FINE-SCALE INVENTORY OF FOREST BIOMASS WITH GROUND-BASED LIDAR

ZHOUXIN XI
Bachelor of Engineering, Tongji University, 2011
Master of Arts, The Ohio State University, 2013

A Thesis
Submitted to the School of Graduate Studies
of the University of Lethbridge
in Fulfilment of the
Requirements for the Degree

DOCTOR OF PHILOSOPHY

Department of Geography
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

FINE-SCALE INVENTORY OF FOREST BIOMASS WITH GROUND-BASED LIDAR

ZHOUXIN XI

Date of Defense: April 23, 2019

Dr. Chris Hopkinson Thesis Supervisor	Professor	Ph.D.
Dr. Laura Chasmer Thesis Examination Committee Member	Assistant Professor	Ph.D.
Dr. Stewart Rood Thesis Examination Committee Member	Professor	Ph.D.
Dr. Derek Peddle Internal External Examiner Department of Geography Faculty of Arts and Science	Professor	Ph.D.
Dr. Richard Fournier External Examiner Université de Sherbrooke Sherbrooke, Quebec	Professor	Ph.D.
Dr. Craig Coburn Chair, Thesis Examination Committee	Associate Professor	Ph.D.

Abstract

Biomass measurement provides a baseline for ecosystem valuation required by modern forest management. The advent of ground-based LiDAR technology, renowned for 3D sampling resolution, has been altering the routines of biomass inventory. The thesis develops a set of innovative approaches in support of fine-scale biomass inventory, including automatic extraction of stem statistics, robust delineation of plot biomass components, accurate classification of individual tree species, and repeatable scanning of plot trees using a lightweight scanning system. Main achievements in terms of accuracy are a relative root mean square error of 11% for stem volume extraction, a mean classification accuracy of 0.72 for plot wood components, and a classification accuracy of 92% among seven tree species. The results indicate the technical feasibility of biomass delineation and monitoring from plot-level and multi-species point cloud datasets, whereas point occlusion and lack of fine-scale validation dataset are current challenges for biomass 3D analysis from ground.

Acknowledgements

First and foremost, thanks are due to my supervisor, Dr. Chris Hopkinson. You are always supportive and tolerant, and most importantly, your honest and cautious attitude towards academic research are lifetime treasure to me.

I am honoured to have Dr. Laura Chasmer, Dr. Stew Rood and Dr. Derek Peddle in my committee. The thesis won't come to shape without all of your advice and more importantly, your encouragement. It is also much appreciated to have Dr. Richard Fournier as my external examiner who kindly provided many valuable comments. Special thanks to Dr. Laura Chasmer for your warmhearted field guidance and publication support. Sincere gratitude to my course lecturers, Dr. Wei Xu and Dr. Craig Coburn.

The thesis cannot survive without funding from S.G.S. International Tuition Award and the Dean's Scholarship from the University of Lethbridge, Campus Alberta Innovates Program (CAIP), and NSERC Discovery Grants Program.

It is fortunate to meet Dr. Yuchu Qin, Dr. Rongxu Qiu, and Dr. Haining Jiang in my beginning years. I also owe thanks to lab colleagues, Mrs. Celeste Barnes, Mr. Thomas Porter, Mr. Maxim Okhrimenko, Mr. David R. McCaffery, Mr. Joshua Montgomery, Dr. Craig Mahoney, and Dr. Reihaneh Peiman. I appreciate the teaching experiences with Dr. Ravinder Virk and Ms. Ariel Pollard-Belsheim, and many memorable moments with Mr. Chaowen Wang, Mr. Li Yu, Dr. Zehan Pan, Dr. Xiao Lin and Mr. Fang (Reggie) Xu.

I won't forget the mentorship from Dr. Shanshan Cai, Dr. Hongshuo Wang, Dr. Jialin Lin, Dr. Zhen Ye, Mr. Zhenxiong Gu, Dr. Shaoming Zhang, Mrs. Linong Liu, Mrs. Xiaobo Jiang and Mrs. Yanfei Zhang in the past. I am grateful for every nice person coming into my life in Lethbridge. To my parents in the end, I owe you many hugs.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AB	Adaboost
AGB	Above-ground Biomass
ALS	Airborne Laser Scanning
BEF	Biomass Expansion Factor
CNED	Characteristic Number Ellipse Detector
CNN	Convolutional Neural Network
DA	Discriminant Analysis
DBH	Diameter at Breast Height
DHP	Digital Hemispherical Photography
DSM	Dynamic Segment Merging
DTM	Digital Terrain Model
ETS	Explicit Tree Structure
FCN	Fully Convolutional Network
FOV	Field of View
FSS	Fusion Scanning System
FWHM	Full Width at Half Maximum
GMM	Gaussian Mixture Models
ICP	Iterative Closest Point
IDW	Inverse Distance Weighting
IoU	Intersection over Union
KNN	K-Nearest Neighbors
LAD	Leaf Angle Distribution
LAI	Leaf Area Index
MLS	Mobile Laser Scanning System
MS	Multiple Scan
NB	Naive Bayes
NFI	National Forest Inventory
OA	Overall Accuracy
PAI	Plant Area Index
PAVD	Plant Area Volume Density
PCA	Principle Component Analysis
QSM	Quantitative Structural Model
RCS	Reference Coordinate System
RF	Random Forest
RMSE	Root Mean Square Error
SS	Single Scan
SVM	Support Vector Machine
TC	Transposed Convolution
TIN	Triangulated Irregular Network
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle
WCS	World Coordinate System

Chapter 1. Introduction

1.1 Towards a Systematic Forest Management Framework

Forests cover one third of the global land surface and are an inseparable part of our living environment. Forests provide multiple ecosystem services such as supporting wildlife, providing fiber and fuel, and regulating atmospheric CO_2 concentrations (Kimmins, 1997). Systematic forest valuation from Pretzsch (2009) points out that 82% of total economic forest values are non-commercial in which nutrient cycling and climate regulation are the two most valuable services accounting for 37% and 15% of total values, respectively. It is important to track the flow of forest nutrients such as nitrogen and mineral into food webs, and evaluate forest carbon consequences in terms of climate change. Therefore, a deliberate design of management rules regarding the complex forest roles in ecosystems is a primary agenda of modern forest sectors and industries.

Yet managing a massive forest area, especially in highly forested countries like Canada, is a systematic project. First, forest diagnosis is site-specific, given varying conditions of species, geography and climate. For example, measurement methods and criteria usually vary with scale units such as individual trees, species, or ecozones (Vidal et al., 2016). Forest mapping and evaluation is thus a balanced consideration of different areas and scales. Second, forest conditions also require periodic inventory and monitoring. Forest dynamics contribute the greatest carbon flux (120 Pg/year) to the atmospheric carbon pool in the global terrestrial ecosystem (Lal, 2008). It is a frequent mission to track the amount and change of forests such as afforestation, harvesting, burning and natural disturbances. Finally, decision making is progressive and long-term. Valuing forests from ecosystem perspective is still relatively new among environmental programs and energy industries. In the long run, it will be important to formulate a practical guidance of forest management, built upon a thorough knowledge of forest ecosystem functions and a persistent agreement from nationwide or intergovernmental decision makers such as Reducing Emissions from Deforestation and Forest Degradation (REDD and REDD+) (COP11, 2005; Miles & Kapos, 2008) and UN Convention on Biological Diversity (CBD) (Balmford et al., 2005; Campbell et al., 2009).

A fundamental requirement of systematic forest management is to timely delineate and report the amount, condition and values of forests. The government of Canada has provided a referable example. Since 1980, Natural Resources Canada has been maintaining an annual report of the State of Canada's Forests. The amount of forests in Canada is characterized by several indicators (variables or attributes in other contexts), typically, the area indicator and wood volume indicator. Wood volume data are provided with a lookup table of more than seven species groups, eleven age classes and twelve terrestrial ecozones (Natural Resources Canada, 2018). The condition of forests is categorized by ownership, province and natural disturbance types. The amount and condition of forests are monitored based on periodical nationwide census such as National Forestry Database, National Forest Inventory (NFI) (Gillis et al., 2005), and Statistics Canada. The amount and conditions of forests are then converted to quantitative values of carbon emission, trade market and job employment to represent greenhouse effect, industrial productivity and social impact, respectively. Obviously, the State of Canada's Forests has established a general framework for diagnosing, monitoring and valuing forests.

In the current framework of forest management, how to precisely convert the forest inventory and census data into forest values remains a challenge, partly due to a lack of suitable indicators. An overwhelmingly used indicator is the area indicator. Yet failure to describe the structural information makes the area indicator insufficient to diagnose accurate forest values destroyed for example by wild fire, insect infection, and deforestation, or recovered from regeneration and afforestation. A three-dimensional (3D) indicator, wood volume, is mostly used to count harvesting for the timber market. Counting wood volume changes in tree growth, reproduction, and mortality, where volume is most related, is not yet compiled for the annual report. Wood volume is also not directly convertible to carbon-related valuation. A more informative 3D indicator, biomass, combining information of wood volume, species, age and nutrition (Poorter et al., 2012; van der Sande et al., 2015), can be a suitable indicator linking forest inventory and valuation.

Biomass is defined as dry mass of organs, including living mass of trees, lichen, shrubs and vines, and also dead mass of the litter and soil (sometimes referred to as necromass). It has been used to predict fiber quality and timber value in silviculture, logging, and harvesting industry (Blanchette et al., 2015; Kankare et al., 2013a; Luther et al., 2013; Peuhkurinen et al., 2007). Because a constant portion of biomass is carbon (45% or 50%) (Vashum & Jayakumar, 2012), biomass is also pivotal to value carbon budget, ecosystem productivity, and biofuel availability (Houghton et al., 2009). Therefore, compared to area, wood volume and most other indicators, the biomass indicator has an advantage of closely tying to framework of forest management. A more precise annual report of forest status should be biomass-oriented with regard to the versatility and significance of the biomass indicator. The importance of biomass has already been recognized (Foody, 2003) and is listed as one of the main attributes in Canada’s Forest Inventory (CanFI) founded in 1981, NFI in 1997, Earth Observation for Sustainable Development of forests (EOSD) in 1999, and National Forest Information System (NFIS) in 1999 (Morrison et al., 1999).

1.2 Estimating Biomass from Ground: Methods and Uncertainties

The biomass indicator still encounters the dilemma of rather few applications in forest management framework, due to many constraints and uncertainties of biomass measurement and modeling methods from ground.

For a single tree in boreal or temperate forests on average, about 60% of live biomass is allocated in the stem and bark, 25% in branches and stump, and 15% in roots and foliage, whereas the proportions vary greatly in terms of species and ages (Lauri et al., 2014). To accurately measure its biomass, above-ground biomass (AGB) is first harvested, dried and weighed, and root biomass is then inferred using default look-up table of shoot-to-root ratio. The shoot-to-root ratio has about 15% average relative error summarized from eighteen forest types (Mokany et al., 2005), a considerable uncertainty term of individual-tree biomass. Destructive methods for obtaining AGB is accurate, but has a known limitation of onerous labor and unrepeatable measurement. In practice, AGB is most often inferred from easy-to-measure tree metrics. Intuitively, AGB is a product of above-ground wood volume and density. Volume can be approximated by various structural variables, whereas stem diameter and tree height are among most influential ones. Thus AGB is often inferred using stem diameter-based parameters capped with constant coefficients in polynomial or logarithmic equations (Jenkins et al., 2003; Lambert et al., 2005). The most common stem diameter parameter is diameter at breast height (DBH) with the breast height usually defined as 1.3 m. These coefficients vary with species, locations, and tree components (e.g. bark, stem, branches, leaves). This type of inferring individual-tree biomass is named allometric model, which is an operational procedure to estimate forest attributes in many nationwide biomass projects such as the Energy from the Forest research

(ENFOR) program (Lambert et al., 2005; Ung et al., 2008; Ung et al., 2017) and the U.S. forest carbon inventories (U.S. EPA, 2008). In other studies, only partial biomass, particularly stem biomass, is known from harvesting or inference data. A biomass expansion factor (BEF) is then used to convert the partial biomass to tree biomass. For example, Heath et al. (2009) develops a component ratio method (CRM) which defines BEF as constant biomass percentages of bark, bole, tops, limbs, and stump. The biometric database from the Fluxnet-Canada Data Information System (DIS) specifies BEF of stem for each species and age class (Hopkinson et al., 2016). The aforementioned shoot-to-root ratio can be also categorized as BEF.

Using stem diameter-based allometric equations is convenient, yet uncertainty cannot be ignored. Jenkins et al. (2003) shows a low agreement of ± 30% between allometric estimates after comparing 389 biomass equations for over 100 eastern U.S. species from 104 literature sources. Unfortunately, the suggested nationwide allometric method from Jenkins et al. (2003) underestimates individual-tree biomass to a bias of −19.5% (Stovall et al., 2018). Colgan et al. (2012) points out that choice of allometric equations accounts for one-third to one-fifth of total biomass estimation error benchmarked by harvesting method in savannas area. Calders et al. (2015) further concludes that an underestimation of approximately 30% biomass is caused by simple allometric modeling of canopy structure. The underestimation effect is greatest for large trees with DBH above 70 cm (Gonzalez de Tanago et al., 2018). Coincident findings are also available from forest studies using terrestrial laser scanning (TLS), a cutting-edge surveying technology capable of digitizing trees into millimeter-level 3D points. Hauglin et al. (2014) extracts DBH and height of individual trees from TLS point clouds, and calculates biomass of 29 Norway spruces based on allometric equations. The relative root-mean-square-error (RMSE%) is

32.4%, benchmarked by destructively sampled biomass. An RMSE%, of about 34.6%, is found in similar studies by Lindberg et al. (2012). Apparently, introducing TLS cannot reduce large uncertainty caused by diameter-based allometric modeling. According to Shettles et al. (2016), using inventory data for biomass, RMSE proportions from measurement, sampling, and allometric modeling are 11%, 23%, and 66%, respectively, whereas using TLS scans, proportions become 5%, 25%, and 70%, respectively. Accuracy of sampling and diameter-based allometric modeling does not rely on fine-scale TLS data.

In open area with explicit individual trees, entire tree forms can be captured in TLS point clouds, leading to a more ambitious goal of biomass extraction other than simple metrics of stem diameter and height or destructive sampling. Recent studies exhibit that a 3D tree geometry can be reconstructed from fine-resolution point clouds using cylinder-fitting algorithms (Côté et al., 2012; Hackenberg et al., 2014; Yan et al., 2009), whereas above-ground wood volume can be automatically delineated along every inch of the tree geometry. Final wood biomass is a product of wood volume and a basic wood density constant. The density constant can be based on direct *in-situ* measurement (Calders et al., 2015; Hackenberg et al., 2015) or from some standard density databases such as the global wood density database (Chave et al., 2009; Zanne et al., 2009). Interestingly, the pioneering 3D reconstruction models, sometimes referred to as quantitative structural models (QSM), have shown preliminary success in accuracy. With harvested AGB of 65 Eucalyptus trees as reference, biomass estimation attains an RMSE% of 9.7% in Calders et al. (2015). Hackenberg et al. (2015) compares the reconstructed and harvested biomass of *Q. petraea*, *E. fordii*, and *P. massoniana*, 12 trees for each species with varying scanning conditions, and reports an RMSE% of 16.6%, 10.9%, and 15.0%, respectively. The high accuracies of the two studies meet typical requirements (10–20% error) of biomass measurement in NFIs

(Liang et al., 2016), making QSM a promising alternative of allometric models to provide individual-tree level biomass information (Kankare et al., 2013a).

Considering the plot level where homogeneous trees cluster, the full form of trees (both stem and branches) as well as understory wood are not exhaustively measurable using inventory or TLS. In practice, other than diameter-based allometric models, merchantable stem volume is used to predict plot biomass, due to strong relationship between volume and biomass (Boudewyn et al., 2007). Biomass of understory sapling, non-merchantable volume can all be predicted from the merchantable stem volume. Other volume-based predictors such as growing stock volume and stand volume (total wood volume) are also used in the U.S. Public National FIA Dataset (FIADB). Then a scaling factor, also called BEF, converts plot volume to plot biomass (Boudewyn et al., 2007). This volume-based allometric prediction is considered more accurate than the diameter-based, but it is unfortunately a destructive method, not repeatable for nationwide and long-term management purpose. Recent efforts have been focused on using TLS plot scans to extract stem volume without the requirement for destructive sampling. Stem extraction relies on stem reconstruction models, a simpler version of QSM. The TLS-based stem volume method for biomass estimation can be highly accurate, with Yu et al. (2013) showing an RMSE% of 12.5%. An overarching work by Kankare et al. (2013a) investigates diverse structural metrics from TLS point clouds, including stem volume, stem diameters, crown size, crown shape, and many others. The reported biomass RMSE% is 12.93% for pine plots and 11.90% for spruce plots. Interestingly, among all structural metrics, the authors demonstrate that stem volume has the highest correlation (0.8-0.9) with predicted tree biomass, whereas metrics about stem curve and crown size are also contributive and independent predictors. Combining metrics of stem volume, stem curve and crown size can achieve best accuracy.

Currently, allometric modeling based on stem volume is the most mature solution to biomass estimation at plot level. Extracting explicit branch volumes from QSM without manual trimming and modification will encounter practical problems, for example, failure of fitting small branches with diameter below 5 cm and overestimation of wood biomass (Momo Takoudjou et al., 2018). Discarding small branches (usually < 10 cm in diameter) to compensate branch volume overestimation from QSM becomes a rule of thumb in recent studies (Gonzalez de Tanago et al., 2018; Lau et al., 2018). Gonzalez de Tanago et al. (2018) tuned the diameter parameter of the QSM (Calders et al., 2015; Raumonen et al., 2015; Raumonen et al., 2013) based on 29 harvested large topical trees, and approaches an concordance correlation coefficient (CCC) of 0.95 compared to the CCC of 0.73–0.82 from traditional allometric method. However, their tuned QSM still encountered a high RMSE% of 28.4%. Compared to the stem-based allometric models, the branch-based QSM method still needs refinement with varying species and complex scenes.

1.3 Estimating Biomass from Ground: More than Accuracy

Accuracy is not the only measure of uncertainty. Rapidity, generalizability, and robustness of biomass calculation are other important measures of uncertainty as well which may prevent a study from approaching truth. First of all, rapidness cannot be ignored in order to collect quality TLS scans. To cover a plot of trees, TLS is sometimes irritating for long scanning time, as much as 2.5 hours per plot according to Rice et al. (2014). During long periods of instrument transportation, deployment and scanning, the environment can shift (e.g. wind or sun changes) and scanning targets may not stay static, giving rise to artificial or jittering points in final point clouds. It usually forces surveyors to reduce scanning resolution or field of views as compromise. Second, generalization uncertainty is known by poor algorithm extension to change of data size and study site, and as well as unrepeatable or inconsistent results caused by human interferences. Increasing sample size can be a solution to improving algorithm extensibility, yet reducing human interference is challenging. Human interference is ubiquitous in the processes of registering image or point clouds, sampling training and testing dataset, parameterizing morphological models, and determining thresholds for model parameters. For example, Jenkins et al. (2003) notices that manual selection of allometric equations and variable combinations poses great difficulty to result compilation and generalization among existing biomass studies. Therefore, it is useful to design automatic or semi-automatic algorithms, especially rapid automatic algorithms, so as to remove or isolate the manual errors and facilitate exposing other sources of uncertainty. Automation is useful for applications with high requirement of usability, repeatability, comparability, and progressivity. For example, NFI takes a frequency of five years to record the biomass change. Hopkinson et al. (2008) suggested that the frequency of airborne LiDAR surveys be minimally three years in order to quantify the tree height growth beyond vertical error of point clouds. It can be concluded that finely monitoring the annual growth of trees is challenging. A strong control of error sources is needed and realizing the automation of analysis is one of the choices.

To automate and generalize tree biomass retrieval in a natural plot, individual-tree volumes should not be the only concerned variable of biomass. Incomplete consideration of understory and biased sampling among species are the common issues preventing broader generalization of biomass modeling. From TLS scans, leafy crown and understory are locations where wood points are hardly differentiable. The differentiation problem could limit TLS applications to leaf-off deciduous trees (Calders et al., 2015; Stovall et al., 2018; Yan et al., 2009) or cause branch volume overestimation (Kunz et al., 2017; Momo Takoudjou et al., 2018). Many studies recognize the importance of independent wood filtering algorithm prior to QSM. Simple wood filtering algorithms have been proposed, which can be categorized into seeking high laser intensity (Béland et al., 2014; Côté et al., 2011; Côté et al., 2009; Hackenberg et al., 2015), seeking high point density (Calders et al., 2018; Hackenberg et al., 2014; Hackenberg et al., 2015) or comparing dual-wavelength difference (Danson et al., 2014; Zhan et al., 2013). These ‘hard’ thresholds in filter configuration can lead to unstable filtering performance in varying scanning scenarios. Recent progresses provide more robust solutions to the wood filtering problem. Ma et al. (2016a) and Ma et al. (2016b) combine multiple low-level threshold filters with unsupervised Gaussian mixture models (GMM) classifiers to automate leaf segmentation, with an average validating agreement of surface areas above 90%. Disney et al. (2018) and Vicari et al. (2019) use similar multi-filtering and GMM methods and their average overall accuracy using simulated point clouds and field reference is 83% and 89%, respectively. Wang et al. (2018) designs an unsupervised method Dynamic Segment Merging (DSM) to partition point clouds into small regions and tunes up a linearity threshold to classify wood points. Their overall accuracy is 88% on average based on both one tree scan and four plot scans, outperformed the overall accuracy of 86% using random forest method. Ferrara et al. (2018) uses a density-based iterative clustering method to extract wood points with an overall accuracy between 95%-97%. These complex solutions to the leaf/wood separation problem have greatly benefited QSM biomass analysis (Disney et al., 2018; Malhi et al., 2018) and leaf biomass inference (Åkerblom et al., 2018), yet still need proof of threshold invariance and species generalizability.

Species mixture is common in natural forest plots. Field inventory cannot exhaustively cover all trees, especially young trees of different species, where uncertainty problem occurs. In practice, species classification is rarely mentioned in the contemporary ground-based LiDAR studies, and scan data is preferably collected in a homogenous stand or a plot with simple species composition. This is probably due to the complexity of processing mixed species plots for biomass estimation. However, scan datasets can explode with the time and detailed species surveys cannot be always available. Rapid scan collection from mobile-based laser scanning system (MLS) and unmanned aerial vehicle (UAV) are envisioned to further release the labor of biomass inventory (Liang et al., 2014a; Schneider et al., 2019). Automatic species classification from ground-based LiDAR will become a demanding compartment of biomass modeling framework. Technically, species classification from ground tree scans is not a challenging problem, considering the success of species classification using airborne laser scanning (ALS) technologies such as Kim et al. (2009) and Ørka et al. (2009). Holmgren and Persson (2004) point out that the basic structural metrics such as height, mean stem diameter, and proportion stems could produce 95% accuracy when classifying between Norway spruce, Scots pine, and deciduous trees in Scandinavia forests. Comparably, TLS scans hold diverse understory portions with more short saplings and trees with richer morphology features. Yet due to lack of multi-species scans with plot size, species classification from TLS remains a problem.

The focus of automation and generalization is usually related to processing and analyzing biomass data, whereas how data should be collected is often ignored. Indeed, the high expense and inconvenience in forest environment have prevented ground-based LiDAR (e.g. TLS, MLS and handheld scanner) from being distributed in large quantities as a broad-scale biomass monitoring tool. Unlike airborne LiDAR, ground-based LiDAR is mostly used in localized studies on a temporary basis such as physiological analysis (Malhi et al., 2018), allometric adjustment (Momo Takoudjou et al., 2018) or radiative transfer modeling (Calders et al., 2018). Few cases demonstrate its temporal and repeatable values in, for example, assisting the widespread meteorological monitoring network to capture spatiotemporal variability of ecosystem carbon and energy flux. A cost-effective and flexible ground-based scanning system is needed for broad-scale environmental studies with emphasis on automation and repeatability instead of accuracy perfection. Several ground-based sensing systems have been endeavored for the plant monitoring purpose such as the use of structured light 3D cameras (Nock et al., 2013), stereo cameras (Koci et al., 2017), and portable TLS (Eitel et al., 2013). These experimental systems have obvious drawbacks of having short range, low vegetation sensibility and low affordability, respectively, which is inadequate for a practical biomass surveying purpose.

1.4 Objective and Rationale

On the whole, fine-scale estimation of biomass from ground is not off-the-shelf without adjusting the way biomass is measured and analyzed. The objective of interest in this thesis is to *optimize 3D tree-level biomass estimation accuracy and automation, in an effort to address current complexities of forest management in diagnosis, monitoring and valuation processes.*

Under time and technical constraints, this thesis is limited to point cloud data format and corresponding processing. Although many relevant methods have been investigated, there is no standard or universally accepted approach for estimating biomass in an accurate and efficient manner. The thesis focuses on four objectives as outlined below, to compensate particular sources of gap or imperfection in existing studies, with detailed formation of problems, gaps, aims, methods, results, and discussions separated into paper-based chapters:

- *accurate and efficient extraction of stem form from natural forest plots;*
- *fine delineation of biomass components from natural forest plots with varying tree density, species and scanning conditions;*
- *automatic species identification from individual tree scans in support of biomass prediction;*
- *a lightweight fusion scanning system for canopy biomass change monitoring.*

These four objectives fit in an interconnected and complementary framework. Stem form (stem curve and diameters) obtained from Objective 1 are part of wood reconstruction algorithm in Objective 2, where new functions like branch reconstruction and wood point filtering are added. With filtered wood points of individual tree from Objective 2, species can be identified using point features in Objective 3. Combination of Objective 1-3 provides a practical solution to automatic biomass analysis at plot size. Objective 4 provides a low-cost and temporal biomass scanning solution from data acquisition perspective. The four objectives correspond to four chapters with a summary chapter in the end.

Chapter 2. Automating Plot-level Stem Delineation from TLS

2.1 Context

A baseline question of plot-level biomass analysis is how to retrieve accurate inventory attributes, particularly, stem volume. Another expression of stem volume is 3D stem form, represented as stem curve (medial axis) and axial diameters, or stem taper functions equivalently (Kublin et al., 2013). Earlier studies only focused on gross dimensions of stem curve, such as DBH and tree height (Aschoff & Spiecker, 2004; Bienert et al., 2006; Hopkinson et al., 2004; Maas et al., 2008). DBH can be estimated accurately using the circle-fitting method with an approximate error under 1 cm (Dassot et al., 2011; Kankare et al., 2013a; Pueschel et al., 2013). To unveil more stem form details, more recent studies have progressed to extracting stem taper from TLS point clouds (Ravaglia et al., 2017), stem diameters at any height (Aschoff et al., 2004; Bienert et al., 2006; Henning & Radtke, 2006; Maas et al., 2008; Pueschel, 2013; Srinivasan et al., 2015), or diameters that are orthogonal to the stem directions based on cylinder fitting (Liang & Hyyppä, 2013; Liang et al., 2014c; Liang et al., 2012; Pfeifer & Winterhalder, 2004).

Note that the cylinder fitting accuracy relies on a clear classification of stem points. Indeed, measuring upper-stem diameters with branches in the foreground is the most challenging part of stem form quantification due to strong occlusion and shadowing effects (Berger et al., 2014). Stem classification in heavy branching regions needs to be treated carefully. In process of delineating stem form, three problems are commonly confronted: point occlusion, stem shape variation, and noise interference. Point occlusion breaks a continuous surface down to a mixture of narrow or wide pieces, giving rise to a zigzagging pattern of the extracted stem node and diameters (Pueschel et al., 2013). Stem shape variation adds complexity to the fitting models and parameters (Dassot et al., 2012; Pfeifer & Winterhalder, 2004). Noise points such as ghost points can affect accuracy of local circle fitting (Brolly, 2013; Huang et al., 2011). Solutions to these problems are still exploratory (Liang et al., 2016).

Stem detection is a preprocessing step of delineating stem form attributes from plot scans. Stem locations can be detected efficiently based on 2D features of a point cloud slice at a specified height (e.g., 1.3 m). The 2D features include angular stem width (Lovell et al., 2011; Strahler et al., 2008), circle fitting error (Bienert et al., 2006; Maas et al., 2008; Pueschel, 2013), or convergence after Hough transformation (Aschoff et al., 2004; Ravaglia et al., 2017; Schilling et al., 2011). Other studies extract stem locations after 3D classification of stem points based on geometric characteristics such as flatness of a point cluster, point density, and vertical histogram (Huang et al., 2008; Lefsky et al., 1999; Liang et al., 2012; Srinivasan et al., 2015; Wezyk et al., 2007). The accuracy of tree detection is dependent on the number of scans (Thies & Spiecker, 2004), stem density (Watt & Donoghue, 2005), branching occlusion (Bienert et al., 2007), and understory occlusion (Ducey & Astrup, 2013). In general, 3D stem classification is more stable than stem detection using only one slice at a certain height, because it is less affected by point occlusion at the specific height (Brolly, 2013). However, the existing 3D classification methods are point-wise and require time-consuming spatial computation such as nearest neighbor search. An efficient stem detection based on 3D geometrical features is needed. Data reduction such as Digital Terrain Model (DTM) removal is also helpful to improve the efficiency and accuracy of stem detection (Litkey et al., 2011; Yang et al., 2013).

The chapter aims to design an automatic approach for stem detection and stem form extraction, capable of reducing occlusion effect, tolerating stem shape variation, and filtering out noise points from stem points. It is intended that these automatic methods will improve the robustness and rapidity in which TLS data can be analyzed, with results of interest to the forest industry.

2.2 Methodology

2.2.1 Data and Analysis Framework

The study site is a Scott pine (*Pinus sylvestris* L.) plot with a stem density of 479 stem per ha in Evo, Finland (61.19 °N, 25.11 °E). Our study plot is among a larger set of plots acquired for the EuroSDR International TLS benchmark project: Benchmarking of Terrestrial Laser Scanning for Forestry Applications (Hyyppa & Liang, 2013). The plot was scanned using Leica HDS6100 by the Finnish Geodetic Institute (FGI) during the summer of 2014. Two TLS datasets were acquired from FGI at this site: a single scan (SS) from the plot center and multiple scans (MS) from both the center and four corners. Both datasets were trimmed to the size of 32 m × 32 m with a point resolution of 15.7 mm and a distance accuracy of ±2 mm at the distance of 25 m per plot. The setting of plot size provides sufficient samples for stem analysis and the setting of scan resolution enables branch-level detail. Validation data are field mensuration of stem location, stem DBH, and tree height, as well as offline measurement of stem form and DTM from manual or semi-automated software tools. Stem form is represented as diameters and horizontal center coordinates at each height, at 1 m height intervals up to the highest measurable height inside crown. DTM reference has a resolution of 0.2 m.

A suite of algorithms was developed in MATLAB to automate the retrieval of stem attributes from point clouds without manual interference. The workflow consists of four steps: DTM extraction and removal, stem location detection and separation, branch and stem segmentation, and stem form extraction. Figure 2.1 illustrates a flow diagram of methods used in this study, and described in the following sections.

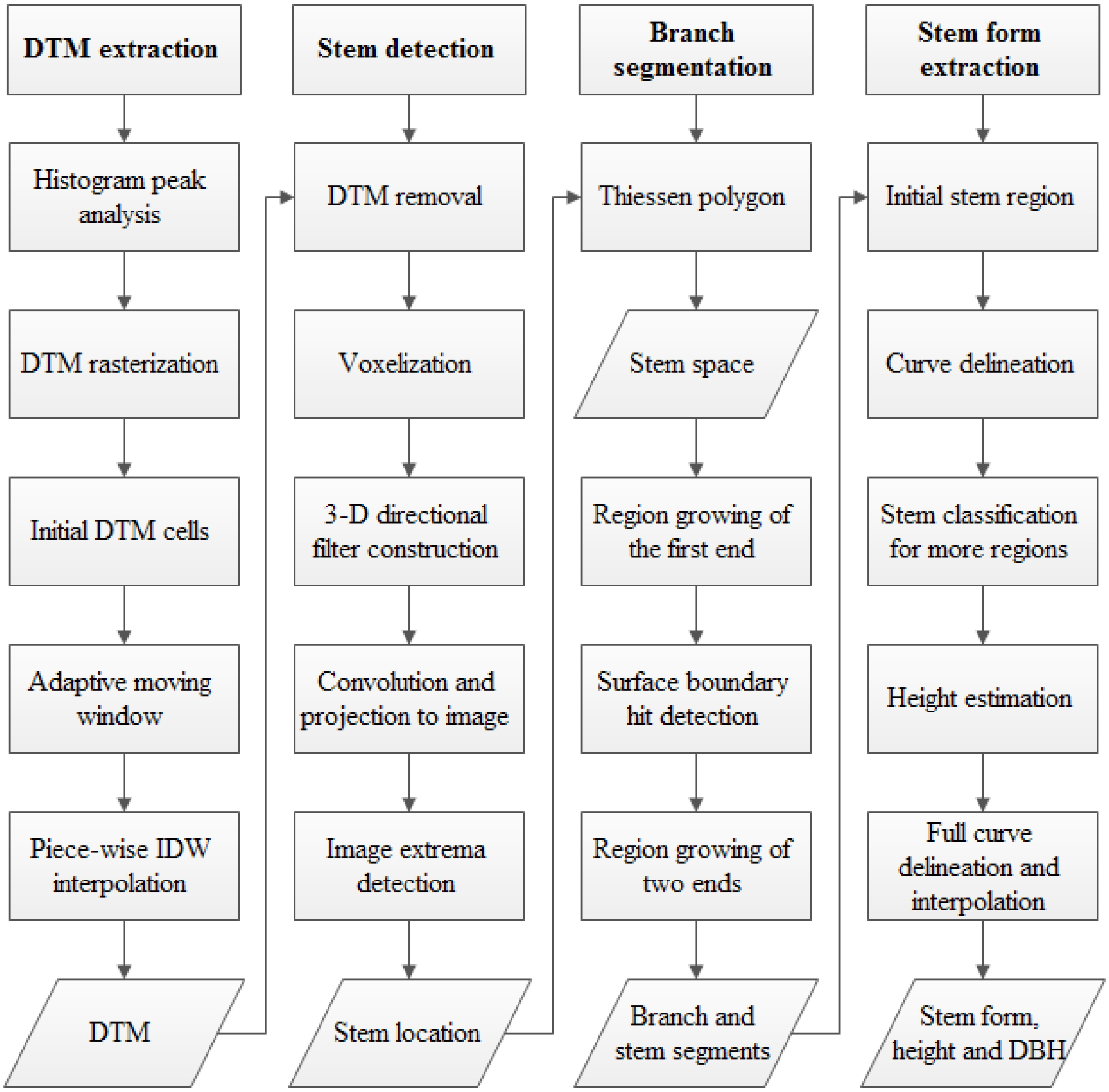


Figure 2.1 Flow-diagram overview of methods used to automatically extract plot-level tree stem attributes. DTM: digital terrain models; IDW: inverse distance weighted; DBH: diameter at breast height.

2.2.2 DTM Extraction

The extraction and removal of DTM is a preliminary procedure for the following step of stem detection and stem form extraction. The general approach of DTM extraction is to detect the candidate ground points and interpolate full ground points from the neighboring candidate ground points.

In forested terrain, initial separation of ground and canopy is a necessary procedure prior to fine DTM extraction (Kobler et al., 2007). An upper threshold (E_{upper}) for DTM is needed to distinguish ground and canopy. The E_{upper} can be estimated based on the histogram of point elevations (Figure 2.2) (Bienert et al., 2006; Douillard et al., 2011). The first histogram peak (E_{p1}) is detected when the slope of the histogram profile reduces to 0. The width of the first peak (W_{p1}) is calculated as full width at half maximum (FWHM). The upper threshold for the DTM elevation (E_{upper}) is defined following Equation 2.1:

$$E_{upper} = E_{p1} + t_w W_{p1} \tag{2.1}$$

where t_w is the controlling parameter for adjusting E_{upper}. The candidate ground points are considered lower than E_{upper}. In this study, we set t_w to be as large as 5, in order to incorporate all ground points as candidate ground points.

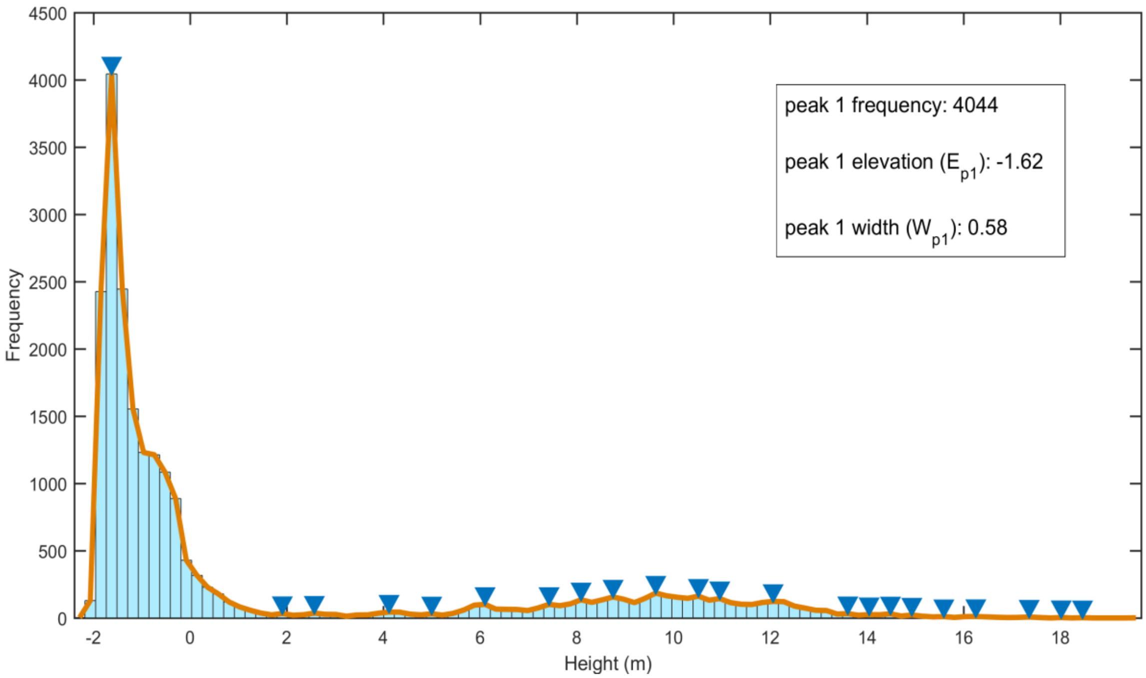


Figure 2.2 Histogram filtering for DTM region. Triangles represent local maximum values of histogram profile. In this example, DTM is found to be lower than 2 m in elevation, which can be estimated based on the width of first histogram peak.

In order to extract the local lowest points, point clouds are rasterized first. Compared to raw irregular points or Triangulated Irregular Network (TIN), raster format can greatly expedite search for spatial neighbors (Meng et al., 2010). All 3D points (p_i) are mapped into a raster cell by normalizing the p_i with DTM resolution (σ). We set σ to be 0.2 m. The elevation of p_i is assigned to be a cell value. Only the lowest elevation inside the cell is kept as the cell value. Empty cells, often a result of occlusion of laser returns, are initially assigned with a positive large value (e.g., 10,000). A square moving window at an initial length of 5 cells is then applied to each cell. The selection of the initial size can be arbitrary because the initial size has little influence on the final results. The local lowest points (MW_{min}) are thus computed as the lowest cell value inside the moving window.

The candidate ground points can be filtered using Equation 2.2:

$$v_{thresh} = \min(E_{upper}, \max(E_{p1}, MW_{min}) + \delta) \\ v_{cand} \leq v_{thresh} \tag{2.2}$$

where v_{cand} is the cell value of candidate ground points, δ (0.5 m in this study) is the elevation tolerance in the moving window, and v_{thresh} is the threshold to filter v_{cand}. The candidate ground point cells are used as support cells to interpolate the center cell value inside the moving window.

The ground points represented by the candidate ground points are likely to have large gaps in canopy areas. Our solution follows the concept from Zhang et al. (2003) which adaptively increases moving window size to proliferate candidate ground points. The size of the moving window keeps increasing until the window covers at least N_{mv} candidate ground point cells. N_{mv} should be large enough (e.g., 10) to avoid noise cells. Our intention is to preserve the lowest elevation information and constrain abrupt elevation changes in the step of interpolation. This can be achieved by designing a robust weight function in light of Kobler et al. (2007) and Kraus and Pfeifer (1998). The center cell value is set to be either unchanged or inverse distance weighted (IDW) neighbor values. It is determined by a piecewise function as Equation 2.3:

$$v(m,n)_{\mathrm{final}} = \begin{cases} v(m,n) & \text{if } v(m,n) < v_{thresh} \\ \dfrac{\sum_{i=m-\frac{s}{2}}^{m+\frac{s}{2}} \sum_{j=n-\frac{s}{2}}^{n+\frac{s}{2}} \dfrac{v(\mathrm{i},\mathrm{j})}{d(i,j,m,n)}}{\sum_{i=m-\frac{s}{2}}^{m+\frac{s}{2}} \sum_{j=n-\frac{s}{2}}^{n+\frac{s}{2}} \dfrac{1}{d(i,j,m,n)}} & \text{else if } v(m,n) \geq v_{thresh} \end{cases} \tag{2.1}$$

$$\text{where } d(i,j,m,n) = \infty, \text{if } v(i,j) \geq v_{thresh}$$

where symbol s is the size of the moving window, symbol $d(i,j,m,n)$ is the distance between cell (m,n) and cell (i,j), and $v(m,n)_{\mathrm{final}}$ is the final ground point value at row m

and column n in the result raster. Only points > 0.1 m above the final ground point value are kept in the following analysis. Those points are regarded as the above-ground points.

2.2.3 Stem Location Extraction and Isolation

Tree stems are identified using template matching (or filter) in 3D space explained in Brunelli and Poggiot (1997). A 3D template is developed to detect tree stems that extend upwards from the ground surface, and have a leaning angle smaller than $\theta_{\max}$. First, above-ground point clouds are divided into voxels at a resolution of $0.1 \times 0.1 \times 0.5$ m (xyz). The voxel resolution is a user-defined parameter. The horizontal voxel resolution of 0.1 m is about half of DBH and is sufficient to detect stem locations. The vertical resolution of 0.5 m can present vertical heterogeneity which is smaller than horizontal heterogeneity. Each voxel is assigned the value of the point count inside the voxel, thereby representing point density. Voxels from the point clouds are multiplied respectively by a 3D template, or what we call a 3D directional filter in our study. The 3D directional filter is also composed of voxels. A sum of the product of point cloud voxels and filter voxels is called convolution. It is efficient to enhance the near-vertical characteristics through 3D convolution, by assigning "1" or any positive value "C_v" to a filter voxel as a reward, and "-1" as a penalty. The structure and size of the filter should be designed to reward all stem points within the stem cylinder, and penalize non-stem points outside the cylinder. If the stem points are partially occluded, the stem points are still rewarded except that reward strength is weakened. The mathematic description of our 3D directional filter is shown in Figure 2.3. We use the parameter s to represent the horizontal radius of "C_v" voxels (reward voxels), numerically close to horizontal stem cylinder radius. We then set the ring thickness of "-1" voxels (penalty voxels) to be s as well. The filter height is b (1 m in this study). The filter width and depth are both $0.5b$ $\tan\theta + 2s$, where θ is the current leaning angle. A rough estimation of s is half DBH (0.2 m in this study). The voxel representation of the filter is demonstrated on the right side of Figure 2.3. Overall, the directional filter is a cylinder shape filled with the penalty voxels and reward voxels. Assuming lean angle is horizontally symmetric, the reward voxels as a whole appear like a revolution geometry, which is formed by rotating the two left thickest lines (Figure 2.3) 360 °around the vertical axis. The penalty voxels appear like another revolution geometry, formed by rotating the two right thickest lines (Figure 2.3) 360 ° around the vertical axis. The remaining filter voxels are filled with "0". We define ρ as a voxel's distance to the centerline, k as the voxel's distance along the rotational line to the line's top end. For each voxel, when $\rho < |0.5b - k| \tan\theta - s$ or $\rho > |0.5b - k| \tan\theta + 2s$, the voxel value is assigned "−1"; when $|0.5b - k| \tan\theta - s \leq \rho \leq |0.5b - k| \tan\theta + s$, the voxel value is assigned C_v; when ρ doesn't satisfy the above two conditions, the voxel value is kept "0". To represent different leaning angles, five filters are created with the leaning angles of 0 °, 5 °, 10 °, 15 °, and 20 °, respectively. The five filter masks are then averaged to get an overall 3D directional filter.

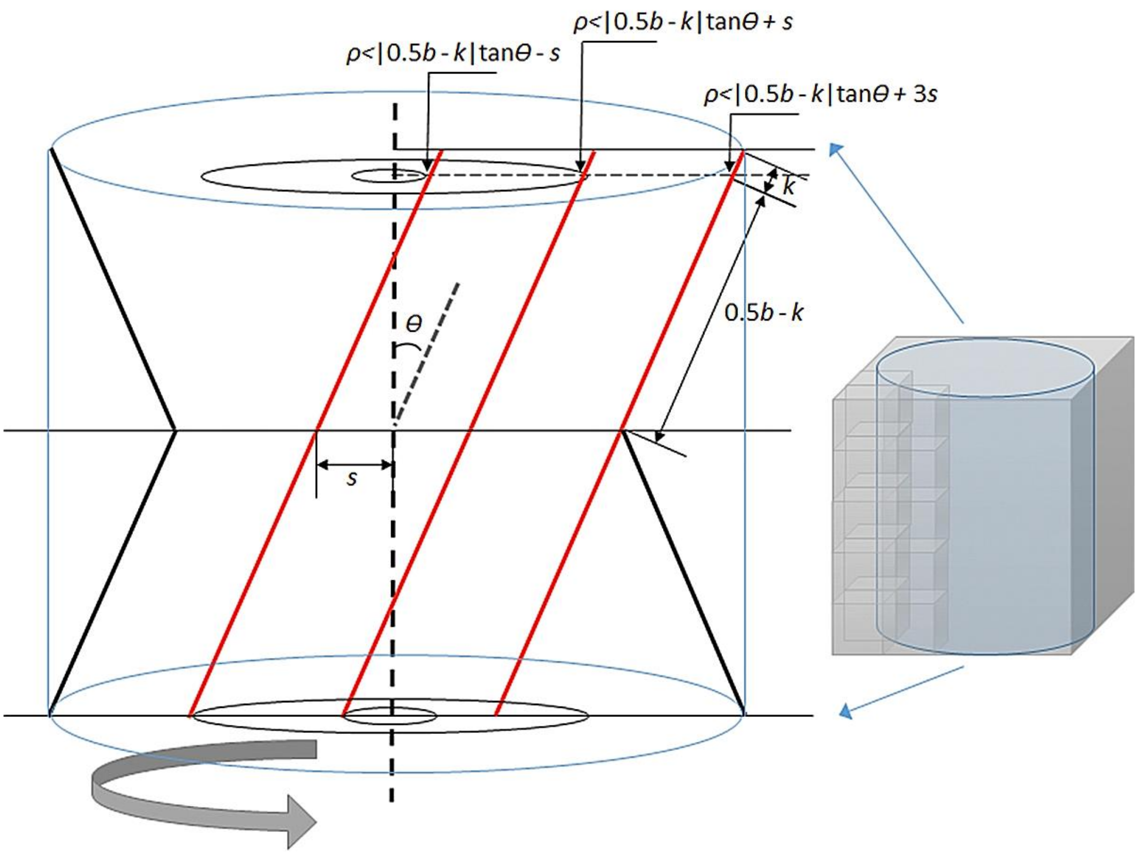


Figure 2.3 3D directional filter: a cube composed of voxels in the right. Mathematic expressions about the filter geometry and structure are noted aside.

