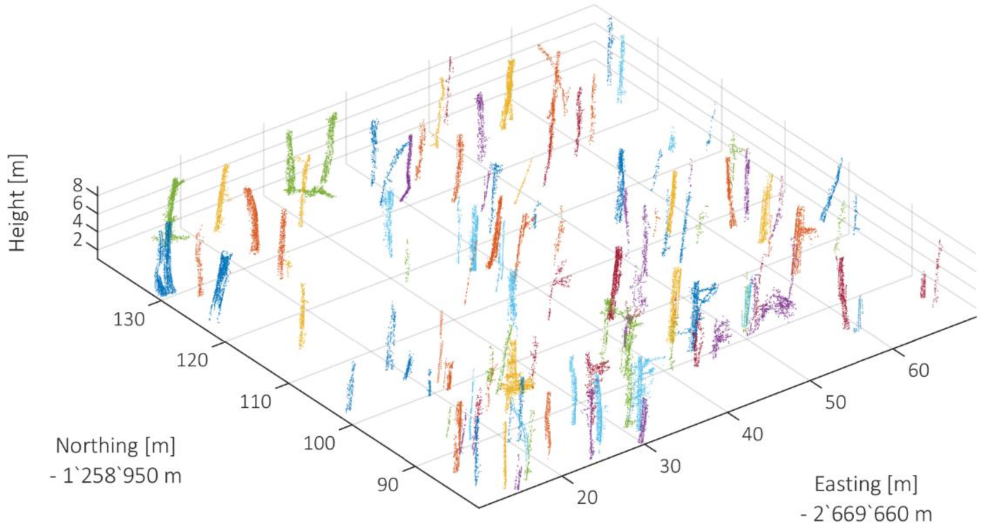


Figure 3-6: Detected and filtered clusters.

3.2.3 Stem Enhancement

To get denser stems, the stem clusters are enhanced with all points from the vegetation cloud, which are within a maximal distance (0.1m) to the points of the clusters. Like this we include for example points with low intensity which are also a part of the stem.

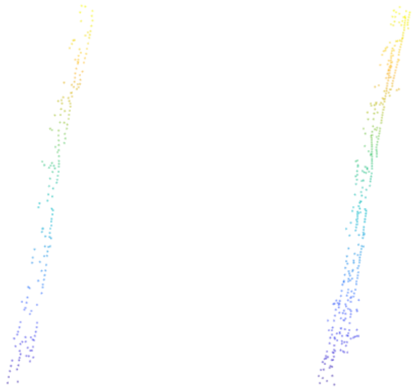


Figure 3-7: Enhanced Stems. Left the filtered version, right the enhanced.

3.2.4 Calculating Stem Measures

The resulting clusters from the stem detection are used to calculate some common measures like diameter at breast height (DBH), tilt, stem position and a ratio between the vertical and the horizontal standard deviation. To calculate the diameter, tilt and stem position, the cluster is divided into three height compartments. We calculated the mean coordinates of all points in the compartments to estimate the tree position at each height. The ratio between the vertical and the horizontal standard deviation is calculated to get a measure for the geometrical distribution of the cluster.

Equation 2: Ratio between horizontal and vertical standard deviation.

$$ratio_{v/h} = \frac{\mathrm{std(Height)}}{\mathrm{std}(Easting, Northing)}$$

3.3 Tree Separation

3.3.1 Stem Subsets

Before we continue with the tree separation, we create another subset of the vegetation points, referred to “no understory”. This time we take all points higher than the stem cloud and filter them by intensity and density. Thus, when combined with the stem clouds, we receive all vegetation points needed for the tree separation. In order to reduce the workload during the graph creation one cloud create subsets around each stem and use those as input for the following graph creation.

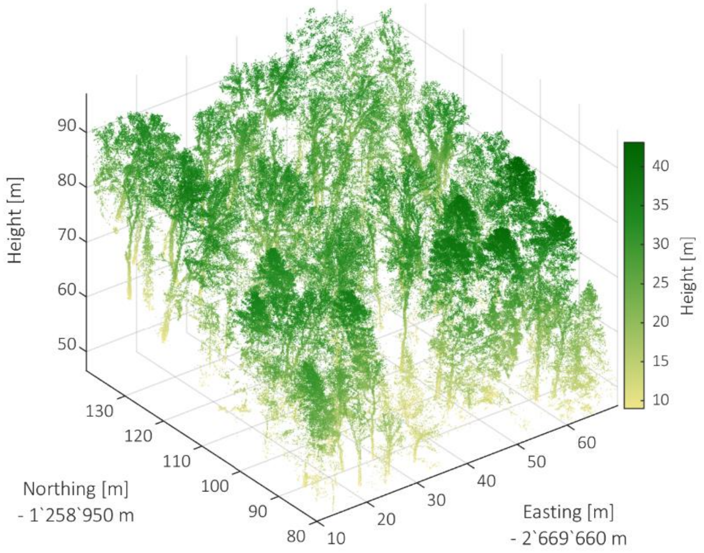


Figure 3-8: Vegetation points above threshold.

3.3.2 Graph Creation and Tree Separation

The graph creation is achieved in a way similar to the one used in the stem detection part. We now use the recently gained subsets as input clouds, yet the z value remains unchanged and the maximal distance between the points is considerably bigger (TLS: 1.5m, UAV: 1m). The distances, which later determine the edge weights, are then transformed into a logarithmic scale. Hence, shorter distances between points are even more favoured in the following shortest path calculation.

Equation 3: Creating logarithmic edge weights.

$$Edgeweight = 10^{1+Distance}$$

The tree separation itself is achieved by a shortest path comparison. We compare all shortest paths from each point to all stem locations in the subset area. Each point then gets assigned to the stem that has the shortest path. Having all the trees separated we enhance the trees in a way similar to the one used in the stem enhancement step. The graph creation and shortest path calculation is shown in the supplementary materials.

3.4 Quantitative Structure Model

To run the QSM algorithm it is important to have a dense point cloud of a tree. Therefore, we apply a similar point cloud enhancement as in the stem enhancement chapter. This time we take the separated trees from the previous step as one part of the input and the original vegetation point cloud as the second part.

In order to create the quantitative structure models of our trees we apply the QSM creation function (Raumonen et al., 2013) on those of our separated trees exceeding the minimal height of 10m. The code used for this will be shown in the supplementary materials under the subsection code.

3.5 Tree Measures

Beside the models we try to detect simple tree measures as diameter at breast height, canopy height, trunk position, canopy position, canopy offset and covered area for each tree. We extract the DBH with two different approaches. One works with the mean distance (Easting, Northing) of all points within a height interval of 1-2m to the centre of the trunk. The other approach applies the cylinder fitting function of the MATLAB point cloud package on the same points. The canopy height is estimated by the maximal height extent of the separated tree. The trunk position is either the centre of the cylinder model resulting from the cylinder fitting process, or the mean position (Easting, Northing) of the trunk points. The canopy position is the mean position (Easting, Northing) of all points within the top meter. Thus, the canopy offset is the difference in Easting and Northing between trunk and canopy position. Detailed code for extracting those measures can be found in the supplementary materials chapter.

4 Results

4.1 Pre-Processing

The implemented quadtree approach enables us to create a high resolution DTM using both UAV-LS as well as TLS data. Ten iterations are made resulting in a 1024x1024 grid with a resolution of 78.2x78.2 square millimetre. Due to the detailed DTM generation, the separation of the point cloud into vegetation points and grounds is easily achieved. The following table shows the obtained amount of vegetation and ground points for each dataset. Additionally, we are able to calculate precise above ground measures for each point which would be less detailed with a coarser DTM.

Table 4-1: Point count: Vegetation- and ground points.

	UAV-LS	TLS
Ground Points	20’499’533	41’277’349
Vegetation Points	86’516’414	64’307’630

To minimize the influence of the unequal point density in the TLS point cloud, we decide to reduce the study area to a 60x60m area, located in the middle of the point cloud. Thus, the vegetation points are reduced to an amount of 59’746’672 points in the TLS data and to 11’995’690 points in the UAV-LS data. The results of the DTM generation are described in further detail in the supplementary material.

As a result of the voxel creation, the amount of vegetation points is reduced as shown in the following table. Furthermore, we are able to remove voxel points representing too little vegetation data.

Table 4-2: Voxel count: Amount of voxel vegetation points in the UAV-LS and TLS data.

	UAV-LS	TLS
Amount of voxel points	5’615’099	3’417’251

4.2 Stem Detection

In total, we are able to detect 132 stem clusters in the UAV dataset and 279 stem clusters in the TLS dataset compared to 77 trees counted and localized in the field survey dataset. All filtering parameters and the resulting amounts of points are shown in table 4-3. The number of clusters for each step is shown in table 4-4.

Table 4-3: Filter parameters and the resulting point count.

Filter	UAV		TLS	
Height	1-9 m	770’883	1-3 m	231’656
Intensity	0.6	62’484	0.5	39’611
Point Density	10 P	52’902	10 P	37’229

Table 4-4: Number of clusters in each step.

	UAV	TLS
No filter	4424	1218
Big Clusters	486	342
Stem Clusters	132	279

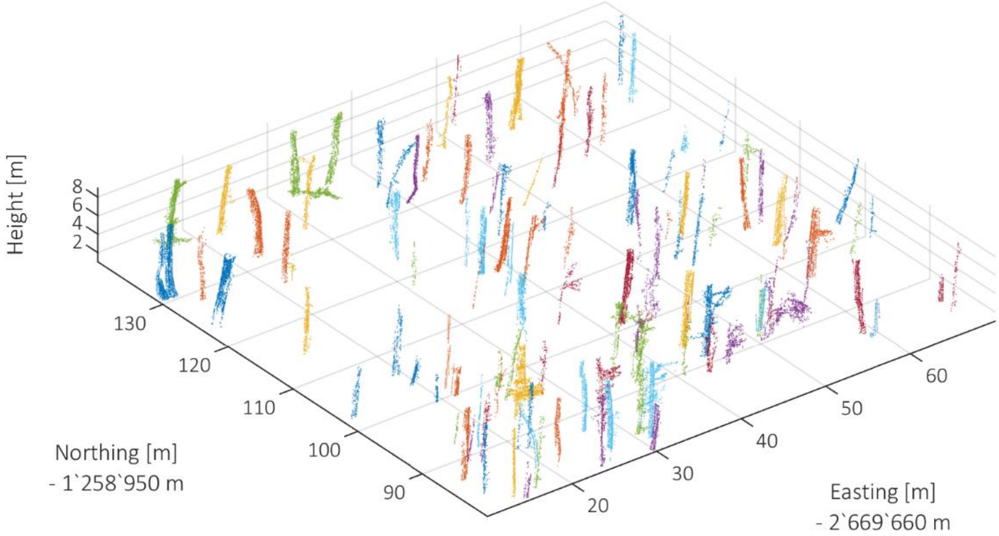


Figure 4-1: Clustered voxel points representing tree trunks.

These Clusters were than enhanced and the stem measures extracted. The stem measures obtained at this stage are only used for orientation purposes during the field survey.

4.3 Tree Separation

After reducing the study area to a 60x60m square and adding a voxel grid, the stem subsets became unnecessary as we are able to run the graph creation on the whole dataset at once. Running the stem subset creation step is important if the computational power is not capable of supporting the size of the point cloud.

In case of the UAV-LS data we are able to separate 111 different objects out of the dataset. Whereas in the TLS data we receive 211 objects. The difference is mainly due to small trees or shrub detected as stems in the TLS dataset. In figure 4-2 trees higher than 10m are displayed. Reducing the number of trees in this way results in 71 objects in the UAV-LS data and 141 objects in the TLS data. Whether the smaller trees should be recognized is to be determined by the survey.

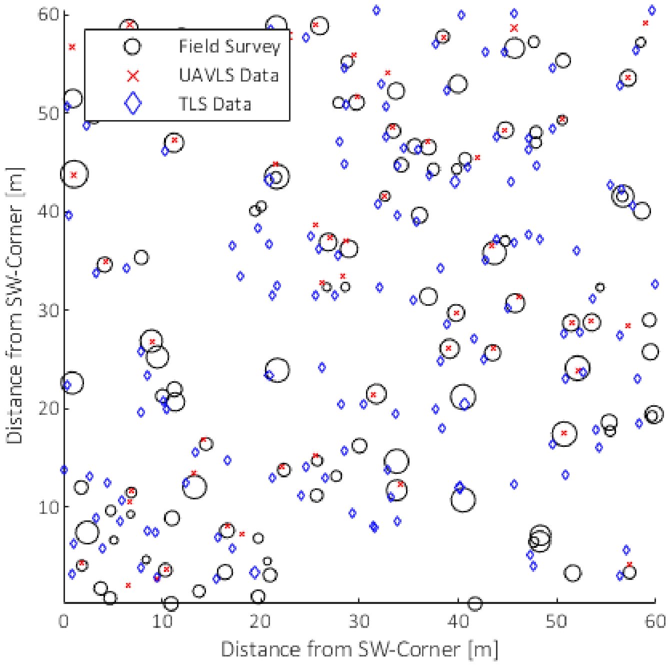


Figure 4-2: All Tree Positions: Field survey, UAV-LS and TLS data.

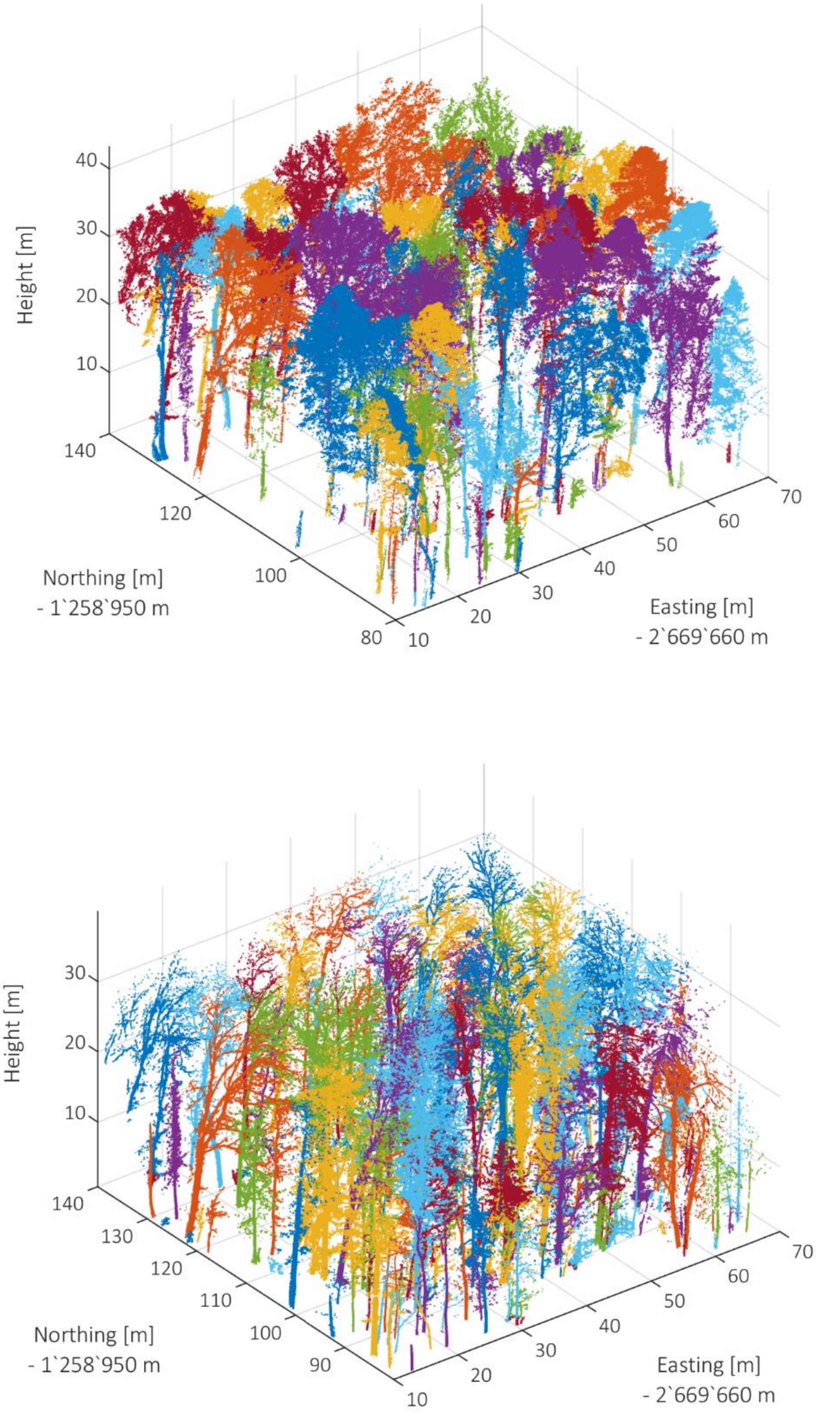


Figure 4-3: Separated trees derived from the UAV-LS (top) and TLS (bottom) data.

