AN ABSTRACT OF THE THESIS OF

Chu Qi for the degree of Master of Science in Sustainable Forest Management presented on December 9, 2020.

Title: Individual Tree Detection and Measurement from Terrestrial Point Cloud

Abstract approved: _____

Bogdan M. Strimbu

Remote sensing forest inventory gained increased attention in the last decade triggered by the decrease in the price of sensors and the explosion of data availability and formats. Evermore, the constant advances in the hardware processing the data emphasized the necessity to develop algorithms to extract forest relevant information from the images or point clouds acquired using remote sensing techniques. The present research develops a method to identify individual trees and estimate stem dimensions from a terrestrial point cloud by integrating three procedures: semantic segmentation, density-based clustering, and sector-based attribute measurement. The semantic segmentation, implemented by modifying the PointNet++ algorithm, classifies the points as ground, stem and crown, whereas the density-based clustering, implemented by adjusting the DBSCAN algorithm, delineated individual trees from the point cloud. To test the algorithm's robustness, universality and accuracy I have used two lidar datasets: the multiple scan international benchmarking and the McDonald-Dunn Research forest. For the international benchmark data, the dominant and codominant trees were more than 95% correctly identified, whereas the completeness was at least 94%, except for one of the plots which was 84%. Most of

the missing trees are caused by boundary effect and inconsistent point density, but the proposed method still outperformed the state-of-the-art tree measuring algorithms using point clouds. The bias in estimation of the diameter at breast height (DBH) is < 3 mm (similar to the best results of 1 mm), except in one plot where the bias reached 6.2 mm. However, the variability of the proposed method is almost half (i.e., <14 mm) compared with the best result (i.e., 20 mm). The situation was mirrored for height estimation, with bias <7% (except for one plot which exhibited 11% bias). The application of the algorithm to the McDonald-Dunn Research Forest supplied a completeness of 94.25% and a 96.47% correctness, even though the algorithm was developed on completely different species and the trees were not seen from all sides (i.e., incomplete description of the stem). The diameter bias at 1.5 m and 2.5 m was at most 1.3%, confirming the accuracy and generality of the algorithm. The algorithm provides operational worthy results in tree recognition and diameter estimation for point clouds that almost completely describe the trees. However, the algorithm seems to be sensitive to the stem coverage with points, which should be addressed in future research.

©Copyright by Chu Qi

December 9, 2020

All Rights Reserved

Individual Tree Detection and Measurement from Terrestrial Point Clouds

By
Chu Qi

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented December 9, 2020
Commencement June 2021

Master of Science thesis of Chu Qi presented on December 9, 2020

APPROVED:

Major Professor, representing Sustainable Forest Management

Head of the Department Forest, Engineering, Resources and Management

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Chu Qi, Author

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to all my committee members, especially for Dr. Bogdan Strimbu. He encourages me when I encounter challenges, inspires me when I am confused, provides practical advisory and funds me when I meet problems in academic research. Also, Fuxin Li showed deep knowledge in deep learning area and introduced me state-of-the-art technology in computer vision area. Michael Wing taught me fundamental knowledge in aerial survey, and I learned critical thinking from Jim Kiser. As my previous and current Graduate Council Representatives, Bo Zhao and Yong Chen provide suggestions from geography and applied economy areas, which makes me understand the connections between the different disciplines. All my committee members have a great impact on me for these years, both academics and thoughts. I cannot finish this work without you. Also, Temesgen Hailemariam taught me basic information about forest inventory and gives me chance to experience real forest work. Thank you!

I addition, as an important part of my life, my wife Jing Li provide strong supports. She can manage our family well, leaving me more time on research. Also, my mother sponsors on daily cost. You both help me establish great confidence in research area. Thank you!

Moreover, as an international student at OSU, I meet a lot of real friends, including Bryce Frank, Cory Gams, Al Pancoast, Quinton BigKnife, as well as the international student service. You all helped me these years and made me understand the culture of U.S. Thank you!

CONTRIBUTION OF AUTHORS

Dr. Bogdan Strimbu contributed to all chapters of the thesis. Dr. Fuxin Li and Wenxuan Wu contributed to chapter 3 in 3-D deep learning. Carver Heine helps review the whole articles.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

1 Introduction

Researchers developed a plethora of methods to monitor and measure the growth and yield of trees using contact or remote sensing methods. The first attempt to measure height of the trees was though photogrammetry when scientists analyzed two-dimension (2-D) images from aerial surveys. The advent of multispectral data allowed the development of various indices, such as the Normalized Difference Vegetation Index (NDVI), that helps in tree identification as well as in the assessment of health and vigor of vegetation. The increase in computational power, with significant impact from machine learning, shifted the focus of the forestry research toward a more detailed description of the ecosystems, specifically locating individual trees and classifying species from remote sense images. However, a 2-D nadir view of the forest is limited, as little information below the main tree canopy is known. Furthermore, forest ecosystems exhibit a hierarchical vertical structure, which reduces the ability of the nadir 2-D images to capture stems or low vegetation. Therefore, researchers enhanced the 2-D images with three-dimension (3-D) data.

However, most state-of-the art 3-D research are focused on man-made environments, particularly cities or indoor settings, which exhibits a relatively reduced variability between features. The reduced variability of the human built environment catered to the usage of machine learning techniques in identification and dimensional measurement of various features with the rendered reality. However, in the forest environment the similarity between trees challenges the forestry inventory based on 3-D data.

The overarching objective of my research is to develop a method that accurately and precisely estimates the dimensions of individual trees from terrestrial point clouds. The method includes recognizing stems, locating individual trees and computing multiple attributes of the

stems. To assess the new forest inventory procedure, I compared it with the state-of-the-art approaches currently used for measuring trees from terrestrial laser scans. The comparison is focused on accuracy, precision, and feasibility for forest inventory.

2 Literature Review

Typically, there are three approaches to represent real entities as a 3-D object: with voxels (i.e., cubes with a predefined side, like pixels but operating in a 3-D space), with meshes (i.e., multiple surfaces positioned in a 3-D space), and with point clouds (i.e., points with three coordinates). The voxels have the advantage to describe the shape of an object accurately and fast, but the details depend on the size of voxels. The meshes are suited to the surface of the 3-D objects, most of the time with high precision, but lacking accuracy, as they are confined to the exterior of the object. Furthermore, the meshes smooths the object, which reduces the accuracy of 3-D representation. The point clouds are simply a set of points with 3-D coordinates, each one indicating the existence of matter at the (X, Y, Z) location. Among the three approaches that renders reality in 3-D space, the point clouds offer the most complete picture of an object, though noise and incompleteness are the main problems. Point clouds can be generated with active sensors, such as lidar, or with passive sensors thru computer vision techniques, such as structure from motion or simultaneous location and mapping. Structure from motion and simultaneous location and mapping algorithms reconstruct the environment with multi-view images; therefore, the completeness of the point cloud depends on the overlap of the image and the light available to the sensor, so darker images are processed with limited success. Lidar is produced by laser scanners that send laser beams, measure the reflection time and computes distance between sensors and targets. Stationary terrestrial laser scanners (TLS) can accurately and precisely describe reality with point clouds (Cabo et al., 2018; Fang and Strimbu, 2017, 2019; Hackenberg et al., 2015; Raumonen et al., 2013), but require an expensive device, and because of the slow process of data acquisition are not feasible for forest inventory. However, mobile TLS are becoming more popular in forest inventory because they are fast, and in many

instances cheaper than stationary TLS (Cabo et al., 2018; Garms et al., 2020). In my study I have used both stationary and mobile TLS, to assess the ability of the algorithm to operate across multiple data acquisition platforms.

A significant amount of effort was dedicated to automatically identifying individual trees and measuring their attributes, such as diameter at breast height (DBH), tree height, location, taper and biomass, from remotely sensed data (Liang et al., 2018a). Yang et al. (2016) creates multiple 2-D layers and locate individual trees by checking point density of different layers. An alternative method is creating voxels in a point cloud, selecting a representative point in the voxel, and checking the distance or flatness between points to identify stems (Liang et al., 2018c). Xi et al. (2012) transfers points into voxels and counts voxels distribution, whereas (Hackenberg et al., 2015; Huang et al., 2011) locates stems by detecting clusters in a 2-D projection plane. Irrespective of the proposed method, no pre-existing knowledge on stem and non-stem points are assumed. If stem point density or flatness is no different than ground or crown points, the results are below expectations due to the reduced number of points needed for measurements. The impact of point numbers on the estimated dimensions is obvious for the two major stem modelling approaches: cylinder fitting and circle fitting at different height. The most common cylinder fitting method is a random sample consensus or RANSAC (Liang et al., 2018c; Olofsson and Holmgren, 2016). Arguably, the most popular circle fitting strategies are least squares, RANSAC and Hough transform (Conto et al., 2017; Kuželka et al., 2020). However, each strategy comes with its own weakness. Least squares method allocates the same weight to each point, which means outliers can significantly influence results, whereas Hough transform requires knowledge of circle radius. Some methods use brute-force to search a radius (Kuželka et al., 2020), but the precision depends on the interval between radii. As a robust fitting strategy,

RANSAC has an acceptable performance, but it cannot always reach the optimal solution if outliers have a competitive size with stem points.

A possible approach to overcome the lack of existing knowledge is to use a twostep approach: first, classify the point clouds into forest inventory relevant classes, even if the initial results are not satisfactory, and second, improve the initial classification, followed by individual tree measurements. In 2017 the PointNet++ algorithm was launched (Qi et al., 2017), which solves the classification of irregular features within the point clouds. Before PointNet, three methods were mainly applied for point cloud classification. The first method directly applies convolution neural network (CNN) on each point, which is computationally intensive and requires significant memory resources. Since the number of points describing a forest is usually more than 10 million, this approach is infeasible. The second method converts the points into voxels, and then classifies the voxels, which sacrifices the details available in the point cloud. Furthermore, if the voxel size is small, it also claims large memory; whereas, if the voxels are too large, the classification cannot supply satisfactory performances, usually expressed at millimeter level. The third method projects the point cloud into 2-D images, classifies objects in 2-D image, then re-projects it back. The last method is currently one of the most popular methods in tree segmentation. However, the direction of the projection is an important parameter, as if the projected images do not cover enough features, results may be not satisfactory. In this study I used PointNet++, as it does not alter the point cloud but executes a semantic segmentation at point level. The PointNet++ semantic segmentation will help in identification of individual trees by providing knowledge on the stem.

3 Method

The method that I developed to identify and measure tree dimensions from TLS contains three components: 1) raw classification of point cloud, 2) individual tree detection, and 3) estimation of stem attributes (*Figure 1*). To ensure feasibility of the computation, a data-preprocessing phase is needed, namely data thinning, which reduces the number of points if the point cloud is too dense. Different from the most existing tree measurement method based on TLS, the present method first classifies the point cloud with the PointNet++ algorithm as ground, stem, and crown, which is followed by individual tree detection, which locates each tree by clustering points according to stem point density. The method concludes by computing the stem attributes with a sector-based method. The algorithm was implemented on a PC with Ubuntu 18.04 operating system and equipped with an Intel i7-9700k CPU, a Nvidia Geforce RTX 2080 Ti and 32G RAM Memory.

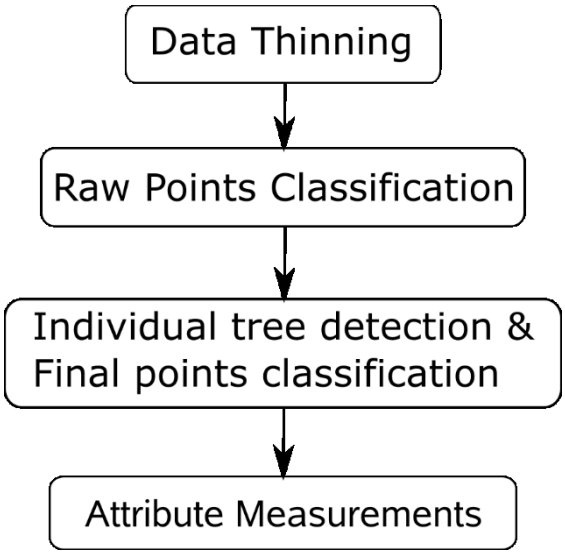

Figure 1. *Method Overview*

3.1 Data

The performances of the proposed method to estimate dimensions of individual trees from TLS were assessed on two datasets: one used as a benchmark for many studies (Liang et al., 2018a) and one used for comparison of various remote sensing platforms suitability for forest inventory (Garms et al., 2020). The two TLS differ not only by the intent but also by location and method of acquisition. The benchmarking data is from a southern boreal forest in Finland, which are subsequently referred to as the benchmark data based on the title of the paper presenting the data (Liang et al., 2018a), whereas the data used for platform comparisons from the American Pacific Northwest, namely from the McDonald - Dunn Research Forests in Oregon, USA, subsequently referred to as McDunn. Furthermore, the method of acquiring the data is not the same for the two datasets, as the benchmark data was created using stationary TLS whereas the McDunn data was obtained using a mobile TLS.