This convolution step can be accelerated using 3D Fourier transformation. Convolution in frequency domain is mathematically equivalent to convolution in spatial domain, but the computation complexity in frequency domain is often less. Interested readers can refer to Nixon (2008) about the theory. Specifically, we calculate the 3D Fourier form of the point cloud voxels and the 3D Fourier form of the overall 3D directional filter. The two Fourier forms are multiplied and the inverse Fourier form of multiplication is our convolution result. The convolution only enhances any near-vertical points within the filter size. To aggregate the enhancement for the whole stem, the 3D convolution value in each voxel is summed again vertically, and is projected to the horizontal plane as a 2D image. In this manner, near-vertical points are displayed as bright pixels in the 2D image, whereas points with other spatial patterns turn to dark pixels. Only pixels with brightness greater than a threshold remain. The threshold is defined as the product of point density, sum of award voxels in the overall filter, and as well as a controllable parameter P_S (0.5 in this study). If a remaining bright pixel has brightness greater than neighboring pixels, it is defined as a local extreme. The stem locations can be determined by searching the local extrema of the bright pixels from the 2D image. The image coordinates are converted back to real-world coordinates by inverting the voxelization step.

The above-ground point clouds can be separated in to stem-wise occupation areas based on the detected stem locations. We define stem occupation area approximately as the Thiessen polygon around the stem location. From the point cloud dataset, we observed that pine stems did not move beyond 1 m horizontally from the center location. The point clouds are therefore trimmed to be within 1 m horizontal radius from the stem location, so as to reduce computing overhead. Trimming based on prior knowledge or exploratory analysis is not a necessary step, but helps improve automation efficiency.

2.2.4 Stem Form Extraction: Segmentation

A region growing method is useful to approximate stem and branch morphology based on the geometric properties of a surface, including local connectivity and fractals, as well as the dynamics of the geometry and curvature (Masutani et al., 1996; Roggero, 2002). An illustration of our 3D region growing algorithm is shown in Figure 2.4, where the algorithm is used to segment whole point clouds into numerous cylinders without a need for initial seeds. The region growing algorithm starts from one arbitrary point as shown in Figure 2.4a, and incorporates its neighboring points to be the 1st grown region. In this study, the neighboring points are defined as any remaining points whose absolute coordinate differences from a growing region are smaller than R_S. Intuitively, the neighboring points lie inside nearest cubes (size $R_s \times R_s \times R_s$) from the grown region. The setting of R_s is related to the point density. A large R_s could increase robustness against point density variation, since region growing would not be affected by point gap smaller than the R_s. Yet a large R_s may cause coarse edges of grown regions. In the same manner, the next region growing iteration incorporates neighboring points around the boundary of the 1st growing region as the 2nd growing region, which is highlighted in blue in Figure 2.4a. The 4th, and 6th growing regions are also highlighted in different colors. At this stage, the 6th growing region is no longer continuous. The upper side of the 6th growing region (“end 1”) is separate from the lower side of the 6th growing region (“end 2”). A visual illustration of “end 1” and “end 2” is also shown in Figure 2.4a. We consider the status of the 6th region growing as a “hit” of the surface boundary. The boundary “hit” condition is judged by near-zero width increment of the nth grown region. Upon the “hit” of surface boundary, the grown regions together (from 1st to 6th) exhibit a cylindrical shape. After the “hit”, the growing region (i.e., “end 1” and “end 2”) continues growing, until the width of the “end 1” or “end 2”increases by more than a user-determined width parameter W_{ir} (0.8 in this study), or the direction of the “end 1” or “end 2”changes by more than a user-determined angle parameter D_c (60 ° in this study). The first termination condition indicates that the growing region reached a branch or a stem portion with a width at least W_{ir} greater than the current width. The second termination means that the growing region reaches another branch with more than an angular difference of D_c. Note that fixing these parameters does not mean the resulting cylinders of a branch or a stem would be limited to certain prototype geometry with fixed angle or width. The parameters W_{ir} and D_c only serve to control the degree of fragmenting point clouds into small regions. Those regions will be merged or connected in later steps to form complete branch or stem curves. After the current region growing is terminated, we select a new starting point from the rest of the points and repeat the above region growing, until no new points can be found from the remaining point cloud.

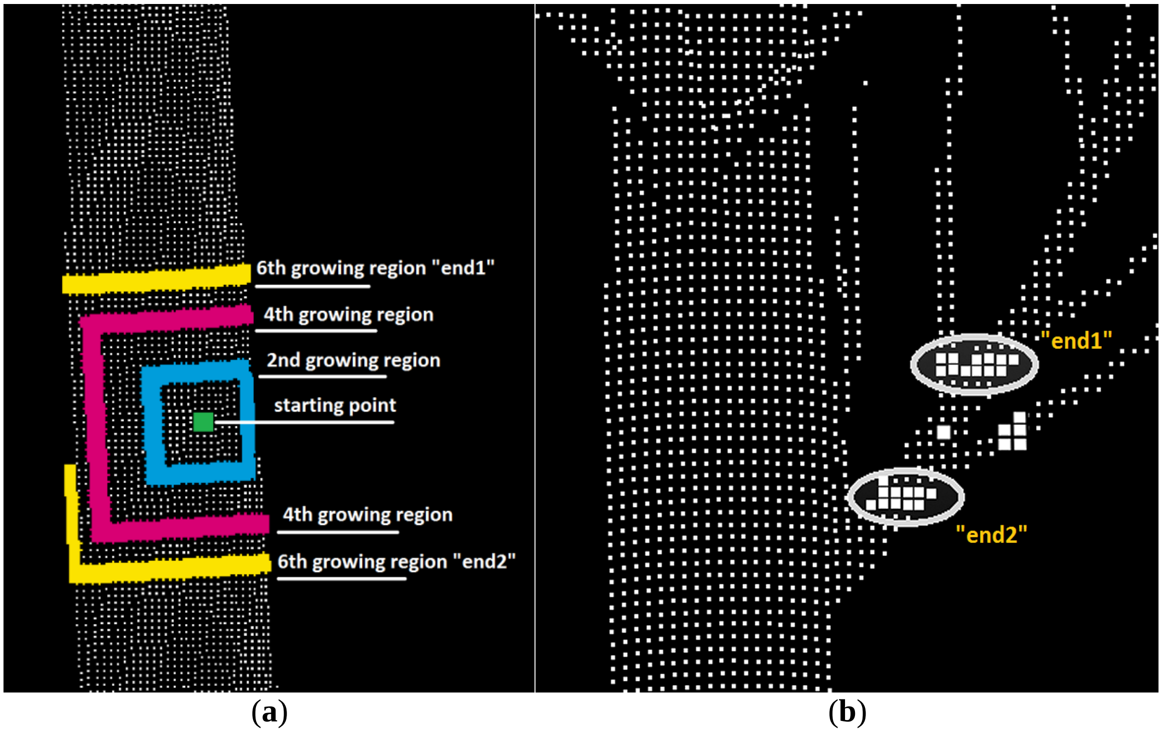


Figure 2.4 Region growing example. (**a**) The starting, 2nd, 4th, and 6th growing regions colored in green, blue, pink, and yellow, respectively; (**b**) "end 1" and "end 2" (circled) recognized from a growing region with four clusters.

Figure 2.4a displays a simple stem point cloud, whereas Figure 2.4b has the more complex situation with branching bifurcation. We intend to control region growing only along the same stem/branch surface. So bifurcated branch points need to be excluded during region growing. This is realized by partitioning the *n*th growing region points into clusters, and only keeping the largest two clusters. The nearest distance between any two clusters should be greater than a user-determined parameter G_t. In other words, G_t is a gap parameter to differentiate region connectivity. In this study, G_t is assigned to be a constant 0.01 m but can be also set as a multiple of the average point resolution, to adapt to different point cloud datasets. Here we focus on the largest two clusters. The largest cluster is assigned to be the “end 1”, and the second largest cluster to be the “end 2”, so as to limit the growing region to be on the stem surface. However, due to point occlusion, the branch may be larger than the stem, and “end 2” may thus be a branch. As such, we add a restriction that “end 2” is classified as the second largest cluster only if that cluster is located in the opposite growing direction of “end 1”. The whole segmentation algorithm and its required parameters are described in Appendix 1. Note that our segmentation algorithm is not aimed to filter a perfect whole stem, instead, it prepares cylindrical segments for the next step of stem classification.

2.2.5 Stem Form Extraction: Classification

The grown region with the most points is selected to initiate stem classification. The first principle component of the principle component analysis (PCA) for the stem region is chosen as an overall stem direction. The initial stem points are then perpendicularly projected to the stem direction to create their projected lengths along the stem direction. The stem points are separated into slices by splitting the projected lengths at equal intervals of 0.2 m. The purpose of slicing at equal intervals is merely to conform to validation data format. Each slice is projected along the stem direction into a plane. A circle-fit algorithm which minimizes the geometric error using least square (Gander et al., 1994) is adopted to find the circle center and radius from the projected points. As a result, the form of the initial part of stem is delineated by a sequence of the circle centers (stem nodes), circle radii (node radii), and circle directions (node directions).

More regions need to be classified as part of the stem. One condition is set to satisfy: the region points shall fall inside a cylinder. The centerline of the cylinder starts from the closest node and is collinear with that node direction. The radius of the cylinder is the closest node radius. A concentric buffer cylinder is used to search new regions. The radius of the buffer cylinder is equal to the node radius timed by D_t. Here D_t is a user-defined ratio parameter indicating the degree of tolerance to the buffer cylinder radius (1.3 in this study). Priority is given to closer and larger regions. After a new region is classified as part of the stem region each time, the stem form is re-delineated using the circle fitting algorithm. Upper-stem and occluded regions may have few points to support circle fitting. Neighboring regions are then searched until sufficient points (e.g., 200) are found. The whole classification process terminates when no regions are found to fall inside the cylinder.

Upper-stem occlusion is common in plot-level TLS scans. Due to upper-stem occlusion, the extracted stem form contains only visible components, yet the full stem form should extend continuously from the bottom to the tree top. Tree height is determined in our algorithm as the vertical distance between the highest and lowest points inside each isolated stem point cloud. The previous delineated stem form is linearly interpolated to zero radius at the tree top. Deficient stems with height lower than 2 m, or diameter smaller than 5 cm, usually not included in tree inventory, are removed from the final results of stem form extraction.

2.2.6 Validation

DTM, stem detection, stem DBH, stem height, and stem form can be validated using the reference data from FGI. The DTM error is defined by the mean distance between extracted DTM points and corresponding nearest points in reference DTM. We only focus on how well the extracted locations from TLS matched the surveyed locations. Our detection rate is defined as the number of detected stems within a tolerable distance to the closest reference stems, divided by the total number of reference stems. The tolerable distance is set to be 0.5 m, so as to allow for possible surveying error of reference locations. This automated calculation of detection rate produces the same results as manual matching in our study. The reference stem nodes manually extracted from each stem point clouds by FGI are organized as 3D coordinates of the stem nodes following a height sequence of 0.65 m, 1.3 m, 2 m, 3 m, 4 m, and so on. Automatically extracted stem nodes are also interpolated to the same height sequence in order to match the reference stem nodes. The match of our nodes and reference nodes is based on the nearest neighbor search from the reference nodes. For each detected stem curve, RMSE of diameters, and r^2 of diameters between the extracted nodes and the reference nodes are investigated.

The parameters needed in all steps are displayed in Table 2.1. The parameter values in the right column can be used as default values in different TLS datasets. Other values were attempted in a few trials and errors, and extraction results did not vary greatly, because the algorithms allow reasonable variations of scanning conditions, point density, and stem geometry. For example, if R_s or G_t is set to smaller values (larger than minimum point resolution), the point clouds will be segmented into smaller pieces, but the following classification step will still recognize stem points based on D_t. The set of parameters, especially the controlling parameters in Table 2.1, are tunable for other datasets to maximize accuracy. For this study, we did not specifically optimize the parameters since we expected the algorithms to be unsupervised and efficient which could avoid tedious tuning processes. We fixed the parameter values as in the right column of Table 2.1 throughout the processing of both SS and MS datasets.

Table 2.1 Parameter and settings for automated stem attribute analysis from TLS data.

	1. DTM extraction	
	DTM resolution σ	0.2 (m)
*	Local elevation tolerance in a moving window δ	0.5 (m)
	Minimum initial ground points in a moving window N_{mv}	10
*	DTM upper threshold factor t_w	5
	2. Stem detection	
	Horizontal resolution for voxelization	0.1 (m)
	Vertical resolution for voxelization	0.5 (m)
*	Cylindrical filter width s (roughly close to DBH)	0.2 (m)
	Cylindrical filter height b	1 (m)
	Maximally-allowed cylindrical filter leaning angle $\theta_{\max}$	20 °
*	Cylindrical filter value C_v (larger value creates more peaks)	1
	Filtered peak detection strength P_s (larger value detects weaker peaks)	0.5
	3. Branch and stem segmentation	
*	Region growing step R_s	0.02 (m)
	Point gap tolerance G_t	0.01 (m)
*	Maximum branch width increase ratio W_{ir}	0.8
	Maximum branch direction change D_c	60 °
	4. Stem curve extraction	
	Stem node interval	0.2 (m)
*	Stem radius tolerance D_t	1.3

* Controlling parameters.

2.3. Results and Discussion

2.3.1 DTM Extraction

Both SS and MS point clouds from plot-1 were processed fully automatically based on the *a-priori* parameters without referring to any mensuration data. The 3D points of the extracted DTM from the MS dataset are shown in Figure 2.5. The point color is rendered by the distance between extracted DTM point and its nearest point in reference DTM. Points with the distance <0.2 m are shown in blue, 0.2–0.4 m green, 0.4–0.5 m yellow, and >0.5 m red. As expected, our DTM extraction algorithm preserves local low points and smoothens upper points. Local low points are not smoothened, so distance is high in the area of pits and bulges. The upper land on the bottom right of Figure 2.5 also shows high distance value, because ground points are inadequate in that vegetated area to support accurate interpolation. For plot-1, average DTM error is 0.16 m (SS) and 0.10 m (MS), respectively. The DTM errors are reasonable, for both the reference and the produced DTMs have a resolution of 0.2 m.

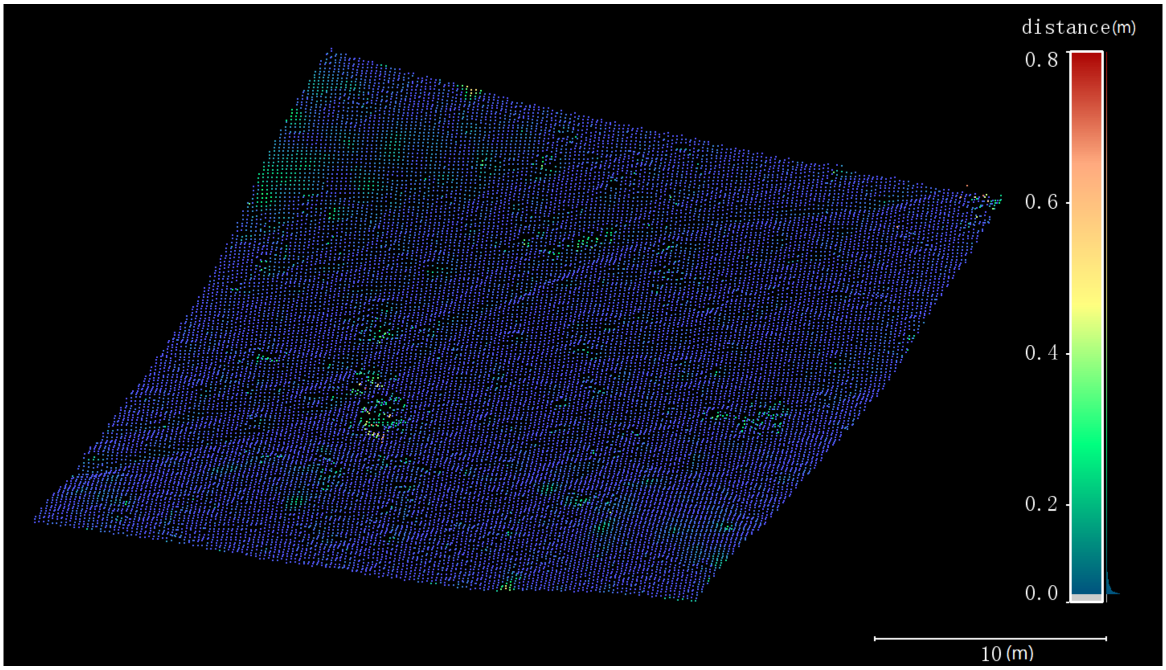


Figure 2.5 DTM 3D points extracted from the plot-1 multiple scans (MS) dataset. Point distance between extracted DTM and reference DTM is rendered using color legend on the right. A scale bar is placed at lower right corner.

2.3.2 Stem Location Extraction and Isolation

The projected 2D image of 3D directional filter convolution using MS dataset is shown in Figure 2.6b, offering comparison to the projected image without applying the directional filter in Figure 2.6a. It can be seen that irrelevant matters such as branch and foliage are cleared using our directional filter in Figure 2.6b. The pixel brightness stays consistently high near the likely stem locations and drops drastically further away. The brightness change (or sharpness) is controlled by the 3D filter size parameter s and filter value C_v. A high degree of sharpness produces speckled effect of the 2D image and leads to over-detection of stem locations. To avoid high sharpness, based on our practical experience, the choice of s is suggested to approximate or exceed the size of stem target and C_v stay below 2. In addition, high sharpness can be mitigated by increasing the maximum-allowed leaning angle $\theta_{\max}$. A large leaning angle (e.g. >40°), however, causes a risk of detecting oblique branches.

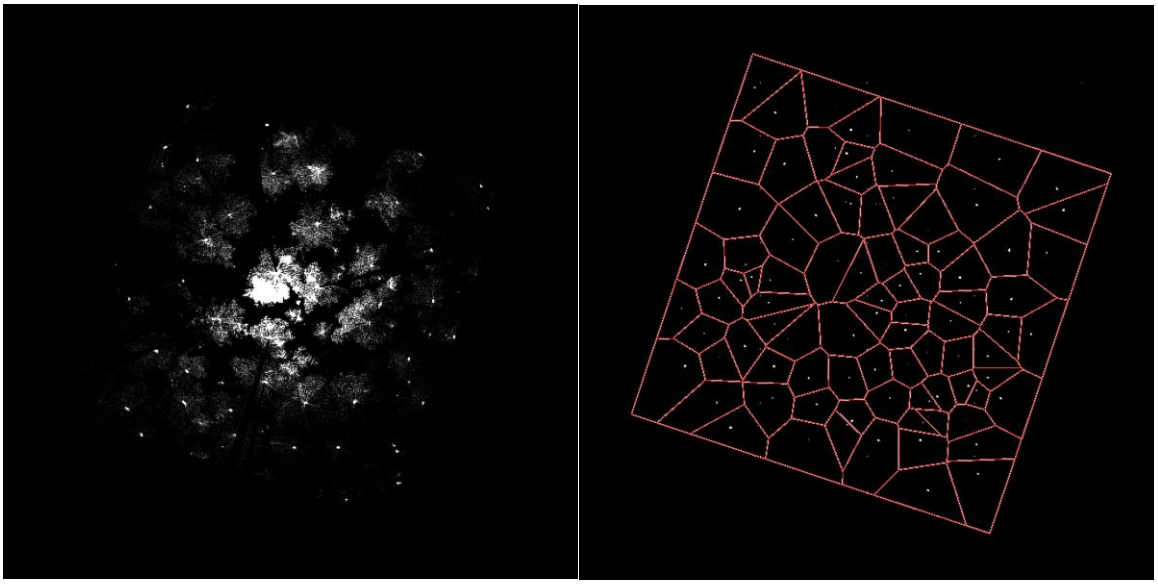

(**a**) (**b**)

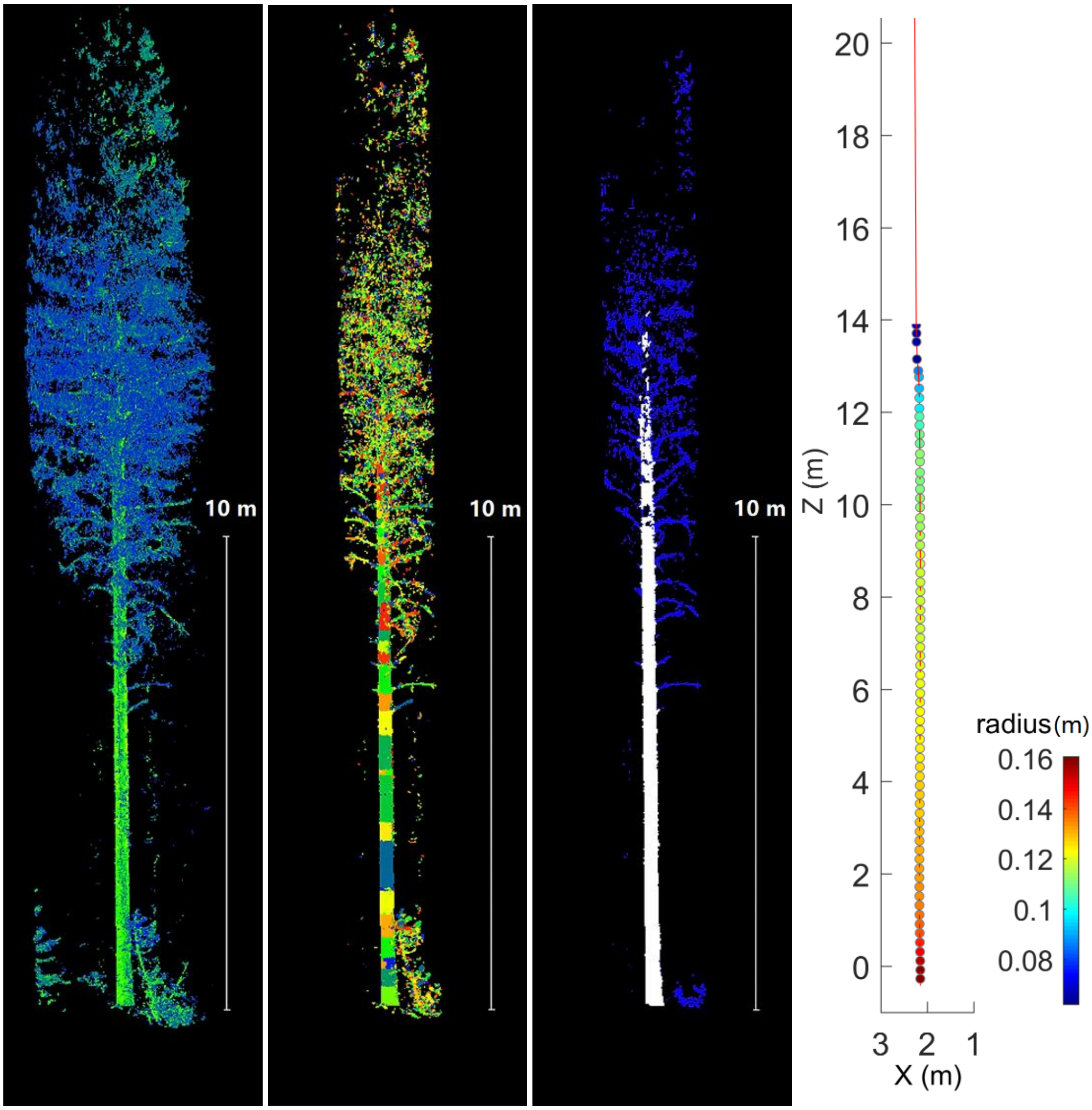
Figure 2.6 Stem location detection. (**a**) 2D image directly projected from point cloud voxels as a comparison; (**b**) 2D image projected from the convolution of 3D directional filter and point cloud voxels, with Thiessen polygons around the extracted locations.

In Figure 2.6b, Thiessen polygons are shown as lines surrounding each extracted stem location. Among the detected stems from filter convolution, deficient stems were further discarded in the stem classification step. As a result, a total of 50 (SS) and 58 (MS) stems were detected, where 40 (SS) and 41 (MS) stems match to reference stems (49 stems). Therefore, a few trees exist in TLS scans but are not sampled in the reference data. This coincides with the finding from Liang and Hyyppä (2013) that around 10% of reference stems do not show in their SS dataset. The detection rates are 81.6% (SS) and 75.5% (MS). A relatively lower detection accuracy from the MS is due to a greater commission error.

Indeed, either SS or MS has a greater commission error than its omission error. Our algorithms detect several understory young trees and thin saplings not included in the reference data. In general, our detection rates are satisfactory, considering Brolly (2013)'s summary of nine papers that detection rates are between 22%–94% (SS), and 52%–100% (MS). The detection rate from SS alone is also competitive, in comparison to the common range of 68%–90% from SS (Liang et al., 2014b). It is not absolute though, for detection rate relies on plot conditions such as tree species, stem density, age, and elevation. For example, detection rates of four different species with over 40 trees in Liang and Hyyppä (2013) range from 81% to 100%.

2.3.3 Stem Form Extraction

A visualisation of the stem curve extraction is shown in Figure 2.7 for a single tree. The stem points are heavily shaded by branches and leaves at the height above 10 m (Figure 2.7a). The loss of stem points in upper stems above 10 m is also reported in Henning and Radtke (2006), and above 7.8 m reported in Bienert et al. (2007). The extraction algorithm used is able to identify the stem shape and taper through the canopy (Figure 2.7c) by segmenting branches (Figure 2.7b). Optimal region growing automation ideally requires the avoidance of *a-priori* estimates of stem or branch radius. A number of existing branch reconstruction studies use a predefined sphere as an initial guess of radius (Hackenberg et al., 2014; Hackenberg et al., 2015; Raumonen et al., 2015). The classified stem reaches a height of 14.8 m, 67% of field measured height. Heavy occlusion above that height prevents our stem classification algorithm from detecting further stem surfaces. In general, the average proportion of our classified stem height reaches 64% (SS) and 74% (MS), greater than the average proportion of FGI's software measured stem height 62% (SS) and 61% (MS). Stem diameter narrowing in the crown is another bottleneck of stem classification from branch clumps. A potential solution is to sacrifice time efficiency in order to gain classification accuracy. This could be, for example, finer region growing in terms of smaller R_s and W_{ir}, or greater classification tolerance in terms of larger D_t. Figure 2.7d plots the shape and radius of the extracted curve nodes. Visually, the overall curve shape is smooth. The occlusion of the crown introduces biases in the node direction above 12 m and radius above 6 m.



(**a**) (**b**) (**c**) (**d**)

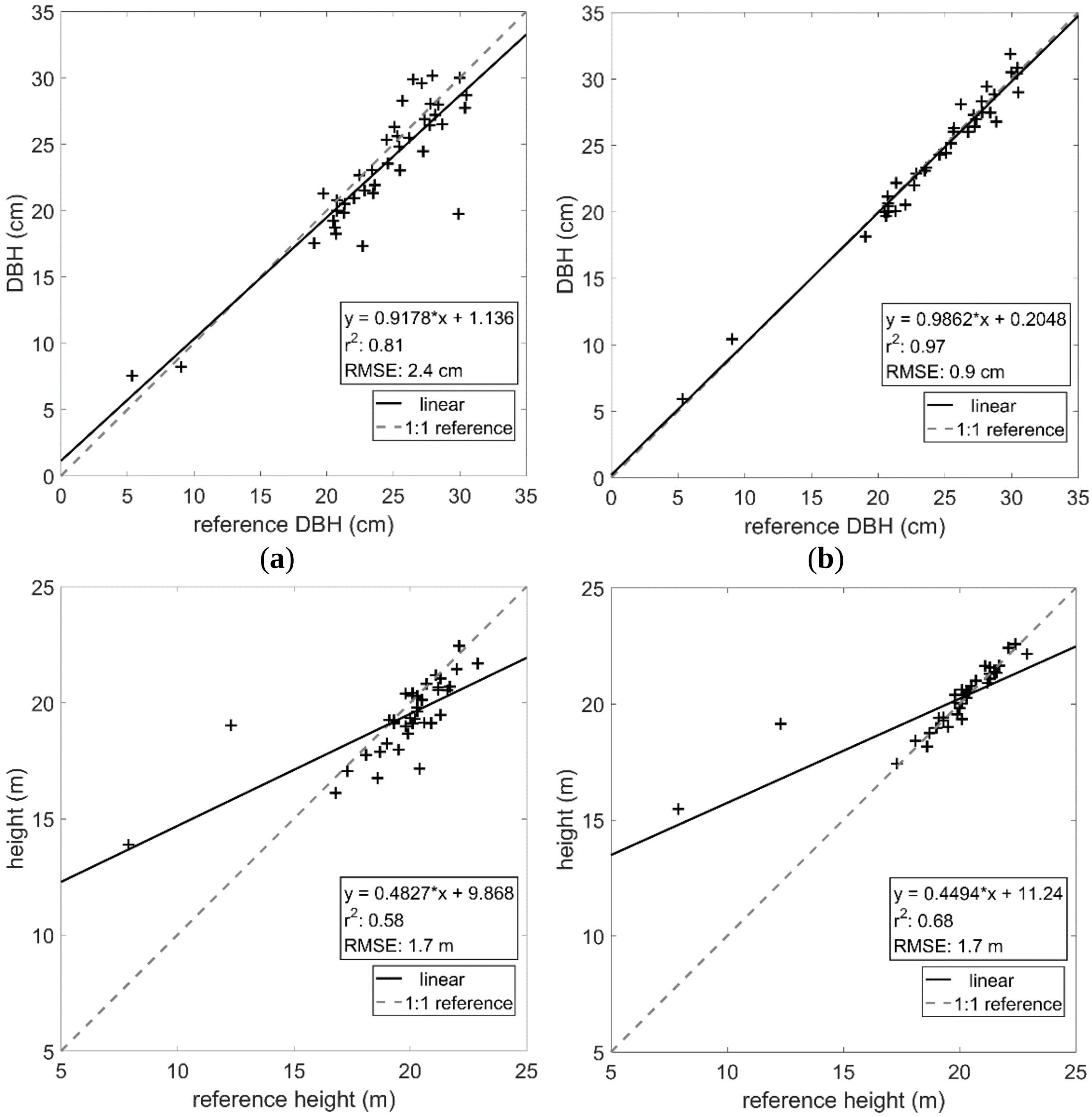
Figure 2.7 Automation outcomes for tree No.23 from plot-1 MS. (**a**) raw isolated point clouds, colored by point intensity; (**b**) segmentation of the point clouds within 1 m horizontal distance to the stem location, colored by the segment random identity numbers; (**c**) a vertical section cutting tree in half, with branches in blue and classified stem in white; (**d**) stem radius and stem curve, represented by colored stem node circles and red connection line of stem nodes, respectively. The results are shown in local Cartesian coordinate system with plot center as the origin.

Correspondence between TLS detected stem DBH, height, and form (for those identified within 0.5 m of the reference stems) and reference is provided in Table 2.2. According to Liang et al. (2014c)'s accuracy assessment of over five studies about stem attribute extraction, DBH RMSE ranges between 0.82–9.17cm (mean: 3.8 cm), and stem diameter RMSE between 1.13–7.0 cm (mean: 3.9 cm). Our DBH and stem diameter results from MS show improvement over the average at 0.9 cm RMSE, yet results from SS are less satisfactory at 2.4 cm RMSE. The difference of DBH RMSE 1.5 cm represents 6.5% of the mean reference DBH. Here, we define "error" as the absolute difference between measured and extracted. A right-tailed two-sample *t*-test of DBH error comparing SS and MS shows a *p*-value of 0.0008 at the 5% significance level, and the same test of height error has a *p*-value of 0.13 at the 5% significance level. It can be concluded that DBH error from SS is significantly greater than from MS and height error from SS is not. The detailed disparity between measured and extracted DBHs can be seen in Figure 2.8a,b. Systematic underestimation error can be found from Figure 2.8a where 67.5% of extracted DBHs from SS and 94.6% from MS are smaller than the reference. It is believed that the systematic error and bias are likely due to the circle-fitting algorithm's imperfection in terms of asymmetric stem sampling from the TLS. The measured and extracted heights are compared in Figure 2.8c,d. The leading error is presented by two obvious outliers that are far from the dashed regression line in both figures. The outliers show errors greater than 5 m. We visually examined the corresponding stems and found crown tops were actually overlaid by several higher crowns from neighboring trees. At such, tree crown overlapping contributes much of the error. The distance to scanner is plotted against DBH error (SS and MS) in Figure 2.9. From Figure 2.9a, it is clear that the largest DBH errors (>5 cm) from SS correspond to a distance greater than 10 m to the plot center. Indeed, with the increase of distance to plot center, stem surface points become thinner and more scattered (Litkey et al., 2008), and therefore challenge DBH extractions. In contrast, extracting DBH from MS is less affected by distance to plot center (Figure 2.9b), owing to additional corner scans.

This further indicate that using a circle-fitting model cannot really overcome the 3D sampling limitation from the field.

Table 2.2 Accuracy assessment for stem attributes.

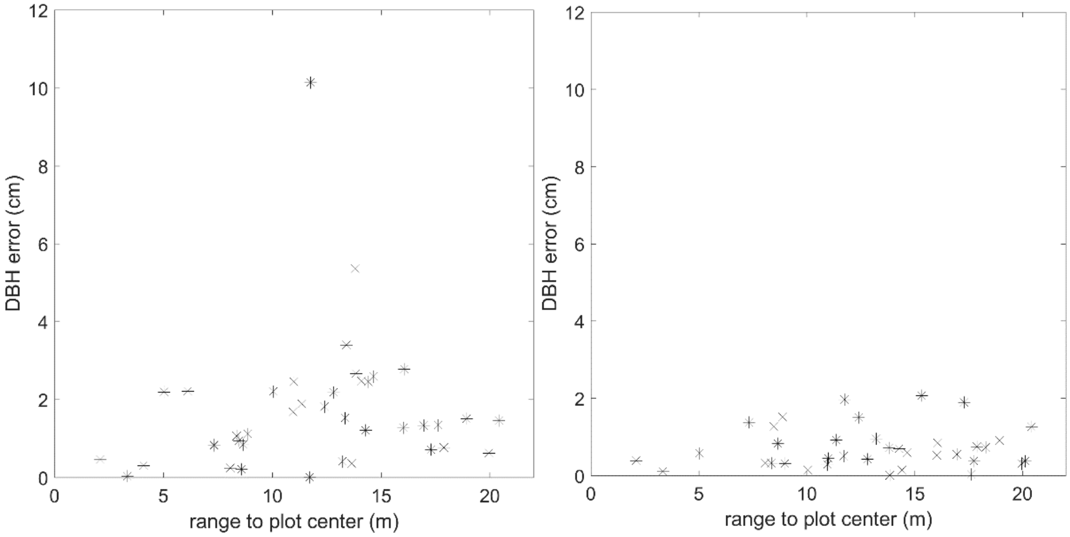
	RMSE	r^2
DBH (SS)	2.4 cm	0.81
DBH (MS)	0.9 cm	0.97
Height (SS)	1.7 m	0.58
Height (MS)	1.7 m	0.68
	Average RMSE	**Average** r^2
Diameter (SS)	3.2 cm	0.58
Diameter (MS)	2.4 cm	0.76



(**a**) (**b**)

(**c**) (**d**)

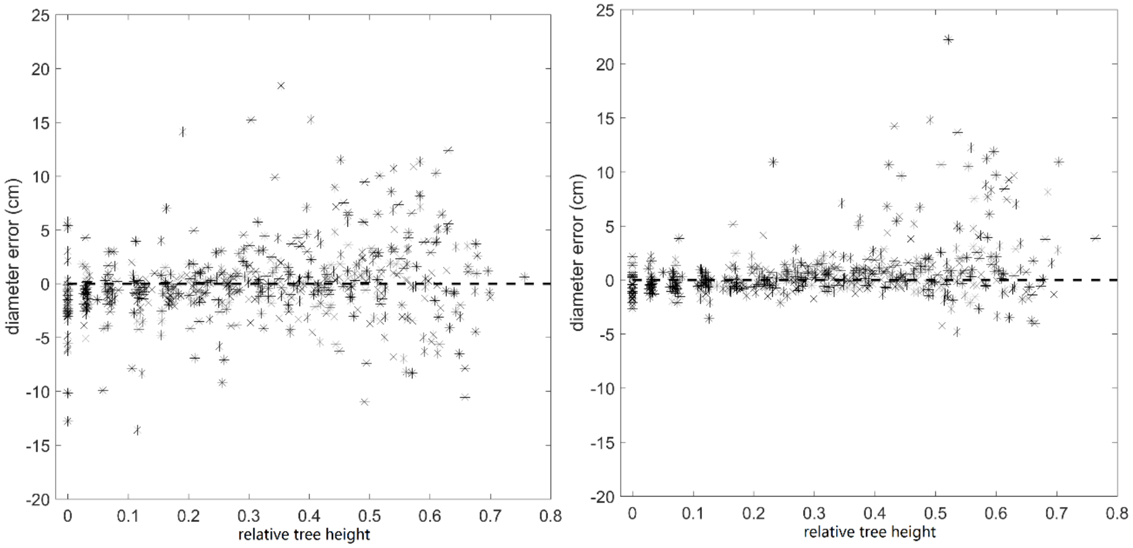
Figure 2.8 Scatter plot between the measured and the extracted: (**a**) DBH from single scan (SS); (**b**) DBH from multiple scan (MS); (**c**) height from SS; (**d**) height from MS.



(**a**) (**b**)

Figure 2.9 Scatter plot between range and DBH error: (**a**) from single scan (SS); (**b**) from MS. Here, error is defined as the absolute difference between the measured ground truth and the extracted.

Extracting full stem diameters is more challenging than extracting DBH. The increased difficulty is clear in the mean RMSEs of extracted stem diameters (SS: 3.2 cm, MS: 2.4 cm) in Table 2.2 compared to that of DBH (SS: 2.4 cm, MS: 0.9 cm). The average RMSEs of extracted diameters are 3.6 cm (SS) and 3.1 cm (MS) above half tree height, contributing 65.4% and 73.3% to total RMSEs, respectively. The relationship between diameter extraction errors and relative tree heights is illustrated in Figure 2.10. Outliers are mostly distributed near the ground and above 40% height. It is clear that more diameters are underestimated from SS (Figure 2.10a) than from MS (Figure 2.10b). Trying different circle-fitting algorithm may overcome this problem. A detailed comparison of different stem form extraction algorithms is illustrated in Liang et al. (2018).



(**a**) (**b**)

Figure 2.10 Scatter plot between relative tree height and diameter error: (**a**) from SS; (**b**) from MS. Here, error is defined as the absolute difference between the measured diameters and the extracted.

2.3.4 Time Efficiency

In the process of stem form extraction, searching for the neighboring points is accelerated by a voxel indexing method. The voxel indexing method first groups the points into voxels and any nearby voxels are then searched directly by the spatial index. This method reduces the computation and storage cost of iterative point cloud subdivision in the octree method. Partitioning of the growing region into clusters is also facilitated by the voxel indexing method. In our test, the overall computation time is reduced to half by applying the voxel indexing method.

Our overall processing time for the plot-1 using an Intel i7 2 GHz with 8 GB memory are 27 min (SS) and 2 h 28 min (MS), respectively. The processing time for MS is distributed approximately in Table 2.3. Note that processing time varies with the tree geometry, point number, scan quality, and computer specification. The numbers of points are 23,568,575 (SS) and 111,065,193 (MS). The point cloud files are 0.5 GB (SS) and 2.5

GB (MS) and isolated stem files had a median of 1 MB (SS) and 10 MB (MS), respectively. Table 2.3 illustrates that the time complexity of our algorithm is highly proportional to the point number, which is partly because our algorithms limit nonlinear traversal of point clouds. A countable processing time is essential to an unsupervised, ceaseless, and achievable product line of plot-size inventory. Foresters can rehearse processing smaller point clouds, determine desired point size and resolution, and then arrange a favorable processing timeline. Only a limited number of studies share information about processing time. Aschoff and Spiecker (2004) reported that their algorithms allocate 16.2 second per million points (sppm) to DTM creation, 10.8 sppm to tree recognition, but a large 865 sppm to noise reduction and export, based on a 2.0 GHz CPU. Bienert et al. (2006) provided sppm statistics for three SS plots and one MS plots using Intel Xeon CPU (2.4 GHz) with 1.0 GB RAM. Their algorithm of DTM filtering, segmentation and classification has around 240 sppm (SS) and 180 sppm (MS), and their algorithm of stem form extraction has 15–22 sppm (SS) and 23 sppm (MS), respectively. In comparison, our algorithms exhibit more stable sppm between SS and MS.

Table 2.3 Time distribution in the automation algorithms in minutes.

Process	**SS**	**MS**
Reading point cloud (sppm) *	1 (2.5)	4 (2.2)
Extracting and removing DTM (sppm)	1 (2.5)	4 (2.2)
Extracting stem location (sppm)	0.2 (0.5)	1 (0.5)
Isolating stem space (sppm)	1 (2.5)	4 (2.2)
Extracting stem form (sppm)	24 (61.1)	135 (72.9)
Overall (sppm)	27 (68.7)	148 (80.0)

* sppm: seconds per million points.

2.4 Conclusions

This study presents a new methodology for the automated extraction of tree locations, complex DBH cross sections and detectable stem heights within a plot with single and multiple TLS scan lines. The study uses automated methods to (a) filter and retain ground returns and high resolution variability of the ground surface using an automated, piece-wise interpolation function; (b) identify stem locations via rapid matching of 3D shapes within point clouds in entire 3D space instead of focusing on predefined slices or point-wise PCA described previously in the literature (Aschoff et al., 2004; Bienert et al., 2006; Maas et al., 2008; Schilling et al., 2011); and (c) applying an automated stem form extraction methodology for all identified trees within the plot such that issues associated with complex stem shape and point occlusion due to canopy interception of laser pulses are addressed. Although numerous steps are required to perform the analysis, the integration of these methods can be applied to answer numerous questions about canopy attributes that would otherwise require greater user intervention and processing time. For example, our segmentation algorithm can provide branch segments as a preliminary input to the cylinder fitting step in most branch reconstruction algorithms (Côté et al., 2011; Hackenberg et al., 2014; Livny et al., 2010; Newnham et al., 2015; Raumonen et al., 2013; Schilling & Maas, 2014). Differentiating branch segments is also an essential step to isolate overlapping trees (Raumonen et al., 2015; Yang et al., 2016). Our stem detection algorithm can also aid the process of multi-scan co-registration (Henning & Radtke, 2006).

While this study may not provide the most accurate results compared with Liang et al. (2014c), the automation of steps for complete plot-level assessment will reduce time spent in the field and more manually intensive TLS-based extraction methods (Dassot et al., 2012; Eysn et al., 2013). Our future work needs to improve the stem detection rate and the circle-fitting algorithm. The convolution filtering and local extrema extraction processes presented in this study are similar to the concept of the convolutional neural network (CNN) to be mentioned in the following chapters, except that the CNN model is a tunable process. It would be interesting to apply CNN specific for the stem detection and localization purpose. Algorithm versatility is also an aim of improvement. As Côté et al. (2011) pointed out, current TLS measurement still faces the problem of object occlusion, multi-scan oversampling, wind sway, and tree component identification. Solving these problems entirely is indeed a long run.

Chapter 3. Filtering Biomass Components from TLS Plot Scans

3.1 Context

Detailed stem form extraction from TLS improves inventory-based approach by quantifying stem volume in non-destructive and precise manners. Using empirical BEF at the plot level can convert stem volume to plot wood volume including other components like branch, root, understory, and saplings. Further evidence is found that reconstructing branch wood components of individual trees using QSM can attain high accuracy (Calders et al., 2015; Hackenberg et al., 2015). The QSM-based branch modeling can be a promising replacement of allometric-derived rules to convert plot-level biomass. One strong requirement of applying QSM at branch level is clear presence of branch points from TLS scans. Apparently, the requirement is hardly satisfied in leafy crown and understory. A reliable and versatile wood filtering algorithm is needed for handling all these complexities in 3D space.

The complexity of scanning scenarios is a constraint to developing successful wood filtering algorithms. Fortunately, artificial intelligence, especially the revolutionary deep learning network, is tailored to handle complex recognition and filtering problems (LeCun et al., 2015). In general, deep learning networks comprise multiple layers with millions of parameters that explain complex data structures and achieve high intelligence. Initial parameter values need to be trained in a gradual optimization process where gradients between the predicted and the desired are backpropagated and minimized. The configuration of layers, however, has various forms and depends on application type. A close application to our wood filtering case is image classification, which is also termed image semantic segmentation, under the context of computer vision and machine learning.

Correspondingly, the most popular deep learning network is the Fully Convolutional Network (FCN). As the name implies, FCN typically has hierarchical convolutional layers that extract abstract image features in hyperspace, then fully-connected layers that define wall-to-wall mappings between the abstract features and classes, and finally transposed convolution layers that interpolate class predictions to fine resolution. The convolutional layers are also termed Convolutional Neural Network (CNN) layers. Such a layer configuration of FCN has had significant success in practice for the capability of automatic feature selection and high-degree nonlinearity. Different FCN realizations and variations are reviewed in Garcia-Garcia, et al. (Garcia-Garcia et al., 2017).

Applying FCN to 3D wood filtering has several practical difficulties. First, training an FCN structure with 3D point clouds is time and storage challenging. So far, the convolution step of FCN from most deep learning programs does not support a discrete point format, and 3D point clouds need to be voxelized first. However, inputting the FCN structure with all voxels in fine resolution and entire scan range will easily lead to computation overflow. This tradeoff between computation load and 3D information abundance calls for additional transformation of 3D point data or FCN structure. Existing transformation methods involve projecting 3D points to multi-view or multi-feature 2D images (Boulch et al., 2017; Lawin et al., 2017; Yang et al., 2017), subdividing point clouds into blocks (Çi çek et al., 2016; Huang & You, 2016), or enabling sparse computation of CNN layers (Graham, 2015; Riegler et al., 2017). Second, a gradient vanishing problem arises. Interpreting point clouds as a uniform grid of voxels will lead to many empty voxels. This is mainly because only LiDAR point clouds only scans a target surface and will have empty gaps inside a target. A great number of empty voxels surrounding a non-empty voxel will usually depress or eliminate the contribution of non-empty voxel during the CNN convolution process using a 3D moving window. The depressing effect will pass on to deeper layers of the CNN. Similarly, backpropagation of gradients to shallower layers hardly affects non-empty voxels. This phenomenon is a symptom of gradient vanishing (Hochreiter et al., 2001), which significantly slow the training process or even lead to training failure. The gradient vanishing problem for the point cloud data has no sound solution yet, albeit compromises can be replacing CNN with a multi-layer perceptron structure that weakens spatial relationships (Qi et al., 2016; Qi et al., 2017). Third, it is difficult to have either ground truth or manual reference of wood points. The training process of FCN requires a considerable number of samples with correct labels. Yet it is laborious to either obtain ground truth by repetitively scanning a plot with and without leaves or create a manual reference by picking and labeling individual points from dense and noisy scans at branch level.

A 3D FCN structure was customized in this study, aiming to filter stems and branches from TLS point clouds in a robust and flexible manner. TLS scans were collected with varying scanning and geometric conditions from three sites dominated by three common species. An automatic region-growing method was developed in aid of manual reference generation. The filtering results from 3D FCN were finally input to a QSM model to test application possibility for plot-level wood delineation. From this experimental study, we hope to unveil the potential of applying deep learning methods in complicated 3D processing tasks, particularly in parsing forest plot scans, in pursuit of automatic and intelligent wood resource management.

3.2 Methodology

3.2.1 Data and Analysis Framework

In the summers of 2015 and 2016, plot scans were collected for three test monospecific sites dominated by three common species in Canada: sugar maple (*Acer saccharum*) from Vivian Forest (Ontario), trembling aspen (*Populus tremuloides*) from West Castle (Alberta), and lodgepole pine (*Pinus contorta*) from Cypress Hills (Saskatchewan). Maple, aspen, and pine plots have an approximate stem density of 400, 200, and 700 ha^{-1}, and average tree height of 17, 12, and 20 m, respectively. All sites were scanned with an Optech ILRIS HD (1535 nm) from plot center and three-to-five corners. The mean point spacing was between 10–20 mm and range accuracy was ±2 mm at a distance of 20 m from the scanner for all sites. Laser intensity values ranged from 0–255. For the sampling and evaluating purpose, all our scans were not trimmed in size. The maximum range of scans from the plot center is about 100-200 m.