4.4 Quantitative Structure Model

Replacing all points of the separated trees with corresponding points of the original vegetation point cloud and their neighbours within 0.1m distance improved the density of the separated trees. Running the treeQSM algorithm of Raumonen et al. (2013) automatically on all separated trees, higher than 10m, resulted in 61 (of 71 separated object) models in the UAV-LS dataset and 124 (of 141 separated object) models in the TLS dataset.

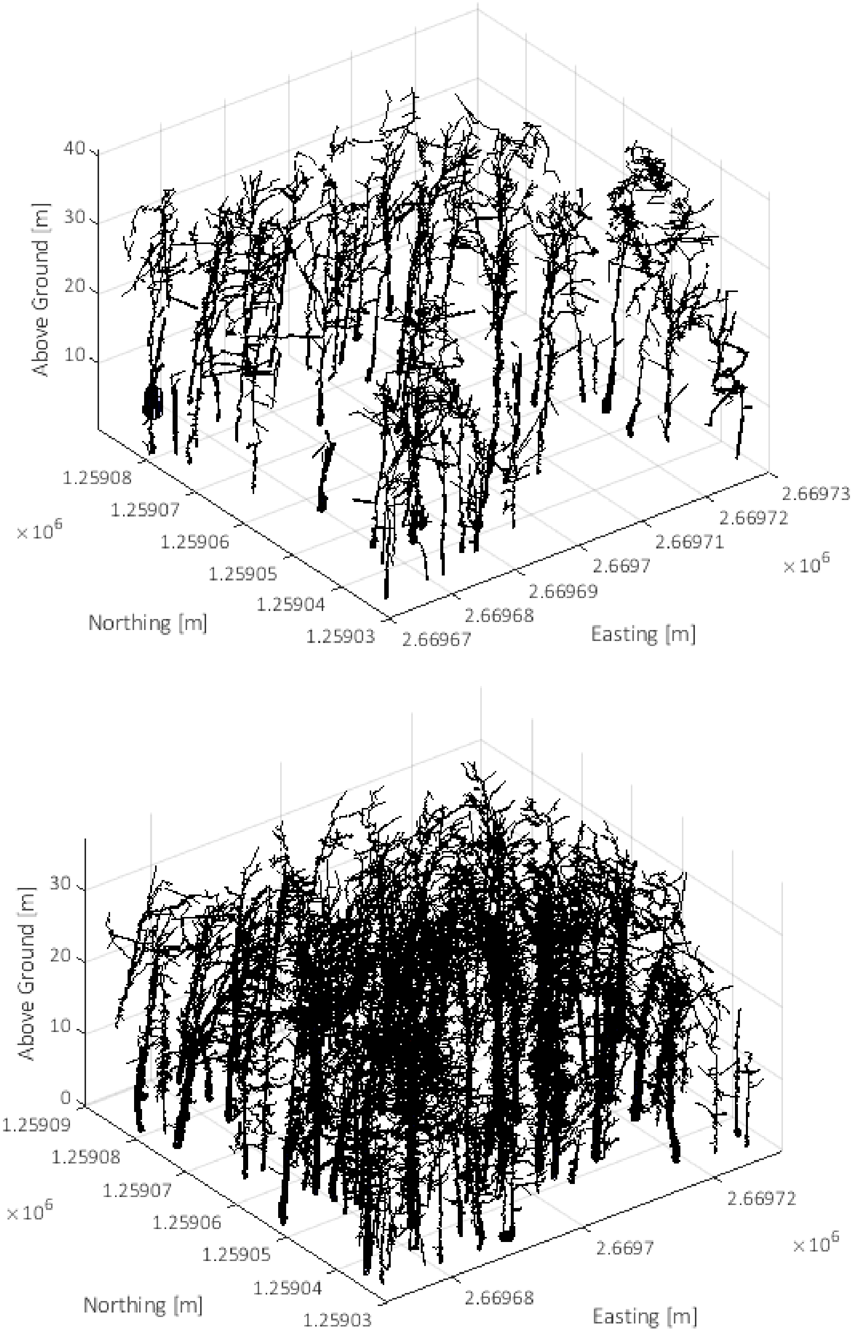


Figure 4-4: Cylinder models generated based on UAV-LS (top) and TLS (bottom) data.

4.5 Tree Measures

The additional tree measure extraction results in a table for each tree containing information about the following measures:

Table 4-5: Extracted tree measures.

ID	Identification
Height	Tree Height
DBH_cyl	DBH with Cylinder Fitting
DBH_d	DBH with mean distance from centre
Area	Covering Area with alpha shape & area
Offset	Offset between Trunk and Canopy (Easting, Northing)
Points	Amount of Points
Trunk_X	Position of Trunk (Easting)
Trunk_Y	Position of Trunk (Northing)
Trunk_Z	Position of Trunk (Height)
Canopy_X	Position of Canopy Centre (Easting)
Canopy_Y	Position of Canopy Centre (Northing)
Canopy_Z	Position of Canopy Centre (Height)

These measures are taken for all trunk position. Thus, we are able to compare an estimated diameter at breast height, for all separated trees. In addition, we are able to answer the research question about the extraction of morphological traits complexity.

Comparing the values of the different diameter at breast height extraction options (figure 4-5) to the ground truth data of the field study results in the RMSE shown in table 4-6.

Table 4-6: RMSE for DBH extraction methods.

TLS			UAV-LS		
QSM	Cylinder	Mean Dist.	QSM	Cylinder	Mean Dist.
0.1167 m	0.1867 m	0.2073 m	0.3014 m	0.8120 m	0.3986 m

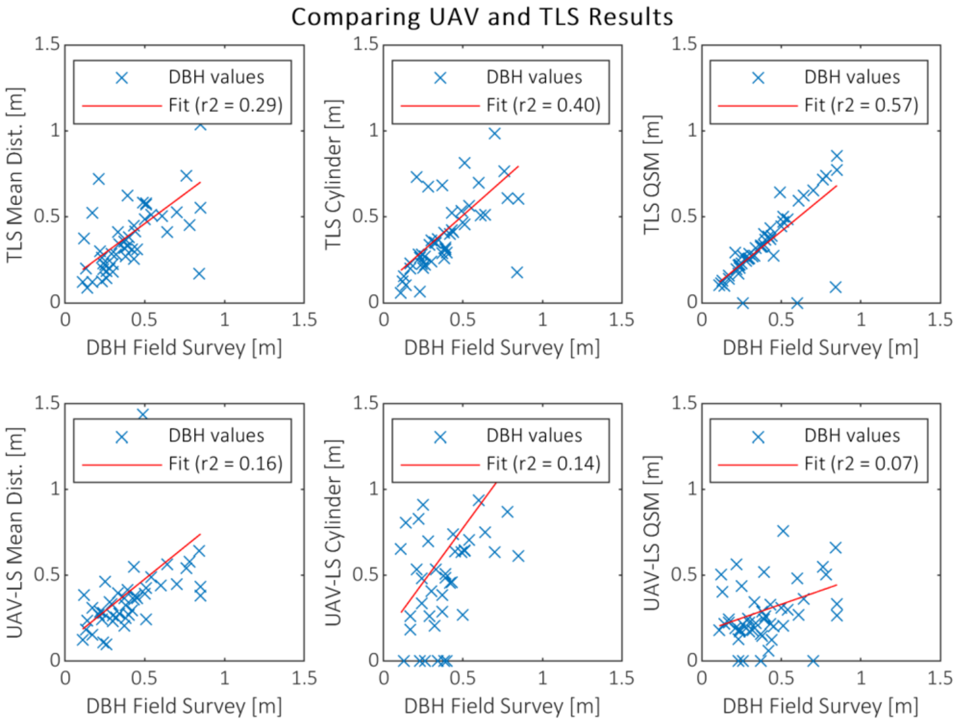


Figure 4-5: Different DBH extraction methods compared to the measurements in the field study.

In addition to the diameter at breast height, we also estimate values for covering area, height, and the offset between trunk and canopy. The results are shown in table 4-7. Our field survey data do not cover these values, but we may compare the UAV-LS tree to the TLS trees.

Table 4-7: Mean tree height, derived from TLS and UAV-LS data.

	TLS	UAV-LS
Mean height [m]	26.51	28.79
Total Crown Surface [m^2 *10^3]	3.0631	3.6680

5 Discussion

5.1 Pre-Processing Steps

Airborne laser data are commonly used for Digital Terrain Model creation. Petzold (1999) describes one of the first cases and ever since, various approaches have evolved (Liu et al., 2008; Petzold et al., 1999; Sithole and Vosselman, 2004). Our approach, combining a quadtree with a common grid approach creates a very detailed product similar to the QuadTIN approach presented by Pajarola et al. (2002) creating a triangulated irregular network to representing the terrain. Enhancement could be achieved by running various classification algorithms that distinguish between non-ground points representing vegetation or artificial objects and ground points. However, in our study area there is only one man-made object that complicates the DTM generation.

Using voxel grid filters on LiDAR data is a widespread method to reduce the workload (Calders et al., 2018; Hackenberg et al., 2015a). If one has the processing power to run the whole algorithm without this reduction, it is highly recommended to dismiss this step.

5.2 Stem Detection

Our approach is built to detect stems in the point cloud with a minimum of effort and input variables. There is a big potential for improvement in this section, especially for the application on the UAV-LS dataset. There are multiple possibilities to enhance the stem detection. For example, one could run additional point classification algorithms like the one presented by Åkerblom et al. (2017), which classifies patches with a principal component analysis and the assumption that stems are vertical. This would lead to a better distinction between the main stems and other clusters especially in the crown area. Another approach to detect stems was conducted by Calders et al. (2018) in their workflow to provide data for a radiative transfer model. For the stem identification they used a height filtering combined with RANSAC cylinder fitting and the angle of the cylinder model in comparison to the DTM surface. Those studies were principally performed with TLS data but could be adjusted for UAV-LS data.

Comparing UAV-LS and TLS data to the field survey highlights the direct correlation between sensor footprint and diameter at breast height threshold as well as the impact of the sensor view angle on the occlusion. Due to the lower overall point density of the

UAV-LS data, which is even amplified in the trunk area (figure 5-1) due to the different perspective, the omission off stems is significantly higher. In addition, the larger footprint of the UAV-LS sensor returns lower intensity signals from small objects leading to the removal of these objects during the filtering of the vegetation cloud. This also explains the higher number of trunks detected in the TLS data set. Both, UAV-LS and TLS derived stems are not filtered by their DBH yet. Therefore, the amount of stems is much higher in comparison to the field study.

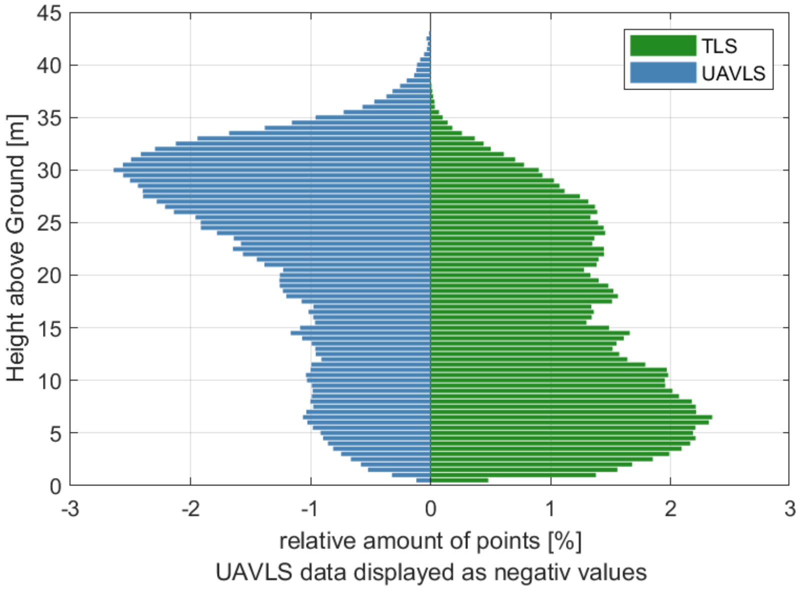


Figure 5-1: Point distribution in height bins.

In addition, we are able to highlight the influence of the tree species on the accuracy of the stem detection with the different sensor types. Best example in our case is a fir (abies alba) with a DBH of approximately 0.8 m which got completely lost in the UAV-LS data due to the occlusion of the stem by the dense branches and shoots but still visible in the TLS data due to the different view angle. In figure 5-2 a transect of the vegetation cloud is shown, including the location of the fir. We see a high occlusion in the crown area and a dense trunk surface in TLS data (left). The UAV-LS data (right) shows a nearly complete occlusion of the stem but in return we have a nice surface of the tree crown.

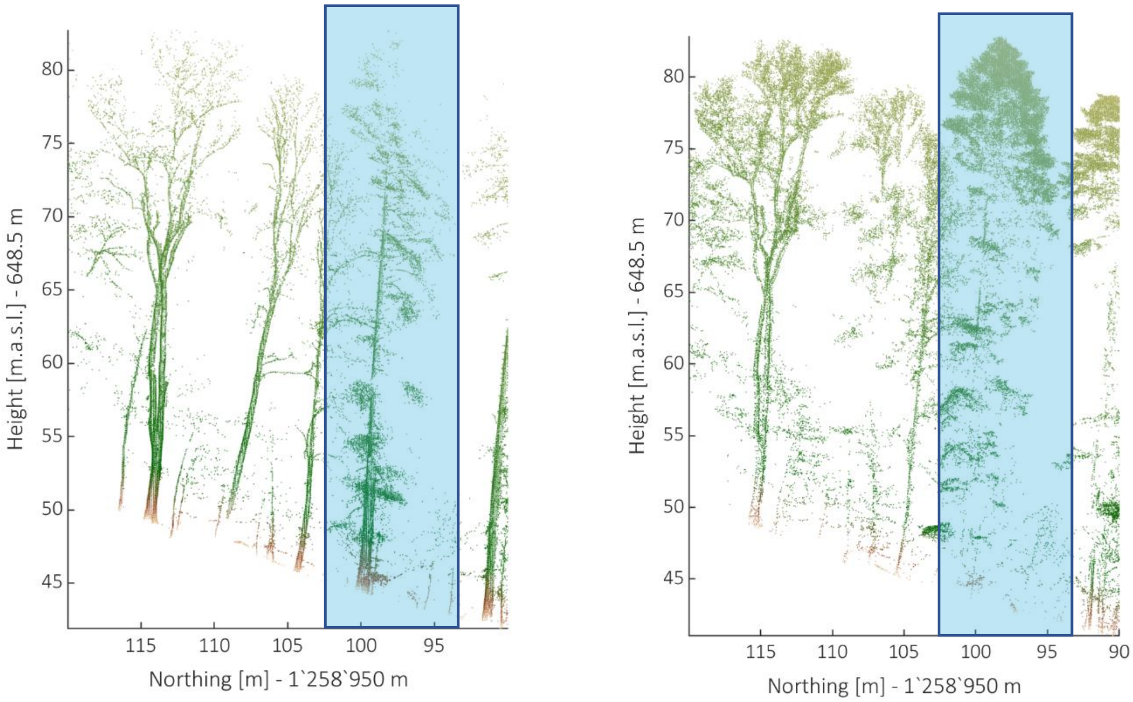


Figure 5-2: Transect of TLS (left) and UAV-LS (right) vegetation cloud showing the different occlusions due to the view direction.

5.3 Tree Separation

After visual evaluation of the resulting trees we are able to categorize three main errors, caused by various reasons. First, peripheral error caused by branches which have their trunk outside the study area and therefore are assigned to their closest trees. Second, there are interwoven trees or tree crowns. In this case, single branches are inseparable mostly due to insufficient resolution of the sensor. As long as the distance within one branch remains shorter than the distance to next branch, the logarithmic edgeweight mostly prevents a "branch hopping" during the shortest path calculation. A third error occurred due to undetected stems. The missed stems in the stem detection section lead to a consequential error in the tree separation.