3.1.1 International Benchmark TLS Data

To improve the persuasiveness of many automated forest inventory approaches researchers from 18 organizations launched an international TLS benchmark project in 2014 (Liang et al., 2018a). The project took place in the boreal forest from Finland, where six plots of approximately 1000 m^2 were scanned (i.e., 32 m x 32 m). The major species present are Scots Pine (*Pinus sylvestris L*), Norway spruce (*Picea abies L*) and Downy Birches (*Betula pubescens Ehrh.*). The authors named the plots as easy, medium and hard, according to the tree density and diameter distribution. However, in this study only the easy and medium plots were used, to ensure compatibility of the benchmark data with the McDunn data (Table 1). Each category had two plots, and for each tree the location, DBH and height was recorded (*Figure 2*).

Table 1: Summary statistics describing the four plots from the TLS benchmark data (Liang et al., 2018a) used in this study. The values for DBH and tree height show the mean followed by the standard deviation

Plot	Plot label	Tree Density	DBH	Tree height	# Points
		stem/ha	cm	m	Million
1	Easy	498	22.8±6.6	18.7±3.9	111.0
2	Easy	820	15.9±6.9	13.7±4.0	113.7
3	Medium	1445	14.8±7.3	15.4±6.8	119.9
4	Medium	761	19.6±14.1	16.1±10.2	129.4

The plots were scanned in the spring of 2014 with a Leica HDS6100, which uses a continuous wave of 650–690 nm to measure the distances with an accuracy of 2 mm at 25 m from the scanner. The data acquisition was set to "High Density" mode, which gives a point spacing of 15.7 mm at 25 m from the scanning location and produces a point cloud in approximately 3 minutes/location. The scanning device was positioned at five locations throughout the plot, such that most of the trees would have the stem surrounded by points. For consistent representation of the trees with point clouds within the study (i.e., benchmark and McDunn data), only the data called "multiscan" from the international benchmarking study was used.

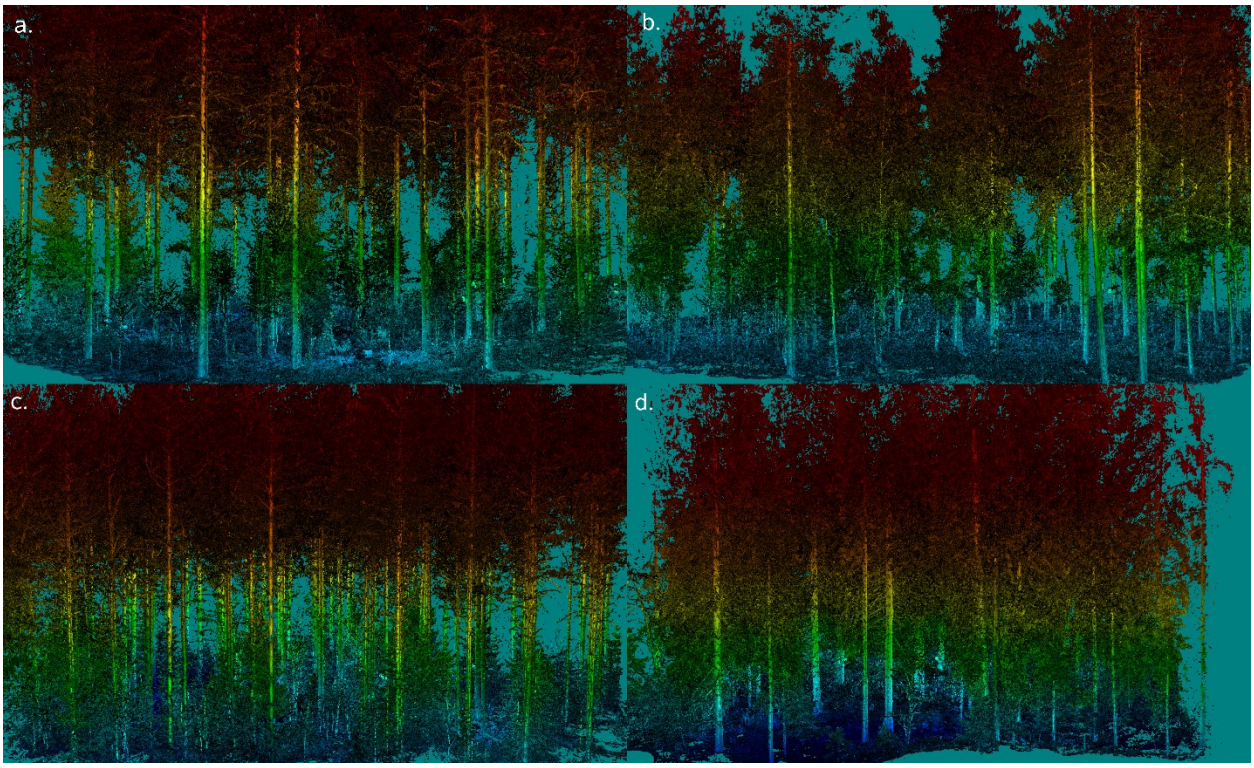

Figure 2: *Lidar point cloud showing the four benchmarking plots. a. plot 1 b. plot 2 c. plot 3 d. plot 4*

3.1.2 McDonald - Dunn Research Forests Data

The McDonald Dunn Research Forest data was acquired in March 5, 2018, with a Velodyne HDL-64E unit installed on a Toyota Tacoma. The system was equipped with a Topcon IP-S2 HD GNSS-aided inertial navigation system. The Velodyne HDL-64E, which has 64 channels, and can record approximately 2 million returns per second coming from a range of at most 120 m. An area of 2.7 ha was scanned with a total number of more than 81.2 million points (Garms et al., 2020). The trees were located on both sides of a road, positioned at the top of a hill, which allows a useful rendering of the forest with lidar point clouds.

The area scanned contains on 86 mature Douglas firs (*Pseudotsuga menziesii Mirb.*), with the height of the dominant and codominant trees of approximately 41 m and an average DBH

around 73.0 cm. The sensor recorded heights around 30 m, which significantly underestimated the actual tree heights. Not only was the height was underestimated, but the canopy was incorrectly created based on the point cloud (*Figure 3*), as multiple tops for the same tree were created by the terrestrial point cloud. Therefore, identification of trees using above canopy algorithms supplied erroneous results (Garms et al., 2020).

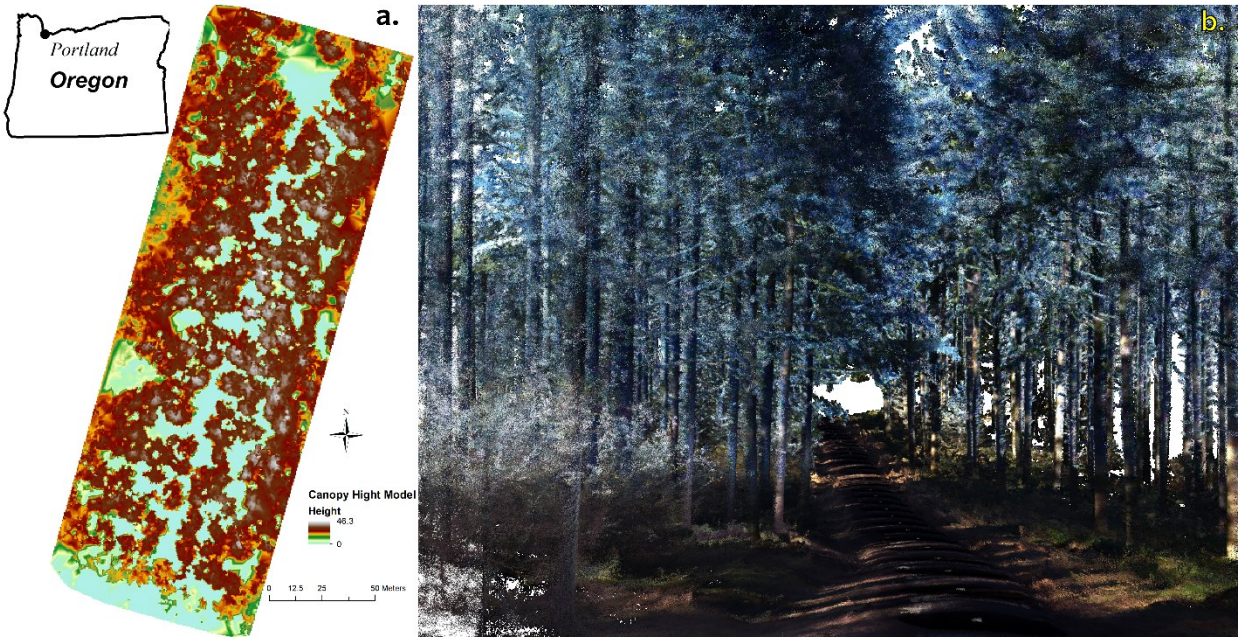

Figure 3: *McDunn lidar data. a. Location of the area and the canopy height models from the mobile TLS, (individual tree crowns cannot be delineated accurately). b. colored lidar point cloud (the blueish color of the canopy was allocated by the point coloring algorithm, even that the leaves were green).*

3.2 Individual tree identification

The procedure developed in this study starts by classifying the point cloud using PointNet++, which requires a training dataset. The training dataset is a point cloud that contains only points that belong to a predefined class. Because our study is focused on estimation of the stem dimensions, we defined three classes: stem, ground, and crown. In actuality, the class "crown" does not contain only points located in the tree crown but also shrubs and the stems of

small trees. However, for the simplicity of the discourse we called the non-ground and no-stem points “crown”, as most of the points in this class are in the crown. We labeled the points with a four-step process: 1) crop target trees; 2) separate the trees into sections that are in the same class; 3) label the points as stem (i.e., code 1), ground (i.e., code 3), and others (i.e., code 2); and 4) merge the labeled data. We labeled 10 trees, considered representative by visually inspecting the point clouds, with QTM version 8.11 (Applied Imagery, 2017). Half of the labeled trees were individually separated (i.e., detached from their neighbors) and half were grouped trees (i.e., a neighborhood of five trees)

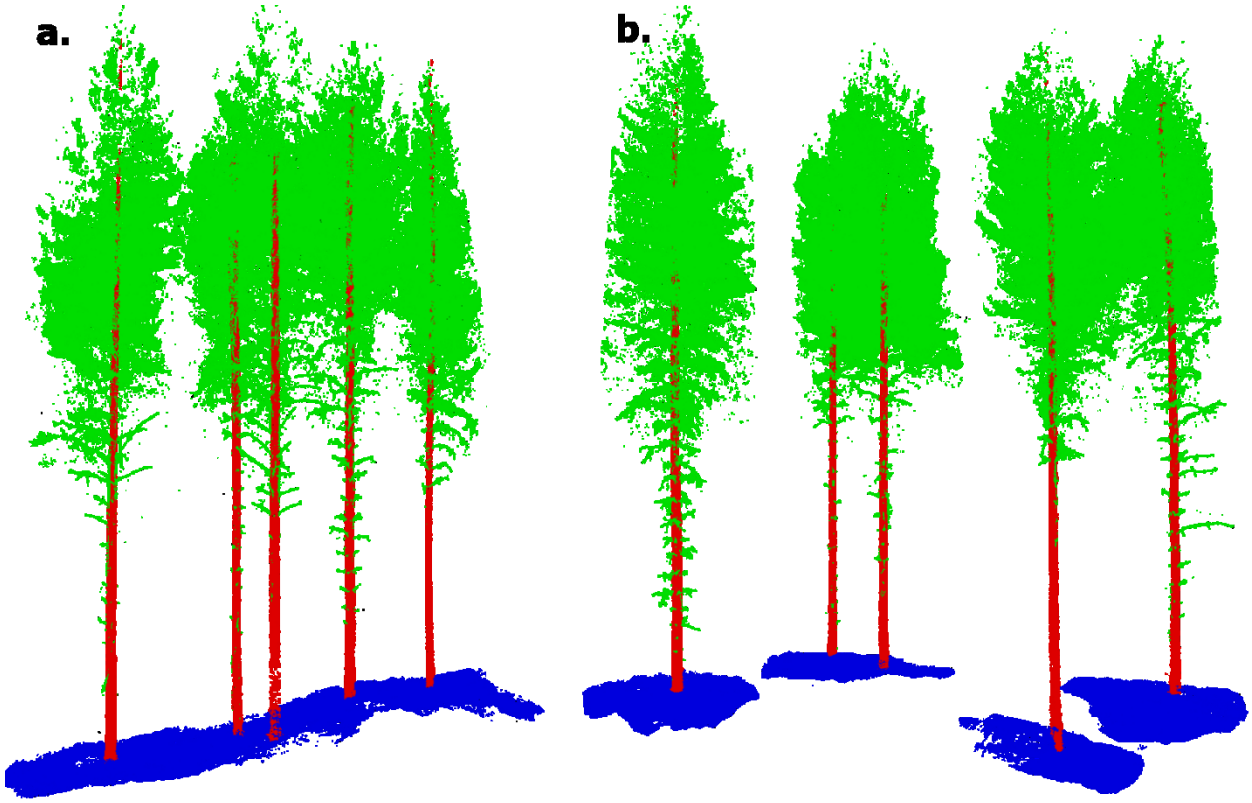

Figure 4: *Labeled tree used for training the PointNet++ network. a. the group of 5 neighboring trees b. the 5 individual trees. The stem is red; crown is green; ground is blue.*

3.2.1 Data thinning

TLS data contains hundreds of millions of points, which are not practical to be classified. Therefore, before starting point classification, a data thinning is applied to ensure feasibility of the subsequent algorithms. The points that will be used further in the classification should be representative to the entire point cloud, as they were used to train the network from the PointNet++. The simplest procedure of sampling a representative point cloud is by using a random based procedure (Cochran, 1977). Therefore, two sampling strategies were investigated: the farthest point sampling (FPS) and simple random sampling (SRS). The choice of FPS is natural, considering that FPS can describe the shape of an object without sacrificing the necessary information (Qi et al., 2017). Intuitively, FPS is a stochastic heuristic that places the sample points in the least-known region of the sampling domain (Moenning and Dodgson, 2003). The FPS starts with a random point, then marks the farthest point from it as the second one and repeat this process until all the points are ordered according to the farthest criterion. However, FPS needs to compute all pair-wised distance in every loop, which is computationally expensive. Because of the size of the datasets encountered in forestry applications, it is not feasible for real TLS data. SRS, on the other hand, is much faster than FPS, but important points (i.e., points describing specific features or contributing to subsequent operations) may be lost during sampling. For example, some stems of short trees that are far from the scanner would lose their shape information during sampling because of the randomness. However, due to the large number of points, rather than unpractical FPS, I found that SRS was functional for this project. In general, each original TLS benchmark dataset had approximately 120 million points. The SRS thinned datasets contained 12 million points (i.e., >10,000 points / m^2), which is not only

close to the operational capacity of the hardware, but is dense enough to provide the needed details for accurate forest inventories.

