DIAMETER AND HEIGHT ESTIMATION USING A LIGHTWEIGHT HANDHELD TERRESTRIAL LASER SCANNER IN LONGLEAF PINE PLANTATIONS

by

PORTER, STEVEN M., Jr.

(Under the Direction of Cristian R. Montes)

ABSTRACT

In this thesis a lightweight handheld terrestrial laser scanner was used to estimate tree diameters and heights for longleaf pine plantations. A method was developed to isolate the tree stem and estimate diameters despite significant occlusion of stem by branches and foliage. To further improve diameter estimation values, an ad-hoc bias correction function was fitted to the residual data. The bias function parameters were found using a cross validation method. Multiple diameters derived out of a terrestrial laser scanner were then further used to estimate height using an inverse taper model of a segmented polynomial form. It was found that upper stem diameters were less biased predictors of height than lower-stem diameters.

INDEX WORDS: LiDAR, terrestrial laser scanner, tree taper, forest inventory, tree measurement, tree height estimation, upper-stem diameters, forest biometrics, longleaf pine

DIAMETER AND HEIGHT ESTIMATION USING A LIGHTWEIGHT HANDHELD TERRESTRIAL LASER SCANNER IN LONGLEAF PINE PLANTATIONS

By

PORTER, STEVEN M., Jr.

B.S., Mississippi State University, 2018

A Thesis Submitted to the Graduate Faculty of The University of Georgia in partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2020

© 2020

PORTER, STEVEN M., Jr.

All Rights Reserved

DIAMETER AND HEIGHT ESTIMATION USING A LIGHTWEIGHT HANDHELD TERRESTRIAL LASER SCANNER IN LONGLEAF PINE PLANTATIONS

By

PORTER, STEVEN M., Jr.

Major Professor:	Cristian R. Montes
Committee:	Bronson P. Bullock
	Joseph Dahlen

Electronic Version Approved:

Ron Walcott
Interim Dean of the Graduate School
The University of Georgia
August 2020

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Forest inventory provides relevant stand level information to support management decisions. One of the main purposes of the forest inventory process is to characterize stand attributes that are important for stock and value determinations. In general, and with a very few exceptions, the methods used to record forest inventory information have remained the same over the last 100 years (LaBau et al. 2007). While including new measurement techniques to speed up the process of collecting tree measurements have been implemented, the fundamental tree measurements are still the diameter at breast height (DBH) and total height (H). From these variables, tree volume can be estimated (Clutter et al. 1983). To save cost and to improve sampling speed, a limited number of trees are measured for H, therefore H-DBH paired observations are used in a regression framework to calibrate a non-linear model using log transformations (e.g., Schumacher 1939) or a polynomial model to impute missing heights. A wide variety of statistical sampling techniques have been used to reduce the number of sampled trees while ensuring a good representation of the stand. This information is later expanded to a whole stand to the whole stand to provide per area estimates (Shiver & Borders 1996). The general process is well known in the forest community, and it provides reasonable volume and green weight estimates assuming the trade off of accepting some estimation error at the benefit of reducing sampling cost. However, the forest inventory process could possibly be improved with remote sensing technology such as Light Detection and Ranging (LiDAR).

LiDAR is a mechanical device that uses a laser beam to scan a 3-dimensional environment into a series of spatially referenced points (Campbell & Wynne 2011). A LiDAR sensor relies on the speed of light to calculate the time it takes the laser beam to travel the distance between the sensor and a reflective object. The sensor records this time while using its position in space, which may be determined with an inertial measurement unit and/or a GPS. Depending on the precision of the LiDAR equipment, these estimates will have varying levels of error inherent. There are a variety of LiDAR sensors available, but perhaps more important than the sensor itself is the sensor platform as LiDAR is a line of sight sensor. In forestry there are two platforms- airborne and terrestrial platforms.

Aerial laser scanning (ALS) systems have shown their value as they provide low cost estimates for tree height (Popescu et al. 1999), canopy size and segmentation (Popescu et al. 1999, Strimbu & Strimbu 2015). Approaches incorporating limited ground data have been used to estimate diameters from these heights (Evans & Parker 2004) for an area as well as on an individual tree basis or for an area (Evans & Parker 2009). However, none of these methods directly measure the most valuable part of the tree- the bole, and therefore these values must be estimated with a model and some ground data. Lastly, ALS based inventories are not economically feasible unless applied over large areas (Tilley et al. 2004).

One of the benefits of Terrestrial Laser Scanning (TLS) is the ability to scan tree stems from a similar perspective to a forester walking through the woods, providing a direct measure of tree stems. In order to measure tree stems with a TLS there are several data processing steps that need to be made. First, it is necessary to isolate a tree from its environment. Second it is necessary to isolate the tree stem from the tree branches. Since this thesis focuses on measuring single tree scans, tree segmentation was not addressed, although stem segmentation is addressed in detail

because all of the trees in this sample had branches. Many diameter measurement methods from TLS data have been developed. Some methods are simpler, where the point clouds are manually measured in the point cloud with virtual calipers (Fang & Strimbu 2017). One method uses eigenvalue decomposition to find points lying on vertical services to find trees (Liang et al. 2012, Conto et a.l 2017) because trees stems should be relatively flat but vertically oriented surfaces. Many papers use some variation of cylinder fitting to model the stem (Thies et al. 2004, Liang et al 2012, Moskal & Zheng 2012, Pitkanen et al. 2019, Srinivasan et al. 2015), which has the benefit of simultaneously estimating lean and orientation of the tree stem. More recently, another method is the application of the Hough Transform (Hough 1959) for circle detection (Conto et al. 2017, Liu et al. 2018) building on the assumption that tree stems are approximately circular in cross section. Similar to this, trees have been modeled by detecting vertical sections and analyzing them like terrain, with the tree stems having characteristics like ground points and branches like trees (Pitkanen et al. 2019). For approaches where cylinder fitting was not used to derive diameters, various circle fitting techniques have been implemented (Henning & Radtke 2006, Conto et al 2017, Cabo et al 2018, Hyppa et al 2020).

Height estimation from TLS data has been studied to a lesser extent than diameter estimation. Using just the highest point from each tree has produced mixed results with Saarinen et al. (2017) reporting high error and a consistent bias towards under-prediction. Using similar methodology, but with an additional error checking step, Cabo et al. (2018) reported low error and apparent unbiasedness. Additionally, in Liu et al. (2018) this approach was also found to be acceptable, although an additional step was used to separate out canopies underneath and overlapping the target tree. No studies have attempted to indirectly estimate height and have simply used the naïve criteria of using the highest observed point from the TLS data. The main problem

relies on the TLS sensor's inability to have a clear line of sight to the top of the tree in a practical inventory. This is particularly true in dense stands such as the un-thinned pine plantations used in this thesis. Also, it is important to note that all studies reviewed utilized tripod mounted sensors. The one exception to this is Hyppa et al. (2020) where a Mobile TLS (MTLS) was mounted to a drone that was manually flown underneath the canopy to scan trees. In this thesis a handheld MTLS is utilized for data collection.

Measurement of tree diameter and height are closely related to tree taper modelling. A tree taper model simply describes the shape of a tree as a series of diameters at different heights (Burkhart & Tome 2012). These models are useful for computing volume and product class when DBH and H are available from inventory data. Taper models are one of the most heavily researched topics in forest biometrics and can be broadly placed into three categories (Burkhart & Tome 2012). Simple models, such as a polynomial (Kozak et al. 1969) have an attractive characteristic of simplicity but might not perform as well as other models. The second type of taper model is the segmented model where each segment is a function that has a similar form to the natural shape of a tree section, with the Max & Burkhart (1976) being a very successful taper model of this form. Lastly there are the more complex variable exponent models, such as Kozak's (1988) model. Taper models are fit to a sample of trees that are felled because upper stem measurements of standing trees are difficult to obtain. With MTLS it might be possible to collect taper data cheaply because trees might not need felling.

In this thesis the problem solved is twofold- the measurement of tree diameters and heights. In chapter 2, diameters are found for *P. palustris* with stems that are highly occluded by tree branches. In chapter 3 the problem of biasedness in diameter measurement with respect to height

is addressed. In chapter 4 multiple diameters are used for finding total height of these trees by using a taper model fitted to destructively sampled data.

CHAPTER 2.

FULLY AUTOMATIC MEASUREMENT OF TREE DIAMETERS WITH A LIGHTWEIGHT HANDHELD TERRESTRIAL LASER SCANNER

Introduction

Tree diameter is an extremely useful metric in forestry- with the Diameter at Breast Height (*DBH*) being the most relevant characteristic to be looked for in an individual tree attribute (Avery & Burkhart 2002). While *DBH* is very useful for describing trees, sometimes information about diameters at different points in a tree is needed. Tree taper is the characterization of tree shape, showing the variation in diameter along the tree stem. Diameters are generally assumed to decrease as height increases, and diameter at top height of tree stem is assumed to be zero in excurrent (e.g., trees with a single stem) species like *Pinus palustris* and other commercially valuable conifers. The common way to measure *DBH* is using a calibrated tape in pi units. The forester simply wraps the tape around the tree at breast height (1.37m above ground) and reads off the diameter measurement. Similarly, the upper stem diameters necessary to characterize taper are typically measured with a tape at the expense of felling the tree, although these measurements can be clumsily recorded with laser calipers and dendrometers. This information can be used to develop taper equations, which are generally used to compute volumes, upper stem diameter, and merchandising information from DBH and height (H) as input variables (Burkhart & Tome 2012). While DBH alone can be used for merchandising and H imputation, an actual upper stem diameter measurement could potentially be useful. In this chapter a procedure to isolate stems and measure

stem diameters is detailed in mobile terrestrial laser scanner (MTLS) scans of *P. palustris* collected in South Georgia, U.S.A., from June to July 2019.

Methods

Data Collection

A set of 3 1/10th ac plots were measured in 24 stands in South Georgia from June to July of 2019, of which 20 were selected for destructive sampling. The stand locations are shown in figure 2.1. At each plot, trees were evaluated for diameters and heights. For each stand, a diameter distribution was worked up for each stand, and 20 trees from the stand were individually selected to represent this distribution. All stands were un-thinned longleaf pine plantations with previous cover type corresponding to old field or cutover sites. In each one of these stands, a group of trees were scanned using a handheld mobile terrestrial laser scanner (MTLS) and were destructively sampled for taper data acquisition.

Trees destructively sampled were not necessarily in the inventory plots and were generally selected to have either excellent form or a few defects in order to meet sampling goals to select both defect free and defective trees. After selection and scanning, trees were measured for diameter with a diameter tape to the nearest 1/10th in at 0.5ft, 2ft, and 4.5ft above the ground. This corresponds to diameter to the nearest 0.2540cm at 0.1524m, 0.6096m, and 1.3716m above the ground respectively. DBH (in) and total height (ft) were measured with a tape and hypsometer before felling. Felled trees were later measured at 8ft (2.4384m) above the ground and then every 4ft (1.2192m) until diameter decreased to 3in (7.6200cm). Total height was also recorded. With the trees laying on the ground, height from 3in (7.6200cm) top to tip of tree was measured. Each tree was later cut at these measurement points and was weighed to the nearest 1/10thlb (0.0454kg).

Lastly, at each measurement point a disc was removed and stored for collection of data related to diameter inside bark, moisture content, and specific gravity for a related project. For defective trees, height and length of each defect was recorded. For forked trees, each fork was measured for diameter at the base of where the forked section begins, and then each forked stem was measured at the base and finally the 3in top was found for each forked stem. Some of these trees were scanned with MTLS but were not the focus of this study and were not included. See appendix 1 for summary statistics of destructively sampled data and stand reconnaissance.

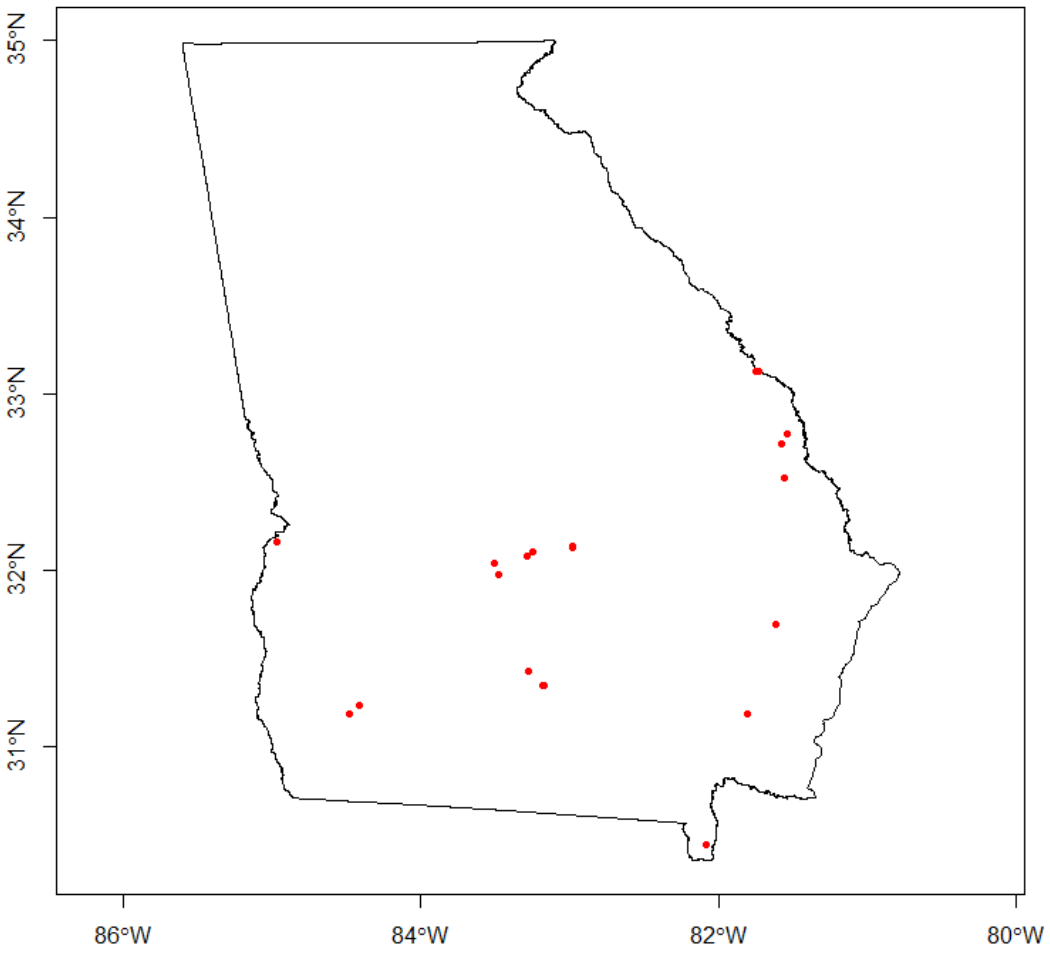

Figure 2.1. Map of sampled longleaf stands in The State of Georgia, U.S.A.

Tree Scanning Procedure

The tree scanning procedure was executed as follows: The operator walked a loop around the target tree while keeping the sensor oriented towards the tree as much as conditions allowed, and then returned to the origin point to close the loop. The exact distance from the tree varied through the scanning process, but the line of sight to each tree was maintained as much as possible. Completing a scan typically took anywhere from 1-3 minutes depending on field conditions. Target trees were further identified using an aluminum pole of approximately 1 meter in length to allow for later identification in the point cloud. The pole created a highly reflective target that was extremely distinct looking when the point cloud was graphically explored. The pole was set into the ground immediately adjacent to the base of the tree at approximately a 45-degree angle from horizontal. Figure 2.3 shows the plot center target against a contrasted background. Only one tree was impossible to identify and was excluded from the final dataset. See figure 2.2 for a diagram of the scanning process and figure 2.4 for a picture of the MTLS unit as well as a typical *P. palustris* site. After completing the scan, the data was transferred to a flash drive. This data was later transferred to a computer and processed with GeoSLAM Hub- a proprietary program compatible with the GeoSLAM Zeb Horizon MTLS sensor used for data collection. The files were exported as .las files and individual trees were clipped out in CloudCompare. See appendix 2 for technical specifications of the MTLS unit.

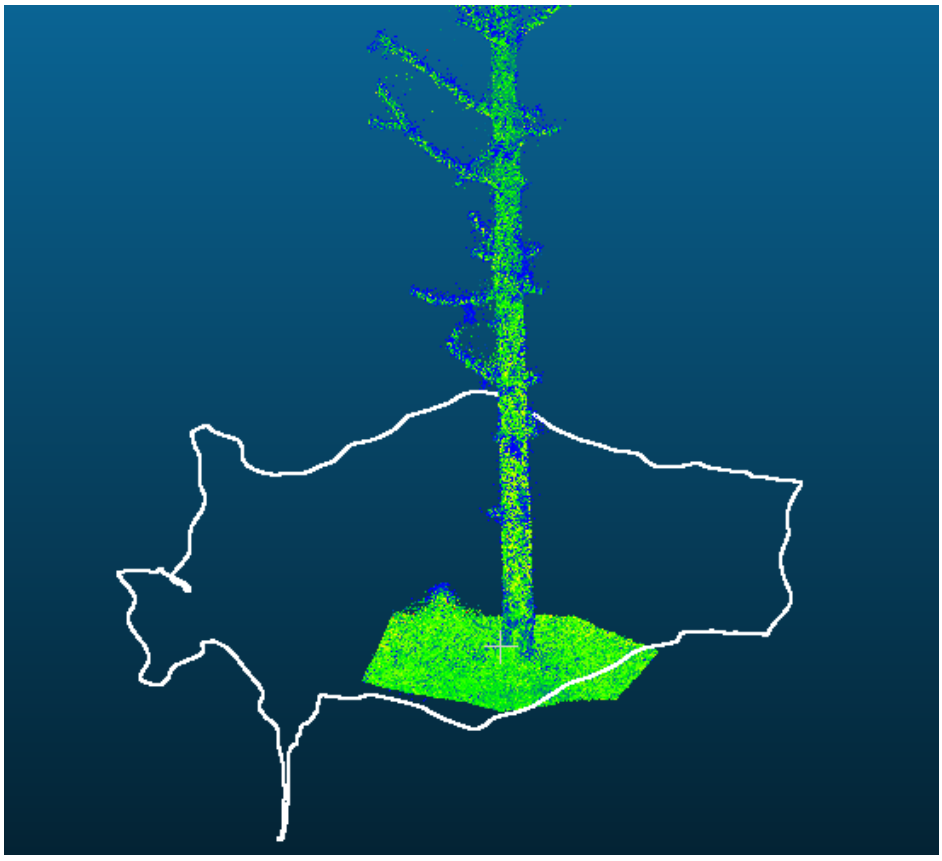

Figure 2.2. Visualization of scanning process. The white line is the sensor path around the tree.

Figure 2.3. Visualization of the plot center stick target- circled in red.

Figure 2.4. Left- GeoSLAM Zeb Horizon MTLS unit and plot center stick target. Right- typical *P. pallustris* stand used for pine straw production.

Classification of Ground Points:

The first processing step was to classify the point cloud into ground and non-ground points. Here a cloth simulation filter (CSF), part of the lidR package, was used for classification (Zhang et al 2016). The process inverts the data and seeks for the boundary plane that includes all the points from above by simulating a cloth falling onto the surface. This cloth is flexible and will conform to some topology. However, objects like trees are very narrow and deep pits from this perspective and therefore the cloth will not "fall" into this pit. A rigidity parameter was adjusted to better reflect the flat ground condition present in this dataset. After classification of ground points, the point cloud was normalized so that the ground points are the new reference point for 0

in the Z-axis. This was accomplished with a simple k-nearest neighbor inverse distance weighting algorithm.

Detection of circular objects in the **x,y** plane:

To detect circular objects in the **x,y** plane of the entire point cloud the Adaptive Circular Hough Transform (ACHT) was used. Since tree stems are assumed to be uniquely circular in the **x,y** plane this algorithm was found to be a good fit. The algorithm, implemented in the the TreeLS package, was used to assign a binary classification as to whether a point was part of a circular pattern or not (Conto et al. 2017). The ACHT originated as the Hough Transform (Hough, P.V.C. 1959), which was originally developed to detect lines in images. This method has been extremely popular for automated edge detection (Hart P.E. 2009), and the function has been adapted to circular object detection (Illingworth & Kittler 1987), as well as other shapes. See figure 2.5 for an example of what the CSF and ACHT do to the point cloud.

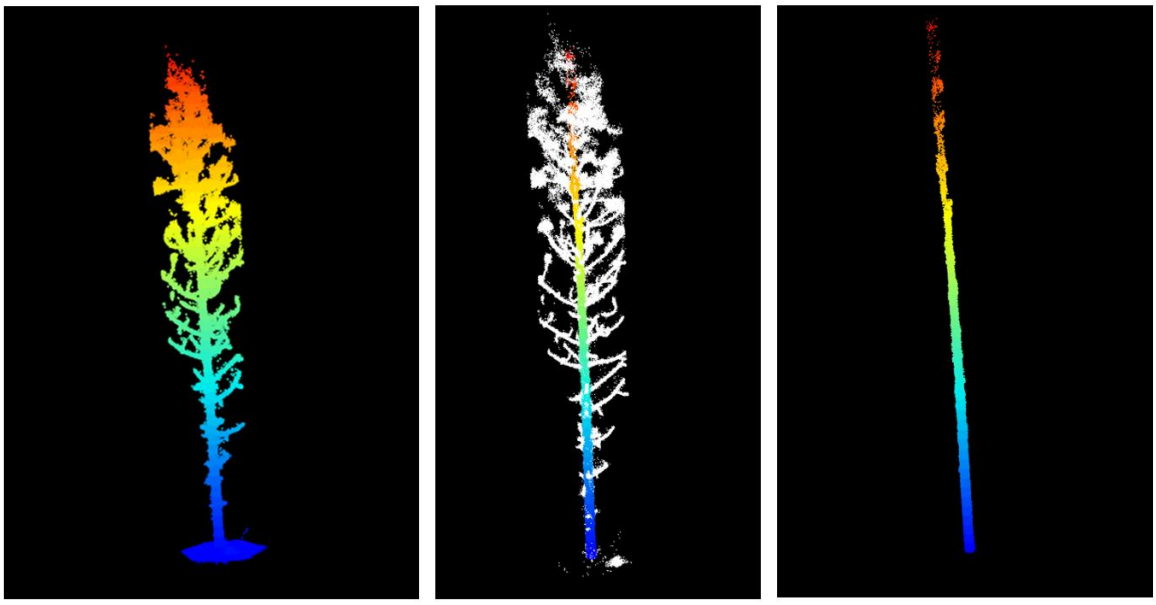

Figure 2.5. Visualization of a tree before and after applying the ACHT (left & center). White points (center) are the non-stem points. Ground points also removed by the CSF routine (middle). After these two subsets all that remains are mostly stem points (right).

