To Aanu and Amma
ACKNOWLEDGMENTS

I have been fortunate to have had the privilege of meeting some of the most wonderful people during the course of my thesis work. First of all, I extend my most sincere thanks to Dr. Reza Ehsani, my mentor and guide, for giving me a platform to utilize my computer skills in the area of precision agriculture. He was very approachable and patient throughout my research work and directed me to do my work right. I wholeheartedly thank my supervisory committee, Dr. Masoud Salyani, Dr. Fritz Roka and Dr. Won Suk Lee, for letting me know what modifications could be done to improvise my work and guiding me to understand the subject matter better. I would love to thank Ms. Sherrie Buchanon, who helped me in analyzing all my results, from the beginning of my research until the very end. I would like to thank Dr. Joe Mari Maja, Dr. Lav Khot, Dr. Sindhuja Sankaran, Ms. Christy Ngu, and Ms. Jessie Lee for helping me with the data collection. I would like to thank Dr. Gene Albrigo and his team for the manual data collection in the fields. My sincere thanks to Ms. Jennifer Dawson, who helped arrange and schedule all the meetings. I extend my heartfelt gratitude to my sisters Dr. Anitha Shenoy, Dr. Savitha Shenoy, and my brother-in-law Dr. Vinayak Shenoy, who were the motivation for pursuing my masters. I would love to thank Ms. Farangis Anjom and Ms. Ujwala Jadhav for being wonderful friends and guides. My thanks would not be complete without thanking Mr. Prateek Kulshreshtha, who has been a source of immense support and inspiration throughout. Last, but not the least, I extend my precious thanks to my wonderful parents who were always there for me as a pillar of support and who have been guiding me in every phase of my life.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>4</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>8</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>10</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>12</td>
</tr>
<tr>
<td>Citrus Characteristics</td>
<td>12</td>
</tr>
<tr>
<td>Global Positioning System</td>
<td>12</td>
</tr>
<tr>
<td>Canopy Characteristics</td>
<td>13</td>
</tr>
<tr>
<td>Significance of the Study on Canopy Characteristics</td>
<td>13</td>
</tr>
<tr>
<td>Research Objectives</td>
<td>14</td>
</tr>
<tr>
<td>2 HISTORY AND LITERATURE REVIEW</td>
<td>16</td>
</tr>
<tr>
<td>Canopy Characteristics with Laser Sensor Data</td>
<td>17</td>
</tr>
<tr>
<td>Canopy Characteristics with Ultrasonic Sensors</td>
<td>21</td>
</tr>
<tr>
<td>Yield Estimation</td>
<td>23</td>
</tr>
<tr>
<td>Summary</td>
<td>25</td>
</tr>
<tr>
<td>3 MATERIALS AND METHODS</td>
<td>29</td>
</tr>
<tr>
<td>Materials</td>
<td>29</td>
</tr>
<tr>
<td>Citrus Groves and Sensor</td>
<td>29</td>
</tr>
<tr>
<td>Global Positioning System Receiver</td>
<td>29</td>
</tr>
<tr>
<td>Example of a GGA String</td>
<td>30</td>
</tr>
<tr>
<td>Example of a RMC String</td>
<td>31</td>
</tr>
<tr>
<td>Methods</td>
<td>32</td>
</tr>
<tr>
<td>Data Collection</td>
<td>32</td>
</tr>
<tr>
<td>Processing the Input File</td>
<td>33</td>
</tr>
<tr>
<td>Area, Height and Volume Calculation</td>
<td>35</td>
</tr>
<tr>
<td>Experimental Setup to Test the Efficacy of the Designed Algorithm</td>
<td>36</td>
</tr>
<tr>
<td>Density Calculations</td>
<td>36</td>
</tr>
<tr>
<td>Yield Estimation</td>
<td>38</td>
</tr>
<tr>
<td>4 RESULTS AND DISCUSSION</td>
<td>52</td>
</tr>
</tbody>
</table>
Experimental Data Results .......................................................................................................................... 52
Results on the Grove Data .......................................................................................................................... 52
Discussion .................................................................................................................................................. 53
Conclusion ................................................................................................................................................. 55
Future Work ............................................................................................................................................... 56

APPENDICES
A  MATLAB CODE FOR HEIGHT, VOLUME, AND DENSITY CALCULATION ........ 61
B  JAVA CODE FOR HEIGHT, VOLUME, AND DENSITY CALCULATION ............ 77
LIST OF REFERENCES ................................................................................................................................. 101
BIOGRAPHICAL SKETCH ......................................................................................................................... 103
<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Sensor specifications</td>
</tr>
<tr>
<td>3-2</td>
<td>Density data at high and low speeds</td>
</tr>
<tr>
<td>4-1</td>
<td>Density output file</td>
</tr>
<tr>
<td>4-2</td>
<td>Volume and height output file</td>
</tr>
<tr>
<td>4-3</td>
<td>PLS statistics</td>
</tr>
<tr>
<td>4-4</td>
<td>PCR statistics</td>
</tr>
<tr>
<td>4-5</td>
<td>SVM statistics</td>
</tr>
<tr>
<td>4-6</td>
<td>MLR statistics</td>
</tr>
<tr>
<td>4-7</td>
<td>CLS statistics</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1-1</td>
<td>Citrus grove variability</td>
</tr>
<tr>
<td>2-1</td>
<td>Laser scanning</td>
</tr>
<tr>
<td>2-2</td>
<td>Surface area and volume calculations by the manual method</td>
</tr>
<tr>
<td>2-3</td>
<td>Tractor mounted with the laser scanner and GPS receiver</td>
</tr>
<tr>
<td>2-4</td>
<td>Intersection of solids</td>
</tr>
<tr>
<td>3-1</td>
<td>Ft. Basinger citrus grove in Florida</td>
</tr>
<tr>
<td>3-2</td>
<td>LMS 200-30106 SICK laser scanner</td>
</tr>
<tr>
<td>3-3</td>
<td>Types of GPS</td>
</tr>
<tr>
<td>3-4</td>
<td>Veris EC equipment for measuring soil conductivity</td>
</tr>
<tr>
<td>3-5</td>
<td>Experimental setup</td>
</tr>
<tr>
<td>3-6</td>
<td>Laser data collection</td>
</tr>
<tr>
<td>3-7</td>
<td>Snapshot of the CVMS software</td>
</tr>
<tr>
<td>3-8</td>
<td>Snapshot of the file converter software</td>
</tr>
<tr>
<td>3-9</td>
<td>Example resultant .csv file</td>
</tr>
<tr>
<td>3-10</td>
<td>Schematic representation of the left and right tree rows with the variables</td>
</tr>
<tr>
<td>3-11</td>
<td>Experimental setup</td>
</tr>
<tr>
<td>3-12</td>
<td>White pole used to measure 100% hit region</td>
</tr>
<tr>
<td>3-13</td>
<td>Measuring the distance between the tree rows and center of the tractor lane</td>
</tr>
<tr>
<td>3-14</td>
<td>Removing the tree leaves manually</td>
</tr>
<tr>
<td>3-15</td>
<td>Weighed tree leaves after removal</td>
</tr>
<tr>
<td>3-16</td>
<td>Density reading zones</td>
</tr>
<tr>
<td>3-17</td>
<td>Section of the tree showing the measurements</td>
</tr>
<tr>
<td>3-18</td>
<td>Flowchart of the algorithm</td>
</tr>
</tbody>
</table>
3-19 Density Calculation Flow Chart