3.2.2 Point Cloud Raw Classification

The classification of the point cloud, even with errors, is based on the assumption that existing knowledge of the stem enhances accuracy of points clustering into individual trees and parts of the tree (i.e., stem and crown). False positives and false negatives are common in 3-D classifiers, and by using a coarse classification we expect that the number of unnecessary outliers will be significantly reduced. Evermore, the positive identification of points of stems can represent a raw map of tree location, even when false positives are present (Strimbu et al., 2019). However, I developed a method to eliminate the wrongly classified points, which is presented in detail in section 3.3.

Different from structed pixels in 2-D images or voxels in 3-D space, point clouds have an irregular feature, which makes tracking patterns difficult. On the other hand, point clouds possess two important features, namely permutation invariance and transformation invariance, which can enhance point classification. Permutation invariance is the property of an object to remain the same no matter how the point order changes, whereas transformation invariance refers to an object ability to holds its features following rotations or translations. The two properties, particularly permutation invariance, contribute to feature extraction, ensuring that similar objects described with different number of points can be represented by PointNet++ into similar objects in future classification. To ensure representativeness and train the network within the PointNet++, an area of interest and the simple symmetric algebraic functions (e.g., maximum, summation, or average) are used to identify patterns. To preserve the geometric features of the point clouds, before symmetric functions are applied, the PointNet++ adds a multilayer

perceptron (MLP). The MLP is a type of artificial neural network composed of multiple layers (Zell et al., 1994), within which the information moves only forward, from the input to the output nodes, and the connections between the nodes do not form a cycle. If there are N points in a point cloud, each with three coordinates (x, y, z), the data used in the MLP will be represented as a n×3 matrix. Because MLP increases dimensionality, redundant data can be generated that decreases the effectiveness of the PointNet++ symmetric function.

To enhance the ability of detecting local features, PointNet++ employs hierarchical learning (*Figure 5*), which uses balls of multiple sizes to detect features. A ball defines a region within which representative points are selected. When smaller balls are used to extract representative training points, the farthest point sampling method is employed. Once the training is completed, PointNet++ groups the trained points from every ball of the same size, then re-samples them, and finally retrains the sampled points in the next hierarchy; a process that reduces the size of training data. Because of the size of the data, I have adjusted the PointNet++ by first focusing it on smaller regions of the forest stand, which will be large enough to encompass sufficient points for valid classification but small enough such that computations are executed in a feasible amount of time with hardware that can be acquired by most companies. I designed such an area as a prism that would be randomly positioned throughout the forest stand. The random location of the prism inside the stand should not preclude the chance of each point being selected at least once. Given the point density of the TLS, which has at least 1000 points / m^2, an area of 5 m × 5 m provided an adequate number of points to be classified (i.e., at least 25,000 points). The sample size of the randomly distributed prisms can be as small as 40 (i.e., 1000 m^2 / 25 m^2), which will provide a chance of selecting every point of 2.5%. To increase the probability of selecting a point, I have quadrupled the sample size, which increased the chance of

a point being in 25 m^2 prism to 10%. Within each prism a predefined number of points were sampled, the number determined by the hardware. Given the configuration used in this study, the number of points selected was 8192. After the selection of the 8192 points, the PointNet++ algorithm was implemented (*Figure 5*).

// Adjusted Point Net ++

Import *Point Cloud*

Decide the number of sampling boxes: b

Decide the number of hierarchies: h

Decide the number of centroid points: n_j (j from 1 to h)

Decide the number of points around the centroid points: k

Decide the sizes of the query ball: r

Sample *the point cloud with randomly positioned Bounding Boxes (5m × 5m × infinity)*

For *each Bounding Box*

Sample *n points randomly (n=8192 max memory given the hardware)*

For each *hierarchy*

Sample *n_j points, j from 1 to h, called nj centroids, from the previous sampled points (n for j=1, n_1 for j=2, n_2 for j=3 etc)*

Group *k points within the 5 x 5 box around each n_j centroid within the r meters query ball (k=32)*

Apply *PointNet*

Output *one representative point in each ball*

//PointNet Semantic Segmentation

Import *Point cloud*

Apply *MLP (64, 64)* *//MLP with two layers, each with 64 nodes*

Output *Local Feature (n×64)*

Apply *MLP (64,128,1024)* *//MLP with three layers, 1st with 64 nodes, 2nd with 128 nodes and 3rd with 1024 nodes*

Output *LayerData (n×1024)*

MaxPool *LayerData to Global Feature (1×1024)*

Replicate *Global feature by n (n×1024)*

Combine *Local Feature and Global Feature*

Output *CombineFeature (n×1088)*

Apply *MLP (512,256,128)* *//MLP with three layers, 1st with 512 nodes, 2nd with 256 nodes and 3rd with 128 nodes*

Output *Score*

Figure 5: *Pseudocode of the raw point classification based on the PointNet++*

A point cloud can be classified with a multitude of methods based on objectives, such as semantic segmentation, object identification or object part classification (Qi et al., 2017). To identify the stem points, I have executed a semantic segmentation, as implemented in PointNet++, of the point clouds, which supplied a raw classification, meaning many points were erroneous classified. However, identification of some of the stem points represents the starting point that culminates with the final classification, on which only a few points will be misclassified. Because PointNet++ training is limited by the number of points, sub-sampling is executed after the initial sampling. Therefore, I randomly selected 8192 points within a 5 m × 5 m box, which were subsequently classified. Considering tree diameters, the radius for hierarchical spheres are 0.1 m, 0.5 m, 1.0 m, and 2.0 m. The main disadvantage of the hierarchical sampling with a sphere is that the trees located at near the boundaries of the forest stand do not have the same chance of being sampled as interior trees. The reduced probability of sampling for trees located close to the boundaries of the forest stand is the result of the random selection of the 5 ×5 m boxes. Because classified points from different 5 × 5 m boxes intersect, the trees near the edges have lower overlapping probability than trees located inside the forest stand. Therefore, the trees close to the edge of the forest stand have a lower number of points being labeled than the points located inside the forest stands.

The raw classification of the points varies not only with the parameters of the PointNet++ but also with the forest (i.e., easy and medium). In PointNet++, training parameters like batch number and batch size are decided by hardware. On the other hand, in real forest, high complexity forests are expected have lower classification accuracy for the points within the crown class compared with the less complex forest. Nevertheless, the extra information provided by the points labeled as stem will improve the subsequent steps focused on the tree identification.

3.2.3 Individual Tree Detection

Classification of the points as stem, ground and other with PointNet++ was partially successful in identification of the stems as a class, but not as segments, namely individual tree detection. Therefore, to detect individual trees, which is mandatory for individual-tree based forest inventory, I have used the DBSCAN (density-based spatial clustering of applications with noise) algorithm (Ester et al., 1996) applied to the raw classified point cloud. Nevertheless, to accurately identify individual trees and their position within the canopy, I have incorporated DBSCAN into a serial procedure that includes three steps (*Figure 6*): 1) project the raw classified stem points into a 2-D space, 2) cluster stem points within the 2-D space and 3) delineate trees according to their position within the canopy

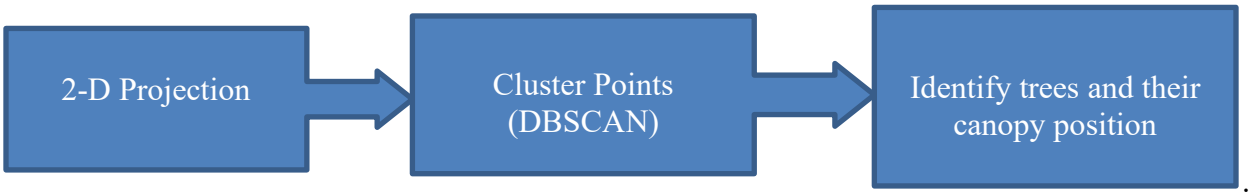

Figure 6: *Overview of individual tree detection*

Raw classification provided a distribution of stem points, important not only to the true positive but also the true negative and false positive points. Since most stem points are true positive, I extracted them and projected into a 2-D space, in essence a map of the points with high likelihood of being stem. Because false positive points are still present in the projected map, I have used DBSCAN to identify the points representing individual stems.

DBSCAN starts by calculating the Euclidean distance between points. For each point in the projected map, if the number of surrounding points within a preset distance, called epsilon by the DBSCAN authors, satisfies a predetermined threshold, then the point is defined as belonging

to a cluster, namely a stem. Points that are farther than the preset distance from a cluster but are the nearest points to a cluster are classified as a border point. Otherwise, the point is considered as not being a part of the stem.

DBSCAN has three properties that make it suitable for individual stem identification (Schubert et al., 2017): 1) it is robust to noise, 2) does not require knowledge on the number of stems, and 3) insensitive to shapes of the objects. Because the raw classification contains many false positive points, there are many non-stem points, which must be eliminated to avoid unnecessary calculation and bias the estimations. The robustness to noise of the DBSCAN (Schubert et al., 2017) ensures that most false positive points are eliminated. Additionally, the DBSCAN does not require the input of a preset number of clusters, namely stems, which is different from some popular cauterization methods, like k-mean algorithm. Therefore, DBSCAN is a parsimonious algorithm (i.e., reduced parametrization), as there is no required knowledge about the number of trees in the point cloud. Lastly, data completeness has limited influence on the performance of DBSCAN, meaning that individual trees can be delineated when described with points surrounding the entire stem or only portions of the stem. On the other hand, DBSCAN faces three challenges(Schubert et al., 2017), each associated with different properties of the point cloud. First, for a large amount of data, such as terrestrial laser scans, it is time consuming, as it has an n^2 run time, where n is the number of points. Second, it has even more parsimonious than other procedures, DBSCAN still requires two parameters: the threshold number of points and the preset distance, henceforth referred to as epsilon. Third, if point density changes significantly within the map, it is possible to have classified stems as false negative for having only a few points (e.g., trees close to the stand boundaries that have a semicircular stem description). The impact of points density depends on the distance between trees and laser scans,

as further trees may not be clustered, since the number of points decreases with increasing distance.

To identify a suitable combination of parameters, I assumed that the trees are described with points similarly, as the same method of acquiring data was used throughout the forest stand. Furthermore, the two parameters defining the DBSCAN algorithm are not only interdependent, but they influence the computational performances, as increasing epsilon leads to inclusion of more points in one cluster with the loss in the number of clusters, whereas decreasing it reduces the size of the cluster (i.e., number of points describing a stem). Therefore, I developed a strategy that identifies both DBSACN parameters are identified simultaneously without computational challenges. The strategy is based on the observation that a regular polygon with 48 sides estimates π with precision at least 0.01m, which is assumed to represent a circular shape. Furthermore, the taper for a lower section of a trunk that is 10 cm long exhibits insignificant taper. Therefore, if sections of 1 m length are used to estimate tree dimensions, it is expected that 480 points would suffice. To ensure that the optimal number of points are used by the DBSCAN algorithm, I have implemented a factorial experiment that considered a range of number of points from 100 to 1000, in 100 increments. For each of the 10 values within the experiment, I estimated the epsilon by using the scree method (Tabachnick and Fidell, 2001) applied to the histogram build from pairwise distances between the stem points. To ensure consistency with the number of points, I built the histogram using only the 100^{th}, 200^{th}, … 1000^{th} nearest point, such that a point was associated with only one other point. Therefore, the histogram contains the same number of values as the number of stem points identified by the PointNet++. I selected epsilon according to four criteria: two of them being adjustments of method 1 and 3 from Rencher (2002) to the DBSCAN requirements, namely the 95 percentile will define the epsilon (method

1) and a horizontal line (i.e., the line between two consecutive values in the histogram) has a slope of at most 2° (method 3), and two of them defined by the dimensions of the stems, namely epsilon is approximately half of the average DBH and at minimum 10 cm. I selected the 10 cm value because in many areas of the world it marks a significant ecological stage, namely transition from sapling to forest (Ministerul silviculturii;, 1988; Reynolds et al., 1992), and represents larger trees, particularly dominant and co-dominant, which store the largest amount of value, volume, and biomass within a forest stand.