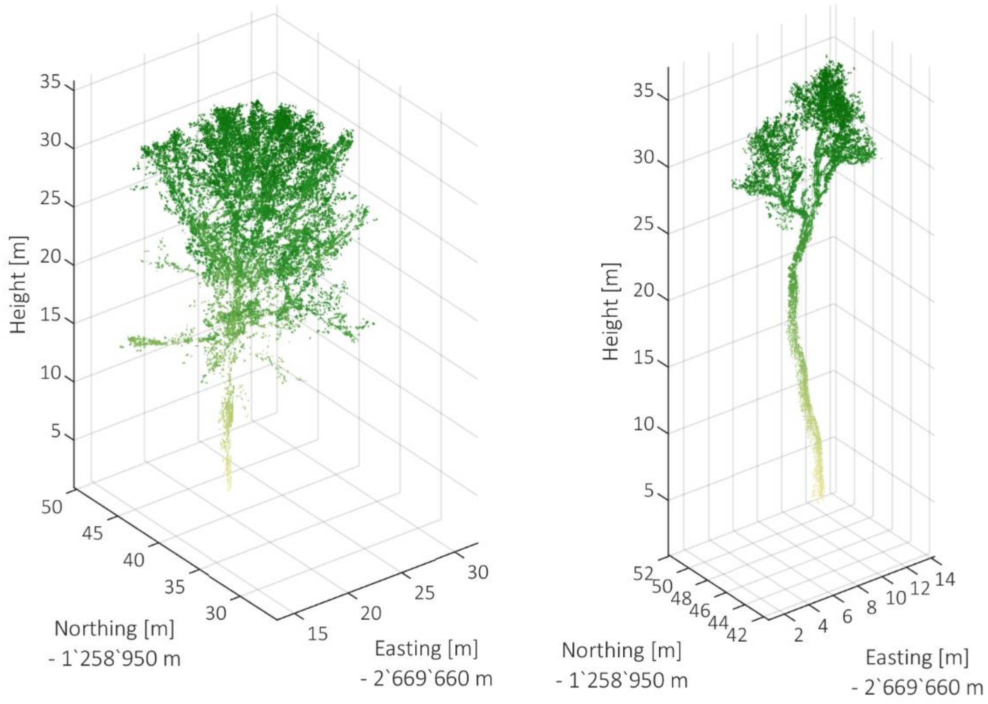


Figure 5-3: Two examples of precise separated trees.

The peripheral error is inevitable if the research area is limited to a part of a forest. We identify this error at approximately one out of seven trees higher than 20 m in the UAV-LS results. In the TLS results, this error occurs at approximately one out of five tall trees.

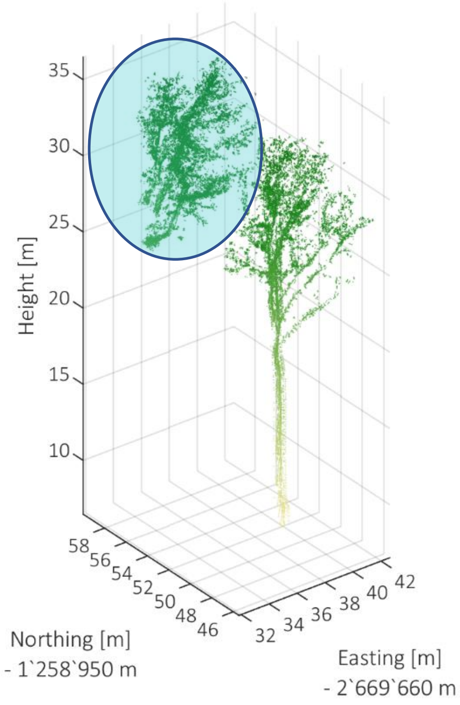


Figure 5-4: An Examples of a peripheral error. Marked with a blue ellipse.

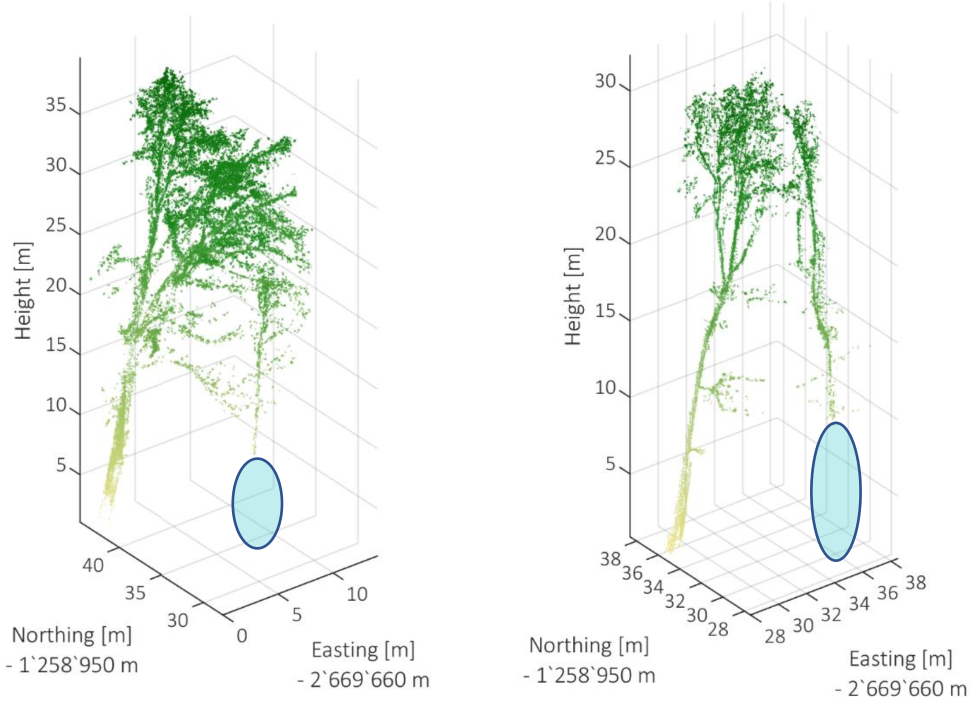


Figure 5-5: Two examples of missed trunks. Marked with blue ellipses.

The error resulting from missing stems happens more frequently in the UAV-LS than in the TLS data. In the UAV-LS results it occurs in approximately one out of five tall trees. In the TLS results it appears only one single time of all 72 compared trees.

The error of the interwoven canopies is difficult to distinguish from the error due to the missing stems. If a trunk position is missing for the separation process, the whole tree will be added to its surrounding trunks. Therefore, it is to distinguish if a tree is split up in total, or if there is only a part of the canopy removed. In case of the UAV-LS dataset we observe this error in one out of ten examples and in case of the TLS in one out of 18. There is much potential left for further processing steps, which could remove branches which do not belong to the targeted tree.

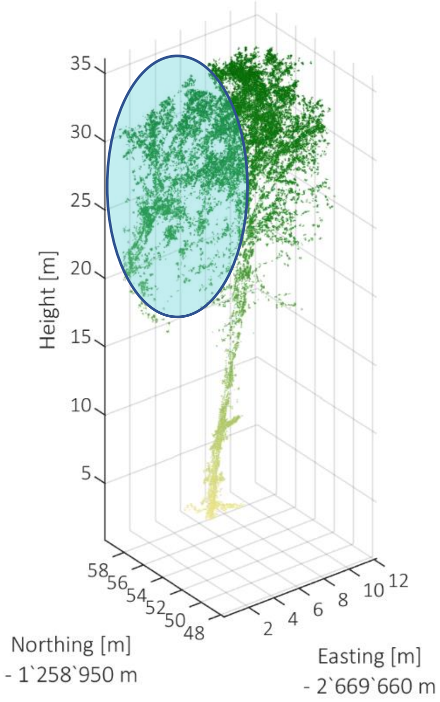


Figure 5-6: Example of two interwoven tree crowns. Marked with a blue ellipse.

The tree separation based on the UAV-LS dataset mainly suffers from the missed stems in the stem detection section. Due to the higher accuracy of the stem detection using the TLS data, it would be of further interest to examine whether the usage of the TLS stem positions could enhance the tree separation of the UAV-LS data. If so, one could use a single dataset of TLS stem positions and combine it with multiple UAV-LS point clouds of a long-term study. The question that therefore arises is as follows: In which time interval are new TLS surveys necessary?

5.3 Quantitative Structure Model

Point clouds acquired with terrestrial laser scanners serve as common input data for QSM approaches. Various studies describe the possibilities and limitation of QSM creation with TLS Data (Hackenberg et al., 2015a; Raumonen et al., 2013, 2015). Thus, it was anticipated that the separated trees from the TLS dataset are well processed. In our knowledge, using UAV-LS Data as input was not done before for a dense and complex (topography) forest as the Laegern. We identified two complications that occurred using the separated trees from the UAV-LS dataset. On the one hand, the tree separation is strongly affected by consequential errors of the missed stems and therefore does not deliver the same detailed separation as in the TLS data. On the other hand, the point density within the stem of the trees is significantly lower in the UAV-LS point cloud. In contrary to the point clouds produced with a TLS, the ones acquired with an UAV-LS lose point density towards the ground due to the view angle and the related occlusion. As the trunk is crucial for the over-all model generation (Hackenberg et al., 2015a; Raumonen et al., 2013), this difference is severe. Possibilities to enhance the stem detection are described in the previous chapter. Reducing the density difference cloud be done with some point interpolation algorithms or comes with the advancing UAV or laser scanner technology.

5.4 Tree Measurements

The extraction of the tree measures enables us to obtain some basic traits of those separated trees, which do not fit the requirement for the quantitative structure model creation. Comparing the three datasets, reveals the high accuracy of the QSM model used on the TLS data. Except from one outlier the calculated DBH of the produced models correlates very accurately with the field survey. However, looking at the UAV-LS data and model generation, we do not determine the same accuracy. This derives mostly from the lower point density in the trunk area. Nevertheless, the result has a lower RMSE than the simplistic mean distance approach (table 4-6). With a RMSE just slightly higher than the one of the QSM approach, we deliver an additional DBH estimation, without the need of a model creation.

6 Conclusion

In this thesis we presented a workflow with the aim of separating trees in a forest with no prior knowledge. Further the separated trees served as input for the QSM approach presented by Raumonen et al. (2013). We used both TLS and UAV-LS data of a tempered mixed forest on the Laegern (Switzerland) as input datasets. In addition, we were in the possession of ground truth data for tree location and diameter at breast height, gathered by A. Rudow and his team. In case of the TLS data our approach led to a satisfactory result with only slight deviations from the ground truth. In case of the UAV-LS data we confronted with two main obstacles. One problem derived from the missing trunks in the stem detection part. The other originated from the lower point density in the trunk area leading to errors in the QSM approach.

Harking back to the research questions:

"Do the point cloud distributions of UAV-LS fit the requirement of tree detection as well as TLS point clouds?"

After visual comparison of the separated trees and the created models, we can conclude that there is a significant difference between the results of the two input datasets. The main difference is based on the lower point density in the trunk area of the UAV-LS data and the higher precision in the TLS data. This leads to the omission of trunks and errors in the cylinder fitting of the QSM in the UAV-LS dataset. It would be interesting to further examine if manual processing of the single trees could reduce the gap between the two results.

"Which traits can be derived from the UAV-LS point cloud?"

The trait identification depends on the spatial resolution. On the one hand, we are able to detect traits such as the diameter at breast height, the tree height or the crown diameter, but on the other hand we cannot detect more complex traits, as for example the angle of branches or the specific wood volume of different branch compartments. Additionally, the identification of the traits is only possible on precisely separated trees.

To put it in a nutshell, we can conclude that there is high potential in the data derived from UAV-LS surveys, nonetheless a QSM generated based on those data does not reach the same accuracy as the one generated from TLS data. The progress in sensor and UAV technology as well as improvements in the implemented approach will certainly enable the extraction of more complex morphological traits from UAV-LS data in the near future. Beyond that it would be of further interest to examine the benefits of merging UAV-LS and TLS point clouds in order to obtain high point densities in the trunk area as well as in the crown.

7 References

Åkerblom, M., Raumonen, P., Mäkipää, R., Kaasalainen, M., 2017. Automatic tree species recognition with quantitative structure models. Remote Sens. Environ. 191, 1–12. https://doi.org/10.1016/j.rse.2016.12.002

Barbeito, I., Dassot, M., Bayer, D., Collet, C., Drössler, L., Löf, M., del Rio, M., Ruiz-Peinado, R., Forrester, D.I., Bravo-Oviedo, A., Pretzsch, H., 2017. Terrestrial laser scanning reveals differences in crown structure of Fagus sylvatica in mixed vs. pure European forests. For. Ecol. Manage. 405, 381–390. https://doi.org/10.1016/J.FORECO.2017.09.043

Bellard, C., Bertelsmeier, C., Leadley, P., Thuiller, W., Courchamp, F., 2012. Impacts of climate change on the future of biodiversity. Ecol. Lett. 15, 365–377. https://doi.org/10.1111/j.1461-0248.2011.01736.x

Bergen, K.M., Goetz, S.J., Dubayah, R.O., Henebry, G.M., Hunsaker, C.T., Imhoff, M.L., Nelson, R.F., Parker, G.G., Radeloff, V.C., 2009. Remote sensing of vegetation 3-D structure for biodiversity and habitat: Review and implications for lidar and radar spaceborne missions. J. Geophys. Res. 114. https://doi.org/10.1029/2008JG000883

Bienert, A., Georgi, L., Kunz, M., Maas, H.-G., von Oheimb, G., Bienert, A., Georgi, L., Kunz, M., Maas, H.-G., von Oheimb, G., 2018. Comparison and Combination of Mobile and Terrestrial Laser Scanning for Natural Forest Inventories. Forests 9, 395. https://doi.org/10.3390/f9070395

Bowker, G.C., 2000. Mapping biodiversity. Int. J. Geogr. Inf. Sci. 14, 739–754. https://doi.org/10.1080/136588100750022769

Butchart, S.H.M., Walpole, M., Collen, B., van Strien, A., Scharlemann, J.P.W., Almond, R.E.A., Baillie, J.E.M., Bomhard, B., Brown, C., Bruno, J., Carpenter, K.E., Carr, G.M., Chanson, J., Chenery, A.M., Csirke, J., Davidson, N.C., Dentener, F., Foster, M., Galli, A., Galloway, J.N., Genovesi, P., Gregory, R.D., Hockings, M., Kapos, V., Lamarque, J.-F., Leverington, F., Loh, J., McGeoch, M.A., McRae, L., Minasyan, A., Hernández Morcillo, M., Oldfield, T.E.E., Pauly, D., Quader, S., Revenga, C., Sauer, J.R., Skolnik, B., Spear, D., Stanwell-Smith, D., Stuart, S.N., Symes, A., Tierney, M., Tyrrell, T.D., Vié, J.-C., Watson, R., 2010. Global biodiversity: indicators of recent declines. Science 328, 1164–8. https://doi.org/10.1126/science.1187512

Calders, K., Newnham, G., Burt, A., Murphy, S., Raumonen, P., Herold, M., Culvenor, D., Avitabile, V., Disney, M., Armston, J., Kaasalainen, M., 2015. Nondestructive estimates of above-ground biomass using terrestrial laser scanning. Methods Ecol. Evol. 6, 198–208. https://doi.org/10.1111/2041-210X.12301

Calders, K., Origo, N., Burt, A., Disney, M., Nightingale, J., Raumonen, P., Åkerblom, M., Malhi, Y., Lewis, P., Calders, K., Origo, N., Burt, A., Disney, M., Nightingale, J., Raumonen, P., Åkerblom, M., Malhi, Y., Lewis, P., 2018. Realistic Forest Stand Reconstruction from Terrestrial LiDAR for Radiative Transfer Modelling. Remote Sens. 10, 933. https://doi.org/10.3390/rs10060933

Cardinale, B.J., Duffy, J.E., Gonzalez, A., Hooper, D.U., Perrings, C., Venail, P., Narwani, A., Mace, G.M., Tilman, D., Wardle, D.A., Kinzig, A.P., Daily, G.C., Loreau, M., Grace, J.B., Larigauderie, A., Srivastava, D.S., Naeem, S., 2012. Biodiversity loss and its impact on humanity. Nature 486, 59–67. https://doi.org/10.1038/nature11148

Cianciaruso, M. V., Batalha, M.A., Gaston, K.J., Petchey, O.L., 2009. Including intraspecific variability in functional diversity. Ecology 90, 81–89. https://doi.org/10.1890/07-1864.1

Ehbrecht, M., Schall, P., Ammer, C., Seidel, D., 2017. Quantifying stand structural complexity and its relationship with forest management, tree species diversity and microclimate. Agric. For. Meteorol. 242, 1–9. https://doi.org/10.1016/J.AGRFORMET.2017.04.012

Ehbrecht, M., Schall, P., Juchheim, J., Ammer, C., Seidel, D., 2016. Effective number of layers: A new measure for quantifying three-dimensional stand structure based on sampling with terrestrial LiDAR. For. Ecol. Manage. 380, 212–223. https://doi.org/10.1016/J.FORECO.2016.09.003

Eugster, W., Zeyer, K., Zeeman, M., Michna, P., Zingg, A., Buchmann, N., Emmenegger, L., 2007. Methodical study of nitrous oxide eddy covariance measurements using quantum cascade laser spectrometery over a Swiss forest. Biogeosciences 4, 927–939. https://doi.org/10.5194/bg-4-927-2007

Hackenberg, J., Spiecker, H., Calders, K., Disney, M., Raumonen, P., 2015a. SimpleTree - An efficient open source tool to build tree models from TLS clouds. Forests 6, 4245–4294. https://doi.org/10.3390/f6114245

Hackenberg, J., Wassenberg, M., Spiecker, H., Sun, D., Hackenberg, J., Wassenberg, M., Spiecker, H., Sun, D., 2015b. Non Destructive Method for Biomass Prediction Combining TLS Derived Tree Volume and Wood Density. Forests 6, 1274–1300. https://doi.org/10.3390/f6041274

Hilker, T., Coops, N.C., Newnham, G.J., van Leeuwen, M., Wulder, M.A., Stewart, J., Culvenor, D.S., 2012. Comparison of Terrestrial and Airborne LiDAR in Describing Stand Structure of a Thinned Lodgepole Pine Forest. J. For. 110, 97–104. https://doi.org/10.5849/jof.11-003

Jaisankar, I., Velmurugan, A., Sivaperuman, C., 2018. Biodiversity Conservation: Issues and Strategies for the Tropical Islands, in: Biodiversity and Climate Change Adaptation in Tropical Islands. Academic Press, pp. 525–552. https://doi.org/10.1016/B978-0-12-813064-3.00019-3

Kaasalainen, S., Krooks, A., Liski, J., Raumonen, P., Kaartinen, H., Kaasalainen, M., Puttonen, E., Anttila, K., Mäkipää, R., Kaasalainen, S., Krooks, A., Liski, J., Raumonen, P., Kaartinen, H., Kaasalainen, M., Puttonen, E., Anttila, K., Mäkipää, R., 2014. Change Detection of Tree Biomass with Terrestrial Laser Scanning and Quantitative Structure Modelling. Remote Sens. 6, 3906–3922. https://doi.org/10.3390/rs6053906

Leadley, P., Pereira, H.M., Alkemade, R., Fernandez-Manjarrés, J.F., Proença, V., Scharlemann, J.P.W., Walpole, M.J., 2010. Biodiversity Scenarios: Projections of 21st century change in biodiversity and associated ecosystem services. Secretariat of the Convention on Biological Diversity, Technical. ed. Secretariat of the Convention of Biological Diversity, Montreal.