2-D Kernel Density Estimation:

The kernel density for each tree was estimated for all points in the **x,y** plane, with the free parameters in *K* being *g* for grid size and *b* for bandwidth. Thus, *K(***x, y,** *g, b)* was the kernel used for each tree. Kernel density can be thought of as a histogram but generalized to more than one dimension. The rationale behind this was that the tree stem might be more reflective than the foliage and is also the center point of the scan. Considering this, the stem points should be greater in density than the foliage/branches in the **x,y** dimensions because there will be more returns in these areas. To compute density, the package KernSmooth (Wand 1994) was selected as it is fast (Deng & Wickham 2011) and implemented in R. Using *K* it was possible to penalize the estimation of circular parameters by assigning values from *K* to corresponding points in the point cloud. This is shown in equation 2.5. It was found that *K(***x, y,** *g, b*) is best computed after the ground points are subset out but before the ACHT is computed.

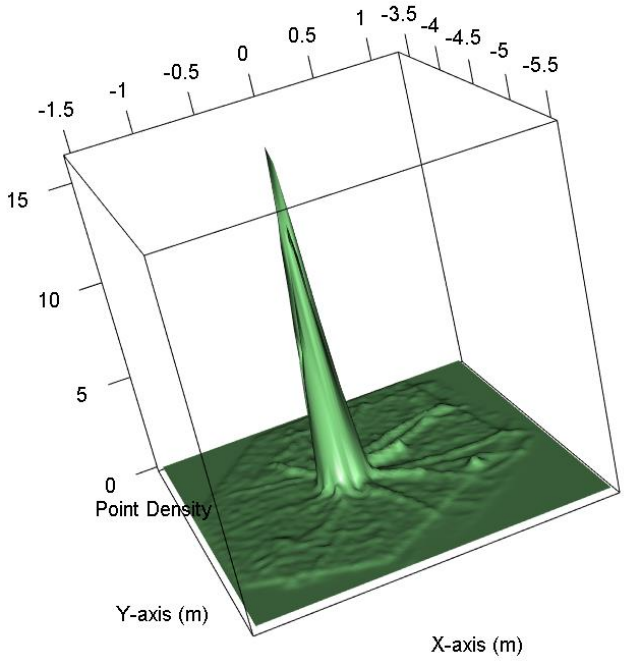

Figure 2.6. *K(***x, y,** g, b*)* of a tree after CSF and normalization of ground points.

Local Estimation of Tree Diameters:

Trees are approximately circular in cross-section; therefore a tree can be represented as a series of circles at different heights. Circles are mathematically described by the parameters *a,b* and *r*, with *a,b* representing the location of the center and *r* representing the radius. Equation 2.1 is the equation of the circle in cartesian coordinates.

$$r^2 = (x - a)^2 + (y - b)^2 \quad (2.1)$$

Where *r* is the radius, *x,y* are any 2 dimensional coordinate lying on a line of length *r* rotated around the center point *a,b* at any angle of rotation. Considering this, and the type of data, a simple objective function describing the circle equation was defined as:

$$0 = r - \sqrt{(x_i - a)^2 + (x_i - b)^2} \quad (2.2)$$

With $x_i, y_i = \mathbf{x}, \mathbf{y}$ vectors of coordinate data of length equivalent to the number of points in a circular pattern. Unfortunately for this particular case, this equation cannot be solved exactly for zero for parameters *a,b,r* with respect to data **x,y** because the data in any circular segment of a tree, **x,y,** does not represent a perfect circular pattern. Note, that these coordinates **x,y** are not the same centers for the whole tree. Rather, **x,y** for each circle were selected from a range in the data by the **z** axis. This can be conceptualized as a narrow disc sliced from the tree. A similar formulation was implemented in Kasa (1976) but this problem was extended to use maximum likelihood estimation as the parameter searching method.

The circular parameters *a,b,r* were found with an iterative maximum likelihood estimation (MLE) optimization algorithm. In this case MLE was implemented so that an additional parameter, σ, could be estimated simultaneously. The parameter σ represents the variation in ***x,y*** about the estimated radius *r* from center coordinates *a,b*. This method is best conceptualized as a

normal distribution rotated about *a,b* with the mean centered on *r* and σ being the dispersion of x_i, y_i point distance about *r*. Equation 2.3. is the MLE expressed as a loss function to be minimized in order to find the best *a,b,r* and σ. Lastly, it is assumed that error distribution of *r* is $N(0, \sigma^2)$.

$$L(\mathrm{a}, \mathrm{b}, \mathrm{r}, \sigma|\mathbf{x}, \mathbf{y}) = argmin \sum_{i=1}^{n} -\log\left(\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{\left((r_i)-\sqrt{((x_i-a)^2+(y_i-b)^2)}\right)^2}{2\sigma_i^2}}\right) \quad (2.3)$$

This MLE formulation assumes a normal distribution, and therefore should be numerically equivalent to least squares estimation under the conditions of circular parameter regression. In equation 2.3, equation 2.2 was substituted into the normal distribution in place of the mean. It is important to note that equation 2.3. is a log-likelihood in interest of computational efficiency.

A penalty was added to the likelihood function as a generalization, formulated as $L(a, b, r, \sigma|\boldsymbol{x}, \boldsymbol{y}) * P_i$, which is simply the loglikelihood function multiplied by the penalty. Essentially this penalty means that points with lower density values from the kernel function *K*, have a lower weight than points with a higher density, which favors points that represent the stem. While it is possible to include more sophisticated penalty, in this case penalties were taken directly from the KDE. The penalty is explicitly defined in equation 2.4, with the application to MLE defined in equation 2.5. While the P_i term is simply derived directly from K, this formulation leaves open the possibility of more exotic penalty structures for future research.

$$P_i = K(\mathbf{x}, \mathbf{y}, g, b) \qquad (2.4)$$

$$L(\mathrm{a}, \mathrm{b}, \mathrm{r}, \sigma|\mathbf{x}, \mathbf{y}) = argmin \sum_{i=1}^{n} -\log\left(\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{\left((r_i)-\sqrt{((x_i-a)^2+(y_i-b)^2)}\right)^2}{2\sigma_i^2}}\right) * P_i \qquad (2.5)$$

The L-BFGS-B algorithm (Byrd et al. 1994) native to the R optim() function was selected for parameter estimation. Extreme values of **x,y** in each circle were used to define the box constraints, which ensures that the algorithm will at least converge to parameters in the data space. Individual circles with fewer than 10 points or where a convergence code indicating some type of error in L-BFGS-B were removed by either a try-catch or subset operation. A pseudo-error statistic was computed for each individual circle fitted. This value, defined as $RMSD_c = \sqrt{\frac{\sum_{i=1}^{n}(\sqrt{(a_c-x_i)^2(b_c-y_i)^2})^2}{n}}$, can be conceptualized as the root mean square Euclidian distance of all points in the circular pattern from the fitted radius rotated about a, b. Note that this is not a true error as there is not necessarily a physical measurement at the exact point the error is computed. Lastly, an extra argument was added to arbitrarily constrain the maximum radius, and was selected to be larger than the largest expected radius. This method ensured no extremely large radii were computed, although it required some fore-knowledge of the expected maximum tree size encountered. An application of circle fitting to a single tree dataset at 1.37m of height is shown in figure 2.7 and 6.1m in figure 2.8. It can be seen in figure 2.7 that after application of CSF, *K*, and the ACHT that the removal of outlying points provided slightly better data for circle fitting, particularly in locating a, b inside the tree stem. Figure 2.8 is a more dramatic demonstration as the slice was taken far from the ground and well into the crown of the tree where occlusion of the tree stem by branches was usually more pronounced. Furthermore, in figure 2.9 the profile of a

tree (i.e., radius at heights) was shown, where 2.9 was a "naïve" circle fitting procedure with no CSF, *K*, or ACHT. In contrast figure 2.10 shows the same tree profile but after application of CSF, *K* and the ACHT. Figure 2.10 is clearly closer to what is expected of the tree stem shape of an excurrent tree species.

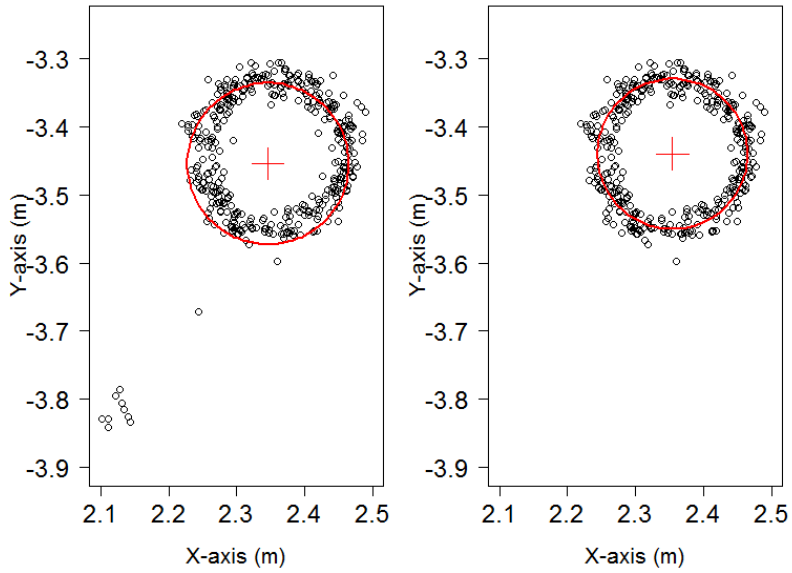

Figure 2.7. Circles fitted to slice of tree 1.37m above ground and 1cm thick. The left panel is before the ACHT and computation of K, but after removal of ground points. The right panel is after these steps are performed.

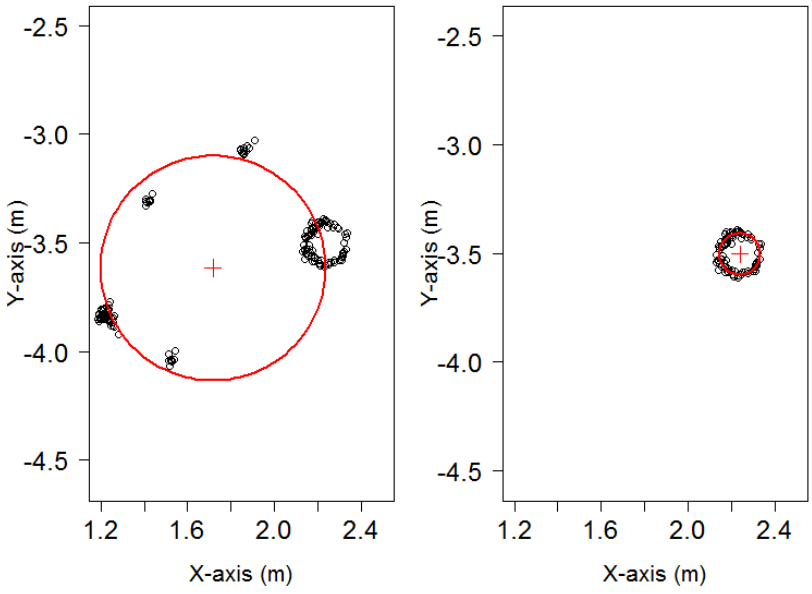

Figure 2.8. These circles are fitted to slice 6.1m above ground and 1cm thick. The left panel is before the ACHT and computation of K. The right panel is after these steps are performed.

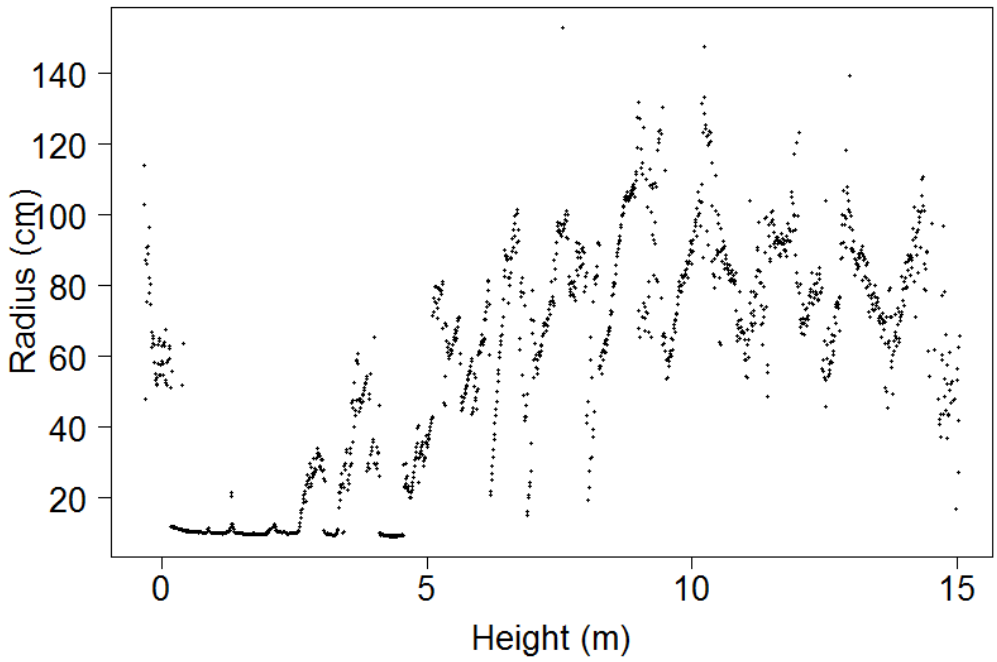

Figure 2.9. Tree with circles fit every 0.01m, but without CSF, *K* or an ACHT. This is an example of using circles fit in a similar manner to how the left-hand panels of figures 2.7 and 2.8 were generated.

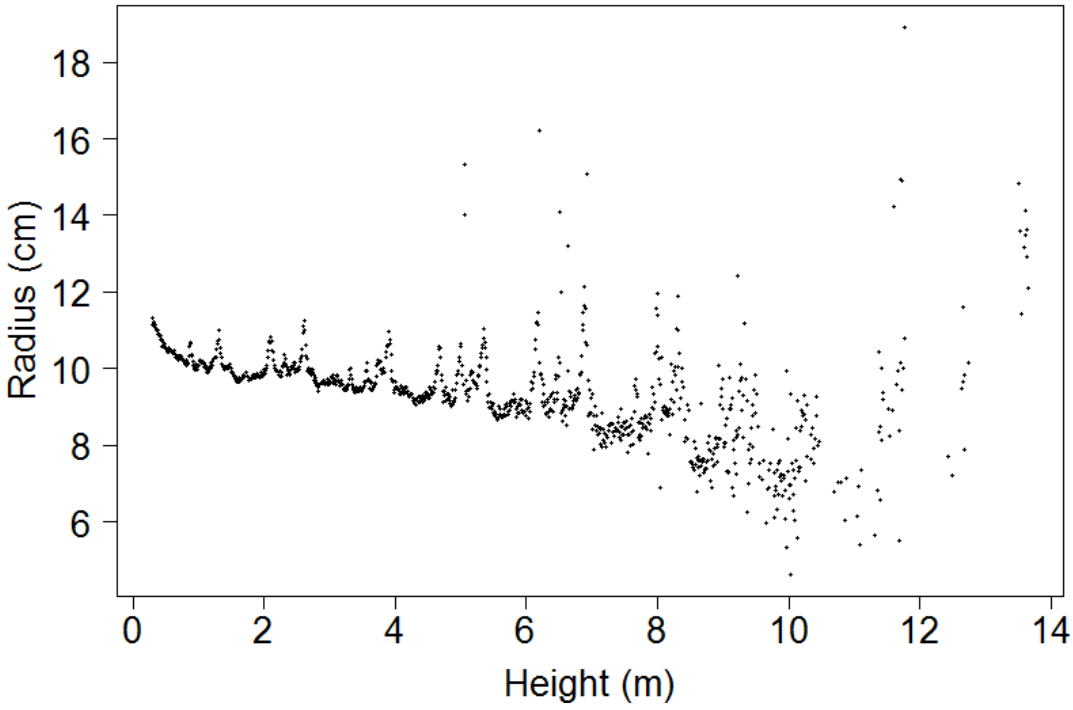

Figure 2.10. Tree with circles fit every 0.01m but with CSF, *K* and the ACHT.

Overview of Computing Segment Diameters

As shown in figure 2.10, a large number of diameters could be computed for each tree. However, only a few diameters are necessary to adequately describe trees, and considerable noise is seen in figure 2.10 despite the best efforts of the methods previously described. Therefore, two methods to estimate diameters were compared. Tape measured diameters were used as ground truthing values. The first method simply took a 2cm thick slice of the tree and measured diameter. The second method used all of the points in the tree to compute segment diameter by assimilating a global model of diameters as a function of height and a segment level model of the same form, for diameters at ever 1cm of height. Both methods used all of the previously described procedures.

Computing Segment Diameters- Method #1

The most conceptually simple method to processing the amount of information shown in figure 2.10 was to simply take a small slice of the tree at the desired height and find the diameter. Instead of computing diameter every 0.01m in height up the tree, only a few slices were extracted for diameter measurement. Less formally this method can be thought of as a virtual tape measure approach. First the CSF, ACHT, and *K* steps as described previously were applied to the whole tree point cloud. Then, for each physically measured $d_h,\ h$ pair in each tree, a subset of points in the corresponding MTLS scan was defined as $\pm$ 0.01(m) from h in the *Z*-axis. Essentially this was a 2cm thick slice of points at each physical measurement point. After these points were selected, a circle was fit using the local circle fitting procedure described previously. The 2cm slices of a tree are shown in figure 2.11. After computing radius in meters, each measurement was converted to diameters in centimeters to simplify interpretation. A simplified flow chart of method #1 is shown in figure 2.12. See appendix 3, blocks A.3.1 & A.3.2 for the R-code used to estimate diameters using method #1.

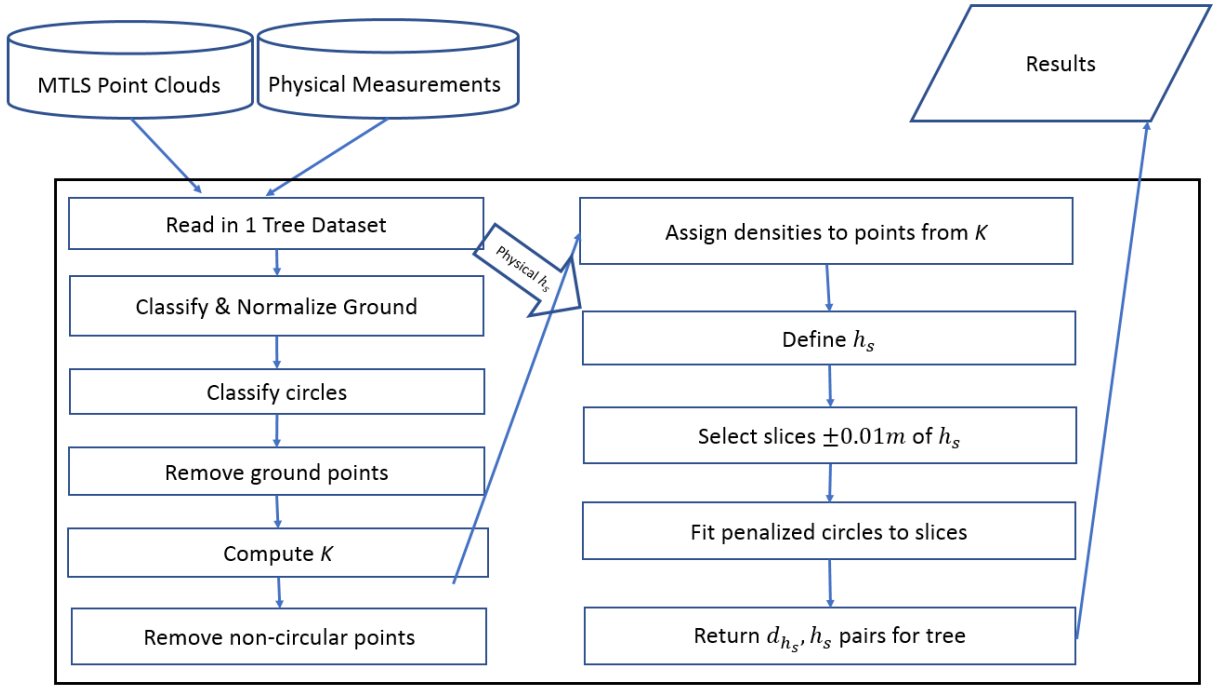

Figure 2.11. Simplified flow diagram of diameter estimation method #1.

Figure 2.12. Tree sliced into 2cm thick segments. *K* penalized circles were fitted to each slice in the same manner as the right-hand panels of figure 2.7 & 2.8.

Computing Segment Diameters- Method #2

As shown in figure 2.10 it was clear that many circles can be fitted with this method. In order to reduce this information to a manageable scale as well as remove some noise, linear models were used to account for this variability. A model form of $\widehat{d_i} = \beta_0 + \beta_1 h_i + \epsilon$ with weights of $w_i = \frac{1}{RMSD_c}$ was fit to each segment of the tree. Each segment was defined by the heights closest to the height at which a physical measurement was taken for each tree. I.e., all the points in the region around a measurement point were used to compute a model in that segment of the tree. This exact same model for was fit for all $\widehat{d_i}, h_i$ in the MTLS data (e.g., circles every 1cm of vertical height). Lastly, for each segment, these models were assimilated at an h_s (segment height) defined by the destructively sampled heights. This assimilation was weighted by the inverse standard error of prediction for the two models at the point h_s. This approach is graphically described in figures 2.13-2.16. A simplified flow diagram of method #2 is presented in figure 2.17. See appendix 3, blocks A.3.1 & A.3.3 for the R-code used to estimate diameters using method #2.