Once the DBSCAN parameters were established, all the stem points are separated into clusters, namely individual trees. Because DBSCAN operates in a 2-D plane, the point allocation must be translated back into the 3-D space. However, direct conversion into the point cloud of the tree level information provided by DBSCAN can lead to suboptimal results. First, because the false negative points are ignored, and second, considering that the density of points varies along and across the stem, the points from sparse density regions are likely to be classified as non-stem. To ensure that most of the stem points are classified as stems, I did not limit the DBSCAN results to the clustered points (i.e., points allocated to a stem), but mainly identified the location of the stem. To ensure that more points than the ones allocated by the DBSCAN algorithm are considered for stem delineation, I focused on a region, henceforth called region of interest (ROI) defined by the points identified by DBSCAN as individual stems. The points included in the final stem delineation are located inside a rectangle encompassing the circle fitted by minimizing the sum of squared radial deviation of the clustered points (i.e., points deemed by DBSCAN as being stem and belonging to a tree). The actual rectangle used in stem delineation is defined by the center of the fitted circle and twice its radius. I have chosen a value larger than one radius, to comprise some of the stem points that are ignored by the least square method.

Hierarchical structure of the forest ecosystem challenges the tree height estimation, particularly for shorter vegetation, such as shrubs or young trees. Height is a practical restriction for the points in ROI, but the definition of height as the difference between the largest elevation and the lowest elevation of the points inside the ROI is not suitable for delineation of the understory vegetation. Therefore, I developed a heuristic method to estimate the height of a tree. Initially, the largest vertical distance within the ROI is calculated; then points within ROI are divided into 1 m long sections. The number of points in each section of the trunk, starting from the bottom, are checked against a preset value considered sufficient for positioning the tree within the canopy. The preset value depends on the scanning procedure, but it should be at least 6, which is the minimum number of points needed to fit a cylinder. By trial-and-error I have selected 20, as lower values lead to unrealistic trees whereas larger values produced trees shorter than they actually are. When two consecutive sections do not meet the density requirement, the highest point in the upper section is considered the top. The height calculated based on the above procedure allows selection of the dominant and codominant trees, as well as identification of the lower vegetation. The height estimation method addresses the vertical hierarchy of the forest stand but is sensitive to the presence of points on the lower portion of the stem. To ensure that enough points are available to make correct decisions regarding the height, areas one-meter larger than the radius of the fitted circle in the projection data are searched. The larger area is used to identify the ground, which I assumed to be the lowest point found inside the searched area. Once the ground elevation is identified, the process of computing the tree height starts at 3 meters above the ground. I have chosen the value of 3 m, as rarely shrubs grow taller than 3 m, which means that the vegetation above this value is part of a tree.

In brief, the procedure of individual tree detection includes three parts: projection of stem point cloud in a 2-D space and estimation of DBSCAN's parameters, identification of individual trees using DBSCAN algorithm, and identification of the possible stem points according to tree position within the canopy (*Figure 7*).

Segment individual trees

Project the points classified by PointNet++ as stem into a 2-D plane. The projected points would be further referred as "2-DPNetStems".

Compute the pairwise distance between the points from the 2-DPNetStems

Create a histogram using only the x^{th} nearest neighbor from the pairwise distance, further referred as minPts, similar to (Ester et al., 1996; Schubert et al., 2017), where x is one of the hundred values from 100 to 1000.

Select epsilon according to the adjusted scree criterion

Run DBSCAN on the projected points

Input parameters: epsilon and the number of points used to create the histogram on which epsilon was estimated.

Label all points as "undefined"

Assign the value of 1 to the variable identifying stems: s = 1.

For each point p in 2-DPNetStems and label(p) = "undefined" do // Start iteration

$N = \{p_i \in 2DPNetStems | \|p_i - p\| \leq epsilon\}$ *//find the neighbor points*

if card(N) < minPts then label(p) := "non-stem" // label non-stem points

if card(N) ≥ minPts then label(p) := s //allocate the point to a stem

S = N \ {p} // Expand neighborhood of p

foreach q in S do

if label(q) = "non-stem" then label(q) := s //allocate the point to stem s

$N_q = \{p_i \in 2DPNetStems | \|p_i - q\| \leq epsilon\}$ *//neighborhood of q*

label(q) := s //initial labeling of q

if card(N) < minPts then continue // check is stem

$S = S \cup N_q$

Position individual trees within the canopy

Compute the center (x_s, y_s) and radius (r_s), of the circle fitted to points associated to each individual stem as identified by the DBSCAN

Create a prismatic ROIs within PointNet++ classified point cloud centered in (x_s, y_s) with sides $4 \times r_s$, ($x_s \pm 2 \times r_s$, $y_s \pm 2 \times r_s$).

Segment the points within each ROI every meter, were $Seg_{..}$ is the set of points

Count the points within each segment, card($Seg_{s,i}$), where $Seg_{s.i}$ is the segment i of tree s.

Initialize the variable identifying the segments that can be used as trunks: $T_{s,.}=0$

For each segment do

If card($Seg_{s.i}$) ≥20 then $T_{s,i} = 1$

If card($Seg_{s.i}$) <20 then

If card($Seg_{s.i+1}$) ≥20 then $T_{s,i} = 1$ //check the segment above

Else $T_{s,i} = 0$

Set highest point inside tree s as the height in the segment $T_{s,i+1}$

Figure 7: *Pseudocode used to identify individual trees and their position within forest canopy. Card stand for the cardinal of a set and || . || stands for the Euclidean distance*

3.2.4 Dimensional measurements of the stem

Once the ROI and canopy position for each tree was delineated, the steam measurements, namely diameter along the bole, should be executed. However, the ROI contains points that are not stem, therefore, before estimation of tree dimensions, the PointNet++ raw classification must be distilled such that almost all the stem points are identified. Because the taper is assumed to be negligible for short sections of the stem, I assumed that for sections of the stem shorter than 1 m the trunk has a prismatic shape. Considering that cross-section of stems are not perfect circles, I developed a sector-based method to compute a surrogate radius, which is the radius of a circle that best approximates the trunk. The procedure used to estimate the diameter along the stem includes six steps (*Figure 8*): 1) verticalize the trees, 2) fine tune the classification of stem points, 3) segment trees into sections, 4) project points of each section into the 2-D plane, 5) use a sector-based method to select representative points, and 6) compute attributes of interests.

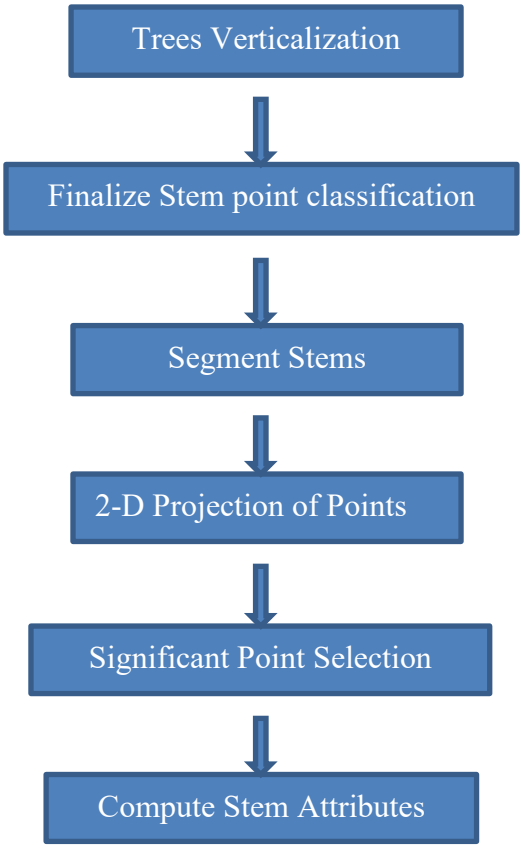

Figure 8: *Flowchart of the procedure used to estimate tree attributes*

The initial step is to verticalize the trees, as the diameter estimation of the stem segements requires verticalization of the trunk. Therefore, leaning trees must be rotated such that they are vertical. To ensure vertical direction of the stems, I selected a portion of 15% to 30% of total height as main stems, and fit cylinders with maximum likelihood estimation sample and consensus (MLESAC) algorithm (Torr and Zisserman, 2000). Different than normal RANSAC (Fischler and Bolles, 1981), MLESAC computes the likelihood to evaluate a consensus set. However, any RANSAC-based cylinder fitting cannot find the best solution because of the number of outliers on stems. Therefore, I enhanced the |MLESAC with a Principal Component Analysis (PCA) - cylinder fitting strategy, which estimates the main orientation of data by minimizing the average Euclidean distance. For real trees, branches can grow in any direction,

which can point the PCA in a direction off vertical. However, if PCA and MLESAC exhibit a small difference in direction, then the cylinder fitting result represents main stem orientation. Once the main stem orientation is established, the entire tree will be rotated.

The second step is focused on fine tuning the stem classification, which because of the PointNet++ raw classification and the ROI selection consists only of elimination of the outliers, namely the branches. I considered a point as an outlier if it is located more than three standard deviations from the mean of positive stem points inside ROI after verticalization. Because the point cloud is created with TLS, most points within ROI are concentrated near stems, which would maintain most of the true stem points. A limitation of this approach is exhibited when the point cloud does not fully cover the tree. In these situations, some stem points are removed because the mean of all stem points is closed to denser region, which contains the stem and branches. Nevertheless, the remaining stem points would ensure that prisms can be fit to the portions of the stem without complete coverage.

Once the stems are verticalized and the stem points identified, I split the trunk into sections of 2 m length and fit cylinders to these sections. The tree separation into 2 m sections targets the estimation of tree axis location and provides an initial radius for future computations. A length of two meters for cylinder fit ensures enough points for locating the section axis inside the stem and practically eliminate the sweep associated with longer sections. I have chosen to fit cylinders instead of circles in the 2-D space because the MLESAC cylinder provides additional orientation information, which does not exist in the 2-D space. The orientation of the fitted cylinder is used to align the sections of the trunk.

For precise estimations of diameter, the two-meter sections are too large. Therefore, I continue segmenting each section into more sub-sections, then project the sub-section points into

the 2-D plane. It should be noticed that the sub-section size has an inverse relationship with the effects of outliers, as shorter sub-sections exhibits larger influence of outliers because a branch usually grows at a particular height. By trial and error, I found that sub-sections with length of a half meter supplies results with accuracy less than 1 cm for DBH. Considering that a 0.5 m long section of the trunk exhibits practically no taper while ensuring accurate results, all the subsequent computations are executed using subsections of half meter.

To estimate tree dimensions, I selected the representative points by developing a sector-based method. Inspired by the derivation of π, I considered that if enough points along stem perimeter are available, then the stem attributes could be computed from these points. The selection of the representative points uses an axis and an approximate radius of the stem. Because cylinders were already fitted for the 2 m long sections, I have used their axis as proxy for the actual the cylinder. To select an approximate radius, I assumed that it should not change significantly from section to section. Since cylinder's radius is sensitive to fitting parameters, I have used a histogram-based method to estimate the radius, which starts by computing the distance between stem points and the axis. For almost all 2 m long sections, the histograms of distances follow a gamma distribution (the Kolmogorov-Smirnov goodness of fit test showed a p value>0.05). Using Rencher's scree method 1, I found that the radius can be estimated within the 90^{th} percentile. The mean of these distances represents the approximate radius. After achieving an approximate axis and radius, I selected the representative points by first dividing the sub-sections into sectors collapsed into a 2-D plane. The number of sections is a required parameter, and I found that a value of 28 ensures DBH values with an accuracy less than 1.0 cm. A larger number of sections increases the computation time and minimally increases accuracy. Using the approximated center (i.e., the projection of the axis on the 2-D plane) and radius, I identified the

sectors with at least one point. To ensure that the stem is represented by sectors with points rather than predicted points, I computed the ratio between the sectors with points and the total number of sectors. If the ratio reaches a preset threshold (i.e., by trial and error I found 80% supplied accurate results), then I selected the representative point of a sector as the average of all points within the sector that are located less than three standard deviations from the mean. The representative points are likely not real points within the point cloud. If the ratio is below the threshold, I increased the radius by multiplying it with a preset constant. An increase of 1.05 was found to produce accurate results. If the radius has increased more than 15 times (i.e., the radius doubled its original value) without achieving the desired convergence of the ratio between sectors with points and the total number of sectors, then randomly move the center and reset the radius.