From corner scans, we manually clipped common areas to be our sample areas. Our training sample included six maple areas, six aspen areas, and two pine areas, whereas our testing sample includes three maple areas, two aspen areas, and two pine areas. We intentionally selected samples with varying tree numbers and scanning conditions, with a quantitative summary in Table 3.1. We also created a sample quality ranking between ‘1’ and ‘10’ based on observation, explained in Table 3.2, indicating how difficult it is to infer inventory details from human interpretation. Point sparsity and occlusion are important references to our subjective quality ranking, which can be partly reflected by the high Pearson’s correlation (r) of 0.72 between quality ranking and point spacing, and by the low p value of 0.0003. It is noteworthy that a quality ranking of ‘1’ does not mean perfect quality. All of our samples encounter a certain degree of point occlusion, especially in upper crown. Such scan quality imperfection and inconsistency are notorious characteristics of natural forest scans and this is where the challenge is. Great quality variation can help develop a robust or universal filtering model to maximize plot data usage.

Table 3.1 Quantitative characteristics of each training and testing sample. DBH min-max means minimum and maximum DBH extracted from sample point clouds at a height of 1.3 m. A DBH with 0 value means unmeasurable. Height means vertical distance between lowest and highest points. Point spacing means median nearest neighbor distance. Quality ranking is described in Table 3.2.

Dataset (Size)	Sample ID	Tree Count	DBH Min-Max (cm)	Height (m)	Point Spacing (cm)	Quality Ranking
Training sample (14)	maple 1	2	9.8–33.3	24.4	0.9	2
	maple 2	3	7.5–11.7	17.5	0.6	7
	maple 3	4	0–39.4	25.3	1.1	3
	maple 4	3	36.8–42.8	28.8	2.0	4
	maple 5	2	0–0	20.1	3.1	4
	maple 6	1	0–0	25.0	2.2	6
	aspen 1	4	0–24.2	14.1	3.2	6
	aspen 2	1	0–0	12.5	4.7	10
	aspen 3	2	0–28.3	14.9	0.9	1
	aspen 4	6	9.9–22.0	33.6	1.7	3
	aspen 5	7	0–0	11.8	7.9	9
	aspen 6	3	10.1–22.4	13.2	2.7	4
	pine 1	4	17.9–29.9	23.6	1.0	1
	pine 2	1	0–0	23.3	1.0	5
Testing sample (7)	maple 7	4	10.7–11.9	27.3	2.1	8
	maple 8	2	5.9–34.9	18.5	0.6	2
	maple 9	2	0–54.0	26.0	1.8	3
	aspen 7	1	23.7–23.7	15.7	0.6	2
	aspen 8	1	34.6–.34.6	14.9	0.5	1
	pine 3	1	42.3–42.3	21.9	2.4	5
	pine 4	4	0–0	23.0	0.5	3

Table 3.2 Criteria for sample quality ranking.

Quality Ranking	Rationale
1	explicit branch and stem forms, very low occlusion degree
2	moderate branch and stem details, full stem form, very low occlusion degree
3	full stem form, low occlusion degree, few branch details
4	Fragmented stem form, moderate occlusion degree
5	partial stem form, low occlusion degree
6	partial stem form, moderate occlusion degree
7	partial stem form, high occlusion degree
8	partial stem form, sparse branch or stem points, high occlusion degree
9	indiscernible stem points
10	indiscernible stem points, surrounded with noisy points

All sample areas were labeled with three classes: ‘stem’, ‘branch’, and ‘other’ components. The labeling step was semi-automatic, to be described in Section 3.2.1. For completeness, stems and branches of understory saplings (height < 4 m) were also labeled in the sample areas. Finally, center scans and corner scans were manually co-registered in preparation for plot-level filtering visualization and tree reconstruction.

Our experimental design of training, evaluating, and applying a 3D FCN filtering model in wood reconstruction is presented in Figure 3.1. Detailed explanations will be provided in the following sections, with the process of generating training and testing reference in Section 3.2.1, specification of the 3D FCN in Section 3.2.2, and schemes of plot-level QSM to retrieve tree structure and volume in Section 3.2.3.

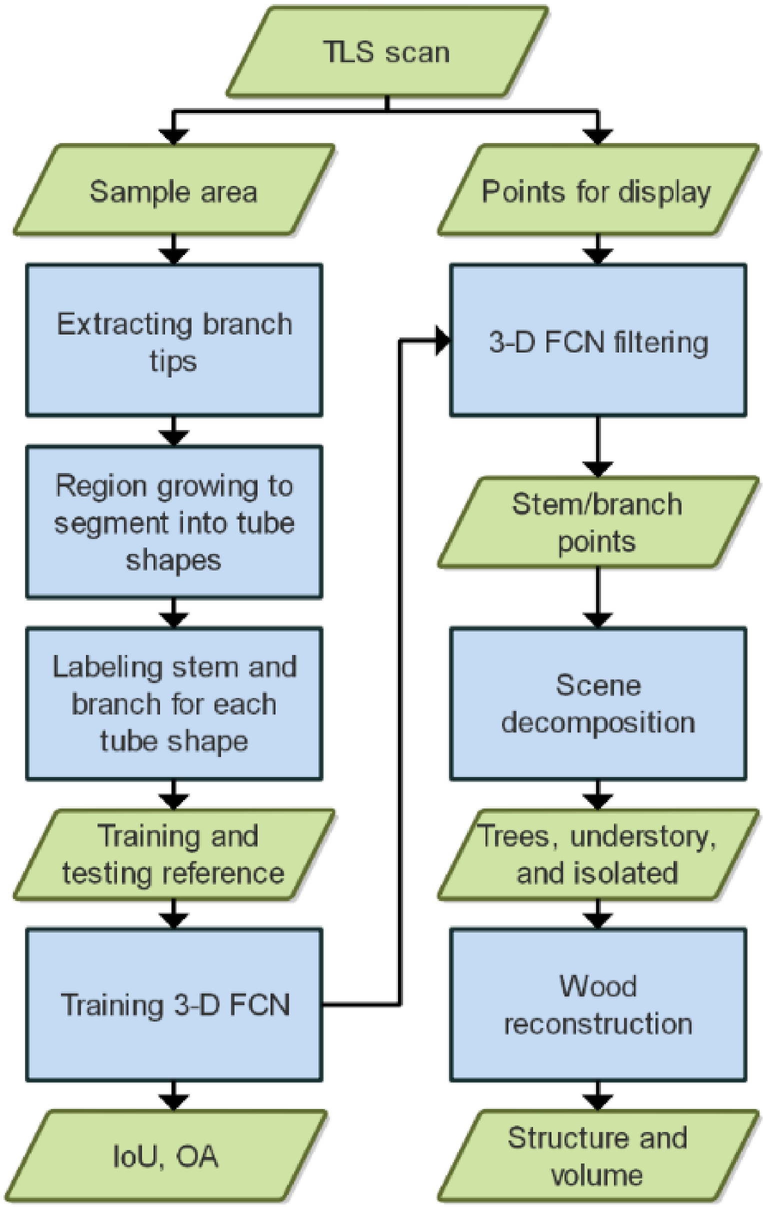


Figure 3.1 A framework of wood filtering and reconstruction from TLS plot scans. IoU and OA are both accuracy evaluation metrics. IoU: intersection over union for a class. OA: overall accuracy.

3.2.2 Labeling Reference Points

Per-point labeling of subtle branches, especially in crown and understory areas, can be difficult. This study relied on an automatic segmentation algorithm to help locate tube-shaped branches and filter out noise (Xi et al., 2016) (Chapter 2). Generally, the algorithm started from selecting a few core points from point clouds, which iteratively incorporated neighboring points and eventually grew to be tube-shaped regions. Major requirements of the region-growing routine were to constrain growth direction and tube width changes. As a result, most points were segmented into tube-shaped regions, while other remaining points, especially from leaves and ground, had bulky shapes. The tube-shaped regions were extracted based on their three-dimensional features from principal components analysis PCA (Pfeifer & Winterhalder, 2004): any regions with linearity >0.4 and sphericity <0.3. The extracted regions were then exported to CloudCompare Open Source Software for further editing. Specifically, the tube-shaped regions were manually trimmed and refined to represent clean wood segments. Wood segments, with thick long shape, near-vertical direction, tight connection to base part, or high laser intensity, were preferably recognized as stems. The remaining wood segments were thus intended to be branches. We labeled stems as class 2, branches as class 3, and other points as class 1.

The above-stated region-growing algorithm was an iterative process, meaning that later region-growing steps were affected by errors in earlier steps. It was therefore critical to ensure correctness of initial region-growing steps. For example, locating initial region-growing points should avoid areas with ambiguous shapes, such as joints between branches. Our experiments suggested that region-growing initialized from branch tip points would yield most correct shapes and cause less conflict with following region-growing steps. We then proposed a simple method of locating branch tips, to improve the initialization step in Xi et al. (2016) (Chapter 2). First, for operational convenience, all points were converted to voxels at a resolution of $5 \times 5 \times 5$ cm^3 (xyz). A metric termed mean curvature was used to describe the spatial local sharpness (Figure 3.2). Specifically, for each voxel, we calculated its unit vectors directing to all neighboring voxels. The average of those unit vectors was a vector (red arrow in Figure 3.2), the direction and amplitude of which represented surface normal and sharpness degree, respectively. This amplitude value was indeed a discrete expression of surface mean curvature (Sorkine, 2006). Using surface mean curvature, branch tips were coarsely located where a voxel's mean curvature exceeded 0.75. These voxels representing branch tips were regarded as initial region-growing points. Each branch tip was searched in turn to grow initial regions, until no tips were available. Then any arbitrary point was selected to continue region-growing until no points were selectable. Both region-growing and initialization algorithms were programmed in MATLAB. The region growing-based segmentation is based on local geometry and connectivity. The segmentation performance is directly affected by the scan occlusion and leaf noise, but is not sensitive to varying point densities and branch geometries among different trees and species.

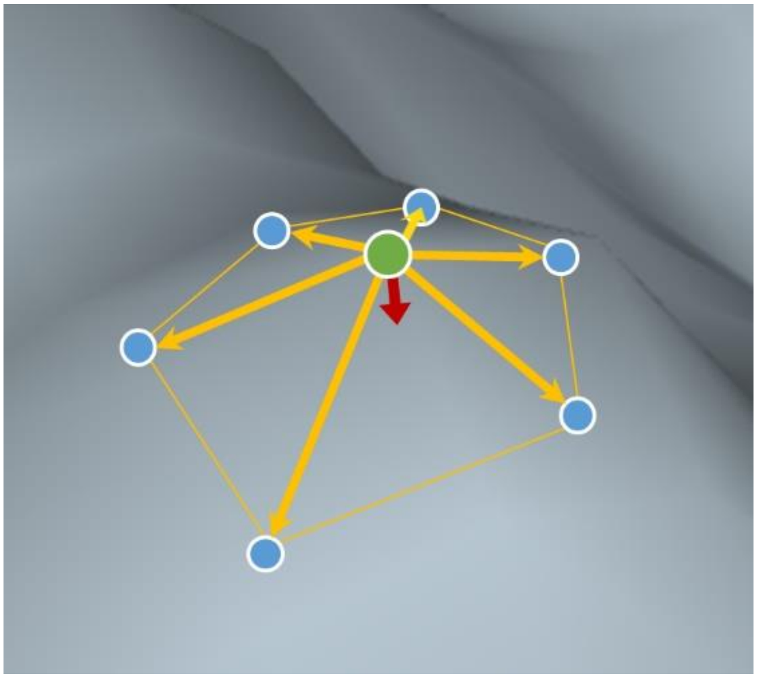


Figure 3.2 A sketch of surface normal and surface mean curvature (red arrow), calculated as average of unit vectors (yellow arrow) pointing from center point (green dot) to neighboring points (blue dots).

3.2.3 Configuring 3D FCN

Our 3D FCN model and evaluation steps were programmed in Tensorflow (GPU edition). We adopted the point cloud subdivision approach (Çiçek et al., 2016; Huang & You, 2016) and limited the model input to be a block of 128 × 128 × 128 voxels, with a voxel size of 5 × 5 × 5 cm^3. Larger block size and smaller voxel size were tested but could cause computing overflow problems of our graphic card memory. Based on our preliminary tests, wood classification accuracy would decline slightly with a smaller block size or a greater voxel size, but drastically with a block size smaller than 64 × 64 × 64 voxels or voxel size greater than 10 × 10 × 10 cm^3. We cut raw point clouds evenly into blocks and then shuffled blocks randomly before the training process. Each voxel was assigned with three attributes, derived from summary statistics of all points within that voxel. The first attribute was binary, with 1 meaning point occupancy and 0 otherwise. The second attribute was the average of the laser intensity values, whereas the third attribute was the average of the point heights (z value). With three attributes, the final FCN input was a 128 × 128 × 128 × 3 block. In addition, the class label of a voxel was calculated as the dominant label from all points within the voxel. If a voxel was empty, its class was set to be 0. Hence, the total class number was four.

Our FCN structure (Figure 3.3) was customized based on the FCN-VGG-16 structure described in Long et al. (2015), where VGG denotes Oxford's Visual Geometry Group. Our FCN structure expanded the 2D layers of the FCN-VGG-16 to a 3D version. It was basically a joint of five 3D CNN layers to subsample and extract important features, three fully connected layers to expand feature dimension and weight high-level features, and three transposed convolution (TC) layers to oversample and recover spatial resolution. Each CNN layer started with two or three convolution layers to filter a block and ended with a max pooling layer to extract local extrema. The TC layers have the problem of losing spatial details and therefore they are fused with the low-level CNN layers for better recovering spatial resolution. We designed the fusion strategy between TC and low-level pooling, shown by the plus sign in Figure 3.3. Our strategy was fusing 1st TC directly with 3rd pooling results, and 2nd TC with 2nd pooling. The omission of fusing 4th pooling results was intended to reduce blurring degree on final prediction, since the spatial resolution of 4th pooling block is low (16x16x16) for fusion. The final prediction output a 128 × 128 × 128 × 4 block, with last dimension denoting the probability of each class. The final class label was predicted as the maximum value in the last dimension of the probability block.

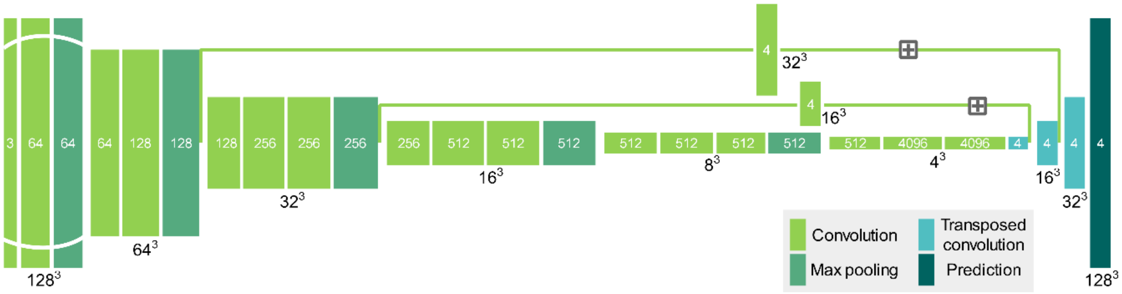


Figure 3.3 Configuration of 3D FCN structure. Colored bars denote different types of network layers, black numbers outside bar denote input size in 3D domain, white numbers inside bar denote input size in feature domain, and gray plus sign denote addition of layer outputs.

We trained our model from scratch because existing 2D pre-trained models were hardly adaptable to our scenes and data structure. Our convolution filter size was 3 ×3 ×3, and dropout ratio was 0.5, following the routines of CNN. The weights of convolution and fully-connected layers were initialized using default values in Tensorflow, and the weights of transposed convolution layer were based on trilinear interpolation. To train our 3D FCN model, errors (or disagreement) between probability block and reference, namely loss function, were measured using a mean softmax cross entropy metric (Bridle, 1990). Nevertheless, directly minimizing this loss function could encounter the gradient vanishing issue. Our solution was to calculate loss function only based on non-empty voxels in the probability block. We used an Adam optimizer (Kingma & Ba, 2014) with an initial step of 10^{-4} to iteratively minimize the loss function. The setting of initial step was based on trial and error experiments. It was observed that loss function from only non-empty voxels could always descend to a minimum level, albeit the convergence process was slow and variable.

Prediction accuracy (or filtering accuracy) was defined as an agreement between predicted and referenced labels for this study. Three accuracy metrics were adopted: Overall Accuracy (OA), Intersection over Union (IoU) per class, and its mean value (mIoU) (Garcia-Garcia et al., 2017). The OA divided the total number of matched labels between prediction and reference by the total number of points. It represented an overall matching degree. The IoU was class-specific, which divided the total number of matched labels by the total number of labels for one class. The mIoU was an average of IoUs among all classes. Compared to OA, mIoU weighed per-class accuracy equally, and therefore less influenced by the accuracy of dominant classes. The metrics of OA, IoU, or mIoU all ranged from 0–1, with higher value indicating greater agreement.

3.2.4 Reconstructing 3D Tree Geometries

We designed a follow-up wood reconstruction model (or QSM) at plot-level to showcase a potential application value of our 3D FCN. The results of FCN filtering were labelled voxels. We extracted stem points from stem voxels, and branch points from branch voxels. The non-wood points were discarded, as well as the height and intensity attributes from wood points. Only 3D coordinates of wood points were reserved for the reconstruction model. The wood reconstruction modeling (or QSM) was a top-to-bottom and bottom-to-top process: decomposing the plot to individual trees, extracting nodes from branch or stem points, connecting nodes along separate branches or stems, and finally connecting branches to stems. Success of all reconstruction steps relied significantly on clean filtering of stems and branches from the 3D FCN. All wood points were assumed to be tube-shaped.

At the plot level, we used a simple region-growing algorithm to decompose plots into individual trees. Region growing process started from an arbitrary point, kept incorporating its buffer points within 0.1 m, and terminated when no buffer points were found. Results were isolated components of branches and stems. We applied a simple rule to merge isolated stem components into complete individual stems. Any two stem components were identified as being from the same stem if one stem component was completely above the other, and the upper components fell within five times stem radius of the lower component. Then, with individual stems identified, branch components were assigned to their nearest stem, together to form an individual tree. We limited the nearest distance between branch components and stem to be no greater than 2 m, otherwise the branch components would be identified as isolated branches. We also identified any branches not higher than 4 m from the ground as understory wood. Those branches were not assigned to their nearest stem. Consequently, a plot of wood points was decomposed into individual trees, isolated branches, and understory wood.

At the individual-tree level, stem and branch points were segmented respectively using the same region-growing algorithm as described in Section 3.2.2, except for filtering tubes based on linearity and sphericity criteria. The resulting grown regions were evenly sliced along their main axis at 1 cm intervals, to form a sequence of nodes representing the tube. The main axis direction was estimated from PCA (Xi et al., 2016), and each node location was at the center of the slice. At each node location, we applied a circle-fitting algorithm for its neighboring points (Gander et al., 1994). The search radius of neighboring points was set to be three times the tube width retrieved from the region-growing process. The fitted circle center became the new node location, the circle radius became node radius, and the circle plane normal becomes the node direction.

At the individual-stem or branch level, nodes were connected based on a weighted similarity score (Equation 3.1) utilizing distance D, angle θ, and radius R:

$$\begin{gathered}\mathrm{Score}_{2,1} = (w_1|\tan\theta_1| + w_2|\tan\theta_2| + w_3D + w_4|\Delta R|)^{-1}, \\ where\ \theta_1, \theta_2 < 90° \end{gathered} \tag{3.1}$$

Figure 3.4 has intuitive illustration of these geometric terms, with D meaning distance between two nodes, θ included angle between node direction and node connection, and ΔR node radius difference. Any large θ, ΔR or D can lead to a low similarity score. The weight of each term was customizable, and this study empirically set $w_1 = 0.4, w_2 = 0.4, w_3 = 0.1$ and $w_4 = 0.1$. Using this set of weights can rapidly approximate consistent and smooth curves for varying branch geometries among all our scan datasets including the complex aspen trees. Connection results are not sensitive to the weight choice because in most situations the best potential node have the smallest θ, ΔR or D at the same time, with a similarity score much smaller than the second potential node. Ambiguity occurs only in highly occluded or noisy area. Our process of node connection started with an arbitrary node, searched its neighboring nodes within 0.1 m, connected nodes with highest similarity scores, and repeated search and connection, until no node is available. In rare situation, nodes have $\theta \geq 90°$ which were discarded to avoid turn-around of connection lines. We then reversed the direction of the starting node and performed the same connection process, so that both sides of starting nodes could be searched and connected. The entire connection process produced individual stems and branches.

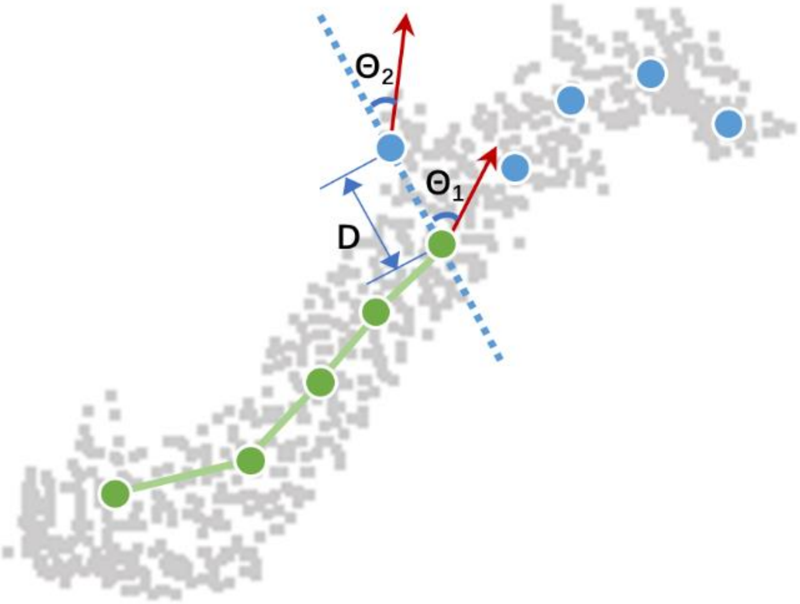


Figure 3.4 An illustration of similarity score composition. Gray points are example of scanned branch points, green dots are connected nodes, blue dots are candidate nodes to a target node, D is distance between target node and one of the candidate nodes, and θ is included angle between their connection (dashed line) and node direction (red arrow). For clarity, fitted circles around each node are not shown here.

At the individual-tree level, each branch, only represented by its starting and ending nodes, was connected to the stem or another branch based on the similarity score shown in Equation 3.2:

$$\mathrm{Score}_{2,1} = (w_2|\tan\theta_2| + w_3 D)^{-1}, where\ \theta_2 < 90° \tag{3.2}$$

Compared to Equation 3.1, no penalty was required for radius difference or direction difference between target and connection line. The connection process was iterative. First, branches nearest to the stem were connected to a stem node with highest similarity score. This branch-to-stem connection shaped a general 3D tree form. Then, remaining branches nearest to the tree form were connected to it based on a similarity score, and so forth, until no branches were available. The resulting tree form was further refined by combining similar branches (Figure 3.5). Any two branches with similar branch directions (<50 ° difference) would join at their shared node, if the joint branch were to be longer than both branches.

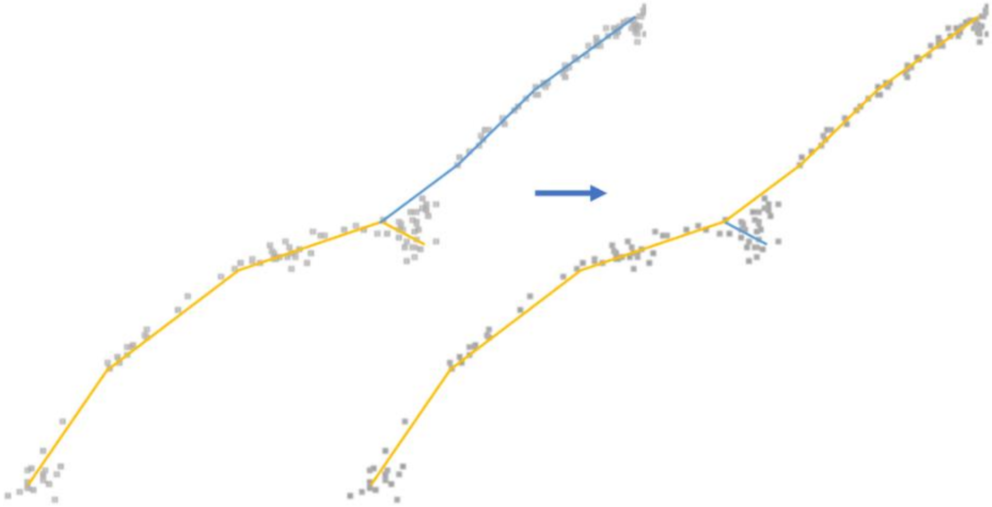


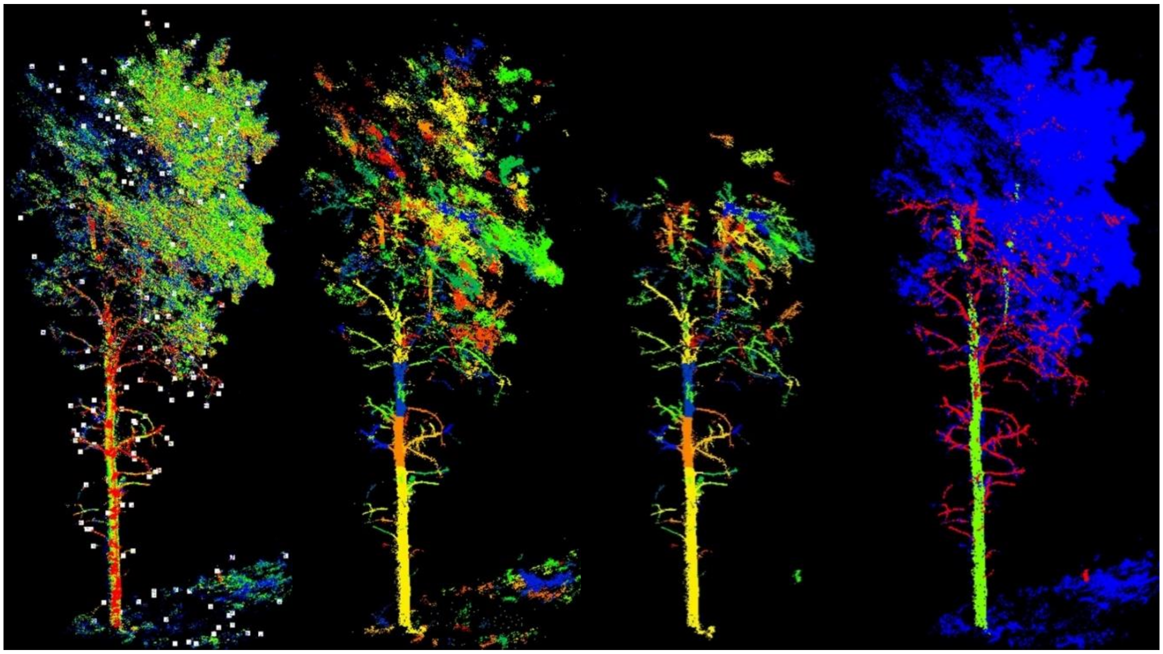
Figure 3.5 Example of branch connection refinement. Gray dots are scanned branch points, yellow and blue lines are connection lines. The right panel shows a more natural way of connection than the left.

Returning to the plot level, in addition to individual trees, isolated branches and understory wood were simply reconstructed following the steps of node connection. We did not apply branch-to-branch connection to avoid an excessive number of plausible connection lines. Wood volumes of individual trees, isolated branches and understory were all computed. All the QSM parameter settings were the same for different plots.

3.3 Results and Discussion

3.3.1 Reference Creation

Results of branch tip extraction from ‘aspen 3’ are shown as white dots in Figure 3.6a. Initialized with these branch tips, our region-growing algorithm can segment point clouds mostly into tube shapes. Those segments are assigned random colors in Figure 3.6b to visualize differences. After segment-wise PCA filtering, manual selection and minor trimming, the resultant wood reference is displayed in Figure 3.6c. The stem and branch points after splitting are shown in green and red, respectively, whereas other points are in blue (Figure 3.6d). Two trees are covered in this sample area, with only one tree exposing its main stem. Both trees contain many ghost points, particularly near the crown area, due to the difficulty of interpretation and trimming. These are typical quality issues in preparing reference data which hinder deep learning algorithms from recognizing branches correctly. Yet eliminating reference errors simply from human interference is limited and costly. It is anticipated in the future that a benchmarking reference dataset, e.g. both leaf-on and leaf-off scans or virtual forest simulation scans, could be developed for an algorithm optimizing purpose where scene complexity and data quality could be finely controlled.



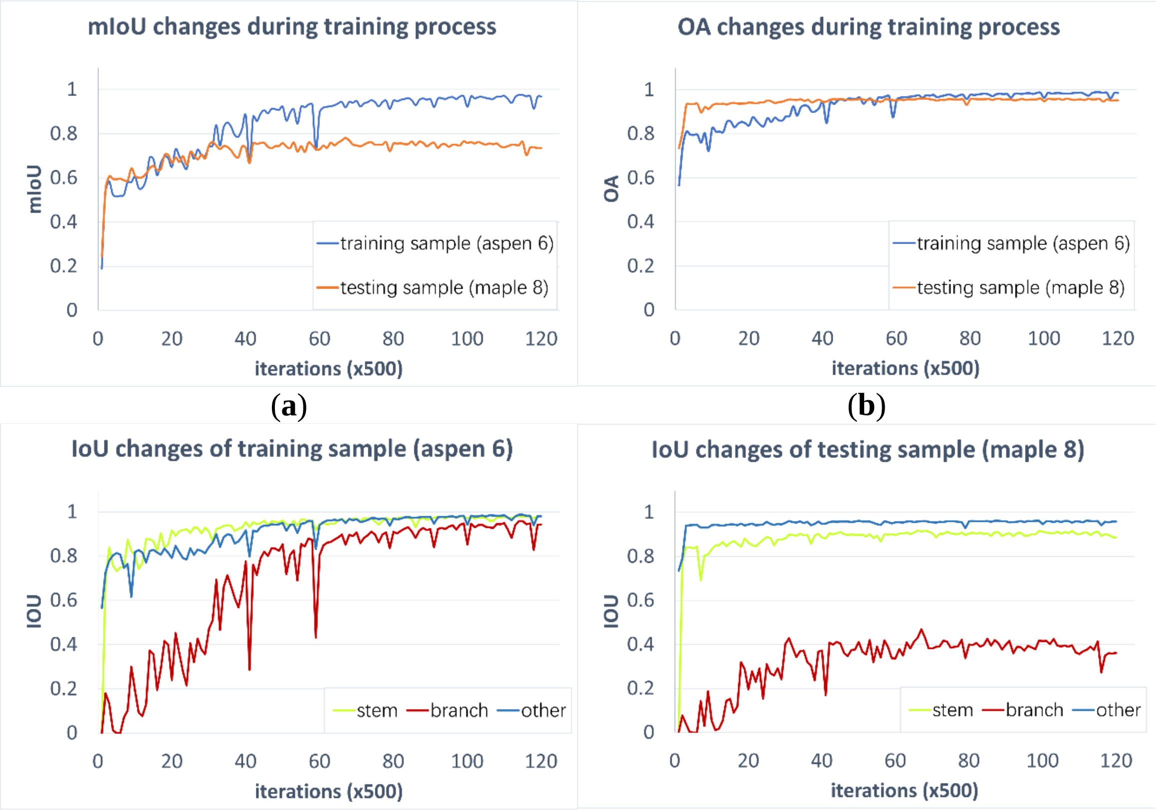
(**a**) (**b**) (**c**) (**d**)

Figure 3.6 Example of preparing aspen reference sample (‘aspen 3’): (**a**) clipped point clouds colored by laser intensity, overlaid with branch tip extraction in white dots; (**b**) region-growing segments in random colors; (**c**) segments with only stem and branches; and (**d**) reference tree with stem, branch, and other components classified in green, red, and blue colors, respectively. This sample area covers two trees: one exhibiting detailed stem form and the other showing only partial upper stem due to occlusion.

3.3.2 3D FCN Filtering and Evaluation

Figure 3.7a,b tracks accuracy changes during the training process for a training sample and a testing sample, corresponding to the 'aspen 6' and 'maple 8' in Table 3.1, respectively. A total of 60,000 training iterations are presented, corresponding to a running time of 30.19 h using a computer with Intel® Core™ i7-6700K 8 × 4.00GHz, 32GB RAM and NVIDIA GeForce GTX 1070 8GB RAM. As iteration increases, both training sample and testing sample show an increasing trend for mIoU and OA, with only one or two abrupt slumps, indicating our 3D FCN is a global optimization process. Note that an IoU or OA of 1.0 means perfect prediction. After 30,000 iterations, mIoU and OA of training sample converges to 0.98, indicating the strong fitting ability of the 3D FCN. On the other hand, mIoU and OA of testing sample levels off to a limit of 0.76 and 0.96, respectively, after 10,000 iterations. This result indicates that the generalizability of FCN would saturate and that the later learning phase after 10,000 iterations may only focus on classifying marginal features. The progressive classification manner by feature complexity is also observed in Bilal et al. (2018).

The testing sample has a high OA limit but relatively lower mIoU limit. This result indicates that the majority of wood points are filtered accurately, but some classes may be predicted incorrectly. Class-wise IoU changes for the training and testing samples are thus examined in Figure 3.7c,d. An increasing trend is shown for all classes of both training and testing samples. Apparently, branch points, due to geometric complexity, have slower and lower convergence values than stem and other points. Therefore, the FCN model can identify simple classes in the early phase, but learning difficult classes takes much longer and may not achieve the same high accuracy. The testing sample has a low branch IoU convergence of 0.4, which also explains the low mIoU in Figure 3.7a. The branch IoU issue can be caused by imperfect branch reference and complex branch structure, which should be mitigated by adding training samples, or refining the deep learning model with higher spatial resolution and stronger feature extraction ability.

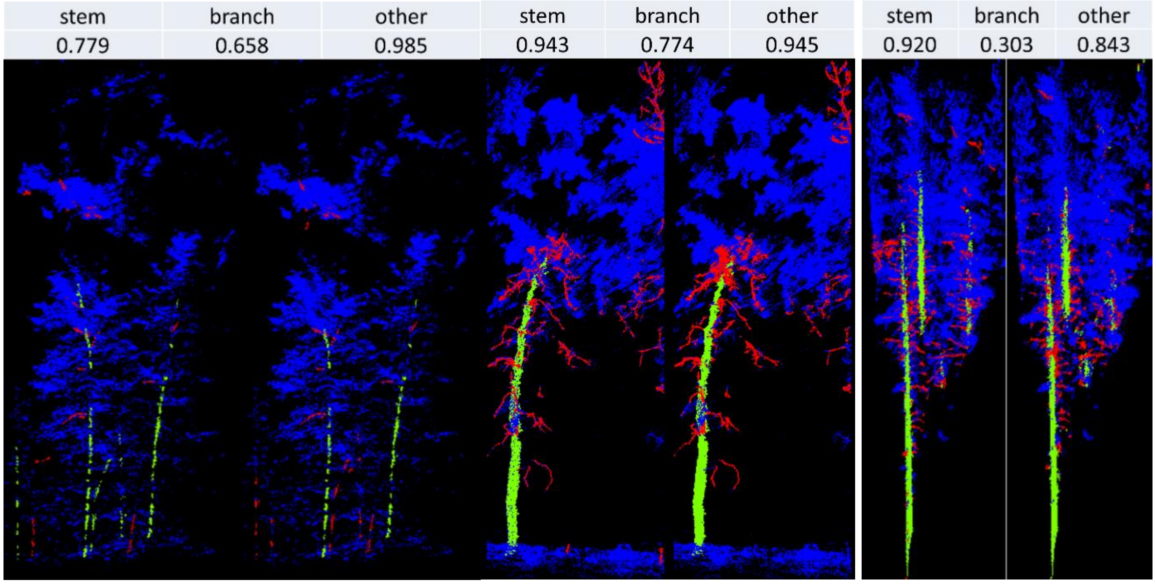


(**a**) (**b**)

(**c**) (**d**)

Figure 3.7 Relationship between accuracy changes and training iterations, for 'aspen 6' from training dataset and 'maple 8' from testing dataset: (**a**,**b**) fluctuation of mIoU and OA; (**c**,**d**) fluctuation of stem IoU, branch IoU, and other components' IoU.

Filtered wood points and reference can be visually contrasted using examples in Figure 3.8a–c. The main wood points are recognized and most difference occur in crown branches in Figure 3.8a,c, understory stems in Figure 3.8a and ghost points in Figure 3.8b,c. The classified wood appears slightly fatter than reference wood partly due to the setting of 5 cm voxel resolution.



(**a**) (**b**) (**c**)

Figure 3.8 Visualization of reference trees (left of each panel) and filtered trees (right of each panel) from testing dataset, including (**a**) ‘maple 7’, (**b**) ‘aspen 8’, and (**c**) ‘pine 4’. Per-class IoU is denoted on top of each panel. Stem, branch, and other points are in green, red, and blue colors, respectively.

The final accuracies of all training and testing samples are provided in Table 3.3. An average of stem IoU, branch IoU, ‘other’ IoU, mIoU, and OA is 0.90, 0.72, 0.96, 0.86, and 0.97 for the training sample, respectively, and 0.89, 0.54, 0.93, 0.79, and 0.94 for the testing sample, respectively. The mIoU shows no significant difference ($p = 0.15$) between training and testing samples based on a two-sample *t* test, implying that deep learning model is generalizable. The mIoU difference between the three species is also insignificant ($p = 0.16$) based on an ANOVA test using all samples. This result presents a possibility of having a universal wood filtering solution without need for exhaustive sampling of species.

Not all IoU results from Table 3.3 are satisfactory. Many factors could contribute to the low IoU situation, e.g. geometrical complexity and details. Results from a left-tailed *t* test show that branch IoU is significantly lower than stem IoU ($p < 0.001$), probably due to complex forms of branches. More details, denoted by a smaller point spacing, would lead to a lower mIoU. This statement can be indicated by a high Pearson’s *r* of 0.51 between point spacing from Table 3.1 and mIoU from Table 3.3. Similarly, worse quality ranking (Table 3.1) tends to produce higher branch IoU ($r = 0.39$) and lower stem IoU ($r = -0.21$). As quality ranking descends, stem form becomes fragmentary or blank at a rising risk of misclassification, whereas branches become simplified and conducive to classification. However, a clear understanding of IoU variation requires systematic investigation of model configuration, reference uncertainty and sample sufficiency, which is beyond the scope of this study.

Table 3.3 3D FCN filtering accuracies (IoU, mIoU, and OA) among training and testing samples.

Dataset (size)	Sample ID	IoU (Stem)	IoU (Branch)	IoU (Other)	mIoU	OA
Training sample (14)	maple 1	0.974	0.501	0.950	0.808	0.960
	maple 2	0.278	0.361	0.973	0.537	0.968
	maple 3	0.979	0.764	0.955	0.899	0.973
	maple 4	0.943	0.835	0.935	0.904	0.956
	maple 5	0.860	0.489	0.995	0.781	0.975
	maple 6	0.980	0.933	0.996	0.970	0.996
	aspen 1	0.972	0.562	0.952	0.829	0.964
	aspen 2	0.970	0.988	0.998	0.985	0.998
	aspen 3	0.929	0.563	0.935	0.809	0.942
	aspen 4	0.921	0.728	0.950	0.866	0.953
	aspen 5	-	1.000	1.000	1.000	1.000
	aspen 6	0.933	0.863	0.968	0.921	0.967
	pine 1	0.984	0.792	0.942	0.906	0.975
	pine 2	0.940	0.742	0.907	0.863	0.933
Testing sample (7)	maple 7	0.779	0.658	0.985	0.808	0.981
	maple 8	0.936	0.308	0.954	0.733	0.955
	maple 9	0.704	0.323	0.888	0.638	0.873
	aspen 7	0.963	0.629	0.958	0.850	0.963
	aspen 8	0.943	0.774	0.945	0.887	0.961
	pine 3	0.980	0.782	0.930	0.898	0.955
	pine 4	0.920	0.303	0.843	0.689	0.889

Attribution of input voxels plays a vital role in our 3D FCN filtering accuracy. As Table 3.4 shows, with only point occupancy, the 3D FCN model can still achieve an impressive accuracy, for example, a stem IoU of 0.922 for the ‘aspen 6’. It is clear that the

3D FCN filter mostly relies on local spatial pattern for classification. Adding the intensity attribute shows a slight accuracy improvement of 0.02 on average. By contrast, introducing the height attribute has greater accuracy increase of 0.07 on average, because branch availability is height-dependent.

Table 3.4 Effect of voxel attribution on 3D FCN filtering accuracies (per-class IoU, mIoU, and OA) at the 25,000th iteration.

Sample ID	Attribute Combination	IoU (Stem)	IoU (Branch)	IoU (Other)	mIoU	OA
aspen 6	occupancy	0.922	0.630	0.878	0.810	0.911
	occupancy + intensity	0.924	0.648	0.881	0.817	0.914
	occupancy + intensity + height	0.958	0.854	0.938	0.909	0.958
maple 8	occupancy	0.789	0.284	0.951	0.674	0.923
	occupancy + intensity	0.847	0.355	0.960	0.721	0.942
	occupancy + intensity + height	0.911	0.445	0.960	0.772	0.960

Traditionally, simple metrics such as intensity threshold are used to filter wood points (branch + stem). The accuracy of applying an intensity threshold filter to our sample is examined in Figure 3.9. We represent wood filtering accuracy with the ‘other’ IoU metric. For each sample, ‘other’ IoU is plotted against a range of intensity thresholds, and the highest ‘other’ IoU is marked with a red triangle. It is obvious that no universal threshold exists that can promise the highest wood filtering accuracy for all the samples. The upper boundary of ‘other’ IoU from intensity filter also vary drastically between 0.5 and 0.9. In comparison, ‘other’ IoU from our FCN filter, denoted by the blue starred line, is drawn in Figure 3.9. An overwhelming advantage of using FCN filter is presented, considering the accuracy level and stability among all the samples.

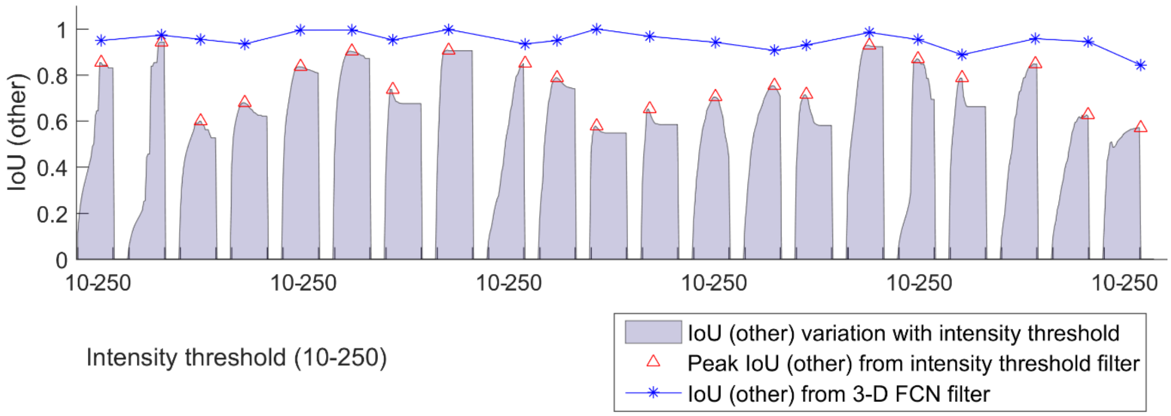


Figure 3.9 IoU (other) of all the 21 training and testing samples from two filters. The area plot in lavender grey shows IoU (other) variation from an intensity filter based on 'hard' thresholds (10–250), with peak IoU (other) in red triangle. The blue starred line shows IoU (other) from a 3D FCN filter.

After the training process, the 3D FCN model can be directly applied to filter stem and branch points from entire plots, based on a moving block of 128 × 128 × 128 voxels. Our plot scan has an average file size of about 1 GB. Filtering an aspen plot of about 500 trees requires 3.12 hrs, a maple plot of 1000 trees 1.95 hrs, and a pine plot of 2000 trees 2.23 hrs.

3.3.3 Wood Reconstruction

Individual wood reconstruction results are illustrated in Figure 3.10 with 'aspen 7' provided as an example. Branch and stem curves are reconstructed separately from FCN-filtered branch and stem points, and branch curves are finally connected to the stem curve. Shown in Figure 3.10, our node connection can overcome a certain degree of stem occlusion in the crown area, and the slice-based circle fitting algorithm is tolerant to some branch noise in the top of the canopy. The extracted stem volume is 0.45 m^3, and the branch volume 0.22 m^3. Replacing FCN prediction with reference labels and using the same QSM settings produces a stem volume of 0.38 m^3, and a branch volume of 0.23 m^3. Based on a wood density of 0.35 g cm^{-3} (Chave et al., 2009), the resulting stem biomass and branch biomass from FCN-QSM are 159 kg and 77 kg, respectively, and from reference-QSM are 133 kg and 80 kg, respectively. The ground truth of biomass is not available in this study. A referable biomass can be allometric biomass (Lambert et al., 2005) based on inventory information of species, sites, DBH, and height. The allometric stem biomass (including bark) is 149 kg and allometric branch biomass 26 kg. In this case, the branch biomass from QSM is probably overestimated. The causes can be noise and ghost points (as indicated in Figure 3.10) occurring during scanning collection, reference preparation and voxelization. However, considering all 21 sample areas, the mean branch biomass from FCN-QSM is only 15% of allometric branch biomass, due to high degree of branch occlusion in most sample areas. For branch biomass, the relative root-mean-square-error (RMSE%) and r^2 between FCN-QSM and allometric approach are 130% and 0.05, respectively, compared to the RMSE% of 19.6% and r^2 of 0.95 for stem biomass.