Liang, X., Hyyppä, J., 2013. Automatic Stem Mapping by Merging Several Terrestrial Laser Scans at the Feature and Decision Levels. Sensors 13, 1614–1634. https://doi.org/10.3390/s130201614

Liang, X., Kankare, V., Hyyppä, J., Wang, Y., Kukko, A., Haggrén, H., Yu, X., Kaartinen, H., Jaakkola, A., Guan, F., Holopainen, M., Vastaranta, M., 2016. Terrestrial laser scanning in forest inventories. ISPRS J. Photogramm. Remote Sens. 115, 63–77. https://doi.org/10.1016/J.ISPRSJPRS.2016.01.006

Liu, X., Zhang, Z., Peterson, J., Chandra, S., 2008. Large area DEM generation using airborne LiDAR data and quality control. Accuracy Geomatics Proc. 8th Int. Symp. Spat. Accuracy Assess. Nat. Resour. Environ. Sci. 2, 79–85.

Malanson, G.P., Alftine, K.J., 2016. Ecological Impacts of Climate Change, in: Biological and Environmental Hazards, Risks, and Disasters. Academic Press, pp. 397–426. https://doi.org/10.1016/B978-0-12-394847-2.00022-X

Morsdorf, F., Eck, C., Zgraggen, C., Imbach, B., Schneider, F.D., Kükenbrink, D., 2017. UAV-based LiDAR acquisition for the derivation of high-resolution forest and ground information. Lead. Edge 36, 566–570. https://doi.org/10.1190/tle36070566.1

Mura, M., McRoberts, R.E., Chirici, G., Marchetti, M., 2015. Estimating and mapping forest structural diversity using airborne laser scanning data. Remote Sens. Environ. 170, 133–142. https://doi.org/10.1016/J.RSE.2015.09.016

Pajarola, R., Antonijuan, M., Lario, R., 2002. QuadTIN: quadtree based triangulated irregular networks, in: Proceedings of the Conference on Visualization ’02. IEEE, Boston, pp. 395–402.

Pereira, H.M., Ferrier, S., Walters, M., Geller, G.N., Jongman, R.H.G., Scholes, R.J., Bruford, M.W., Brummitt, N., Butchart, S.H.M., Cardoso, A.C., Coops, N.C., Dulloo, E., Faith, D.P., Freyhof, J., Gregory, R.D., Heip, C., Höft, R., Hurtt, G., Jetz, W., Karp, D.S., McGeoch, M.A., Obura, D., Onoda, Y., Pettorelli, N., Reyers, B., Sayre, R., Scharlemann, J.P.W., Stuart, S.N., Turak, E., Walpole, M., Wegmann, M., 2013. Essential biodiversity variables. Science. 339, 277–278. https://doi.org/10.1126/science.1229931

Petzold, B., Reiss, P., Stössel, W., 1999. Laser scanning—surveying and mapping agencies are using a new technique for the derivation of digital terrain models. ISPRS J. Photogramm. Remote Sens. 54, 95–104. https://doi.org/10.1016/S0924-2716(99)00005-2

Pretzsch, H., 2014. Canopy space filling and tree crown morphology in mixed-species stands compared with monocultures. For. Ecol. Manage. 327, 251–264. https://doi.org/10.1016/J.FORECO.2014.04.027

Raumonen, P., Casella, E., Calders, K., Murphy, S., Åkerbloma, M., Kaasalainen, M., 2015. Massive-Scale Tree Modelling From Tls Data. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. II-3/W4, 189–196. https://doi.org/10.5194/isprsannals-II-3-W4-189-2015

Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., 2013. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. Remote Sens. 5, 491–520. https://doi.org/10.3390/rs5020491

Rötheli, E., Heiri, C., Bigler, C., 2012. Effects of growth rates, tree morphology and site conditions on longevity of Norway spruce in the northern Swiss Alps. Eur. J. For. Res. 131, 1117–1125. https://doi.org/10.1007/s10342-011-0583-4

Schneider, F.D., Leiterer, R., Morsdorf, F., Gastellu-Etchegorry, J.-P., Lauret, N., Pfeifer, N., Schaepman, M.E., 2014. Simulating imaging spectrometer data: 3D forest modeling based on LiDAR and in situ data. Remote Sens. Environ. 152, 235–250. https://doi.org/10.1016/J.RSE.2014.06.015

Sithole, G., Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 59, 85–101. https://doi.org/10.1016/J.ISPRSJPRS.2004.05.004

Sverdrup-Thygeson, A., Ørka, H.O., Gobakken, T., Næsset, E., 2016. Can airborne laser scanning assist in mapping and monitoring natural forests? For. Ecol. Manage. 369, 116–125. https://doi.org/10.1016/J.FORECO.2016.03.035

Vihervaara, P., Mononen, L., Auvinen, A.P., Virkkala, R., Lü, Y., Pippuri, I., Packalen, P., Valbuena, R., Valkama, J., 2015. How to integrate remotely sensed data and biodiversity for ecosystem assessments at landscape scale. Landsc. Ecol. 30, 501–516. https://doi.org/10.1007/s10980-014-0137-5

Vosselman, G., Maas, H.-G., Van Genderen, J.L., 2011. Airborne and terrestrial laser scanning. Int. J. Digit. Earth 4, 183–184. https://doi.org/10.1080/17538947.2011.553487

Williams, P.H., Gaston, K.J., Humphries, C.J., 1997. Mapping biodiversity value worldwide: combining higher-taxon richness from different groups. Proc. R. Soc. B Biol. Sci. 264, 141–148. https://doi.org/10.1098/rspb.1997.0021

Yamasaki, E., Altermatt, F., Cavender-Bares, J., Schuman, M.C., Zuppinger-Dingley, D., Garonna, I., Schneider, F.D., Guillén-Escribà, C., van Moorsel, S.J., Hahl, T., Schmid, B., Schaepman-Strub, G., Schaepman, M.E., Shimizu, K.K., 2017. Genomics meets remote sensing in global change studies: monitoring and predicting phenology, evolution and biodiversity. Curr. Opin. Environ. Sustain. 29, 177–186. https://doi.org/10.1016/J.COSUST.2018.03.005

Zhao, K., García, M., Liu, S., Guo, Q., Chen, G., Zhang, X., Zhou, Y., Meng, X., 2015. Terrestrial lidar remote sensing of forests: Maximum likelihood estimates of canopy profile, leaf area index, and leaf angle distribution. Agric. For. Meteorol. 209–210, 100–113. https://doi.org/10.1016/J.AGRFORMET.2015.03.008

8 Supplementary Material

8.1 Functions and Code

During the course of this thesis numerous functions were developed. Below, table 8-1 presents those functions and their application. The crucial ones are highlighted with bold font and closer examined in separate chapters.

Table 8-1: Implemented functions.

Illustrations	
rgb_colors	Function providing various colours
rgb_colormap	Creates colormaps from input colours
scatter3_pc	Plotting 3d point cloud with title, labels, font, etc.
scattter3_t	Same as scatter3_pc, just for tables

Calculations	
dividePC_cellfun	DTM creation and point classification - divides the point clouds into four sub-clouds - separates vegetation and ground - creates DTM & DSM
chm_cellfun	Creates CHM from DTM & DSM
substract_dtm_cellfun	Calculates above ground height
createvoxel3dveg_cellfun	Uses the cells from the dividePC_cellfun result to create voxels
adding_x_y_id_cellfun	Adds x,y ids to the voxel in order to relocated the represented points
dbscan	Find the number of neighbours within a radius
create_edgelist	Creates edgelist from point cloud, using max distance, max nodes
create_graph 4voxeldata	Graph creation
calculate_Stem_measures 4voxeldata	Filters stem cloud and calculates basic stem measures
enhance_stem_PC	Enhances stems with points from the vegetation point cloud
growing_Forest 4voxel	Separating trees
enhance_tree	Enhancing trees with vegetation points
extract_tree_data	Extracting tree measures

Structural	
cell2tablePC	Changes cell structure from DTM creation to table structure
combining_Stems	Combines all stems point clouds in a single point cloud

8.1.1 Digital Terrain Model Creation

In order to use the advantages of a quad tree approach in the digital terrain model creation we applied the following workflow:

- creating the first cell
- saving the whole Point Cloud in it
- saving the absolute minimum of the point cloud

The next steps are to be iteratively repeated until the resolution is on the scale needed for the study:

- quartering each cell
- saving all points in its corresponding cell
- comparing the difference between old and new minimum
- if beneath a selected threshold, replacing the old minimum

In the last iteration the DTM can be smoothed to reduce the noise emerging from cells in which the lowest points do not correspond to the ground. The points contained in each cell are divided into ground points, vegetation points and highest points. The thresholds, distinguishing ground points and vegetation points, are linked to the cell resolution which means a maximal slope needs to be determined for the study site.

We ran 10 iterations which leads to a grid of 1024x1024 cells. Each cell has a resolution of 78.2 x 78.2 square millimetre. In the DTM created using the TLS data, the influence of the quadtree is well depicted due to different grid resolutions across the study area. The closer to the periphery the coarser the resolution becomes. This directly corresponds to a decrease in point density. The spots with the lowest resolutions are also the ones with the biggest height differences between UAV-LS and TLS DTM which is the effect from ground occlusion in the TLS point cloud. This occlusion appears predominately due to changes in the slope caused by the forest road located in the southern part of the study area.

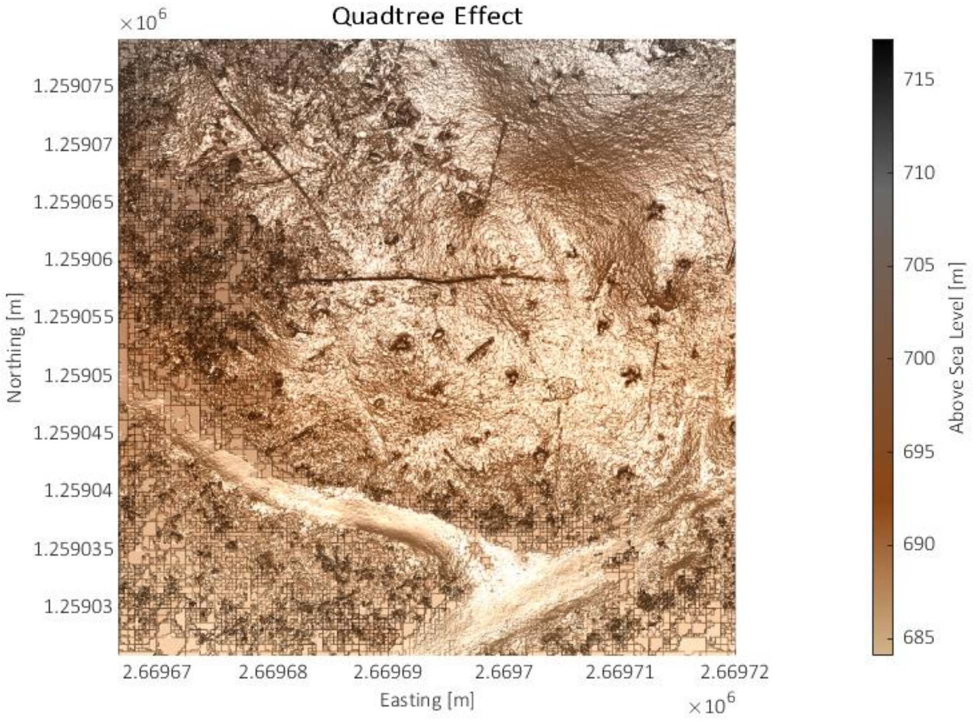


Figure 8-1: Effect of the Quadtree approach.

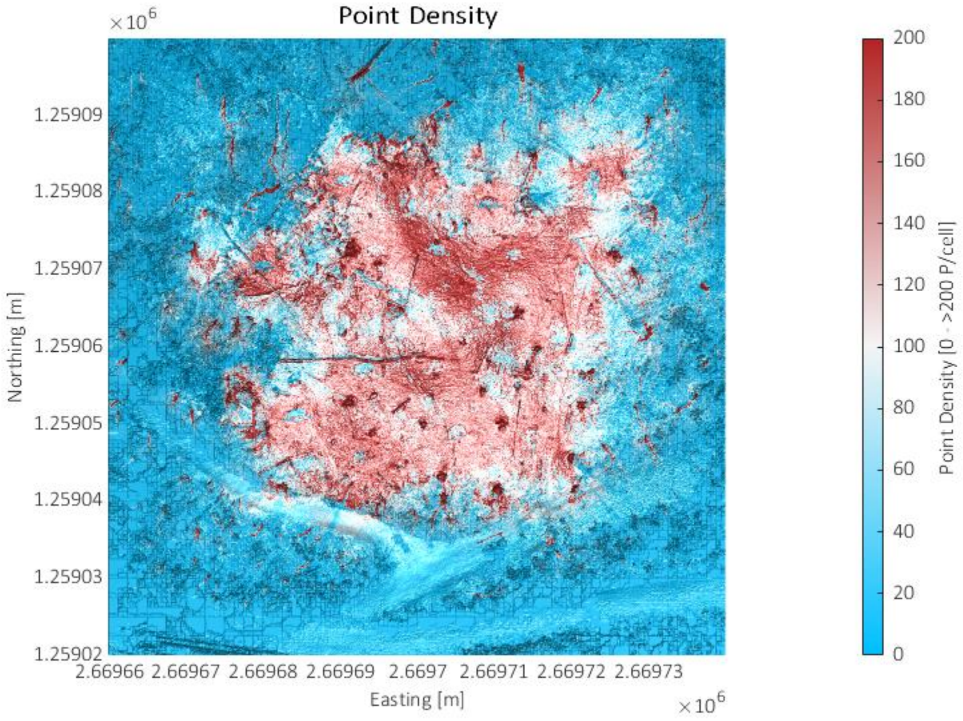


Figure 8-2: Point density of the TLS Point Cloud.

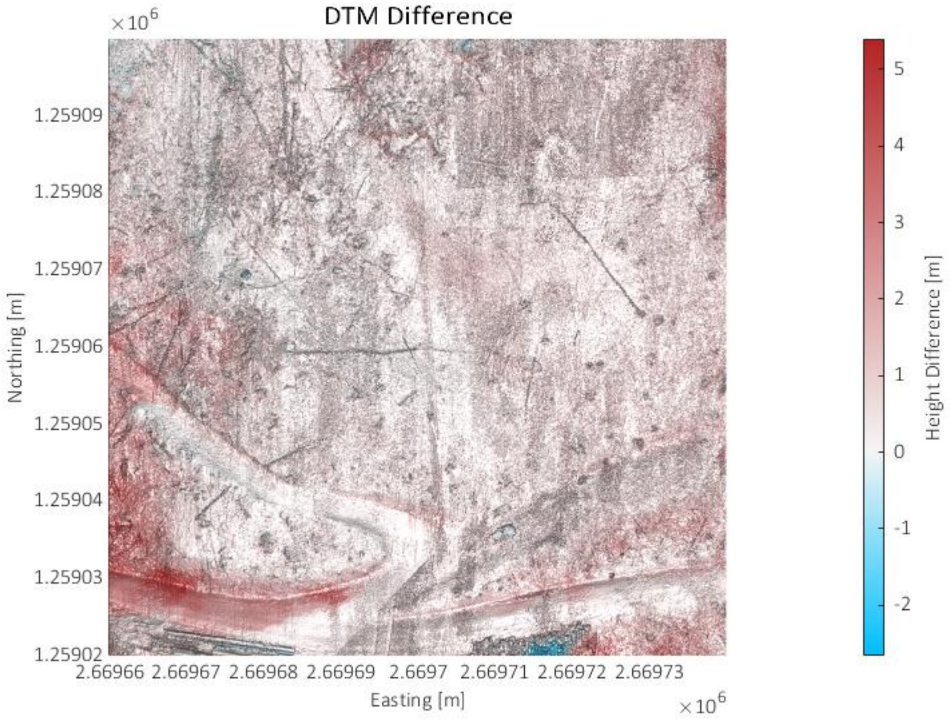


Figure 8-3: Difference between UAV-LS and TLS DTM.