4-1 Density readings for a single tree

4-2 Relationship between the EC and yield

4-3 Relationship between the elevation and yield

4-4 Measured volume overlaid on the aerial image using Arc GIS software
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLS</td>
<td>Classical least squares</td>
</tr>
<tr>
<td>CREC</td>
<td>Citrus research and education center</td>
</tr>
<tr>
<td>CVMS</td>
<td>Canopy volume measurement system</td>
</tr>
<tr>
<td>DGPS</td>
<td>Differential global positioning system</td>
</tr>
<tr>
<td>GGA</td>
<td>Global positioning system fix data</td>
</tr>
<tr>
<td>GPS</td>
<td>Global positioning system</td>
</tr>
<tr>
<td>LAD</td>
<td>Leaf area density</td>
</tr>
<tr>
<td>LAI</td>
<td>Leaf area index</td>
</tr>
<tr>
<td>LASER</td>
<td>Light amplification by stimulated emission of radiation</td>
</tr>
<tr>
<td>LIDAR</td>
<td>Light detection and ranging</td>
</tr>
<tr>
<td>MLR</td>
<td>Multiple linear regression</td>
</tr>
<tr>
<td>NDVI</td>
<td>Normalized difference vegetation index</td>
</tr>
<tr>
<td>PCR</td>
<td>Principal component regression</td>
</tr>
<tr>
<td>PLS</td>
<td>Partial least squares</td>
</tr>
<tr>
<td>PVC</td>
<td>Polyvinyl chloride</td>
</tr>
<tr>
<td>RMC</td>
<td>Recommended minimum specific GPS transit data</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machine</td>
</tr>
<tr>
<td>TRLV</td>
<td>Tree row LIDAR volume</td>
</tr>
<tr>
<td>UTC</td>
<td>Coordinated universal time</td>
</tr>
</tbody>
</table>
Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

A METHODOLOGY TO DETERMINE AND MAP THE VARIABILITY OF CITRUS TREE
CANOPY CHARACTERISTICS

By

Akshatha Ramesh Shenoy

August 2013

Chair: Reza Ehsani
Major: Agricultural and Biological Engineering

The estimation of the geometric characteristics of a citrus tree canopy can
provide great insight into the tree growth variations and productivity predictions. A
canopy variability map can be used for a variable rate shaking system on a mechanical
harvesting machine or a variable rate spraying system. It can also be used for
estimating yield. An algorithm was designed to gauge the tree canopy characteristics
from the laser sensor data collected, which in turn was used to estimate the yield of the
citrus groves. The estimated volume and height gave a percent error of 8.2% and 2.5%,
respectively. The experimental analysis with density calculations produced a coefficient
of variation of 0.94. The estimated yield and predicted yield produced a percentage
error between 13.8-15.5% for all the supervised learning models. The algorithm had a
lower runtime in the Java environment when compared to the Matlab environment.
CHAPTER 1
INTRODUCTION

Citrus Characteristics

Citrus groves in Florida have always been fairly non-uniform. The groves are prone to various diseases resulting in a lot of variability across the state. This variability has affected a lot of factors concerning with the grove maintenance and yield productivity. Figure 1-1 shows a citrus grove in Florida with a lot of variability.

Some of the diseases that are prevalent in citrus trees are Alternaria, Anthracnose, Brown Rot, citrus blight, citrus canker, Mal Secco disease, powdery mildew, Satsuma Dwarf, citrus yellow mosaic, citrus greening, HLB. These diseases result in a lot of variability across the groves as the health of the trees depreciate along with its canopy size in some cases.

This study aims at developing an algorithm to calculate the major tree canopy characteristics of height, volume, and density using laser scanned data. The knowledge of tree canopy characteristics at a given location can be used to control the pesticide or chemical applications based on the size of the tree, which would be very cost effective.

Global Positioning System

This study makes use of the Global Positioning System (GPS) receivers in order to locate the point on the ground at which the canopy characteristics are being estimated. The output files generated from the algorithm, containing the volume, height, and density data, correspond to a geo-referenced location that helps the growers to relate the output data to their groves.

GPS is a collective term used to refer to a constellation of 31 satellites along with the ground stations which form a worldwide radio navigation system. It makes use of
these satellites to determine the location. GPS was first developed to aid the defense services. It gives every square meter on earth a specific address making it unique. GPS is being widely used on a commercial basis in cars, boats, farm machinery, laptop, computers, etc.

**Canopy Characteristics**

The term canopy is used to define the orientation of leaves in a tree which absorb sunlight to carry out the process of photosynthesis. The arrangement of leaves in the canopy varies enormously, which in turn decides the exposure of the leaves to the sun and the amount of the sunlight absorbed. This plays a vital role in the growth of the tree and the yield it produces.

Yield prediction, biomass estimation, growth rate, water consumption, site-specific management and long term productivity of a citrus tree can be directly associated with canopy characteristics.