I computed the diameter by fitting a circle thru the representative points using the least square method, considering that the representative points are distributed in an almost circular pattern. The proposed algorithm is robust to outliers, and it performs as expected even for sparse or incomplete data. I estimated the height above the ground corresponding to the sub-section diameter as the mean height of the representative within the sub-section.

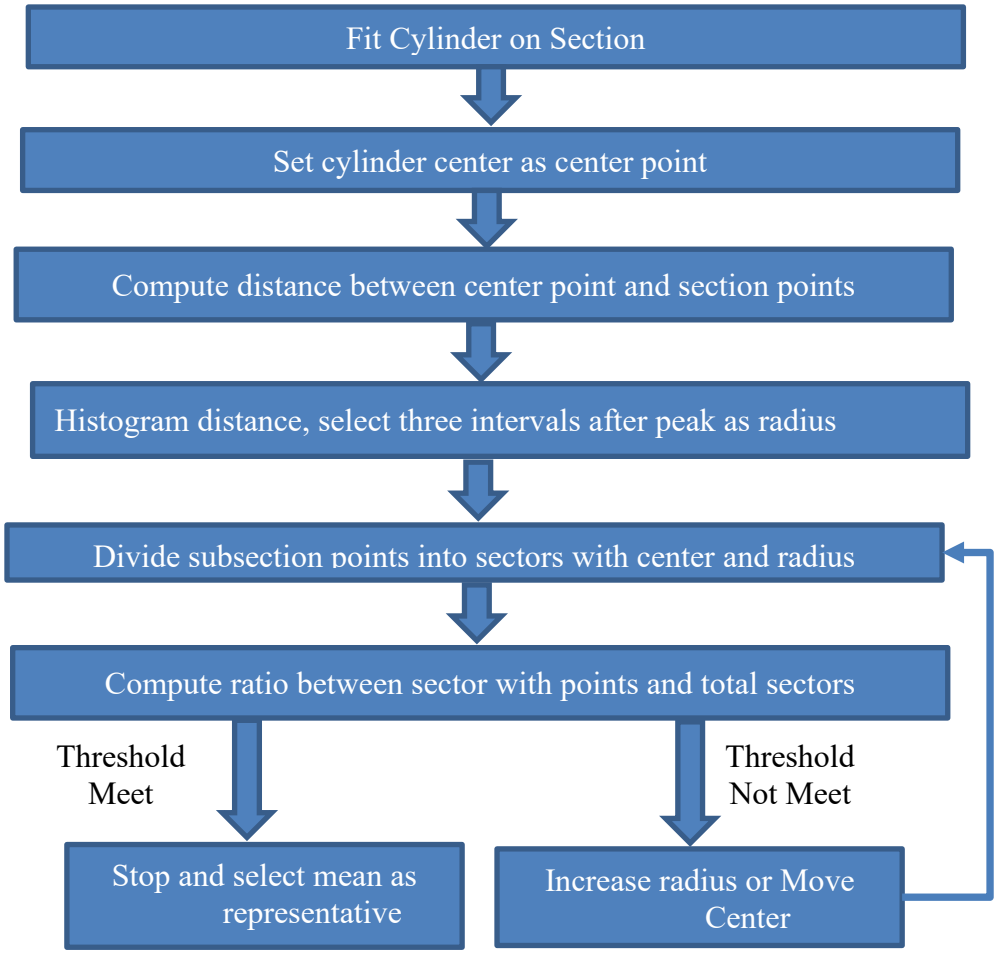

Figure 9: *Sector-based method flow chart for estimation of diameters along the stem*

3.3 Assesment of the proposed forest inventory method

The main attributes of interest in this study are individual trees, identified by their location, and the diameter height pair. Because the benchmark data and McDunn data have the DBH measured, either in the field or at the office, I have use the DBH for assessing the ability of the proposed method to measure tree dimensions. The metrics used to assess the performance of the proposed methods are completeness, sometimes called sensitivity or recall (Powers, 2007), and correctness, sometimes called precision (Saito and Rehmsmeier, 2015), for individual tree detection, similarly to Liang et al (2018b), and bias and root mean square error (RMSE) for

location, DBH and height, similarly to Garms et al (2020). The followings equations were used in evaluation.

$$\boldsymbol{Completeness} = \frac{\boldsymbol{Tree_{match}}}{\boldsymbol{Tree_{ref}}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad \textbf{1}$$

$$\boldsymbol{Correctness} = \frac{\boldsymbol{Tree_{match}}}{\boldsymbol{Tree_{detect}}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad \textbf{2}$$

$$\boldsymbol{Bias} = \frac{\mathbf{1}}{\boldsymbol{Tree_{match}}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{Tree_{match}}} (\boldsymbol{\hat{y}_i - y_i}) \quad \textbf{3}$$

$$\boldsymbol{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{Tree_{match}}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{Tree_{match}}} (\boldsymbol{\hat{y}_i - y_i})^{\mathbf{2}}} \quad \textbf{4}$$

$$\boldsymbol{Bias}\% = \frac{\boldsymbol{Bias}}{\bar{y}} \times \mathbf{100}\% \quad \textbf{5}$$

$$\boldsymbol{RMSE}\% = \frac{\boldsymbol{RMSE}}{\bar{y}} \times \mathbf{100}\% \quad \textbf{6}$$

where *Tree match*, *Tree detect* and *Tree ref* represents the number of trees detected, matched, and reference, respectively; TP, FN, and FP stand for true positive, false negative and false positive, respectively; $\bar{y}$ represents the mean attribute for which bias is computed, such as DBH or height.

I assessed the performance of the algorithm as a combination of completeness and correctness. Therefore, I considered that high correctness, but low completeness is not necessarily a satisfactory algorithm, whereas high completeness and low correctness is also not preferred, as it shows an overestimation algorithm. Consequently, the focus is to obtain an algorithm with high completeness and correctness. For such algorithm, the assessment involves the dimensional and location measurements, which is executed using bias and RMSE.

4 Results

4.1 Selection of the main parameters

The proposed method of measuring trees from TLS and classifying the point cloud relies on three algorithms: PointNet++, DBSCAN and the newly developed sector-based dimensional measurements. Each algorithm depends on many parameters, some impacting the performances and computation time, such as epsilon from DBSCAN, while some affect the performance, such as the number of sectors or length of a section in the sector-based method of estimating the cross-section of the tree. There are two time-consuming tasks that define the procedure: the raw point classification, based on PointNet++, and individual tree segmentation, based on DBSCAN. Depending on the computational configuration, the estimated time for the five areas used in this study the raw point classification is completed in approximately 30 min, whereas the tree segmentation is executed on average in 45 minutes. The parameters of PointNet++ used in this procedure were not tailored to tree identification, as they were based on previous point classification studies (Qi et al., 2017). Evermore, because the raw point classification is not required to have a significant degree of accuracy, the parameter selections are not crucial in the overall performance of the algorithm. However, the parameters defining the sampling strategy of the point cloud to ensure that PointNet++ is operational within the financial constraints is important. Using a trial-and-error approach I found that a box with cross section of 5 m provides enough points to the PointNet++ algorithm without sacrificing the spatial description of the trees and ground. Also, given the dimensions of the trees in respect with the point cloud density provided by the TLS, the size of the spheres used in hierarchical learning were chosen to vary from 0.5 m to 2.0 m, in 0.5 m increments. The selection of the sphere sizes ensured that the point cloud is classified with the accuracy needed by the subsequent steps.The tree segmentation part

of the algorithm plays a pivotal role in dimensional measurement, therefore its parameters require proper selection. The histogram-based procedure of selecting the epsilon and the number of points within the neighborhood (minPts), recommended an almost consistent value for all plots from the benchmarking study, which was used for training the PointNet++ algorithm. The number of points within the neighborhood of a point has a significant impact on the DBSCAN performance not only in term of computation time, but also in terms of magnitude of the epsilon. Values from 100 to 300 supplied significantly different results for epsilon depending on the plot, whereas values larger than 600 significantly increased the computation time without providing superior results. Considering that several of the 480 points were needed to accurately describe a 1m section of the stem (as argued previously) I have chosen 500 as minPts. Furthermore, for 500 points, the scree method pointed to a consistent epsilon, namely 0.1 m for plot 1 and 3, according to the slope and 10 cm criteria, and 0.11 m for plot 2 and 4, according to the 95% criterion (*Figure 10*). Being only one centimeter apart, I have chosen 0.1 m as epsilon.

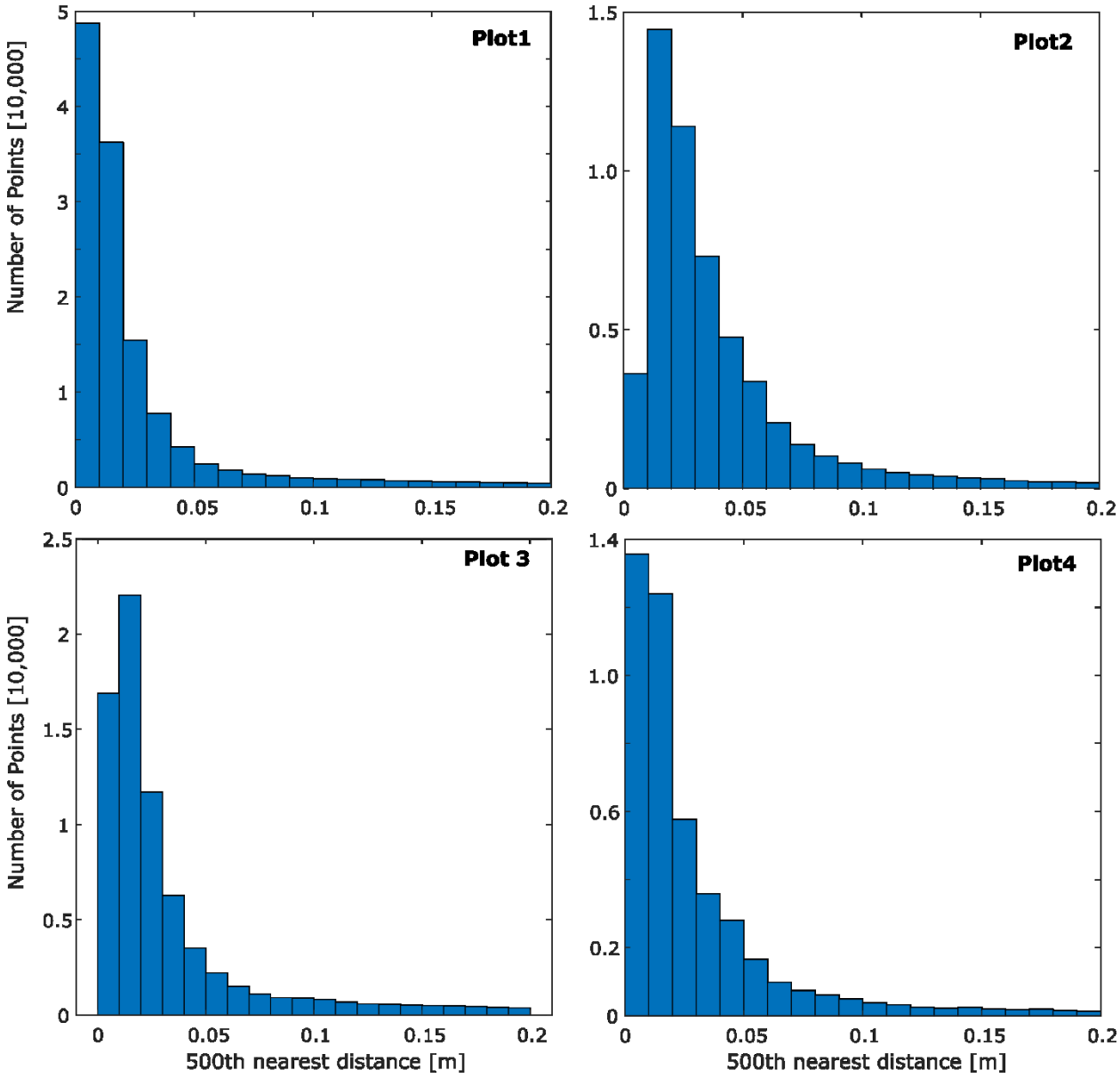

Figure 10: *Histogram of 500th nearest distance in each plot. Larger distance has higher probabilities of not being stem.*

4.2 Point classification and estimation of cross-sectional dimension

The raw classification served as the input for the subsequent individual tree identification and stem measurements. The subsequent steps in the algorithm not only that eliminated most of the points that were wrongly identified as stem but also included points that were initially considered crown when in fact were stem. Depending on the plot, the final number of stem

points increased compared to the raw classification, except for plot 1, where the initial classification supplied accurate results that needed to be distilled rather than enhanced (Table 2). The raw classification was in fact successful in identifying the ground points, as the difference between the raw and final classification was approximately 1%, regardless the plot (Table 2), whereas the number of stems differed by more than 2 % in some cases (i.e., Plot 4).