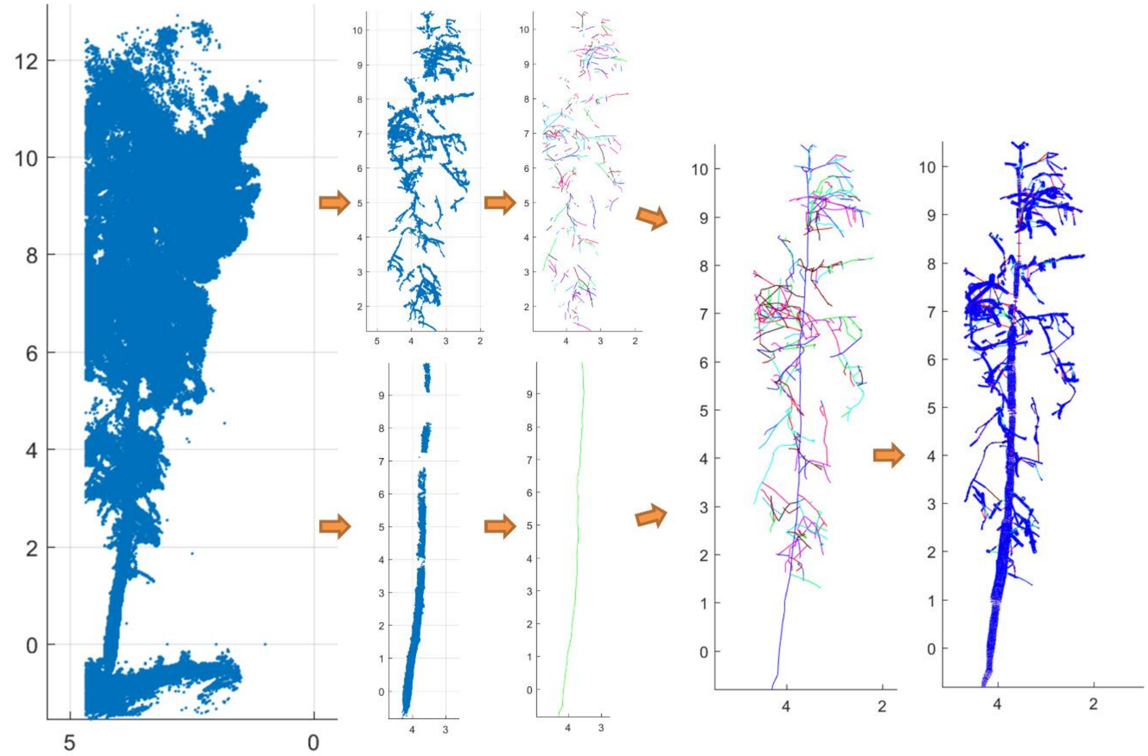
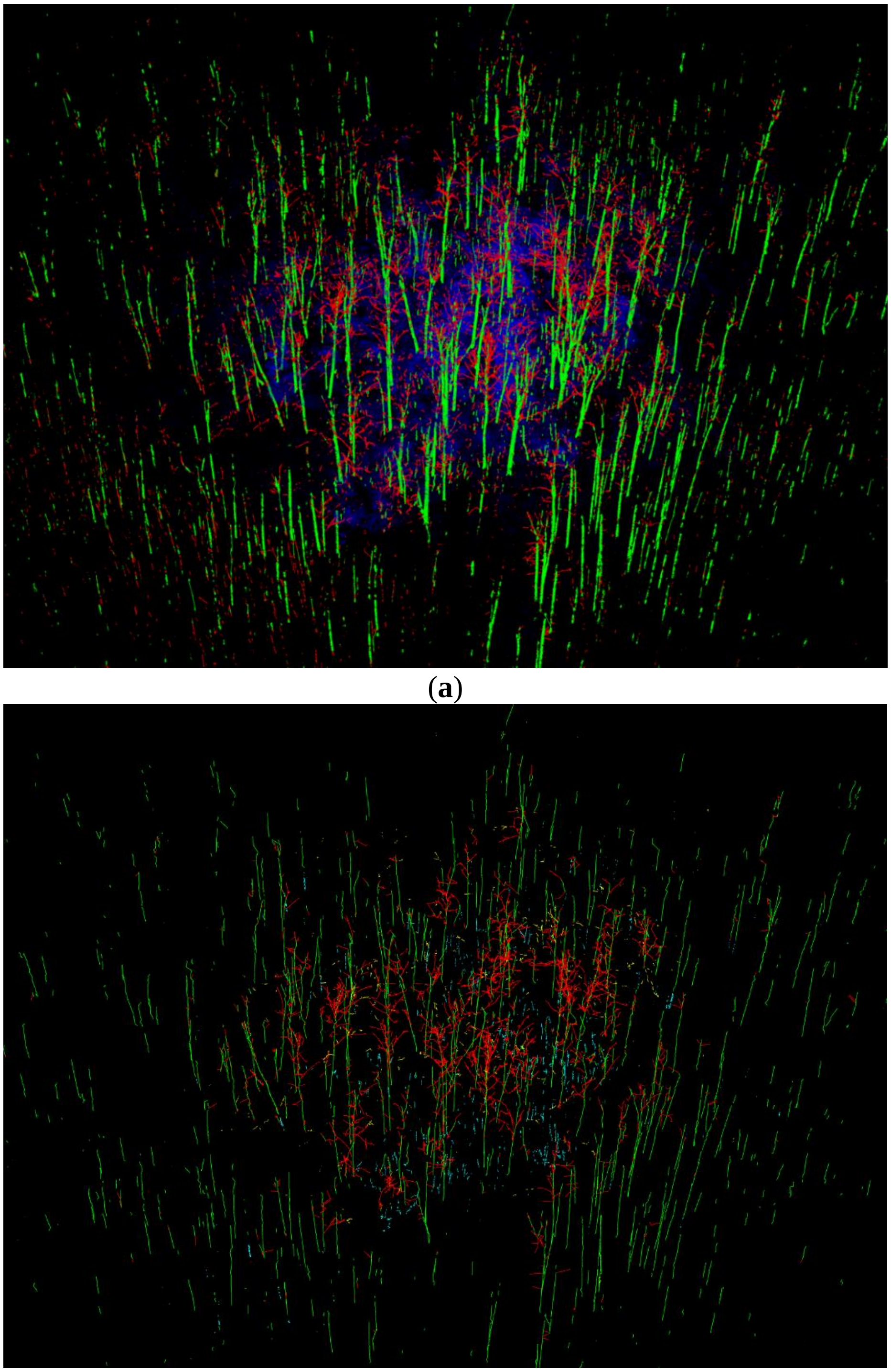


Figure 3.10 Illustration of individual-tree QSM process for ‘aspen 7’. From left to right are raw points, stem and branch points filtered from the 3D FCN model, stem and branch curve in random colors, tree curve connection in random colors, and tree curve overlaid with cross-section circles.

An example of plot-level wood reconstruction results is visualized in Figure 3.11. It is the maple plot, with FCN-filtered components (stem, branch and other) in different colours in Figure 3.11 (a). Its wood reconstructed stem, branch, isolated, and understory are also assigned with different colours in Figure 3.11 (b). The plot scan comprises five corner scans and one center scan, stretching over 100 m distance from plot center. In our maple plot, the farthest reconstructed tree is 148 m away from the plot center. However, a problem is also apparent in that reconstructing farther woods is strongly affected by a lower signal-noise ratio. Availability of both stem and branch points descends rapidly with the distance from the plot center. The reconstructed maple plot has 80% of stem volumes within a distance of 44 m, and 80% of branch volumes within 26 m. The individual tree branch-to-stem volume ratio also descends, from 0.15 within 10 m, to 0.13 within 25 m, to 0.05 within the entire plot. To mitigate the weak wood signal problem, we consider 25 m as an effective plot size for evaluating plot-level wood reconstruction. The reconstructed wood volume of maple plot has a composition of 92.3% stem, 6.5% branch, 0.3% isolated, and 0.9% understory. Correspondingly, the reconstructed aspen plot has 81.2% stem, 17.6% branch, 0.05% isolated, and 1.2% understory, and the pine plot has 91.5% stem, 8.4% branch, 0.03% isolated, and 0.1% understory. The average branch-to-stem ratio of individual tree with branch detected is 0.15 for maple, 0.24 for aspen, and 0.13 for pine. According to Ung and colleagues' biomass database (Ung et al., 2017), the average branch-to-stem ratio and its standard deviation should be 0.27 ± 0.24 for sugar maple, 0.14 ± 0.14 for trembling aspen, and 0.15 ± 0.12 for lodgepole pine. Compared to maple and aspen in leaf-on conditions, lodgepole pine has better visibility of wood points, leading to less bias of branch-to-stem ratio. A future study should involve rigid evaluation and calibration of occlusion effect on QSM's volume and branch-to-stem ratio calculation.



(**a**)

(**b**)

Figure 3.11 Example wood reconstruction from a maple plot (1 ha): (**a**) a 3D FCN filtered plot scan, with stem in green, branch in red, and other points in blue; (**b**) QSM reconstructed wood curves, with stem in green, branch in red, understory in cyan, and isolated wood in yellow. To balance the level of graphic details, blue points in (**a**) are assigned with transparency degree of 0.99, and stem curves are set four times thicker than all other curves.

Finally, it should be clarified that reconstruction results were not validated, due to lack of ground truth measurement of branch curve and volume. The parameters of reconstruction model are also not optimized, for the same reason of lacking accurate reference. The problem of lacking validation and reference should be mitigated in the future by using repeated scans of leaf-on and leaf-off deciduous trees or virtual forest simulation models. Yet one function of this QSM model can be exposing detailed wood components, and reflecting any issues of integrating filtering and QSM for natural forest scans. The delineated tree curves can also provide high-level geometric features, in terms of raw point metrics, to help develop more accurate allometric models of wood volume or biomass.

3.4 Conclusions

Delineating wood components in complex natural forest environments has attained wide research interest, represented by QSM development, due to its importance to modern wood management in pursuit of correct and complete biomass, carbon budget, and tree physiology. At the plot level, the wood amount in dense and noisy areas such as crown and understory are however, uncharted or gross in common QSM models. Recent QSMs incorporate simple wood filtering methods aimed to reduce noise. In contrast, this study introduces the use of a deep learning model FCN in 3D space to filter both stem and branch points from complex forest scans, with an average testing accuracy of 0.89 (stem IoU), 0.54 (branch IoU), 0.79 (mIoU), and 0.94 (OA) over three plots. We further explored the potential application of wood filtering in constructing a plot-level QSM and extracting the wood volume component. From visual inspection, wood filtering generally produces tube shapes beneficial to QSM modeling.

Some challenges remain, however. Creating reference data, although assisted with a semi-automation algorithm, inevitably contains interpretation errors and further induces low branch filtering accuracy. Besides, point cloud is a discrete form of 3D data, but continuous voxel form is the only supported choice in deep learning tools. As a result, coarse voxel resolution was fixed to tradeoff deep learning accuracy with sufficient computation efficiency. Last but not least, wood filtering cannot address the most severe problem of wood delineation, which is occlusion. Improving TLS scan coverage and QSM modeling, although sophisticated, are still the most important solutions to overcome occlusion.

Deep learning modeling offers significant potential to intellectualize processing and analysis in various data-driven applications. This study only intends to probe the feasibility of filtering 3D wood points of interest, without activating diverse options of refinement. Filtering low-quality scans, particularly from needle-leaf species (e.g., pine and spruce), remains a challenge. Tentative refinement may introduce more complex network structures such as ResNet (He et al., 2016), Inception (Szegedy et al., 2017; Szegedy et al., 2015), and CRF (Zheng et al., 2015), sparse point network such as PointNet (Qi et al., 2016; Qi et al., 2017), instance segmentation network such as R-CNN (Girshick et al., 2014; Ren et al., 2015). However, refinement of the all-purpose deep networks are not optimal solutions to many practical 3D problems. It is strongly hoped that deep network mechanisms can be rationalized systematically in the future, in a similar manner to Vapnik–Chervonenkis (VC) learning theory (Vapnik, 1995) explaining the statistical nature of machine learning. For example, it is important to avoid computation overflow by previewing the complexity or dimension of 3D data structure before finding an optimal deep learning model with matching complexity. It is also critical to find the relationship between sample size, training time, and generalizability for deep learning. There also need many research efforts to convert the enormous numeric features created by deep learning into analytic and explanatory features, before deep learning could be involved to help develop or calibrate existing models with mathematic forms.

Chapter 4. Classifying Plot Species from Wood Point Clouds

4.1 Context

Biomass is species-specific, considering that both wood density and expansion factors vary with species (Chave et al., 2009; Teobaldelli et al., 2009). Commercial TLS typically can detect targets up to and over hundreds of meters. There can be many species represented per plot. With TLS's strong ability to sense tree 3D structural detail, species classification from TLS plot scans is a justifiable research goal. Sampling within a mixed plot with TLS can also provide detailed species mixture information useful for broader scale studies.

Relevant studies are rare due to the shortage of multispecies scans and the complexity of dense scan processing. A pilot study from Othmani et al. (2013) employed the 3D geometric texture of bark to classify hornbeam, oak, spruce, beech and pine from TLS scans with an overall accuracy of 88%. This method is limited to several close-range trees and lacks universality. More reliable classification should take account of rich structural information from species. As Falster and Westoby (2003) point out, the variation of most architectural traits between species is systematically greater than the variation within a species. This is apparent because vegetative morphology is widely used as basic taxonomic knowledge to identify species. Lin and Herold (2016) are among the first to explore TLS's explicit tree structure (ETS) features ranging from tree, stem, branch, crown to leaf level. A support vector machine (SVM) classifier is applied to differentiate four species among 40 trees. After performing cross-validation, their best overall accuracy is 90.0% and they suggest 77.5% is a more generalizable accuracy. Fully automatic species classification based on QSM branch features and machine learning classifiers was first demonstrated in Åkerblom et al. (2017). Their classification accuracy is above 93% among three species: silver birch, Scots pine and Norway spruce. Due to insufficient reference studies, it is still questionable whether the selected branch features can remain effective to a wider choice of species. Another constraint of Åkerblom et al. (2017)'s method is the dependency on complex computation processes, including QSM reconstruction and branch-level feature extraction. A more flexible and universal classification method needs to be investigated with an ultimate goal of replacing manual interpretation for the NFI's tree species lists.

As investigated in Chapter 3 and evaluated in many studies, the deep learning model the convolutional neural network (CNN), is considered a feature-free replacement of machine learning classifiers, since feature selection or feature engineering can be automated using CNN. The CNN typically comprises a long sequence of layers, with one layer a function of its previous layer. The long-sequence structure is vividly described as 'deep'. Recent progress further enables model optimization directly from sparse point-cloud format. Calculation of 3D convolution in deep layers can be simplified and convergence of classification accuracy can be expedited, whereas the best accuracy is competitive with the CNN-based models. At present, representative point-based CNN models are PointNet (Qi et al., 2016), Kd-Net (Klokov & Lempitsky, 2017) and PointCNN (Li et al., 2018c), whereas other versions such as OctNet (Li et al., 2018a) and So-Net (Riegler et al., 2017) adopt analogous schemes. Basically, the PointNet assumes arbitrary 3D math functions and transformations including the 3D convolution can be implicitly assembled by deep 2D layers under hyper feature space. Through the deep layers, signals (or point features) can be enhanced and filtered until features become easily classifiable. The authors further overcome the sparsity issue caused by enormous parameters by penalizing model complexity with a regularization cost function. Indeed, the PointNet totally ignores local spatial connectivity and the equivalence between its 2D layer setup and 3D is built upon a strong assumption of spatial homogeneity. The PointNet model was later developed to be the PointNet++ (Qi et al., 2017) with additional point grouping attributes to utilize local contextual information. The Kd-Net, as its name suggests, decomposes point clouds into an organized Kd-tree structure (Bentley, 1975) and optimizes a small amount of affine transformation parameters at each tree level. An apparent advantage of Kd-Net is its short training time. The drawback is also notable that Kd-Net may fail where 3D geometries are improperly presented in a Kd-tree structure, such as what is occurring in a ‘real’ tree scan. The PointCNN provides a more flexible solution than Kd-Net without a strong presumption of data structure. It basically replaces the 3D convolution layer with a sparse version called *X* convolution which assigns learnable parameters to local k nearest neighbors. Their final network structure simply includes two *X* convolution layers and a conventional fully-connected layer. All these point-based deep learning models are reasonable in theory, yet lacking practical assessment about classifying species from complex tree scans, since these models are customized for indoor datasets such as ModelNet40 (Wu et al., 2015) and S3DIS (Armeni et al., 2017) featuring small objects, smooth surfaces, explicit boundaries and numerous classes.

This chapter will investigate whether the state-of-the-art deep learning models can be tailored to species classification scenario based on branch information and what are their accuracies. A number of wood segmentation algorithms based on machine learning are also involved in contrast against the 3D FCN model in Chapter 3, to exhibit more possibilities in wood biomass analysis.

4.2 Methodology

4.2.1 Material and Framework

We selected seven monospecific natural plot scans (about 30 × 30 m^2) during the summers of 2014, 2015 and 2016, corresponding to seven species: sugar maple (*Acer saccharum*) (44.077 °N,79.323 °W), trembling aspen (*Populus tremuloides*) (49.349 °N,114.411 °W), lodgepole pine (*Pinus contorta*) (49.646 °N,114.095 °W), red pine (*Pinus resinosa*) (44.077 °N,79.323 °W), Scots pine (*Pinus sylvestris*) (61.205 °N, 25.069 °E), Norway spruce (*Picea abies*) (61.205 °N, 25.069 °E) and silver birch (*Betula pendula*) (61.205 °N, 25.069 °E). The first four species were scanned in Canada with an Optech ILRIS HD (1535 nm) and the remaining three species were scanned in Finland by the FGI with a Leica HDS6100 (650–690 nm). All the plot scans consisted of one center scan, and 3-5 corner scans. The average alignment accuracy of the first four Canadian plot scans was 3 cm and the alignment accuracy of the FGI scans were 2.1 mm. The accuracy difference was caused by different scan settings. An FGI plot scan encompasses only five single scans, each with a wide field of view of 360 ° × 310 °, and the accuracy was calculated based on *in-situ* targets (Liang et al., 2018). In comparison, a Canadian plot scan was composed of more than 20 individual scans due to limited field of view, and the accuracy was based on the Iterative closest point (ICP) alignment algorithm. All the FGI and Canadian plot scans were trimmed to a fixed circular radius of 20 × 20 m^2 and uniformly resampled with a point resolution of 2 cm using the CloudCompare for further analysis, with stem density and average and standard deviation of tree height shown in Table 4.1. It is clear that stem density and height vary among different plots, but each plot is homogeneous considering the within-plot tree height has small variation. It is also apparent that tree height alone cannot differentiate species, since the same species such as sugar maple, red pine and Norway spruce have similar tree heights.

Table 4.1 Characteristics of the seven monospecific plots for species classification.

	Tree totals	Tree density (ha^{-1})	Tree height average (standard deviation) (m)
sugar maple	54	430	24.77 (1.42)
trembling aspen	80	637	13.96 (0.69)
lodgepole pine	130	1035	20.96 (1.02)
red pine	85	676	24.05 (0.87)
Scots pine	69	549	20.38 (1.32)
Norway spruce	63	501	24.18 (2.63)
silver birch	101	804	17.25 (0.82)

The processing framework included wood point segmentation, species classification and classification accuracy evaluation. We investigated multiple machine learning algorithms to filter wood points, decompose plots into separate trees including stems and branches, and evaluate five deep learning classifiers including VGGNet, PointNet, PointNet++, Kd-Net, and its variation contextual Kd-Net (Zeng & Gevers, 2018).

4.2.2 Wood Segmentation based on Point-wise Machine Learning

It is possible to classify species directly using deep learning model without wood point classification. Yet a deep learning model using entire crown is not generalizable to the same deciduous species in a leaf-off condition. It is therefore necessary to filter wood points before species classification. The design of wood segmentation classification is outlined in Figure 4.1. A total of 45 features were extracted for each point using the software CloudCompare (v2.11), with definitions listed in Table 4.2. Except for the three features, namely, Z, intensity and bilateral filtered intensity (Tomasi & Manduchi, 1998), the other 42 features are local geometrical features summarized from three search windows

(5, 10, and 20 cm in radius, respectively) centered on each point. The configuration of three search windows was intended to cover three typical scales corresponding to twigs, branch and stems. Figure 4.1 lists the three features and 14 features per scale. Based on the 45 per-point features, six supervised machine learning classifiers including K-Nearest Neighbors (KNN), Naive Bayes (NB), Adaboost (AB), Random Forest (RF), SVM and Discriminant Analysis (DA) were trained in the MATLAB (r2018b) to classify three classes: stem, branch and the other. Parameter K of the KNN was set to 20, the number of bagged trees for the RF were set to 200. Settings of other classifiers remained default values in MATLAB. In addition, point cloud features were also normalized for the KNN and SVM classifiers. Our purpose was not to maximize each classifier's performance for wood classification, since the following species classification based on deep learning does not rely on perfect wood filtering results. Our point instead was to seek a convenient and practical way of reducing crown noise and exposing branch geometries. A small portion of leaf or twig points might not be classified correctly, but in general, it did not affect the species classification. The mathematic and statistical rationale of the machine learning classifiers are explained in Bishop (2006) and Kuhn and Johnson (2013). Accuracy metrics used to evaluate these methods were Overall Accuracy (OA), Intersection over Union (IoU), precision and recall with manual wood classification as reference.

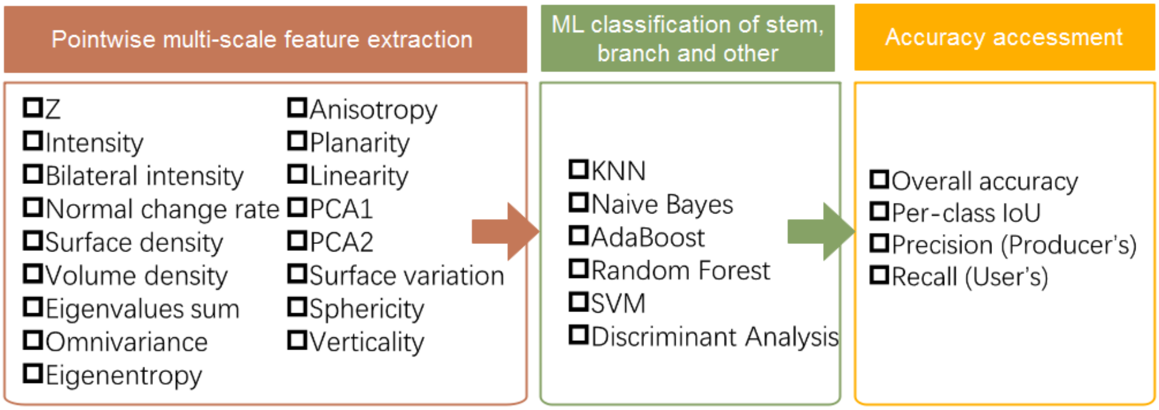


Figure 4.1 Feature-based wood point classification and evaluation.

Table 4.2 Definitions of geometric features from CloudCompare. $\boldsymbol{L_1}$, $\boldsymbol{L_2}$ and $\boldsymbol{L_3}$ denotes the first, second, and third eigenvalues calculated from neighborhood points, respectively.

Feature	Definition
Z	Coordinate along vertical axis
Intensity	Intensity of the laser return (0-255)
Bilateral intensity	Intensity enhanced with a bilateral filter
Normal change rate	Point orientation change rate
Surface density	Number of neighbors / neighborhood area
Volume density	Number of neighbors / neighborhood volume
Eigenvalues sum	$L_1 + L_2 + L_3$
Omnivariance	$\sqrt[3]{L_1 L_2 L_3}$
Eigenentropy	$-(L_1 \ln L_1 + L_2 \ln L_2 + L_3 \ln L_3)$
Anisotropy	$(L_1 - L_3)/L_1$
Planarity	$(L_2 - L_3)/L_1$
Linearity	$(L_1 - L_2)/L_1$
PCA1	$L_1/(L_1 + L_2 + L_3)$
PCA2	$L_2/(L_1 + L_2 + L_3)$
Surface variation	$L_3/(L_1 + L_2 + L_3)$
Sphericity	L_3/L_1
Verticality	1- |orientation projection along vertical axis|

Point clouds of five trembling aspens, five sugar maples, one birch and one lodgepole pine were selected and clipped from the corresponding plot scans in CloudCompare. The stem, branch and other component points were manually labelled with class 1, 2 and 3 to be our reference sample. Note that the reference dataset was slightly different from that in Chapter 3. Chapter 3 aimed to use arbitrary plot clips to evaluate deep learning performance under complex situations, but the primary purpose of wood filtering in this chapter was shifted to species classification for individual trees. Trees with great occlusion of branch patterns could lead to great uncertainty of species classification, and were not included as reference data. Among the 12 tree clips, four aspen, four maple, and one lodgepole pine clips (nine in total) were used as our training sample and the remaining one aspen, one maple and one birch clips (three in total) were regarded as testing sample for accuracy assessment purposes. The other three species or larger sample size were not included because wood segmentation here does not need to be perfect, and wood segmentation is considered species-independent based on the finding in Chapter 3. Indeed, the sample size (>100,000 features per point cloud) was considered sufficient for the point-based machine learning classifiers, and no improvement of classification accuracy was found with more species or point clouds.

Entire plot scans with each point containing 45 features (a.k.a. predictors) could give rise to tremendous data volume hazardous to storage and computation. Features with marginal importance to classification accuracy should be discarded. Among various feature importance criteria and feature selection methods, random forest was a common choice which characterizes predictors' influence by the change degree of out-of-bag errors gathered from repeated permutation tests (Loh, 2002). Influential features were selected for overall consideration and were calculated for each point of the plot scans. The random forest classifier was then trained again using the selected features and all the reference samples, to predict wood component classes for the entire seven plot scans. The stem and branch points were extracted based on the predicted classes, and then merged into individual tree point clouds using the method in Chapter 3. Each of the wood point clouds were inspected manually, and the highly occluded trees without few branch points were not used for the following species classification process, since our objective is to inspect if a deep learning classifier can classify species based on branch points.

4.2.3 Species Classification using Point-based Deep Learning Models

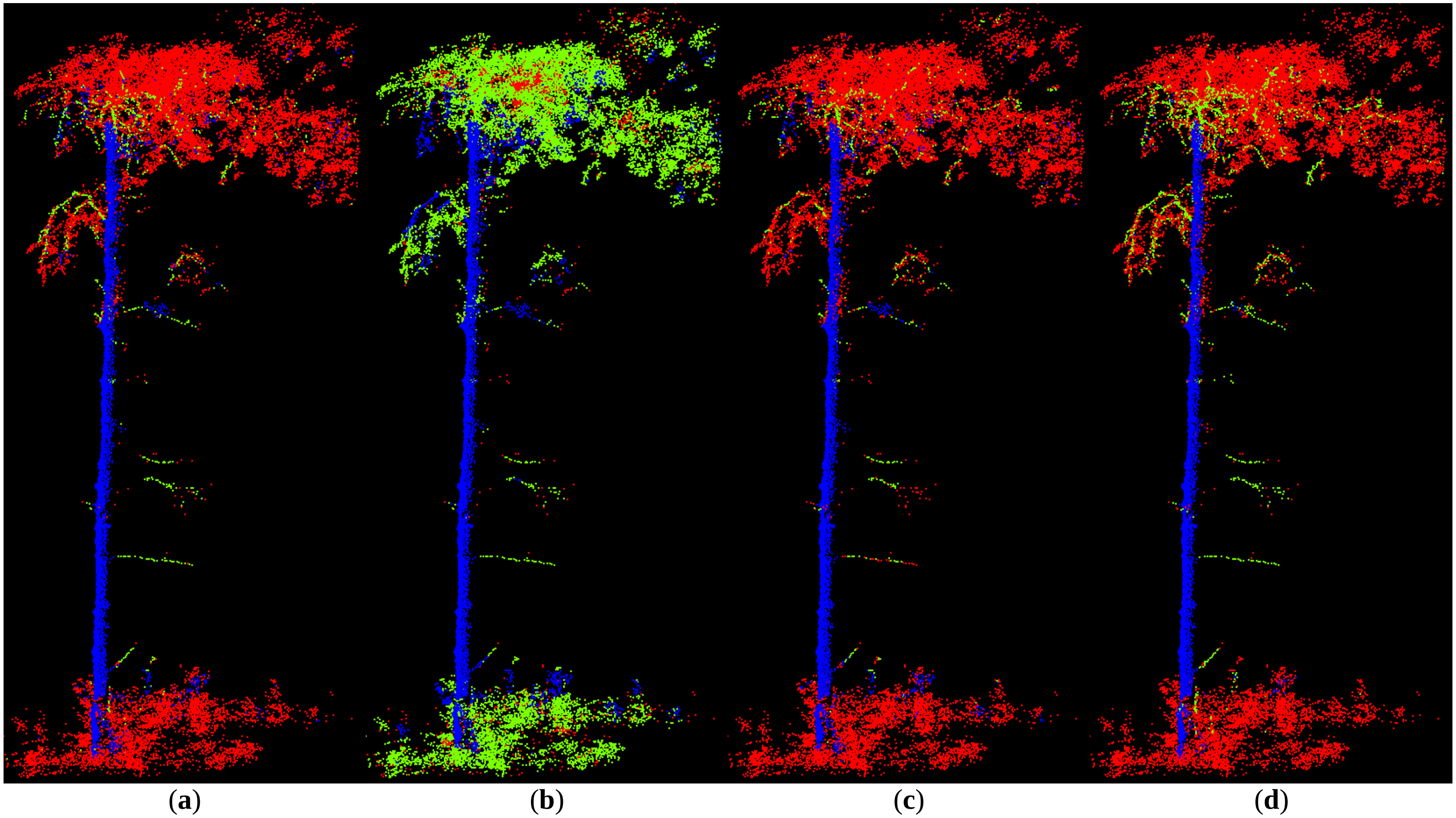
Only the 3D coordinates of wood points (stem and branch as a whole) are needed for species classification using point-based deep learning models. More attributes such as laser intensity or height can be useful information for species classification. Yet intensity and height are not always available from any arbitrary laser scanning datasets, and it is also one of our objectives is to justify the feasibility of feature-free deep learning model against the feature-based machine learning models such as Lin and Herold (2016). That is why only 3D coordinates of individual trees are chosen to be the model input. After manual inspection and removal of occluded trees, a total number of 313 trees from the seven species were selected for species classification, where 80% of the total, namely, 250 trees were randomly picked for the training purpose and the remaining 63 trees for the testing purpose. Five point-based deep learning classifiers VGGNet, PointNet, PointNet++, Kd-Net and contextual Kd-Net were investigated briefly in terms of classification accuracy. All five classifiers only allowed a fixed choice of data size to match their layer templates. In our case, the number of points from each individual tree ranged between 5,000-40,000. The final point number for the deep learning models was 2048 after a balanced consideration of training speed and accuracy based on some exploratory tests. The huge loss of points could mean a great loss of information for the deep learning classification. The question is therefore how to choose a sampling strategy that best preserves important points from each tree point clouds. The sampling method adopted in this study was based on k-means algorithm, which initially assigned 2048 seeds within the point clouds and iteratively shifted each seed to the centroid of clusters around the seed. The results of the k-means algorithm were meaningful clusters and the centroids of clusters were the input to the deep learning model.

Deep learning models did not need heavy feature engineering or parameter tuning, since most parameters were turned during training process and other constant parameters had a marginal impact on classification accuracy. The main constant parameters for all five deep learning models were dropout ratio, batch size, and optimizer. The dropout ratio, designed to penalize overfitting risks, was fixed to be 0.5 for each dropout layer. The batch size, used to expedite learning process in parallel, was set to 32. The optimizer, controlling and scaling the search step size for minimal training error, was configured to be SMORMS3 (Wichrowska et al., 2017) based on gradient noise. In addition, data augmentation functions, such as translating, rotating and mirroring the input point clouds during the training process, were also enabled on to enhance the robustness of deep learning. The total training iterations were set to be 400. At the end of each training iteration, the training and testing accuracies were tracked based on 20 training and 20 testing samples, as a coarse and rapid way of assessing accuracy convergence. Due to the tree data complexity, training accuracy would fluctuate throughout the training process and even after convergence. Our strategy was to store the classifier status with the minimal testing loss during the entire training process. After training was finished, the best classifier status was then recovered to classify testing datasets for evaluation purposes. The accuracy metrics for evaluation were OA and per-class IoU.

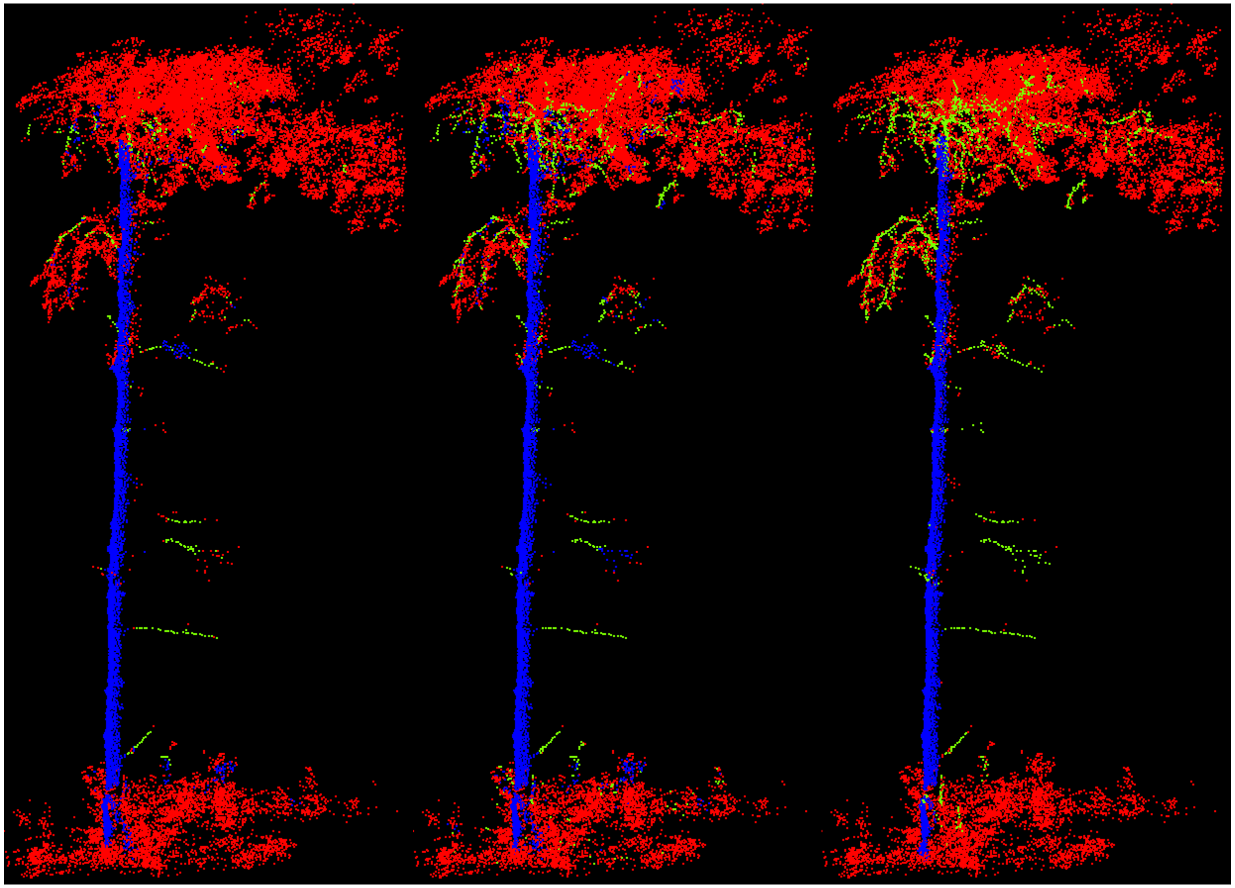
4.3 Results and Discussion

4.3.1 Wood Segmentation

The six classifiers, namely, KNN, NB, AB, RF, SVM and DA, were trained from nine training samples with 45 per-point features. For illustrative purposes, we choose one of the aspen training samples as shown in Figure 4.2, to assess visually their classification differences. Among the six machine learning classifiers, NB is the weakest, seen by its exaggerated classification of both stem and branch points in the ground and crown regions. The RF delivers the visually correct stem and branch classification appearance. The other four classifiers can identify the main stem without difficulties, yet tend to misclassify a number of crown and ground points as stem points. This is because the certain parts of stem, crown, and ground have similarities in size, shape, and surface curvature. The crown and ground are also the most erroneous areas for the four classifiers where many branch points are omitted. Comparably, KNN and DA detects relatively more branches than SVM and AB in this aspen case. The effects of stem overestimation and branch underestimation imply a weakness to differentiate irregular and fine data structure. It further means the complexity of the 3D wood classification problem exceeds the dimensionality of many machine learning models, with RF being an exception, owing to its tree number setting which adds model dimensionality. Another obvious problem is the salt and pepper effect among all the six classifiers. This is a typical problem for pointwise classification, which can be mitigated by increasing the unit of classification or adding a smoothing filter.



(a) (b) (c) (d)



(e) (f) (g)

Figure 4.2 Wood classification of an aspen sample using the classifier of (**a**) KNN: K-Nearest Neighbors, (**b**) NB: Naive Bayes, (**c**) AB: Adaboost, (**d**) RF: Random Forest, (**e**) SVM: Support Vector Machine, (**f**) DA: Discriminant Analysis, and benchmarked by reference (**g**). The stem, branch and other components are shown in blue, green and red, respectively.

Quantitative accuracy assessment of wood segmentation using the entire training and testing samples is shown in Figure 4.3. Note that precision measures the agreement within the reference sample and recall measures agreement within prediction. Among the training accuracy results (bars in Figure 4.3), the NB has the lowest accuracies on the whole except for the branch precision where KNN performs worst. The RF has much higher accuracies than the other five classifiers. The five classifiers except NB have high OAs above 0.85, owing to successful stem classification with stem IoUs of around 0.80. The testing accuracies (crosses in Figure 4.3) are systematically lower than the training accuracies with a few exceptions from SVM and NB. Among the six classifiers, the RF contributes to the greatest difference between training and testing accuracies. Its branch IOU, for example, drops drastically from 0.70 to 0.36, indicating a strong overfitting effect. Causes of the universal overfitting effect are manifold, generally related to the sample size limitation, reference imperfection, and lack of global features. Unlike training accuracies, testing accuracies do not vary greatly among different classifiers, excluding the extreme bad results from the NB. For example, the stem IoU of the remaining five classifiers keeps tightly to the level of 0.8, and the difference of branch IoU is no greater than 0.2. The SVM, KNN, RF and DA classifiers, in particular, have comparable predictive power of both stem and branches, seen from the testing accuracies. The RF has only slightly higher testing accuracies than the other three except for the stem precision. The results of classifier similarity and the RF’s small advantage in accuracy are in line with the findings from Wang et al. (2017). For both training and testing accuracies, and among all classifiers except NB, stem precisions are systematically higher than stem recalls, and branch precisions lower than branch recalls. This coincides with the visual interpretation in Figure 4.2 that stem classification tends to be aggressive and branch more conservative.

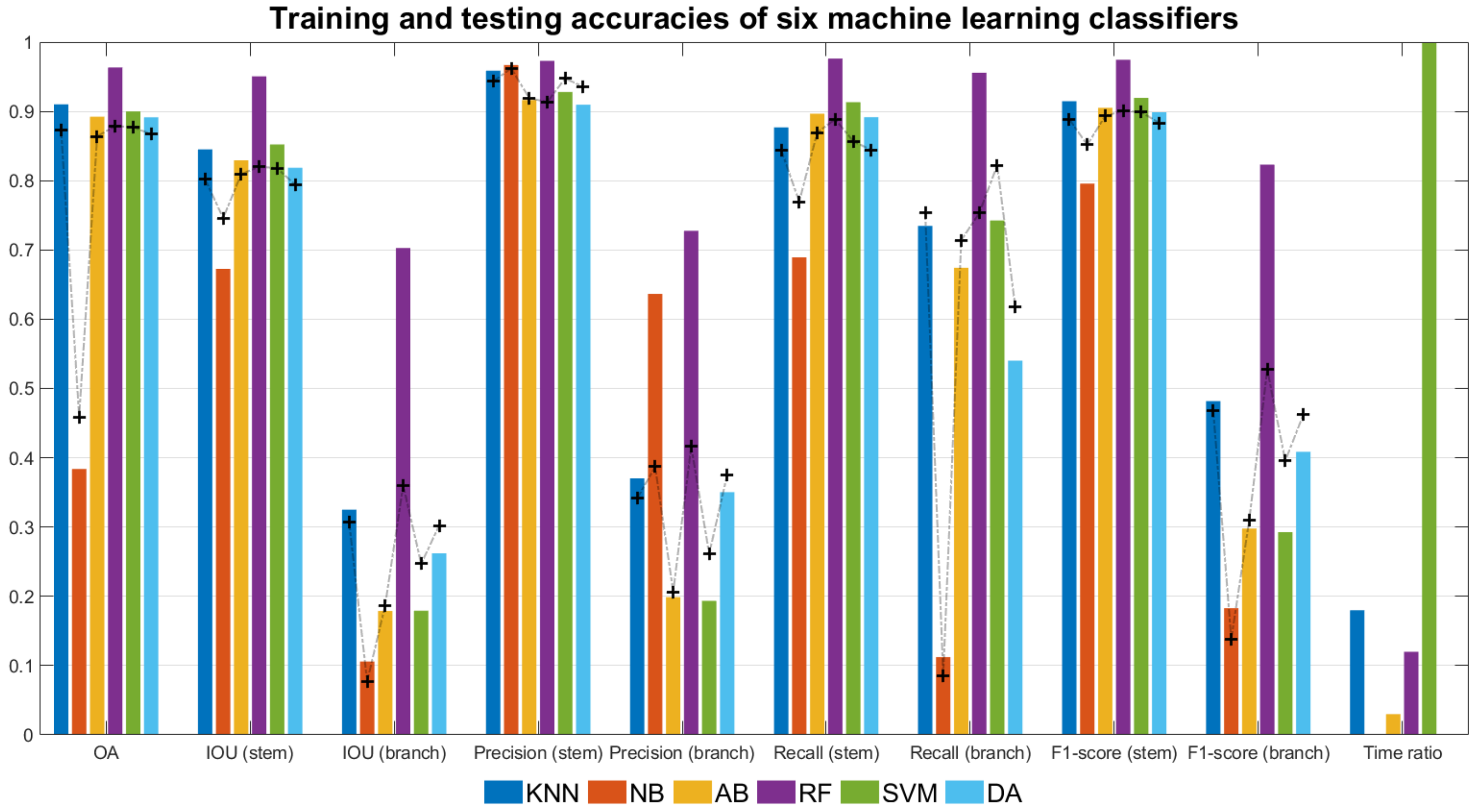


Figure 4.3 Accuracies of the six classifiers trained with 45 per-point features. Each training accuracy is an average value from nine training samples shown as a colored bar, and each testing accuracy is an average from three testing samples in dotted lines. The horizontal axis lists the names of nine accuracy metrics, as well as a time ratio metric near the right edge. The time ratio measures the total training and testing time scaled to a range between 0-1.

Specific classification accuracies for each sample is summarized in Table 4.3. Among the three species and averaging from the training samples, the sugar maple has the highest OA (0.98), stem IoU (0.97) and branch IoU (0.66), the lodgepole pine has the same highest stem IoU (0.97), whereas the trembling aspen has overall lowest accuracies. Among the three testing species, the sugar maple has the highest stem IoU (0.86) but lowest branch IoU (0.29) probably due to the confusion of stem and big branches. The trembling aspen's OA (0.83) and stem IoU (0.76) are still the lowest among the three species, due to its wood complexity.

Table 4.3 Classification accuracies of each sample based on the random forest model using 45 features, with TA short for trembling aspen, LP lodgepole pine, SB silver birch and SM sugar maple.

	OA	IoU (stem)	IoU (branch)	Precision (stem)	Precision (branch)	Recall (stem)	Recall (branch)
TA 1	0.97	0.94	0.76	0.99	0.83	0.95	0.91
TA 2	0.93	0.91	0.61	0.96	0.62	0.95	0.98
TA 3	0.96	0.91	0.63	0.94	0.66	0.97	0.94
TA 4	0.96	0.95	0.62	0.98	0.63	0.97	0.97
TA 5*	0.83	0.76	0.40	0.90	0.51	0.83	0.64
LP	0.94	0.97	0.67	0.98	0.68	0.99	0.97
SB*	0.89	0.84	0.40	0.92	0.42	0.91	0.90
SM 1	0.98	0.97	0.87	0.98	0.89	0.99	0.97
SM 2	0.98	0.96	0.70	0.96	0.73	0.99	0.95
SM 3	0.98	0.98	0.75	0.99	0.78	0.99	0.95
SM 4*	0.91	0.86	0.29	0.93	0.32	0.92	0.71
SM 5	0.98	0.97	0.72	0.98	0.74	0.99	0.97

* denotes testing sample

Importance of the 45 features estimated from random forest is shown in Figure 4.4. The importance measure is unitless, and a relatively higher value means greater contribution to the variation of prediction errors. In general, most features have identical importance values varying around 1.7. The greatest gap exists between the top seven influential features and the remaining. Those seven features with a descending order of importance are Z value, verticality (r = 0.1 m, 0.3 m, 0.2 m, respectively), bilateral filtered intensity, intensity and omnivariance. Similar importance analyses with different choices of training samples indicate that the Z value, the two intensity features, and verticality are consistently most important features for wood classification. In particular, the Z value of a point is interpreted as most helpful to differentiate branch, stem or other classes. No other geometrical features including omnivariance were ranked high importance. The average level of importance is alike (1.73, 1.74 and 1.66, respectively) among the three predefined scales (0.1, 0.2 and 0.3 m). This scale likelihood effect indicates that a fixed scale may not be an appropriate unit to calculate representative branch-level features, due to the morphological uncertainties from different sites and species. After combined consideration of computation speed and feature representativeness, eight features are selected, namely, Z value, intensity, sum of eigenvalues (r = 0.3 m), first two PCA components (r = 0.2 m), and three PCA derivatives (planarity, linearity and verticality with r = 0.1 m).

Classification accuracies before and after feature selection are provided in Table 4.4. Reducing feature numbers from 45 to 8 causes accuracy decreases, 0.08 (9%) for stem and 0.06 (11%) for branch on average. Yet testing accuracy drops by 0.10 (17%), greater than the training accuracy decrease of 0.04 (5%). This noticeable stability issue caused by feature selection is a known limitation of feature-based machine learning methods (Li et al., 2018b).

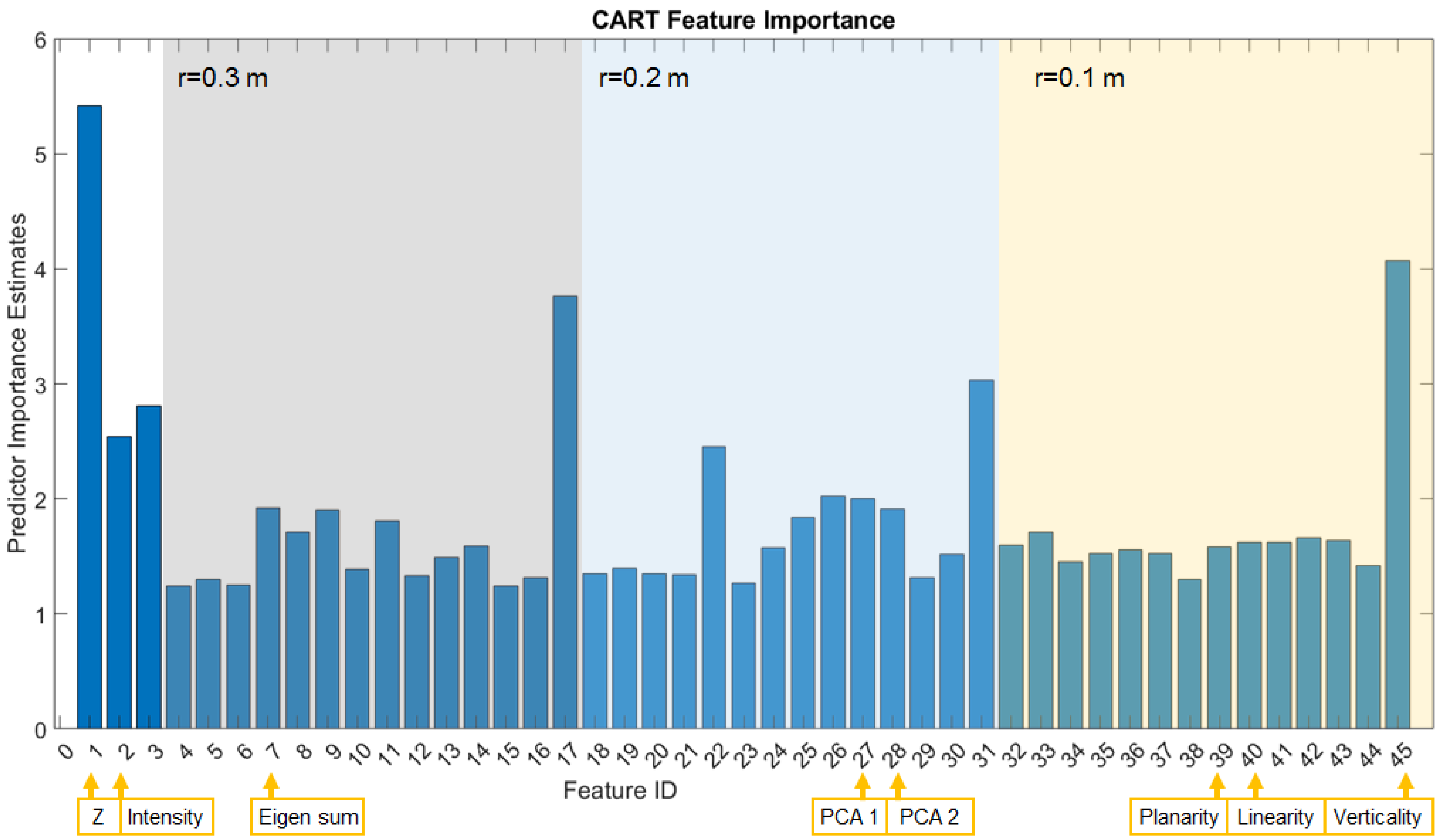
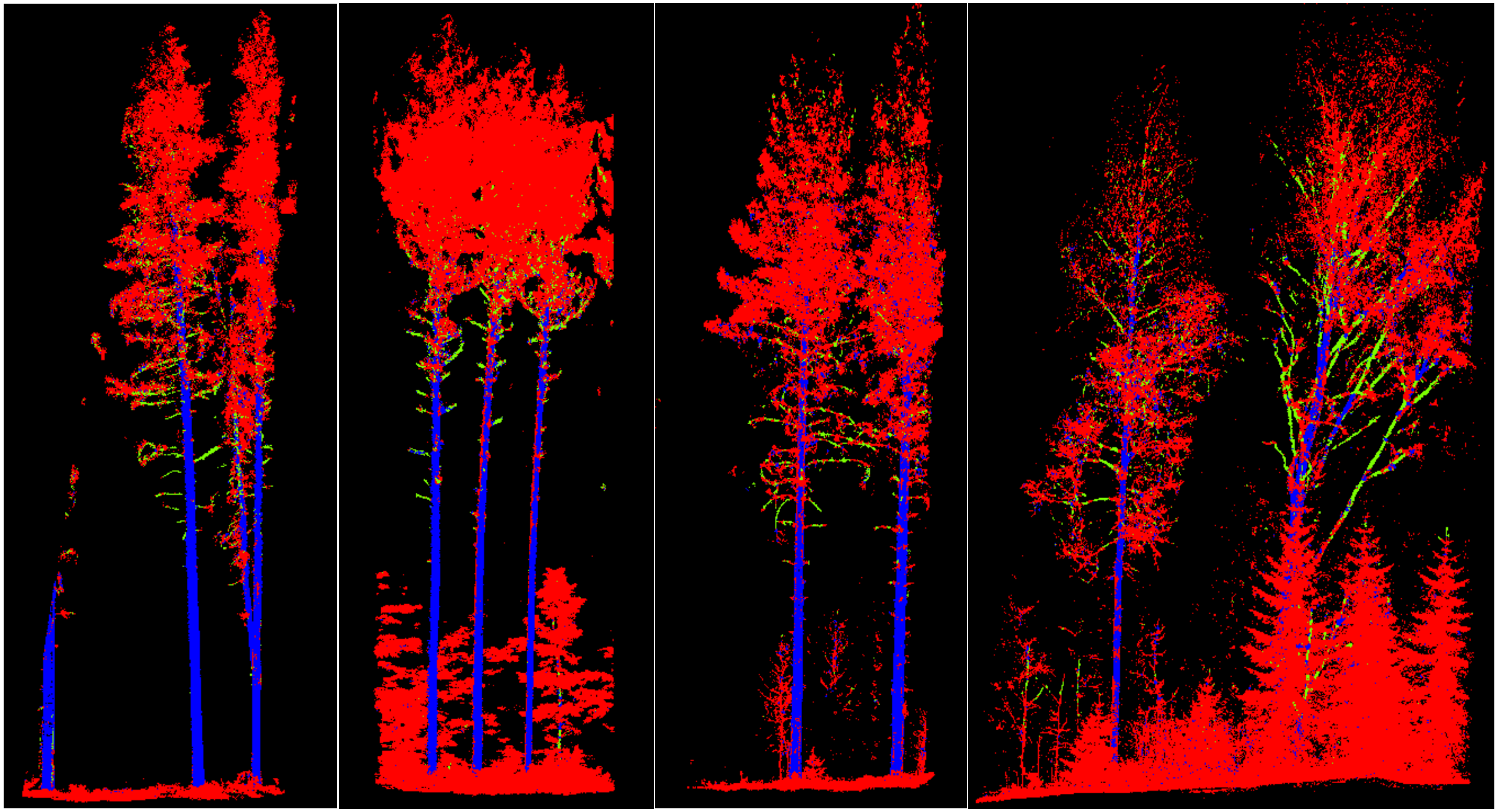


Figure 4.4 Feature importance estimation from random forest among the 45 candidate features. The **r** refers to the radius of nearest neighbor search window for feature calculation. Eight representative features (denoted in the bottom boxes) were finalized to be the predictor variables of the random forest classifier.