The importance of estimating canopy characteristics is that:

- It can be used for estimating yield.
- A canopy variability map can be used for a variable rate shaking system on a mechanical harvesting machine or a variable rate spraying system.
- It can also provide information on tree inventory such as size of trees which is needed for crop insurance, both by growers and crop insurance companies.

**Significance of the Study on Canopy Characteristics**

There has been a need to calculate the canopy characteristics in the groves for a long time now. As the groves span over acres of land, it becomes practically impossible to manually have the canopy characteristics measured and documented. Thus, there is a requirement to design a system which can help the growers make an estimate of the tree growth at all the locations and to capture areas that have sparse tree growth. The
The methodology proposed aims at gauging the tree canopy characteristics from the laser sensor data collected. It helps to determine the non-uniformity of the canopy size in the groves.

**Research Objectives**

Specifically, the aim is to develop an algorithm that can efficiently estimate the major tree canopy characteristics of height, volume, and density of the trees based on the data collected from the citrus groves using a commercially available laser scanner. The algorithm was tested for accuracy by comparing it with the manually collected data.

This study also aims at developing a model to estimate the yield based on the calculated canopy factors from the algorithm and manually collected data on soil conductivity and elevation.

The output of the algorithm would create tree variability maps that can be used by citrus growers to estimate the yield and tree growth rate, and would locate the areas where the tree growth is sparse or nil.

Autonomous tractors are being developed which have computer-controlled system along with an inbuilt laser scanner. They would give the information on tree canopy, which can be used to determine the tree height, volume, and density in a cost-effective manner.
Figure 1-1. Citrus grove variability
CHAPTER 2
HISTORY AND LITERATURE REVIEW

Precision agriculture deals with managing farms such that the resources are efficiently utilized and preserved to have maximum returns with the given inputs. Information technology, satellite imagery, and satellite positioning systems are some of the aspects of precision agriculture. The estimation of tree canopy characteristics has always been a key area of research as the results would give an in-depth knowledge about the quantity of fertilizers and chemicals to be used and the variability present in the groves.

As mentioned in Chapter 1, the absorption of sunlight is a key factor contributing to plant growth and the increase in the size of the canopy. It was in the early twentieth century that the first measurements of CO\textsubscript{2} exchange were carried out (Boysen-Jensen, 1918; Henrici, 1921; Lundegardh, 1922; Baldocchi, 2001). In citrus groves, it was noted that for the photosynthetic activity to be at its peak, the optimum conditions required were somewhere between full shade and full sunlight. It was also noted that interception of approximately 30% of the annual radiation is a necessity for efficient yield production. Overall, sections of the canopy unexposed to direct sunlight tend to have low foliage density and comparatively decreased fruit production than sections of canopy exposed to direct sunlight. The above phenomenon is more apparent at the lower sections of the tree which receive negligible sunlight (Tucker et al., 1994).

Leaf Area Index (LAI) is used to determine the number of leaf surfaces, a crop has per acre of land (Tucker et al., 1994). Tucker et al. (1994) measured the LAI by the following approaches: measuring the tree area covering the ground, measuring the tree canopy surface area, measuring the total volume of the tree and measuring the outer
three feet of the tree canopy which represented the bearing volume. The bearing volume refers to the part of the canopy which produces or bears the maximum fruits. It was noted that the production potential of a citrus tree is directly proportional to the canopy volume and that the first three outer feet (from the outside to inside) of the canopy receives 90% of the solar radiations.

**Canopy Characteristics with Laser Sensor Data**

Wei and Salyani (2004) designed algorithms to measure the canopy characteristics using data collected from a laser scanner. Fixed-length polyvinyl chloride (PVC) pipes were used to test the correctness of the algorithm. For the height calculations, the pipes were placed within the tree canopy at fixed distances. Figure 2-1 shows the analysed image obtained after processing the data.

The distance of these pipes and their length were measured manually which represented the ground truth. The laser scanner scanned through the canopy, and the PVC pipes were distinctly identifiable from the resulting pseudo-images. For volume calculation, a rectangular parallelepiped box with certain dimensions was used to represent the ground truth for establishing the accuracy of laser data measurements.

The results showed that the accuracy of the length measurement obtained from the designed algorithms were above 97%. The volume measurements using the algorithms in comparison to the ground-truth resulted in coefficients of variation (CV= standard deviation/mean) of 5.4% and a relative error (Relative error= (measured data-actual data)/actual data) of 4.4%.

Polo et al. (2008) developed a structured algorithm to determine the tree volume and total crop surface area using the data collected by a light detection and ranging (LIDAR) system mounted on a tractor scanning through the rows of trees on both sides.
Manual measurements of a few selected representative trees were carried out to establish the ground truth. The leaf area was measured by two methods. In the first method, a relationship was deduced between the results obtained from LIDAR data and the manual measurements of the combined foliar area. This helped in the determination of the Leaf Area Index (LAI). The second method followed Beer’s law. Beer’s law states that the transmitted light attenuates exponentially when it passes through a plant.

\[ I(r) = I_0 e^{-\alpha r} \]

where \( I_0 \) is the initial beam intensity, and \( I(r) \) is the final beam intensity passing through the tree canopy, and \( \alpha \) is an extinction coefficient related to the leaf area density. The experiments were carried out in apple orchards, pear orchards, and vineyards. This study showed that the LIDAR based system could be used to estimate the tree volume and leaf-area in a non-destructive manner; and also the results obtained from the LIDAR data measurements and manual measurements had a good correlation with each other.

Lee and Ehsani (2009) developed laser scanner based measurement system to determine the tree canopy geometric characteristics of height and surface area. The study used a laser scanner, a GPS receiver, and an inertial sensor mounted on a tractor to scan the tree rows to both left and right. Figure 2-2 shows the manual surface area estimation approach.

Three different techniques were used to estimate the geometric characteristics of the tree: a) Convex-Hull method b) Savitzky-Golay filter method and c) the original laser data. The tractor had a constant speed of 0.63 m/s; and the angular resolution of the laser was maintained at 0.25°. It was observed that the results obtained from Convex-Hull method were close to the results obtained from the original laser dataset. The Savitzky-
Golay filter method was not in the same range as that of the original dataset results. Figure 2-3 shows the data collection using a laser scanner in a citrus grove.