The MATLAB code of the function:

```
function [cloud_out,dtm_out,dsm_out,gp,vp,hp] = dividePC_cellfun(cloud,dtm,dsm,input,var)
% [out,z_out] = dividePC(in,z_in)
% Function to divide Point Clouds in four Subclouds each.
%
%
% Input:
%  in   -> cell() containing pointclouds (:,3)
%  z_in -> matrix or single value: same size as in containing the z_min
%         per cloud
%  slope-> maximal slope accounted
%
% Output:
% out  -> cell() with divided pointclouds
% z_out-> matrix of same size as out, containing the min values of each
%         cloud
%
% manuel.luck@gmail.com

%% First: detecting size of input and creating the output variables
 treesize  = size(cloud);
 cloud     = make_M_greater(cloud,2);
 dtm_out   = make_M_greater(dtm,2);
 dsm_out   = make_M_greater(dsm,2);

 switch var.splitting_PC
   case 'no'
     gp= [];
     vp = [];
     hp = [];
 end

%% Second: getting Resolution and X,Y Coords
  res(1:2)  = input.plotsize(1,1:2)./(treesize(1:2)*2);
 xcoord   = input.min(1,1):res(1):input.max(1)-res(1);
 ycoord   = input.min(1,2):res(2):input.max(2)-res(2);

 threshold = max(res)*var.slopefactor;
 if threshold < 0.5
   threshold = 0.5;
 end

 [X,Y] = meshgrid(xcoord,ycoord);

 X = num2cell(X);
 Y = num2cell(Y);

 cloud_out = cellfun(@(C1,C2,C3) quad_cloud(C1,C2,C3,res),cloud,X,Y,'UniformOutput',false);

 [dtm_out,dsm_out] = cellfun(@(C1,C2,C3)
dtm_dsm_cellfun(C1,C2,C3,threshold),cloud_out,dtm_out,dsm_out,'UniformOutput',false);

 switch var.splitting_PC
   case 'yes'
     [gp,vp,hp] = cellfun(@(C1,C2,C3) classi_gp_vp_hp_cellfun(C1,C2,threshold),cloud_out,dtm_out,'UniformOutput',false);
 end


 switch var.smoothing
   case 'yes'
     dtm_out = cell2mat(dtm_out);
     dtm4smooth = cat(1,dtm_out(1,:),dtm_out(1,:),dtm_out(1,:),...
       dtm_out,...
       dtm_out(end,:),dtm_out(end,:),dtm_out(end,:));
     dtm4smooth = cat(2,dtm4smooth(:,1),dtm4smooth(:,1),dtm4smooth(:,1),...
       dtm4smooth,...
       dtm4smooth(:,end),dtm4smooth(:,end),dtm4smooth(:,end));

     [ext1,ext2,~] =size(dtm_out);

     dtm4mean=....
       cat(3,dtm_out,...
       dtm4smooth(1:ext1,1:ext2),...
       dtm4smooth(1+1:1+ext1,1+1:1+ext2),...
       dtm4smooth(1+2:2+ext1,1+2:2+ext2),...
       dtm4smooth(1+3:3+ext1,1+3:3+ext2),...
       dtm4smooth(1+4:4+ext1,1+4:4+ext2),...
       dtm4smooth(1+5:5+ext1,1+5:5+ext2),...
       dtm4smooth(1+6:6+ext1,1+6:6+ext2),...
       dtm4smooth(1+1:1+ext1,1:ext2),...
       dtm4smooth(1+2:2+ext1,1:ext2),...
       dtm4smooth(1+3:3+ext1,1:ext2),...
       dtm4smooth(1+4:4+ext1,1:ext2),...
       dtm4smooth(1+5:5+ext1,1:ext2),...
       dtm4smooth(1+6:6+ext1,1:ext2),...
       dtm4smooth(1:ext1,1+1:1+ext2),...
       dtm4smooth(1:ext1,1+2:2+ext2),...
       dtm4smooth(1:ext1,1+3:3+ext2),...
       dtm4smooth(1:ext1,1+4:4+ext2),...
       dtm4smooth(1:ext1,1+5:5+ext2),...
       dtm4smooth(1:ext1,1+6:6+ext2));
     dtm_mean = mean(dtm4mean,3);
     dtm_std = std(dtm4mean,0,3);

     bin = abs(dtm_out-dtm_mean) < dtm_std;
     dtm_final(bin) = dtm_out(bin);
     dtm_final(bin==0) = dtm_mean(bin == 0);
     dtm_out = reshape(dtm_final,size(dtm_out,1),size(dtm_out,2));
     dtm_out = num2cell(dtm_out);
 end
%
end

function [out] = make_M_greater(in,multiplier)
%% [out] = make_M_greater(in,multiplier)
```

```
% replaces each grid-unity of in with multiplier x multiplier units
%
% manuel.luck@gmail.com

 for i = 1:size(in,1)
  for j = 1:size(in,2)
   out((i*multiplier+1)-multiplier:(i*multiplier),(j*multiplier+1)-multiplier:(j*multiplier)) = in (i,j);
  end
 end
end

function [out] = quad_cloud(cloud,X,Y,res)
%% out = quad_cloud(cloud,X,Y,res)
% selects the subclouds for each cell
%
% manuel.luck@gmail.com

 out = cloud(cloud(:,1) >= X & cloud(:,1) < X + res(1) &...
    cloud(:,2) >= Y & cloud(:,2) < Y + res(2),:);
end

function [dtm_out,dsm_out] = dtm_dsm_cellfun(cloud,dtm_in,dsm_in,threshold)
%% dtm_out,dsm_out] = dtm_dsm_cellfun(cloud,dtm_in,dsm_in,threshold)
% creates a DTM value and a DSM value by:
% determining if the difference between the lowest point in the cloud and
% the actual dtm are withhin a specific threshold in order to either change
% the dtm for the cell or to stay with the old value.
%
% manuel.luck@gmail.com

 if size(cloud,2) > 1
  mini = min(cloud(:,3));
  if abs(mini-dtm_in) <= threshold
   dtm_out = mini;
  else
   dtm_out = dtm_in;
  end
  dsm_out = max(cloud(:,3));
 else
  dtm_out = dtm_in;
  dsm_out = dsm_in;

 end
end

function [gp,vp,hp] = classi_gp_vp_hp_cellfun(cloud,dtm,threshold)
%[gp,vp,hp] = classi_gp_vp_hp_cellfun(cloud,dtm,threshold)
% classifies each point as vegetation point, ground point or highest point.
%
% manuel.luck@gmail.com
 if size(cloud,2) > 1
  hp = cloud(cloud(:,3) == max(cloud(:,3)),:);
  gp = cloud(abs(cloud(:,3)-dtm) <= threshold,:);
  vp = cloud(abs(cloud(:,3)-dtm) > threshold,:);
 else
  gp = [];
  hp = [];
  vp = [];
 end
end
```

8.1.3 Growing Forest

Growing Forest is the function used for separating the trees. To achieve this separation, three input variables are needed. First the *Stem_Points*, an array containing the coordinates of each trunk point and information about its intensity, the number of points represented by this voxel point, three voxel identification numbers as well as the trunk id. The second variable is the array *no_understory*. This variable contains the filtered voxel-vegetation points with a minimal height above the maximal height of the separated trunks. The array has the same structure as the *Stem_Points* with the only difference that there is no trunk id. The third input variable is the *max_d*. It contains the maximal distance allowed between two nodes to build an edge.

Starting with these three variables, we add an additional trunk id (9999) to the *no_understory* points and concatenate the two arrays to a single point cloud. We use this point cloud and the maximal distance value for the edgelist production and the graph creation. As we want a logarithmic scale to the egdeweights, we use the 10th logarithm of the distance between the nodes. Instead of using the connected compartments as in the trunk separation before, we now compare the shortest path from each node in the graph to the highest point of each trunk. The points are then assigned to the trunk connected with the shortest overall distance.

Supplementary Material

The MATLAB code of the function:

```
function [Tree, Forest] = growing_Forest4voxel(Stem_Points, no_understory, max_d)
%% [Tree, Forest] = growing_Forest4voxel(Stem_Points,no_understory,max_d)
% Uses shortes path within a build graph to separate Trees.
%
% input:  Stem_Points:  Previously separated stem point cloud
%      no_understory: Point cloud containing vegetation higher than Stem_Points
%      max_d:     maximal distance to build graph
%
% output:  Tree:      cells containing all points of a tree
%      Forest:     point cloud containing all points with an id for
%             each tree
%
% manuel.luck@gmail.com

%% Code
  under_s = Stem_Points;
  upper_s = no_understory;
  upper_s(:,10) = 9999;


  Cloud = [upper_s;under_s];

  G = create_graph_by_edgelist([Cloud(:,1:2),Cloud(:,3)],max_d);
  G.Nodes.cloud  = Cloud(1:size(G.Nodes,1),:);

  disp('Searching Stems')
  stem_ids = unique(under_s(:,10));
  stem_ids(stem_ids == 9999) = [];

  disp('extracting')
  Easting   =  G.Nodes.cloud(:,1);
  Northing   =  G.Nodes.cloud(:,2);
  AboveDTM   =  G.Nodes.cloud(:,3);
  Intensity  =  G.Nodes.cloud(:,4);
  AboveSea   =  G.Nodes.cloud(:,5);
  Voxelcount  =  G.Nodes.cloud(:,6);
  Dim1     =  G.Nodes.cloud(:,7);
  Dim2     =  G.Nodes.cloud(:,8);
  Dim3     =  G.Nodes.cloud(:,9);
  Stem_Nr   =  G.Nodes.cloud(:,10);

  Out     =  table(Easting, Northing, AboveDTM,...
            Intensity,AboveSea,Voxelcount,...
            Dim1,Dim2,Dim3,Stem_Nr);

  disp('Calculating Shortest Path')
  D = zeros(size(stem_ids,1),size(G.Nodes,1));
  for iv = 1:size(stem_ids,1)
    disp(iv)
    id_min = max(G.Nodes.cloud(G.Nodes.cloud(:,10) == stem_ids(iv,1),3));
    idx_g = find(G.Nodes.cloud(:,3) == id_min & G.Nodes.cloud(:,10) == stem_ids(iv,1));
    [~,D(iv,:)] = shortestpathtree(G,idx_g(1),'all');
  end

  [min_D,idx_shortest_Path] = min(D,[],1);

  idx = stem_ids(idx_shortest_Path);
  idx(min_D == inf) = NaN;

  Out.Stem_Nr = idx;
  Forest = Out;

  Tree{1,size(stem_ids,1)} = [];
  for i = 1:size(stem_ids,1)
      Tree{i} = Out(Out.Stem_Nr==i,:);
  end
end
```

```
function G = create_graph_by_edgelist(in,max_d)
%% G = create_graph_by_edgelist(in,max_d)
% creates graph with an edgelist connecting all points within max_d of
% eachother.
% input:  in:   PointCloud(x,y,z)
%      max_d: Maximal Distance (d)
%
% output:  G:   Graph with logrithmic edgeweight

disp('building Tree')
  Mdl = KDTreeSearcher(in);

disp('kNN-Search')
  [IdxNN_a,D_a] = rangesearch(Mdl,in,max_d);

disp('Edgelist')

  edgelist = cellfun(@(C1,C2) create_edgelist_cellfun(C1,C2),IdxNN_a,D_a,'UniformOutput',false);
  edgelist  = cell2mat(edgelist(:));

  G = graph(edgelist(:,1),edgelist(:,2),10.^(edgelist(:,3)+1));
end
```

8.1.4 Extract Tree Data

Extract Tree Data has the goal to extract basic measures for the case, that the QSM generation did not work.

In order to find the trunk position, we use a height interval from 1-2 meters. Then we try to fit a cylinder in the cloud using the MATLAB cylinder fitting function. If this works, we use the centre of the cylinder model as our trunk centre and two times the Radius as DBH. If the cylinder fitting does not find enough inlier points in order to create a model, we use the mean position of each point in this height interval. The DBH is calculated as the mean distance (only Easting and Northing) of each point within the interval to the centre.

The canopy position is estimated by calculating the mean position of each point in the top meter of each tree. Thus, the Offset between canopy and trunk is the difference in Easting and Northing between the two positions.

The tree height is the highest point in the cloud.

How much area is covered by the tree is estimated using a two dimensional alphashape of the point cloud.

The MATLAB code of the function:

```
function tree_data = extract_tree_data(tree_cloud)
%% tree_data = extract_tree_data(tree_cloud)
% Extracting measures from tree point clouds
%
% manuel.luck@gmail.com

%% Code
  tree_ids = unique(tree_cloud(:,6));

  ID      = zeros(size(tree_ids,1),1);
  Height  = zeros(size(tree_ids,1),1);
  DBH_cyl  = zeros(size(tree_ids,1),1);
  DBH_d    = zeros(size(tree_ids,1),1);
  Area     = zeros(size(tree_ids,1),1);
  Offset   = zeros(size(tree_ids,1),1);
  Points   = zeros(size(tree_ids,1),1);
  Trunk_X  = zeros(size(tree_ids,1),1);
  Trunk_Y  = zeros(size(tree_ids,1),1);
  Trunk_Z  = zeros(size(tree_ids,1),1);
  Canopy_X = zeros(size(tree_ids,1),1);
  Canopy_Y = zeros(size(tree_ids,1),1);
  Canopy_Z = zeros(size(tree_ids,1),1);

  tree_data  =
table(ID,Height,DBH_cyl,DBH_d,Area,Offset,Points,Trunk_X,Trunk_Y,Trunk_Z,Canopy_X,Canopy_Y,Canopy_Z,'VariableNames',
{'ID','Height','DBH_cyl','DBH_d','Area','Offset','Points','Trunk_X','Trunk_Y','Trunk_Z','Canopy_X','Canopy_Y','Canopy_Z'});

  for i = 1:size(tree_ids,1)
    % subcloud
    cloud    = tree_cloud(tree_cloud(:,6) == tree_ids(i),:);

    % Cylinderfitting
    ptCloud = pointCloud(cloud(cloud(:,3)<=2,1:3));
    [model,~,~] = pcfitcylinder(ptCloud,0.1,[0,0,1],5);

    % alphashape 2d
    shp2d = alphaShape(cloud(:,1:2));

    % Canopy
    canopy = mean(cloud(cloud(:,3)>=max(cloud(:,3))-1,1:3),1);

    if model.Center(1,1) > 0
      Center = model.Center;
      offset = pdist2(canopy(1,1:2),model.Center(1,1:2));
    else
      Center = mean(cloud(cloud(:,3)<=min(cloud(:,3))+max(cloud(:,3))./10,1:3),1);

      offset = pdist2(canopy(1,1:2),Center(1,1:2));
    end

    % Saving
    tree_data.ID(i)      = tree_ids(i);
    tree_data.Height(i)   = max(cloud(:,3));
    tree_data.DBH_cyl(i)    = model.Radius.*2;
    tree_data.DBH_d(i) = mean(pdist2(mean(cloud(cloud(:,3)<=2,1:2)),cloud(cloud(:,3)<=2,1:2)))*2;
    tree_data.Area(i)     = area(shp2d);
    tree_data.Offset(i)   = offset;
    tree_data.Points(i)   = size(cloud(:,1),1);
    tree_data.Trunk_X(i)  = Center(1,1);
    tree_data.Trunk_Y(i)  = Center(1,2);
    tree_data.Trunk_Z(i)  = Center(1,3);
    tree_data.Canopy_X(i)  = canopy(1,1);
    tree_data.Canopy_Y(i)  = canopy(1,2);
    tree_data.Canopy_Z(i)  = canopy(1,3);
    clear model canopy shp2d cloud
  end
end
```

8.1.5 Raumonen et al. treeqsm.m

We used the provided input structure (Raumonen et al., 2013) and added a while loop around the whole given structure. Like this we are able to continue easy if an error occured in the function.

Our input variable is called Tree_Cloud_final, containing the coordinates, tree id, intensity and voxel ids for each point. The output variable is a cell structure containing all outputs from the treeqsm function in a separate cell for each tree.