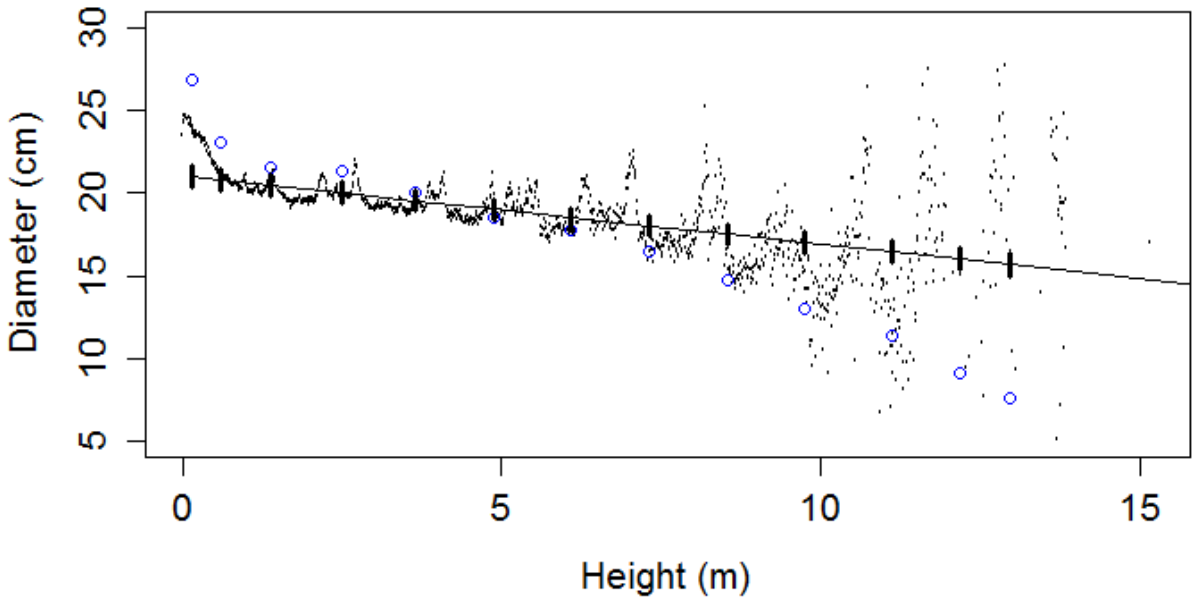

Figure 2.13. Graphical description of the method #2 whole tree weighted linear model. Black dots are raw radii, black line is whole tree linear model, blue dots are the true measured diameter, and vertical black lines are the model error.

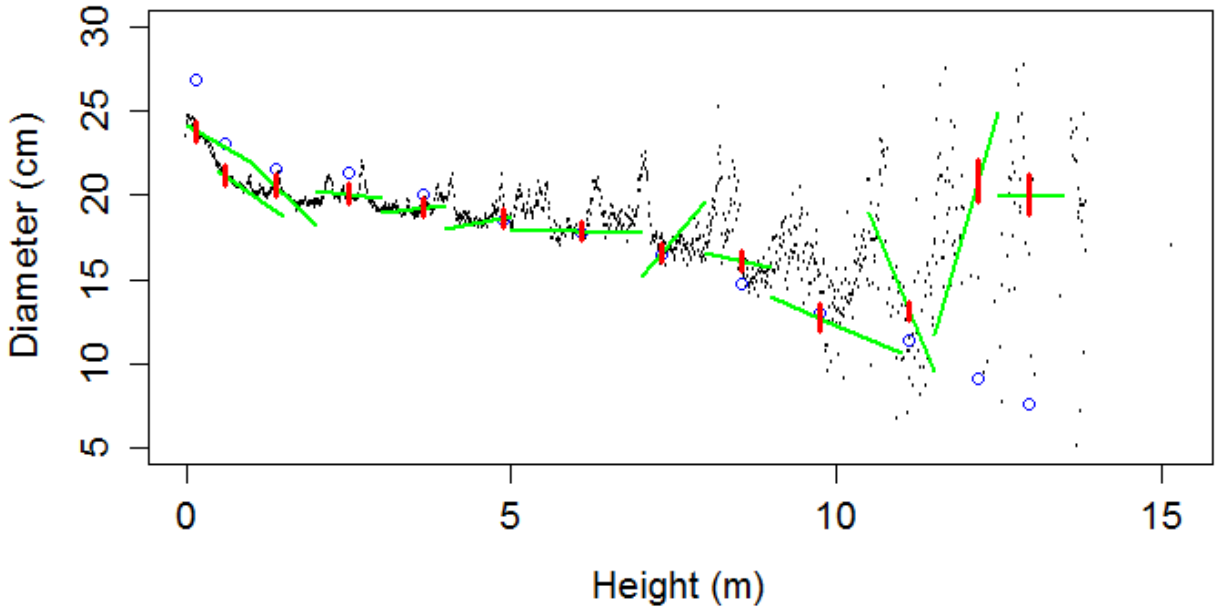

Figure 2.14. Graphical description of the method #2 segment models. The green lines are the local weighted linear model with the red vertical lines being the model error.

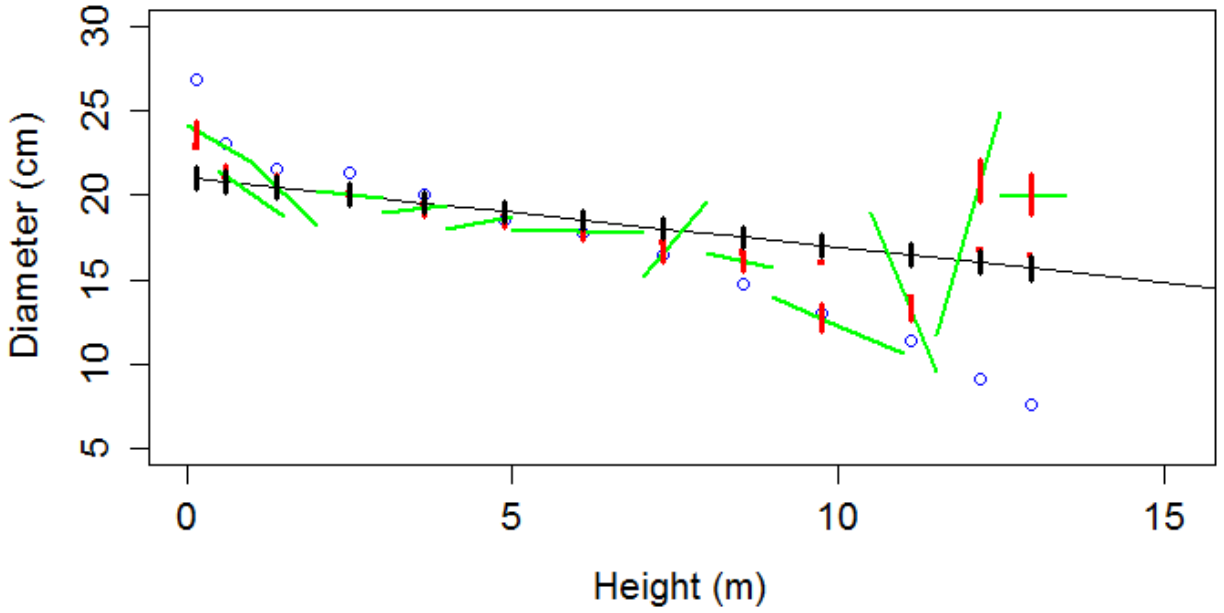

Figure 2.15. Graphical description of the method #2 model assimilation. The red squares are the assimilated values from the whole tree and local models.

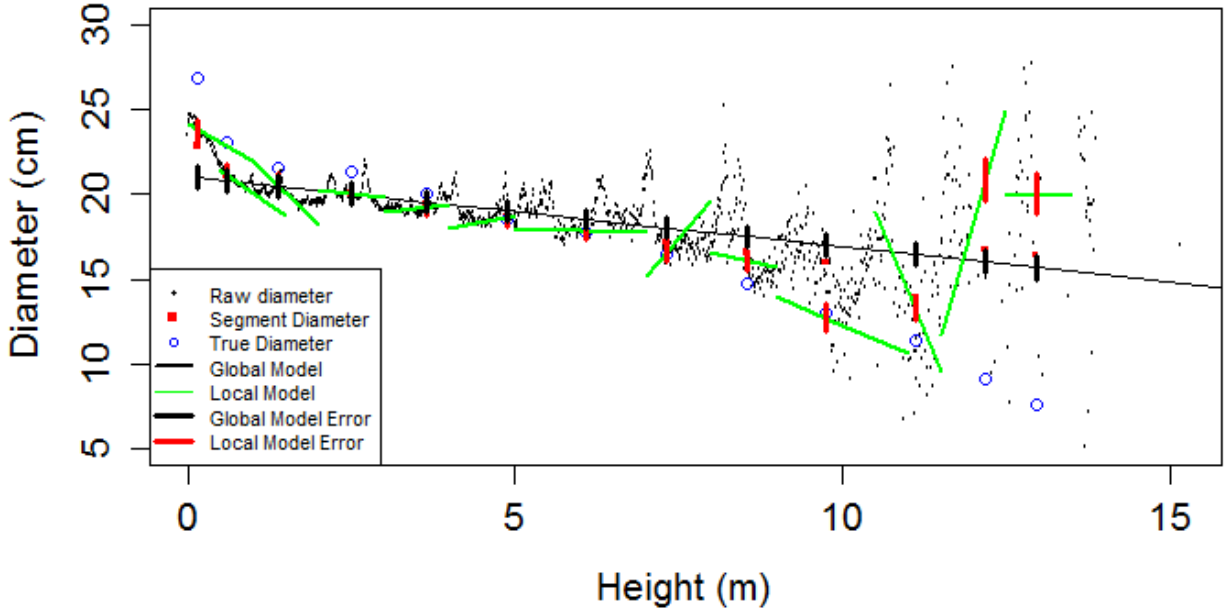

Figure 2.16. Complete graphical description of method #2.

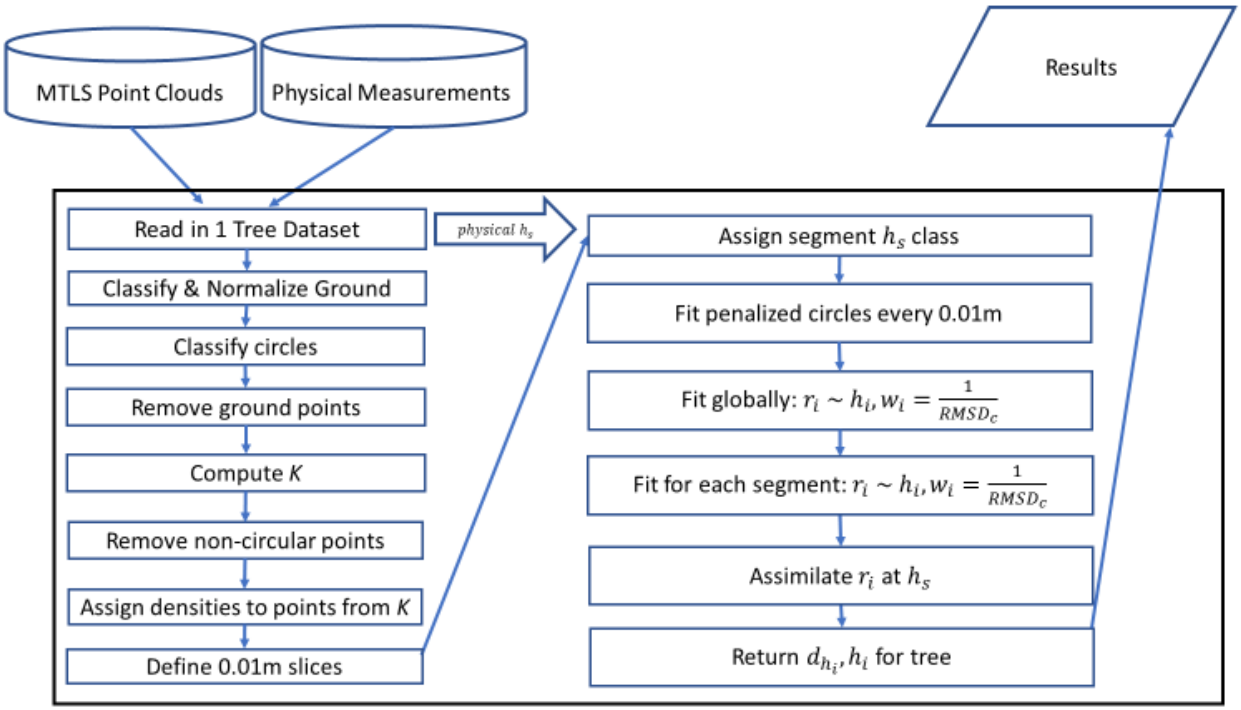

Figure 2.17. Simplified flow diagram of method #2.

Error Analysis

The error in centimeters for each individual segment diameter was defined as: $\epsilon_s = d_s - \hat{d}_s$, with d_s being the known diameter from physical measurements, and $\hat{d}_s$ being the predicted segment radius from a processing method. Therefore, for a positive ϵ_s the MTLS diameter resulted in an underprediction, likewise a negative ϵ_s resulted in an overprediction. For each method, statistics of ϵ_s of interest were root mean square error (RMSE), signed mean error (SME) and the distribution of ϵ_s, and ϵ_s with respect to segment height (h_s). RMSE for each method across all segments was defined as $RMSE_s = \sqrt{\frac{\sum_{s=1}^{n_s} \epsilon_s^2}{n_s}}$, where n_s corresponds to the number of segments in the population of segment measurements by method. Similarly, SME was defined as $SME_s = \frac{\sum_{i=1}^{n}(\epsilon_s)}{n}$, which can be thought of as the mean of residuals. Error by h_s was investigated by the linear model in equation 2.6.

$$\epsilon_s = \beta_0 + \beta_1 h_s \quad (2.6)$$

The β_1 describes the rate of change in ϵ_s with respect to either h_s, with the associated parameter *t* statistic and *P*-value determining if this rate is statistically significant. Lastly, performance in measuring *DBH* was evaluated with the same $RMSE_s, SME_s$ formulations.

Results

Before evaluating ϵ_s, it must be considered that the diameter tape used to take the physical measurements only possessed a resolution of 0.1in, or 0.254cm but was assumed to completely error and bias free. Additionally, the MTLS sensor utilized had a relative accuracy of 1-3cm for each point (see appendix 2). The results on estimating d_s, DBH for each method are presented below. As shown in table 2.1, method #1 outperforms method #2 in accuracy except for SME of DBH, indicating that method #2 might be slightly less biased than method #1 in estimating DBH, although both methods are far from zero in for this measure. However, method #1 exhibits a smaller β_1 bias value of bias as a function of height as is shown in table 2.2. It is readily apparent that the d_s and *DBH* MTLS measurements are biased as a function of height from figures 2.18-2.19 and table 2.2. The d_s measurements must be as unbiased as possible with respect to h_s in order for methods described in chapter 4 to be successfully implemented. Of principle importance was *DBH* as discussed in the introduction- an unbiased *DBH* measurement is crucial for interoperability with current forest modelling systems that assume an unbiased and inerrant *DBH* measurement. For both methods the SME of DBH was positive and far from zero- indicating a bias towards underprediction, although method #2 was slightly better in this category. The estimated DBH is shown against known DBH in figure 2.18. The distribution of all ϵ_s is shown

in figure 2.19 and it is clear that all diameters are slightly biased towards underprediction for both methods because the center of the distribution is to the right of zero.

Table 2.1. $RMSE_s, SME_s$ of d_s and *DBH* for methods #1 & #2.

Method	RMSE (cm) of d_s	RMSE (cm) of DBH	SME (cm) of d_s	SME (cm) of DBH	n_s
#1	2.8074	1.7395	0.0450	1.4282	1578
#2	4.4829	2.9676	-0.3983	1.0815	1724

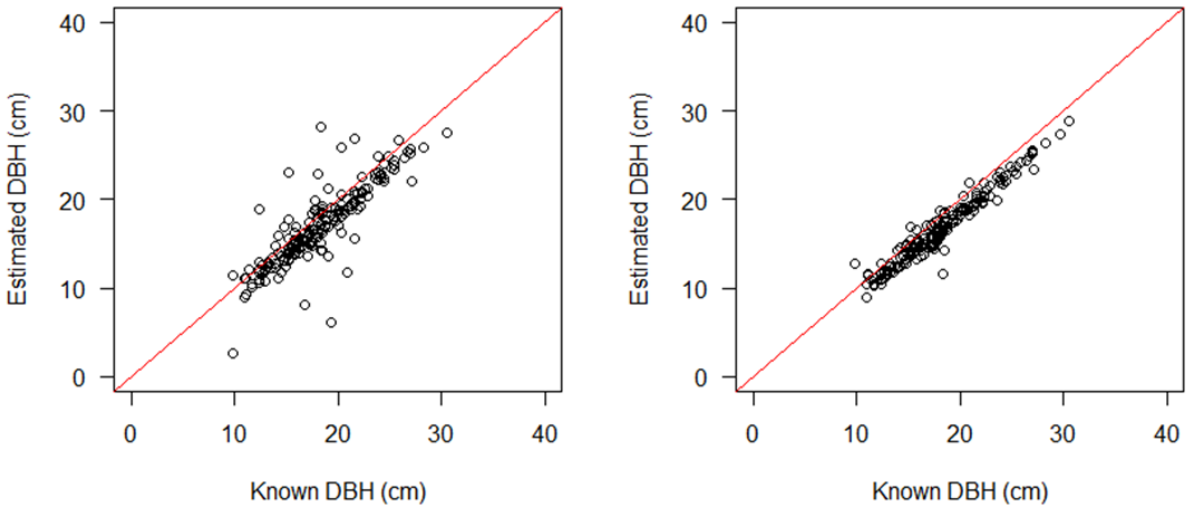

Figure 2.18. Estimated *DBH* vs known *DBH* for method #2 (left) and method #1 (right).

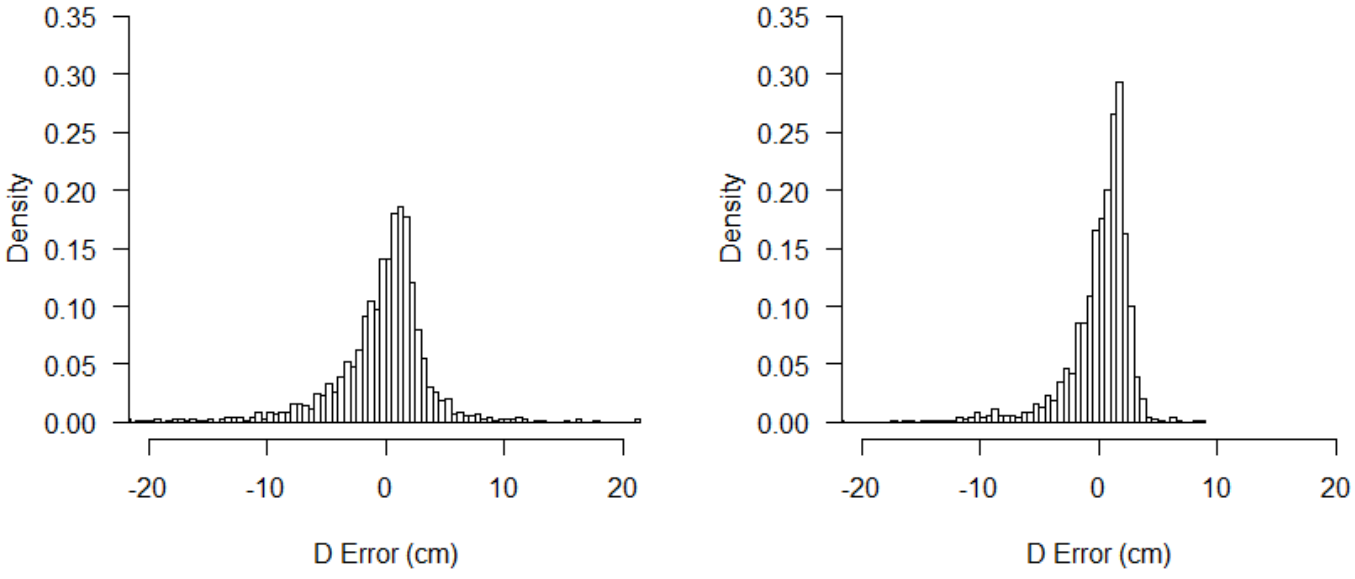

Figure 2.19. Density histograms of all ϵ_s (method #2 left, #1 right).

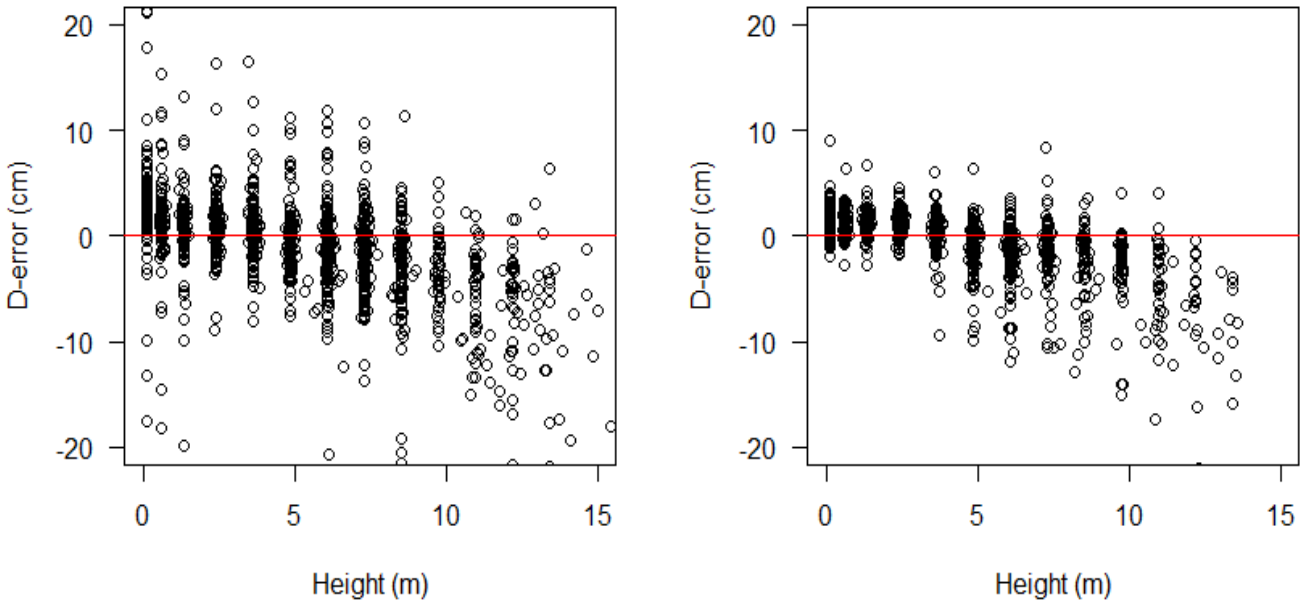

Figure 2.20. Residuals by height with method #2 left and method #1 right.