Rosell et al. (2009) continued with experiments to measure the structural characteristics of trees. In this study, the three dimensional images of trees using a 2D LIDAR system were obtained. Computer Aided Design (CAD) software was used to visualize the cloud of three-dimensional points collected by the LIDAR scanner. The 3D digitalized images of the trees were used to determine the height, total volume, width, leaf surface area and LAI. The 3D model helped in producing images which were highly comparable to real crops, which made the system very efficient. The results obtained were comparable to the manually measured volume, height, and other characteristics.

Fernández-Sarría et al. (2013) used the Terrestrial Laser Scanner (TLS) to estimate the crown volumes of the urban *Platanus hispanica* trees. This study manually calculated the dendrometric parameters such as the tree width, height, crown diameter, and distance from the base of the tree crown to the soil. Conical, paraboloid, and hemisphere geometric models were used to estimate the volume of the tree crown. The manual measurements were taken for 30 specimens. The TLS scanned these trees which resulted in a point cloud of laser data for every tree, which were then used to calculate the volumes of the respective crowns.

There were four methodologies employed to obtain the crown volume (Fernández-Sarría et al., 2013). The first was to calculate the convex hull of the point cloud obtained and apply it to the entire crown. The second method was again a convex hull, but the difference being that the crown was divided into slices, 10 cm in height from bottom to top of the tree crown, and the convex hull for each slice was obtained.
The third was to determine the XY triangulation in horizontal sections. This method made use of the Delaunay triangulation. The crown data was divided every 10 cm in height from top to bottom similar to the second method. The surface area was calculated for each slice based on the Delaunay triangulation. The total volume was calculated by summing up the surface area of each slice and multiplying the result with height (10 cm).

The fourth was by the method of voxel discretization. This method used a grid in the 3D space to divide the point cloud in small units each having a volume of its own. The total volume was calculated by multiplying points inside each voxel with the number of voxels by the voxel volume. 

Total Volume= points in voxel * number of voxels/voxel volume;

The results showed that the convex hull calculated on the sections (second method) and the XY triangulation methods (third method) were more appropriate than the global convex-hull method (first method) as it reduces the over-estimation of crown volume. The voxel method generated good results. However, one of the drawbacks with this method was the hidden points in the central part of the canopy that cannot be ignored during the volume estimations.

Sanz et al. (2013) used the LIDAR 3D dynamic measurement system along with the 2D Terrestrial Laser LIDAR (TLS) for the geometric characterization of tree crops. The study included calculating the tree row LIDAR volume (TRLV) from the laser data and establishing a non-linear relationship with the leaf area density (LAD). The TRLV was calculated from the principal of intersection of two solids. The solids were obtained from the left and right scanned 3D data. The intersection of these solids gave the TRLV of the tree.
LAD = −0.36\ln(TRLV) + 3.69

The above equation gave the relationship between the TRLV and LAD. The coefficient of determination (R^2) was 0.87 and it showed that there was a good logarithmic fit obtained. Figure 2-4 shows the solids generated from the left and right scans.

**Canopy Characteristics with Ultrasonic Sensors**

Zaman and Salyani (2004) used ultrasonic sensors to find a relationship between the foliage density and the vehicle ground speed with the tree volume. For the study, 15 densely foliated trees and another 15 partially-defoliated citrus trees were considered. The ultrasonic system used 10 ultrasonic sensors. The divergence angle of each sensor was maintained at 6 degrees and they pulsed at a rate of 40 kHz. The sensors were at a distance of 0.46 m. The sensors were sequentially fired to avoid interference between neighboring sensors. The sensors were divided into groups: (1, 5, 9), (3, 7) and (2, 6, 10). The distance from the sensor to the foliage was captured by every individual sensor. The whole system was mounted on a tractor which moved between rows for collecting the data.

The canopy volume was calculated by the given equation (Tumbo et al., 2002):

\[
U = \sum_{1}^{n} \frac{S \cdot D_{d} \cdot D_{s}}{S_{r}}
\]

where,

U = Calculated ultrasonic volume
S = ground speed of the tractor
D_{d} = distance from the tree row line to the periphery of the canopy in meter
D_{s} = vertical spacing between adjacent sensors
The results obtained from the ultrasonic sensor system data were at a 95% confidence level compared to the manually calculated results. It was reported that the volume variability increased proportionally with the ground speed in partially defoliated trees. The results corresponding to the two halves of the tree were found to be asymmetric; hence the tree had to be scanned from both sides for accurate measurements.

Schumann and Zaman (2005) used ultrasonic sensors system to map the tree canopy size in real-time. The main aim of the study was to a) develop a software system to calculate the real-tree canopy size b) develop a graphical diagnostic interface, and c) evaluate the accuracy of the developed software in commercial groves.

The tree canopy volume and height were measured accurately by the software at a rate of 13.6 trees per minute which was far more efficient than the manual measurements, which would have taken approximately a minute for every tree.

Rosell and Sanz (2012) reviewed widely used sensor technologies and methodologies to estimate the geometric characterization of the tree crops. The most important technologies which have been greatly successful in estimation of the geometric characteristics of trees are the LIDAR laser scanners and the stereo vision systems. The study also discusses many of the positive aspects and drawbacks of the LIDAR laser scanners and the stereo vision systems. The most important ones being that the LIDAR laser scanner was found to lack in terms of the post processing of data, as the collected data had to be processed in a software to determine the canopy characteristics, whereas,
for the stereo vision systems, the drawback was the cost of the equipment used and its large scale deployment.

Llorens et al. (2011) combined the ultrasonic sensor application and the LIDAR application to carry out estimate the canopy characteristics. Canopy characterization is an important factor in estimating the fertilizer and pesticide application in crops. This study compared the benefits of using ultrasonic and LIDAR sensors over the conventional manual measurements. It was reported that the ultrasonic sensors were beneficial in calculating the average canopy characteristics, whereas, LIDAR sensors provided more details and were more applicable for canopy measurements.