Table 2: Summary statistics of the point cloud classification for the benchmark data

Plot	Stem [%]		Ground [%]		Crown [%]	
	Raw	Final	Raw	Final	Raw	Final
1	10.9	9.6	64.1	62.6	25.0	27.8
2	7.0	7.9	59.3	59.0	33.6	33.1
3	9.8	11.3	38.7	37.7	51.6	51.0
4	6.3	8.9	45.0	44.6	48.7	46.5

When the raw classification identified stem points for all the trees, even with errors, the fine tuning further improved the allocation of points into the stem class (*Figure 11*, a and b), but did not necessarily identify more trees. However, the fine-tuning classification was important when the original classification produced many false negatives, including missing trees, which was the case of plot 4 (*Figure 11*, c and d). Therefore, the fine tuning will not only improve stem identification, but will also reduce the number of missing trees, which is the case for forest stands that have a large dimensional variability (*Figure 11* d).

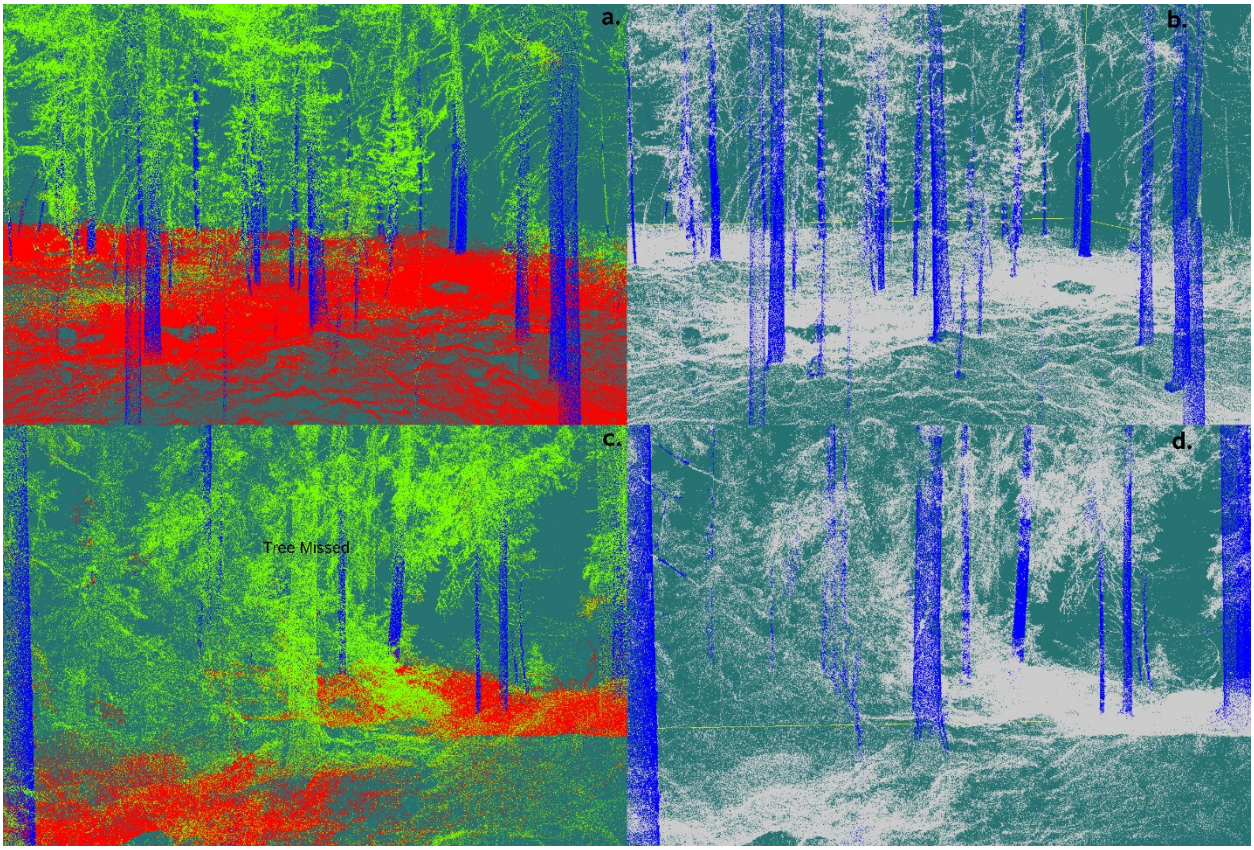

Figure 11: *Impact of adjusting the PointNet++ classification. a. Plot 2 raw classified points b. Plot 2 final classification c. Plot 4 raw classification with a large tree having the stem points classified as crown d. Plot 4 final classification with the stem points identified for all trees. The color scheme is: blue stem, red ground, and green crown.*

The sector-based procedure used for dimensional measurements proved to identify the stem accurately, even in the presence of many branches or non-obvious circularity. *(Figure 12*). Furthermore, the presence of obstructions inside canopy or because of the sensor's location in respect to the tree had limited impact on the estimations, as accurate values were provided even for sections described with almost half circular distribution of points around the stem (*Figure 12*).

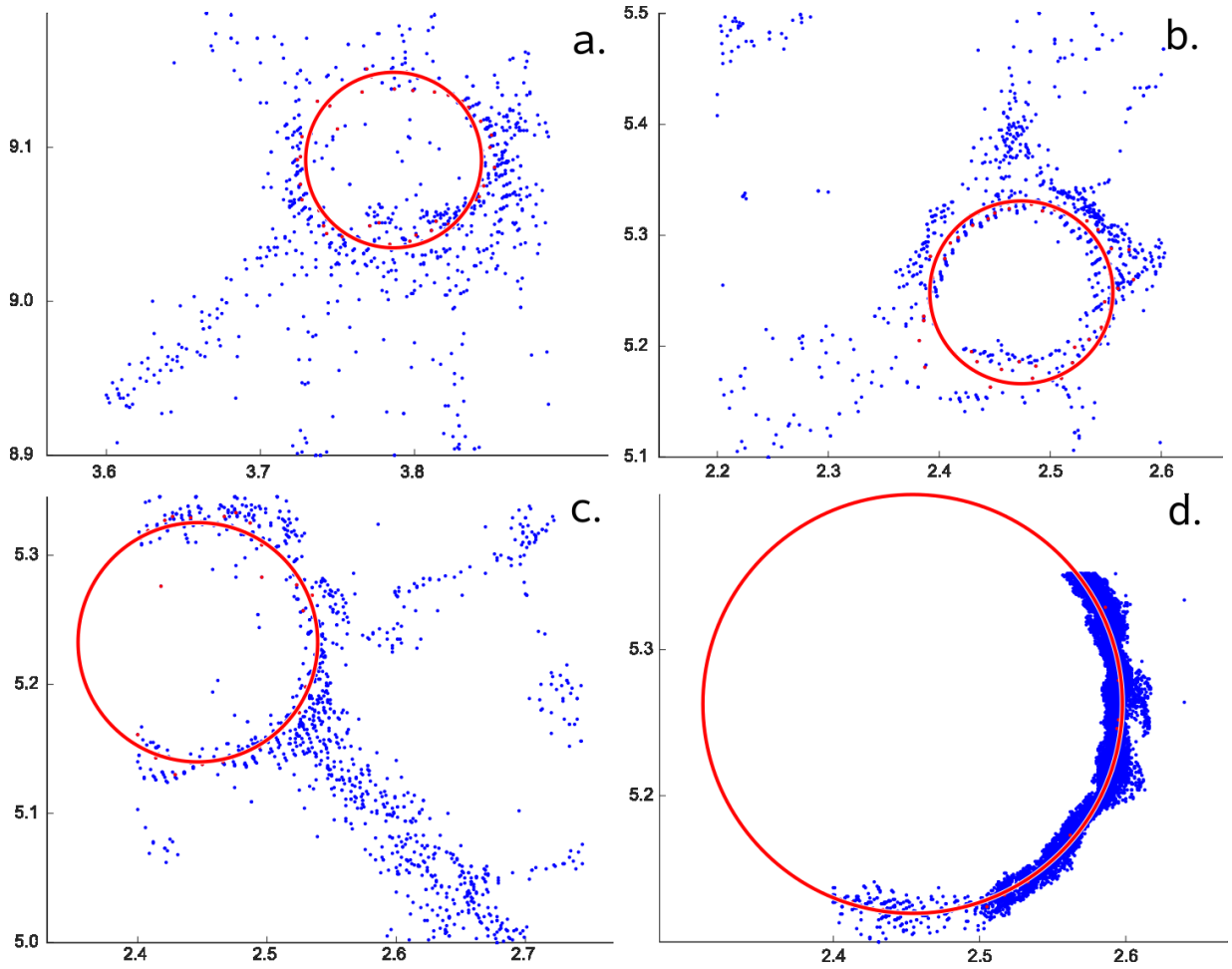

***Figure 12**: Examples of the sector-based estimates showing the algorithm robustness to: a. outliers (points that are not stem); b. erroneous points (inside the stem) c. outliers and sparse data; d. Incomplete data.*

4.3 International TLS Benchmark Data

4.3.1 Individual Tree Detection

Forest inventory is focused on dominant and codominant trees; therefore, the method developed in this study concentrates on trees located in the upper canopy. To differentiate the understory from the upper canopy I considered that trees have a height larger than 10 meters, which is approximately 1/3 of the height of the dominant and codominant trees for benchmarking

data and ¼ for the McDunn data. All of the algorithms tested on the international benchmark data were outperformed by the proposed method in term of correctness (Table 3). The results were almost mirrored for the completeness, as the proposed algorithm outperformed all the benchmarking algorithms, except for plot 4 (Table 3). The conclusions are valid not only for the dominant and codominant trees, but also for the dominated and suppressed trees. The underperformance of the developed algorithm on Plot 4 is the direct result of the reduced number of points for the trees close to the boundary, which were not fully covered points (*Figure 13*). The location of the scanners led to an incomplete description of the trees not only for plot 4, but for all the plots. Nevertheless, the impact was most severe in plot 4 because the trees near the edges had not only few points but also a reduced circularity, given the location of the scanner and the obstruction from the rest of the trees.

Table 3: Completeness and correctness for multiple scan data from the benchmarking study. The Best presents the best results from the benchmarking study whereas the All trees and (Co) Dominant shows the results of the proposed method of measuring trees from TLS for all the trees and for the Dominant/Codominant trees, respectively.

Plot	**Completeness [%]**			**Correctness [%]**		
	Best	**All Trees**	**(Co)Dominant**	**Best**	**All Trees**	**(Co)Dominant**
1	90.4	92.16	97.83	93.6	100	100
2		91.9	96.5		100	94.9
3	88	80.8	94.2	89.2	98.8	95.3
4		74.1	84.6		95.2	91.7

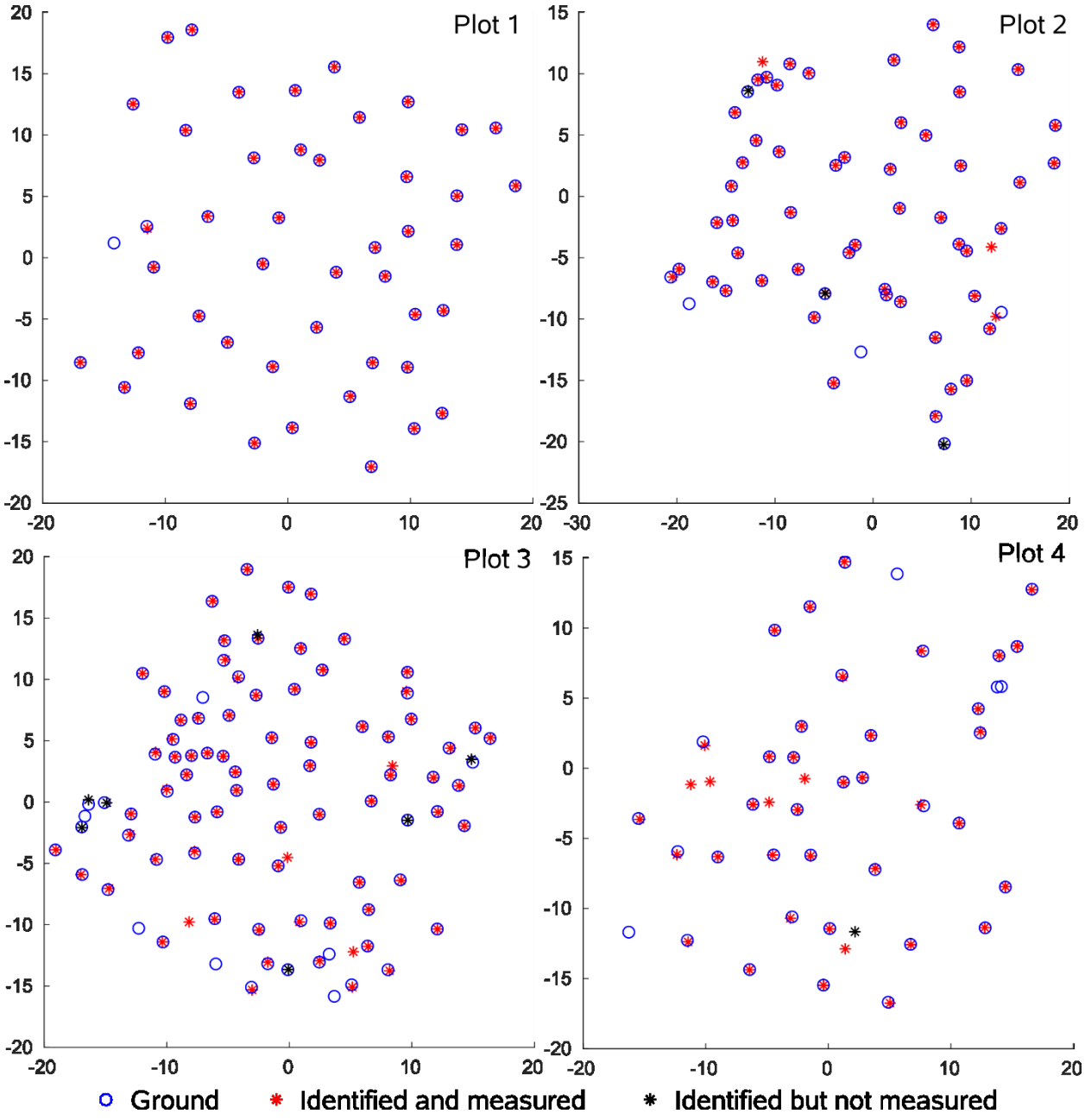

Figure 13: *Detection of dominant and codominant tree from the benchmark data. The black stars are trees that were identified but unable to have the attributes estimated because of reduced number of points.*