Table 4.4 Accuracy comparison before and after feature selection from random forest.

	OA	IoU (stem)	IoU (branch)	Precision (stem)	Precision (branch)	Recall (stem)	Recall (branch)
Training (45 features)	0.96	0.95	0.70	0.97	0.73	0.98	0.96
Testing (45 features)	0.88	0.82	0.36	0.91	0.42	0.89	0.75
Training (8 features)	0.95	0.91	0.65	0.96	0.69	0.95	0.92
Testing (8 features)	0.80	0.69	0.29	0.78	0.36	0.87	0.61

The point clouds of the seven species classified by the random forest model using 8 features are displayed in Figure 4.5 (clipped view) and Figure 4.6 (plot view, wood points only). Main stems are detected with some noise around the edge of stems (e.g. Figure 4.5f). Certain crown portions are misclassified as stems, and certain top crown portions as branches, clearly seen by the plot views in Figure 4.6. The redundancy in the crown area is not absolutely harmful in the context of species classification. Instead, additional crown information might enhance the species identity for the deep learning classifiers. Identifying the seven species at plot level is not difficult from manual interpretation, considering a variety of discernible traits including tree height, first branch height, branch density, branch angle, and stem verticality. However, identifying trees within a plot without reference can be challenged by incomplete or irregular tree forms. For example, the lodgepole pine tree in Figure 4.5a and the Scots pine in Figure 4.5c have visually similar crown shape and main branch angles. Wood classification effect of the sugar maple plot can be contrasted in Figure 4.7 between the random forest and the deep learning classifiers in Chapter 3. More branches and also more noise points were classified by the random forest, but this comparison is just coarse because different data were used to train the two classifiers.



(a)
(b)
(c)
(d)

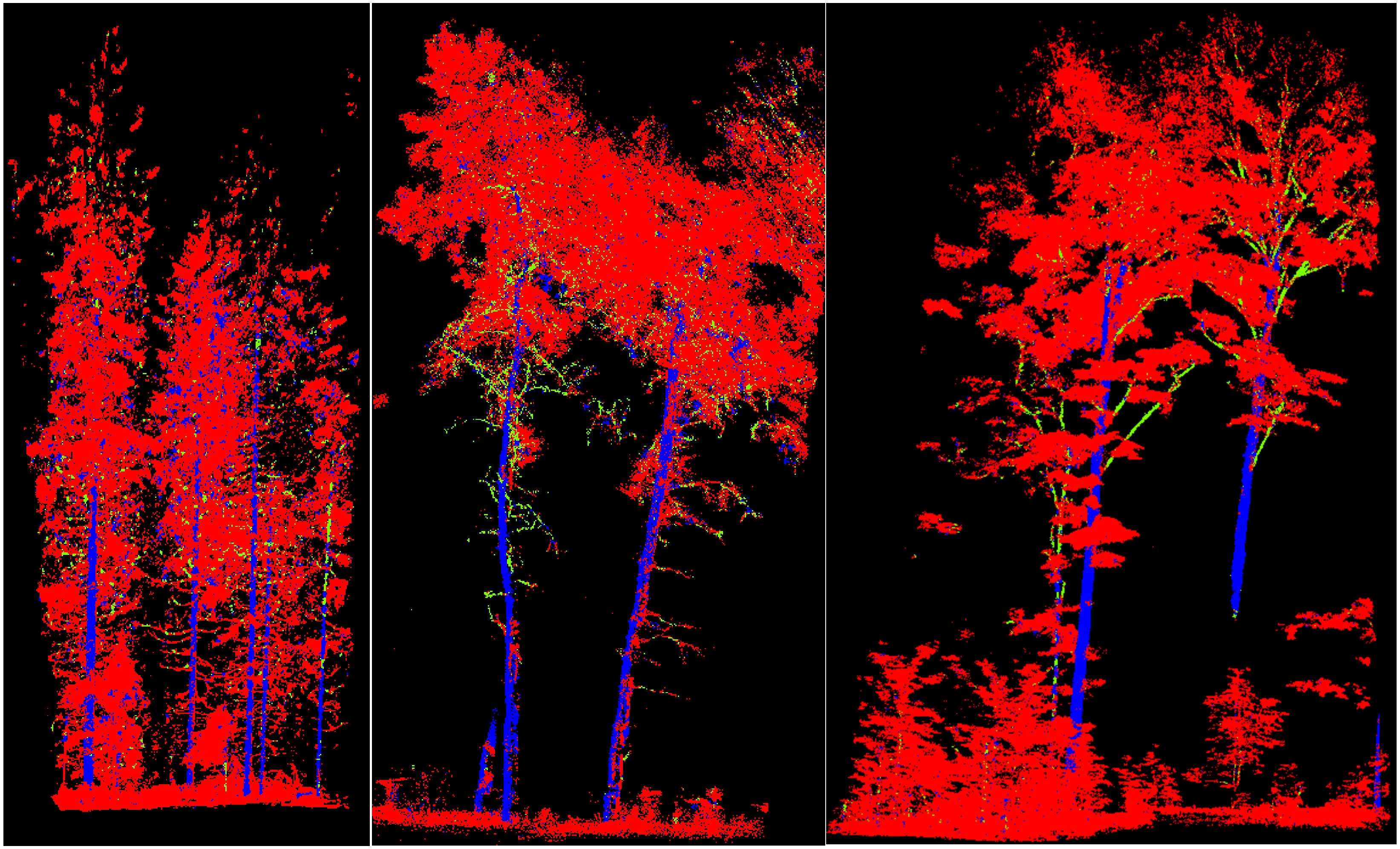
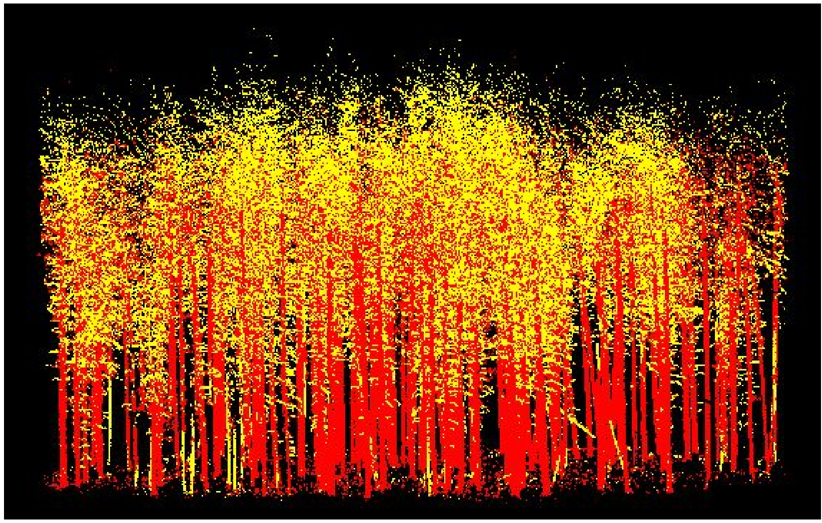
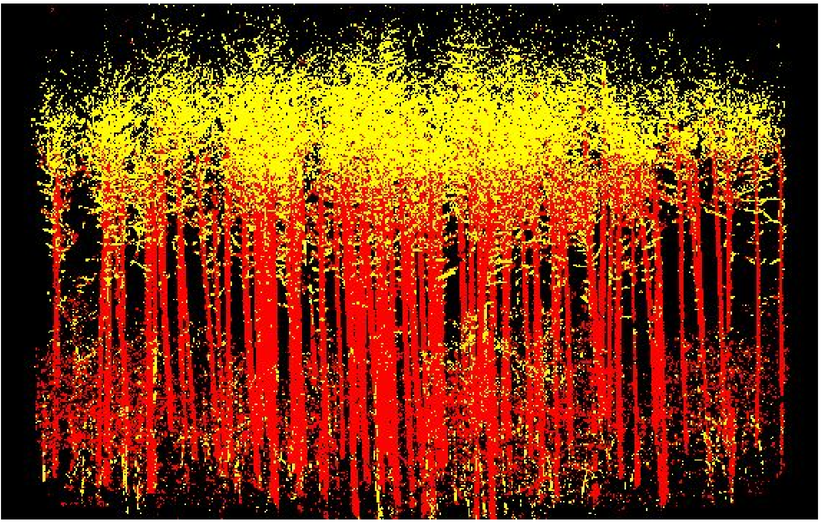


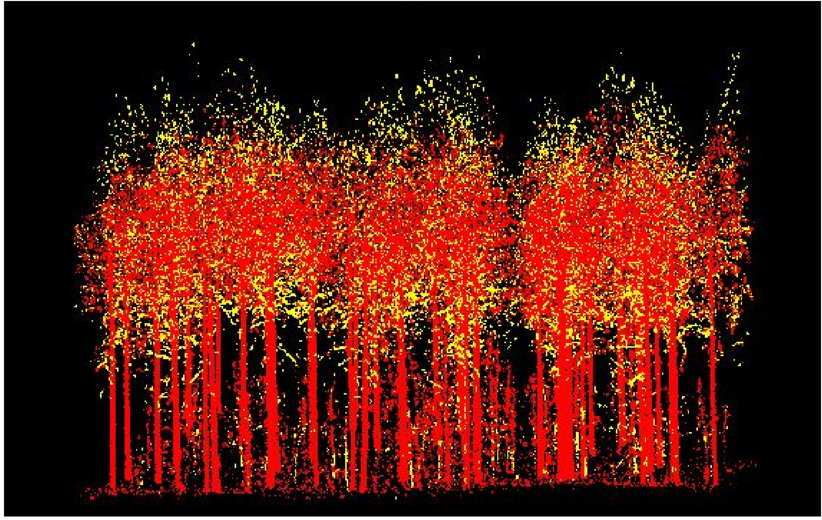
Figure 4.5 Random forest classification of an example (**a**) lodgepole pine, (**b**) red pine, (**c**) Scots pine, (**d**) silver birch (**e**) Norway spruce, (**f**) trembling aspen, and (**g**) sugar maple clip from plot scans, with stem, branch and other components in blue, green and red.



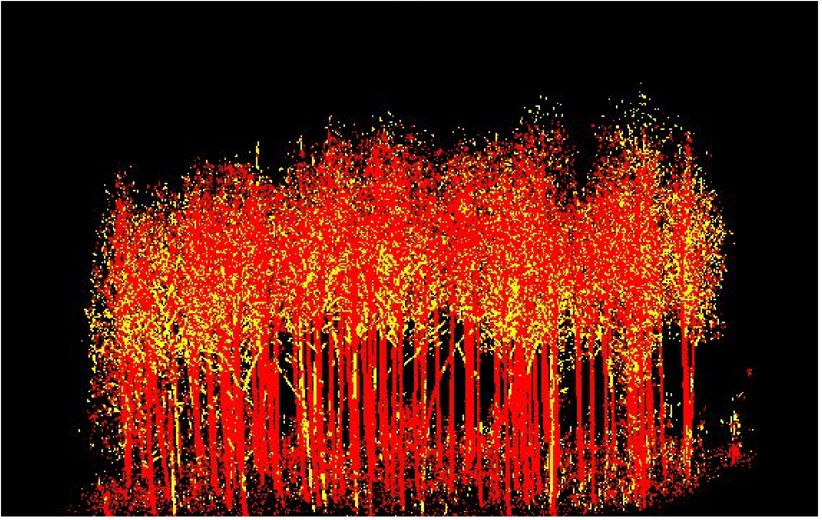
(a)



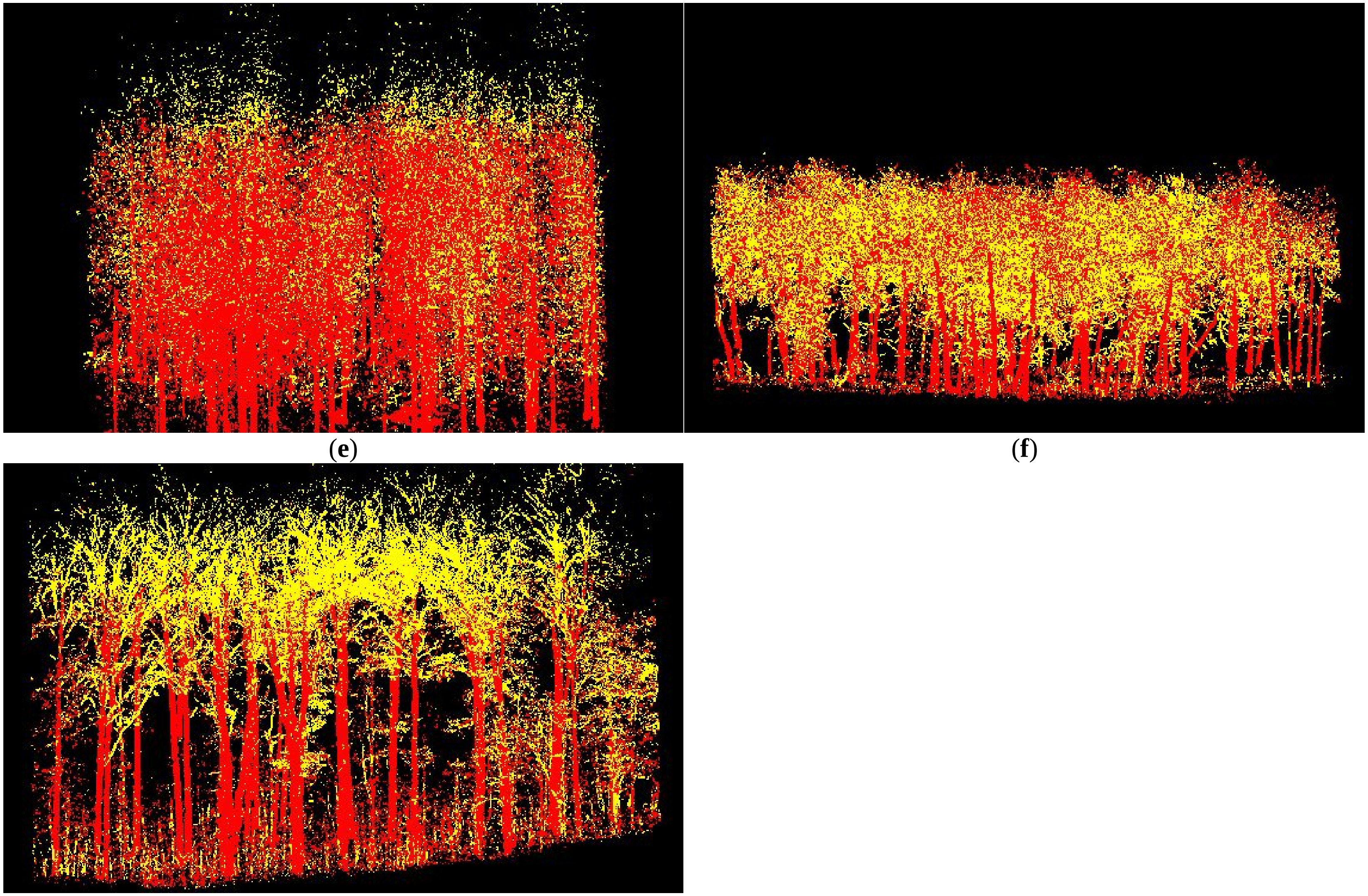
(b)



(c)



(d)



(e) (f)

(g)

Figure 4.6 Plot-level wood filtering results using random forest: (**a**) lodgepole pine, (**b**) red pine, (**c**) Scots pine, (**d**) silver birch, (**e**) Norway spruce, (**f**) trembling aspen, and (**g**) sugar maple, where stem and branch points are shown in red and yellow.

Figure 4.7 A visual comparison of sugar maple wood classification between random forest (left) and 3D-FCN (right). Individual trees were assigned with contrasting colors to help visual differentiation.

4.3.2 Species Classification

For each of the 250 manually selected trees, the 3D coordinates of wood points (stem + branch) classified by RF are used to train deep learning models. After training accuracy converges, the VGGNet, PointNet, PointNet++, Kd-Net and contextual Kd-Net attain an overall testing accuracy of 0.88, 0.75, 0.92, 0.63 and 0.70, respectively. PointNet++ outperforms all other models. The reported training and testing accuracy of PointNet++ during the training process is shown in Figure 4.8. The two accuracies are based on the subset 20 training and 20 testing samples, just for the accuracy tracking purpose. The accuracy changes follow a regular convergence pattern akin to that of wood filtering in Chapter 3. After the 100th iteration, the reported training accuracy tends to level off, but the testing accuracy continues fluctuating between 0.7-1.0 and below the training accuracy. An abnormal drop of both training and testing accuracies is noticeable at the 200th iteration step, due to a manual pause of training process to detect convergence. After the training process is restored, the training accuracy drops by almost 0.1 and undergoes more than 50 iterations to recover the previous status of highest accuracy. The entire 400 iterations of the training process can be completed within 15 minutes.

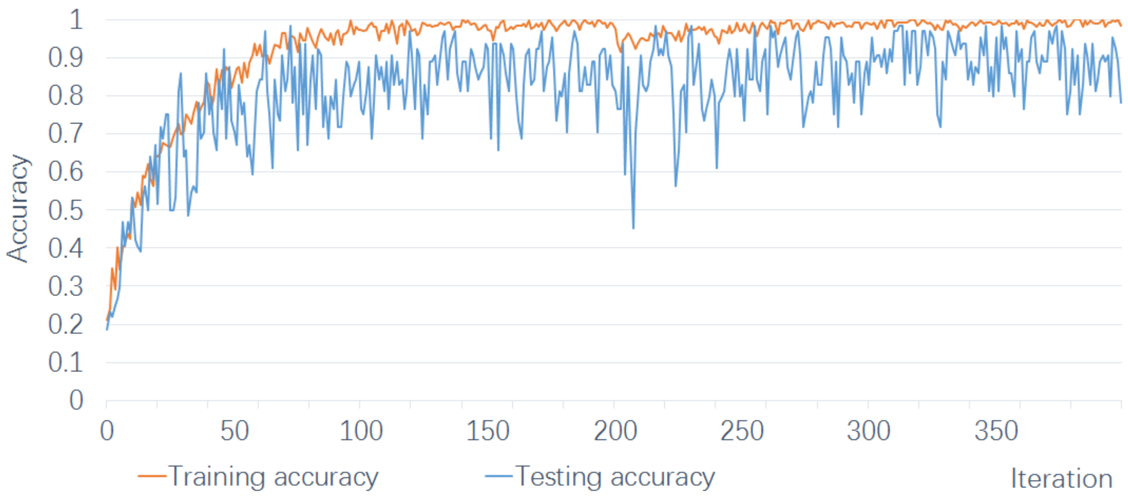


Figure 4.8 Convergence of the PointNet++'s training and testing accuracy.

The PointNet++ model with minimal testing loss based on the 20 testing samples is defined as the optimal classifier. The optimal classifier is applied to predict our entire testing dataset (63 samples) and the result OA is 0.92. The species-specific accuracy metrics are shown in Table 4.5. Among the seven species, except for the lodgepole pine, all IoUs have high values above 0.8. The red pine, silver birch and Norway spruce are classified perfectly. The precision and recall accuracies are also identical except the lodgepole pine, indicating strong prediction stability. The problem of lodgepole pine can be found based on the confusion matrix in Table 4.6. It is misclassified as aspen, scots pine or silver birch. Based on manual interpretation from Figure 4.6, lodgepole pine shows certain branch structural similarity with the misclassified species, yet its low first branch height can be a strong identifier for classification. Therefore, the PointNet++ uses its own interpretation mechanism, which differs from human approaches, and the features 'seen' from the PointNet++ may be less global or abstract compared to a human's vision.

Table 4.5 Classification accuracy for each species of the testing dataset based on PointNet++.

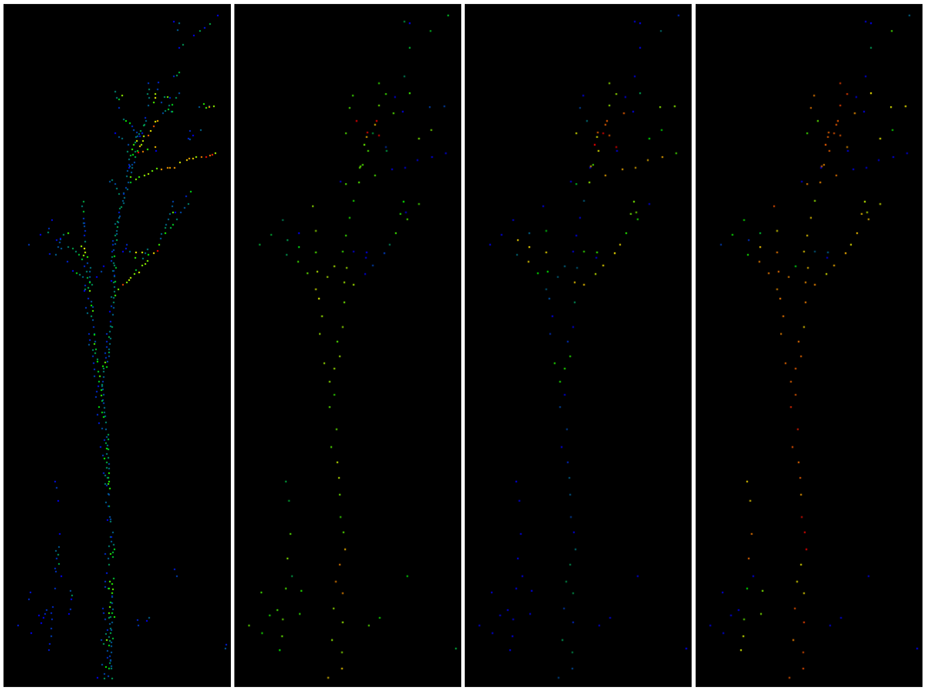
	Trembling aspen	Lodgepole pine	Red pine	Sugar maple	Scots pine	Silver birch	Norway spruce
IoU	0.86	0.55	1.00	1.00	0.83	0.95	1.00
Precision	0.92	0.67	1.00	1.00	0.91	1.00	1.00
Recall	0.92	0.75	1.00	1.00	0.91	0.95	1.00

Table 4.6 Classification confusion matrix for each species of the testing dataset based on PointNet++, with TA short for trembling aspen, LP lodgepole pine, RP red pine, SM sugar maple, SP Scots pine, SB silver birch, and NS Norway spruce. The reference classes are in columns and predicted classes in rows.

	TA	LP	RP	SM	SP	SB	NS
TA	12	1	0	0	0	0	0
LP	1	6	0	0	1	0	0
RP	0	0	4	0	0	0	0
SM	0	0	0	4	0	0	0
SP	0	1	0	0	10	0	0
SB	0	1	0	0	0	18	0
NS	0	0	0	0	0	0	4

The filtering results of the first two layers (a.k.a. sampling and grouping layers) from the PointNet++ are illustrated in Figure 4.9. Raw points become sparser after being filtered by each layer, but the distribution of filtered points basically retains the main structure of the raw point clouds (Figure 4.9a-c). It can be inferred that global position information can propagate through layers of the PointNet++ model for high-level interpretation. The filtering layers also extend the feature dimensionality of the raw coordinate data, in our case, from 3 to 64 by the first layer, and from 64 to128 by the second layer. The abundance in feature space can be helpful to high-level interpretation. To identify the most important feature dimensions among the 64 or 128 total dimensions, feature amplitudes are averaged among all points in each feature dimension, which quantifies the overall importance of each feature dimension. Figure 4.9d-i show three most important feature dimensions filtered by the first two layers of the PointNet++. Warmer point color denotes higher feature amplitude. From Figure 4.9d-f, the feature dimensions filtered by the first layer are correlated and not distinct, probably due to low-level filtering effect. In contrast, the three feature dimensions filtered by the second layer show more representativeness in Figure 4.9g-i. Nevertheless, it is difficult to understand what the feature amplitude represents from a human perspective. Other suitable visualization and interpretation methods are needed to relate the human and deep learning classification processes. For example, using the class activation map method (Zhou et al., 2016), important image zones for each class can be visualized by averaging convolutional features with weights and interpolating the feature map into a hot zone map.

(a) **(b)** **(c)** **(d)** **(e)**



(f) **(g)** **(h)** **(i)**

Figure 4.9 Sugar maple point clouds filtered by the first two layers of the PointNet++: (**a**) example sugar maple point clouds, colored by laser intensity, (**b**) points filtered by the first layer, (**c**) points filtered by the second layer, (**d**,**e**,**f**) points filtered by the first layer, colored by feature strength, and (**g**,**h**,**i**) points filtered by the second layer, colored by feature strength.

4.4 Conclusions

Generalizing biomass analysis from multisource scan data requires an automatic scheme of species classification. This chapter proposed a cutting-edge framework of species classification from complex TLS data. Seven machine-learning classifiers were evaluated to classify wood components from TLS point clouds with benchmarking values of contrasting the deep learning model in Chapter 3. The optimal classifier, namely, random forest was applied to filter wood points from seven plot scans with different species. Four point-based deep learning models were investigated to classify species from wood points of individual trees. The species classification from the best model PointNet++ reached a high overall accuracy of 0.92, a competitive accuracy level compared to contemporary studies using dimension- or QSM-based classification methods.

A few problems are also exposed in this chapter. During the wood filtering process, the best machine learning classifier only shows a branch classification accuracy of 0.36, compared to the branch IoU of 0.54 using a typical deep learning model with a more complex dataset. In addition, training the machine learning classifiers needs trials and errors of feature extraction, feature selection and parameter tuning, while a deep learning model does not. The deep learning models presented in Chapter 3 and this chapter only require point cloud input, instead of features. In our case, the prediction time of the best machine learning classifier, random forest, is about 20-30 minutes per plot, compared to the 3D FCN's 30-60 minutes per plot with a same plot size. However, the random forest classifier needs additional feature generation time of about 15-20 minutes. The overall prediction time difference between random forest and 3D FCN is not large. Therefore, in terms of both accuracy and time efficiency, deep learning is recommended to be the most practical choice currently to solve wood segmentation problems.

Imperfections are also encountered to adapt the point-based deep learning model originally developed for indoor datasets to the problem of species classification. First, the existing point-based classifiers require trimming point cloud data to a predefined size. This inevitably leads to a loss of useful details. Second, testing accuracy continues with great fluctuation after convergence. This model instability issue could induce a risk of missing optimal and generalizable parameters for classification. Third, the mechanism of deep learning is still a black box. Deep learning technology essentially relies on a stochastic search among an enormous dimensionality of parameters to minimize errors. How to converge to a global minimum instead of local minimum both accurately and effectively is a bottleneck in current deep learning research. The learning process of deep learning models basically starts from scratch and expects a long duration before marginal features and data bias can be properly identified and discarded. It lacks interaction and guiding information from the user to reduce unnecessary ‘guess work’. The first obstacle of understanding and communicating with the deep learning black box is the enormous dimensionality of deep learning feature parameters and lack of effective tools to probe and visualize that feature structure. The success of point-based deep learning such as PointNet++ and Kd-net is actually a good example that conventional CNN network structure is modified based on the user’s *a priori* knowledge of point cloud structure (e.g. multi-scale structure or kd-tree structure).

Deep learning technology is just emerging in many application areas. Yet owing to the rate of technical development, empirical methods are widely applied in practice where physical understanding is limited. For example, the best point cloud classifiers such as PointNet++ usually decompose point clouds into meaningful clusters or groups first and then classify clusters with high intelligence. A global classification question can be converted to pointwise segmentation question based on FCN strategies. Batch normalization operation can greatly improve learning convergence. Unlike traditional machine learning, such deep learning settings are not fully explained by a framework of statistical or optimization theories yet. They gradually become widely accepted by the deep learning community from a practical perspective. The statistical theorization of those rules and settings needs time and will eventually enhance our understanding of species differentiation processes and attributes.

Chapter 5. Developing a Lightweight Leddar Optical Fusion Scanning System (FSS) for Canopy Change Monitoring

5.1 Context

Monitoring *in-situ* canopy variables such as leaf area index (LAI) combined with meteorological and biochemical measurements are valuable for developing canopy light interception and biomass growth models (Sumida et al., 2018; van der Sande et al., 2015). The leaf area variable, in particular, can be used to quantify leaf biomass, since leaf biomass is approximately a product of leaf density, leaf thickness and total leaf area (Das, 2014). Passive optical sensors such as digital cameras are cost-effective ground-based tools and have been applied to monitor canopy variables such as LAI and the fraction of absorbed photosynthetically active radiation (fPAR) (Kim et al., 2019). However, the passive optical approach provides limited accuracy due to lack of stem and crown level details. For example, the LAI estimated from the Beer Lambert geometric-optical model (de Wit, 1965; Ross, 1981), also termed effective LAI, is usually 55-65% of the true LAI (Zheng & Moskal, 2009). The constraint of sensing tools inevitably leads to sophisticated tuning efforts for the geometric-optical model, such as the introduction of gap size distribution, clumping factor, and needle-to-shoot area ratio (Yan et al., 2019).

The emerging terrestrial laser scanning (TLS) technology greatly mitigates the LAI modeling problems. The 3D datasets from TLS not only enables straightforward estimation of canopy model variables including leaf angle distribution (LAD) (Li et al., 2018d; Zhao et al., 2015), clumping index (Zhu et al., 2018), canopy foliage profile (Hopkinson et al., 2013b), gap fraction (Hancock et al., 2014) and plant area volume density (PAVD) (Calders et al., 2014), but has also led to the development of more accurate canopy models. For example, the non-randomness of leaf distribution, conventionally described using the clumping index or gap size distribution, can be explicitly modeled in path length distribution equations (Hu et al., 2018; Hu et al., 2014). Basically, the path length distribution (PATH) model relates the light extinction degree with the detailed optical paths extractable from 3D point clouds. The LAI variable can be estimated from the PATH with high accuracy and stability benchmarked by the true LAI (Hu et al., 2014; Yan et al., 2019). Therefore, capturing 3D information has great implications for modeling canopy variables.

Conventional TLS sensors have shown impressive measurement precision and reliability of capturing 3D canopy information, yet lack portability and affordability for widespread use at landscape level. An alternative solution is to integrate low-cost LiDAR sensors into an existing ground sensor network for broad-scale monitoring purposes. Manufacturers such as Faro and Leica produce portable terrestrial or mobile LiDAR at a price range of 4,000 USD and 20,000 USD with moderate frequency (20k-300k) and centimeter-level resolution. Those sensors are not cost-effective, power-saving, compact and flexible enough for widespread 3D biomass monitoring. In recent years, tiny LiDAR scanners such as from Velodyne, Ouster, Hokuyo, SICK, Ibeo and Scanse have entered the market with a price level between 100 - 4,000 USD. Most tiny scanners have limited field of view (FOV) and low point detection frequency. Many low-cost tiny scanners scan in only 2D dimensions: recording laser distance returns with a spinning mirror or motor. Multiple scanlines are thus required to produce sufficient vertical points, while low frequencies can cause serious 3D distortions from fast-moving platforms or targets (Zhu & Liu, 2013).

Using a multi-beam laser scanner can be a balanced choice of budget and frequency. Instead of repetitive scanning in vertical direction, multi-beam scanner relies on detection arrays to record multiple distances instantly. Each beam from detection arrays usually has high detection frequency over 50 Hz. Among the multi-beam scanners, a LED-based LiDAR from LeddarTech, or termed as Leddar, stands out as an economical choice with 16 beams, 100Hz frequency and a thousand-dollar cost. A Leddar sensor implements patented algorithms to estimate traveling distances of each pulse emitted from a LED light source and detected by an array of 16 PIN photodiodes (Olivier, 2015). Each beam corresponds to a solid angle of around 2.8 ° x 7.5 ° and the field of view for 16 beams is customizable to be between 9 ° and 95 °. The sensor can record multiple distance returns from multiple objects at different distances. Leddar's capability of rapid data acquisition and multiple object detection has enabled growing applications in canopy detection (Gangadharan et al., 2019), autonomous driving (Arnay et al., 2018; Mimeault & Cantin, 2013), traffic analysis (Godejord, 2018; Thakur, 2016), parking assistance (Mimeault, 2014), and drone altitude estimation (Elaksher et al., 2017; Hentschke et al., 2018). However, due to limited FOV and sparse beams, the Leddar sensor alone cannot compete with conventional static TLS for detailed 3D canopy modeling or geometric mensuration.

Adding a rotational robot to the Leddar sensor can be a cost-effective solution to expand its FOV and enrich point clouds. The boresight of integrated system, however, needs to be calibrated in order to deliver precise and consistent 3D data or point clouds. One type of LiDAR calibration method can be statistical correction based on a known target. For example, Bohren et al. (2008) corrects 3D ground point clouds from SICK and Velodyne scanners based on the planarity constraint of ground. A more rigorous calibration method is to model the physical relationship between the LiDAR system and calibration targets. For example, Muhammad and Lacroix (2010) uses a planar target to calibrate five intrinsic parameters of Velodyne HDL-64E S2 system mounted on a static rotator, including two beam angle and three origin parameters. Atanacio-Jiménez et al. (2011) expands the calibration target to be a room of five planes and calibrates intrinsic and extrinsic parameters of Velodye HDL-64E. Zhu and Liu (2013) estimate origin and orientation of the HDL64E S2 sensor by aligning point clouds based on pole-shaped features. Several other studies provide convenient physical calibration approaches without a requirement for measuring reference target coordinates. For example, Levinson and Thrun (2011) propose a global calibration method for 192 orientation or distance parameters generated from a moving trajectory of Velodyne HD-64E S2 sensor. Point clouds are self-calibrated by maximizing local planarity without a need for a deliberate reference target. Similarly, Sheehan et al. (2012) proposes an entropy-based self-calibration method maximizing the crispness of point clouds for three SICK LMS-151 laser scanning units.

The above calibration methods inevitably require accurate position and orientation data from external sources such as wheel encoder, GPS or IMU, and also dense points to support point cloud alignment and registration. These requirements are usually not affordable in terms of a low-cost scanning system with Leddar and rotational motor. A possible solution is to integrate a camera sensor. The extrinsic parameters such as pose and origin can be estimated using photogrammetric methods. The predominant approach from existing studies is to re-project LiDAR point clouds to a 2D plane and co-register with an image based on corresponding features (Bodensteiner et al., 2011; Budge et al., 2014; Fremont & Bonnifait, 2008; Li et al., 2015; Park et al., 2014; Zhou & Deng, 2012). The calibration accuracy from these studies greatly depends on highly dense point clouds which a low-cost Leddar sensor is unable to produce. Among the few studies on calibrating sparse LiDAR data, Debattisti et al. (2013) places focus on edge points of artificial targets visible from both an image and a SICK LMS221 sensor. Debattisti et al. (2013) also points out that point clouds from low-cost LiDAR usually leads to laborious scanning in pursuit of sufficient corresponding points from both image and point cloud. Considering the characteristics of a low-cost scanning system, this study will utilize a planar calibration target to avoid the prerequisite for dense point clouds and also integrate a camera to provide extrinsic pose estimation.

Adding a camera sensor to the scanning system not only satisfies the calibration and alignment need, but also provides useful texture details. A question of interest is how to integrate the texture information from camera with the sparse point clouds from a Leddar scanning system, to produce dense colored point clouds. Indeed, a variety of existing literature, Hartley and Zisserman (2003) in particular, already illustrate the feasibility of reconstructing dense 3D point clouds from stereo or multiple images directly without LiDAR distance. However, with regard to our small scanning system with one rotational camera, the short movement baseline will lead to poor 3D reconstruction quality. Assuming the target is far from the camera, the error of depth estimate (ΔD) is related to the error of stereo matching between two images (Δx) in Equation 5.1:

$$\Delta x = \frac{fd\Delta D}{\mu D^2} \tag{5.1}$$

Assuming the baseline d of the small scanning system is about 3 cm, focal length f 3.6 mm, pixel size μ 2.8 μm and target distance D is 20 m, then a small Δx of 0.1 pixel will lead to a ΔD of 1 m. We can conclude that using the monocular camera alone without Leddar point clouds in this case cannot lead to any meaningful point clouds. Therefore, a camera cannot replace Leddar in recovering depth (Z) information, but can be useful to fill 2D planar (XY) gaps between the beams from the interpolation point of view. The methods for interpolating or filling point clouds are many, yet few studies consider this level of fusion. De Silva et al. (2018) is among the few studies which matches resolution between image and point clouds using a Gaussian process model. This study will combine both the classic structure from motion (SFM) method and posterior interpolation method for dense 3D recovery.

The objective of this chapter is to (1) configure a compact and low-cost fusion scanning system (FSS) including multi-beam Leddar, monocular camera and rotational robotics, and (2) propose an entire framework of calibration and fusion algorithms that produce dense colored point clouds covering a hemispherical view for 3D canopy monitoring. The dense point clouds can provide rapid canopy information such as gap fraction and LAI, useful for monitoring foliage biomass status and changes. Most parts of the framework are automatic and aimed to reduce potential manual cost. For example, only a few ground truth measurements and one horizontal scan is required to complete calibration, and only ten minutes of field scanning is necessary to reconstruct dense point clouds over a hemispherical view. We expect our work could promote more research into lightweight scanning systems for cost-effective 3D environmental mapping and monitoring.

5.2 Methodology

5.2.1 Hardware Customization and Data Processing Framework

The Leddar optical FSS mainly consists of two sensors: a 16-beam Leddar sensor (Leddar M16) and a web camera sensor in a 3D printed enclosure (13.3 x 9.1 x 4.1 cm). The two lightweight sensors (< 300 gram) sit on top of a tilting arm (DDT560 Direct Drive Tilt) and a panning base shown in Figure 5.1. Beside and beneath the pan-tilt arms are rotational servos which drive the pan-tilt movement and determine the angular resolution and span of the rotation. Specifically, the tilting servo (Hitec HS-5485HB) is a standard digital servo that rotates between 0-118 ° with highest resolution of 0.6 °. The pan servo (Dynamixel MX-12W) can have 360 ° rotation with 0.08 ° resolution and can feedback rotation angles in real time. Specifications of Leddar, camera, and servos are provided in Table 5.1. Camera video and Leddar distance data are collected and stored by Raspberry Pi and servo rotations are manipulated by an Arduino Mega 2560 board. The detailed connections between sensors, pan-tilt robotics and electronic controllers are shown in Figure 5.2.

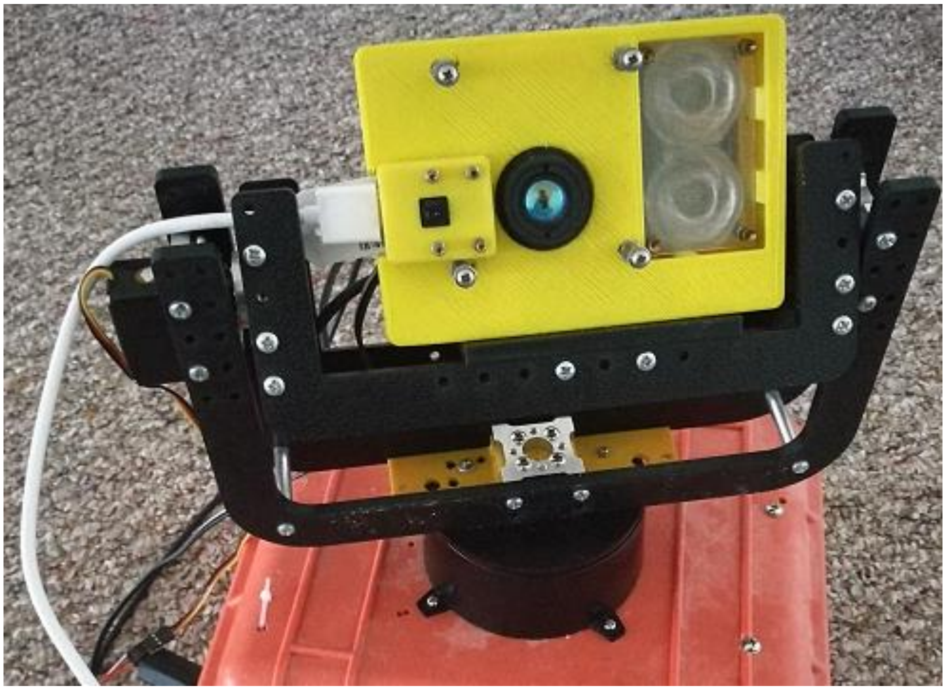


Figure 5.1 A Fusion Scanning System (FSS) with Leddar and monocular camera sensors.

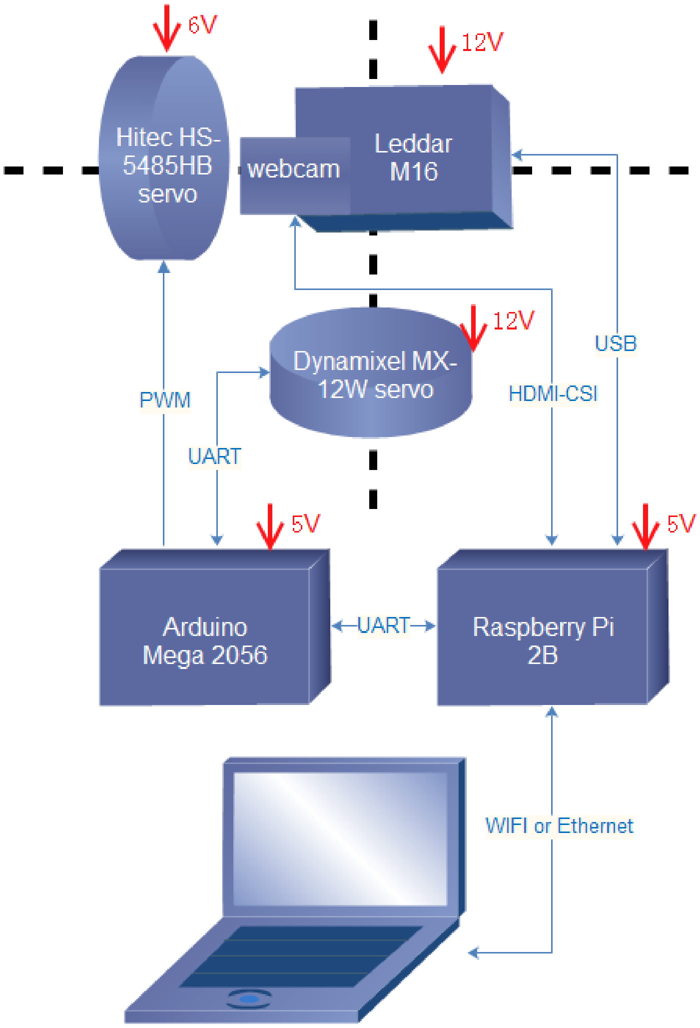


Figure 5.2 Hardware components and connections of the FSS. PWM: Pulse Width Modulation. UART: Universal Asynchronous Receiver/Transmitter. CSI: Camera Serial Interface.

Table 5.1 Major hardware specifications.

Camera	Leddar	Tilt servo	Pan servo
OmniVision OV5647	M16 module	Hitec HS-5485HB	Dynamixel MX-12W
FOV: 54 °x 41 °	Distance: 0 to 50 m	Max angle: 118 °	Max angle: 360 °
Lens: f=3.6 mm, f/2.9	Frequency: ≤100 Hz	PWM: 750-2250 µs	Steps: 4096
Calibration: no IR filter	Wavelength: 940 nm	Deadband: 8 µs	Resolution: 0.088 °
Application: IR filter	Power: 12/24 V, 4 W	Power: 4.8-6.0V	Voltage: 12V

To scan a wide field of view (e.g. hemispherical view), the pan-tilt system rotation follows a common raster scanning scheme: fixing tilting angle for one horizontal scanline and changing tilting angle for the next. For each horizontal scanline, recordings from Leddar, camera and servo are stored asynchronously in separate files by Raspberry Pi. The Leddar sensor outputs timestamp, beam distance, echo amplitude and echo quality index with an updating frequency of 100 Hz. Raspberry Pi camera captures 720p video with a rate of 25 frames per second (FPS). The rotation angle from pan servo is saved every 4°. The Leddar and pan angle readings are synchronized based on their millisecond-level timestamps tagged by Raspberry Pi. The camera videos do not have timestamps and their timing is inferred from motion detection.

Constructing the hardware of the FSS is technically straightforward. The system components are relatively inexpensive and uncomplicated compared to most commercial LiDAR scanning systems, yet fusion of different low-resolution data sources to generate dense point cloud is the challenge. The framework of our multi-source data fusion is illustrated in Figure 5.3, including: (1) mapping discrete Leddar distances onto individual video frames to create “3D pixels”, (2) aligning video frames globally to cover a panorama field of view, and (3) adjusting frame alignment and extrapolating 3D point clouds based on the “3D pixels”.