Field tests were carried out for three years using the system which constituted of a laser scanner and three ultrasonic sensors. Ultrasonic sensors helped in estimating the canopy volume and LAI with high accuracy. The LIDAR sensors gave detailed results; and it was observed that the post processing of the data posed some difficulty as the software had to be developed to calculate the canopy characteristics.

**Yield Estimation**

Ye et al. (2008) used airborne hyperspectral imagery to estimate the yield in citrus crops. Pixel-based average spectral reflectance values were used to identify the features of the canopy at different wavelengths from the acquired images. These were then used to design a yield estimation model.

Five methodologies were used to estimate yield: a) a combination of several vegetation indices, b) a simple correlation analysis to determine key wavelengths, c) principal components from principal component regression, d) Partial Least Squares (PLS) factors, and e) important wavelengths established by a B-matrix obtained from PLS regression.
The yield data was collected during the 2007 harvest season. An airborne imaging spectrometer for applications (AISA) Eagle Systems, was used to acquire the hyperspectral images. The data was then processed by using calibration coefficients, which were determined in the laboratory. Before estimating the yield model, the data had to be filtered, as the data faced a multicollinearity problem (high correlation existing between wavelengths).

The results showed that wavelengths in the range of 407 to 898 nm had a high correlation with the citrus yield. The five methodologies used to estimate yield proved to be highly comparable to the yield data collected manually.

Zaman et al. (2006) devised a study to estimate the yield by relating it to the tree size which was measured using ultrasonic sensors. An automated ultrasonic system was used to estimate the tree canopy volume and fruit yield of ‘Valencia’ groves and a sensor-based automatic yield monitoring system was used to map the data. Collected spatial data was divided into 40 equal-sized plots using ArcView GIS software. The resulting data was used to establish a relationship between the tree canopy size and fruit yield.

The fruits were manually harvested and placed in tubs; these were passed on to the yield monitoring system which stored the position of every full tub. ArcView GIS software was used to map the ultrasonically-obtained tree size information with the fruit yield. For each of the 40 plots, the correlation was mapped using linear regression.

The results showed that the ultrasonically-sensed tree sizes were strongly correlated to the fruit-yield such that if the tree size was small, the yield was small and vice versa.
Summary

The most important studies carried out to estimate the canopy characteristics deal with ultrasonic sensors and laser scanners.

Both of these methodologies have their pros and cons. Laser scanner based studies require a lot of post processing of the data for the results to be shown, but are very efficient in the estimation of the canopy, whereas, the ultrasonic sensors do not require any post processing but they capture a lot of noise along with the actual data, making it not very efficient in the canopy approximation.

Most of the techniques used until now are not commercially used as they are either very expensive or are not practical. This study aims at calculating the canopy characteristics on a commercial level and building a cost effective grower tool.
Figure 2-1. Laser scanning A) Hypothetical distant image B) Scanning illustration for a single slice (Source: Development of a laser scanner for measuring tree canopy characteristics: Phase 1. Prototype development, Wei and Salyani, 2004)

Figure 2-2. Surface area and volume calculations by the manual method. (Source: A Laser scanner based measurement system for quantification of citrus tree geometric characteristics, Lee and Ehsani, 2009)
Figure 2-3. Tractor mounted with the laser scanner and GPS receiver (Source: A Laser scanner based measurement system for quantification of citrus tree geometric Characteristics, Lee and Ehsani, 2009)
Figure 2-4. Intersection of solids A) Solids generated with the right-scan data B) Solids generated from the left-scan data C) intersection of two solids generated from a and b (Source: Relationship between tree row LIDAR-volume and leaf area density for fruit orchards and vineyards obtained with a LIDAR 3D Dynamic Measurement System, Sanz et al., 2013)
CHAPTER 3
MATERIALS AND METHODS

This chapter provides an in-depth understanding of the experiments conducted and the materials used for this study. The height, area and volume calculation approach has been adopted from Lee and Ehsani, 2009. The new approaches exhibited in this study are the area calculation, density calculation and the yield estimation. The algorithms were written in both Java and Matlab environments.

Materials

Citrus Groves and Sensor

All the experiments were conducted on Valencia orange groves located in Fort Basinger, owned by Lykes Brothers, and CREC groves, Lake Alfred, Florida. Figure 3-1 shows a typical Florida citrus grove.

The SICK laser sensor provides the distance between the laser and the target in millimeter. The sensor moves vertically or horizontally, depending on the way it is positioned, and collects data at different angular resolutions. For the experiment, we maintained the scan angle to 1 degree; and the laser scanner was mounted such that it provided the data from 0 to 180 degrees perpendicular to the ground. The maximum distance the SICK laser scanner can scan is 8191 mm. The SICK scanner generated 12 scans per second and each scan consisted of 0 – 180 degrees. Figure 3-2 shows the SICK sensor along with its internal working mechanism.

Global Positioning System Receiver

A GPS (GPS18-5 Hz, Garmin International Inc.) was mounted on top of the tractor to collect the location of the scanned rows and also to determine the speed of the tractor. Figure 3-3 shows the types of GPS receivers used in this study.
The latitude-longitude co-ordinate system was used in the experiments. The latitude at a point is defined as an angle that is formed at the juncture of the equatorial plane and an imaginary line that runs through that point and is normal to the surface of the reference ellipsoid which is a depiction of the shape of the earth. The longitude at a point is defined as the angle formed between a reference meridian to the east or to the west to another meridian that runs through that point.

The Global Positioning System (GPS) receiver can be set to collect data in different formats. The experiments conducted collected GPS readings in global positioning system fix data (GGA) or recommended minimum specific GPS transit data (RMC) formats. These provide a string of data with many attributes which are explained below. The GGA string gives the elevation at every location which is not provided in an RMC string; whereas, RMC provides the speed details of the GPS carrier which is not provided in a GGA string.