The MATLAB code of the function:

```
while i <= size(forest_tree_id,1)
     if max(Tree_Cloud_final(Tree_Cloud_final(:,10)==forest_tree_id(i),3)) -
     min(Tree_Cloud_final(Tree_Cloud_final(:,10)==forest_tree_id(i),3)) >= 20
   % The following parameters can be varied and should be optimised (they
   % can have multiple values given as vectors, e.g. [4 6]):
     inputs.PatchDiam1 = 0.1; % Patch size of the first uniform-size cover
     inputs.PatchDiam2Min = [0.02 0.03]; % Minimum patch size of the cover sets in the second cover
     inputs.PatchDiam2Max = 0.06; % Maximum cover set size in the stem's base in the second cover
     inputs.lcyl = [2 6]; % Relative (length/radius) length of the cylinders
     inputs.FilRad = 3; % Relative radius for outlier point filtering
   % The following parameters can be varied and but usually can be kept as
   %
   % shown (i.e. little bigger than PatchDiam parameters):
     inputs.BallRad1 = inputs.PatchDiam1+0.01; % Ball radius in the first uniform-size cover generation
     inputs.BallRad2 = inputs.PatchDiam2Max+0.01; % Maximum ball radius in the second cover generation
   % The following parameters can be usually kept fixed as shown:
     inputs.nmin1 = 3; % Minimum number of points in BallRad1-balls, generally good value is 3
     inputs.nmin2 = 1; % Minimum number of points in BallRad2-balls, generally good value is 1
     inputs.OnlyTree = 1; % If 1, point cloud contains points only from the tree
     inputs.Tria = 0; % If 1, produces a triangulation
     inputs.Dist = 1; % If 1, computes the point-model distances
   % Different cylinder radius correction options for modifying too large and
   % too small cylinders:
   % Traditional TreeQSM choices:
   inputs.MinCylRad = 0.0025; % Minimum cylinder radius, used particularly in the taper corrections
   inputs.ParentCor = 1; % Radii in a child branch are always smaller than the radii of the parent cylinder in the parent branch
   inputs.TaperCor = 1; % Use partially linear (stem) and parabola (branches) taper corrections
   % Growth volume correction approach introduced by Jan Hackenberg,
   % allometry: GrowthVol = a*Radius^b+c
   inputs.GrowthVolCor = 0; % Use growth volume (GV) correction
   inputs.GrowthVolFac = 2.5; % fac-parameter of the GV-approach, defines upper and lower bound. When using GV-approach,
consider setting: TaperCorr = 0, ParentCorr = 0, MinCylinderRadius = 0.

  % Other inputs

   % These parameters don't affect the QSM-reconstruction but define what is
   % saved, plotted, and displayed and how the models are named/indexed
   inputs.name = strcat('Tree ',num2str(i)); % Name string for saving output files and naming models
   inputs.tree = i; % Tree index. If modelling multiple trees, then they can be indexed uniquely
   inputs.model = 1; % Model index, can separate models if multiple models wit h the same inputs
   inputs.savemat = 0; % If 1, saves the output struct QSM as a MATLAB-file into \result folder
   % If name = 'pine', tree = 2, model = 5,
   % the name of the saved file is 'QSM_pine_t2_m5.mat'
   inputs.savetxt = 0; % If 1, saves the models in .txt-files
   inputs.plot = 0; % If 1, plots the model, the segmentation of the point cloud and distributions
   inputs.disp = 2; % Defines what is displayed during the reconstruction: 2 = display all; 1 = display name, parameters and distances;
0 = display only the name

   QSM(i) = {treeqsm(Tree_Cloud_final(Tree_Cloud_final(:,10)==forest_tree_id(i),1:3),
          inputs)};
  end
  i = i + 1;
end
```

8.2 Additional Figures

8.2.1 Research Area, TLS & UAV-LS Point Cloud

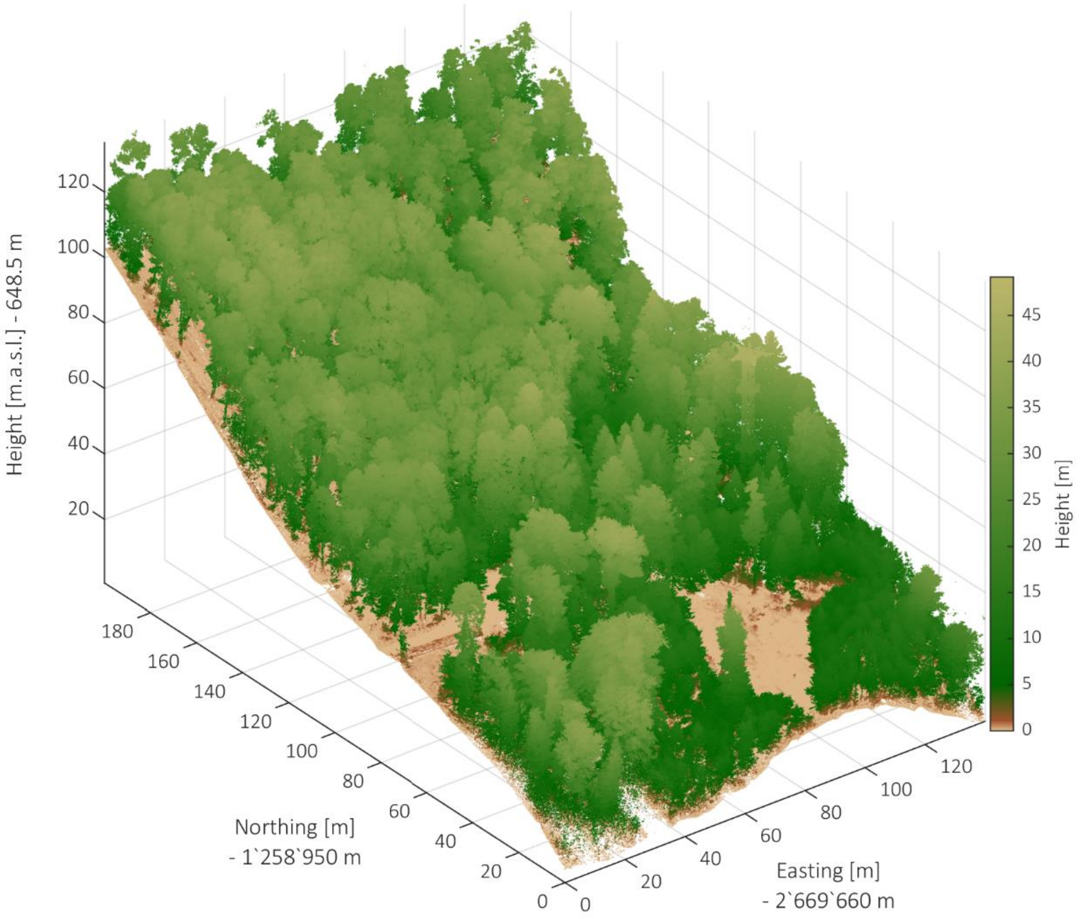


Figure 8-4: UAV-LS Point Cloud.

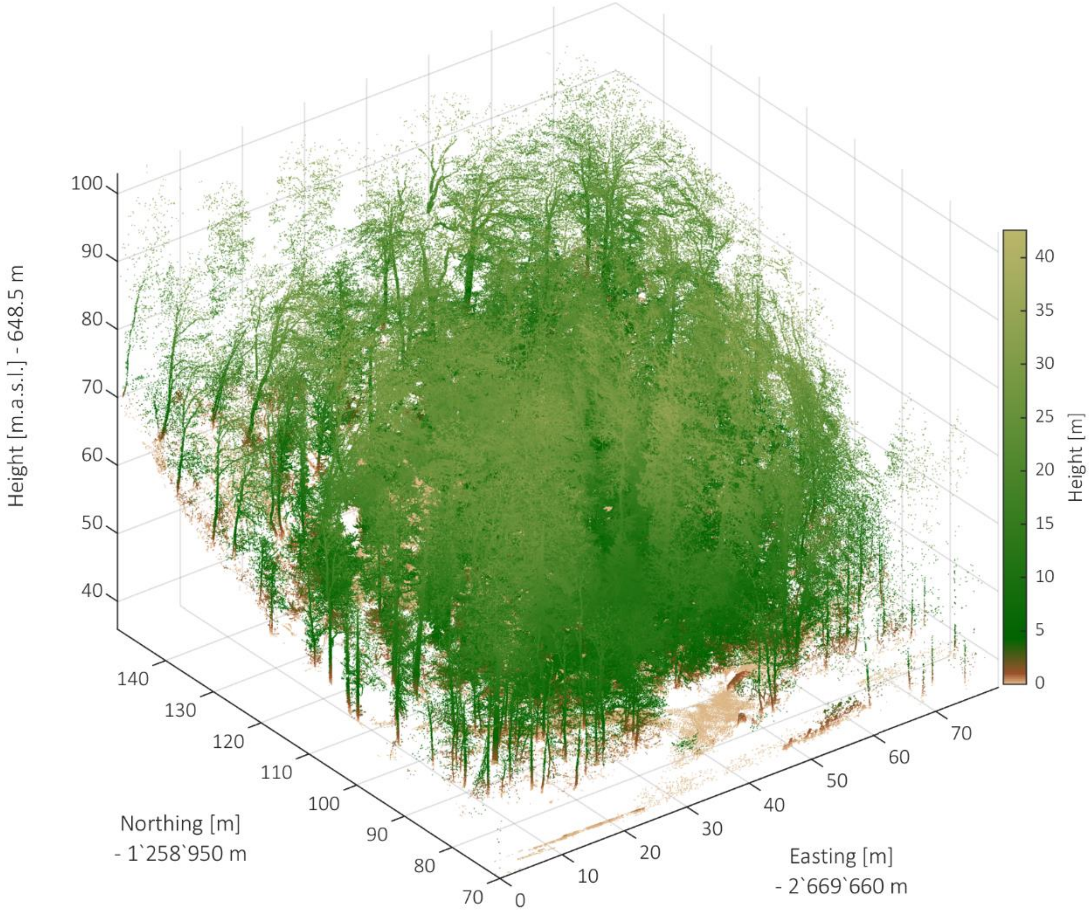
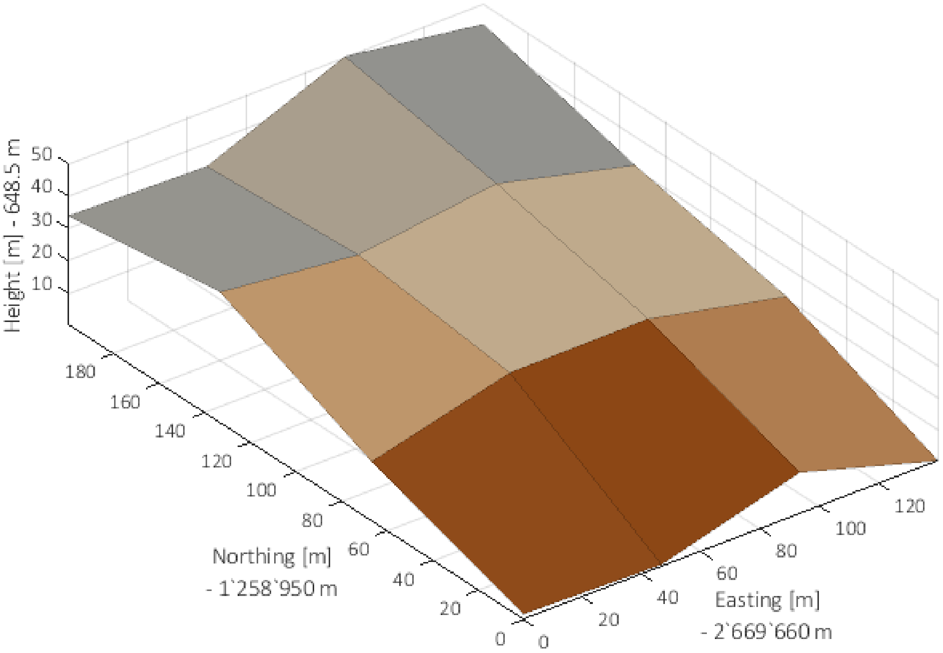


Figure 8-5: TLS Point Cloud.

8.2.2 DTM Creation



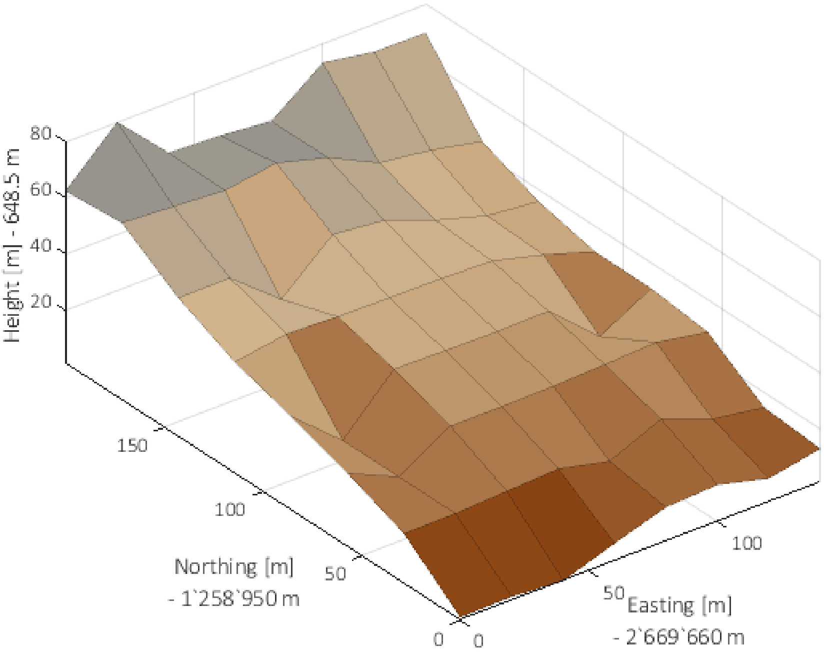
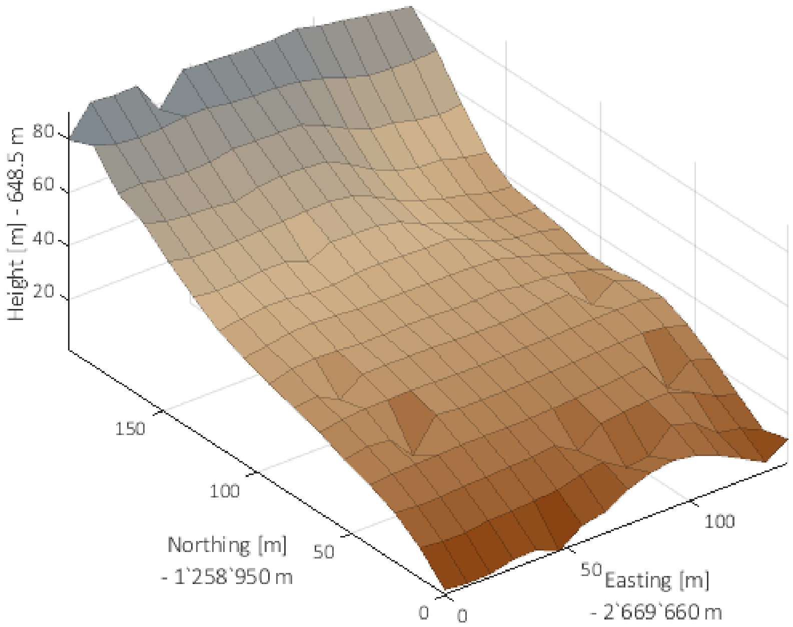


Figure 8-6: DTM generation, iteration 2 (top) and 3 (bottom).



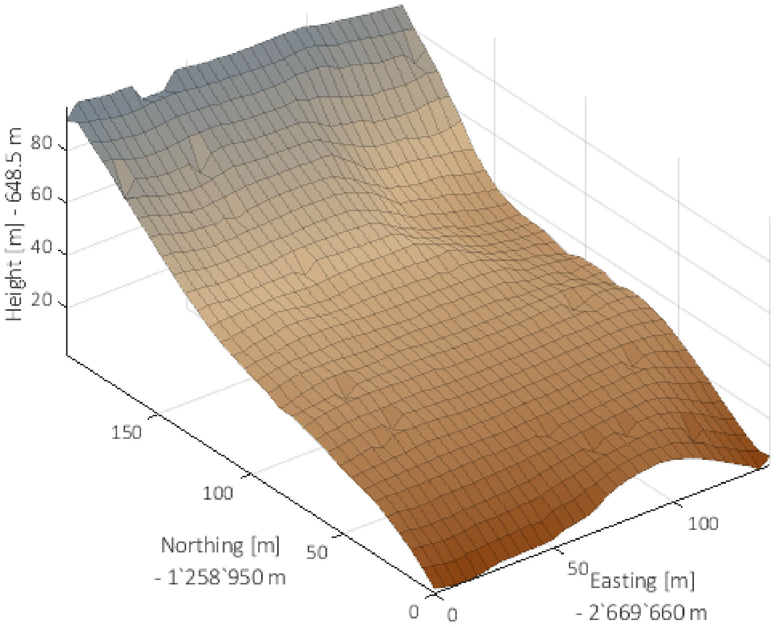
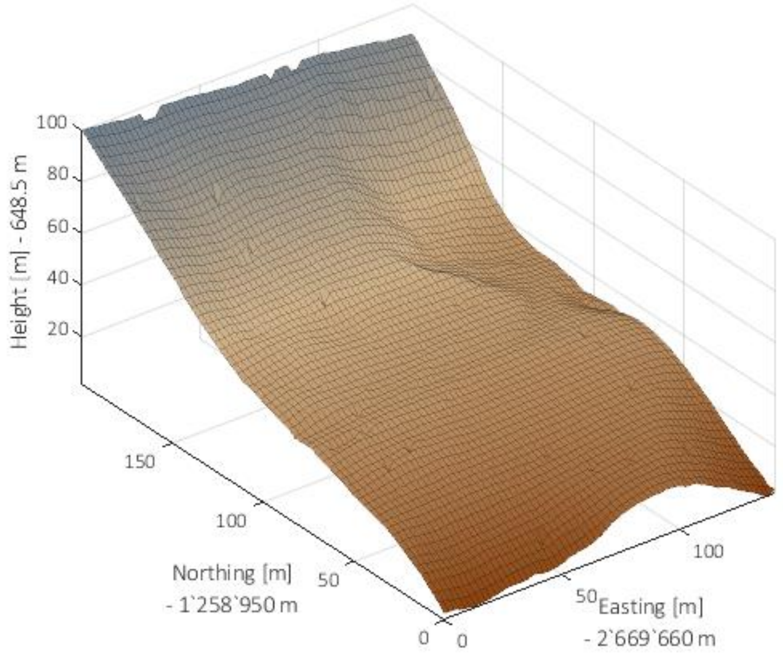


Figure 8-7: DTM generation, iteration 4 (top) and 5 (bottom).