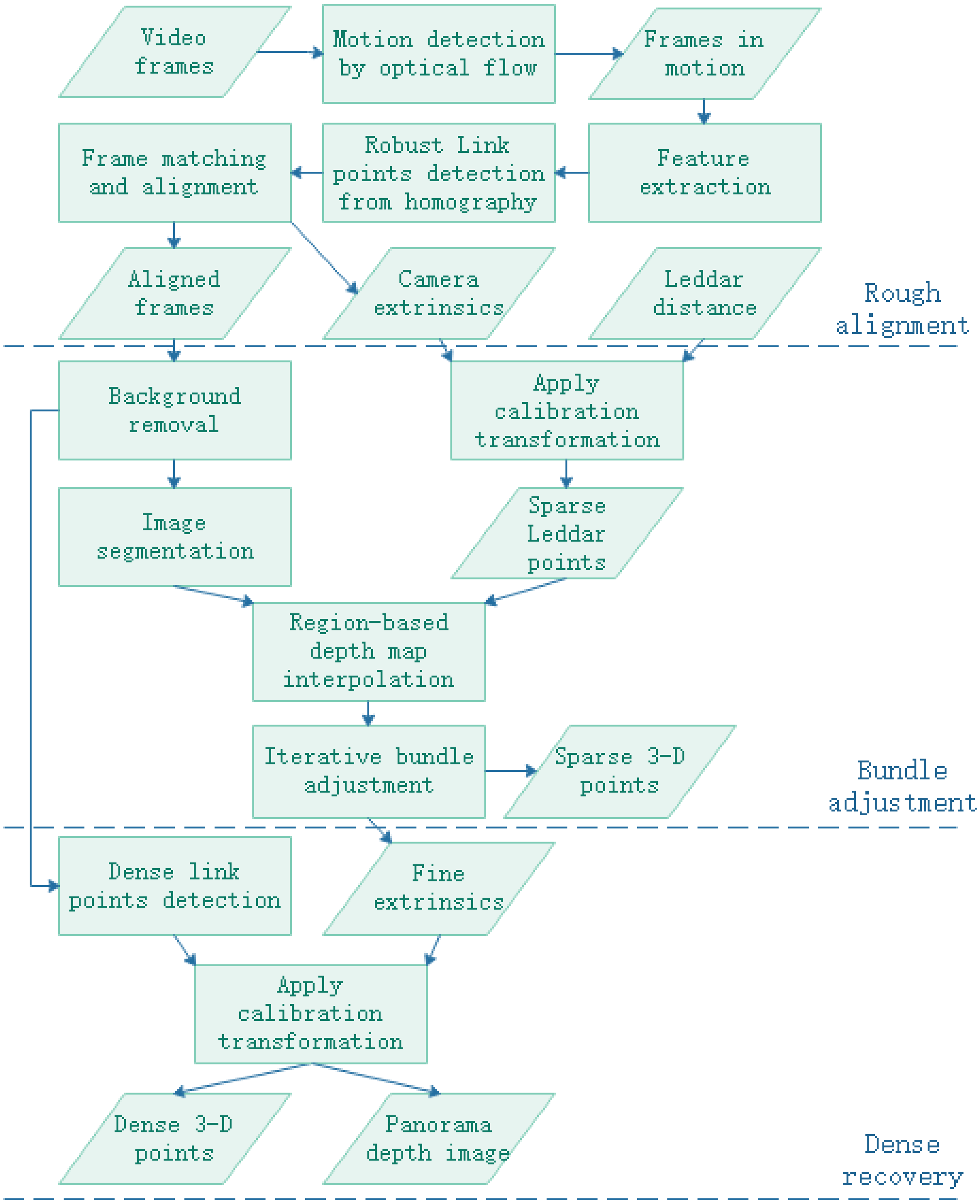


Figure 5.3 Framework of fusion point cloud recovery from monocular camera videos and sparse Leddar beams.

5.2.2 Coordinate System Conversion for Calibration

Image frames decomposed from camera video do not have timestamps directly linkable to Leddar timestamps. It is necessary to match the camera motion with the Leddar motion and assign timestamps to camera frames. The start and stop of camera motion are detected by optical flows from neighboring frames: frames with average pixel velocity above a threshold of 0.6 pixel are considered moving. With start and stop timestamps, the entire timestamps of the moving frames can be recovered based on linear interpolation.

The synchronization of camera and Leddar enables one-to-one mapping between Leddar distances and camera frames. We need to further locate Leddar footprints on the pixels of each video frame. This is a problem of calibrating the FSS, or specifically, determining unknown boresight parameters that convert Leddar coordinate system to the camera coordinate system. Calibrating the FSS is only needed once which enables the step of applying calibration transformation in the framework in Figure 5.3.

Our reference coordinate system (RCS) is a right-handed Cartesian coordinate system. It has same units as world coordinate system (WCS), with its origin at the camera optical center, x axis along the transverse direction of the image plane, y axis up along longitudinal direction of the image plane, z axis along camera optical direction facing viewer. The target 3D coordinates $\overline{\boldsymbol{P}}_{\mathrm{t}}$ in RCS can be parametrized in Equation 5.2:

$$\overline{\boldsymbol{P}}_{\mathrm{t}} = \begin{pmatrix} T \\ 0 \\ 0 \end{pmatrix} + (D_{it} + D_b) \begin{pmatrix} \cos(\theta_i)\sin(\psi_i) \\ \sin(\theta_i)\sin(\psi_i) \\ \cos(\psi_i) \end{pmatrix} \tag{5.2}$$

where $\psi_i = \psi_0 + (8.5 - i)\psi_\Delta$, $\theta_i = \theta_0$, $i = 1,2,3, \dots ,16$

where t means a specific time point, T the horizontal location of Leddar optical center in RCS, polar angles θ_i and ψ_i the orientation of i^{th} beam segment in RCS, D_{it} the distance measurement of i^{th} beam segment, D_b the bias of distance measurement, and $\overline{\boldsymbol{P}}_{\mathrm{t}}$ the target 3D coordinates in RCS. We consider T, D_b, θ_i and ψ_i to be constant during pan-tilt rotation for no relative movement between Leddar and camera. Assuming all the beam segments are equiangular, angle ψ_i can be presented by an arithmetic sequence parametrized with ψ_0 and ψ_Δ (Equation 5.2).

The real-world coordinates of target $\boldsymbol{P}$ can be converted from $\overline{\boldsymbol{P}}_{\mathrm{t}}$ in RCS through camera extrinsic parameters in Equation 5.3:

$$\boldsymbol{P} = \mathrm{R}_t\overline{\boldsymbol{P}}_{\mathrm{t}} + \boldsymbol{T}_t \tag{5.3}$$

where R_t is the rotation matrix, $\boldsymbol{T}_t$ the translation vector. R_t can be characterized by Euler angles $(\alpha_t, \beta_t, \gamma_t)$ following z- y- x rotation order. Assuming the initial rotation matrix is R_0 or Euler angles $(\alpha_0, \beta_0, \gamma_0)$, the temporal change of R_t during a horizontal scan can be represented by rotation matrix R_w in Equation 5.4:

$$\mathrm{R}_t = \mathrm{R}_0\mathrm{R}_w; \tag{5.4}$$

$$\mathrm{R}_w = \begin{pmatrix} 1-2(u_yu_y+u_zu_z) & 2(u_xu_y-u_wu_z) & 2(u_wu_y+u_xu_z) \\ 2(u_xu_y+u_wu_z) & 1-2(u_xu_x+u_zu_z) & 2(u_yu_z-u_wu_x) \\ 2(u_xu_z-u_wu_y) & 2(u_yu_z-u_wu_x) & 1-2(u_xu_x+u_yu_y) \end{pmatrix};$$

$$\boldsymbol{T}_t = \begin{pmatrix} X_{ct} \\ Y_{ct} \\ Z_{ct} \end{pmatrix} = \begin{pmatrix} X_{c0} + a_1\phi_x + a_3\phi_z - (a_{01}\phi_x + a_{03}\phi_z) \\ Y_{c0} + b_1\phi_x + b_3\phi_z - (b_{01}\phi_x + b_{03}\phi_z) \\ Z_{c0} + c_1\phi_x + c_3\phi_z - (c_{01}\phi_x + c_{03}\phi_z) \end{pmatrix};$$

where

$$\begin{pmatrix} u_x \\ u_y \\ u_z \\ u_w \end{pmatrix} = \begin{pmatrix} sin(\phi_\alpha)cos(\phi_\beta)sin(\omega_t/2) \\ cos(\phi_\alpha)sin(\omega_t/2) \\ sin(\phi_\alpha)sin(\phi_\beta)sin(\omega_t/2) \end{pmatrix};$$

$$\mathrm{R}_t = \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix}, \mathrm{R}_0 = \begin{pmatrix} a_{01} & a_{02} & a_{03} \\ b_{01} & b_{02} & b_{03} \\ c_{01} & c_{02} & c_{03} \end{pmatrix}.$$

where ϕ_α, ϕ_β define the horizontal rotation axis and w_t is the horizontal rotation angle measured by the servo. The camera optical center $\boldsymbol{T}_t$ also slightly moves during the horizontal scan, whose temporal change can be parameterized by its initial position (X_{c0}, Y_{c0}, Z_{c0}) and rotation origin (ϕ_x, ϕ_z) in Equation 5.4.

Solving the calibration equations Equation 5.2, 5.3 and 5.4 requires measurement of $\boldsymbol{P}$ and D_{it} from the same target point. However, since the exact Leddar point is invisible from the web camera, it is impossible to measure the exact 3D coordinates for $\boldsymbol{P}$ in the real world. Instead, we can reduce the requirement of the 3D $\boldsymbol{P}$ by finding a planar target with constant Z values (Z_0) and arbitrary X and Y values. Therefore, combining Equation 5.3 and 5.4, $\overline{\boldsymbol{P}}_{\mathrm{t}}$ and in RCS should satisfy a planar constraint in Equation 5.5:

$$(c_1 \quad c_2 \quad c_3)\overline{\boldsymbol{P}}_{\mathrm{t}} + Z_{ct} = Z_0 \tag{5.5}$$

where Z_{ct} is the Z component of $\boldsymbol{T}_t$. With Equation 5.2 and 5.5, the only two required measurements are D_{it} at multiple time points and Z_0 of the planar target.

The above four equations form a set of nonlinear calibration equations with nine unknown intrinsic terms (T, D_b, θ_0, ψ_0, ψ_Δ, ϕ_α, ϕ_β, ϕ_x, ϕ_z) and six initial extrinsic terms (X_{c0}, Y_{c0}, Z_{c0}, α_0, β_0, γ_0). Our solution is iterative. The initial extrinsic terms (X_{c0}, Y_{c0}, Z_{c0}, α_0, β_0, γ_0) are solved using least-square regression of camera collinearity equations (Equation 5.6):

$$\boldsymbol{x}_t = \mathrm{K}\overline{\boldsymbol{P}}'_{\mathrm{t}} = \mathrm{K}\mathrm{R}_t^{\mathrm{T}}(\boldsymbol{P}' - \boldsymbol{T}_t) \tag{5.6}$$

$$\mathrm{K} = \begin{pmatrix} \frac{f}{\mu} & 0 & x_0 \\ 0 & \frac{f}{\mu} & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

given additional measurements of pixel coordinates $\boldsymbol{x}_{t=0}$ and the corresponding world coordinates $\boldsymbol{P}'$ (details in Section 5.2.3). The $\mathrm{R}_t^{\mathrm{T}}$ in Equation 5.6 denotes transposed R_t in Equation 5.3. For simplicity, the camera intrinsic parameters in Equation 5.6 are fixed, including camera focal length f, pixel size μ, half image width x_0 and half height y_0 in pixels. Lens distortion is not considered in this study. Combining Equation 5.4 and 5.6, both extrinsic terms (X_{c0}, Y_{c0}, Z_{c0}, α_0, β_0, γ_0) and intrinsic (ϕ_α, ϕ_β, ϕ_x, ϕ_z) can be inferred by non-linear least-square regression, and further substituted into Equation 5.2, 5.3 and 5.5 to finally estimate Leddar intrinsic parameters (T, D_b, θ_0, ψ_0, ψ_Δ).

After knowing all intrinsic and extrinsic parameters, locating Leddar points on camera images is feasible using Equation 5.2 and 5.6. It is also possible to roughly estimate a Leddar point $\boldsymbol{P}$ given D_{it} and w_t from pan-tilt angles based on Equation 5.2, 5.3 and 5.4. Instead of using pan-tilt angles, we can also rely on camera global alignment in the following section for more precise inference of R_t and $\boldsymbol{T}_t$ and thus estimate $\boldsymbol{P}$ from Equation 5.2.

5.2.3 Calibration Experiment

This section presents a case experiment of system calibration, with a flat wall ($Z_0 = 0$) being our calibration target (Figure 5.4). The left bottom corner of the wall was defined as the WCS origin. The sensors scanned the wall with a fixed tilt angle of about 15 ° and continuous horizontal angle from 50 ° to 140 ° (2.4 ° per second). A total of 70 frames were subsampled from the video with equal interval for calibration. On the front wall was a 16 x 9 grid from an optical projector, for the purpose of calibrating camera extrinsic parameters in Equation 5.6. As mentioned, the calibration equation (Equation 5.6) required measuring the WCS coordinates $\boldsymbol{P}'$ of each projected circle center and extracting the corresponding pixel coordinates $\boldsymbol{x}_t$ from camera frames. We used circle grid instead of chessboard pattern to be more robust to edge detection error during automatic extraction of $\boldsymbol{x}_t$ from camera images. Since the camera had limited field of view of around 50 °, not all circles appeared in camera images. Therefore, an ID number was assigned to each circle to help link $\boldsymbol{x}_t$ and $\boldsymbol{P}'$ automatically.

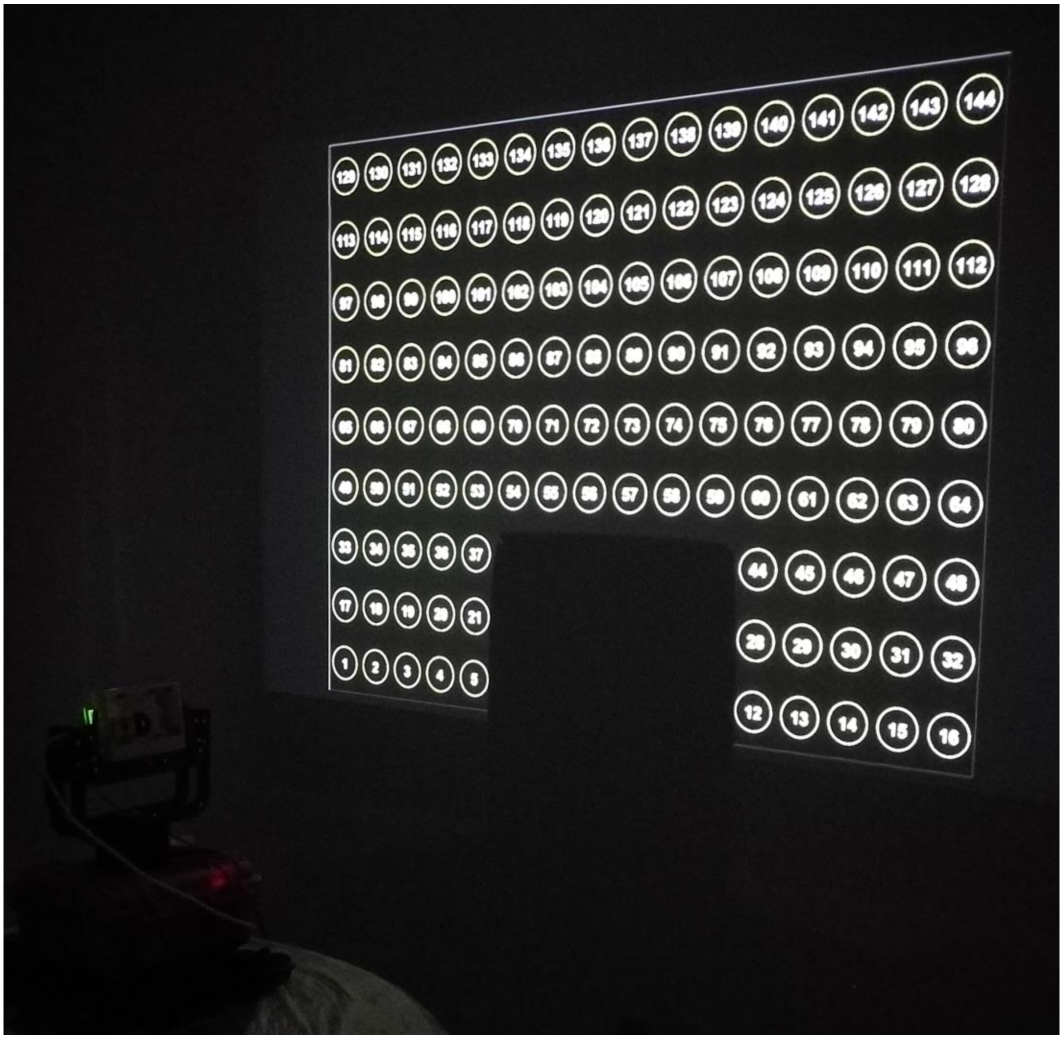


Figure 5.4 Experiment setup for the FSS calibration.

Assuming the projector lens had no distortion, we manually measured the $\boldsymbol{P}'$s of four corner circle centers on the wall and then simply applied bilinear interpolation to get $\boldsymbol{P}'$s of all the 144 circle centers. The process of extracting $\boldsymbol{x}_t$ was challenging, because circle projection on the wall from camera's view would exhibit ellipse shapes. Extracting ellipses centers is obviously more difficult than extracting circle centers. We adopted Jia et al. (2017)'s Characteristic Number Ellipse Detector (CNED) to coarsely detect ellipse parameters (centers and axis). All the settings of the CNED were default except that the CNL parameter of rejecting linearity was set to 10.0 instead of 3.0. Due to thickness of ellipses edges on the wall, redundant ellipses could be detected using the CNED. The next process simply averaged the center and axis parameters over all redundant ellipses, thinned the ellipse edges using morphological operator, and finely fit ellipse (Fitzgibbon et al., 1996) within each ellipse region outlined by its axis parameter. The ID number within each ellipse region was also identified by Optical Character Recognition (OCR) in MATLAB, with character set confined to number 0-9. Using ellipse IDs could greatly expedite the laborious search for correspondence between $\boldsymbol{x}_t$ and $\boldsymbol{P}'$ from 70 camera frames. The accuracy of OCR recognition using MATLAB was about 80%. Improper IDs were later corrected by voting from the nearest four IDs and our final recognition accuracy was 100% among the 70 frames. Given the corresponding $\boldsymbol{x}_t$ and $\boldsymbol{P}'$, the least-square Newton-Raphson iterative method (Newton, 1711; Newton & Colson, 1736; Ypma, 1995) was applied to minimize residuals in the Equation 5.2-5.6 following the aforementioned steps in Section 5.2.2. At the end of each iteration, the robust Huber's function (Huber, 1981) was applied to reduce the effect of residual outlier. Initial estimates of parameters were also required by the nonlinear Newton-Raphson method, in which $(\alpha_0, \beta_0, \gamma_0)$ were roughly set as (0 °, 40 °, 0 °), (X_{c0}, Y_{c0}, Z_{c0}) manually measured as (1.80, 1.08, 1.09) in meters, and $(T, D_b, \theta_0, \psi_0, \psi_\Delta, \phi_\alpha, \phi_\beta, \phi_x, \phi_z)$ to be (0.03m, -0.44m, 180 °, 90 °, 2.5°, 0°, 0 °, 0.06m, 0.00m).

5.2.4 Fusion-based Dense Point Cloud Recovery from FSS

Calibration is a preliminary requirement for fulfilling the framework in Figure 5.3. The framework focuses on generating and optimizing point clouds from multiple field scans. This section mainly describes a suit of visual odometry algorithms being used with a following experiment in Section 5.2.5. Our field scans cover a hemispherical view with four scanlines spanning a vertical angle between 0 ° - 120 °, albeit more scanlines can be added. To avoid data overhead, we choose systematic sampling of the moving frames. Hence, our full hemispherical scan contains 150x4 moving frames with horizontal overlap over 90% and vertical overlap about 80%. Aligning frames becomes a first problem since the extrinsic parameters in the lab calibration environment are not repeatable in a new location. The intrinsic parameters such as (T, D_b, θ_0, ψ_0, ψ_Δ) from the Leddar sensor remain unchanged. To estimate extrinsic parameters, target-based calibration or ground control points would be tedious for the hemispherical view of 600 frames. Instead, we directly use rich photogrammetric information from the video to approximate camera poses R_t and then incorporate Leddar distance into bundle adjustment for fine camera extrinsic parameters (R_t and $\boldsymbol{T}_t$).

A common way of aligning multiple frames is to extract invariant features in each frame, match features between frames, and optimize camera colinear equation (Equation 5.6). Frames with multiple scanlines also needs iterative correction of scanline skewness caused by uneven distribution of matching features. The set of global alignment has been supported by image stitching software such as PtGUI for this study. The only weakness of using existing stitching software is the limited number of matched pixels, insufficient for the bundle adjustment needed. Therefore, intensive extraction of SURF features (Bay et al., 2006) is added to the workflow in Figure 5.3, with outlier features filtered out using homograph-based RANSAC algorithm (Hartley & Zisserman, 2003). The extracted SURF features are then matched between frames. Note that a 3D point can correspond to a set of matched pixels from multiple frames. The matched pixels from more than three frames are called “key pixels” here, whose features can be considered stable and will be used for the bundle adjustment later.

The global alignment using PtGUI exports Euler angles of each frame, which can be used to interpolate Euler angles $(\alpha_t, \beta_t, \gamma_t)$ or R_t at any timepoint when Leddar distances are measured. Then Leddar point clouds $\boldsymbol{P}_t$ can be roughly recovered using Equation 5.2 and 5.3, assuming $\boldsymbol{T}_t$ is zero vector. Projecting the Leddar point clouds $\boldsymbol{P}_t$ back to each image frame will add depth information to a few pixels (or “3D” pixels here). Then the sparse “3D” pixels can interpolate depth values of all the “key pixels” which will be input to the bundle adjustment later. Our method of interpolating depth value is region-based: (1) filtering foreground in each image using k-means clustering method (k=2), (2) segmenting images using Statistical Region Merging algorithm (level=8) (Nock & Nielsen, 2004), and (3) implementing Inverse distance weighting (IDW) interpolation for each key pixel within each region. Compared to global interpolation, using region-based interpolation better maintains sharp boundary lines between different image regions.

Based on the 3D “key pixels”, both transformation matrix R_t and $\boldsymbol{T}_t$ and WCS coordinates $\boldsymbol{P}_t$ can be finely estimated using iterative bundle adjustment. First, $\boldsymbol{P}_t$ estimated from 3D “key pixels” are averaged among different frames and reprojected to each image frame using Equation 5.7. The reprojected points are normalized by Z coordinate and compared to the 2D coordinates of “key pixels”. The least square error is minimized using nonlinear regression, with camera extrinsic parameters ($\mathrm{\alpha_t}$, $\mathrm{\beta_t}$, $\mathrm{\gamma_t}$, $\mathrm{X_{ct}}$, $\mathrm{Y_{ct}}$, $\mathrm{Z_{ct}}$) as variable (Equation 5.8). New $\boldsymbol{P}_t'$ corresponding to “key pixels” is estimated using Equation 5.9 based on the optimized camera extrinsics. Note that $\boldsymbol{P}_t$ is sensitive to small errors of “key pixels” and camera extrinsics due to the ill-posed mono-camera geometry.

A robust solution to the ill-posed optimization problem is to use ridge regression (Hoerl & Kennard, 1970) which partially minimizes the least square error between $\boldsymbol{P}_t$ and $\boldsymbol{P}'_t$ in Equation 5.10. Its regularization parameter λ is set to be 0.01. The optimized $\boldsymbol{P}_t$ from Equation 5.10 is again reprojected to each image frame in Equation 5.7, and repeat iterations from Equation 5.7 to Equation 5.10 until the error in Equation 5.8 is locally minimal. This iterative bundle adjustment for 3D recovery is similar to the Gold Standard Method from Hartley and Zisserman (2003), but with depth information provided from sparse Leddar distance instead of dense stereo geometry. The resulting point clouds will exhibit rich details in the 2D planar direction but limited variation in the depth direction.

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} = \mathrm{KR}_t^{\mathrm{T}} \left[\begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} - \boldsymbol{T}_t \right] \tag{5.7}$$

$$\min_{\alpha_t, \beta_t, \gamma_t, \mathrm{X}_{ct}, \mathrm{Y}_{ct}, \mathrm{Z}_{ct}} \left\| \begin{pmatrix} x'_t \\ y'_t \end{pmatrix} - \begin{pmatrix} x_t/z_t \\ y_t/z_t \end{pmatrix} \right\|, \text{ using nonlinear regression} \tag{5.8}$$

$$\begin{pmatrix} X'_t \\ Y'_t \\ Z'_t \end{pmatrix} = \mathrm{R_t K^{-1}} \begin{pmatrix} x'_t \\ y'_t \\ 1 \end{pmatrix} + \boldsymbol{T}_t \tag{5.9}$$

$$\min_{\mathrm{X_t, Y_t, Z_t}} \left\| \begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} - \begin{pmatrix} X'_t \\ Y'_t \\ Z'_t \end{pmatrix} \right\| \text{ using ridge regression } (\lambda = 0.01) \tag{5.10}$$

The 3D recovery from bundle adjustment produces point clouds for “key pixels”. The “key pixels” are essentially from SURF feature extraction and are mostly focused on corner pixels with sharp color gradient. Other “internal” foreground pixels should also be incorporated to produce completely dense point clouds. Our method is to extract dense foreground pixels and solve Equation 5.9 and 5.10 to create optimal dense 3D points. To satisfy Equation 5.10, one 3D point requires at least one pair of matching pixels from two frames. We already have matching pixels defined as “key pixels” in previous steps. We need to extrapolate the matching relationship for all foreground pixels. This is a pixel-level dense matching process. First, the foreground pixels need to be subsampled with certain interval (e.g. 10 pixels in this study) to avoid data overhead. Then the disparities of "key pixels" between current frame and one matched frame are calculated. The disparities are used to interpolate a disparity map for all foreground pixels in the current frame. Given a disparity map, the foreground pixel location in the matched frame can be estimated. This dense matching step between a pair of matched frames is repeated for all the matching frames. Each set of matched pixels from multiple frames is given one unique ID, corresponding to one unique $\boldsymbol{P}_t$. Finally, using Equation 5.9 and 5.10, the densely matched sets of pixels can produce dense point clouds. The final RGB colors of dense point clouds are average RGBs from matched pixels.

5.2.5 Application: Tracking Autumn Defoliation Processes with the FSS

The FSS was used to track canopy changes during an autumn season in 2018. Our experiment site was located in an area of cottonwood stands in Lethbridge (49°41'45.2"N, 112°51'54.0"W). The FSS was mounted on a triIpod surrounded by six poplar trees within 20 m, including *Populus angustifolia* and *Populus deltoides* and their hybrids. The irregular shapes of poplar trees lead to difficulty of depth information recovery, but the rich texture of the scene would facilitate feature extraction in frame alignment. The camera lens filter was replaced to block near-infrared light and enable natural-colored images. Multiple scanlines were collected, each corresponding to 360° horizontal rotation at a speed of 2.4° per second. Four scanlines were selected to cover the upper hemispherical view of the scene for further processing. All scans were aligned, optimized, and densified to create colored point clouds using the methods in Section 5.2.4. The same scene was scanned with a Teledyne Optech ILRIS HD (1535 nm) TLS as a benchmark. The scanning angle is

360°x80°, with a small zenith angle between 0-10° uncovered. Only last returns were recorded and point spacing was 3.2 cm at a distance of 20 m from the TLS. A total of 30 ILRIS TLS scans were collected in 30 minutes with three scanlines covering the entire upper hemispherical view. The TLS scans were co-registered by the Iterative Closest Point (ICP) algorithm into one hemispherical scan with an average accuracy below 1.3 cm. The same scanning and processing activities were repeated on September 9th (09/09), September 17th (09/17), October 1st (10/01) and October 17th (10/17) during the autumn defoliation period in 2018, to evaluate the reusability of our static scanning system in a temporal monitoring context. Hemispherical photos based the Digital Hemispherical Photography (DHP) methods were also captured on 09/09, 09/17 and 10/17 for benchmarking purposes.

Canopy vertical volume profile and Plant Area Index (PAI) were extracted from the FSS point clouds to evaluate the capabilities of 3D canopy detection and canopy attribution. The volume profile was defined as a total volume of voxels at each height, where a unit voxel was $0.1\times0.1\times0.1\text{m}^3$ and a height slice was 0.1m. The PAI was calculated based on a path length distribution (PATH) model (Hu et al., 2014; Yan et al., 2019). Specifically, the PATH model consisted of two equations (Equation 5.11 and 5.12) with the gap fraction $\overline{P(\theta)}$ and path distribution p_l to be the only two inputs. To calculate angle-specific gap fractions, a hemispherical image from either FSS or TLS point clouds was first converted to a black and white binary image under a fisheye perspective. The fisheye binary image was equally sliced into 28 rings representing a zenith angle between 15-69° and a ring width of 4°. The overlap between two neighboring rings is 2°. The gap fraction $\bar{P}$ was defined as the ratio of the "hole" pixel numbers to the "filled" pixel numbers within a ring slice. The "filled" pixels represented the overall canopy area and was generated based on image morphological smoothing. The path distribution p_l was defined as the probability density function (PDF) of the optical path length within the crown area, with l representing the within-crown path length. The l ranged between 0-1, scaled by the maximum value l_{max}. The p_l was approximated by the histogram of the l, normalized by the total histogram area. With both the gap fraction $\overline{P(\theta)}$ and path distribution p_l extracted from crown area, the integral equation (Equation 5.11) was solvable based on any root-finding algorithm and the $FAVD \cdot l_{max}$ could be estimated, where $PAVD$ stands for the plant area volume density and $G(\theta)$ the leaf angle distribution. The $G(\theta)$ was set 0.5 in this study corresponding to a spherical leaf angle distribution (Chen et al., 2018). The $PAVD \cdot l_{max}$ was then input to the Equation 5.12 to determine the $PAI_{true}(\theta)$ at a specific zenith angle θ. The final PAI value was a weighted sum of $PAI_{true}(\theta)$ over all zenith angles (Equation 5.13) (Hu et al., 2014).

$$\overline{P(\theta)} = \int_0^1 e^{-G(\theta)\cdot(FAVD\cdot l_{max})\cdot l} p_l dl \text{, where } \int_0^1 p_l dl = 1 \tag{5.11}$$

$$PAI_{true}(\theta) = \int_0^1 \cos(\theta) \cdot (FAVD \cdot l_{max}) \cdot l \cdot p_l dl \tag{5.12}$$

$$PAI_{true} = \frac{\sum_\theta PAI_{true}(\theta) \cdot \sin(\theta)}{\sum_\theta \sin(\theta)} \tag{5.13}$$

An important portion of canopy was foliage and the corresponding index was LAI. The LAI was directly related to photosynthetic processes and carbon productivity, and was a more sensitive index than PAI to reflect seasonal biomass changes. We estimated LAI values by contrasting leaf-on and leaf-off gap fraction values, as illustrated in Equation 5.14, where N was the number of pixels and P was short for the gap fraction $\overline{P(\theta)}$. Specifically, N_{wood}, N_{leaf} and N_{hull} are the numbers of wood, leaf and canopy pixels, respectively, and P_{off}, P_{on}, and P_{leaf} were the gap fractions of leaf-off, leaf-on, leaf-only canopy, respectively. Based on Equation 5.14, P_{leaf} was a simple ratio of P_{off} to P_{on} (Equation 5.15). With known P_{leaf}, LAI was estimated in a similar manner with PAI using Equation 5.11-5.13, except for replacing $\overline{P(\theta)}$ (namely P_{on}) with P_{leaf}.

$$P_{off} = 1 - \frac{N_{wood}}{N_{hull}}, P_{on} = 1 - \frac{N_{wood} + N_{leaf}}{N_{hull}}, P_{leaf} = 1 - \frac{N_{leaf}}{N_{hull} - N_{wood}} \tag{5.14}$$

$$P_{leaf} = \frac{P_{on}}{P_{off}} \tag{5.15}$$

5.3 Results and Discussion

5.3.1 Calibration

An example frame among the 70 frames from the camera video is shown in Figure 5.5a. The original 720p frame is cropped to 1280x500 to be compact. Only part of the 16x9 circle grid is within the field of view. The frame shows no blurry effect, implying that an FPS of 25 is sufficient to match a rotation speed of 2.4 °per second. The purple band in the frame is the near-infrared LED light from Leddar, since the camera lens has no IR filter. The visibility of the LED light provides an intuitive way of validating calibration accuracy: the footprints of all the 16 Leddar beams after calibration should fall within the purple area. The ellipses on the wall are detected with the CNED method, shown in green in Figure 5.5b. It is clear that a few incomplete ellipses are skipped and many redundant ellipses are created. This is a preliminary step of approximating ellipses ranges and locations. Fine ellipses after edge thinning and geometrical fitting are shown in random colors in Figure 5.5c overlaid by the edge image. Edge noises are inevitable but have limited effect on the ellipse fitting results. Each ellipse center is marked as green cross. The integer number inside each ellipse is the circle ID predicted with the OCR and posterior voting method.

The OCR recognition confidence is also placed under each ellipse ID as a decimal number. Ellipses with low OCR confidence are removed, such as the No.142 and No.80. The remaining 33 ellipses still satisfy the minimal requirement of having four control points for the camera collinearity equation. Note that all the 70 frames have been inspected to have four or more ellipses at the beginning. After calibration, point-based footprints of the Leddar beams are projected on the example frame. The Leddar points basically fall within the LED light zone, except for the first beam on less illuminated area.

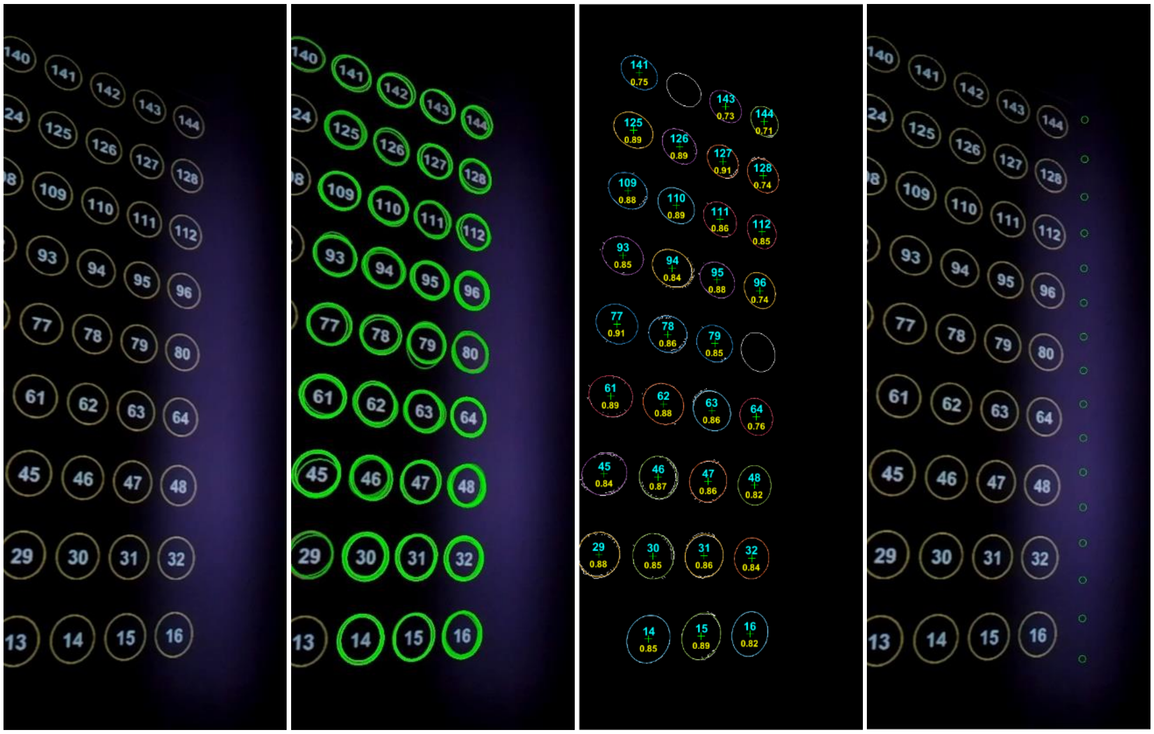
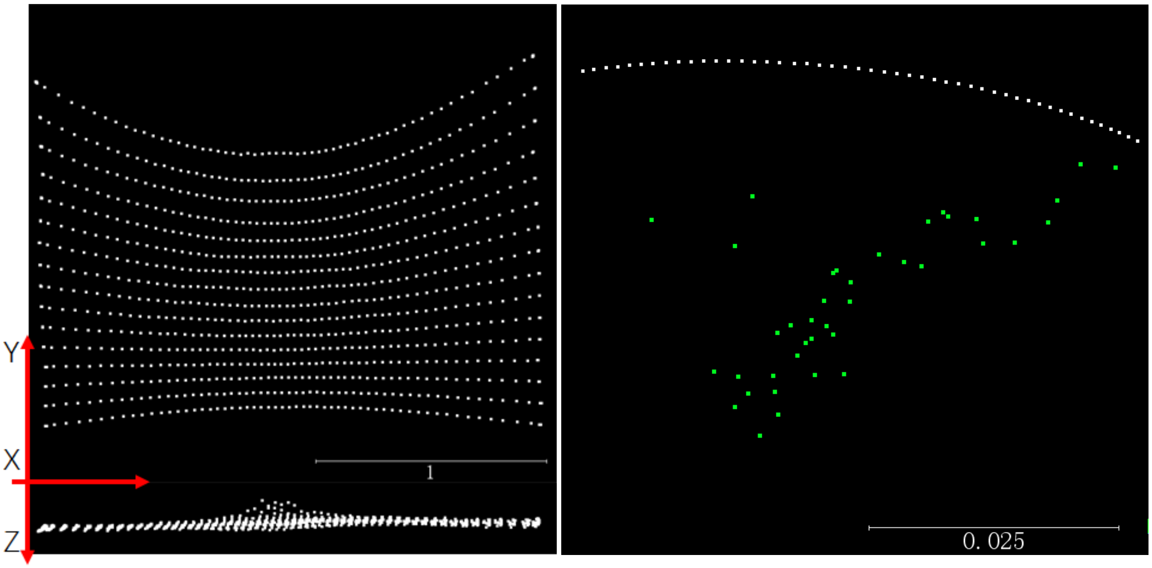

(a) (b) (c) (d)

Figure 5.5 Calibration processing of an example frame: (**a**) circle grid (yellow) and LED light (purple) from camera view, (**b**) ellipse detection (green) by CNED method, (**c**) fine ellipse fitting (random color), ellipse ID from OCR (cyan) and OCR confidence (yellow), and (**d**) calibrated beam points reprojected to the frame image (green).

The point clouds after calibration of Leddar distances and poses are displayed in Figure 5.6a, with the X-Y projection approximating a planar shape and the X-Z projection linear shape. The entire point clouds contain 16 beams, each creating 70 points along the sweeping direction. Several points overlap when Leddar is still static at the beginning or ending. The sweeping pattern of each beam on the X-Y plane is not a straight line. This is because Leddar has a fixed tilt angle of about 15 °upwards. Yet a few lines are not smooth and their noises are not systematic for all lines, probably not due to the servo movement but Leddar's instability of distance measurement instead. The standard error is 1.03 pixel for optimizing the camera collinearity equation (Equation 5.6), 3.84 pixel for optimizing the temporal rotation equation (Equation 5.4 and 5.6), and finally 9.7 mm in WCS for solving Leddar distance equation (Equation 5.2, 5.3 and 5.5). Equation 5.4 is about constraining the rotation matrix to a fixed rotation axis. Without Equation 5.4, solving Equation 5.6 for each frame separately is also feasible and can reduce the standard error to 2.34 pixel. However, the retrieved camera extrinsic parameters, such as the camera center locations $\boldsymbol{T}_t$ shown as the green dots in Figure 5.6b, will lose physical meaning and present irregular movement. Instead, applying Equation 5.4 can reflect real camera movement shown as the white arc points in Figure 5.6b and thus retrieve more reliable calibration parameters. The final estimate of intrinsic parameters (T, D_b, θ_0, ψ_0, ψ_Δ, ϕ_α, ϕ_β, ϕ_x, ϕ_z) are 0.070 m, -0.428 m, 180.84 °, 89.39 °, 3.32 °, -0.69 °, 12.19 °, 0.027 m and -0.0006 m.

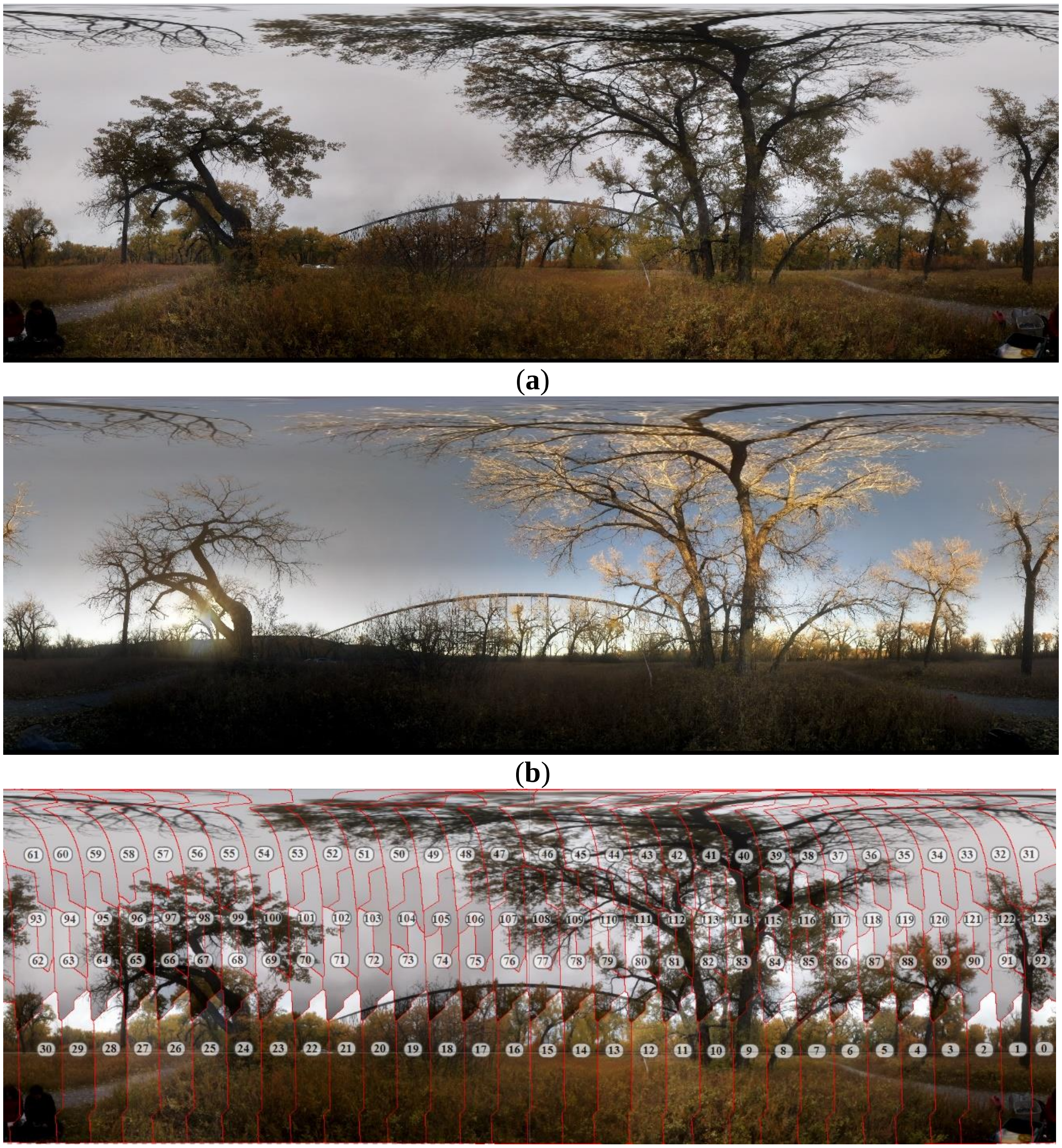


(**a**) (**b**)

Figure 5.6 Calibrated Leddar points and camera trajectory: (**a**) calibrated Leddar points of a flat wall on X-Y plane (above) and on X-Z plane (below), and (**b**) camera trajectory with rotation constraints (above) and without constraints (green below).

5.3.2 Fusion-based Dense 3D Recovery

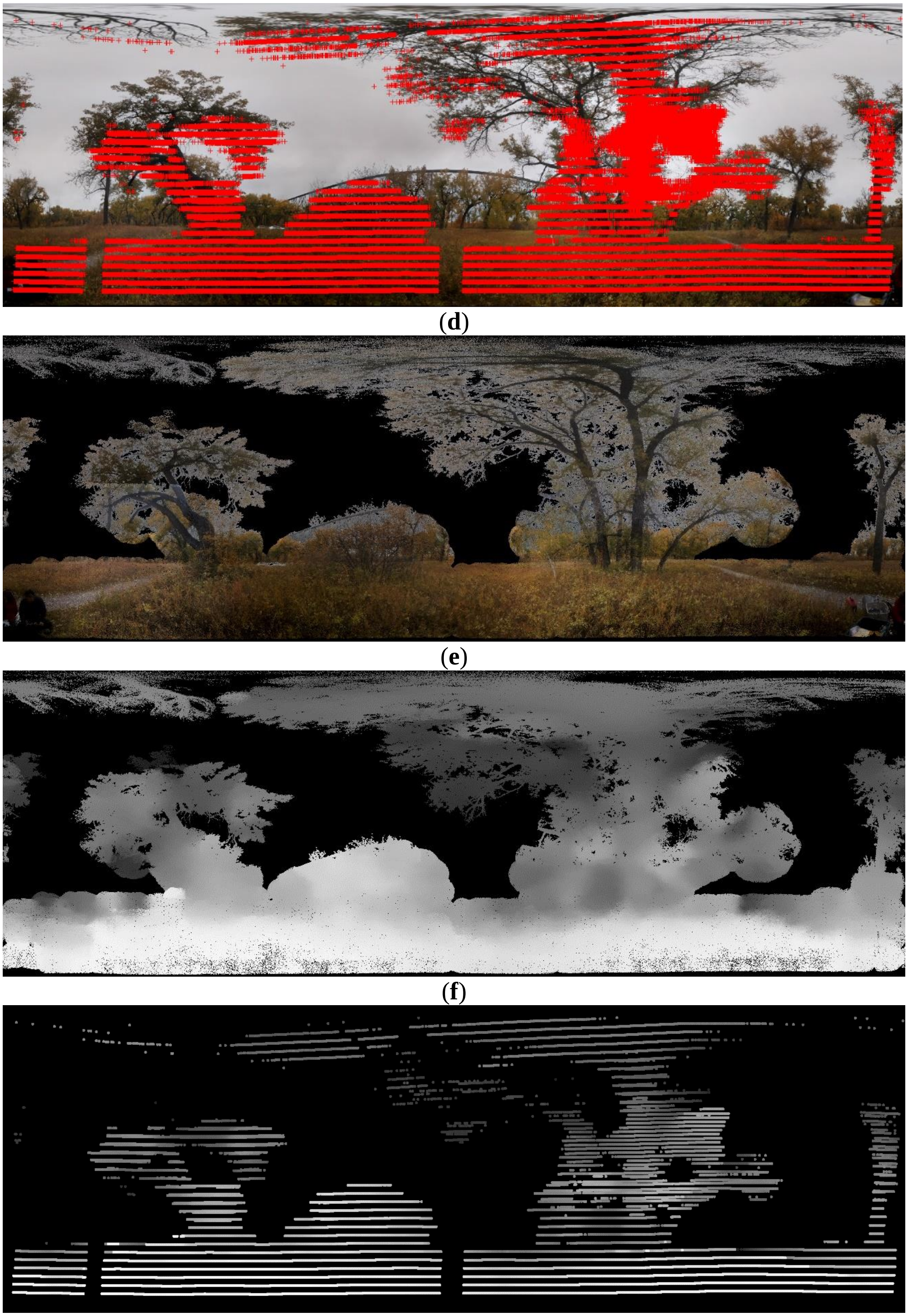
The success of point cloud recovery hinges greatly on the quality of aligning video frames, because camera poses determine the general form and structure of point clouds. Our video scans of poplar trees on four different dates are aligned based on automatic solutions provided from PtGUI, including image matching, feature extraction, feature matching, horizon correction, bundle adjustment and image mosaicking. Example alignment results for the 10/01 and 10/17 videos are visualized as 360 ° x 120 ° spherical panoramas under Equirectangular projection in Figure 5.7a,b. The trees display leaf-on condition on 10/01 and are defoliated completely on 10/17. Both scenes are centered on a railway viaduct and lower part of the panorama is discarded. No obvious alignment gap or inconsistency are found from the two images. The processing results of the leaf-on scene are visualized in Figure 5.7c-h. Figure 5.7c shows alignment of separate images and their seamlines in PtGUI without mosaicking and color blending. The colors of individual tiles in Figure 5.7c differ from each other due to sunlight variation during scan. Yet based on visual inspection, the alignment of tiles is seldom affected by the color difference, indicating strong robustness of PtGUI's feature extraction algorithms. The alignment errors estimated from bundle adjustment in PtGUI are 4.5, 3.0, 2.5, 2.5 and 2.2 pixels for the scenes of 09/09, 09/17,10/01 and 10/17, respectively. The relatively large error of the 09/09 is due to various factors such as windy conditions, thick canopy and cloudy sky.