**Example of a GGA String**

Global Positioning System Fix Data

(Source: http://aprs.gids.nl/nmea/#latlong)

$GPGGA, hhmmss.ss, lati.at, ns, long.at, ew, q, ns, d, p, a, sl, um, g, s, u, a, d, drsi$

Attribute explanation:

hhmmss.ss = UTC of position
lati.at = latitude at that position
ns = North or South
long.at = Longitude at that position
ew = East or West
q = GPS Quality indicator (0=no fix, 1=GPS fix, 2=DGPS fix)
ns = number of satellites in use

d.p = horizontal dilution of precision

a.sl = Antenna altitude above mean-sea-level

um = units of antenna altitude in meters

g.s = Geoidal separation

ug = units of geoidal separation in meters

a.d = Age of Differential GPS data in seconds

**Example of a RMC String**

Recommended minimum specific GPS/Transit data

(Source: http://aprs.gids.nl/nmea/#latlong)

$GPRMC, hhmmss.ss, DS, lati.at, ns, long.at, ew, sk, x.x, ddmmyy, m.d, ew*hh

hhmmss.ss = UTC of position fix

DS = Data status (V=navigation receiver warning)

lati.at = Latitude of fix

ns = North or South

long.at = Longitude of fix

ew = East or West

sk = Speed over ground in knots

x.x = Track made good in degrees True

ddmmyy = UT date

m.d = Magnetic variation degrees

ew = East or West

*hh = Checksum
Arc GIS software was used to map all the results of the volume, height, and density onto the aerial image to view the accuracy of the data. Soil conductivity was measured using Veris EC equipment (Figure 3-4).

The yield analysis was carried out using supervised learning models contained in Matlab 2010 (Natick, MA), which work on the principle of pattern recognition and prediction. The models used were support vector machine (SVM), partial least-squares (PLS), principal components regression (PCR), classical least-squares (CLS), and multiple linear regression (MLR) data prediction models.

All these models divided the input dataset into a training set and a testing set. The training set would recognize the pattern and develop a prediction based on that and test the predicted outcome with the test dataset.

Methods

The laser scanner systems include the need for post processing of the data before it can show results and be analyzed. The experimental data was used as an input to the algorithms to calculate the height, volume, density, and estimate yield of the canopy under consideration. Figure 3-5 shows a schematic representation of the data collection and processing.

Data Collection

The data collection was performed using CVMS software (Figure 3-6). The CVMS software collected the data of the right laser scanner, the left laser scanner, and the GPE reading along with a tag file that synchronizes the GPS point with the corresponding left and right laser scanner reading (Figure 3-7).

Once the data of the right laser scanner, left laser scanner, GPS reading and tag files were collected, they were combined into one single comma separated (.csv) file for it
to be taken as input for the algorithm. This task was done by a file converter software (Figure 3-8).

The output file of the file converter software resulted in a .csv file which acted as the input for the canopy measurement software (Figure 3-9). These .csv files were then used in the algorithm designed to calculate the height, area, volume and density (number of hits) of the tree canopy.

**Processing the Input File**

Figure 3-9 shows a sample input .csv file. One line of GPS string is followed by twelve left and twelve right laser scanner data lines. The “L” refers to the left laser data, which is followed by 181 readings in mm, corresponding to a single scan. Twelve rows of left laser scanner data are followed by 12 rows of right laser scanner data. The “R” symbolizes the start of each right scan. This pattern follows throughout the file.

While collecting the data, the tractor was assumed to be at the center of two tree rows. The horizontal distance between the two tree rows was measured and found to be approximately 7600 mm.

$$TD = \text{the horizontal distance between two tree rows}/2 = 7600/2 = 3800 \text{ mm};$$

TD was calculated to be half the horizontal distance between two consecutive tree rows as the laser was assumed to be mounted right at the center of the tractor, which in turn was running at the center of the tree row. The tree was considered to be symmetric on both sides. It was assumed to be an irregular polygon for calculation purposes. Hence, only one side of the tree was scanned for data. The area and volume calculations that followed were multiplied by two to get the complete picture. Any data in the .csv file which was greater than the TD value was set to 0 as any data that was greater than TD would
be showing false hits. The height of the laser above the ground was measured and termed LD.

The laser scanner collects 12 scans for every second. Therefore, in the .csv file, every GPS string is followed by 12 lines of left data and 12 lines of right data. Each data line corresponds to a scan from 0 to 180 degrees. The laser scanner scanned from the ground up to the sky. To remove the data that scans the ground and sky, the first 20 degrees and last 20 degrees of the each scan was set to 0.

The time between each scan was termed as dt.

\[ dt = \frac{1}{12} = 0.08 \text{ second} \quad (1 \text{ second generates 12 scans}) \]

Figure 3-10 shows a schematic representation of the variables used and the distances between the tree rows and the tractor positioning. The data in the .csv files gave the points in the polar coordinate system. To make the calculations easier, it had to be converted into Cartesian coordinates to get a 2-dimensional picture of the points.

The following formula was used for determining the x and y co-ordinates of every point in the Cartesian plane.

\[ X_{ij} = TD - d_{ij} \times \sin (\theta_{ij}) \]
\[ Y_{ij} = LD - d_{ij} \times \cos (\theta_{ij}) \]

(Lee, Ehsani, 2009)

Where, \( X_{ij} \) is the x-coordinate of the Cartesian plane and \( Y_{ij} \) is the y-coordinate of the Cartesian plane.

LD= height of the laser above the ground
TD= distance between the tractor and the tree row.
As mentioned earlier, for every GPS string in the input file, there were 12 left scans and 12 right scans. The output file was planned in a way that, every scan would give the GPS reading of the location.

**Area, Height and Volume Calculation**

The area was calculated by taking one scan at a time. This approach divided the tree into slices. Each slice was considered to be an irregular polygon. The area was calculated by first doing a convex hull on the data and then getting the area of the irregular polygon formed by the connecting data points in one scan.

\[
\text{Area}_{ij} = \text{Area of the polygon} (X_{ij}, Y_{ij})
\]

The height calculation for every slice of the tree was achieved by finding the maximum value for y-coordinate in a given scan ranging from 0-180 degrees.

\[
\text{Height}_{ij} = \max (Y_{ij})
\]

The height was further filtered to discard any data that was not a tree. Such data would usually be grass or other weeds which might have grown in the groves.

\[
\text{If } (\text{Height}-\text{SH}) > \text{LD}) \\
\text{Height} = (\text{Height} +\text{SH})/1000
\]

where,

\[
\text{SH} = \text{height of the tree trunk, which was assumed as 500 mm.} \\
\text{LD} = \text{Height of the laser above the ground. It was taken to be 1923.5 mm.}
\]

The value was divided by 1000 to convert it into meter.