4.1.1 Location of the trees and dimensions of the stem

The location of the trees reflects the completeness and correctness of the algorithm, with bias approximately 10 mm to all plots and position within the canopy (Table 4). The outstanding performance of the proposed method compared with the methods from the benchmarking study in respect to tree position, is likely the result of the point classification and of sector-based procedure of computing the cross section of the tree (Table 4). The RMSE on the other hand was larger but comparable with the ones from the best methods considered in the benchmarking study. The large RMSE for location suggests a reduced precision in respect to accuracy.

Table 4: Performance of the proposed algorithm in respect to the location of the trees whereas the All trees and (Co) Dominant shows the results of the proposed method of measuring trees from TLS for all the trees and for the Dominant/Codominant trees, respectively.

Plot	**Bias / RMSE [mm]**		
	Best	**All Trees**	**(Co)Dominant**
1	30 / 20	5.4 / 36.9	5.5 / 37
2		5.4 / 41.8	6.0 / 43.7
3	50 / 25	7.2 / 62.8	7.4 / 63.6
4		11.8 / 78	13.8 / 82.7

The DBH bias was less than 3 mm, irrespective of the plot and size of the tree (Table 5), except for plot 2 which had a bias of 5.5 mm. The relative bias was always less than 4%, except for the dominant and codominant trees from plot 3, which was 6.4%. Compared to the best results from the benchmarking study, the proposed procedure did not always supply better results, but did return comparable values (Table 5). Nevertheless, the error associated with the

proposed procedure was within the margins acceptable by the field forest inventories (Ministerul silviculturii, 1986; Robertson, 2000). The relative RMSE was almost always smaller that the best results found in the benchmarking study (Table 5). However, the best results from benchmarking study are not from the same algorithm, but by two algorithms (i.e., Tree Metrics and SLU), as presented by Liang et al. (2018b).

Table 5: DBH bias and RMSE for the benchmarking data. The least unbiased algorithm from the benchmarking study were the Tree Metrics (abbreviated TM) for the easy plots, which was 100% correct, irrespective of the plot, but its completeness was 35%, and by the SLU for the medium plots, which was correct <95% and its completeness was <90%. The least variable estimates were provided by the TM algorithm, which for the medium plot has a completeness of 30%. All trees and (Co) Dominant shows the results of the proposed method of measuring trees from TLS for all the trees and for the Dominant/Codominant trees, respectively.

Plot	**Bias [mm] (Bias [%]) / RMSE [mm] (RMSE [%])**		
	Best	**All Trees**	**(Co)Dominant**
1	-1^{TM} (-1^{TM}) / 2^{TM} (5^{TM})	2.7 (1.1) / 10.4 (4.3)	2.9 (1.2) / 10.4 (4.2)
2		-5.5 (3.1) / 12.3 (7.0)	-6.2 (3.2) / 11.6 (6.1)
3	0.8^{SLU} (1^{SLU}) / 2^{TM} (10^{TM})	2.0 (1.0) / 13.2 (6.9)	1.2 (6.4) / 12.0 (6.1)
4		0.5 (0.2) / 13 (1.3)	0.3 (0.1) / 11.6 (3.8)

The algorithms supplying the least unbiased results from the international benchmark paper underestimated the height for all four plots by 2.2 m, or a relative height bias of -10%, which were almost always larger than the results obtained by the proposed method (Table 6). For plots with larger trees the bias is less than 1 m, within the field guidelines (Robertson, 2000), even though the data was acquired with TLS, which was proven to produce sometimes unreliable estimates for total height (Garms et al., 2020). For smaller trees, the case of plots 3 and 4 results supplied by the proposed method are still superior in absolute value than the benchmarking

study, but the error is larger than the acceptable limits established through the guidelines, which means that the height estimation from TLS should be avoided, if the point cloud is not enhanced by merging with aerial lidar or photogrammetric point clouds (Garms et al., 2020). The proposed method mirrors the findings from the international benchmarking study, with bias increasing in absolute value from plots with larger trees to plots with smaller trees. In addition, the variability of the estimates exhibited a slight improvement compared to the best results presented in the benchmarking study for all plots, except for the dominant/codominant trees from plot 4 (Table 6).

Table 6: Tree height bias and RMSE for the benchmarking data. The least unbiased algorithms from the benchmarking study were the FGI for plots 1 and 2, which were <90% correct and had a completeness < 95%, and by the TUZVO for plots 3 and 4, which were correct <95% and had a completeness of 75%. The least variable estimates were provided by the TM algorithm for plots 1 and 2 and by the TUWien for plots 3 and 4, which had a completeness <65% and a correctness of 95%. The abbreviations were from the Liang et al (2018b). T**he All trees and (Co) Dominant shows the results of the proposed method of measuring trees from TLS for all the trees and for the Dominant/Codominant trees, respectively.**

Plot	**Bias [m] (Bias [%]) / RMSE [m] (RMSE [%])**		
	Best	**All Trees**	**(Co)Dominant**
1	-2.2^{FGI} (-10 ^{FGI}) / 2.8^{TM} (13 ^{TM})	-0.66 (-3.4) / 2.1 (10.8)	-0.89 (-4.4) / 1.73 (8.7)
2		-0.87 (-5.9) / 2.4 (15.8)	-1.27 (-8.1) / 2.14 (13.6)
3	-2.2 ^{TUZVO} (-10 ^{TUZVO}) / 4.4 ^{TUWien} (23 ^{TUWien})	-1.35 (-6.8) / 3.0 (15.0)	-1.75 (-5.6) / 2.4 (11.8)
4		-2.4 (-11.2) / 6.4 (29.2)	-4.16 (-16.6) / 5.8 (23.4)

4.4 McDonald - Dunn Research Forests Data

The results based on the International benchmark data were expected to supply superior results, as the training data was created using the trees from the plot 1 and 3. To test the generality of the method, namely the ability to supply accurate values in other types of forests on which the neural network from PointNet++ was trained, I have applied it without changes to the McDunn dataset (*Figure 14* a). The main difference between the international benchmark data and the McDunn data is the lack of fully covered trees with points, in the sense that all the stems had portions that were completely occluded from the sensors. Therefore, the newly developed algorithm was always using incomplete data. The lack of point coverage seems to have limited

impact on the performances of the method, as the completeness and the correctness was above 94%. (*Figure 14* b and Table 7).

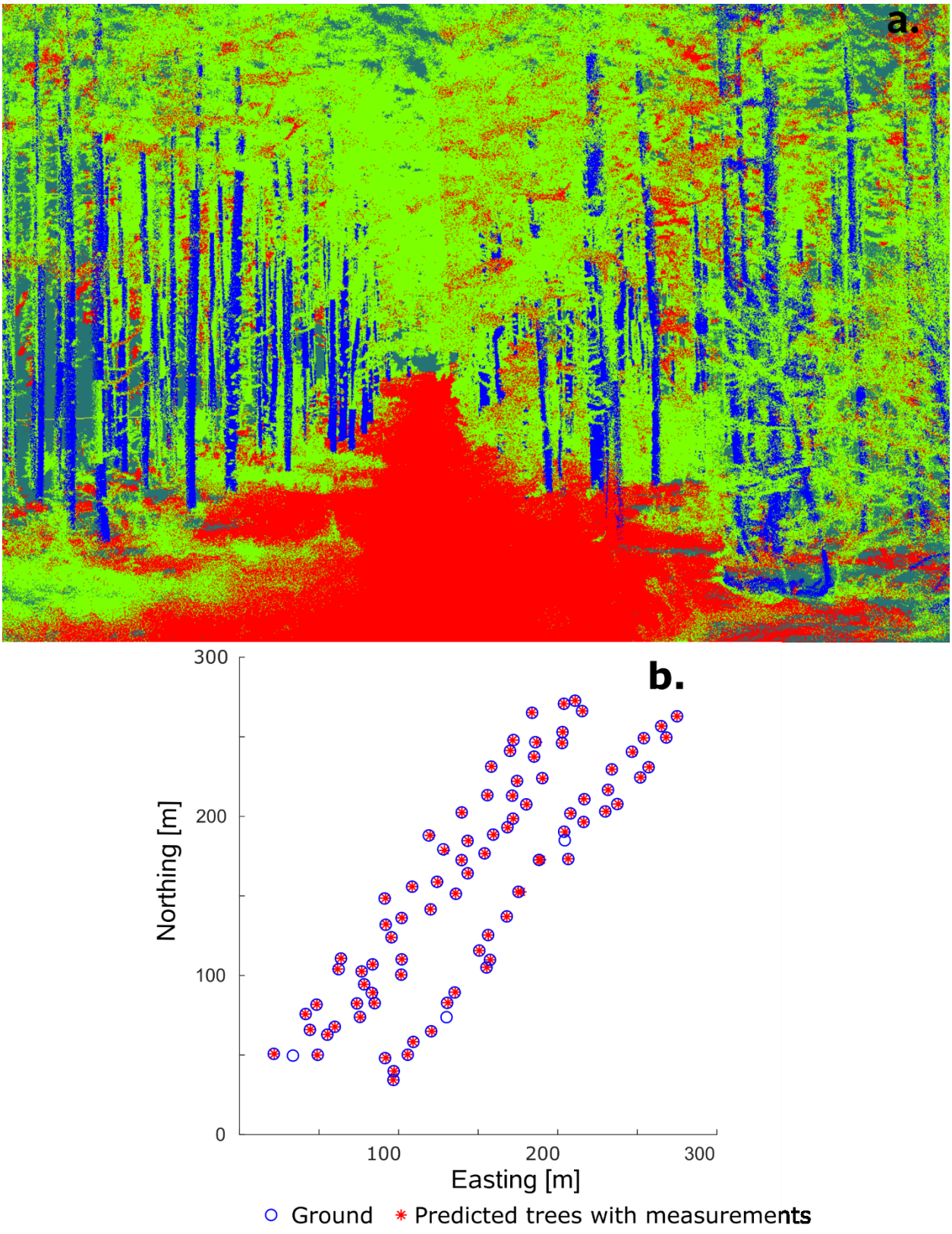

Figure 14: *Classified McDunn Data. a. final classification of the point cloud b. Matching map*

Table 7: Statistics for the McDunn Data for diameters measured at 1.5 m and 2.5 above the ground

Parameters	**Value at 1.5 m**	**Value at 2.5 m**
Completeness	94.25%	94.25%
Correctness	96.47%	96.47%
Location Bias [m]	0.02	0.02
Location RMSE [m]	0.14	0.14
Diameter Bias [mm/%]	8.4 / 1.3	-2.5 / -0.4
Diameter RMSE [mm/%]	62.8 / 9.73	50.2 / 8.57
Height Bias [m/%]	-11.08 / -25.7	-11.08 / -25.7
Height RMSE [m/%]	12.39 / 28.7	12.39 / 28.7

The variability in identifying the location is less than half DBH (i.e., 13.66 cm). The bias of diameter at 1.3 m and 2.5 m height is less than 1 cm, and the relative bias has a magnitude of approximately 1%, well below the 5% measurements standards, suggesting that from practical perspective the algorithm is unbiased. The RMSE for diameter at both heights is <10%, indicating that the method is not only accurate but also precise. However, the height is not acceptable, as it is severely underestimated by 11 m, which corresponds to a relative bias of more than 25%. The height estimates are not only inaccurate but also imprecise, as the relative RMSE is 28.72%. However, the lack of reliability in computing the height could be related to the data itself not necessarily to the method, as TLS was proven to significantly underestimate the height (Garms et al., 2020).