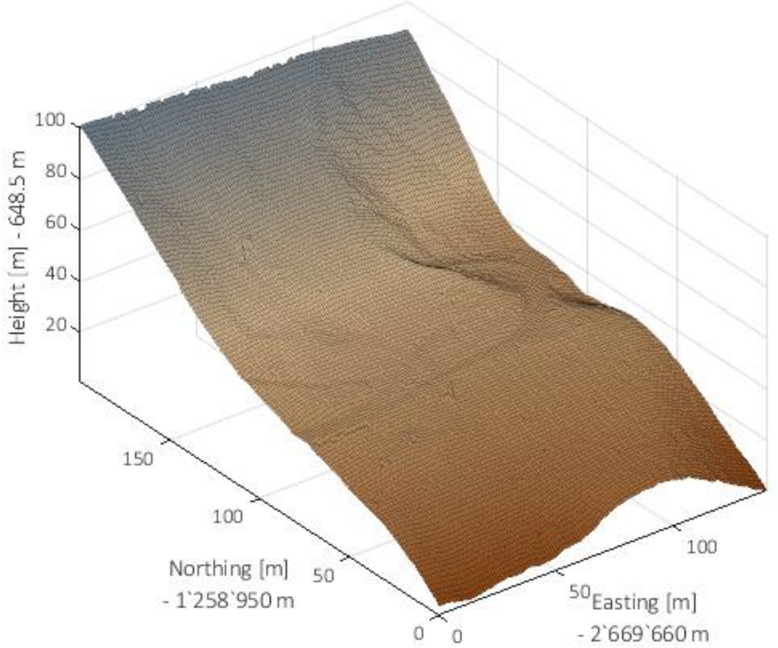
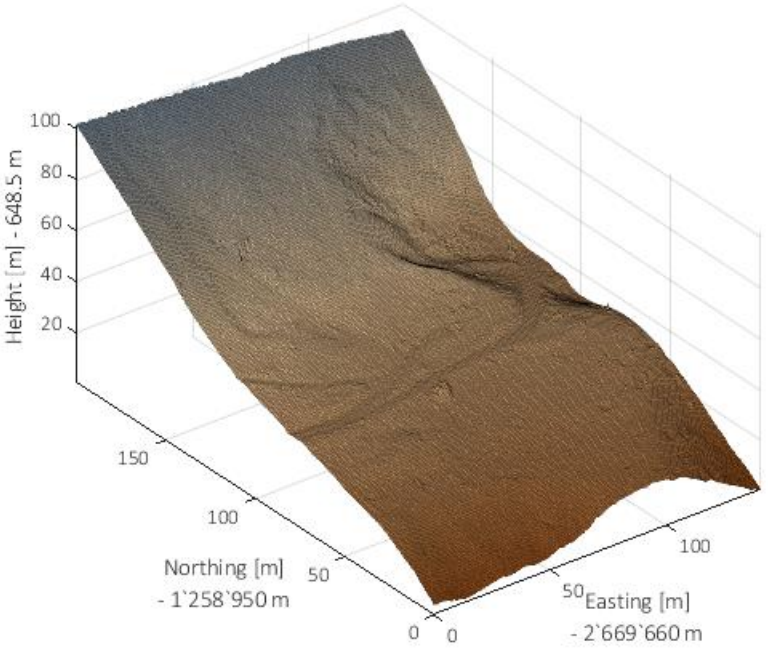


Figure 8-8: DTM generation, iteration 6 (top) and 7 (bottom).



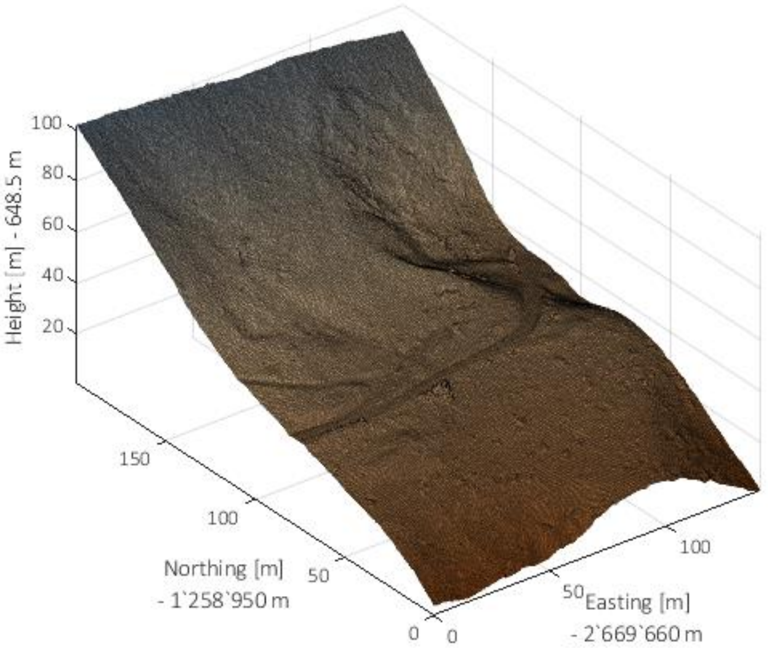
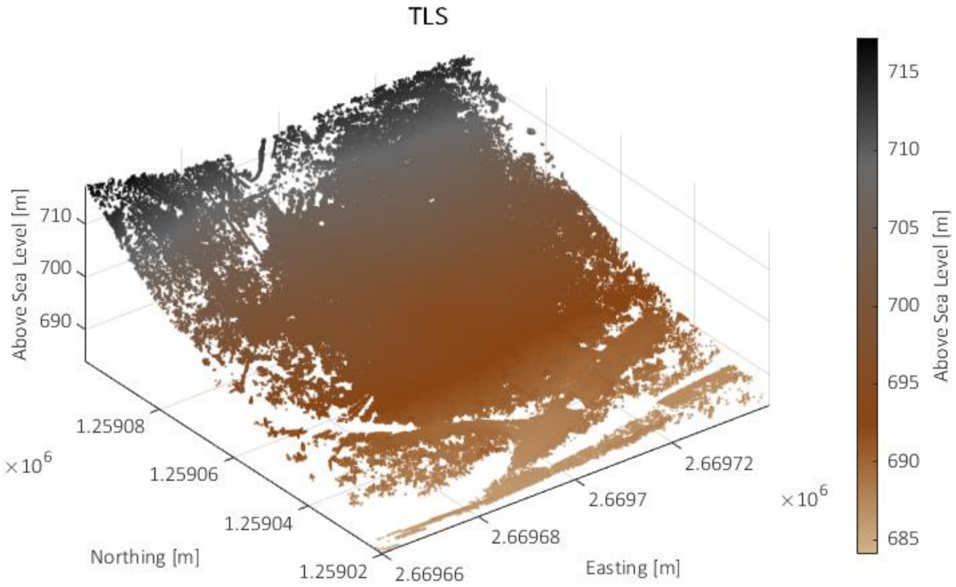


Figure 8-9: DTM generation, iteration 8 (top) and 9 (bottom).



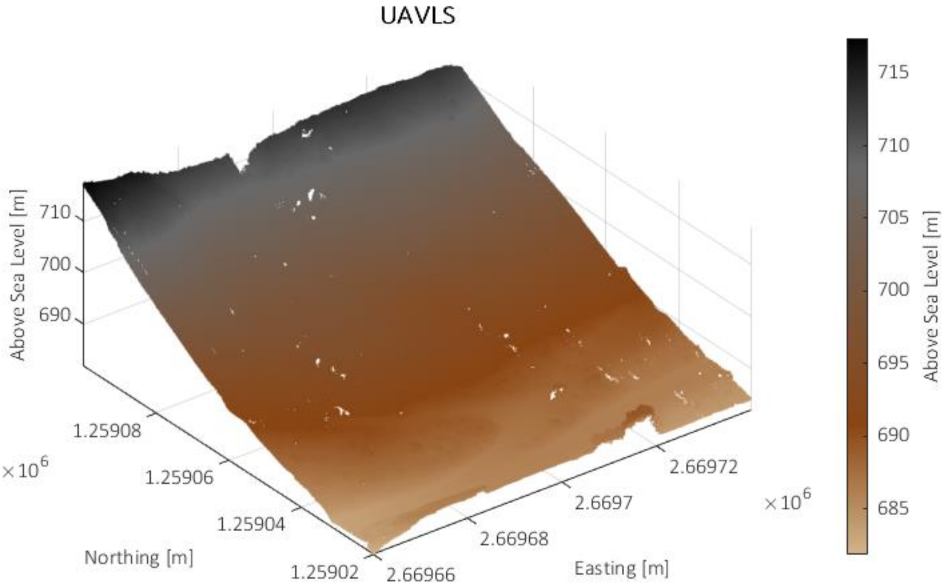


Figure 8-10: Comparison between TLS (top) and UAV-LS (bottom) ground points.

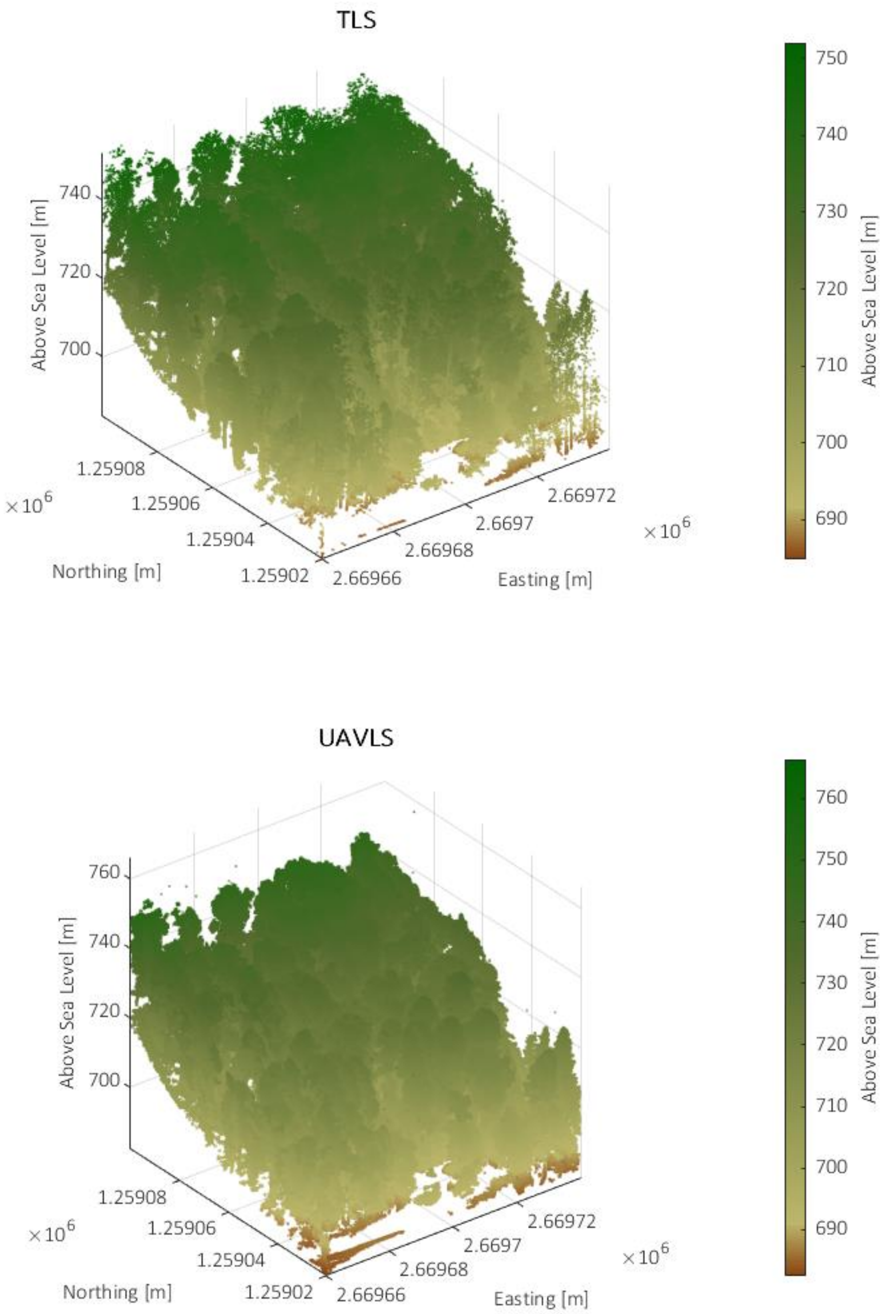


Figure 8-11: Comparison between TLS (top) and UAV-LS (bottom) vegetation points.

8.2.3 TLS Trees

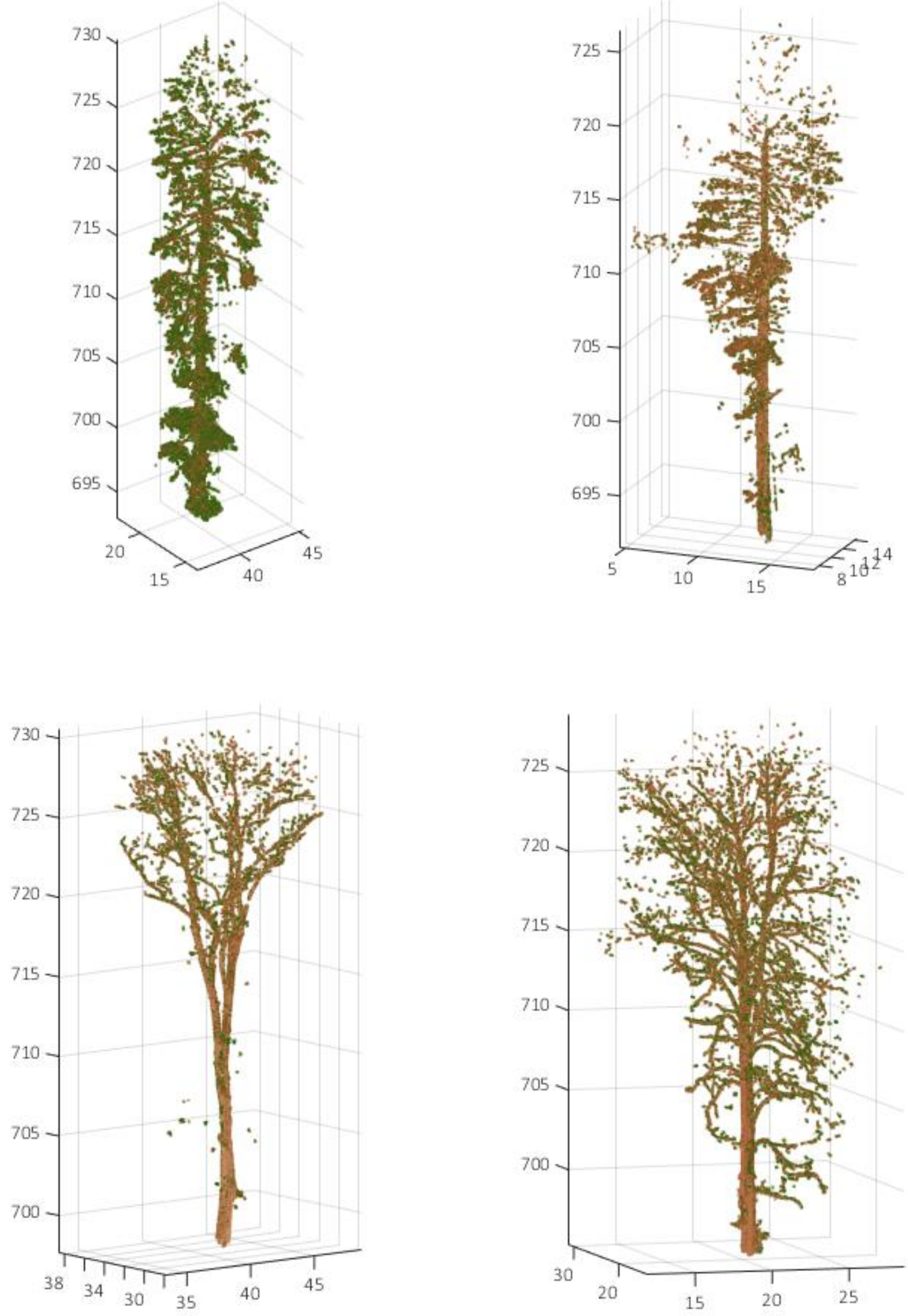


Figure 8-12: Separated trees from TLS Point Cloud.