**Volume calculation:**

\[
\text{Volume} = 2 * \text{Area}_{ij} * \text{veldata} * \text{dt};
\]

where,

\[
\text{Veldata} = \text{the velocity data that can be obtained from the GPS string}
\]
dt= time interval between two scans

**Experimental Setup to Test the Efficacy of the Designed Algorithm**

An experimental setup consisting of wooden boxes, piled one on top of the other and equally spaced, was designed as an imitation of the citrus groves. The tractor was mounted with the lasers and GPS to collect the data using the CVMS software. Figure 3-11 shows the experimental setup, the dimensions of the boxes, and the track where the tractor was run to collect the data. The speed information was collected at each GPS point, which was used for the volume calculations.

**Density Calculations**

The density was calculated based on the number of hits on the top, middle, and low sections of the tree canopy. To get a clear picture of the efficiency of the density calculation algorithm, the data was collected on one citrus tree. Figure 3-12 shows the tree on which the density measurement experiments were conducted.

For the density estimation, A Valencia tree with a heavily-leaved canopy was chosen. To get an estimate of how dense the tree was, first the data was collected from the given tree and this was called the 100% dense condition. The tree was deleaved manually for about 25% and the tree was said to be in 75% dense condition. This was followed by removing another 25% of the tree leaves manually and this condition was called the 50% dense condition. And finally, another 25% of the tree leaves were manually removed to get a 25% dense tree condition. After each set of leaf removal, the removed leaves were weighed and the tree was scanned for data at high and low speeds. Three repetitions were made at every speed. Figure 3-13 shows the manual measurement of the distance between the tree row and tractor, and Figure 3-14 shows the deleaving procedure of the tree.
Table 3-1 shows the amount of leaves collected for 100%, 75%, 50% and 25% of the tree density conditions along with the weight in kg of the leaves collected. The table also shows the time taken for every repetition of the data collection. In the speed column, the “H” represents the high speed and “L” represents the low speed. Figure 3-15 shows the leaves after removal.

The manually measured readings:

- Height of the tree: 104 cm
- Width of the tree: 113 cm
- Distance from laser (rope) to tree trunk: 104 cm
- Pole height: 208 cm
- Distance from rope to pole: 109 cm
- Distance travelled: 161 cm

The density calculation was carried out by the distance of the hits from the laser to the tree canopy. The smaller the distance calculated, the greater the density. The tree canopy was divided into sections. Each section represented 9 degrees in height and 4 scans in width as shown in Figure 3-17.

. The mean of the data points was calculated vertically for every scan for every section. After finding the mean vertically, for every scan, a mean of the four scans for every section was calculated as shown in the Figure 3-19.

Once the mean for every section was calculated, each tree row was divided into top, middle and low density regions. This was done to get a better understanding of how the density is divided at the top, mid and lower sections of the trees.

The top six vertical sections spanning a row were taken as the top density region (sections 1-6). The next six vertical sections spanning the row (sections 7-12) were considered to be the mid-density region and the last eight vertical sections (sections 12-20) were considered as the low-density regions.
The density readings were further classified into four zones for every region.

- Density reading lesser than 1000 mm and greater than 0 was zone 1.
- Density reading lesser than 2000 mm and greater than 1000 was zone 2.
- Density reading lesser than 3000 mm and greater than 2000 was zone 3.

The rest of the readings which showed 0 reading was given a zone of zero, symbolizing no-tree-region. Figure 3-16 shows the schematic representation of the density distances on which the zones are based. Figure 3-18 shows the flowchart of volume, height and density calculation algorithm.

**Yield Estimation**

The major aim of this study was to determine if there was a significant relationship between the canopy characteristics and the yield produced. For this, the data for yield, as in the weight of oranges per row in kg, was determined manually by weighing. The electrical conductivity (EC) of the soil and the elevation/row was also calculated.

The ability of transmitting electrical current through soil is referred to as its electrical conductivity. The EC of the soil was measured for the groves where the data was collected. It is measured in Siemens. The elevation was measured in meter and it indicates the height of the surface above the sea level.

The next goal was to find the relationship between the algorithm-calculated attributes to the manually-calculated yield. The attributes considered were: volume, height, top density, mid density, low density, EC of the soil (deep and shallow) and elevation.

\[ Y = f(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) \]

\[ X_1 = \text{Mean Volume in m}^3 \]

\[ X_2 = \text{Mean height in m} \]
X3 = Sum of top Density in m/scan
X4 = Sum of mid Density in m/scan
X5 = Sum of low Density in m/scan
X6 = Elevation in m
X7 = Deep EC in mS
X8 = Shallow EC in mS

Yield estimation modeling was carried out in Matlab using the SVM, PLS, PCR, CLS, and MLR data prediction models.

38 rows of data were taken for developing the models. Nine trees from every row were taken as a data point to collect the volume, height, density, EC and elevation data. In total there were 342 data points.

The data was averaged by volume to get a smaller number of data points to make it easier for the graphical representation. The data was averaged based on the volume data.

Yield per tree was calculated on a weighted average.

Yield per section = (volume per section * total yield per row)/total Volume per row.

This data was fed into the supervised models (PLS, PCR, SVM, CLS, MLR) in Matlab. The data was analyzed using 75% of the data as the training set and the remaining 25% of the data as the testing dataset for all the models.

The factors determining the model were the sum of top density/row, sum of mid-density/tree, sum of low-density/tree, average EC of soil/tree(shallow and deep), average elevation, mean of the measured volume/tree and mean height/tree.
Figure 3-1. Ft. Basinger citrus grove in Florida

Figure 3-2. LMS 200-30106 SICK laser scanner (Source: Innovative LIDAR 3D dynamic measurement system to estimate fruit-tree leaf area, Sanz et al., 2011)
Figure 3-3. Types of GPS A) Garmin GPS18-5Hz B) Trimble RTK 442

Figure 3-4. Veris EC equipment for measuring soil conductivity (Source: http://www.veristech.com/products/soilec.aspx)
Figure 3-5. Experimental setup
Figure 3-6. Laser data collection

Figure 3-7. Snapshot of the CVMS software
Figure 3-8. Snapshot of the file converter software

```
# CVMS File Converter v2.0
# Citrus Research and Education center - UF
# Support Email: jmaja@ufl.edu
#
# ?
# GPS: ,RMC[N], GGA[Y], PTNL[Y]
# START ANGLE, END ANGLE
0,180
```

Figure 3-9. Example resultant .csv file
Figure 3-10. Schematic representation of the left and right tree rows with the variables.