Table 2.2. Bias statistics- this table describes the information contained in figure 2.20.

Method	β_1 bias cm/m	P of β_1	t-value of β_1	s.e. of β_1	R^2 of model
#1	-0.6016	1.1871×10^{-211}	-36.4686	0.0165	0.4577
#2	-0.7234	1.7351×10^{-150}	-28.9502	0.0300	0.3274

As shown in the above table (2.2), bias as a function of height was clearly seen as a non-negative β_1 value for both methods. This parameter was clearly significant, and even the R^2 of the model pointed to h_s being somewhat useful in explaining the relationship between ϵ_s and h_s. A simple interpretation of the β_1 value is that for method #1 diameter was overpredicted by about 0.6cm per 1m of vertical height. Method #2 behaved in a similar manner but at a rate of 0.7cm per 1m of vertical height. In figure 2.20 a pattern was observed where the MTLS derived lower stem diameters appeared to be underpredicted with upper stem diameters being an overprediction. This is troubling as the β_1 error for both methods will add to an unacceptable amount of bias after just a few meters.

Discussion

In this chapter it was shown that tree diameters can be recovered automatically from MTLS data of single trees. While the $RMSE_s, SME_s$ statistics in table 2.1 are usable, a troubling amount of bias in ϵ_s with respect to h_s was noted to be significantly present. Therefore, uncorrected these methods are not of use. It is unknown if the bias shown in figures 2.18-2.20 result from the tree, sensor or diameter estimation procedure. Determining the source of this bias would make for an interesting study. Additionally, the statistics presented in table 2.1 must be interpreted with caution because the sample size of d_s is very large at 1578 for method #1 and 1724 diameters for method #2. $RMSE_s$ is likely to be higher in a real MTLS inventory, and this effect will be amplified with smaller inventories.

An important aspect to consider when selecting which diameter finding method to use is how segment diameters were defined in the data. In method #1 these are rigidly defined by h_s in the destructively sampled dataset and therefore may exhibit sampling bias. Indeed, in the field h_s was frequently moved to avoid minor stem defects such as branch whorls or rust infections. This may explain the lower $RMSE_s$ exhibited by method #1 as measurement points were essentially hand-picked. In contrast, method #2 utilized the entire region around h_s. Therefore, method #2 utilizes all the data and h_s is not as rigidly defined by the physically sampled dataset and can be said to exhibit less sampling bias. Knowing this, method #2 is likely to be applied to an MTLS inventory with greater success than method #1. This is because the attraction in using MTLS is speedily sampling large numbers of trees, and therefore careful manual selection of measurement points defeats the purpose of using MTLS. Sample sites used in this study were relatively clean of brush compared to typical operational pine plantations. In order to apply the diameter finding methods shown here special care must be taken to isolate trees in the scan. Automated tree

segmentation was not investigated in this thesis. Before application to wide scale pine plantation inventory more testing should be performed with stems that are highly occluded by brush. On the other hand, in the field it was clear than in most of the plantations sampled that significant tree branches were observed even to ground level. These had to be removed from the MTLS point cloud for accurate diameter estimation, which was very highly non-trivial as was shown in the methods section of this chapter.

Conclusion

Diameters were found with usable RMSE values, with method #1 performing better than method #2 in estimating d_s as well as exhibiting a lower β_1, but method #2 will perform better in real inventories because it uses all of the data and in theory exhibits less sampling bias. Method #2 performs better in *DBH* estimation. However, troubling bias in ϵ_s with respect to h_s was observed as significant and non-negative β_1 scores. Therefore, it must be concluded that more work must be done in correcting for this bias. In chapter 3 an ad-hoc bias correction function will be developed and evaluated for method #2 residuals using cross validation.

CHAPTER 3.

GENERALIZING A DIAMETER MEASUREMENT BIAS CORRECTION FUNCTION BY CROSS VALIDATION

Introduction

In chapter 2 a methodology was developed to estimate diameters at different heights using a portable TLS unit. One troubling problem was discovered where ϵ_s was biased with respect to h_s, although somewhat usable $RMSE_s$ values were found. This was described by the significant and nonzero β_1 in table 2.2. and the residuals in figure 2.20, which showed a tendency to under-predict lower stem diameters and over predict upper stem diameters. Similar bias in estimated diameters with respect to height has been reported previously in the literature, notably in Fang & Strimbu (2017) where a simple multivariate model was fit with ordinary least squares to reduce severe bias. Taking inspiration from this, a bias correction function was selected using a linear regression and parameters calibrated using *K*-fold cross validation, where $K = 10$. The motivation behind *K*-fold cross validation was to evaluate model performance on similar but different datasets in situations where total observations are limited. More specifically, the goal of this chapter was to mimic what might occur if these methods are applied to other *Pinus palustris* trees outside of this specific dataset but still in South Georgia, U.S.A. and with a similar MTLS sensor and diameter finding method #2. It can be said that the goal was to generalize the parameters in the bias correction function because the goal was to reduce test error instead of train error (Hastie et al 2009). However, unlike traditional cross-validation where an error statistic (such as mean

squared or mean absolute error) is of interest for the test statistic, the bias statistic β_1 of the form shown in table 2.2 and equation 2.6 was the test statistic of interest. Simply fitting a model to the data without cross validation (e.g., β_1 on all the data) would minimize training error, which can be a poor indicator of test error under the assumption that the relationships of interest in the train and test datasets slightly differ. Ideally, β_1 across *K* folds will have a mean close to 0, or at least significantly less than β_1 in table 2.2. Additionally, the measure of dispersion of β_1 across *K* is standard deviation (S.D.) as it is simple to interpret with respect to the mean of β_1, and a near-zero S.D. should indicate that there is low variance in post-correction bias across *K*. If these conditions hold across *K* then we can be more confident that the model form will be useful on similar but different datasets and methods. Any improvements in $RMSE_s$, and RMSE of *DBH* measurement is a bonus but is still secondary to reducing post-correction β_1 of bias. Additionally, by using only two predictor variables ($\widehat{d_s}, h_s$) either alone or combined, model selection was greatly simplified because the resulting models should be robust and resistant to overfitting because of the amount of data available. Lastly, only diameter measurement method #2 will be used in this chapter, with the rationale being that it probably exhibits less sampling bias and therefore should be more appropriate for a practical MTLS inventory.

Methods

Error and Bias Analysis Review

The error, ϵ_s, in centimeters for each segment radius was defined in equation 3.1.

$$\epsilon_s = d_s - \hat{d}_s \quad (3.1)$$

Where d_s was the known diameter from physical measurements, and $\hat{d}_s$ was the predicted segment radius from MTLS diameter measurement method #2. Statistics of ϵ_s of interest were

root mean square error (RMSE), the distribution of ϵ_s, ϵ_s with respect to segment height (h_s), and ϵ_s with respect to d_s and $\widehat{d_s}$. RMSE for each method across all segments was defined as $RMSE_s = \sqrt{\frac{\sum_{s=1}^{n_s} \epsilon_s^2}{n_s}}$, where n_s corresponds to the number of segment diameters. Similarly, signed mean error (SME) was defined as $SME_s = \frac{\sum_{i=1}^{n}(\epsilon_s)}{n}$, which can be thought of as the mean of residuals. Error by h_s and d_s were investigated by linear models, where β_1 principally describes the rate of error change.

$$\epsilon_s = \beta_0 + \beta_1 h_s \quad (3.2)$$

$$\epsilon_s = \beta_0 + \beta_1 d_s \quad (3.3)$$

The associated parameter *t* statistic and *P*-value determines if this rate was statistically significant. Performance in measuring DBH was evaluated from other diameters with the same $RMSE_s$, SME_s criteria, but only on the diameter closest to 1.37m in h_s for each tree. This definition was due to *DBH* being frequently moved due to defects in the physically measured data. The $RMSE_s$, SME_s statistics are only computed after the best correction model is selected and applied to all of the data.

Cross Validation

Using a *K*-fold cross validation scheme, where $K = 10$, these three models were fit to a training dataset and then evaluated on a test dataset. An important distinction is that $!K$ were the training observations and K were the test observations. Since $K = 10$, the procedure was repeated 10 times by iterating through the data. These datasets came from all the estimated diameters from method #2. The three models to be tested are shown in equations 3.4-3.6. After correction these

models were evaluated with the models described in 3.2-3.3. Thus, three models are tested with two criteria each. The process is described in the rest of this section for a single fold.

$$M1 = \epsilon_s = \beta_0 + \beta_1 h_s \mid !K \quad (3.4)$$

$$M2 = \epsilon_s = \beta_0 + \beta_1 \widehat{d_s} \mid !K \quad (3.5)$$

$$M3 = \epsilon_s = \beta_0 + \beta_1 h_s + \beta_2 \widehat{d_s} \mid !K \quad (3.6)$$

This can be interpreted as segment diameter error, ϵ_s, is a linear function of a variable (h_s, d_s or both h_s and d_s in the case of $M3$) using β's found for the training dataset $!K$. The division of the train & test data is described in 3.1. The fold moves for a total of 10 iterations through the data.

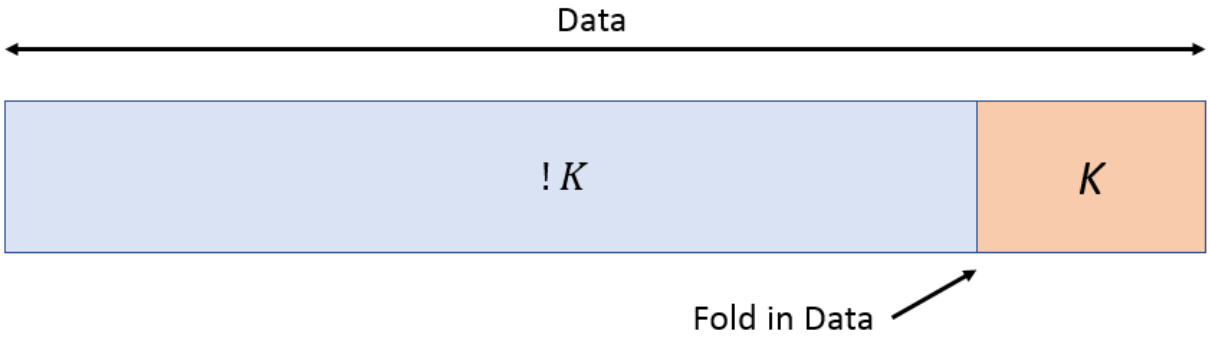

Figure 3.1. Graphical representation of *K*-fold cross validation. The red section, *K*, is the test split of the data with $!K$ train split being represented by the blue section. The division of the datasets is the fold, and is iteratively moved across the data space.

After being fit to $!K$, the models were predicted onto K as shown in equations 3.7-3.9. E.g., the parameters found for equations 3.4-3.6 and data $!K$ were used but with predictor variables from *K* plugged into the equations.

$$\widehat{d_{c1}} = M1 \mid K + \widehat{d_s} \mid K \quad (3.7)$$

$$\widehat{d_{c2}} = M2 \mid K + \widehat{d_s} \mid K \quad (3.8)$$

$$\widehat{d_{c3}} = M3 \mid K + \widehat{d_s} \mid K \quad (3.9)$$

The result can be thought of as taking the original uncorrected MTLS measured diameters, $\widehat{d_s}$, and adding the correction to $\widehat{d_s}$ in the test dataset, K, using parameters found for the test dataset $!K$. This was the post-correction error or test error for a single fold. This was repeated 10 times without replacement.

Evaluating Trained Models on Test Data

For each fold, $M1, M2, M3$ were predicted onto K as $\widehat{d_{c1}}$, $\widehat{d_{c2}}$, $\widehat{d_{c3}}$, and were evaluated on the test dataset *K* to check for bias as a function of height or diameter. The criteria was the same as described in equations 3.2 & 3.3, but it describes the post-correction relationship of error on the K dataset for each model and the variables h_s, $\widehat{d_s}$, resulting in two criteria each for 3 possible models. This can be thought of as β_1 being the pre-correction bias statistic with respect to either h_s or d_s with α_1 being the corresponding post-correction bias statistic. This is described in equations 3.10-3.15.

$$d_s - \widehat{d_{c1}} \mid K = \alpha_0 + \alpha_1 h_s \mid K \quad (3.10)$$

$$d_s - \widehat{d_{c1}} \mid K = \alpha_0 + \alpha_1 d_s \mid K \quad (3.11)$$

$$d_s - \widehat{d_{c2}} \mid K = \alpha_0 + \alpha_1 h_s \mid K \quad (3.12)$$

$$d_s - \widehat{d_{c2}} \mid K = \alpha_0 + \alpha_1 d_s \mid K \quad (3.13)$$

$$d_s - \widehat{d_{c3}} \mid K = \alpha_0 + \alpha_1 h_s \mid K \quad (3.14)$$

$$d_s - \widehat{d_{c3}} \mid K = \alpha_0 + \alpha_1 d_s \mid K \quad (3.15)$$

Thus, the new errors were of the same form as used in equations 3.2 & 3.3, but post-correction for that fold. This could be interpreted as the post correction error for each correction model form as a function of either h_s or d_s. After fitting models 3.10 – 3.15 the post-correction bias statistics, mean and standard deviation of α_1, for each model across all folds were evaluated. Ideally a mean α_1 value was expected be around 0 with a small standard deviation, with α_1 being a measure of post correction bias and S.D. of α_1 a measure of generality. After that, the corrected diameters of the best model were compared to the original diameters (d_S) to show if bias was reduced with particular interest in minimizing bias with respect to height. Lastly, final $RMSE_s$ was be computed after corrections were applied where the parameters of the chosen correction function were the mean of parameters across K. However, it is critical to recognize that the goal was minimizing α_1 and any reduction of $RMSE_s$ was a bonus. Figure 3.2 is a simplified summary of the process described in the above section.

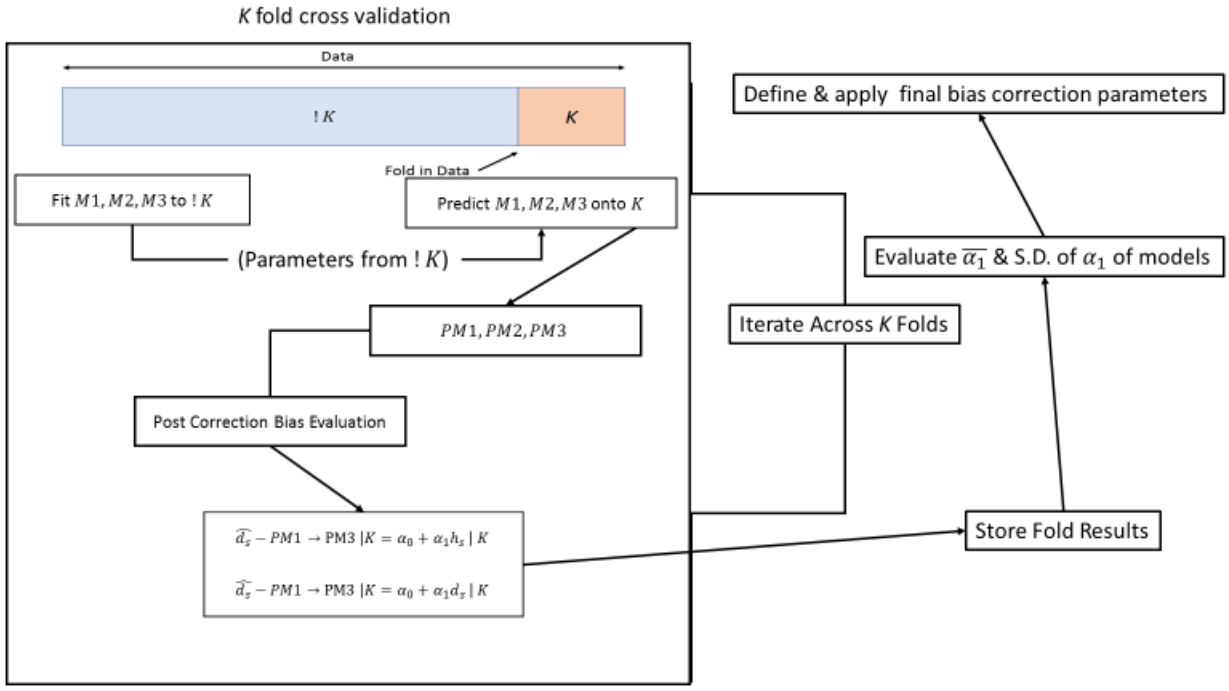

Figure 3.2. Simplified flow diagram of the cross-validation bias correction function development procedure described in the above methods.

Results

Correction model $M1$ was selected as the best bias correction model form. While $M3$ had the lowest $\overline{\alpha_1}$ and associated S.D., it badly biased *DBH* with only a very marginal improvement in bias of all ϵ_s. See table 3.1 for these statistics. See figure 3.3 for the 1-1 plot of $M3$ corrected *DBH* values to see the clear biasedness.

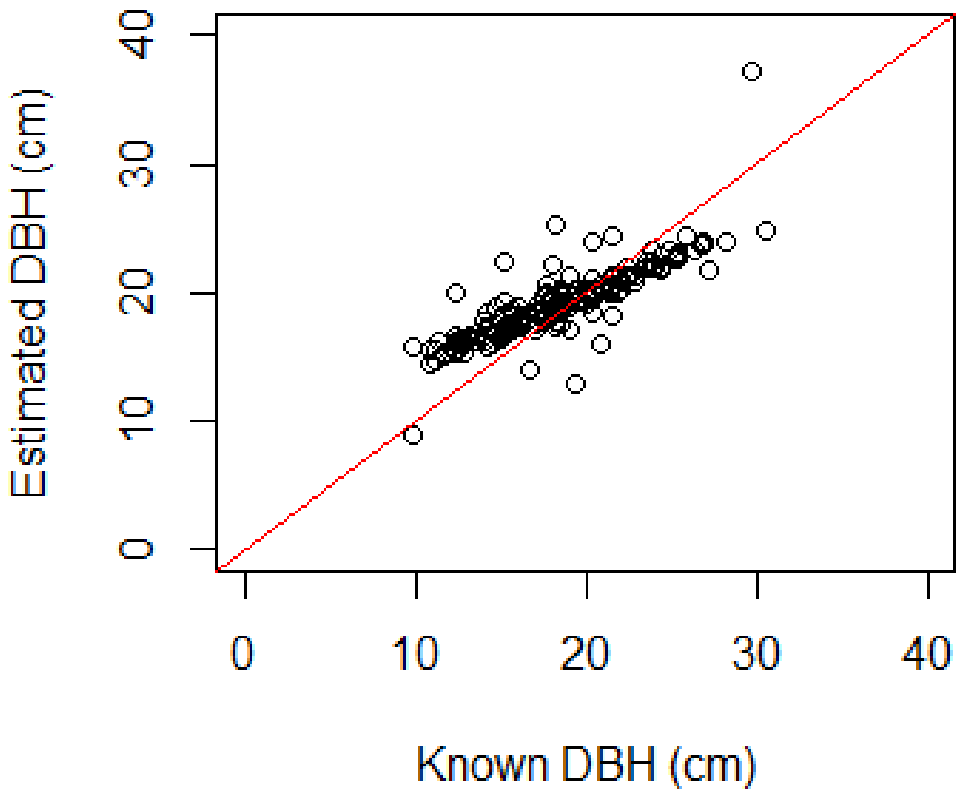

Figure 3.3. *DBH* 1-1 graph of known vs predicted with the $M3$ corrections applied.

The choice of $M1$ over $M3$ is a classic example of why more than just statistics may be needed to make a decision- clearly the behavior shown in figure 3.3 is unacceptable. It is likely that in $M3$ that the h_s predictor contributed to the improvement much more than the $\widehat{d_s}$ predictor, with $\widehat{d_s}$ introducing bias in *DBH* while contributing no meaningful gain. Considering the importance of *DBH*, such blatant bias is unacceptable, especially considering that it is worse than what was observed in chapter 2 for the same relationship. Additionally, using $M2$, (bias as a

function of estimated diameter) did not provide meaningful improvement in reducing bias as a function of diameter and actually worsened bias as a function of height. In contrast, *M*1 reduced bias as a function of height to a more useful level as compared to the pre-correction bias while improving *DBH* biasedness over what was shown in chapter 2 in terms of SME.

Table 3.1. Comparison of bias correction model performance- for diameters from method #2, $\overline{\alpha_1}$ describes the bias after correction across the cross-validation folds, similarly to the β_1 in table 2.

Post-Correction Criteria Function	Method #2 Diameters	
	$\overline{\alpha_1}$ across K	S.D. of $\overline{\alpha_1}$
$d_s - \widehat{d_{c1}} \mid K = \alpha_0 + \alpha_1 h_s \mid K$	-0.0146 (cm/m)	0.1559
$d_s - \widehat{d_{c1}} \mid K = \alpha_0 + \alpha_1 d_s \mid K$	0.1157 (cm/cm)	0.0959
$d_s - \widehat{d_{c2}} \mid K = \alpha_0 + \alpha_1 h_s \mid K$	-0.8144 (cm/m)	0.1046
$d_s - \widehat{d_{c2}} \mid K = \alpha_0 + \alpha_1 d_s \mid K$	0.6285 (cm/cm)	0.0763
$d_s - \widehat{d_{c3}} \mid K = \alpha_0 + \alpha_1 h_s \mid K$	-0.0120 (cm/m)	0.1089
$d_s - \widehat{d_{c3}} \mid K = \alpha_0 + \alpha_1 d_s \mid K$	0.1089 (cm/cm)	0.0633

The following figures and tables describe in detail the improvement over uncorrected diameters that using *M*1 as a correction method allowed for. Notice how ϵ_s as a function of h_s is reduced, with ϵ_s now being centered about zero, and estimated *DBH* is no longer biased towards under-prediction.