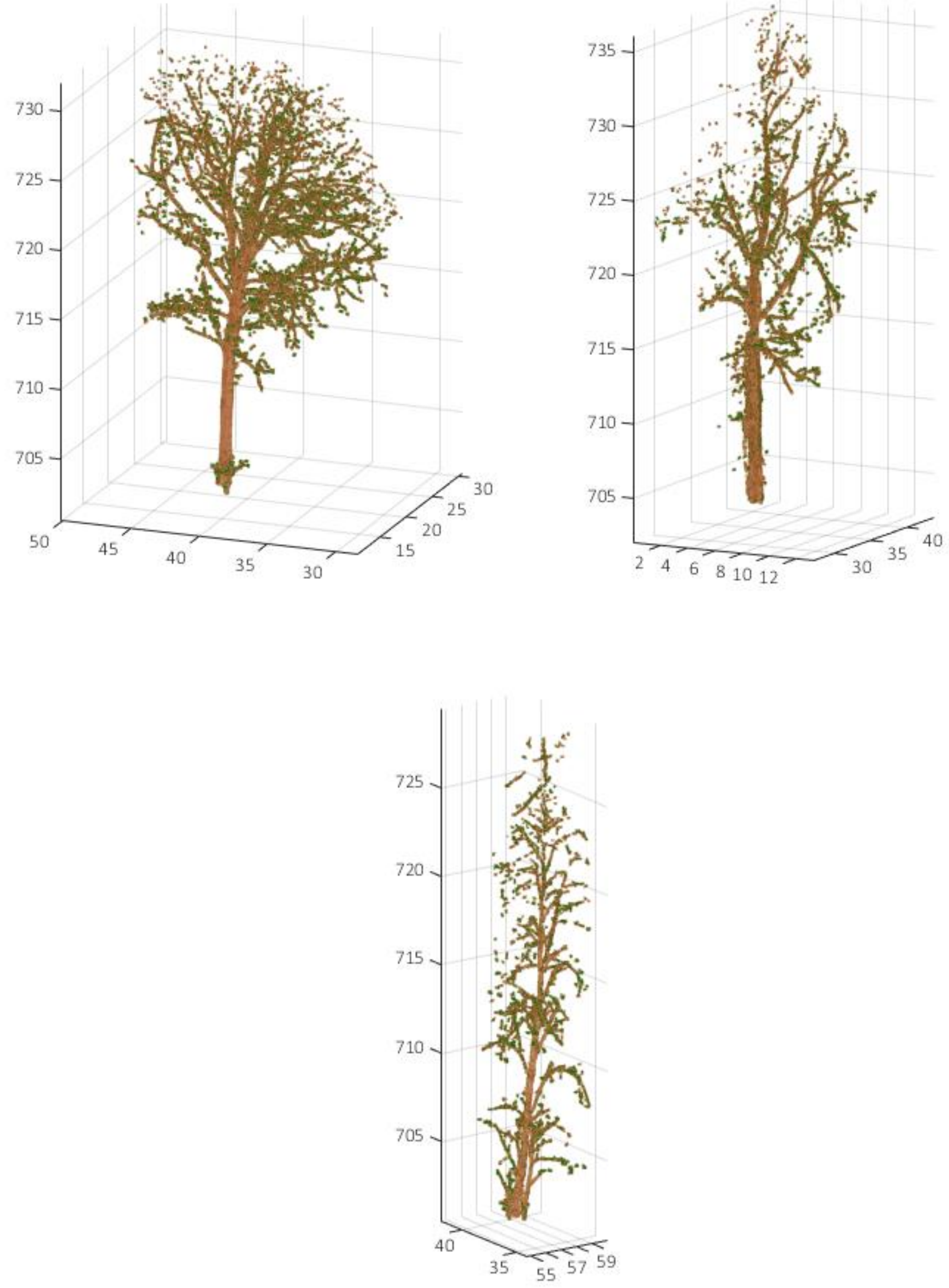


Figure 8-13: Separated trees from TLS Point Cloud.

8.2.4 UAV-LS Trees

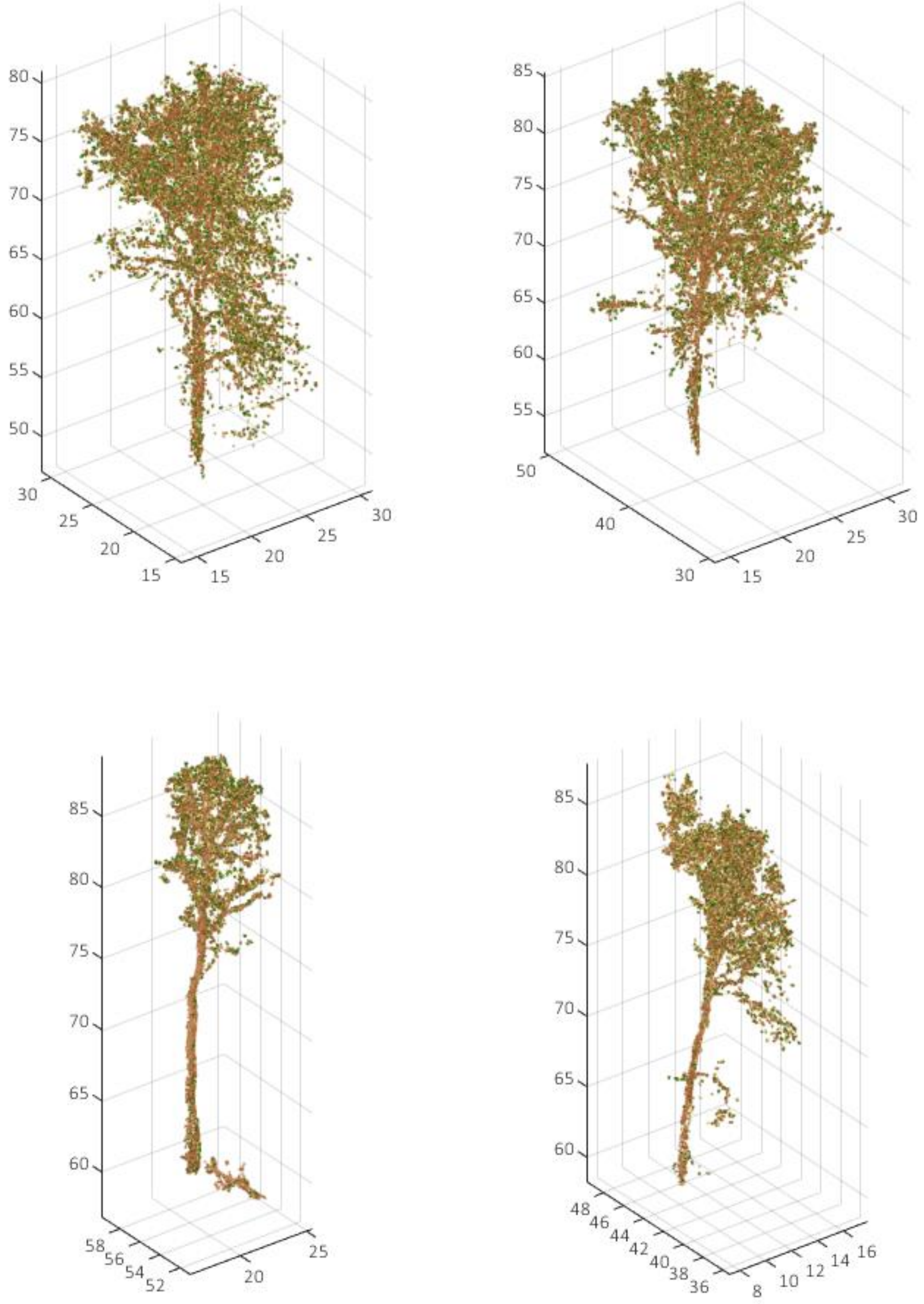


Figure 8-14: Separated trees from UAV-LS Point Cloud.

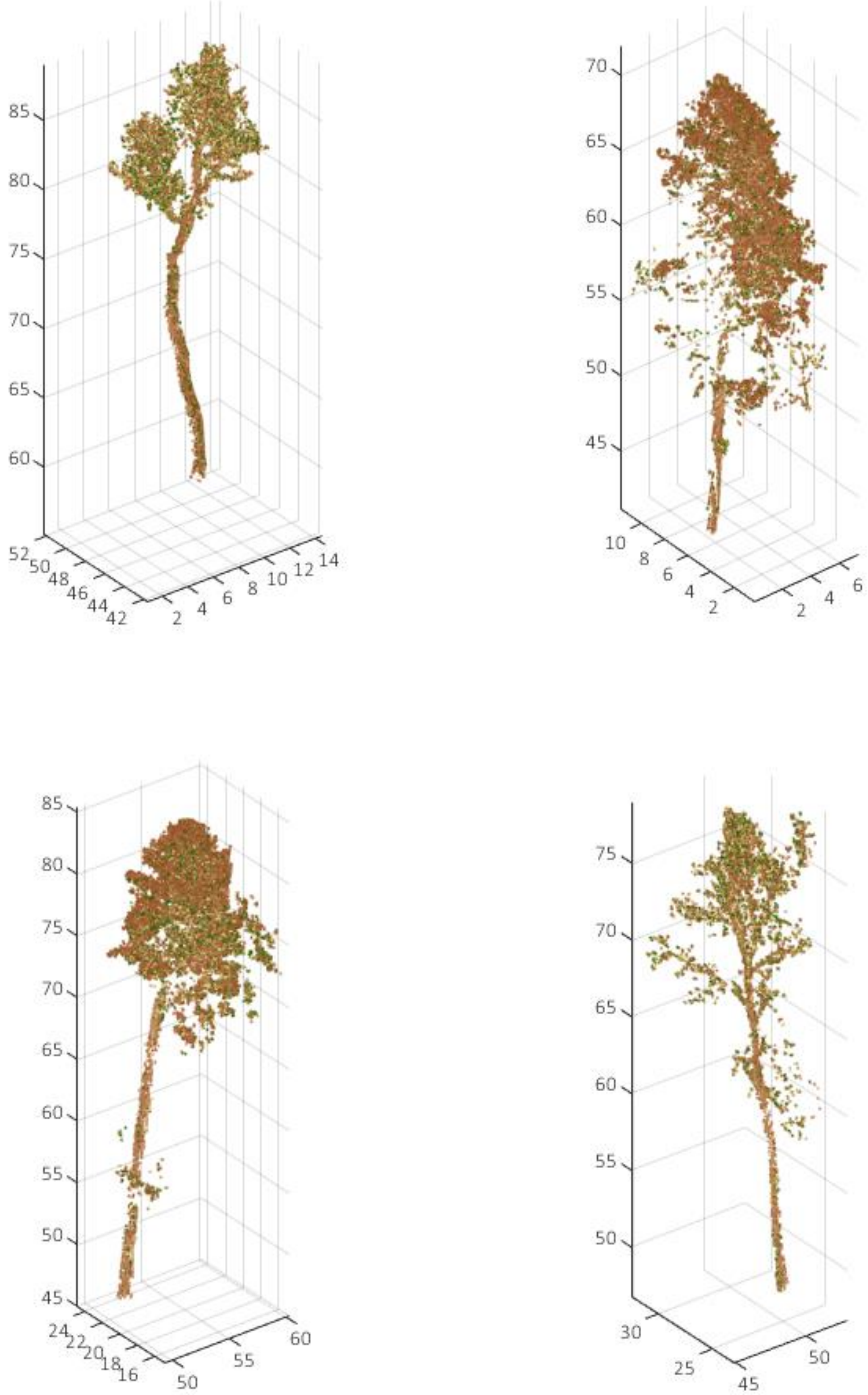


Figure 8-15: Separated trees from UAV-LS Point Cloud.

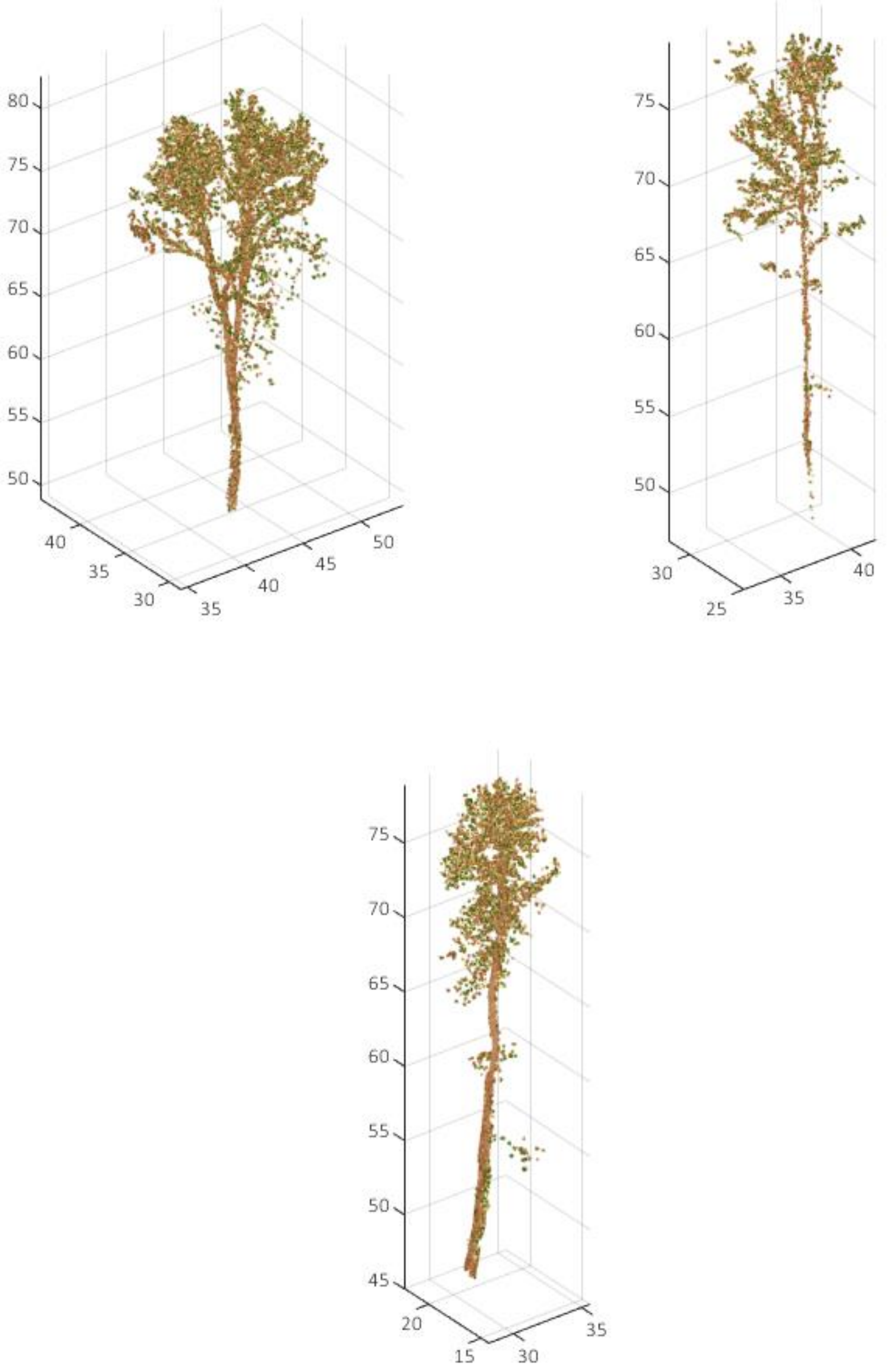


Figure 8-16: Separated trees from UAV-LS Point Cloud.

9 Acknowledgements

I would like to thank my supervisor Felix and co-supervisor Dani for the great support by giving me the chance to pursue my own ideas and encouraging my research. They were always on hand with help and advice.

Additionally, I want to thank the Remote Sensing Laboratory of the University of Zurich represented by Michael E. Schaepman for providing an inspiring environment and supporting this thesis.

A special thanks goes to A. Rudow and his team from the ETH for running the field survey and collecting important data for the verification.

Last but not least I want to thank my family and my girlfriend Hannah for their infinite patience during my endless hours of programming and writing.

Personal Declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

Zürich, 27.10.2018

Place, Date Signature