5 Discussion

The metrics used to assess the performance of the proposed forest inventory method based on TLS are either superior or comparable with the best results supplied by the method considered in the benchmarking study of Liang et al (Liang et al., 2018b). However, the best results were produced by the same method, some of them performing as expected from some perspectives, but poorly from others. I considered the metrics to be sequential in nature, in the sense that the completeness and correctness should be ensured before dimensional measurements are compared. Therefore, from a completeness perspective only, three methods from the benchmarking study supplied results above 90% (namely FGI, SLU and RILOG, as they are label by Liang et al (2018b)), and even in that case only for plots 1 and 2. For plots 3 and 4, the same study showed that no method achieved values >90% completeness, with the FGI and SLU reaching 88%, whereas RILOG attained 85%. From a correctness perspective, only FGI and SLU consistently achieved values around 90%, the RILOG was challenged in two plots (i.e., 1 and 2), where the correctness dropped to 75%. In comparison, the proposed procedure always has the correctness above 90% (>95% when all the trees were considered) and achieved completeness of more than 94% for dominant trees in all plots, except for plot 4, which has the completeness of 84%. Therefore, for inventory purposes the method developed in this study is operational, particularly for stands for which commercial thinning or regeneration harvests are scheduled. Such stands have larger diameters, usually the DBH ≥20 cm, where the proposed method achieved more than 95% completeness and correctness. The 5% error ensures that the accuracy and precision of the forest inventory is superior to any field base inventory, which is very rare and has a confidence interval <10%.

The dimensional estimates of the proposed algorithm were either similar or smaller than the best values from the benchmarking study. However, with two exceptions (i.e., FGI and SLU), the dimensional estimates with the smallest bias or variability were obtained by algorithms with large false positive and false negative errors (i.e., TreeMetrics, TUZVO and TUWien). Compared to the FGI and SLU, the proposed method has virtually the same bias (i.e., 3.3 mm vs. 5 mm (FGI) vs 4 mm (SLU)), while exhibiting significantly less variability (i.e., 4% vs. 10% (FGI) vs. 12% (SLU)). Therefore, diameter estimates that I have obtained are not only accurate, but they are also more precise than the current methods. It should be noticed that the DBH bias is close to the device accuracy (i.e., 2 mm at 25 m). The situation was partially mirrored for height, as the proposed method supplied similar values only with FGI method (i.e., -1.3 m vs. -1.8 m (FGI)), as the SLU attained a bias of -3.0 m. The variability of the estimated height was similar to FGI or 17% and SLU (i.e., approximately 15%), except for one plot, which has a relative RMSE of 29%.

The proposed individual tree measurement method from TLS proved to be not only accurate but also generalist, as it operates with the same parameters in different forest ecosystems and data acquisitions. However, the proposed method is sensitive to variation in the point cloud density across the inventory area, particularly completeness. The causes for reduced performance in low density regions of the forest is twofold. First, during the point classification part of the algorithm, random boxes are distributed throughout the entire scene. Therefore, points close to the boundaries have a lower chance to be present in multiple overlapping boxes, which produces limited representations of the trees. First, regions with reduced point density, such as the ones close to the boundaries of the inventory area, can lead to stem elimination by the DBSCAN algorithm. A practical solution, which involves increasing the field work, is to

increase the scanning area such that all the trees within the region of interests are not located close to the edges.

6 Conclusion

The objective of the present study was to develop a method for identifying individual trees and measuring the diameter and height along the stem from high density terrestrial point clouds. The method proposed in this research delineates points that are stem from the non-stem points by applying semantic segmentation at point level, which is subsequently enhanced during the individual tree identification and attributes estimation. My results show that 3-D deep learning is practical in forest inventory, but its performances depends on the available information (i.e., consistent high point density throughout the forest stand).

The semantic segmentation, implemented by modifying the PointNet++ algorithm, classifies the points as ground, stem and crown, whereas the density-based clustering, implemented by adjusting the DBSCAN algorithm, delineated individual trees from the point cloud. The method was tested on four 1000 m^2 plots from the international benchmark data (Liang et al., 2018b). The dominant and codominant trees were more than 95% correctly identified, whereas the completeness was at least 94%, except for one of the plots which was 84%. Most missing trees are caused by the boundary effect and inconsistent point density, but the proposed method still outperforms the tree measuring algorithms presented by the international benchmarking study. The method estimates DBH with a bias < 3 mm, except in one plot where the bias reached 6.2 mm, but exhibits a variability almost half (i.e., <14 mm) compared with the best result in the international benchmarking study (i.e., 20 mm). The situation was mirrored for height estimation, with bias <7% (except for one plot which exhibited 11% bias). The application of the algorithm to the McDonald-Dunn Research Forest supplied a completeness of 94.25% and a 96.47% correctness, even though the algorithm was developed on a completely different species and the trees were not seen from all sides (i.e., incomplete

description of the stem). The diameter bias at 1.5 m and 2.5 m is at most 1.3%, confirming the accuracy and generality of the algorithm. The algorithm provides operational worthy results in tree recognition and diameter estimation for point clouds that describe almost completely the trees. However, the algorithm seems to be sensitive to the stem coverage with points, which should be addressed in future research.

The performances of the algorithm can be improved in several directions. First, additional training data is needed to improve the raw point classification, with significant impact on the subsequent computations. The present study uses only 10 trees to estimate hundreds of trees, some different species and on different continents. The second improvement should be focused on elimination of the boundary effects from tree identification and dimensional estimation.

7 References

Applied Imagery, 2017. Quick Terrain Modeler. Silver Spring, MD.

Cabo, C., Del Pozo, S., Rodríguez-Gonzálvez, P., Ordóñez, C., González-Aguilera, D., 2018. Comparing Terrestrial Laser Scanning (TLS) and Wearable Laser Scanning (WLS) for Individual Tree Modeling at Plot Level. Remote Sens. 10, 540. https://doi.org/10.3390/rs10040540

Cochran, W.G., 1977. Sampling techniques. John Wiley and Sons, Singapore.

Conto, T., Olofsson, K., Gorgens, E., Rodriguez, L.C., Almeida, G., 2017. Performance of stem denoising and stem modelling algorithms on single tree point clouds from terrestrial laser scanning. Comput. Electron. Agric. 143, 165–176. https://doi.org/10.1016/j.compag.2017.10.019

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96. AAAI Press, Portland, Oregon, pp. 226–231.

Fang, R., Strimbu, B., 2017. Stem Measurements and Taper Modeling Using Photogrammetric Point Clouds. Remote Sens. 9, 21.

Fang, R., Strimbu, B.M., 2019. Comparison of Mature Douglas-Firs' Crown Structures Developed with Two Quantitative Structural Models Using TLS Point Clouds for Neighboring Trees in a Natural Regime Stand. Remote Sens. 11, 1661. https://doi.org/10.3390/rs11141661

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 381–395. https://doi.org/10.1145/358669.358692

Garms, C.G., Simpson, C.H., Parrish, C.E., Wing, M.G., Strimbu, B.M., 2020. Assessing lean and positional error of individual mature Douglas-fir (Pseudotsuga menziesii) trees using active and passive sensors1. Can. J. For. Res. https://doi.org/10.1139/cjfr-2020-0041

Hackenberg, J., Spiecker, H., Calders, K., Disney, M., Raumonen, P., 2015. SimpleTree —An Efficient Open Source Tool to Build Tree Models from TLS Clouds. Forests 6, 4245–4294. https://doi.org/10.3390/f6114245

Huang, H., Li, Z., Gong, P., Cheng, X., Clinton, N., Cao, C., Ni, W., Wang, L., 2011. Automated Methods for Measuring DBH and Tree Heights with a Commercial Scanning Lidar [WWW Document]. https://doi.org/info:doi/10.14358/PERS.77.3.219

Kuželka, K., Slavík, M., Surový, P., 2020. Very High Density Point Clouds from UAV Laser Scanning for Automatic Tree Stem Detection and Direct Diameter Measurement. Remote Sens. 12, 1236. https://doi.org/10.3390/rs12081236

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., Huang, H., Jo, H.-W., Katoh, M., Liu, L., Mokroš, M., Morel, J., Olofsson, K., Poveda-Lopez, J., Trochta, J., Wang, D., Wang, J., Xi, Z., Yang, B., Zheng, G., Kankare, V., Luoma, V., Yu, X., Chen, L., Vastaranta, M., Saarinen, N., Wang, Y., 2018a. International benchmarking of terrestrial laser scanning approaches for forest inventories. ISPRS J. Photogramm. Remote Sens. 144, 137–179. https://doi.org/10.1016/j.isprsjprs.2018.06.021

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., Huang, H., Jo, H.-W., Katoh, M., Liu, L.,

Mokroš, M., Morel, J., Olofsson, K., Poveda-Lopez, J., Trochta, J., Wang, D., Wang, J., Xi, Z., Yang, B., Zheng, G., Kankare, V., Luoma, V., Yu, X., Chen, L., Vastaranta, M., Saarinen, N., Wang, Y., 2018b. International benchmarking of terrestrial laser scanning approaches for forest inventories. ISPRS J. Photogramm. Remote Sens. 144, 137–179. https://doi.org/10.1016/j.isprsjprs.2018.06.021

Liang, X., Kukko, A., Hyyppä, J., Lehtomäki, M., Pyörälä, J., Yu, X., Kaartinen, H., Jaakkola, A., Wang, Y., 2018c. In-situ measurements from mobile platforms: An emerging approach to address the old challenges associated with forest inventories. ISPRS J. Photogramm. Remote Sens., ISPRS Journal of Photogrammetry and Remote Sensing Theme Issue "Point Cloud Processing" 143, 97–107. https://doi.org/10.1016/j.isprsjprs.2018.04.019

Ministerul silviculturii;, 1988. Norme 3 pentru alegerea si aplicarea tratamentelor. Centru de material didactic si propaganda agricola, Bucharest.

Ministerul silviculturii, 1986. Norme 5 pentru amenajarea padurilor. ICAS Lithography, Bucharest.

Moenning, C., Dodgson, N.A., 2003. Fast Marching farthest point sampling (Technical report No. UCAM-CL-TR-562). University of Cambridge, Cambridge UK.

Olofsson, K., Holmgren, J., 2016. Single Tree Stem Profile Detection Using Terrestrial Laser Scanner Data, Flatness Saliency Features and Curvature Properties. Forests 7, 207. https://doi.org/10.3390/f7090207

Powers, D.M.W., 2007. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation (No. SIE-07-001). Flinders University, Adelaide AU.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. ArXiv170602413 Cs.

Raumonen, P., Casella, E., Disney, M., Åkerblom, M., Kaasalainen, M., 2013. Fast automatic method for constructing topologically and geometrically precise tree models from TLS Data. Presented at the 7th International Conference on Functional-Structural Plant Models, Saariselkä, Finland, pp. 89–91.

Rencher, A.C., 2002. Methods of Multivariate Analysis. John Wiley and Sons, New York NY USA.

Reynolds, R.T., Graham, R.T., Reiser, M.H., Bassett, R.L., Kennedy, P.L., Boyce, D.A., Goodwin, G., Smith, R., Fisher, L., 1992. Management Recommendations for the Northern Goshawk in the Southwestern United States (GTR No. RM-217). US Forest Service, Fort Collins CO.

Robertson, F.D., 2000. Timber Cruising Handbook. USDA Forest Service, Washington DC.

Saito, T., Rehmsmeier, M., 2015. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLoS ONE 10. https://doi.org/10.1371/journal.pone.0118432

Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X., 2017. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM Trans. Database Syst. 42, 19:1–19:21. https://doi.org/10.1145/3068335

Strimbu, B.M., Qi, C., Sessions, J., 2019. Accurate Geo-Referencing of Trees with No or Inaccurate Terrestrial Location Devices. Remote Sens. 11, 1877. https://doi.org/10.3390/rs11161877

Tabachnick, B.G., Fidell, L.S., 2001. Using multivariate statistics. Allyn and Bacon, Needham Heights.

Torr, P.H.S., Zisserman, A., 2000. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. Comput. Vis. Image Underst. 78, 138–156. http://dx.doi.org/10.1006/cviu.1999.0832

X. Liang, P. Litkey, J. Hyyppa, H. Kaartinen, M. Vastaranta, M. Holopainen, 2012. Automatic Stem Mapping Using Single-Scan Terrestrial Laser Scanning. IEEE Trans. Geosci. Remote Sens. 50, 661–670. https://doi.org/10.1109/TGRS.2011.2161613

Yang, B., Dai, W., Dong, Z., Liu, Y., 2016. Automatic Forest Mapping at Individual Tree Levels from Terrestrial Laser Scanning Point Clouds with a Hierarchical Minimum Cut Method. Remote Sens. 8, 372. https://doi.org/10.3390/rs8050372

Zell, A., Mache, N., Huebner, R., Mamier, G., Vogt, M., Schmalzl, M., Herrmann, K.-U., 1994. SNNS (stuttgart neural network simulator), in: Neural Network Simulation Environments. Springer, pp. 165–186.