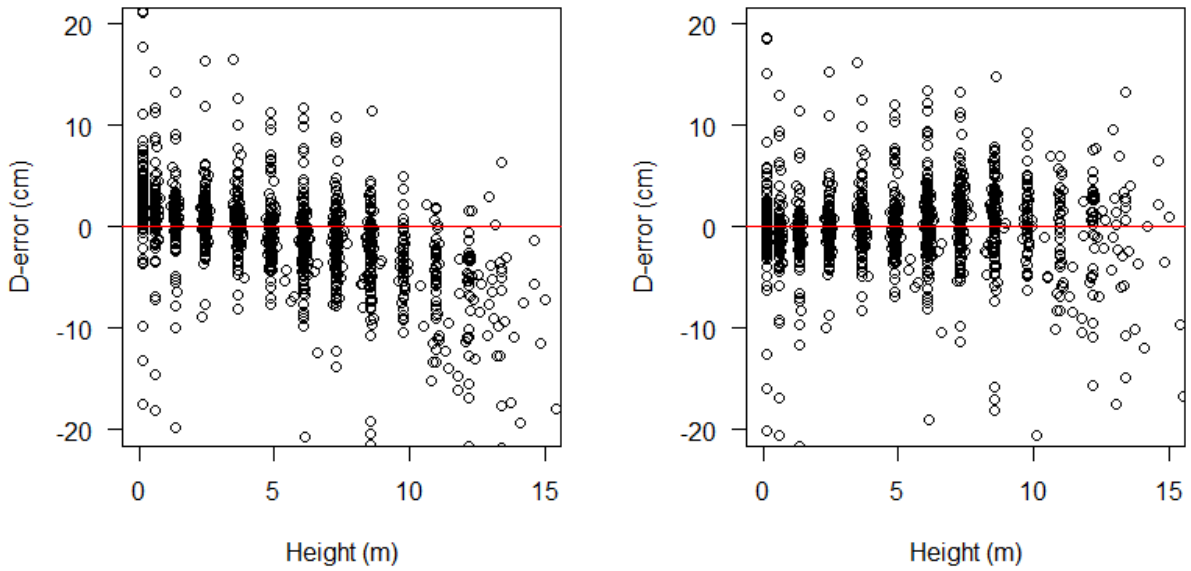

Figure 3.4. Pre-correction (left) vs post-correction (right) with of Method #2 residuals $M1$ where ϵ_s is corrected with respect to h_s.

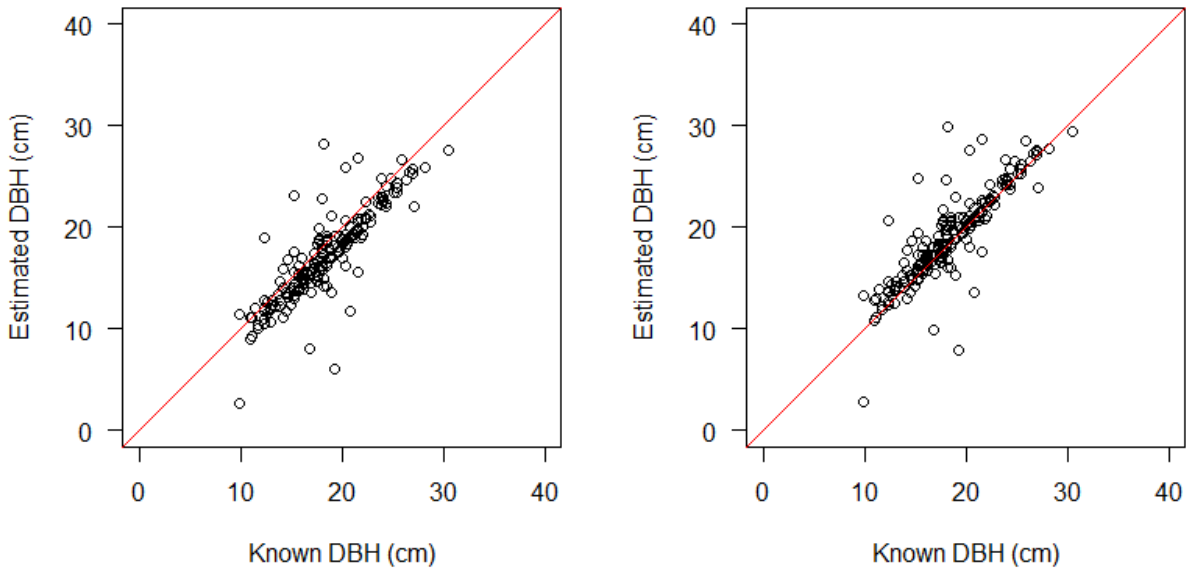

Figure 3.5. 1-1 graph of DBH pre-correction (left) vs post-correction (right) with $M1$ where ϵ_s is corrected as a function of h_s for residuals of method #2.

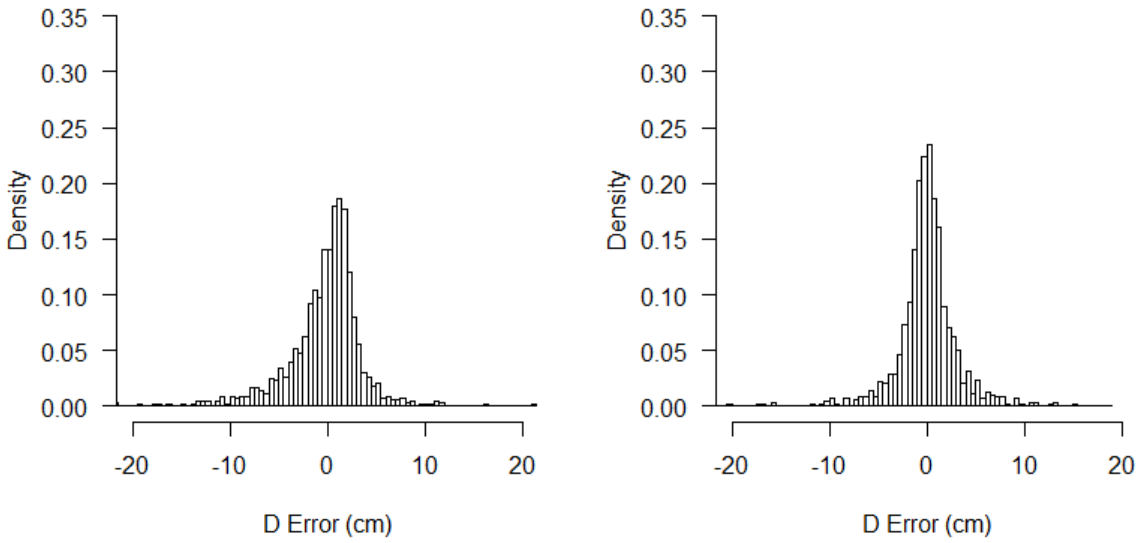

Figure 3.6. Histogram of all ϵ_s, pre-correction (left) vs post-correction (right).

In table 3.2 the post correction bias (ϵ_s with respect to h_s) was reduced to being very nearly 0. Additionally, the associated statistics to β_1 point to h_s alone being an insignificant indicator of ϵ_s, quite an improvement over the same statistics in table 2.2. Bias as a function of height has been eliminated. Additionally, it can be seen in table 3.3 that RMSE and SME has been improved over the corresponding statistics shown in table 2.1.

Table 3.2. Bias statistics post-correction with *M*1 bias correction model for method #2 diameters.

Method	β_1, bias cm/m	P of β_1	t-value of β_1	s.e. of β_1	R^2 of model
#2, *M*1	1.1727×10^{-5}	0.9996	0.0005	0.0250	1.2789×10^{-10}

Table 3.3. Post-correction $RMSE_s$ for each method using bias correction *M*1 where parameters are found through cross-validation.

Method	RMSE (cm) of DBH	SME (cm) of DBH	RMSE (cm) of d_s	SME of d_s
#2, *M*1	2.8752	-0.7166	3.6621	-0.0006

It can be seen in tables 3.3 that in the process of removing bias as a function of height the $RMSE_s$ and SME_s was slightly reduced over what was shown in table 2.1. However, these

statistics must be considered with caution as the sample size was very large at $n_s = 1724$ for all diameters and 185 for DBH.

Discussion

In this chapter it was found that a bias correction function of the form *M*1 (equation 3.4) was successfully used to remove bias in ϵ_s with respect to h_s with cross-validated parameters. Additionally, RMSE and SME of all segment diameters as well as *DBH* was reduced as a secondary effect. It appears that using $\widehat{d_s}$ as a predictor for ϵ_s was not successful because it drastically worsened performance in estimating *DBH* despite producing a slightly smaller α_1 with respect to h_s as compared to only using h_s as the predictor variable in correcting bias. In terms of RMSE of *DBH*, the best traditional diameter measurement tool analogue would be the Biltmore Stick, which is typically subtended to return diameters to a 1 inch resolution, which approximately corresponds to the 2.9 cm RMSE of *DBH* resulting from method #2 after correction with $M1$.

None of the literature reviewed utilized a cross-validated bias correction function for this specific problem of bias in ϵ_s with respect to h_s. An approach using several predictor variables and only ordinary least squares was described in Fang & Strimbu (2017) but with more limited results. The TLS literature reports a wide range of RMSE values for diameters. Most comparable in methodology would be Hyyppa et al (2020), where an RMSE of 0.6cm in DBH was reported. This is particularly interesting because in that study and this thesis an MTLS sensor is used unlike other studies. In terms of RMSE, the most similar study is Srinavasan et al (2015) where an RMSE of about 1.8-2.1cm of DBH estimation was achieved. In some studies, such as Pitkanen et al (2019), better RMSE's of DBH of 0.7cm were reported, but upper stem diameters had an RMSE of 0.6-0.8cm, showing more error at height. This same phenomenon is shown here, where DBH

has a lower RMSE than other diameters despite being a smaller sample size. Interestingly the tendency to underpredict lower stem diameters shown in chapter 2 is not new- it was observed by Thies et al (2004), albeit with a different sensor and in a very different forest type. Despite the high RMSE's in this chapter, in Moskal et al (2012) an RMSE of over 10cm was reported, so the results of this chapter certainly are not the worst. More similarly to this study, Liu et al (2018) reported an RMSE of 1.2cm for DBH, while Henning & Radtke (2006) reported 1-2cm average errors in diameters, with error increasing at greater heights and a general bias towards over-estimation of diameters. Lastly, Cabo et al (2018) reported an RMSE of 0.8-1.3cm. All of these studies mentioned bias in estimated diameters- typically towards underprediction, as was seen in chapter 2. However, upon reading this literature, none of the trees were explicitly described to have severe branching at ground level. Therefore, it is difficult to compare the results of this study with results of the literature because the noise that was contended with by others is unknown, while the noise (i.e., non-stem points) was extremely high in the data used for this thesis. Bias as a function of height was not explored in literature as explicitly as in this thesis except for perhaps in Fang & Strimbu (2017), although the post-correction bias and error is higher in that study than in this thesis. In Pitkanen et al (2019), DBH estimation from TLS showed slight bias to under-prediction with a bias towards uppers stem diameter over estimation, although no correction model was proposed. Summarizing this, the methods presented in chapter 2 & 3 are competitive with existing methods in the literature. Most of the existing literature is solely focused on DBH measurement, with direct height measurements often being a secondary goal. Since the goal of this thesis was to develop an indirect height estimation procedure using an inverse taper model and multiple stem diameters, development of a diameter estimation method exhibiting minimal bias as a function of height was necessary, with $RMSE_s$, SME_s being secondary goals. Additionally, the

MTLS datasets used here could be thought of as "worst-case" datasets because the *P. palustris* trees that were sampled almost all exhibited significant branches in the lower stem, creating large amounts of noise to be removed before stem diameters could be measured. Lastly, the MTLS sensor used in this thesis was mobile and handheld as opposed to being large and tripod mounted (as used in most literature reviewed), leading to the classic tradeoff in speed vs accuracy.

In performing this cross-validation procedure an important question must be asked about the true generality of this bias correction function. It is proposed that this bias correction function is likely to be appropriate for data collected under similar circumstances in similar stands to the data in this study, where the same diameter estimation procedure and MTLS are used. However, it is not known if a new bias correction function will be needed for different stands, species, or MTLS units. If this function and diameter finding method work for other species then perhaps it can be said that the bias results from the diameter finding method or sensor itself. These are important questions that merit a study of their own. Lastly, perhaps there is a better diameter measurement procedure than described in chapter 2 that will eliminate the need for a bias correction function regardless of sensor or forest type- a solution that proved to be elusive.

Conclusion

In this chapter a bias correction function for *P. palustris* in South Georgia, U.S.A., was developed for residuals from MTLS diameter measurement method #2 described in chapter 2. This bias correction function successfully reduced the post-correction β_1 from equation 3.2 to near zero as well as reducing $RMSE_s$, SME_s of both *DBH* and all d_s. Thus, these corrected measurements are now more useful for inventory purposes as well as height prediction using methods to be shown in chapter 4.

CHAPTER 4.

ESTIMATION OF TREE TOTAL HEIGHT WITH MULTIPLE TERRESTRIAL LASER SCANNER DERIVED DIAMETERS BY NUMERICALLY SOLVING A TAPER MODEL FOR HEIGHT

Introduction

Tree height is necessary for computation of tree volume and for characterizing forest productivity when repeated height measurements are available. In traditional forest inventory total tree height (*H*) is measured relatively infrequently as compared to diameter at breast height (*DBH*), where breast height is 1.37m, and tree count per unit area. With these limited height measurements, a relationship between *DBH* and *H* is often established using a non-linear relation, to allow for later *H* imputation using the remaining diameters in the inventory (Avery & Burkhart 2002). The process is well established among forest practitioners who are willing to accept the level of uncertainty present in this kind of estimate. An inventory done with a mobile terrestrial laser scanner (MTLS) as presented in this thesis provides a series of advantages in forest inventory but with some limitations in the height measurement process. No scanner guaranteed to see the top of the trees, and therefore would provide a biased estimator for the total tree height if the highest point is assumed to be the top of the tree. In spite of that, the MTLS provides the advantage of measuring not only one but several diameters height pairs in a tree (d_h, h), opening the possibility of using multiple diameters to estimate height.

Incorporating d_h, h measurements in forest inventory is not new- physical measurement by ladder and tape or by ocular estimation were perhaps first suggested by Mesavage & Girard in 1946 for determining form class. More recently d_h, h measurements have been used to improve taper model diameter prediction (Kozak 1998). This has been extended to the taper model that is the focus of this chapter- the Max and Burkhart (1976) (MB76) model by developing algebraic constraints such that the taper model must pass through a d_h, h measurement (Cao 2009), as well as models that adjust taper parameters (Trincado & Burkhart 2006) from d_h, h measurements on a tree-wise basis (Sabatia & Burkhart 2015). Previous application to taper models have used just a single d_h, h that was measured with more traditional tools than TLS, with varying results in terms of improvement and ideal height at which to take d_h, h measurements. Increasing d_h, h measurement error can affect volume prediction error, and under certain conditions was found to be impractical considering the difficulties of traditional d_h, h measurement methods (Arias-Rodil et al 2017). Additionally, a taper model has been proposed that incorporates many d_h, h measurements into a system of linear equations (Kilkki et al 1978). However, it is unknown if several $d_h,\ h$ pairs have been explicitly used to predict height until now.

The objective of this chapter was to develop a method of using multiple MTLS derived d_h, h pairs for height estimation in such a way that *H* need not be directly measured in an inventory using MTLS or a clinometer. To be successful this model must approximate the performance of existing methods while also being unbiased.

Methods

In the proposed method of estimating total height from a taper model and several d_h, h measurements from MTLS, several assumptions were made. First, the model form and parameters

of a taper model were assumed to be a good representation of tree taper. Second, tree shape was assumed to be related to tree height. Third, the paired d_h, h measurements were assumed to be unbiased and have minimal error. Lastly, the parameters of the taper model were found for a population of trees representative of the population of trees for which height was to be estimated from measured d_h, h pairs. Lastly, a common model (Schumacher 1939) was used as an accuracy benchmark to compare the taper model heights to, for which heights were predicted from *DBH* as in a traditional forest inventory.

Taper & TLS Data Used

The physically measured data used was from a dataset of 400 *P. palustris* trees that were felled and measured in the summer of 2019 in South Georgia, U.S.A. Of these 400 trees, 324 were retained as being non-forked trees because taper models are designed for excurrent trees. Trees were measured for *H* and *DBH* before felling, with measurements of d_h, h being taken after felling at regular intervals up the tree as described in chapter 2. 185 of these trees had paired MTLS measurements from method #2 as shown in chapter 2, with the bias corrected as shown in chapter 3.

Establishing a Benchmark- The Schumacher 1939 Model

The Schumacher (1939) (S39) model, shown in equations 4.1 and 4.2, is commonly used to model the relationship between *H, DBH* pairs in a forest inventory, and then impute *H* for the rest of the measured *DBH* in the inventory. The parameters are simple to interpret, and the model generally assumes the shapes of most *H, DBH* relationships. Additionally, this model can be log-transformed to become linear in the parameters, which allows for a direct least squares solution for the parameters- a useful feature in automated inventory data processing programs.

$$H = \beta_0 e^{\left(\frac{\beta_1}{DBH}\right)} + \epsilon \quad (4.1)$$

Where β_0, β_1 are parameters found through least squares estimation for the selected 324 trees in the destructive sampled dataset. A common variation of this model has a transformed response variable to account for the fact that *DBH* is not measured at ground level and that therefore we must know that the tree is at least 1.3m tall.

$$H - 1.3m = \beta_0 e^{\left(\frac{\beta_1}{DBH}\right)} + \epsilon \quad (4.2)$$

Equations 4.1 and 4.2 were fit to the trees with direct ground measurements in the TLS dataset, with 4.1 performing better with a root mean square error (RMSE) of 2.1m as opposed to 2.5m when fitted with equation 4.2. Therefore, equation 4.1 was used as the benchmark for further comparisons. Note that equation 4.1 was fit to all of the data- in application often only a few trees are measured for height, with these few *DBH*, *H* paired observations forming the relationship necessary to apply an equation like 4.1 to the rest of the DBH observations.

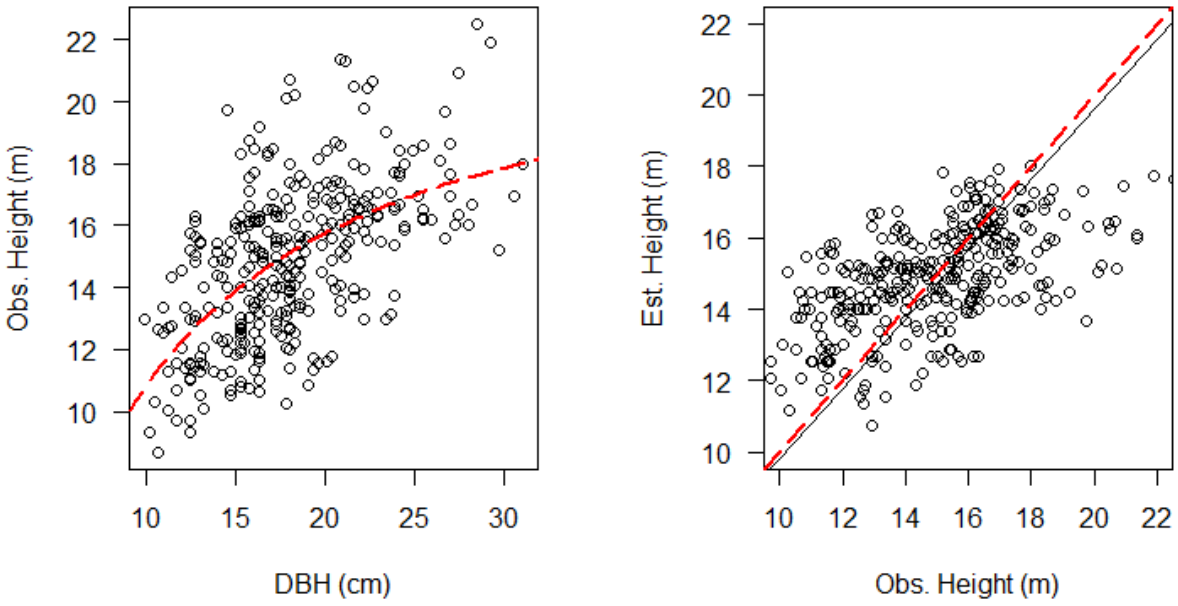

Figure 4.1. *H* as a function of *DBH* (left) with fitted S39 model plotted, and estimated vs fitted heights (right) with the red line being the reference 1-1 relationship and the black line fitted.

The Taper Model

Most taper models in the forestry literature can be summarized in the form presented in equation 4.3. These models take *D*, *H* and a given *h* to return $d_{t|h_i}$ subject to a function *f* and parameters θ with some error ϵ.

$$\widehat{d_{t|h_i}} = f(h_i, D, H, \theta) + \epsilon \quad (4.3)$$

The exact form of *f* and the methods for obtaining θ vary widely in the forestry literature, and are perhaps the most studied topics in forest biometrics. However, most taper models can be generalized to equation 4.3 because they all return an upper stem diameter subject to the desired height of diameter, *DBH*, and *H*. Equation 4.3 is simple to understand with a geometric description of taper model variables and data as shown in figure 4.2.

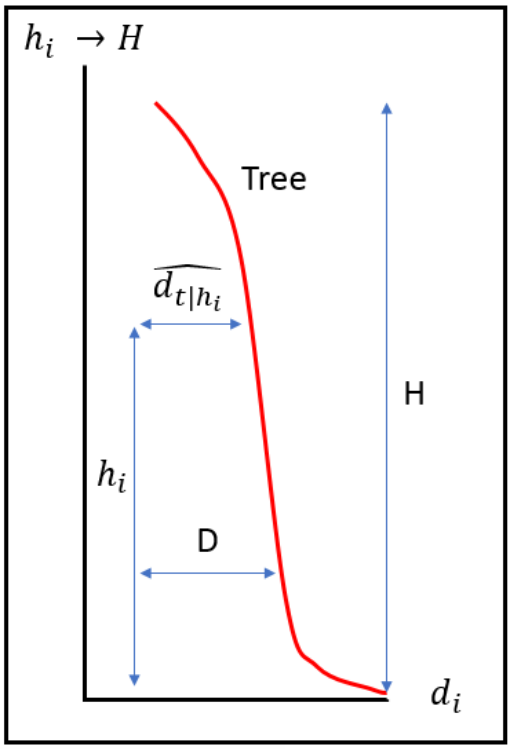

Figure 4.2. Geometric description of taper model variables and data.