(**a**)

(**b**)

(**c**)

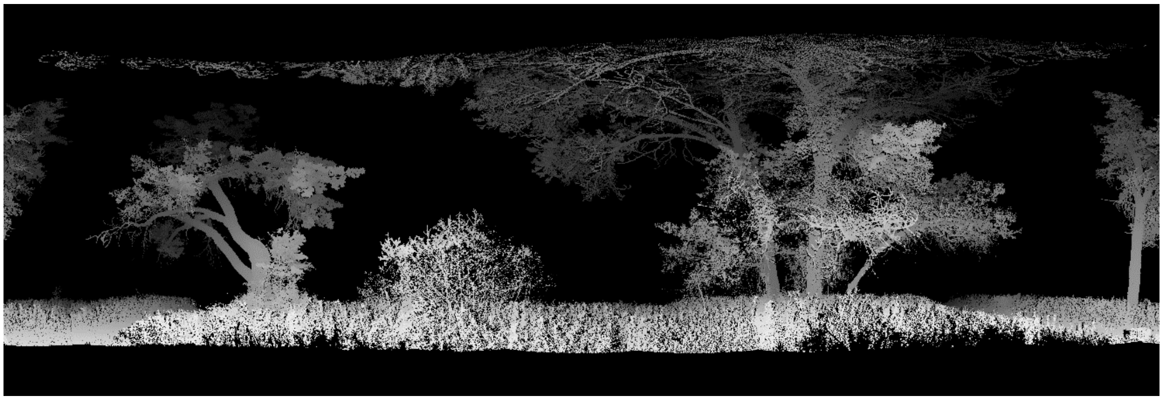


(d)

(e)

(f)

(g)



(**h**)

Figure 5.7 Hemispherical view of processing results: (**a**) image global alignment for the 10/01 scan, (**b**) image global alignment for the 10/17 scan, (**c**) global alignment layout in PtGUI software with image IDs and seamlines for the 10/01 scan, (**d**) Leddar-only point clouds (red cross) reprojected to the hemispherical image, (**e**) RGB colors from fusion-based point clouds, (**f**) depth image from fusion-based RGB point clouds, (**g**) depth image from Leddar-only point clouds (point size enlarged for clearer visualization), and (**h**) depth image from TLS scans. (**e**), (**f**), (**g**) and (**h**) are all using hemispherical projection.

Leddar points are reprojected as the red crosses in Figure 5.7d and overlaid with the panorama view of the 10/01 scene, after applying Leddar intrinsic parameters from calibration and camera extrinsic parameters from PtGUI alignment. The Leddar points capture the basic scene structure nearby except for upper canopy, distant ground and thin branches. The minimum, average and maximum detection range of Leddar in this scene is 1.64, 6.45 and 14.17m, respectively. The Leddar point clouds have obvious gaps between the beams and on the ground due to missing signals. This problem of Leddar data sparsity limits potential applications such as tree surveying and object detection, unless photographic information is integrated. Therefore, the iterative bundle adjustment is applied at the point cloud level to minimize the disagreement between Leddar reprojected pixels, camera pixels, and camera extrinsic. Iterations of the bundle adjustment error, measured in pixels, is plotted for the four scanning dates in Figure 5.8. The initial error of bundle adjustment can be greater than 8 pixels but will converge to a level comparable to

PtGUI alignment error. The final error from bundle adjustment is 2.8, 1.9, 2.0 and 1.8 pixels for 09/09, 09/17, 10/01, and 10/17, respectively.

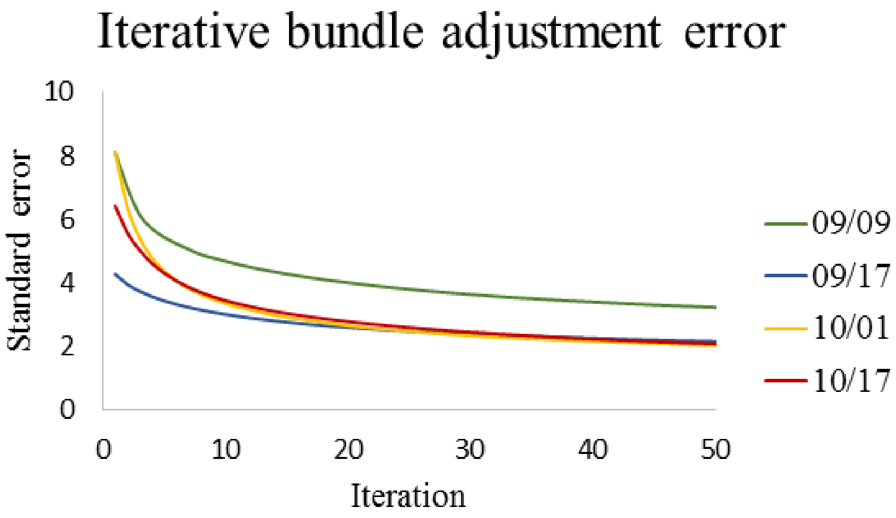
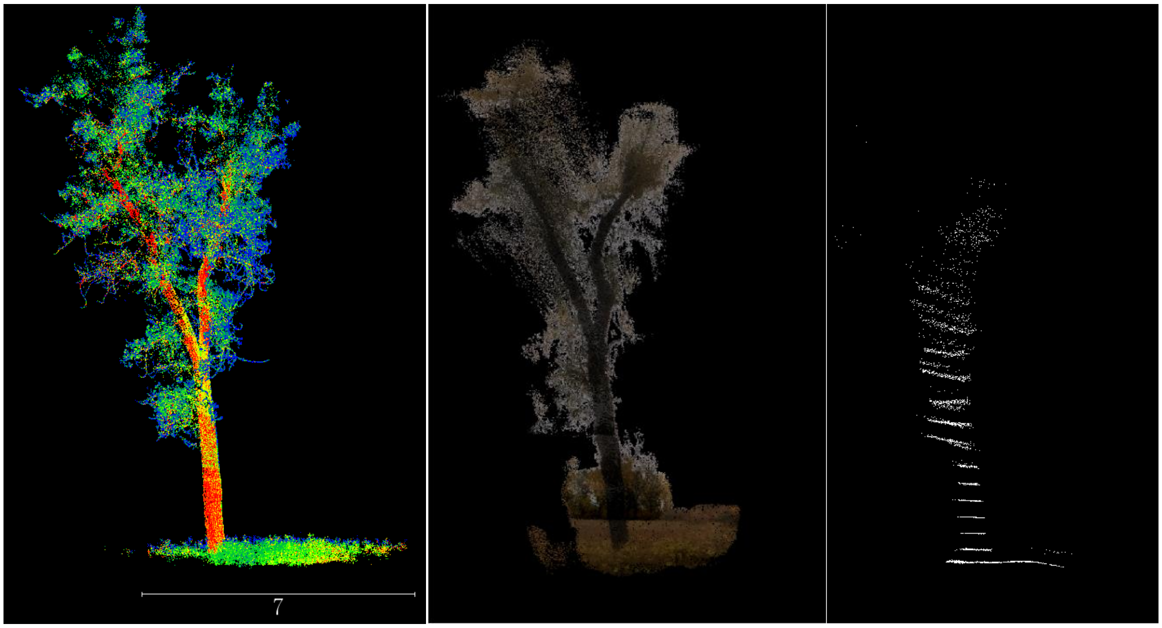


Figure 5.8 Fusion error convergence with iterations (in pixels) on four defoliating dates in 2018.

The fusion-based point clouds after image background removal, iterative bundle adjustment, and dense matching recovery are reprojected into two panorama images shown as Figure 5.7e-f. Figure 5.7e displays reprojected pixels with RGB colors, and Figure 5.7f is the corresponding depth image with nearer objects showing brighter colors. The reprojection from point clouds to a hemispherical-view image is not simply one point per pixel, considering the previous dense recovery process has a subsampling rate of 10 pixels per point. Therefore, each point has a buffer of 10 pixels in a hemispherical image. Similarly, reprojecting TLS point clouds into a hemispherical depth image in Figure 5.7h also needs to consider the footprint of each TLS laser beam. The scanning spacing of each Ilris HD beam (1600 μrad) is set to be the footprint size according to the suggestions in Hancock et al. (2014). The scanning spacing corresponds to a constant pixel size of 1.6 and thus each preprojected pixel of the hemispherical images is dilated by a factor of 1.6.

In contrast to the Leddar reprojection image in Figure 5.7d, the image in Figure 5.7e not only captures rich 2D details but also covers a reasonable extent due to the region-based interpolation. The main problem of the fusion-based point clouds is false interpolation. The problem can be illustrated when comparing specific tree point clouds extracted from TLS, fusion-based point clouds, and Leddar point clouds in Figure 5.9a-c. The TLS point clouds clearly exhibit branch-level details with warmer colors representing higher laser intensity. The fusion-based point clouds have distinguishable stem colors and noisy branches, still highly detailed compared to the obscure Leddar point clouds. Yet the fusion-based point clouds overfill the gaps between branches and also falsely incorporate pixels from remote shrubs. This is inevitable since region-based interpolation and bundle adjustment can mitigate but not eradicate the problem of coarse and sparse depth measurement from Leddar. The depth image in Figure 5.7f displays a strong smoothing effect compared to the depth image reprojected from TLS in Figure 5.7h, but are much more detailed than the Leddar-only point clouds in Figure 5.7g with indiscernible sparse points.

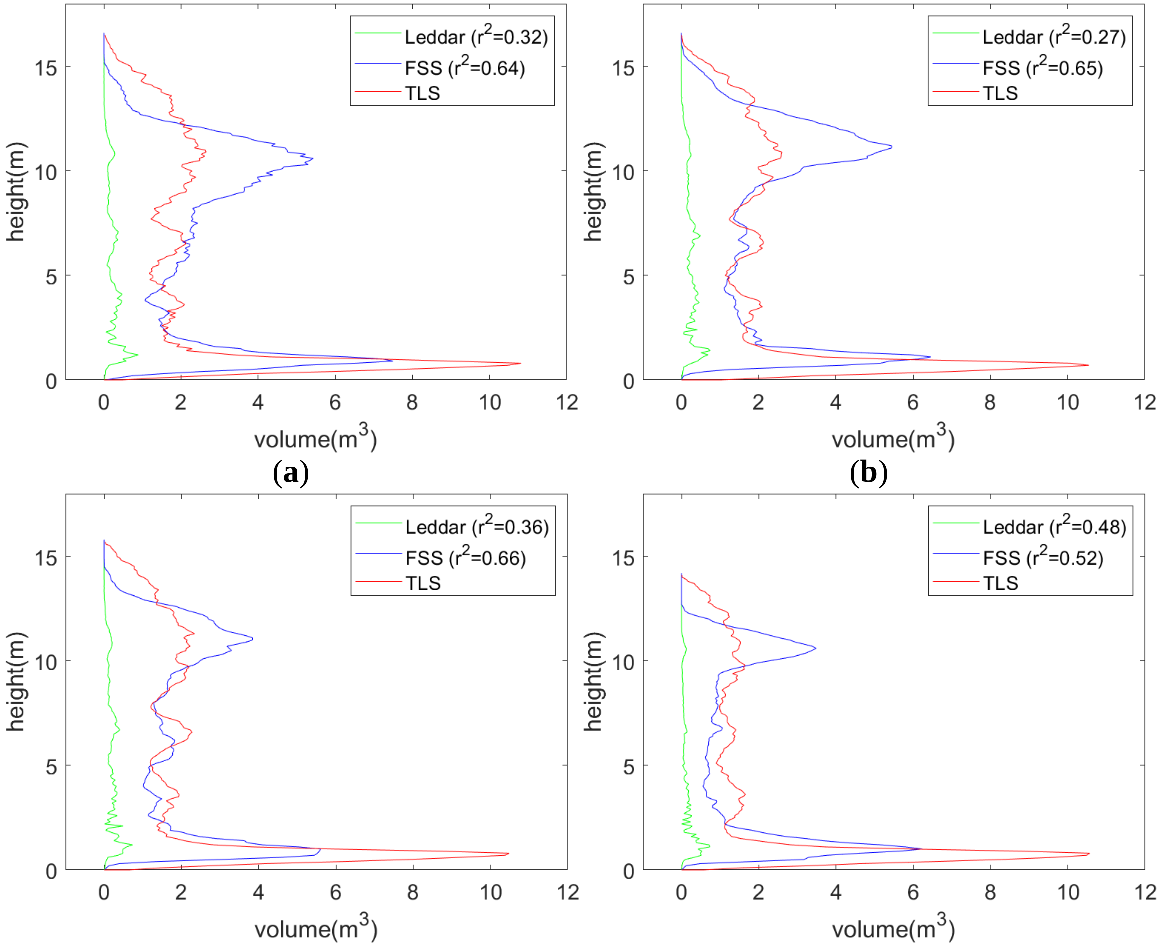


(**a**) (**b**) (**c**)

Figure 5.9 Example tree point clouds from (**a**) TLS scans, (**b**) fusion-based dense point clouds, and (**c**) Leddar-only point clouds.

5.3.3 Tracking changes of canopy vertical volume profile, PAI and LAI

Vertical volume profiles from TLS, fusion-based point clouds, and Leddar point clouds can be contrasted with Figure 5.10a-d. Both profiles from the fusion-based point clouds and the Leddar point clouds are correlated with the TLS profiles, regardless of scanning date. The r^2 of profiles over the maximum tree height range between Leddar and TLS stays around 0.3 from the first three leaf-on scenes and jumps to 0.48 from the last leaf-off scene. In contrast, the r^2 of profiles between the fusion-based and TLS are around 0.65 from the leaf-on scenes, constantly higher than 0.52 from the leaf-off scene. This is because thick crowns and leafy understory lead to Leddar signal loss but do not affect photography-based interpolation. With defoliated trees in the last scene, photography-based interpolation tends to incorporate more false pixels from a remote area and lead to lower accuracy. It is also noteworthy that the fusion-based point clouds demonstrate higher frequency than TLS point clouds in the middle crown area due to the overfilling effect, but also thinner near upper crown due to the loss of Leddar points. Note that the TLS has a slightly narrower scanning view than the FSS, with part of upper crown and ground not sampled by scans. The profile difference around upper crown and ground can be greater than observed in Figure 5.10. This problem of profile distortion might be due to the imperfect hemispherical stitching process.



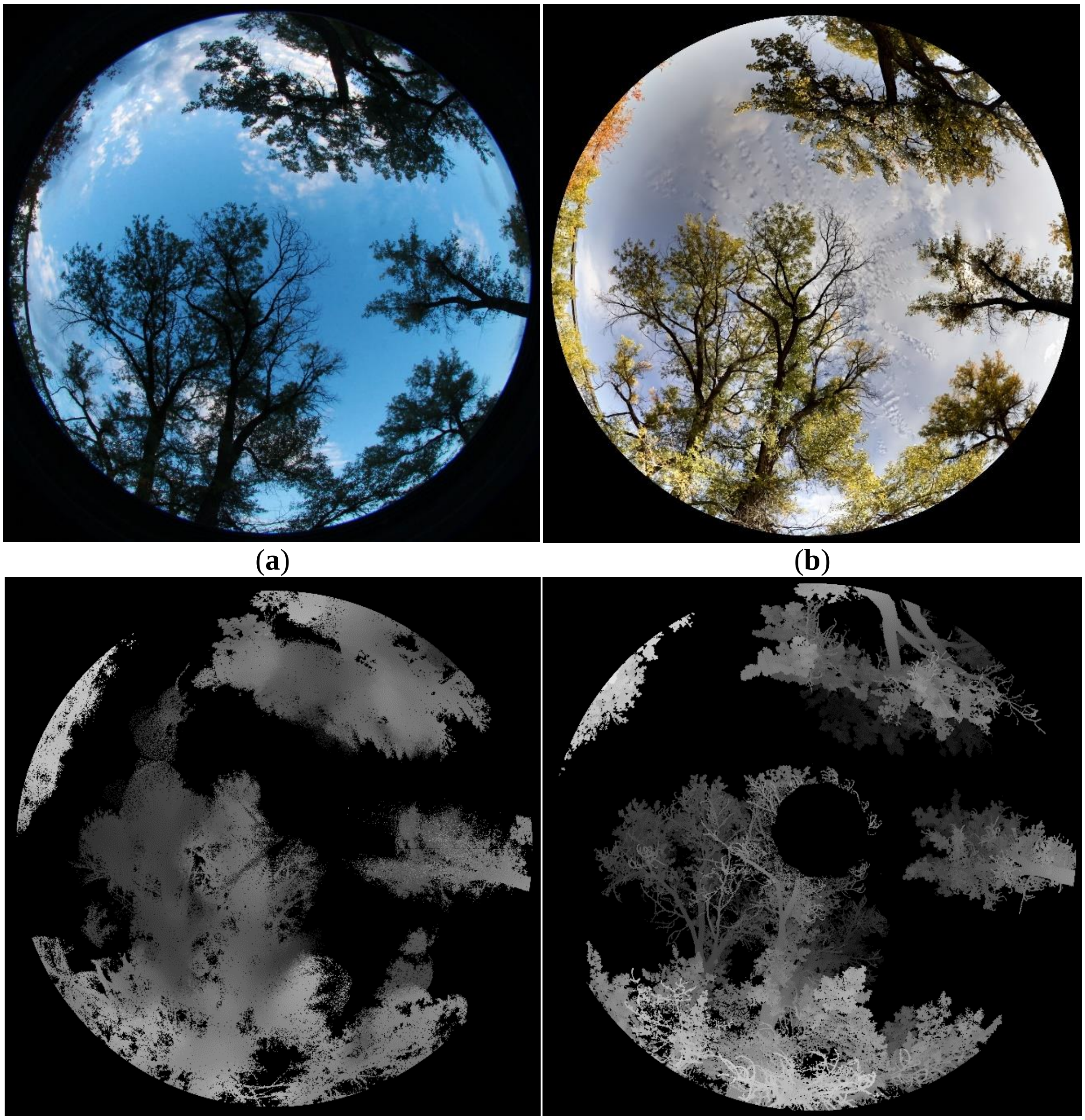
(a) (b)

(c) (d)

Figure 5.10 Vertical volume profiles from TLS, fusion-based and Leddar-only point clouds on (**a**) 09/09, (**b**) 09/17, (**c**) 10/01, and (**d**) 10/17, 2018, where horizontal axis denotes volume of voxels with a unit voxel of $0.1\times0.1\times0.1$ m^3, and vertical axis denotes height in meters.

The advantage of synthesizing both 3D and color information can make FSS a valuable complement to conventional LAI or PAI surveying tools, for example, the DHP. Figure 5.11 compares the fisheye image from FSS with the DHP photo from the same site.

The canopy shapes between the two images are visually identical, seen from Figure 5.11a and Figure 5.11b. The FSS, in addition, captures depth information such as the image in Figure 5.11c, with a benchmarking TLS depth image provided in Figure 5.11d. Note that the upper crown area was not scanned with the TLS due to the field of view constraint. The availability of depth images enables FSS to calculate true PAI and LAI based on the PATH model. Different PAI and LAI estimates based on FSS, TLS and DHP methods, and based on PATH and non-PATH methods are contrasted in Figure 5.12, with bars denoting PAI and crosses denoting LAI. The non-PATH method relies on the LAI models from the Hemisfer software (Schleppi et al., 2007; Thimonier et al., 2010) which combines Lang (1987)'s LAD function, Lang and Xiang (1986)'s clumping correction, and Schleppi et al. (2007) non-linearity correction model. The non-PATH method focuses on the RGB images from DHP or FSS, or the depth images from TLS, whereas the PATH method additionally needs point cloud input from FSS or TLS.



(**a**) (**b**)

(**c**) (**d**)

Figure 5.11 Fisheye-view images compiled from the 09/09 datasets based on (**a**) DHP, (**b**) FSS, (**c**) FSS depth, and (**d**) TLS depth.

For the non-PATH methods in Figure 5.12, PAI and LAI values generally decline with the defoliation dates, with all the LAI values reaching zero level on the leaf-off date 10/10, except that the PAI and LAI value from non-PATH FSS increases on 10/01. The incorrect increase implies the instability of using image-only methods. The possible cause of the incorrect increase is FSS's underestimation of PAI and LAI from the 09/09 and 10/01

FSS images, in contrast to the PAI and LAI values from DHP and TLS. Strong spectral reflectance from the sunlight are observed in the 09/09 and 10/01 FSS images and a small portion of canopy pixels in the FSS image displays a similar color as the sky background. These bright canopy pixels were not successfully identified as leaf area, causing the underestimation effect. The DHP method does not have the underestimation issue since the DHP images were captured near dusk. The TLS method does not have the stability issue as FSS, since depth images are used instead of color images. The TLS method, however, has an issue of overestimating PAI. The leaf-off PAI from TLS is 32% higher than from DHP, compared to the average 3% overestimation of leaf-on PAI from TLS. The overestimation issue of TLS has two typical causes. The depth images, especially the leaf-off ones, contain ghost points or misaligned points around thin branches and twigs. Gaps smaller than the beam width of TLS are also not differentiable from the depth images (Hancock et al., 2014). The TLS's overestimation of twig PAI leads to an underestimation of LAI by 26% in terms of DHP.

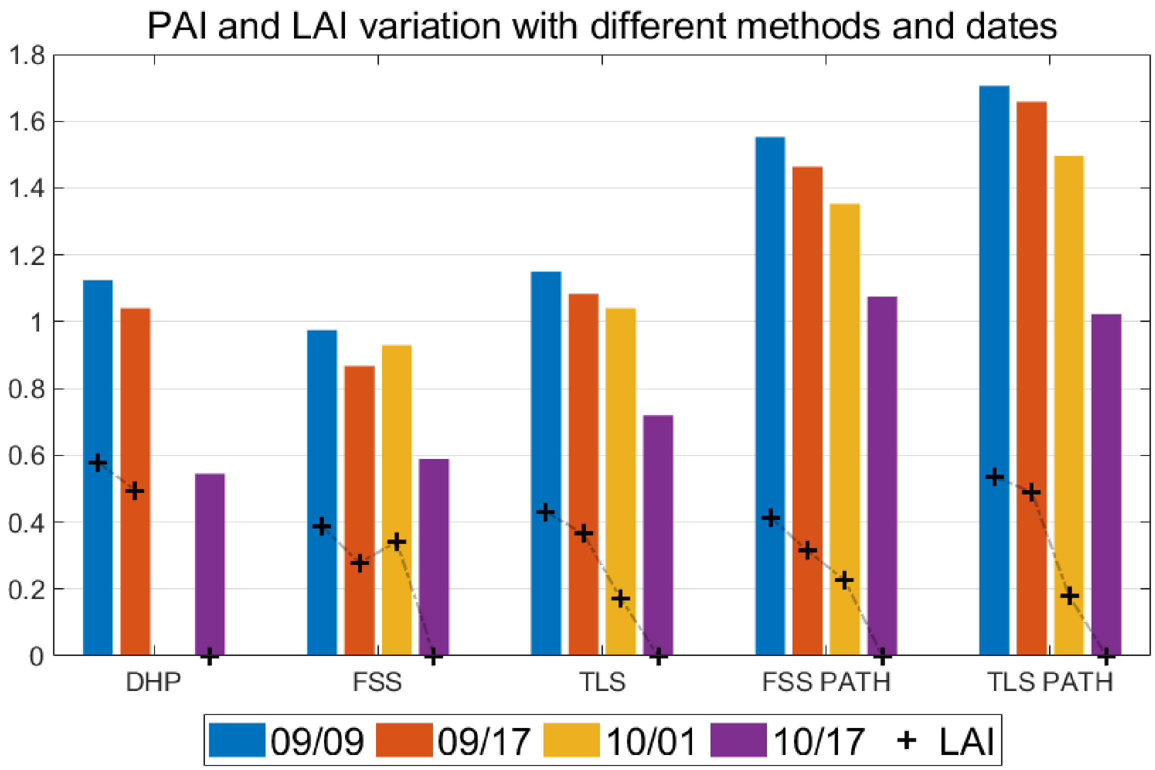


Figure 5.12 Comparing different methods of LAI estimation on four scanning dates. The colored bars stand for PAI, and crosses stand for LAI. The 10/01 DHP dataset is not available. The DHP and the FSS methods are based on RGB images and the TLS based on depth images.

With the PATH model applied to TLS and FSS, the PAI estimates are about 30-45% greater than the non-PATH PAIs. The PATH PAI estimation from FSS does not have the problem of PAI increase on 10/01, indicating the importance of incorporating depth correction. The PATH LAI estimates are also greater than the non-PATH by 14% on average, except for the 10/01 FSS LAI anomaly. Considering the optical image methods usually underestimate true PAI or LAI (Yan et al., 2019) by 20%-60%, it is assumed that the PATH model is a closer approximation of the true PAI or LAI values.

It is important to understand why the PATH model usually outputs higher PAI (or LAI) values than the classic geometrical-optical model. Indeed, the $PAI_{true}(\theta)$ solved from the PATH model does not have a simple analytic form, due to various forms of p_l.

However, if we simply assume p_l is constantly 1, or equivalently, the within-crown path length distribution is uniform, the PATH model then has an analytic solution of PAI, which is basically a Lambert W function of gap fraction $\overline{P(\theta)}$ (Figure 5.13). The traditional LAI model (effective LAI) using Beer's law is also contrasted in the Figure 5.13. It clearly shows that the PATH PAI is consistently greater than non-PATH PAI, especially when gap fraction is small. It is also noteworthy that the PATH model might be overly sensitive to near-zero gap fraction changes. The upper bound of PAI based on the PATH model is $-\frac{\cos(\theta)}{G(\theta)}\frac{l_{max}}{l_{min}}ln\overline{P(\theta)}$ and lower bound is $-\frac{\cos(\theta)}{G(\theta)}\frac{l_{min}}{l_{max}}ln\overline{P(\theta)}$. The wide ranges of PAI indicates strong flexibility of the PATH model, but it is also important for future studies to examine what are the rigorous PAI bounds based on different forms of the p_l functions, and what can be a suitable analytical form of p_l functions or PAI functions with smoother gap fraction sensitivity.

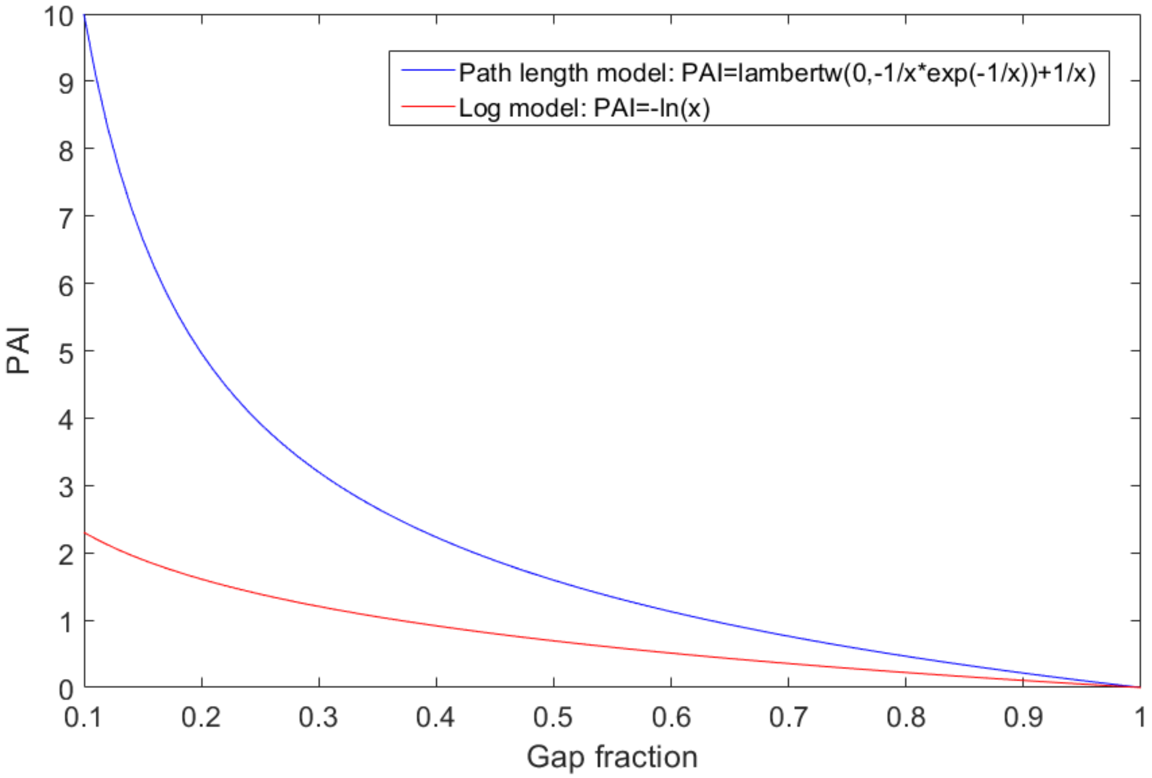


Figure 5.13 Relationship between gap fraction (x) and PAI (or LAI). The red curve shows PAI values from the path length distribution model, compared to the blue curve from the simple Beer's law model.

5.4 Conclusions

Timely monitoring of canopy characteristics is needed to understand the spatiotemporal variation of biomass in a forest ecosystem, and to evaluate carbon budgets as part of forest stand reporting. The advent of low-cost multi-beam LiDAR sensors, Leddar in particular, has presented many successful object tracking applications. Yet the Leddar sensor is still not comparable to TLS in sampling 3D details due to its limited FOV and point resolution. This limitation was mitigated in this study by constructing a low-cost 3D scanning system, FSS, integrating Leddar, camera and pan-tilt robotics. A framework of integration was developed, generally including (1) plane-based physical calibration converting Leddar distance into 3D point and locating Leddar point from images, (2) global image alignment obtaining panorama and coarse camera poses, (3) iterative bundle adjustment optimizing camera poses using both Leddar distance and correspondent pixels at point cloud level, and (4) dense point cloud recovery based on dense matching and interpolation. The calibration error of Leddar points was 9.7 mm at a distance of about 1 m. The set of fusion-based methods was applied to recover hemispherical colored point clouds from multi-temporal poplar tree scans during the autumn defoliation period. The bundle adjustment error was 1-3 pixels, indicating strong agreement between image and Leddar projection on the X-Y plane. Yet great uncertainty exists on the depth (Z) direction due to coarse resolution of Leddar distance. Final fusion-based point clouds were compared to the TLS scans collected on the same spot and date. The volume vertical profiles between TLS and FSS point clouds had an r^2 of 0.5-0.7 over a maximum tree height range, which vary with leaf cover conditions and are higher than the r^2 of 0.3-0.5 between TLS and pure Leddar point clouds. The PAI and LAI metrics were also extracted from FSS, TLS and DHP for leaf-on and leaf-off dates. Using only image data, the PAI and LAI tended to be underestimated with FSS and overestimated with TLS. With point cloud PATH model, both PAI and LAI from FSS or TLS were corrected to approach their true values. In conclusion, by combining both color and depth information, the FSS has demonstrated great versatility and advantages in the application of canopy change monitoring.

The FSS system was mainly developed for a static scanning of environment. For environmental applications such as crown measurement and biomass delineation, low-cost sensor systems such as the FSS cannot parallel TLS and DHP's resolution and precision at present, yet the demand for gross mensuration should not be overlooked and sensor hardware upgrading is always possible. The core contribution of this study is a holistic calibrating and fusing scheme for a lower resolution multi-sensor platform. The advantage of FSS is obvious. It is more portable and low-cost compared to a conventional TLS, and also has a higher frequency, greater detection range and FOV compared to the indoor-oriented LED or flash LiDAR. It is therefore suitable to be deployed in great quantities into sensor networks for broad-scale environmental monitoring. Adapting the FSS to other mobile system such as UAV is also feasible, following a similar framework of calibration, pose approximation, bundle adjustment and densification, except the need for external pose information from GPS and IMU and a dedicated image alignment method. A better viewing geometry such as stereoscopy from a mobile system could greatly improve the precision of 3D recovery especially in the depth direction. A bright future for cost-effective 3D canopy scanning systems is anticipated.

Chapter 6. Portraying the Future of Biomass Analysis and Monitoring

6.1 Thesis Summary

Ground-based LiDAR sensors have been a game changer for biomass research at fine scales and have demonstrated a strong potential for improving the way forests can be inventoried, monitored and evaluated for better forest management. The chapters of this thesis are developed in a progressive order covering a collection of analyses from stem, tree, plot, species, to scalable sites.

Chapter 1 introduced the concepts and challenges of modern forest management and reporting. Biomass delineation plays an important role in filling the gap between forest inventory and forest valuation. Tree biomass measurement accuracy is dependent on volume measurement accuracy. Conventional allometric volume modeling is criticized for its coarseness, whereas promising QSM methods from TLS still calls for validation, automation, generalization and rapidness.

Chapter 2 proposed an automatic stem reconstruction method to provide accurate stem structural metrics for biomass modeling. This includes customized convolution filtering to extract tree locations, constrained region growing and merging to detect stem points, and QSM to quantify stem curves and diameters. The set of automation scripts is not aimed to optimize volume modeling accuracy but to generate descriptive statistics (e.g. DBH and height) integratable within existing biomass modeling frameworks. An early version of the automation scripts, referred to as the UoL method, has been tested in an international benchmarking project (Liang et al., 2018) and a more recent version has been developed into a technical paper (Xi et al., 2016). The core step, constrained region growing, has been developed as an open source program downloadable from https://github.com/truebelief/cylsegment.

Chapter 3 analyzed individual-tree and plot-level wood biomass. The key step is a novel 3D wood point filtering process using emerging deep learning technology, namely FCN, which classifies each point into stem, branch, or non-wood category with a high overall accuracy. Filtered branch points from FCN are further identified to be either part of individual trees, understory, or isolated wood points, based on their nearest distance to stem. Finally, a QSM model is applied to reconstruct both stem and branch forms of individual trees, as well as isolated branches, and understory branches. Stem volume and branch volume are calculated and converted to biomass at plot level. The entire procedure, as published in Xi et al. (2018), depicts a flow of cleaning, parsing, and modeling wood points from complicated plot scans at fine scales, with practical values in systematic biomass resource management.

Chapter 4 probes the possibility of automatic species classification from ground scans to facilitate multi-source biomass analysis. Six different machine learning classifiers are evaluated for wood filtering performance, in contrast with the deep learning method in Chapter 3. The filtered wood points of each tree are input to a point-based deep learning model PointNet++ to classify species. High accuracy is achieved again using the deep learning model, in line with the finding in Chapter 3.

Chapter 5 introduces a prototype of biomass monitoring from ground, a fusion scanning system (FSS) based on a multi-beam Leddar sensor, a webcam and two rotational servos. Methods are designed to calibrate FSS using minimal labor and extended hemispherical scanning field of view. The prototype FSS has been tested to scan poplar trees. Point clouds with both spectral and 3D spatial detail are successfully produced. The advent of compact scanning systems provides new insights into biomass analysis in time-demanding or remote monitoring missions.

6.2 A Future Scalable Biomass Analysis Framework

Ground-based LiDAR and other scanning methods have shown promise for providing accurate tree inventory and biomass estimates. A remaining challenge, however, is the range limitation of ground-based methods. At the scale of landscape, biomass is mostly estimated through other sensing technologies including photography, optical imagery, synthetic aperture radar (SAR) and airborne or spaceborne LiDAR (Koch, 2010). Over decades, landscape-scale biomass has been modeled based on metrics from remote sensors, including spectral bands, vegetation indices, meteorological records, laser return metrics, and radar backscatter coefficients. For example, Boyd (1999) observes that the vegetation reflectance in the middle infrared spectral range (3-5 μm) from satellite imagery is highly sensitive to biomass, probably due to strong spectral absorbance from lignin (wood) and cellulosic concentration (Chen et al., 2010; Xu et al., 2013). Mutanga and Skidmore (2004) further show that narrow spectral bands can overcome the saturation problem of vegetation index (Huete et al., 2002), and are suitable for biomass estimation. Wang et al. (2011) designed the Normalized Dry Matter Index (NDMI), a promising tool to extract biomass from hyperspectral imagery. The L-band of SAR image is found to be predictive of tree age, tree height, diameter at breast height (DBH), and thus biomass (Sun et al., 2002). Airborne laser scanning (ALS), alone or when fused with other sensor data, could provide direct measurement of canopy attributes and outperform optical imaging solutions in high-biomass zones like temperate and tropical forests (Houghton, 2005).

The problem of inferring biomass from remote sensing is obvious: great uncertainty occurs at this broad scale. Among limited studies with accuracy assessments, global biomass estimates can vary by ±25% (Houghton et al., 2009). In the U.S.A, biomass estimation based on MODIS, Landsat imagery and inventory data encounters an average relative error of 74% (Blackard et al., 2008). ALS is considered the most powerful tool to estimate biomass, because rich point cloud metrics can be used to develop allometric models. Gonzalez et al. (2010) compares the uncertainty of biomass calculation using ALS, Quickbird, and field measurements, and reveals that ALS is most consistent and stable with below 1% uncertainty. Yet the relative error between ALS and field mean biomass stays between -18% and -35%. Popescu (2007) displays an RMSE% of 33% in pine biomass estimation from ALS. The RMSE% of ALS biomass based on Hauglin et al. (2013) is 35%, with harvested biomass from 50 Norway spruce trees as reference. Another study by Kankare et al. (2013b) using a similar ALS dataset shows an RMSE% of 36.8% for Norway spruce and 26.3% for Scots pine.

In contrast, a typical RMSE% from TLS studies using QSM models is 10% (Calders et al., 2015; Momo Takoudjou et al., 2018), significantly lower than the typical ALS RMSE% of 30%. The considerable gap between the two scales of biomass accuracies is a limitation for a universal biomass analysis framework. Biomass-related studies integrating TLS and ALS are not rare (Hopkinson et al., 2013a), where role of TLS is basically a proxy of conventional inventory, with most popular integrable metrics being DBH, and height, and crown shape (Chasmer et al., 2006; Hilker et al., 2010; Lovell et al., 2003). Hauglin et al. (2014) points out that replacing simple crown metrics from ALS with the same metrics from TLS leads to a small accuracy increase of 3% in retrieving landscape biomass. Therefore, the success of integrating TLS and ALS for biomass retrieval depends on how to choose useful scalable metrics. The problem of ALS biomass inference can be related to the limited wood-volume metrics at landscape scale. Therefore, a potential solution to maximize integration accuracy is to transfer wood volume information from fine scale (tree or plot) across to landscape scales. Stem and wood volumes are suitable choices for upscaling ALS biomass inference.

An upscaling solution based on fine wood volume metrics is anticipated in the future work. The ground-based approaches presented in this thesis provide a basis for a scalable biomass analysis framework to help forest management. Fulfilling healthy and systematic forest management has particular implications for resource-oriented developing countries to design optimal and united policies on wood trade, employment, carbon reduction and environmental harmony. In near future, it is anticipated that the emerging voxel- or QSM-based virtual forest simulation models (Calders et al., 2018; Côté et al., 2018) become a more popular reference to guide and optimize fine-scale data sampling and spatiospectral reconstruction, which in return would promote the development of simulation models. Point occlusion, currently a major challenge for fine-scale structural modeling, could be successfully overcome by adding scanning views with UAV or other robotic tools (Morsdorf et al., 2018). New ground-based LiDAR sensors and sensor networks would eventually assist fulfillment of long-term and broad-scale biomass sampling and monitoring missions, whereas artificial intelligence and computer vision methods can provide strong support for aligning, extracting, and integrating multi-scale biomass datasets. With the impressive progress of forest sensing tools and analysis methodology, a new quantitative level of systematic forest management and stewardship is on the horizon.

References

Åkerblom, M., Raumonen, P., Casella, E., Disney, M. I., Danson, F. M., Gaulton, R., Schofield, L. A., & Kaasalainen, M. (2018). Non-intersecting leaf insertion algorithm for tree structure models. *Interface Focus, 8*(2), 20170045.

Åkerblom, M., Raumonen, P., Mäkipää, R., & Kaasalainen, M. (2017). Automatic tree species recognition with quantitative structure models. *Remote Sensing of Environment, 191*, 1-12. doi:http://dx.doi.org/10.1016/j.rse.2016.12.002

Armeni, I., Sax, S., Zamir, A. R., & Savarese, S. (2017). Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*.

Arnay, R., Hernández-Aceituno, J., Toledo, J., & Acosta, L. (2018). Laser and Optical Flow Fusion for a Non-Intrusive Obstacle Detection System on an Intelligent Wheelchair. *IEEE Sensors Journal, 18*(9), 3799-3805.

Aschoff, T., & Spiecker, H. (2004). Algorithms for the automatic detection of trees in laser scanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36*(Part 8), W2.

Aschoff, T., Thies, M., & Spiecker, H. (2004). Describing forest stands using terrestrial laser-scanning. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 35*(5), 237-241.

Atanacio-Jiménez, G., González-Barbosa, J.-J., Hurtado-Ramos, J. B., Ornelas-Rodríguez, F. J., Jiménez-Hernández, H., García-Ramirez, T., & González-Barbosa, R. (2011). LIDAR velodyne HDL-64E calibration using pattern planes. *International Journal of Advanced Robotic Systems, 8*(5), 59.

Balmford, A., Bennun, L., Ten Brink, B., Cooper, D., Côté, I. M., Crane, P., Dobson, A., Dudley, N., Dutton, I., & Green, R. E. (2005). The convention on biological diversity's 2010 target. *Science, 307*(5707), 212-213.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). *Surf: Speeded up robust features*. Paper presented at the European conference on computer vision.

Béland, M., Baldocchi, D. D., Widlowski, J.-L., Fournier, R. A., & Verstraete, M. M. (2014). On seeing the wood from the leaves and the role of voxel size in determining leaf area distribution of forests with terrestrial LiDAR. *Agricultural and forest meteorology, 184*(0), 82-97. doi:http://dx.doi.org/10.1016/j.agrformet.2013.09.005

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM, 18*(9), 509-517.

Berger, A., Gschwantner, T., McRoberts, R. E., & Schadauer, K. (2014). Effects of measurement errors on individual tree stem volume estimates for the Austrian National Forest Inventory. *Forest Science, 60*(1), 14-24.

Bienert, A., Scheller, S., Keane, E., Mohan, F., & Nugent, C. (2007). *Tree detection and diameter estimations by analysis of forest terrestrial laserscanner point clouds.* Paper presented at the ISPRS workshop on laser scanning.

Bienert, A., Scheller, S., Keane, E., Mullooly, G., & Mohan, F. (2006). Application of terrestrial laser scanners for the determination of forest inventory parameters. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36*(Part 5).

Bilal, A., Jourabloo, A., Ye, M., Liu, X., & Ren, L. (2018). Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics, 24*(1), 152-162.

Bishop, C. M. (2006). *Pattern recognition and machine learning*: springer.

Blackard, J. A., Finco, M. V., Helmer, E. H., Holden, G. R., Hoppus, M. L., Jacobs, D. M., Lister, A. J., Moisen, G. G., Nelson, M. D., Riemann, R., Ruefenacht, B., Salajanu, D., Weyermann, D. L., Winterberger, K. C., Brandeis, T. J., Czaplewski, R. L., McRoberts, R. E., Patterson, P. L., & Tymcio, R. P. (2008). Mapping U.S. forest biomass using nationwide forest inventory data and moderate resolution information. *Remote Sensing of Environment, 112*(4), 1658-1677. doi:http://dx.doi.org/10.1016/j.rse.2007.08.021

Blanchette, D., Fournier, R. A., Luther, J. E., & Cote, J.-F. (2015). Predicting wood fiber attributes using local-scale metrics from terrestrial LiDAR data: A case study of Newfoundland conifer species. *Forest Ecology and Management, 347*, 116-129.

Bodensteiner, C., Hübner, W., Jüngling, K., Solbrig, P., & Arens, M. (2011). *Monocular camera trajectory optimization using LiDAR data.* Paper presented at the Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on.

Bohren, J., Foote, T., Keller, J., Kushleyev, A., Lee, D., Stewart, A., Vernaza, P., Derenick, J., Spletzer, J., & Satterfield, B. (2008). Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge. *Journal of field robotics, 25*(9), 598-614.

Boudewyn, P., Song, X., Magnussen, S., & Gillis, M. (2007). *Model-based, volume-to-biomass conversion for forested and vegetated land in Canada* (Vol. 411): Pacific Forestry Centre Victoria, BC.

Boulch, A., Saux, B. L., & Audebert, N. (2017). *Unstructured point cloud semantic labeling using deep segmentation networks.* Paper presented at the Eurographics Workshop on 3D Object Retrieval.

Boyd, D. S. (1999). The relationship between the biomass of Cameroonian tropical forests and radiation reflected in middle infrared wavelengths (3.0-5.0 mu m). *International Journal of Remote Sensing, 20*(5), 1017-1023.

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing* (pp. 227-236): Springer.

Brolly, G. B. (2013). *Locating and parameter retrieval of individual trees from terrestrial laser scanner data*. Ph. D. dissertation, University of West Hungary, Sopron, Hungary, 104 p. Online: http://ilex. efe. hu/PhD/emk/brollygabor/disszertacio. pdf,

Brunelli, R., & Poggiot, T. (1997). Template matching: Matched spatial filters and beyond. *Pattern Recognition, 30*(5), 751-768.

Budge, S. E., Badamikar, N. S., & Xie, X. (2014). Automatic registration of fused lidar/digital imagery (texel images) for three-dimensional image creation. *Optical Engineering, 54*(3), 031105-031105. doi:10.1117/1.OE.54.3.031105

Calders, K., Armston, J., Newnham, G., Herold, M., & Goodwin, N. (2014). Implications of sensor configuration and topography on vertical plant profiles derived from terrestrial LiDAR. *Agricultural and forest meteorology, 194*(0), 104-117. doi:http://dx.doi.org/10.1016/j.agrformet.2014.03.022

Calders, K., Newnham, G., Burt, A., Murphy, S., Raumonen, P., Herold, M., Culvenor, D., Avitabile, V., Disney, M., & Armston, J. (2015). Nondestructive estimates of above‐ground biomass using terrestrial laser scanning. *Methods in Ecology and Evolution, 6*(2), 198-208.

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 Sensing, 10*(6), 933.

Campbell, A., Kapos, V., Scharlemann, J., Bubb, P., Chenery, A., Coad, L., Dickson, B., Doswald, N., Khan, M., & Kershaw, F. (2009). *Review of the Literature on the Links between Biodiversity and Climate Change: Impacts, Adaptation and Mitigation.* Paper presented at the Secretariat of the Convention on Biological Diversity, Montreal. Technical Series.