The Max & Burkhart Segmented Polynomial Taper Model

The Max & Burkhart segmented polynomial taper model (MB76) is a popular taper model formulation in the biometrics literature, with a well thought out derivation that is based on observations of tree geometry. In this model a tree is modeled with a frustrum of a neiloid representing the butt section, a paraboloid for the mid bole, and a cone for the tip section (Burkhart & Max 1976).

$$\frac{d_h}{DBH} = \beta_1\left(\frac{h}{H} - 1\right) + \beta_2\left(\frac{h^2}{H} - 1\right) + \beta_3\left(\alpha_1 - \frac{h}{H}\right)^2 I_1 + \beta_4\left(\alpha_2 - \frac{h}{H}\right)^2 I_2 + \epsilon \quad (4.4)$$

Where $I_1 = 1 \ if \ \frac{h}{H} \leq \alpha_1, \ = 0 \ otherwise$ and $I_2 = 1 \ if \ \frac{h}{H} \leq \ \alpha_2, = 0 \ otherwise.$ The I_1, I_2 operators are indicator variables that "turn off" different parts of the equation depending on what part of the tree in terms of relative height the predicted diameter will occur. It was clear that the MB76 model cannot be solved analytically for *H*. Therefore, a numerical solution was developed to solve for *H*. Lastly, this model can be summarized by equation 4.3.

Numerically Solving a Taper Model for Height

Recalling equation 4.3 for brevity, understanding how any taper model, including MB76, can be solved for *H* was simple. Given a known $d_{l|h_i}, \ h, \ DBH$ measurement for a tree and fixed θ's determined from a conventional taper model parameter fit, predicted height, $\widehat{H}$, was found by varying ω_k (substituted for *H* into equation 4.3) in the taper model until the expression $\left(d_{t|h_i} - d_{l|h_i}\right)^2$ was minimized, where $d_{t|h_i} = f(\omega_k, DBH, h_i, \theta)$ and $d_{l|h_i}$ was the TLS measured diameter with *i* measurements in a single tree. This is shown in equation 4.5.

$$argmin \ \omega_k \in \mathbb{R}^n = \ (f(\omega_k, DBH, h_i, \theta) - d_{l|h_i})^2 \quad (4.5)$$

Where $\omega_k \approx H$ of a single tree when the right-hand side of equation 4.5 approaches 0. A simple way of conceptualizing this is for each $d_{l|h_i}$, h_i, DBH combination for each tree, a convex minimization problem was solved. A new ω_k from a pre-defined sequence of *k* length was tested until the squared distance between the taper model predicted diameter, $d_{t|h_i}$, and the TLS measured diameter, $d_{l|h_i}$, was minimized. Additionally, as a simple error checking step, equation 4.5 was also solved where $\omega = 0, 50$ to test for unreasonable $d_{l|h_i}, h_i$ pairs because trees are no 0m tall, and 50m was far taller than the expected tallest tree in the data. ω_k was defined as starting at the highest h_i in all $d_{l|h_i}, h_i$ pairs for a tree and was incremented by 0.2m to a reasonable maximum height of 35m. The core assumption of this approach was that tree shape and tree height were related, with the additional assumption that the selected taper model and parameters represented the shape of the tree appropriately. E.g., θ in f was found for a sample representative of the trees sampled with MTLS. Since H is implied to be a predictor of tree shape in equations 4.3 and 4.4, it can be said that this method is an inverse taper model because tree shape was assumed to be a predictor of H. An example in pseudo code is shown in figure 4.3, as well as a graph of a typical objective function in figure 4.4.

for each tree

for each $d_{l|h_i}$, h_i **pair** in each tree while holding DBH constant

for each $\omega_k \in \mathbb{R}^n$ where $\mathbb{R}^n$ is bound between max h_i of all $d_{l|h_i}$ measurements in the tree and a ω_k that is a reasonable maximum H given prior knowledge of stand characteristics, compute $\left(f(\omega_k, DBH, h_i, \theta) - d_{l|h_i}\right)^2$. ω_k is iterated in 0.2m intervals in this case, to a reasonable maximum H of 35m

return ω_k where function is minimized.

if ω_k = **an extreme value**, remove height estimation as un-reasonable. In this case, 0m and 50m were used as extreme values.

Compute an average of remaining ω_k for a single tree height, assign as $\widehat{H}$

Return $\widehat{H}$ **then iterate to next tree**

Figure 4.3. The height estimation in pseudo-code.

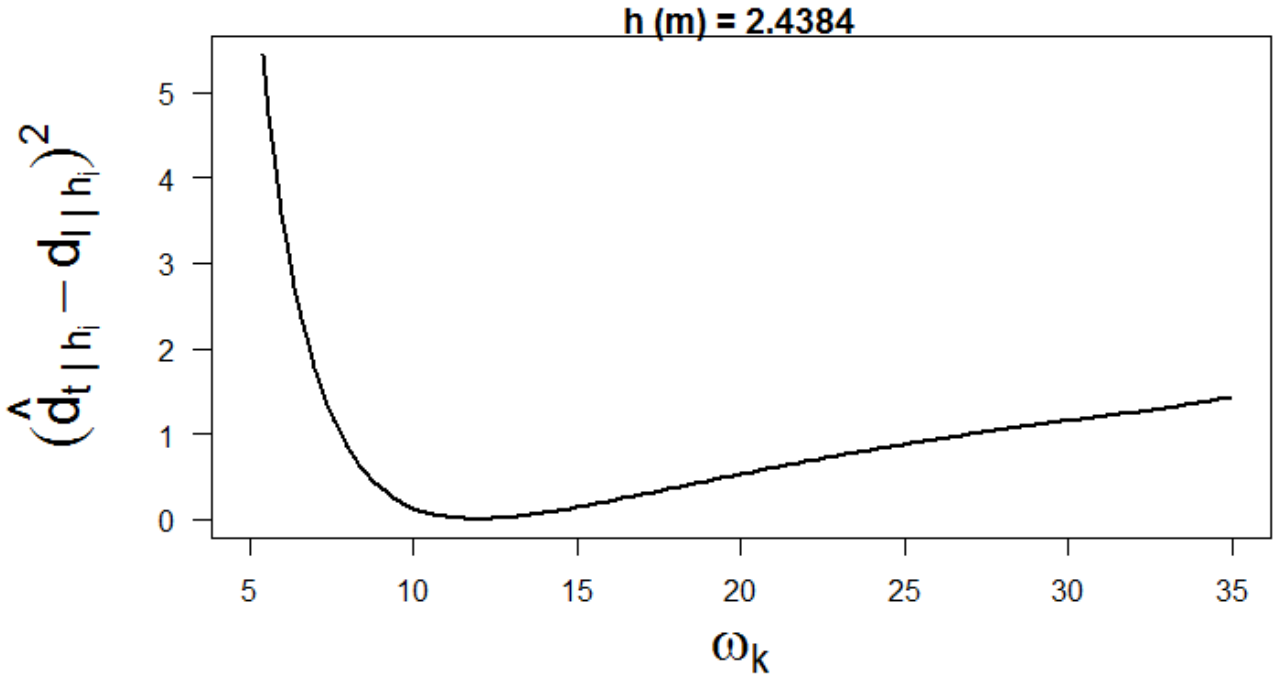

Figure 4.4. Graph of an objective function evaluated across the ω vector.

Weighting Estimated Heights

For each tree at least two or more $d_{l|h_i}, h_i$ pairs were available in the MTLS data. However, for all pairs in a tree, a different height solution is usually found. Therefore, a comparison was made between a simple average as well as weighted averages of these pairs because it was unclear which single $d_{l|h_i}, h_i$ best predicts height for an individual tree. Two weight variables of interest are investigated- h_i of each $d_{l|h_i}, h_i$ pair in the tree and the objective function value (tolerance) from equation 4.5. By using a simple weight and an inverse weight for h_i a simple test to determine whether or not $d_{l|h_i}, h_i$ pairs at a greater h_i are more useful for predicting *H* was created. Lastly, the inverse of equation 4.5 when minimized (convergence) weight was conceived out of the assumption that a taper model that accurately represents tree shape might also accurately represent tree height, and therefore a very small convergence tolerance may mean that the taper model diameter very closely approaches a $d_{l|h_i}$, h_i pair. The weight equations are shown in table 4.1.

Error Analysis Approach

Error analysis in this case was simply in terms of RMSE (m) of height prediction, where $RMSE_H = \sqrt{\frac{\sum_{i=1}^{n}(H_i - \widehat{H_i})^2}{n}}$, where n corresponds to the number of trees and signed mean error $SME_H = \frac{\sum_{i=1}^{n}(H_i - \widehat{H_i})}{n}$ which can be thought of as the mean of residuals. Furthermore, predicted heights were plotted against known heights with the model $\widehat{H_i} = \beta H_i$ fitted to each set of residuals and plotted over the 1-1 plots. A β value of exactly 1 would indicate an exact 1-1 relationship between estimated and observed heights, so therefore the method with β nearest to 1 is the least biased.

Results

Taper Model Performance for Diameter Prediction

Before considering height prediction performance it was necessary to consider the performance of these taper models on diameter prediction on the given data as that is what these models were originally designed to do. After fitting the MB76 model to the data, the MB76 model returned an RMSE of 0.6975 cm in diameter prediction and exhibited a good fit as shown in figure 4.2.

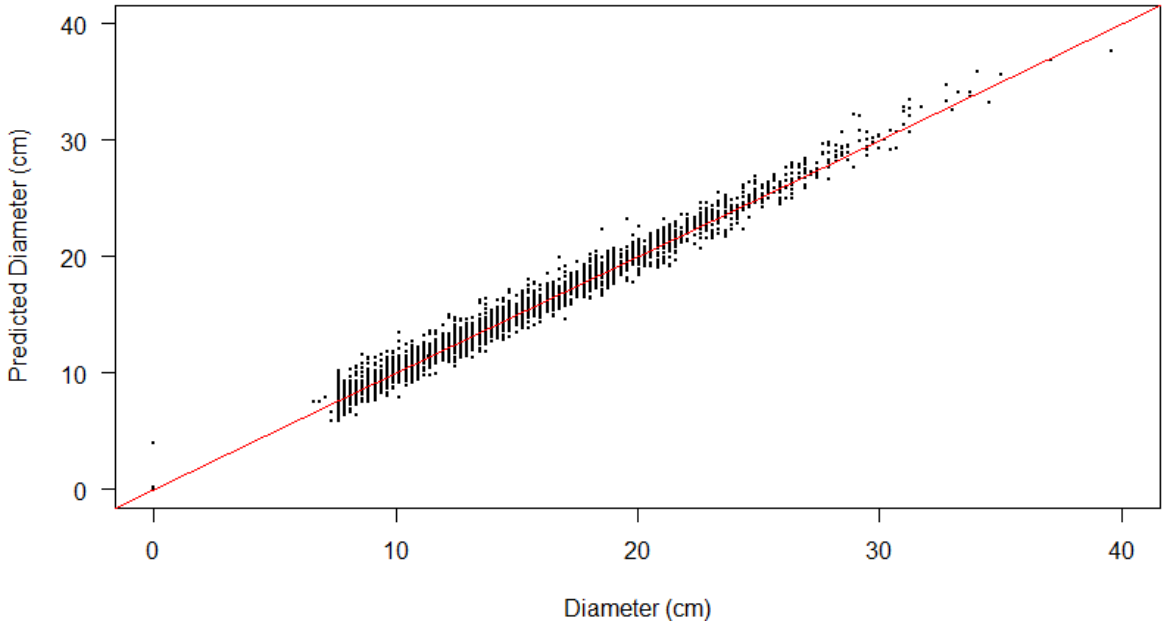

Figure 4.2. Predicted diameter from MB76 against known diameters for all d_h, h pairs in the destructive sample dataset of 324 trees.

Taper Model Height Estimation

Application of the method described above was successful in predicting heights, although with higher $RMSE_H$ than the benchmark value of 2.1m. The unweighted averages had the lowest $RMSE_H$. However, in comparing figures 4.5 and 4.6, it is clear that the h_i weight has a closer to 1-1 relationship than the unweighted average approach and is therefore said to be less biased. This

can also be seen in the β values. Additionally, the SME_H of the h_i weight is closer to zero. These results are shown in table 4.1. Lastly, it is important to note that a reasonable $\widehat{H}$ solution was found for all but one tree and that therefore 184 heights were used for the error assessment.

Table 4.1. Weight formulas applied on a tree-wise basis and resulting $RMSE_H$ across all trees.

Weights Formula	$RMSE_H$ (m)	SME_H (m)	β
N/A	3.11	0.46	0.957
$w_i = h_i$	3.84	0.07	0.982
$w_i = \frac{1}{h_i}$	4.64	1.56	0.886
$w_i = \frac{1}{convergence}$	3.93	-1.47	1.084

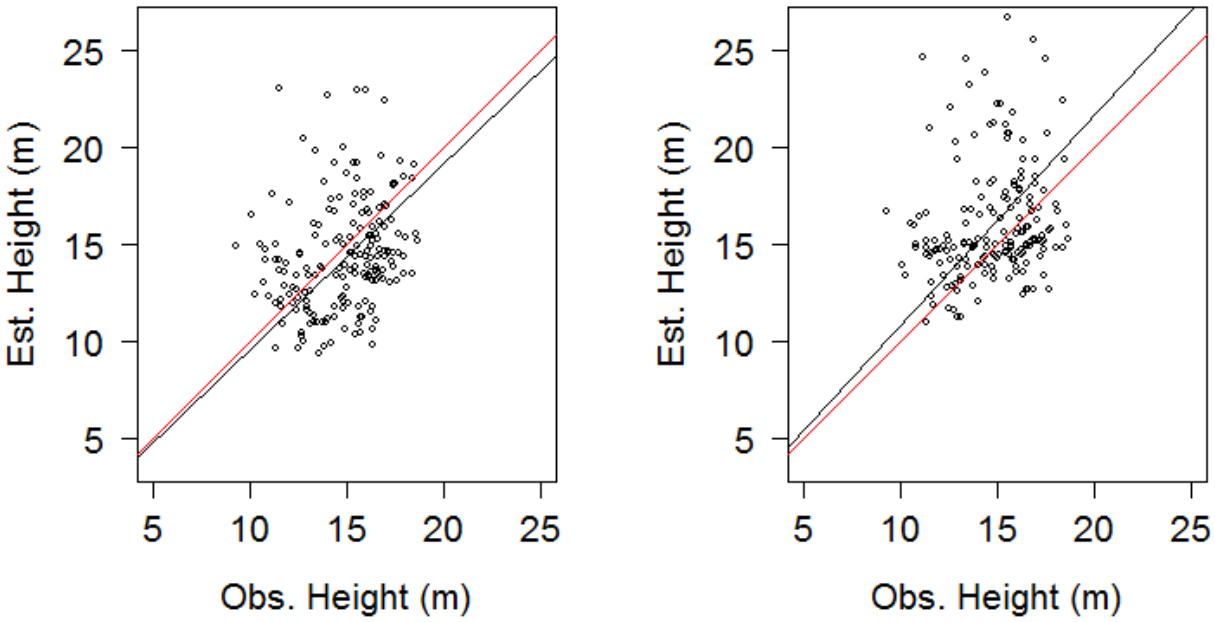

Figure 4.5. Unweighted averages of all $d_{l|h_i}, h_i$ pairs for each tree (left) and $\frac{1}{convergence}$ weight of each $d_{l|h_i}, h_i$ pair for each tree (right). Red lines are theoretical $\beta = 1$ while the black lines are fitted β values.

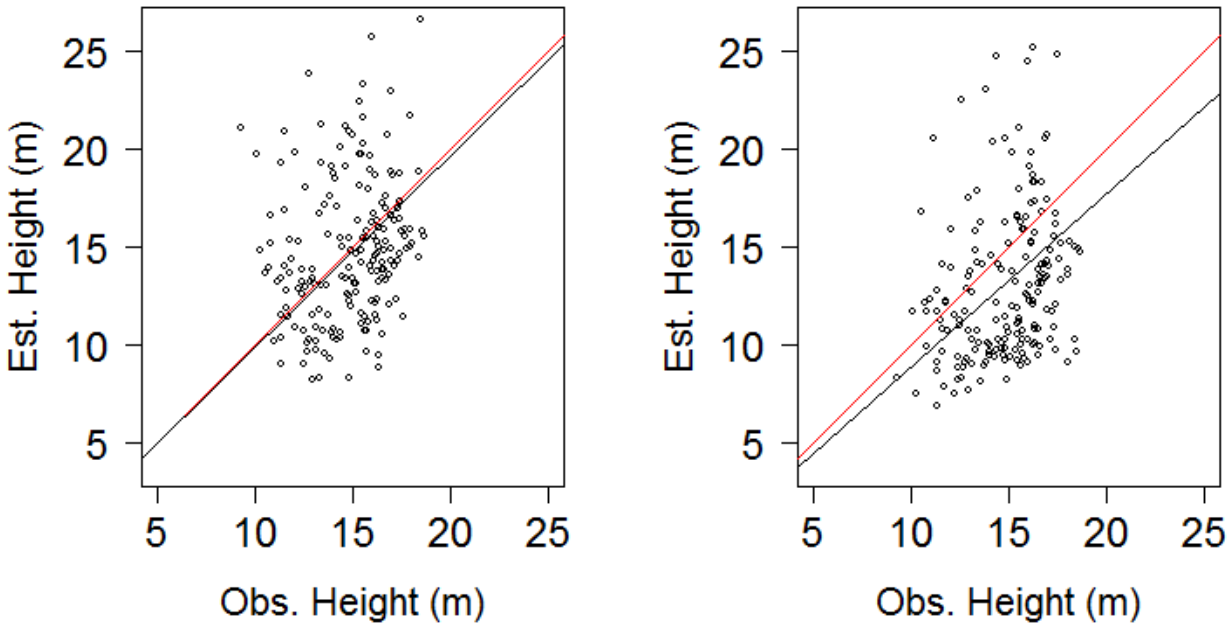

Figure 4.6. h_i weighted average (left), $\frac{1}{h_i}$ weighted averages (right). Red lines are theoretical $\beta = 1$ while the black lines are fitted β values.

Discussion

Using an inverse taper model to solve for total tree height yielded usable results, although with less accuracy than the S39 benchmark. The relatively poor performance of the $\frac{1}{h_i}$ weight as compared to the other methods suggests that diameters at greater heights might be better predictors of total height than lower stem heights when using diameters from MTLS. The principle assumption of this approach to height prediction was that θ in the taper model $f(d_h,\ h,\ D,\ \theta)$ was found for a sample representative of the population to be sampled, with an appropriate f also selected. Depending on species and region, MB76 might not be the best f. The simple analysis in this chapter did not address these questions- the f was chosen to be MB76 because it is a commonly used model with good performance that is well understood, and θ was found for it using all of the destructive sampled measurements as they are good data and are representative of *P. pallustris* in the region of interest. In a sense f, θ captures the global trend in taper (assuming it is

found with region wide data), with the MTLS estimated $d_{l|h_i}, h_i$ being the local taper trend. This leads to the question of how taper models and MTLS measurement error interact in estimating height- in this case the $d_{l|h_i}, h_i$ pairs were assumed to be error free, although we know this is not the case as was shown in chapters 2 and 3. Lastly, in practical application, this method has the advantage of removing height measurements from the field sampling procedure. Since data collection is very fast, collection of height and diameter measurements over large areas in a stand are anticipated to be easy, which might compensate for the higher height estimation error.

Conclusion

In this chapter a method of using multiple MTLS estimated diameters to estimate height was shown with usable precision by numerically solving a common taper model for total height. Assuming that these diameters are found with acceptable precision and in way that is practical to apply in an inventory, estimation of height can be accomplished for almost all trees in the sample. In conclusion, under certain conditions the fully automatic measurement of tree diameter and height from MTLS data is feasible with errors that are possibly useful in forest inventory.

REFERENCES

Arias-Rodil, M., Aranda-Dieguez, U., Burkhart, H.E. Effects of Measurement Error in Total Tree Height and Upper-Stem Diameter on Stem Volume Prediction. Forest Science. 63, 250-260. 2017.

Avery, T.E., Burkhart, H.E. Forest Measurements. McGraw-Hill. 7, 145-167. 2002.

Clutter, J.L., Fortson, J.C., Pienaar, L.V., Brister, G.H., Bailey, R.L. Timber Management: A Quantitative Approach. John Wiley & Sons. 1983.

Burkhart, H.E., Tome, M. Modeling Forest Trees and Stands. Springer. 5, 85-106. 2012.

Burkhart, H.E., Max, T.A. Segmented Polynomial Regression Applied to Taper Equations. Forest Science. 22, 283-289. 1976.

Byrd, R.H., Lu, P., Nocedal, J., Zhu, C. A Limited Memory Algorithm For Bound Constrained Optimization. Northwestern University Department of Electrical Engineering and Computer Science. NAM-08. 1994.

Cabo, C., Ordonez, C., Lopez-Sanchez, C.A., Armesto, J. Automatic dendrometry: Tree detection, tree height and diameter estimation using terrestrial laser scanning. International Journal Applied Earth Observation Geoinformation. 69, 164-174. 2018.

Campbell, J.B., Wynne, R.H. Introduction to Remote Sensing Fifth Edition. The Guilford Press. 2011.

Cao, Q.V. Calibrating a Segmented Taper Equation with Two Diameter Measurements. Southern Journal of Applied Forestry. 33, 58-61. 2009.

Conto, T., Olofsson, K., Gorgens, E.B., Rodriguez, L.C.E., Almeida, G. Performance of stem denoising and stem modelling algorithms on single tree point clouds from terrestrial laser scanning. Computers and Electronics in Agriculture. 143, 165-176. 2017.

Deng, H., Wickham, H. Density estimation in R. 2011.

Fang, R., Strimbu, B.M. Stem Measurements and Taper Modeling Using Photogrammetric Point Clouds. Remote Sensing. 9, 716. 2017

Frescino, T.S., Edwards, T.C.Jr., Moisen, G.G. Modeling spatially explicit forest structural attributes using Generalized Additive Models. Journal of Vegetation Science. 12, 15-26. 2001.

Hart, P.E. How the Hough Transform Was Invented. IEEE Signal Processing Magazine. 18-22. November, 2009.

Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning. Springer. 2009.

Henning, J. G., Radtke, P. J. Detailed Stem measurements of Standing Trees from Ground-Based Scanning Lidar. Forest Science. 52, 67-80. 2006.

Hough, P.V.C. Machine Analysis of Bubble Chamber Pictures. Proceedings to International Conference High Energy Accelerators and Instrumentation. 1959.