Chasmer, L., Hopkinson, C., & Treitz, P. (2006). Investigating laser pulse penetration through a conifer canopy by integrating airborne and terrestrial lidar. *Canadian Journal of Remote Sensing, 32*(2), 116-125. doi:10.5589/m06-011

Chave, J., Coomes, D., Jansen, S., Lewis, S. L., Swenson, N. G., & Zanne, A. E. (2009). Towards a worldwide wood economics spectrum. *Ecology Letters, 12*(4), 351-366.

Chen, H., Ferrari, C., Angiuli, M., Yao, J., Raspi, C., & Bramanti, E. (2010). Qualitative and quantitative analysis of wood samples by Fourier transform infrared spectroscopy and multivariate analysis. *Carbohydrate polymers, 82*(3), 772-778.

Chen, Y., Zhang, W., Hu, R., Qi, J., Shao, J., Li, D., Wan, P., Qiao, C., Shen, A., & Yan, G. (2018). Estimation of forest leaf area index using terrestrial laser scanning data and path length distribution model in open-canopy forests. *Agricultural and forest meteorology, 263*, 323-333. doi:https://doi.org/10.1016/j.agrformet.2018.09.006

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). *3D U-Net: learning dense volumetric segmentation from sparse annotation.* Paper presented at the International Conference on Medical Image Computing and Computer-Assisted Intervention.

Colgan, M. S., Asner, G. P., & Swemmer, T. (2012). Harvesting tree biomass at the stand level to assess the accuracy of field and airborne biomass estimation in savannas. *Ecological Applications, 23*(5), 1170-1184. doi:10.1890/12-0922.1

COP11, U. (2005). *Report of the Conference of the Parties on its eleventh session, Part One: Proceedings*.

Côté, J.-F., Fournier, R. A., & Egli, R. (2011). An architectural model of trees to estimate forest structural attributes using terrestrial LiDAR. *Environmental Modelling & Software, 26*(6), 761-777. doi:http://dx.doi.org/10.1016/j.envsoft.2010.12.008

Côté, J.-F., Fournier, R. A., Frazer, G. W., & Olaf Niemann, K. (2012). A fine-scale architectural model of trees to enhance LiDAR-derived measurements of forest canopy structure. *Agricultural and forest meteorology, 166–167*(0), 72-85. doi:http://dx.doi.org/10.1016/j.agrformet.2012.06.007

Côté, J.-F., Fournier, R. A., Luther, J. E., & van Lier, O. R. (2018). Fine-scale three-dimensional modeling of boreal forest plots to improve forest characterization with remote sensing. *Remote Sensing of Environment, 219*, 99-114. doi:https://doi.org/10.1016/j.rse.2018.09.026

Côté, J.-F., Widlowski, J.-L., Fournier, R. A., & Verstraete, M. M. (2009). The structural and radiative consistency of three-dimensional tree reconstructions from terrestrial lidar. *Remote Sensing of Environment, 113*(5), 1067-1081. doi:http://dx.doi.org/10.1016/j.rse.2009.01.017

Danson, F. M., Gaulton, R., Armitage, R. P., Disney, M., Gunawan, O., Lewis, P., Pearson, G., & Ramirez, A. F. (2014). Developing a dual-wavelength full-waveform terrestrial laser scanner to characterize forest canopy structure. *Agricultural and forest meteorology, 198*, 7-14.

Das, N. (2014). Modeling Develops to Estimate Leaf Area and Leaf Biomass of Lagerstroemia speciosa in West Vanugach Reserve Forest of Bangladesh. *ISRN Forestry, 2014*, 9. doi:10.1155/2014/486478

Dassot, M., Colin, A., Santenoise, P., Fournier, M., & Constant, T. (2012). Terrestrial laser scanning for measuring the solid wood volume, including branches, of adult standing trees in the forest environment. *Computers and Electronics in Agriculture*, *89*, 86-93. doi:http://dx.doi.org/10.1016/j.compag.2012.08.005

Dassot, M., Constant, T., & Fournier, M. (2011). The use of terrestrial LiDAR technology in forest science: application fields, benefits and challenges. *Annals of forest science, 68*(5), 959-974.

De Silva, V., Roche, J., & Kondoz, A. (2018). Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots. *Sensors*, *18*(8), 2730.

de Wit, C. T. (1965). *Photosynthesis of leaf canopies*. Retrieved from

Debattisti, S., Mazzei, L., & Panciroli, M. (2013). *Automated extrinsic laser and camera inter-calibration using triangular targets.* Paper presented at the Intelligent Vehicles Symposium (IV), 2013 IEEE.

Disney, M. I., Boni Vicari, M., Burt, A., Calders, K., Lewis, S. L., Raumonen, P., & Wilkes, P. (2018). Weighing trees with lasers: advances, challenges and opportunities. *Interface Focus*, *8*(2), 20170048.

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., & Frenkel, A. (2011). *On the segmentation of 3D LIDAR point clouds.* Paper presented at the Robotics and Automation (ICRA), 2011 IEEE International Conference on.

Ducey, M. J., & Astrup, R. (2013). Adjusting for nondetection in forest inventories derived from terrestrial laser scanning. *Canadian Journal of Remote Sensing*, *39*(5), 410-425. doi:10.5589/m13-048

Eitel, J. U., Vierling, L. A., & Magney, T. S. (2013). A lightweight, low cost autonomously operating terrestrial laser scanner for quantifying and monitoring ecosystem structural dynamics. *Agricultural and forest meteorology, 180*, 86-96.

Elaksher, A. F., Bhandari, S., Carreon-Limones, C. A., & Lauf, R. (2017). *Potential of UAV lidar systems for geospatial mapping.* Paper presented at the Lidar Remote Sensing for Environmental Monitoring 2017.

Eysn, L., Pfeifer, N., Ressl, C., Hollaus, M., Grafl, A., & Morsdorf, F. (2013). A Practical Approach for Extracting Tree Models in Forest Environments Based on Equirectangular Projections of Terrestrial Laser Scans. *Remote Sensing, 5*(11), 5424.

Falster, D. S., & Westoby, M. (2003). Leaf size and angle vary widely across species: what consequences for light interception? *New Phytologist*, *158*(3), 509-525. doi:10.1046/j.1469-8137.2003.00765.x

Ferrara, R., Virdis, S. G. P., Ventura, A., Ghisu, T., Duce, P., & Pellizzaro, G. (2018). An automated approach for wood-leaf separation from terrestrial LIDAR point clouds using the density based clustering algorithm DBSCAN. *Agricultural and forest meteorology, 262*, 434-444. doi:https://doi.org/10.1016/j.agrformet.2018.04.008

Fitzgibbon, A. W., Pilu, M., & Fisher, R. B. (1996, 25-29 Aug. 1996). *Direct least squares fitting of ellipses*. Paper presented at the Proceedings of 13th International Conference on Pattern Recognition.

Foody, G. (2003). Remote sensing of tropical forest environments: towards the monitoring of environmental resources for sustainable development. *International Journal of Remote Sensing, 24*(20), 4035-4046.

Fremont, V., & Bonnifait, P. (2008). *Extrinsic calibration between a multi-layer lidar and a camera*. Paper presented at the Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on.

Gander, W., Golub, G. H., & Strebel, R. (1994). Least-squares fitting of circles and ellipses. *BIT Numerical Mathematics, 34*(4), 558-578.

Gangadharan, S., Burks, T. F., & Schueller, J. K. (2019). A comparison of approaches for citrus canopy profile generation using ultrasonic and Leddar® sensors. *Computers and Electronics in Agriculture, 156*, 71-83.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv preprint arXiv:1704.06857*.

Gillis, M., Omule, A., & Brierley, T. (2005). Monitoring Canada's forests: the national forest inventory. *The Forestry Chronicle, 81*(2), 214-221.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Godejord, B. (2018). *Characterization of a commercial LIDAR module for use in camera triggering system*. NTNU,

Gonzalez de Tanago, J., Lau, A., Bartholomeus, H., Herold, M., Avitabile, V., Raumonen, P., Martius, C., Goodman, R. C., Disney, M., Manuri, S., Burt, A., & Calders, K. (2018). Estimation of above-ground biomass of large tropical trees with terrestrial LiDAR. *Methods in Ecology and Evolution, 9*(2), 223-234. doi:doi:10.1111/2041-210X.12904

Gonzalez, P., Asner, G. P., Battles, J. J., Lefsky, M. A., Waring, K. M., & Palace, M. (2010). Forest carbon densities and uncertainties from Lidar, QuickBird, and field measurements in California. *Remote Sensing of Environment, 114*(7), 1561-1575. doi:http://dx.doi.org/10.1016/j.rse.2010.02.011

Graham, B. (2015). Sparse 3D convolutional neural networks. *arXiv preprint arXiv:1505.02890*.

Hackenberg, J., Morhart, C., Sheppard, J., Spiecker, H., & Disney, M. (2014). Highly Accurate Tree Models Derived from Terrestrial Laser Scan Data: A Method Description. *Forests, 5*(5), 1069-1105.

Hackenberg, J., Wassenberg, M., Spiecker, H., & Sun, D. (2015). Non Destructive Method for Biomass Prediction Combining TLS Derived Tree Volume and Wood Density. *Forests, 6*(4), 1274-1300.

Hancock, S., Essery, R., Reid, T., Carle, J., Baxter, R., Rutter, N., & Huntley, B. (2014). Characterising forest gap fraction with terrestrial lidar and photography: An examination of relative limitations. *Agricultural and forest meteorology, 189*, 105-114.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*: Cambridge university press.

Hauglin, M., Dibdiakova, J., Gobakken, T., & Næsset, E. (2013). Estimating single-tree branch biomass of Norway spruce by airborne laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing, 79*, 147-156.

Hauglin, M., Gobakken, T., Astrup, R., Ene, L., & Næsset, E. (2014). Estimating single-tree crown biomass of Norway spruce by airborne laser scanning: A comparison of methods with and without the use of terrestrial laser scanning to obtain the ground reference data. *Forests, 5*(3), 384-403.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Heath, L. S., Hansen, M., Smith, J. E., Miles, P. D., & Smith, B. W. (2009). Investigation into calculating tree biomass and carbon in the FIADB using a biomass expansion factor approach.

Henning, J. G., & Radtke, P. J. (2006). Detailed Stem Measurements of Standing Trees from Ground-Based Scanning Lidar. *Forest Science, 52*(1), 67-80.

Hentschke, M., Pignaton de Freitas, E., Hennig, C., & Girardi da Veiga, I. (2018). Evaluation of Altitude Sensors for a Crop Spraying Drone. *Drones, 2*(3), 25.

Hilker, T., van Leeuwen, M., Coops, N. C., Wulder, M. A., Newnham, G. J., Jupp, D. L., & Culvenor, D. S. (2010). Comparing canopy metrics derived from terrestrial and airborne laser scanning in a Douglas-fir dominated forest stand. *Trees, 24*(5), 819-832.

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55-67.

Holmgren, J., & Persson, Å. (2004). Identifying species of individual trees using airborne laser scanner. *Remote Sensing of Environment, 90*(4), 415-423. doi:http://dx.doi.org/10.1016/S0034-4257(03)00140-8

Hopkinson, C., Chasmer, L., Barr, A. G., Kljun, N., Black, T. A., & McCaughey, J. H. (2016). Monitoring boreal forest biomass and carbon storage change by integrating airborne laser scanning, biometry and eddy covariance data. *Remote Sensing of Environment, 181*, 82-95. doi:https://doi.org/10.1016/j.rse.2016.04.010

Hopkinson, C., Chasmer, L., Colville, D., Fournier, R. A., Hall, R. J., Luther, J. E., Milne, T., Petrone, R. M., & St-Onge, B. (2013a). Moving Toward Consistent ALS Monitoring of Forest Attributes across Canada. *Photogrammetric Engineering & Remote Sensing, 79*(2), 159-173.

Hopkinson, C., Chasmer, L., & Hall, R. (2008). The uncertainty in conifer plantation growth prediction from multi-temporal lidar datasets. *Remote Sensing of Environment, 112*(3), 1168-1180.

Hopkinson, C., Chasmer, L., Young-Pow, C., & Treitz, P. (2004). Assessing forest metrics with a ground-based scanning lidar. *Canadian Journal of Forest Research, 34*(3), 573-583. doi:10.1139/x03-225

Hopkinson, C., Lovell, J., Chasmer, L., Jupp, D., Kljun, N., & van Gorsel, E. (2013b). Integrating terrestrial and airborne lidar to calibrate a 3D canopy model of effective leaf area index. *Remote Sensing of Environment, 136*, 301-314.

Houghton, R. (2005). Aboveground forest biomass and the global carbon balance. *Global Change Biology, 11*(6), 945-958.

Houghton, R., Hall, F., & Goetz, S. J. (2009). Importance of biomass in the global carbon cycle. *Journal of Geophysical Research: Biogeosciences (2005–2012), 114*(G2).

Hu, R., Bournez, E., Cheng, S., Jiang, H., Nerry, F., Landes, T., Saudreau, M., Kastendeuch, P., Najjar, G., Colin, J., & Yan, G. (2018). Estimating the leaf area of an individual tree in urban areas using terrestrial laser scanner and path length distribution model. *ISPRS Journal of Photogrammetry and Remote Sensing, 144*, 357-368. doi:https://doi.org/10.1016/j.isprsjprs.2018.07.015

Hu, R., Yan, G., Mu, X., & Luo, J. (2014). Indirect measurement of leaf area index on the basis of path length distribution. *Remote Sensing of Environment, 155*, 239-247. doi:https://doi.org/10.1016/j.rse.2014.08.032

Huang, H., Gong, P., Cheng, X., Clinton, N., Cao, C., Ni, W., Li, Z., & Wang, L. (2008). Forest structural parameter extraction using terrestrial LiDAR. *American Geophysical Union, University of Arizona, USA.*

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. *Photogrammetric Engineering & Remote Sensing, 77*(3), 219-227.

Huang, J., & You, S. (2016). *Point cloud labeling using 3d convolutional neural network.* Paper presented at the Pattern Recognition (ICPR), 2016 23rd International Conference on.

Huber, P. J. (1981). *Robust statistics*. New York: Wiley.

Huete, A., Didan, K., Miura, T., Rodriguez, E. P., Gao, X., & Ferreira, L. G. (2002). Overview of the radiometric and biophysical performance of the MODIS vegetation indices. *Remote Sensing of Environment, 83*(1–2), 195-213. doi:http://dx.doi.org/10.1016/S0034-4257(02)00096-2

Hyyppa, J., & Liang, X. (2013). Project Benchmarking on Terrestrial Laser Scanning for Forestry Applications. Retrieved from http://www.eurosdr.net/research/project/project-benchmarking-terrestrial-laser-scanning-forestry-applications

Jenkins, J. C., Chojnacky, D. C., Heath, L. S., & Birdsey, R. A. (2003). National-scale biomass estimators for United States tree species. *Forest Science, 49*(1), 12-35.

Jia, Q., Fan, X., Luo, Z., Song, L., & Qiu, T. (2017). A fast ellipse detector using projective invariant pruning. *IEEE Transactions on Image Processing, 26*(8), 3665-3679.

Kankare, V., Holopainen, M., Vastaranta, M., Puttonen, E., Yu, X., Hyyppä, J., Vaaja, M., Hyyppä, H., & Alho, P. (2013a). Individual tree biomass estimation using terrestrial laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing, 75*(0), 64-75. doi:http://dx.doi.org/10.1016/j.isprsjprs.2012.10.003

Kankare, V., Räty, M., Yu, X., Holopainen, M., Vastaranta, M., Kantola, T., Hyyppä, J., Hyyppä, H., Alho, P., & Viitala, R. (2013b). Single tree biomass modelling using airborne laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing, 85*, 66-73. doi:http://dx.doi.org/10.1016/j.isprsjprs.2013.08.008

Kim, J., Ryu, Y., Jiang, C., & Hwang, Y. (2019). Continuous observation of vegetation canopy dynamics using an integrated low-cost, near-surface remote sensing system. *Agricultural and forest meteorology, 264*, 164-177. doi:https://doi.org/10.1016/j.agrformet.2018.09.014

Kim, S., McGaughey, R. J., Andersen, H.-E., & Schreuder, G. (2009). Tree species differentiation using intensity data derived from leaf-on and leaf-off airborne laser scanner data. *Remote Sensing of Environment, 113*(8), 1575-1586. doi:http://dx.doi.org/10.1016/j.rse.2009.03.017

Kimmins, J. (1997). Forest ecology: a foundation for sustainable forest management. *Prentice Hall, New Jersey*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klokov, R., & Lempitsky, V. (2017). *Escape from cells: Deep kd-networks for the recognition of 3d point cloud models.* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision.url: https://github.com/Regenerator/kdnets

Kobler, A., Pfeifer, N., Ogrinc, P., Todorovski, L., Oštir, K., & Džeroski, S. (2007). Repetitive interpolation: A robust algorithm for DTM generation from Aerial Laser Scanner Data in forested terrain. *Remote Sensing of Environment, 108*(1), 9-23. doi:http://dx.doi.org/10.1016/j.rse.2006.10.013

Koch, B. (2010). Status and future of laser scanning, synthetic aperture radar and hyperspectral remote sensing data for forest biomass assessment. *ISPRS Journal of Photogrammetry and Remote Sensing, 65*(6), 581-590. doi:http://dx.doi.org/10.1016/j.isprsjprs.2010.09.001

Koci, J., Jarihani, B., Leon, J. X., Sidle, R., Wilkinson, S., & Bartley, R. (2017). Assessment of UAV and ground-based Structure from Motion with multi-view stereo photogrammetry in a gullied savanna catchment. *ISPRS International Journal of Geo-Information, 6*(11), 328.

Kraus, K., & Pfeifer, N. (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing, 53*(4), 193-203. doi:http://dx.doi.org/10.1016/S0924-2716(98)00009-4

Kublin, E., Breidenbach, J., & Kändler, G. (2013). A flexible stem taper and volume prediction method based on mixed-effects B-spline regression. *European Journal of Forest Research, 132*(5-6), 983-997.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). New York: Springer.

Kunz, M., Hess, C., Raumonen, P., Bienert, A., Hackenberg, J., Maas, H., Härdtle, W., Fichtner, A., & Von Oheimb, G. (2017). Comparison of wood volume estimates of young trees from terrestrial laser scan data. *iForest: Biogeosciences and Forestry, 10*(2), 451-458.

Lal, R. (2008). Sequestration of atmospheric CO 2 in global carbon pools. *Energy & Environmental Science, 1*(1), 86-100.

Lambert, M., Ung, C., & Raulier, F. (2005). Canadian national tree aboveground biomass equations. *Canadian Journal of Forest Research, 35*(8), 1996-2018.

Lang, A. (1987). Simplified estimate of leaf area index from transmittance of the sun's beam. *Agricultural and forest meteorology, 41*(3-4), 179-186.

Lang, A., & Xiang, Y. (1986). Estimation of leaf area index from transmission of direct sunlight in discontinuous canopies. *Agricultural and forest meteorology, 37*(3), 229-243.

Lau, A., Bentley, L. P., Martius, C., Shenkin, A., Bartholomeus, H., Raumonen, P., Malhi, Y., Jackson, T., & Herold, M. (2018). Quantifying branch architecture of tropical trees using terrestrial LiDAR and 3D modelling. *Trees, 32*(5), 1219-1231. doi:10.1007/s00468-018-1704-1

Lauri, P., Havlík, P., Kindermann, G., Forsell, N., Böttcher, H., & Obersteiner, M. (2014). Woody biomass energy potential in 2050. *Energy Policy, 66,* 19-31. doi:http://dx.doi.org/10.1016/j.enpol.2013.11.033

Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S., & Felsberg, M. (2017). Deep Projective 3D Semantic Segmentation. *arXiv preprint arXiv:1705.03428*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.

Lefsky, M. A., Harding, D., Cohen, W. B., Parker, G., & Shugart, H. H. (1999). Surface Lidar Remote Sensing of Basal Area and Biomass in Deciduous Forests of Eastern Maryland, USA. *Remote Sensing of Environment, 67*(1), 83-98. doi:http://dx.doi.org/10.1016/S0034-4257(98)00071-6

Levinson, J., & Thrun, S. (2011). *Unsupervised Calibration for Multi-beam Lasers*. Paper presented at the Experimental Robotics: The 12th International Symposium on Experimental Robotics.

Li, J., Chen, B. M., & Hee Lee, G. (2018a). *So-net: Self-organizing network for point cloud analysis*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2018b). Feature selection: A data perspective. *ACM Computing Surveys (CSUR), 50*(6), 94.

Li, J., He, X., & Li, J. (2015). *2D LiDAR and camera fusion in 3D modeling of indoor environment*. Paper presented at the Aerospace and Electronics Conference (NAECON), 2015 National.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018c). *PointCNN: Convolution On X-Transformed Points*. Paper presented at the Advances in Neural Information Processing Systems.

Li, Y., Su, Y., Hu, T., Xu, G., & Guo, Q. (2018d). Retrieving 2-D leaf angle distributions for deciduous trees from terrestrial laser scanner data. *IEEE Transactions on Geoscience and Remote Sensing, 56*(8), 4945-4955.

Liang, X., & Hyyppä, J. (2013). Automatic stem mapping by merging several terrestrial laser scans at the feature and decision levels. *Sensors, 13*(2), 1614-1634.

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. (2018). International benchmarking of terrestrial laser scanning approaches for forest inventories. *ISPRS Journal of Photogrammetry and Remote Sensing, 144*, 137-179. doi:https://doi.org/10.1016/j.isprsjprs.2018.06.021

Liang, X., Hyyppa, J., Kukko, A., Kaartinen, H., Jaakkola, A., & Yu, X. (2014a). The use of a mobile laser scanning system for mapping large forest plots.

Liang, X., Hyyppä, J., Kukko, A., Kaartinen, H., Jaakkola, A., & Yu, X. (2014b). The use of a mobile laser scanning system for mapping large forest plots. *IEEE Geoscience and Remote Sensing Letters, 11*(9), 1504-1508.

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 Journal of Photogrammetry and Remote Sensing, 115*, 63-77. doi:http://dx.doi.org/10.1016/j.isprsjprs.2016.01.006

Liang, X., Kankare, V., Yu, X., Hyyppä, J., & Holopainen, M. (2014c). Automated stem curve measurement using terrestrial laser scanning. *IEEE Transactions on Geoscience and Remote Sensing, 52*(3), 1739-1748.

Liang, X., Litkey, P., Hyyppa, J., Kaartinen, H., Vastaranta, M., & Holopainen, M. (2012). Automatic Stem Mapping Using Single-Scan Terrestrial Laser Scanning. *IEEE Transactions on Geoscience and Remote Sensing, 50*(2), 661-670. doi:10.1109/TGRS.2011.2161613

Lin, Y., & Herold, M. (2016). Tree species classification based on explicit tree structure feature parameters derived from static terrestrial laser scanning data. *Agricultural and forest meteorology, 216*, 105-114. doi:https://doi.org/10.1016/j.agrformet.2015.10.008

Lindberg, E., Holmgren, J., Olofsson, K., & Olsson, H. (2012). Estimation of stem attributes using a combination of terrestrial and airborne laser scanning. *European Journal of Forest Research, 131*(6), 1917-1931.

Litkey, P., Liang, X., Kaartinen, H., Hyyppä, J., Kukko, A., Holopainen, M., Hill, R., Rosette, J., & Suárez, J. (2008). *Single-scan TLS methods for forest parameter retrieval.* Paper presented at the Proceedings of SilviLaser 2008, 8th international conference on LiDAR applications in forest assessment and inventory, Heriot-Watt University, Edinburgh, UK, 17-19 September, 2008.

Litkey, P., Puttonen, E., & Liang, X. (2011). Comparison of point cloud data reduction methods in single-scan TLS for finding tree stems in forest. *Proceedings of SilviLaser*.

Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., & El-Sana, J. (2010). *Automatic reconstruction of tree skeletal structures from point clouds.* Paper presented at the ACM Transactions on Graphics (TOG).

Loh, W.-Y. (2002). Regression tress with unbiased variable selection and interaction detection. *Statistica Sinica*, 361-386.

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully convolutional networks for semantic segmentation.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Lovell, J., Jupp, D., Newnham, G., & Culvenor, D. (2011). Measuring tree stem diameters using intensity profiles from ground-based scanning lidar from a fixed viewpoint. *ISPRS Journal of Photogrammetry and Remote Sensing, 66*(1), 46-55.

Lovell, J. L., Jupp, D. L. B., Culvenor, D. S., & Coops, N. C. (2003). Using airborne and ground-based ranging lidar to measure canopy structure in Australian forests. *Canadian Journal of Remote Sensing, 29*(5), 607-622. doi:10.5589/m03-026

Luther, J. E., Skinner, R., Fournier, R. A., van Lier, O. R., Bowers, W. W., Coté, J.-F., Hopkinson, C., & Moulton, T. (2013). Predicting wood quantity and quality attributes of balsam fir and black spruce using airborne laser scanner data. *Forestry*, cpt039.

Ma, L., Zheng, G., Eitel, J. U., Moskal, L. M., He, W., & Huang, H. (2016a). Improved salient feature-based approach for automatically separating photosynthetic and nonphotosynthetic components within terrestrial lidar point cloud data of forest canopies. *IEEE Transactions on Geoscience and Remote Sensing, 54*(2), 679-696.

Ma, L., Zheng, G., Eitel, J. U. H., Magney, T. S., & Moskal, L. M. (2016b). Determining woody-to-total area ratio using terrestrial laser scanning (TLS). *Agricultural and forest meteorology, 228-229*, 217-228. doi:https://doi.org/10.1016/j.agrformet.2016.06.021

Maas, H. G., Bienert, A., Scheller, S., & Keane, E. (2008). Automatic forest inventory parameter determination from terrestrial laser scanner data. *International Journal of Remote Sensing, 29*(5), 1579-1593. doi:10.1080/01431160701736406

Malhi, Y., Jackson, T., Patrick Bentley, L., Lau, A., Shenkin, A., Herold, M., Calders, K., Bartholomeus, H., & Disney, M. I. (2018). New perspectives on the ecology of tree structure and tree communities through terrestrial laser scanning. *Interface Focus, 8*(2), 20170052.

Masutani, Y., Masamune, K., & Dohi, T. (1996). Region-growing based feature extraction algorithm for tree-like objects. In K. H. Höhne & R. Kikinis (Eds.), *Visualization in Biomedical Computing: 4th International Conference, VBC'96 Hamburg, Germamy, September 22–25, 1996 Proceedings* (pp. 159-171). Berlin, Heidelberg: Springer Berlin Heidelberg.

Meng, X., Currit, N., & Zhao, K. (2010). Ground Filtering Algorithms for Airborne LiDAR Data: A Review of Critical Issues. *Remote Sensing, 2*(3), 833.

Miles, L., & Kapos, V. (2008). Reducing greenhouse gas emissions from deforestation and forest degradation: global land-use implications. *Science, 320*(5882), 1454-1455.

Mimeault, Y. (2014). Parking management system and method using lighting system. In: Google Patents.

Mimeault, Y., & Cantin, D. (2013). Lighting system with driver assistance capabilities. In: Google Patents.

Mokany, K., Raison, R. J., & Prokushkin, A. (2005). *Critical analysis of root: Shoot ratios in terrestrial biomes* (Vol. 12).

Momo Takoudjou, S., Ploton, P., Sonké, B., Hackenberg, J., Griffon, S., de Coligny, F., Kamdem, N. G., Libalah, M., Mofack, G. I., Le Moguédec, G., Pélissier, R., & Barbier, N. (2018). Using terrestrial laser scanning data to estimate large tropical trees biomass and calibrate allometric models: A comparison with traditional destructive approach. *Methods in Ecology and Evolution, 9*(4), 905-916. doi:doi:10.1111/2041-210X.12933

Morrison, R., Low, B., Kucera, H., & Kucera, G. (1999). *A spatial data warehouse for NFIS.* Paper presented at the Proceedings of the 13th Annual Conference on Geographic Information Systems.

Morsdorf, F., Kükenbrink, D., Schneider, F., Abegg, M., & Schaepman, M. (2018). Close-range laser scanning in forests: towards physically based semantics across scales. *Interface Focus, 8*(2), 20170046.

Muhammad, N., & Lacroix, S. (2010). *Calibration of a rotating multi-beam lidar.* Paper presented at the Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on.

Mutanga, O., & Skidmore, A. K. (2004). Narrow band vegetation indices overcome the saturation problem in biomass estimation. *International Journal of Remote Sensing, 25*(19), 3999-4014.

Natural Resources Canada. (2018). *The State of Canada's Forests*. Ottawa: Canadian Forest Service.

Newnham, G. J., Armston, J. D., Calders, K., Disney, M. I., Lovell, J. L., Schaaf, C. B., Strahler, A. H., & Danson, F. M. (2015). Terrestrial Laser Scanning for Plot-Scale Forest Measurement. *Current Forestry Reports, 1*(4), 239-251. doi:10.1007/s40725-015-0025-5

Newton, I. (1711). *De analysi per aequationes numero terminorum infinitas*.

Newton, I., & Colson, J. (1736). *The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines... Translated from the Author's Latin Original Not Yet Made Publick. To which is Subjoin'd a Perpetual Comment Upon the Whole Work... by J. Colson*.

Nixon, M. (2008). *Feature extraction & image processing*: Academic Press.

Nock, C. A., Taugourdeau, O., Delagrange, S., & Messier, C. (2013). Assessing the potential of low-cost 3D cameras for the rapid measurement of plant woody structure. *Sensors (Basel, Switzerland), 13*(12), 16216-16233. doi:10.3390/s131216216

Nock, R., & Nielsen, F. (2004). Statistical region merging. *IEEE transactions on pattern analysis and machine intelligence, 26*(11), 1452-1458.

Olivier, P. (2015). *Leddar Optical Time-of-Flight Sensing Technology*. Quebec City, Quebec,.

Ørka, H. O., Næsset, E., & Bollandsås, O. M. (2009). Classifying species of individual trees by intensity and structure features derived from airborne laser scanner data. *Remote Sensing of Environment*, *113*(6), 1163-1174. doi:http://dx.doi.org/10.1016/j.rse.2009.02.002

Othmani, A., Voon, L. F. C. L. Y., Stolz, C., & Piboule, A. (2013). Single tree species classification from Terrestrial Laser Scanning data for forest inventory. *Pattern Recogn. Lett., 34*(16), 2144-2150. doi:10.1016/j.patrec.2013.08.004

Park, Y., Yun, S., Won, C. S., Cho, K., Um, K., & Sim, S. (2014). Calibration between color camera and 3D LIDAR instruments with a polygonal planar board. *Sensors, 14*(3), 5333-5353.

Peuhkurinen, J., Maltamo, M., Malinen, J., Pitkänen, J., & Packalén, P. (2007). Preharvest measurement of marked stands using airborne laser scanning. *Forest Science, 53*(6), 653-661.

Pfeifer, N., & Winterhalder, D. (2004). Modelling of tree cross sections from terrestrial laser scanning data with free-form curves. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36*(Part 8), W2.

Poorter, H., Niklas, K. J., Reich, P. B., Oleksyn, J., Poot, P., & Mommer, L. (2012). Biomass allocation to leaves, stems and roots: meta‐analyses of interspecific variation and environmental control. *New Phytologist, 193*(1), 30-50. doi:10.1111/j.1469-8137.2011.03952.x

Popescu, S. C. (2007). Estimating biomass of individual pine trees using airborne lidar. *Biomass and Bioenergy, 31*(9), 646-655. doi:http://dx.doi.org/10.1016/j.biombioe.2007.06.022

Pretzsch, H. (2009). Forest dynamics, growth, and yield. In *Forest Dynamics, Growth and Yield* (pp. 1-39): Springer.

Pueschel, P. (2013). The influence of scanner parameters on the extraction of tree metrics from FARO Photon 120 terrestrial laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing, 78*, 58-68. doi:http://dx.doi.org/10.1016/j.isprsjprs.2013.01.005

Pueschel, P., Newnham, G., Rock, G., Udelhoven, T., Werner, W., & Hill, J. (2013). The influence of scan mode and circle fitting on tree stem detection, stem diameter and volume extraction from terrestrial laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing, 77*, 44-56. doi:http://dx.doi.org/10.1016/j.isprsjprs.2012.12.001

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2016). Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*. url: https://github.com/charlesq34/pointnet

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv preprint arXiv:1706.02413*. url: https://github.com/charlesq34/pointnet2

Raumonen, P., Casella, E., Calders, K., Murphy, S., Åkerblom, M., & Kaasalainen, M. (2015). Massive-scale tree modelling from TLS data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2*(3), 189.

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 Sensing, 5*(2), 491-520.

Ravaglia, J., Bac, A., & Fournier, R. A. (2017). Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data. *Computers & Graphics, 66*, 23-33.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards real-time object detection with region proposal networks.* Paper presented at the Advances in neural information processing systems.

Rice, B., Weiskittel, A. R., & Wagner, R. G. (2014). Efficiency of alternative forest inventory methods in partially harvested stands. *European Journal of Forest Research, 133*(2), 261-272.

Riegler, G., Osman Ulusoy, A., & Geiger, A. (2017). *Octnet: Learning deep 3d representations at high resolutions.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Roggero, M. (2002). Object segmentation with region growing and principal component analysis. *International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences, 34*(3/A), 289-294.

Ross, J. (1981). *The radiation regime and architecture of plant stands*. The Hague: Junk Publishers.

Schilling, A., & Maas, H. (2014). Automatic reconstruction of skeletal structures from TLS forest scenes. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2*(5), 321.

Schilling, A., Schmidt, A., & Maas, H.-G. (2011). *Automatic Tree Detection and Diameter Estimation in Terrestrial Laser Scanner Point Clouds.* Paper presented at the Proceedings of the 16th Computer Vision Winter Workshop.

Schleppi, P., Conedera, M., Sedivy, I., & Thimonier, A. (2007). Correcting non-linearity and slope effects in the estimation of the leaf area index of forests from hemispherical photographs. *Agricultural and forest meteorology, 144*(3-4), 236-242.

Schneider, F. D., Kükenbrink, D., Schaepman, M. E., Schimel, D. S., & Morsdorf, F. (2019). Quantifying 3D structure and occlusion in dense tropical and temperate forests using close-range LiDAR. *Agricultural and forest meteorology, 268*, 249-257. doi:https://doi.org/10.1016/j.agrformet.2019.01.033

Sheehan, M., Harrison, A., & Newman, P. (2012). Self-calibration for a 3D laser. *The International Journal of Robotics Research, 31*(5), 675-687.

Shettles, M., Hilker, T., & Temesgen, H. (2016). Examination of uncertainty in per unit area estimates of aboveground biomass using terrestrial LiDAR and ground data. *Canadian Journal of Forest Research, 46*(5), 706-715.

Sorkine, O. (2006). *Laplacian mesh processing.* (Doctoral), Tel Aviv University,

Srinivasan, S., Popescu, S. C., Eriksson, M., Sheridan, R. D., & Ku, N.-W. (2015). Terrestrial laser scanning as an effective tool to retrieve tree level height, crown width, and stem diameter. *Remote Sensing, 7*(2), 1877-1896.

Stovall, A. E. L., Anderson-Teixeira, K. J., & Shugart, H. H. (2018). Assessing terrestrial laser scanning for developing non-destructive biomass allometry. *Forest Ecology and Management, 427*, 217-229. doi:https://doi.org/10.1016/j.foreco.2018.06.004

Strahler, A. H., Jupp, D. L. B., Woodcock, C. E., Schaaf, C. B., Yao, T., Zhao, F., Yang, X., Lovell, J., Culvenor, D., Newnham, G., Ni-Miester, W., & Boykin-Morris, W. (2008). Retrieval of forest structural parameters using a ground-based lidar instrument (Echidna®). *Canadian Journal of Remote Sensing, 34*(sup2), S426-S440. doi:10.5589/m08-046

Sumida, A., Watanabe, T., & Miyaura, T. (2018). Interannual variability of leaf area index of an evergreen conifer stand was affected by carry-over effects from recent climate conditions. *Scientific reports, 8*. doi:10.1038/s41598-018-31672-3

Sun, G., Ranson, K., & Kharuk, V. (2002). Radiometric slope correction for forest biomass estimation from SAR data in the Western Sayani Mountains, Siberia. *Remote Sensing of Environment, 79*(2), 279-287.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.* Paper presented at the AAAI.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). *Going deeper with convolutions.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Teobaldelli, M., Somogyi, Z., Migliavacca, M., & Usoltsev, V. A. (2009). Generalized functions of biomass expansion factors for conifers and broadleaved by stand age, growing stock and site index. *Forest Ecology and Management, 257*(3), 1004-1013. doi:http://doi.org/10.1016/j.foreco.2008.11.002

Thakur, R. (2016). Scanning LIDAR in Advanced Driver Assistance Systems and Beyond: Building a road map for next-generation LIDAR technology. *IEEE Consumer Electronics Magazine, 5*(3), 48-54.

Thies, M., & Spiecker, H. (2004). Evaluation and future prospects of terrestrial laser scanning for standardized forest inventories. *Forest, 2*(2.2), 1.

Thimonier, A., Sedivy, I., & Schleppi, P. (2010). Estimating leaf area index in different types of mature forest stands in Switzerland: a comparison of methods. *European Journal of Forest Research, 129*(4), 543-562.

Tomasi, C., & Manduchi, R. (1998). *Bilateral filtering for gray and color images*. Paper presented at the International Conference on Computer Vision.

U.S. EPA. (2008). *EPA's Report on the Environment (Roe) (2008 Final Report)* (EPA/600/R-07/045F (NTIS PB2008-112484)). Retrieved from Washington D.C.:

Ung, C.-H., Bernier, P., & Guo, X.-J. (2008). Canadian national biomass equations: new parameter estimates that include British Columbia data. *Canadian Journal of Forest Research, 38*(5), 1123-1132. doi:10.1139/X07-224

Ung, C.-H., Lambert, M. C., Raulier, F., Guo, J., & P.Y, B. (2017). *Biomass of trees sampled across Canada as part of the Energy from the Forest Biomass (ENFOR) Program*.

van der Sande, M. T., Zuidema, P. A., & Sterck, F. (2015). Explaining biomass growth of tropical canopy trees: the importance of sapwood. *Oecologia, 177*(4), 1145-1155. doi:10.1007/s00442-015-3220-y

Vapnik, V. N. (1995). *The nature of statistical learning theory*: Springer-Verlag New York, Inc.

Vashum, K. T., & Jayakumar, S. (2012). Methods to estimate above-ground biomass and carbon stock in natural forests-a review. *J. Ecosyst. Ecogr, 2*(4), 1-7.

Vicari, M. B., Disney, M., Wilkes, P., Burt, A., Calders, K., & Woodgate, W. (2019). Leaf and wood classification framework for terrestrial LiDAR point clouds. *Methods in Ecology and Evolution, 0*(0), 1–15. doi:10.1111/2041-210X.13144

Vidal, C., Alberdi, I., Hernández, L., & Redmond, J. J. (2016). *National Forest Inventories: Assessment of Wood Availability and Use*: Springer.

Wang, D., Brunner, J., Ma, Z., Lu, H., Hollaus, M., Pang, Y., & Pfeifer, N. (2018). Separating Tree Photosynthetic and Non-Photosynthetic Components from Point Cloud Data Using Dynamic Segment Merging. *Forests, 9*(5), 252.

Wang, D., Hollaus, M., & Pfeifer, N. (2017). Feasibility of machine learning methods for separating wood and leaf points from terrestrial laser scanning data. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4*.

Wang, L., Hunt, E. R., Qu, J. J., Hao, X., & Daughtry, C. S. (2011). Towards estimation of canopy foliar biomass with spectral reflectance measurements. *Remote Sensing of Environment, 115*(3), 836-840.

Watt, P., & Donoghue, D. (2005). Measuring forest structure with terrestrial laser scanning. *International Journal of Remote Sensing, 26*(7), 1437-1446.

Wezyk, P., Koziol, K., Glista, M., & Pierzchalski, M. (2007). *Terrestrial laser scanning versus traditional forest inventory: First results from the Polish forests.* Paper presented at the Proceedings of the ISPRS Workshop ‘Laser Scanning.

Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., de Freitas, N., & Sohl-Dickstein, J. (2017). *Learned optimizers that scale and generalize.* Paper presented at the Proceedings of the 34th International Conference on Machine Learning-Volume 70.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). *3d shapenets: A deep representation for volumetric shapes.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Xi, Z., Hopkinson, C., & Chasmer, L. (2016). Automating Plot-Level Stem Analysis from Terrestrial Laser Scanning. *Forests, 7*(11), 252.

Xi, Z., Hopkinson, C., & Chasmer, L. (2018). Filtering Stems and Branches from Terrestrial Laser Scanning Point Clouds Using Deep 3-D Fully Convolutional Networks. *Remote Sensing, 10*(8), 1215.

Xu, F., Yu, J., Tesso, T., Dowell, F., & Wang, D. (2013). Qualitative and quantitative analysis of lignocellulosic biomass using infrared techniques: a mini-review. *Applied Energy, 104*, 801-809.

Yan, D.-M., Wintz, J., Mourrain, B., Wang, W., Boudon, F., & Godin, C. (2009). *Efficient and robust reconstruction of botanical branching structure from laser scanned points.* Paper presented at the Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics' 09. 11th IEEE International Conference on.

Yan, G., Hu, R., Luo, J., Weiss, M., Jiang, H., Mu, X., Xie, D., & Zhang, W. (2019). Review of indirect optical measurements of leaf area index: Recent advances, challenges, and perspectives. *Agricultural and forest meteorology, 265*, 390-411. doi:https://doi.org/10.1016/j.agrformet.2018.11.033

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 Sensing, 8*(5), 372.

Yang, X., Strahler, A. H., Schaaf, C. B., Jupp, D. L. B., Yao, T., Zhao, F., Wang, Z., Culvenor, D. S., Newnham, G. J., Lovell, J. L., Dubayah, R. O., Woodcock, C. E., & Ni-Meister, W. (2013). Three-dimensional forest reconstruction and structural parameter retrievals using a terrestrial full-waveform lidar instrument (Echidna®). *Remote Sensing of Environment, 135*(0), 36-51. doi:http://dx.doi.org/10.1016/j.rse.2013.03.020

Yang, Z., Jiang, W., Xu, B., Zhu, Q., Jiang, S., & Huang, W. (2017). A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds. *Remote Sensing, 9*(9), 936.

Ypma, T. J. (1995). Historical development of the Newton–Raphson method. *SIAM review, 37*(4), 531-551.

Yu, X., Liang, X., Hyyppä, J., Kankare, V., Vastaranta, M., & Holopainen, M. (2013). Stem biomass estimation based on stem reconstruction from terrestrial laser scanning point clouds. *Remote Sensing Letters, 4*(4), 344-353.

Zanne, A., Lopez-Gonzalez, G., Coomes, D., Ilic, J., Jansen, S., Lewis, S., Miller, R., Swenson, N., Wiemann, M., & Chave, J. (2009). Global wood density database.

Zeng, W., & Gevers, T. (2018). *3DContextNet: Kd Tree Guided Hierarchical Learning of Point Clouds Using Local and Global Contextual Cues.* Paper presented at the European Conference on Computer Vision. url: https://github.com/corochann/chainer-pointnet

Zhan, L., Douglas, E., Strahler, A., Schaaf, C., Xiaoyuan, Y., Zhuosen, W., Tian, Y., Feng, Z., Saenz, E. J., Paynter, I., Woodcock, C. E., Chakrabarti, S., Cook, T., Martel, J., Howe, G., Jupp, D. L. B., Culvenor, D. S., Newnham, G. J., & Lovell, J. L. (2013, 21-26 July 2013). *Separating leaves from trunks and branches with dual-wavelength terrestrial lidar scanning.* Paper presented at the Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International.

Zhang, K., Chen, S.-C., Whitman, D., Shyu, M.-L., Yan, J., & Zhang, C. (2003). A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *Geoscience and Remote Sensing, IEEE Transactions on, 41*(4), 872-882.

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. *Agricultural and forest meteorology, 209–210*, 100-113. doi:http://dx.doi.org/10.1016/j.agrformet.2015.03.008

Zheng, G., & Moskal, L. M. (2009). Retrieving leaf area index (LAI) using remote sensing: Theories, methods and sensors. *Sensors*, *9*(4), 2719-2745.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., & Torr, P. H. (2015). *Conditional random fields as recurrent neural networks.* Paper presented at the Proceedings of the IEEE International Conference on Computer Vision.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). *Learning deep features for discriminative localization.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Zhou, L., & Deng, Z. (2012). *Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation.* Paper presented at the Intelligent Vehicles Symposium (IV), 2012 IEEE.

Zhu, X., Skidmore, A. K., Wang, T., Liu, J., Darvishzadeh, R., Shi, Y., Premier, J., & Heurich, M. (2018). Improving leaf area index (LAI) estimation by correcting for clumping and woody effects using terrestrial laser scanning. *Agricultural and forest meteorology, 263*, 276-286. doi:https://doi.org/10.1016/j.agrformet.2018.08.026

Zhu, Z., & Liu, J. (2013). *Unsupervised extrinsic parameters calibration for multi-bem LiDARs.* Paper presented at the Proc. 2nd Int. Conf. Computer Science and Electronics Engineering.

Appendix 1. A Constrained Region Growing Method for Cylindrical Segmentation

```
Input:       $PointCloud$
Output:      $Segments$
Parameters:  $R_s, G_t, W_{ir}, D_c$
$RestPoints \leftarrow PointCloud; Segments \leftarrow \emptyset;$
while $(RestPoints \neq \emptyset)$ do
   $p_0 \leftarrow$ ArbitaryPointFrom$(RestPoints);$
   $end_1 \leftarrow p_0; end_2 \leftarrow p_0; GrownRegion \leftarrow p_0;$
   $RestPoints \leftarrow \{RestPoints - \boldsymbol{p}_0\};$
   $StopGrow.end_1 \leftarrow false; StopGrow.end_2 \leftarrow true;$
   $NN_1 \leftarrow \emptyset;\ NN_2 \leftarrow \emptyset; GrownRegion \leftarrow \emptyset;$
   while $(true)$ do
      if $(StopGrow.end_1 = false), \{NN_1\} \leftarrow$
      CubicNeighbors$(end_1, RestPoints, R_s);$
      $RestPoints \leftarrow \{RestPoints - NN_1\};$
      if $(StopGrow.end_2 = false), \{NN_2\} \leftarrow$
      CubicNeighbors$(end_2, RestPoints, R_s);$
      $RestPoints \leftarrow \{RestPoints - NN_2\};$
      if $(NN_1 = \emptyset), StopGrow.end_1 \leftarrow true;$
      if $(NN_2 = \emptyset), StopGrow.end_2 \leftarrow true;$
      if $(StopGrow.end_1\ \&\ StopGrow.end_2 = false)$, break;
      $\{end_1\_next, end_2\_next\} \leftarrow$ PartitionToLargestTwoClusters$(NN_1, NN_2, G_t);$
      if (Width$(end_1)$ < Width$(end_1\_next)$)$, Hit \leftarrow true;$
      if $(Hit),$
         $StopGrow.end_2 \leftarrow false;$
         if (Width$(end_1\_next)$ − Width$(end_1)$)/Width$(end_1)$) > $W_{ir}$
         or Angle(Direction$(end_1\_next, end_1)$, Direction$(end_1, end_1\_prev)$) > $D_c$,
            $StopGrow.end_1 \leftarrow true;$
         if (Width$(end_2\_next)$ − Width$(end_2)$)/Width$(end_2)$) > $W_{ir}$
         or Angle(Direction$(end_2\_next, end_2)$, Direction$(end_2, end_2\_prev)$) > $D_c$,
            $StopGrow.end_2 \leftarrow true;$
         $end_1\_prev \leftarrow end_1; end_1 \leftarrow end_1\_next;$
         if Angle(Direction$(end_2\_next, p_0)$, Direction$(end_1\_next, p_0)$) > 90°,
            $end_2\_prev \leftarrow end_2;$
            $end_2 \leftarrow end_2\_next;$
         $GrownRegion \leftarrow \{GrownRegion\ \cup\ end_1 \cup end_2\}s;$
      $Segments \leftarrow \{Segments \cup GrownRegion\};$
```

Detailed implementation using C++ and MATLAB can be found in https://github.com/truebelief/cylsegment.