Hyyppa, E., Hyyppa, J., Hakala, T., Kukko, A., Wulder, M.A., White, J.C., Pyorala, J., Yu, X., Wang, Y., Virtanen, J-P., Pohjavirta, O., Liang, X., Holopainen, M., Kaartinen. Under-Canopy UAV laser scanning for accurate forest field measurements. ISPRS Journal of Photogrammetry and Remote Sensing. 164, 41-60. 2020.

Illingworth, J., Kittler, J. The Adaptive Hough Transform. IEEE Transactions on Pattern Analysis and Machine Intelligence. 9, 690-698. 1987.

Kasa, I. A Circle Fitting Procedure and Its Error Analysis. IEEE Transactions on Instrumentation and Measurement. 25, 8-14. 1976.

Kozak, A., Munro, D. D., Smith, J. H. G. Taper Functions and their Application in Forest Inventory. The Forestry Chronicle. 278-283. 1969.

Kozak, A. A variable-exponent taper equation**.** Canadian Journal of Forest Research**.** 18**,** 1363-1368**.** 1988.

Kozak, A. Effects of upper stem measurements on the predictive ability of a variable-exponent taper equation. Canadian Journal of Forest Research. 28, 1078-1083. 1998.

Kilkki, P., Saramaki, M., Varmola, M. A SIMULTANEOUS EQUATION MODEL TO DETERMINE TAPER CURVE. Silva Fennica. 12, 120-125. 1978.

LaBau, V.J., Bones, J.T., Kingsley, N.P., Lund, H.G., Smith, W.B. A History of the Forest Survey in the United States: 1830-2004. United States Department of Agriculture Forest Service. FS-877. 2007.

Liang, X., Litkey, P., Hyyppa, J., Kaartinen, H., Vastaranta, M., Holopainen, M. Automatic Stem Mapping Using Single-Scan Terrestrial Laser Scanning. IEEE Transactions on Geoscience and Remote Sensing. 50, 661-670. 2012.

Liu, G., Wang, J., Pinliang, D., Chen, Y., Liu, Z. Estimating Individual Tree height and Diameters at Breast height (DBH) from Terrestrial Laser Scanning (TLS) Data at Plot Level. Forests. 9, 398. 2018.

Magnussen, S., Eggermont, P., LaRiccia, V. N. Recovering Tree Heights from Airborne Laser Scanner Data. Forest Science. 45, 407-422. 1991.

Moskal, L.M., Zheng, G. Retrieving Forest Inventory Variables with Terrestrial Laser Scanning (TLS) in Urban Heterogeneous Forest. Remote Sensing. 4, 1-20. 2012.

Mesavage, C., Girard, J.W. TABLES FOR ESTIMATING BOARD-FOOT VOLUME OF TIMBER. United States Department of Agriculture Forest Service. 1946

Norzahari, F., Turner, R., Lim, S., Trinder, J. Estimating Taper Diameter and Stem Form of *Pinus Radiata* in Australia by Terrestrial Laser Scanning. IEEE. International Geoscience and Remote Sensing. 6491-6494. 2012.

Parker, R. C., Evans, D. L. An Application of LiDAR in a Double-Sample Forest Inventory. Western Journal of Applied Forestry. 19, 95-101. 2004.

Parker, R. C., Evans, D. L. LiDAR Forest Inventory with Single-Tree, Double- and Single Phase Procedures. International Journal of Forestry Research. 1-6. 2009.

Pitkanen, T.P., Raumonen, P., Kangas, A. Measuring stem diameters with TLS in boreal forests by complementary fitting procedure. ISPRS Journal of Photogrammetry and Remote Sensing. 147, 294-306. 2019.

Popescu, S. C., Wynne, R. H., Nelson, R. F. Measuring individual tree crown diameter with lidar and assessing its influence on estimating forest volume and biomass. Canadian Journal of Remote Sensing. 29, 564-577. 2003.

Saarinen, N., Kankare, V., Vastaranta, M., Luoma, V., Pyorala, J., Tanhuanpaa, T., Liang, X., Kaartinen, H., Kukko, A., Jaakkola, A., Yu, X., Holopainen, M., Hyyppa, J. Feasibility of Terrestrial laser scanning for collecting stem volume information from single trees. ISPRS Journal of Photogrammetry and Remote Sensing. 123, 140-158. 2017.

Sabatia, C.O., Burkhart, H.E. On the use of Upper Stem Diameters to Localize a Segmented Taper Equation to New Trees. Forest Science. 61, 411-423. 2015.

Schumacher, F.X. A new growth curve and its application to timber-yield studies. Journal of Forestry. 37, 819-820. 1939.

Shiver, B.D., Borders, B.E., Sampling Techniques for Forest Resource Inventory. John Wiley & Sons. 1996.

Srinivasan, S., Popescu, S.C., Eriksson, M., Sheridan, R.D., Ku, N-W. Terrestrial Laser Scanning as an Effective Tool to Retrieve Tree Level Height, Crown Width, and Stem Diameter. Remote Sensing. 7, 1877-1896. 2015.

Strimbu, V. F., Strimbu B. M., A graph-based segmentation algorithm for tree crown extraction using airborne LiDAR data. ISPRS Journal of Photogrammetry and Remote Sensing. 104, 30-43. 2015.

Thies, M., Pfeifer, N., Winterhalder, D., Gorte, B.G.H. Three-dimensional reconstruction of stems for assessment of taper, sweep, and lean based on laser scanning of standing trees. Scandinavian Journal of Forest Research. 19, 571-581. 2004.

Tilley, B. K., Munn, I.A., Evans, D. L., Parker, R. C., Roberts, S.D. Cost Considerations of Using LiDAR for Timber Inventory. Southern Forest Economics Workers. 43-50. 2004.

Trincado, G., Burkhart, H.E. A Generalized Approach for Modeling and Localizing Stem Profile Curves. Forest Science. 52, 670-682. 2006.

Wand, M.P. Fast Computation of Multivariate Kernel Estimators. Journal of Computational and Graphical Statistics. 3, 433-445. 1994.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. Remote Sensing. 8, 501. 2016.

APPENDIX 1. STATISTICS FOR DESTRUCTIVELY SAMPLED DATA

Table A.1.1. Summary of DBH (cm) of felled trees by stand

Order Sampled	Stand Number	Min DBH	1st Qu. DBH	Med. DBH	Mean DBH	3rd Qu. DBH	Max DBH
1	13	11.2	14.4	16.3	16.0	17.8	20.8
2	2	11.4	15.2	18.4	18.9	22.2	31.0
3	1	10.4	14.1	16.0	16.3	18.0	23.4
4	6	10.2	13.7	15.5	15.4	16.6	20.1
5	7	10.9	15.4	19.9	19.7	23.4	27.9
6	4	10.7	13.6	15.2	14.9	16.1	18.0
7	5	13.0	16.6	18.2	18.3	19.7	23.9
8	10	9.9	15.6	17.3	17.3	18.6	23.9
9	9	10.9	16.3	17.7	17.7	19.4	23.4
10	8	14.2	17.7	19.9	20.3	22.5	27.2
11	12	11.7	14.5	17.1	17.1	19.3	24.1
12	11	11.2	14.0	16.4	15.9	17.5	20.8
13	14	10.7	14.4	16.4	16.5	17.8	23.4
14	15	12.4	15.7	17.1	17.6	19.7	26.7
15	18	12.7	17.4	20.7	20.2	21.7	29.2
16	20	13.0	17.8	20.3	20.4	22.7	27.4
17	21	11.9	16.5	20.2	20.3	23.2	30.5
18	22	11.7	13.2	15.6	16.0	17.7	22.1
19	23	15.7	20.6	22.5	22.1	24.0	26.9
20	24	12.7	17.9	19.2	19.5	21.9	29.7

Table A.1.2. Stand summary statistics from reconnaissance cruise for *DBH* and *H* by stand. Diameters are in cm, with BA/HA in m^2/HA. Heights are reported in meters.

Stand #	TPH	$\overline{DBH}$	BA/HA	Min. DBH	Max. DBH	Mean H	Min. H	Max. H
1	716.6	15.5	14.3	7.1	22.9	11.6	6.7	15.2
2	822.9	18.3	22.1	7.4	25.1	15.5	7.9	20.1
4	773.4	13.5	11.3	8.4	19.6	10.7	7.9	13.4
5	798.1	17.5	19.9	6.4	24.9	11.6	4.0	14.6
6	469.5	15.0	8.7	6.4	21.1	9.1	4.9	11.0
7	758.6	19.1	23.7	5.6	33.0	14.0	4.9	17.7
8	575.8	21.3	21.3	8.9	30.5	14.0	6.7	17.4
9	635.1	22.9	26.4	16.0	30.5	15.5	11.6	18.9
10	1037.8	18.3	27.5	10.9	24.1	13.1	9.8	16.5
11	882.2	15.0	15.9	7.6	21.1	11.6	7.0	14.6
12	790.7	17.8	20.3	9.4	24.1	14.0	7.9	16.2
13	827.8	15.2	15.7	8.4	21.3	11.3	6.4	16.2
14	1119.4	14.5	19.6	4.3	22.9	13.7	4.9	18.3
15	914.3	17.8	23.9	9.4	24.9	17.7	9.1	22.3
16	716.6	18.0	18.8	9.1	25.4	15.8	10.4	20.1
17	1013.1	13.0	14.7	5.8	19.8	12.2	7.0	15.2
19	872.3	16.3	19.4	6.6	27.9	14.0	7.0	18.3
20	882.2	19.3	27.2	8.1	27.9	14.9	5.8	16.8
21	906.9	19.3	27.8	9.1	30.5	14.6	8.8	17.4
22	1045.3	15.2	19.6	7.6	24.4	10.4	6.1	16.5
23	370.7	22.9	15.7	11.4	35.6	15.5	12.2	17.7
24	716.6	15.5	14.3	7.1	22.9	11.6	6.7	15.2

Table A.1.3. Proportion of defects categorized into forks, sweeps, rust, and ramicorns for reconnaissance cruise data by stand, with total proportions of defects.

Stand #	P. Fork	P. Sweep	P. Rust	P. Ramicorn	P. Total
1	0.0690	0.0230	0.1494	0.0575	0.2989
2	0.2300	0.0500	0.1400	0.0300	0.4500
4	0.1489	0.0426	0.0957	0.0957	0.3829
5	0.2371	0.0412	0.1237	0.0516	0.4536
6	0.0175	0.0000	0.0000	0.0526	0.0701
7	0.1739	0.1630	0.0978	0.1413	0.5760
8	0.2286	0.0290	0.0429	0.0860	0.3865
9	0.1948	0.0130	0.0779	0.0520	0.3377
10	0.2460	0.0238	0.0397	0.0476	0.3571
11	0.1495	0.0000	0.0280	0.01870	0.1962
12	0.1042	0.0625	0.0208	0.0833	0.2708
13	0.1418	0.0448	0.0746	0.0224	0.2836
14	0.0588	0.0809	0.0074	0.0000	0.1471
15	0.2162	0.1441	0.0811	0.0180	0.4594
16	0.0805	0.0575	0.0000	0.0115	0.1495
17	0.1463	0.122	0.0976	0.0244	0.3903
19	0.1038	0.1887	0.0283	0.0283	0.3491
20	0.2897	0.2804	0.1028	0.1308	0.8037
21	0.2818	0.4182	0.0910	0.0727	0.8637
22	0.1181	0.1102	0.0630	0.0551	0.3464
23	0.2444	0.2222	0.0444	0.0667	0.5777
24	0.1765	0.2941	0.0588	0.0000	0.5294

APPENDIX 2. GEOSLAM ZEB-HORIZON SPECIFICATIONS

Table A.2.1. List of technical specifications for the GeoSLAM Zeb-Horizon handheld laser scanner used for data collection. These specs are sources from the GeoSLAM website at the url: https://mzt1b2rcaay128n901d0fifo-wpengine.netdna-ssl.com/wp-content/uploads/2020/03/ZEB-Horizon-product-card.pdf Accessed: 07/13/2020/23:08CST.

Range	100m
FOV	360x270
Protection class	IP 54
Processing	Post
Data logger carrier	Backpack or shoulder strap
Scanner weight	3.7kg
Colorized point cloud	Yes
Intensity	Yes
References imagery	Yes
Scanner points/second	300,000
No. of sensors	16
Relative accuracy	1-3cm
Raw data file size	100-200 MB/minute

APPENDIX 3. DIAMETER ESTIMATION R-CODE

Block A.3.1. Supporting functions for diameter estimation, error analysis, and plotting.

```
#  supporting functions
#  necessary packages
library(lidR)
library(KernSmooth)
library(TreeLS)
##########################################
#  extract a tree from the conventional taper data
gitDH <- function(file, treeNum){
  out <- subset(file, tnum == treeNum)
  out <- as.matrix(out)
  out <- out[,5:6]
  #  metric conversions
  out[,1] <- out[,1] * 0.3048
  out[,2] <- out[,2] * 0.0254
  return(out)
}

#  density penalty #1 circle likelihood function
#  log likelihood * penalty

llh.p1 <- function(params, X, Y, P)
{
  a  = params["a"]
  b  = params["b"]
  r  = params["r"]
  sigma = params["sigma"]
  exp.val = sqrt((X-a)**2 + (Y-b)**2)
  llh = sum(dnorm(r, mean=exp.val, sd = sigma, log=TRUE) * P)
  if(is.na(llh))
    llh <- -99999999
  return(-llh)
}
#  wraps about likelihood function and connnects to data
#  basic circle fitting function, takes
#  data = 5 wide matrix X, Y, Z, P, 1mm Z-class
#  f = maximum likelihood function
#  maxr = maximum expected radius in data

cfit <- function(data, f, maxr){
```

```
  zc <- mean(data[,3])
  n <- as.numeric(nrow(data))
  Xh <- as.numeric(max(data[,1]))
  Yh <- as.numeric(max(data[,2]))
  Xl <- as.numeric(min(data[,1]))
  Yl <- as.numeric(min(data[,2]))
  r1 <- as.numeric(sqrt( (Xh - Xl)^2 + (Yh - Yl)^2 ))
  rs <- ifelse(r1 <= maxr, r1, 0.99*maxr)
  X <- as.vector(data[,1])
  Y <- as.vector(data[,2])
  P <- as.vector(data[,4])
  inits <- c(a = mean(X), b = mean(Y), r = rs , sigma = 0.1)
  lower <- as.vector(c(Xl, Yl, 0.0001, 0.0001))
  upper <- as.vector(c(Xh, Yh, maxr, 5))
  out <- optim(inits, fn = f, X = X, Y = Y, P = P,
               lower = c(lower[1], lower[2], lower[3], lower[4]),
               upper = c(upper[1], upper[2], upper[3], upper[4]),
               method = "L-BFGS-B")
  dists <- sqrt((out$par[1] - data[,1])^2 + (out$par[2] - data[,2])^2)
  mse <- 1/nrow(data) * sum(out$par[3] - dists)^2
  return(c(zc, n = n, out$par[1], out$par[2], out$par[3], mse))

}
#  recursive wrapper of cfit and llh functions
#  takes an nx5 matrix of:
#  X, Y, Z, P, Z class
#  specify llhfun for different circle fitting objective functions
#  returns appropriate matrix, length unique Z class and 5 parameters columns
#  Z-class, number of points, a,b,r circle parameters as columns
#  takes maximum expected radius (meters) as third argument
rcfit <- function(d, llhfun, maxr){
  zclass <- unique(d[,5])
  nr <- as.numeric(length(zclass))
  nc <- 6
  nrnc <- nr*nc
  results <- matrix(rep(NA, nrnc), nrow = nr, ncol = nc, byrow = TRUE)
  for(L in 1:length(zclass)){
    results[L,] <- tryCatch(cfit(subset(d, d[,5] == zclass[L]), f = llhfun, m
axr = maxr),
                            error = function(e){return(rep(NA,6))})
  }
  return(results)
}
#  classify fitted circles by the equivalent height class in the
#  physically measured data
#  takes nx6 matrix from rcfit and a nx2 matrix from gitDH
#  returns nx7 matrix, essentially adds column to fitMAT
#  with the height of nearest diameter measurement from
#  destructively sampled data
gitClass <- function(fitMAT, dhMAT){
```

```
  hclasses <- as.vector(dhMAT[,1])
  column <- rep(NA, nrow(fitMAT))
  for(M in 1:length(column)){
    column[M] <- hclasses[which.min(abs(hclasses - fitMAT[M,1]))]
  }
  outmat <- cbind(fitMAT, column)
  return(outmat)
}
#  takes data matrix and kernel object (from 2D bkde)
#  returns nx4 matrix, kernel from each matrix
gitWT_K <- function(datmat, kob){
  dmat <- cbind(datmat, rep(0, nrow(datmat)))
  for(i in 1:nrow(dmat)){
    Xi <- dmat[i,1]
    Yi <- dmat[i,2]
    j <- which.min(abs(Xi - kob$x1))
    k <- which.min(abs(Yi - kob$x2))
    dmat[i, 4] <- kob$fhat[j, k]
  }
  return(dmat)
}
#  function to rescale matrices
rescale <- function(x, x.min = NULL, x.max = NULL, new.min = 0, new.max = 1)
{
  if(is.null(x.min)) x.min = min(x)
  if(is.null(x.max)) x.max = max(x)
  new.min + (x - x.min) * ((new.max - new.min) / (x.max - x.min))
}
##############################################################################
##
#  function for summarizing results
summarize.results <- function(tnums, e, dl, dp, npts, h, method){
  output <- rep(NA, 4)
  #  internal d.f., construct from input vectors to standardize formats
  sdf <- as.data.frame(cbind(tnums, dl, dp, e, h, npts))
  #  diameter vs predicted diameter 1-1 plots, for DBH only
  ut  <- unique(sdf$tnums)
  Dp <- rep(NA, length(ut))
  Dl <- rep(NA, length(ut))
  for(k in 1:length(ut)){
    tID <- ut[k]
    tree <- subset(sdf, tnums == tID)
    htoffset <- abs(1.3 - tree$h)
    Dp[k]  <- tree[which.min(htoffset),3]
    Dl[k]  <- tree[which.min(htoffset),2]
  }
  #  RMSE of DBH
  output[1] <- sqrt(mean((Dp - Dl)^2, na.rm = TRUE))
  output[2] <- mean(Dp - Dl, na.rm = TRUE)
  #  plot function
```

```
par(mfrow = c(1,2))
plot(Dl ~ Dp,
     xlab = "Known DBH (cm)", ylab = "Estimated DBH (cm)",
     xlim = c(0,40), ylim = c(0,40))
abline(0,1, col = "red")
hist(as.numeric(Dp - Dl), xlab = "DBH Error (cm)",
     main = NULL, freq = FALSE, breaks = 50)
mtext(paste("Method:", method, sep = " "), side = 3)
#  diameter vs predicted diameter 1-1 plots
output[3] <- sqrt(mean((sdf$e^2), na.rm = TRUE))
output[4] <- mean(sdf$e, na.rm = TRUE)
#  plot function
par(mfrow = c(1,2))
plot(sdf$dl ~ sdf$dp,
     xlab = "Known D (cm)", ylab = "Estimated D (cm)",
     xlim = c(0,50), ylim = c(0,50))
abline(0,1, col = "red")
hist(sdf$e, xlab = "D Error (cm)",
     main = NULL, freq = FALSE, breaks = 100,
     ylim = c(0, 0.35), xlim = c(-20, 20))
mtext(paste("Method:", method, sep = " "), side = 3)
#  bias models of diameters
par(mfrow = c(1,2))
plot(e ~ dl, xlab = "Est-D (cm)", ylab = "D-error (cm)", data = sdf)
abline(0,0, col = "red")
plot(e ~ dp, xlab = "Known D (cm)", ylab = "D-error (cm)", data = sdf)
abline(0,0, col = "red")
mtext(paste("Method:", method, sep = " "), side = 3)
#  bias model of Height
par(mfrow = c(1,2))
plot(e ~ h, xlab = "Height (m)", ylab = "D-error (cm)", data = sdf)
abline(0,0, col = "red")
plot.new()
m <- lm(e ~ h, data = sdf)
ms <- summary(m)
b1 <- m$coefficients[2]
P  <- ms$coefficients[,4][2]
t  <- ms$coefficients[,3][2]
se <- ms$coefficients[,2][2]
rs <- ms$r.squared
mtext(paste("Method:", method, sep = " "), side = 3)
mtext(paste("b1 of e,h bias,cm/m:", b1, sep = " "), side = 3, line = -1)
mtext(paste("P of b1:", P, sep = " "), side = 3, line = -2)
mtext(paste("t of b1:", t, sep = " "), side = 3, line = -3)
mtext(paste("s.e. of b1:", se, sep = " "), side = 3, line = -4)
mtext(paste("R-Squared:", rs, sep = " "), side = 3, line = -5)
#   Returns:
#  RMSE of DBH, signed mean error (SME) of DBH,
#  RMSE of all diameters, SME of al diameters
```

```
  return(output)
}
```

Block A.3.2. Diameter estimation method #1.

```
taper <- read.csv("//taper.csv", header = TRUE)
path  <- "//clipped_nodefect"
filenames <- dir(path = path)
count <- seq(1, length(filenames), by = 1)
time <- proc.time()
Tbw = 0.1
GS  = 10
results <- as.matrix(cbind(as.numeric(), as.numeric(), as.numeric(),
                           as.numeric(), as.numeric(), as.numeric(),
                           as.numeric()))
for(i in count){
  print(paste("Beginning #", i, sep = " "))
  file = filenames[i]
  filepath <- paste(path, file, sep = "/")
  tnum <- as.numeric(strsplit(file, "\\D+")[[1]][-1])
  tree <- readLAS(filepath)
  phys <- gitDH(taper, tnum)
  tree <- lasground(tree, csf(cloth_resolution = 0.1, class_threshold = 0.1,
rigidness = 3))
  tree <- lasnormalize(tree, knnidw(k = 20, p = 2))
  tree <- stemPoints(tree)
  kdf <- as.matrix(subset(as.data.frame(tree@data), Classification == 1))
  K <- bkde2D(x = as.matrix(kdf[,1:2]), bandwidth = c(Tbw, Tbw), gridsize = c
(GS, GS))
  K$fhat <- rescale(x = K$fhat, x.min = min(as.vector(K$fhat)), x.max = max(a
s.vector(K$fhat)),
  new.min = 0, new.max = 1)
  MAT <- as.matrix(subset(as.data.frame(tree@data), Classification == 1 & Ste
m == TRUE)[,1:3])
  MAT <- gitWT_K(MAT, K)
  MAT <- MAT[order(MAT[,3]),]
  MAT <- cbind(MAT, rep(NA, nrow(MAT)))
  rMAT <- as.matrix(cbind(as.numeric(), as.numeric(), as.numeric(),
                          as.numeric(), as.numeric()))
  for(z in 1:length(phys[,1])){
    ss <- subset(MAT, MAT[,3] > (phys[z,1] - 0.01) & MAT[,3] < (phys[z,1] + 0
.01))
    ss[,5] <- rep(phys[z,1], nrow(ss))
    rMAT <- rbind(rMAT, ss)
  }
  print("Initializing Circle Fits")
  cfMAT <- rcfit(rMAT, llh.p1, 0.5)
  cfMAT <- subset(cfMAT, cfMAT[,2] >= 5 & cfMAT[,5] >= 0.0001)
  classMAT <- gitClass(cfMAT, phys)
```

```
  uclass <- unique(classMAT[,7])
  luclass <- length(uclass)
  compMAT <- matrix(rep(0, luclass*6), nrow = luclass, ncol = 6)
  for(y in 1:luclass){
    ss <- subset(classMAT, classMAT[,7] == uclass[y])
    #  height, npts, rk, rbar, err, pseudo error
    compMAT[y,] <- as.vector(c(uclass[y], ss[2], (phys[y,2]*100),
                               (ss[5]*200), ((phys[y,2]*100) - (ss[5]*200)), (s
s[6]*200)))
  }
  outmat <- cbind(rep(tnum, luclass), compMAT)
  results <- rbind(results, outmat)
}
(proc.time() - time)/60
sum(results[,2])
head <- list(V1 = "tnum", V2 = "h", V3 = "npts", V4 = "r", V5 = "rbar", V6 =
"err", V7 = "pse")
res <- as.data.frame(results)
names(res) <- head
#####################################################################
res <- subset(res, is.na(err) == FALSE & rbar > 0.001)
write.csv(res, file = "//m1.csv")
with(res, summarize.results(tnum, err, rbar, r, npts, h, "M1"))
```

Block A.3.3. Diameter estimation method #2.

```
#  thesis method #2
#  destructive sample dataset
taper <- read.csv("//taper.csv", header = TRUE)
#  path to MTLS data
path  <- "//clipped_nodefect"
filenames <- dir(path = path)
count <- seq(1, length(filenames), by = 1)
time <- proc.time()
results <- as.matrix(cbind(as.numeric(), as.numeric(), as.numeric(),
                           as.numeric(), as.numeric(), as.numeric()))
Tbw = 1
GS  = 10
for(i in count){
  print(paste("Beginning #", i, sep = " "))
  file = filenames[i]
  filepath <- paste(path, file, sep = "/")
  tnum <- as.numeric(strsplit(file, "\\D+")[[1]][-1])
  tree <- readLAS(filepath)
  phys <- gitDH(taper, tnum)
  tree <- lasground(tree, csf(cloth_resolution = 0.1, class_threshold = 0.1,
rigidness = 3))
  tree <- lasnormalize(tree, knnidw(k = 20, p = 2))
  tree <- stemPoints(tree)
```

```
  kdf  <- as.matrix(subset(as.data.frame(tree@data), Classification == 1))
  K <- bkde2D(x = as.matrix(kdf[,1:2]), bandwidth = c(Tbw, Tbw), gridsize = c
(GS, GS))
  MAT <- as.matrix(subset(tree@data, Stem == TRUE & Classification == 1)[,1:3
])
  MAT <- gitWT_K(MAT, K)
  MAT <- cbind(MAT, round(MAT[,3] * 100)/100)
  MAT <- MAT[order(MAT[,5]),]
  print("Initializing Circle Fits")
  cfMAT <- rcfit(MAT, llh.p1, 0.5)
  cfMAT <- subset(cfMAT, cfMAT[,2] >= 5 & cfMAT[,5] >= 0.0001)
  classMAT <- gitClass(cfMAT, phys)
  uclass <- unique(classMAT[,7])
  luclass <- length(uclass)
  classMAT <- data.frame(classMAT)
  compMAT <- matrix(rep(0, luclass*5), nrow = luclass, ncol = 5)
  gm     <- lm(I(V5*200) ~ V1, weights = (1/(V6*200)), data = classMAT)
  #modMat <- matrix(rep(0, luclass*2), nrow = luclass, ncol = 2)
  #lpreds <- matrix(rep(0, luclass*2), nrow = luclass, ncol = 2)
  #gpreds <- matrix(rep(0, luclass*2), nrow = luclass, ncol = 2)
  for(q in 1:length(uclass)){
    ss <- subset(classMAT, column == uclass[q])
    h_c     <- uclass[q]
    mod <- lm(I(V5*200) ~ V1, weights = 1/(V6*200), data = ss)
    localPred <- predict(mod, data.frame(V1 = h_c), se.fit = TRUE)
    localRbar <- as.numeric(localPred$fit)
    localSigma <- as.numeric(localPred$se.fit)
    globalPred <- predict(gm, newdata = data.frame(V1 = h_c), se.fit = TRUE)
    globalRbar <- as.numeric(globalPred$fit)
    globalSigma <- as.numeric(globalPred$se.fit)
    rbar <- weighted.mean(x = c(localRbar, globalRbar),
                          w = c(1/localSigma, 1/globalSigma))
    rk <- (phys[q,2])*100
    err <- rk - rbar
    n_sec <- sum(ss[,2])
    compMAT[q,] <- as.vector(c(n_sec, h_c, rk, rbar, err))
    #modMat[q, ] <- as.vector(c(coef(mod)[1], coef(mod)[2]))
    #lpreds[q, ] <- as.vector(c(localRbar, localSigma))
    #gpreds[q, ] <- as.vector(c(globalRbar, globalSigma))
  }
  outmat <- cbind(rep(tnum, luclass), compMAT)
  results <- rbind(results, outmat)
}
(proc.time() - time)/60
sum(results[,2])
head <- list(V1 = "tnum", V2 = "npts", V3 = "h", V4 = "r", V5 = "rbar", V6 =
"err")
res <- as.data.frame(results)
names(res) <- head
#################################################################
```

```
res <- subset(res, is.na(err) == FALSE & rbar > 0.001 & rbar < 50)
write.csv(res, file = "m2.csv")
######################################################################
with(res, summarize.results(tnum, err, rbar, r, npts, h, "M2"))
```

APPENDIX 4. DIAMETER BIAS CORRECTION FUNCTION CROSS-VALIDATION R-CODE

Block A.4.1. Cross-validation code for residuals from diameter estimation method #2.

```
#  evaluating bias correction functions with cross-validation
m5 <- read.csv("//m2.csv") #  m5 = m2 in thesis
m13 <- read.csv("//m1.csv") # m13 = m1 in thesis
#  model summary function
gitSum <- function(m){
  ms <- summary(m)
  b1 <- ms$coefficients[2]
  P  <- ms$coefficients[,4][2]
  t  <- ms$coefficients[,3][2]
  se <- ms$coefficients[,2][2]
  rs <- ms$r.squared
  return(as.vector(c(b1, P, t, se, rs)))
}
ut <- unique(m5$tnum)
k  <- 1:10
K  <- rep(NA, length(ut))
for(i in 1:length(ut)){
  K[i] <- sample(k)
}
km <- cbind(ut, K)
out <- matrix(rep(NA, 34*10), ncol = 34, nrow = 10)
m1mat <- matrix(rep(NA, 2*10), ncol = 2, nrow = 10)
m2mat <- matrix(rep(NA, 2*10), ncol = 2, nrow = 10)
m3mat <- matrix(rep(NA, 3*10), ncol = 3, nrow = 10)
for(j in 1:length(k)){
  tss <- subset(km, K != k[j])
  tss <- tss[,1]
  #  train
  tr <- subset(m5, tnum %in% tss == TRUE)
  # test
  te <- subset(m5, tnum %in% tss == FALSE)
  lm1 <- lm(err ~ h, data = tr)
  lm2 <- lm(err ~ rbar, data = tr)
  lm3 <- lm(err ~ h + rbar, data = tr)
  m1mat[j, ] <- as.vector(c(lm1$coefficients[1], lm1$coefficients[2]))
  m2mat[j, ] <- as.vector(c(lm2$coefficients[1], lm2$coefficients[2]))
  m3mat[j, ] <- as.vector(c(lm3$coefficients[1], lm3$coefficients[2],
```

```
                              lm3$coefficients[3]))
  #  predict on test from train
  p1 <- (predict(lm1, newdata = te) + te$rbar)
  p2 <- (predict(lm2, newdata = te) + te$rbar)
  p3 <- (predict(lm3, newdata = te) + te$rbar)
  #  new RMSE
  rmse1 <- rmse(te$r, p1)
  rmse2 <- rmse(te$r, p2)
  rmse3 <- rmse(te$r, p3)
  rmsevec <- c(rmse1, rmse2, rmse3)
  #  bias after bias correction
  LML <- list()
  LML$hb1 <- lm(I(r - p1) ~ h, data = te)
  LML$db1 <- lm(I(r - p1) ~ r, data = te)
  LML$hb2 <- lm(I(r - p2) ~ h, data = te)
  LML$db2 <- lm(I(r - p2) ~ r, data = te)
  LML$hb3 <- lm(I(r - p3) ~ h, data = te)
  LML$db3 <- lm(I(r - p3) ~ r, data = te)
  modvec <- vector(length = 0)
  for(l in 1:6){
    modvec <- c(modvec, gitSum(LML[[l]]))
  }
  #  k, post-correction(b1, P, t, se, rs,), post-correction rmse
  modvec <- c(k[j], modvec, rmsevec)
  out[j, ] <- modvec
}
#  b1, P, t, se, rs
head <- list(V1 = "K",
             V2 = "M1_h_b1", V3 = "M1_h_P", V4 = "M1_h_t", V5 = "M1_h_se", V6
= "M1_h_rs",  #  r-p ~ h of post-correction for model e ~ h,
             V7 = "M1_r_b1", V8 = "M1_r_P", V9 = "M1_r_t", V10 = "M1_r_se", V
11 = "M1_r_rs",  #  r-p ~ r of post-correction for model e ~ h,
             V12 = "M2_h_b1", V13 = "M2_h_P", V14 = "M2_h_t", V15 = "M2_h_se"
, V16 = "M2_h_rs",  #  r-p ~ h of post-correction for model e ~ rb,
             V17 = "M2_r_b1", V18 = "M2_r_P", V19 = "M2_r_t", V20 = "M2_r_se"
, V21 = "M2_r_rs",  #  r-p ~ r of post-correction for model e ~ rb
             V22 = "M3_h_b1", V23 = "M3_h_P", V24 = "M3_h_t", V25 = "M3_h_se"
, V26 = "M3_h_rs",  #  r-p ~ h of post-correction for model e ~ h + rb,
             V27 = "M3_r_b1", V28 = "M3_r_P", V29 = "M3_r_t", V30 = "M3_r_se"
, V31 = "M3_r_rs",  #  r-p ~ r of post-correction for model e ~ h + rb
             V32 = "RMSE_M1" , V33 = "RMSE_M2", V34 = "RMSE_M3"  # post-corre
ction of test dataset for models: e ~ h, e ~ rb, e ~ h + rb
             )
out <- as.data.frame(out)
names(out) <- head
######################
#  mean/sd of post-correction bias betas (alphas in chapter 3)
mean(out$M1_h_b1)
sd(out$M1_h_b1)
mean(out$M1_r_b1)
```

```
sd(out$M1_r_b1)
mean(out$M2_h_b1)
sd(out$M2_h_b1)
mean(out$M2_r_b1)
sd(out$M2_r_b1)
mean(out$M3_h_b1)
sd(out$M3_h_b1)
mean(out$M3_r_b1)
sd(out$M3_r_b1)
# mean/sd of RMSE
mean(out$RMSE_M1)
sd(out$RMSE_M1)
mean(out$RMSE_M2)
sd(out$RMSE_M2)
mean(out$RMSE_M3)
sd(out$RMSE_M3)
#  mean parameters to use for final bias correction
m1_b1 <- mean(m1mat[,1])
m1_b2 <- mean(m1mat[,2])
m2_b1 <- mean(m2mat[,1])
m2_b2 <- mean(m2mat[,2])
m3_b1 <- mean(m3mat[,1])
m3_b2 <- mean(m3mat[,2])
m3_b3 <- mean(m3mat[,3])
#########################################################################
#  evaluate diameter finding methods with above code, summarize results as be
low
#  corrected errors
#  M2 in thesis = M5 in code
c1 <- m1_b1 + m1_b2*m5$h
cd1 <- c1 + m5$rbar
c2 <- m2_b1 + m2_b2*m5$rbar
cd2 <- c2 + m5$rbar
c3 <-  m3_b1 + (m3_b2*m5$h) + (m3_b3*m5$rbar)
cd3 <- c3 + m5$rbar
summarize.results(m5$tnum, e = as.numeric(m5$r - cd1),
                  cd1, m5$r, m5$npts, m5$h, "# 5M1")
summarize.results(m5$tnum, e = as.numeric(m5$r - cd2),
                  cd2, m5$r, m5$npts, m5$h, "# 5M2")
summarize.results(m5$tnum, e = as.numeric(m5$r - cd3),
                  cd3, m5$r, m5$npts, m5$h, "# 5M3")
#########################################################################
#  M1 in thesis = M13 in code
#  corrected error
c1 <- m1_b1 + m1_b2*m13$h
cd1 <- c1 + m13$rbar
c2 <- m2_b1 + m2_b2*m13$rbar
cd2 <- c2 + m13$rbar
c3 <-  m3_b1 + (m3_b2*m13$h) + (m3_b3*m13$rbar)
cd3 <- c3 + m13$rbar
```

```
summarize.results(m13$tnum, e = as.numeric(m13$r - cd1),
                  cd1, m13$r, m13$npts, m13$h, "# 13M1")
summarize.results(m13$tnum, e = as.numeric(m13$r - cd2),
                  cd2, m13$r, m13$npts, m13$h, "# 13M2")
summarize.results(m13$tnum, e = as.numeric(m13$r - cd3),
                  cd3, m13$r, m13$npts, m13$h, "# 13M3")
#  pick "best" model and write output to drive
c1_m13 <- cd1
c1_m5  <- cd1
m5m1 <- cbind(m5, c1_m5)
write.csv(m5m1, "m5m1.csv") #M2 in chapter 2
m13m1 <- cbind(m13, c1_m13) #M1 in chapter 2
write.csv(m13m1, "m13m1.csv")
```

APPENDIX 5. TAPER MODEL NUMERICAL HEIGHT ESTIMATION R-CODE

Block A.5.1. Code for fitting MB76 taper model with non-linear least squares

```
#  destructively sampled dataset
data <- read.csv(file = "//taper.csv", header = TRUE)
#  metric conversions
data$DBH <- data$DBH * 2.54
data$d   <- data$d * 2.54
data$h   <- data$h * 0.3048
data$THT <- data$THT * 0.3048
#  indicator function
Ind<-function(rH,jP)
{
  val<-ifelse(rH <= jP, 1, 0)
  return(val)
}
#  prediction form of Max & Burkhart 1976 Taper Model
MBpred <- function(D, H, h, p){
  rH <- h/H
  d <- D*(p[1]*(rH-1)+ p[2]*(rH^2-1)+ p[3]*(p[5]-rH)^2* Ind(rH, p[5])+
p[4]*(p[6]-rH)^2*Ind(rH, p[6]))
  return(d)
}
#  fitting MB76 with nonlinear least squares
MBfit <- nls(as.numeric(d/DBH) ~ (b1*((h/THT)-1)+ b2*((h/THT)^2-1)+ b3*(a1-
(h/THT))^2* Ind((h/THT), a1)+ b4*(a2-(h/THT))^2*Ind((h/THT), a2)),
             data = data, start = list(b1 = 0.7, b2= -1.5, b3 = 1.5, b4 =
100, a1 = 0.5, a2 = 0.06))
summary(MBfit)
parms <- coefficients(MBfit)
#  testing prediction form
pd <- MBpred(data$DBH, data$THT, data$h, parms)
#  define RMSE function
RMSE <- function(M, P){return(sqrt(mean((M - P)^2, na.rm = TRUE)))}
#  signed mean error function
SME <- function(M, P){return(mean(M - P, na.rm = TRUE))}
#  evaluate MB76 diameter prediction performance
with(data, RMSE(data$d, pd))
par(mfrow = c(1,1), las = 1)
plot(pd ~ data$d, xlab = "Diameter (cm)", ylab = "Predicted Diameter (cm)",
     cex = 0.5, pch = 20, xlim = c(0, 40), ylim = c(0,40))
abline(0,1, col = "red")
```

Block A.5.2. Evaluation & fitting of the S39 H ~ DBH model with non-linear least squares

```
#  evaluation of Schummacher 1939 model on same data
DH <- subset(data, h < 0.2)
D_m <- DH$DBH
H_m <- DH$THT
s39 <- nls(H_m ~ b0*exp(b1/D_m), start = list(b0 = 50, b1 = -2))
s39P <- coefficients(s39)[1]*exp(coefficients(s39)[2]/D_m)
RMSE(H_m, s39P)
par(oma = c(0.1,0.1,0.1,0.1), mar = c(4, 4, 1, 3), mfrow = c(1,2), las = 1)
plot(H_m ~ D_m, pch = 1, xlab = "DBH (cm)", ylab = "Obs. Height (m)",
     las = 1)
curve(coefficients(s39)[1] * exp(coefficients(s39)[2]/x),
      from = 0, to = 35, add = TRUE,
      col = "red", lty = 5, lwd = 2)
plot(s39P ~ H_m,
     xlim = c(10, 22), ylim = c(10, 22) ,
     xlab = "Obs. Height (m)", ylab = "Est. Height (m)")
abline(0,1, col = "red", lty = 5, lwd = 2)
abline(lm(s39P ~ H_m-1), lty = 1, lwd = 1)
```

Block A.5.3. Code for solving & evaluating solutions for H with a taper model

```
#  TLS derived diameters, method #2 in thesis with correction model #1
res <- read.csv(file = "//m5m1.csv", header = TRUE)
res <- na.omit(res)
res$DBH <- NA
#  Assign a lidar derived DBH
for(i in 1:nrow(res)){
  tID <- as.integer(res[i,3])
  tree <- subset(res, tnum == tID)
  htoffset <- abs(1.37 - tree$h)
  res[i,10]  <- tree[which.min(htoffset),9]
}
#  assign a height from the destructively sampled datset
res$THT <- NA
for(s in 1:nrow(res)){
  tn <- res[s,3]
  H <- as.numeric(subset(data, tnum == tn)[1,3])
  res[s,11] <- H
}

#  vector of tree numbers
ut <- unique(res$tnum)
ccv <- c(0, 50) # convergence check values, rejects non-convex minima
#  storage space for results
mHMB <- matrix(rep(NA, 6*length(ut)), ncol = 6, nrow = length(ut))
#  tree-wise outer loop
```

```
for(i in 1:length(ut)){
  #i = 105 #  un-comment for graph production of objective functions
  tree  <- subset(res, tnum == ut[i]) #  observations from tls data
  tnum  <- unique(tree$tnum) #  tree number
  Hp    <- mean(tree$THT)    #  physical measurement height for comparison
  max_h <- tree[nrow(tree),]$h #  height of highest segment diameter from TLS
  D     <- unique(tree$DBH)    #  TLS derived DBH
  #  define storage space for each d,h pair in tree
  pair_res <- matrix(rep(NA, 3*nrow(tree)), ncol = 3, nrow = nrow(tree))
  graph_update <- as.matrix(cbind(as.numeric(), as.numeric(), as.numeric()))
  #  middle loop for each measurement in tree
  for(j in 1:nrow(tree)){
    row <- tree[j,] #  d,h pair
    hud <- row$h    # h where d occurs
    dl <- row$c1_m5  #  d from TLS
    Hrange <- c(seq(from = max_h, to = 35, 0.2), ccv) #  possible ansers
    dt   <- rep(NA, length(Hrange)) #  storage space of predicted d
    #  inner loop for objective function
    for(k in 1:length(Hrange)){  #  predict d at h,Hrange,D
      dt[k] <- MBpred(D, Hrange[k], hud, parms)
    }
    #  solution for dl, hl combo
    diff <- dl - as.numeric(dt) #  compute differences
    #dminloc <- which.min(abs(diff))
    dminloc <- which.min((diff^2)) #  index of minimum value
    tol_MB <- diff[dminloc]  #  convergence of minimum value
    hcap_MB <- Hrange[dminloc]  #  output height and convergence check
    hcap_MB <- ifelse(hcap_MB == 50 | hcap_MB == 0,
                      hcap_MB <- NA, hcap_MB <- hcap_MB)
    #  replace convergence failures with NA
    pair_res[j,] <- as.vector(c(tol_MB, hcap_MB, hud)) #  store results
    #graph_dat    <- cbind((diff^2), Hrange, rep(hud, length(Hrange)))
    #graph_update <- rbind(graph_update, graph_dat)
  }
  #  if all NA, return NA for tree
  if(all(is.na(pair_res[,2])) == TRUE){
    output <- c(c(tnum, Hp), rep(NA, 4))
  }
  #  else averaging all estimated heights
  else{
    #  unweighted height
    m1 <- mean(pair_res[,2], na.rm = TRUE)
    #  height weighted
    m2 <- weighted.mean(x = pair_res[,2], w = pair_res[,3], na.rm = TRUE)
    #  inverse height weighted
    m3 <- weighted.mean(x = pair_res[,2], w = 1/pair_res[,3], na.rm = TRUE)
    #  inverse convergence tolerance weighted
    m4 <- weighted.mean(x = pair_res[,2], w = 1/abs(pair_res[,1]), na.rm = TR
UE)
    #  record tree number, true height (for comparison), and estimated height
```

```
s
    output <- as.vector(c(tnum, Hp, m1, m2, m3, m4))
  }
  #  output single tree solution
  mHMB[i,] <- output
}
#  packaging into data.frame
mHMB <- as.data.frame(mHMB)
head <- list(V1 = "tnum", V2 = "Hp",
             V3 = "UW", V4 = "HW", V5 = "IHW", V6 = "CW")
names(mHMB) <- head
mHMB[!complete.cases(mHMB),]  #  check for missing information and convergenc
e failures
##########################################################
```