A

Distance between objects
1.2m 1.2m 1.2m 1.2m 1.2m

B

Figure 3-11. Experimental setup A) the lateral view of the boxes (top) and B) the tractor track and top view of the boxes (bottom) (the figure is not to scale)
Figure 3-12. White pole used to measure 100% hit region.

Figure 3-13. Measuring the distance between the tree rows and center of the tractor lane.
Figure 3-14. Removing the tree leaves manually
Figure 3-15. Weighed tree leaves after removal

Figure 3-16. Density reading zones
Figure 3-17. Section of the tree showing the measurements

Figure 3-18. Flowchart of the algorithm
Figure 3-19. Density Calculation Flow Chart
Table 3-1. Sensor specifications

<table>
<thead>
<tr>
<th>Model name</th>
<th>LMS 200-30106</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field of view</td>
<td>180°</td>
</tr>
<tr>
<td>Angular resolution</td>
<td>1°</td>
</tr>
<tr>
<td>Scanning range</td>
<td>8 m</td>
</tr>
<tr>
<td>Dimensions (L X W X H)</td>
<td>156 X 155 X 210 mm</td>
</tr>
</tbody>
</table>

Table 3-2. Density data at high and low speeds

<table>
<thead>
<tr>
<th>Canopy volume</th>
<th>Weight of shredded leaves (kg)</th>
<th>Speed</th>
<th>Replication</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>~100%</td>
<td></td>
<td>L</td>
<td>1</td>
<td>19.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>12.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>12.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H</td>
<td>1</td>
<td>5.51</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>4.98</td>
</tr>
<tr>
<td>~75%</td>
<td>0.8</td>
<td>L</td>
<td>1</td>
<td>10.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>8.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>9.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H</td>
<td>1</td>
<td>8.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>6.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>4.86</td>
</tr>
<tr>
<td>~50%</td>
<td>1.3</td>
<td>L</td>
<td>1</td>
<td>12.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>12.51</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>12.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H</td>
<td>1</td>
<td>4.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>4.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>5.21</td>
</tr>
<tr>
<td>~25%</td>
<td>2.1</td>
<td>L</td>
<td>1</td>
<td>10.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>11.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>12.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H</td>
<td>1</td>
<td>5.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>5.02</td>
</tr>
</tbody>
</table>
CHAPTER 4
RESULTS AND DISCUSSION

This chapter provides the graphs and resultant tables.

Experimental Data Results

It was seen that the measured height ranged from 2.2 to 2.7 m for the experimental setup with the boxes or minibulks. The actual height of the minibulks were 2.3 m and 2.2 m. The minibulks of 1.16 m were not considered as they were taken as short trees by the algorithm. The experimental results for the density calculation considering a single tree gave satisfactory results with coefficient of determination of 0.94 with various density scenarios. Figure 4-1 shows that that the density decreases as the percentage of the leaves removed increases.

Results on the Grove Data

Table 4-1 shows the output of density at a given GPS location with top, middle and low densities accompanied by it’s zone. The density decreased from zone 1 to zone 3, and zone 0 represented no tree detected. Table 4-2 shows the output file for the volume and height data along with their respective GPS locations.

It was seen that as the height of the tree increases, there is a considerable growth in the volume of the tree, owing to the foliage increase. There were also specific cases in the output data which showed a low estimated volume despite a good height. This could be a result of a unhealthy tree in the grove.

Figure 4-2, shows the variation of yield based on the EC data. Though there was no relationship as such when the EC data was mapped to the manually collected yield, it was seen that on removing the EC data from the yiled estimation model there was a
considerable increase in the percentage error. This shows that EC data does play a role in the yield estimation in citrus groves.

In Figure 4-3, the relationship between the elevation and the citrus yield is mapped. Similar to EC data, the correlation between the yield and the elevation data did not show good results; yet, in the yield estimation modelling, on removing the elevation data from the analysis, the percentage error increased considerably.

**Discussion**

In the experimental data analysis, we could see that the volume estimation gave a percent error of 8.2%. This error could be related to the fact that, the objects used for the experiments were rectangular minibulks and not trees as such. The geometric differences between the rectangular minibulk and the tree has to be taken into consideration as the volume calculation was carried out with the same algorithm, keeping the geometric characteristic of a tree in mind. That is, the distance between the tree row and the laser is exactly TD when it comes to the rectangular boxes, but in the real scenario, the foliage of the tree overlaps this distance.

The height estimation on the other hand produced a percent error of 2.5%. This is because the algorithm calculates the height of every section by taking the mean of maximum point it records, which is similar in both cases of the rectangular minibulks and citrus trees.

The measured volume, height and density were overlaid on the aerial image using the ArcGIS software (Figure 4-4) and the regions where there were missing trees were precisely indicated by the algorithm. The estimated height in the groves was at par with the actual tree sizes.
The volume and height data were very much in sync with the actual trees in the groves under consideration. The algorithm was able to predict the sparse regions and the dense regions in the groves accurately. The output file gave a clear picture of the density distribution. In areas with no trees, the volume, height and density gave a result of 0, signifying the absence of the tree.

The yield estimation was carried out with five different regression models in a MATLAB 2010b environment. The factors used in developing this model were the top-density, mid-density, low-density, volume, height, EC (shallow and deep) and elevation. Table 4-3, Table 4-4, Table 4-5, Table 4-6, and Table 4-7 give the analysis statistics for the PLS, PCR, SVM, MLR and CLS regression models, respectively.

The five models produced approximately the same results with slight variations. The SVM and CLS model gave a percent error of 13.8% and 15.5%, respectively, when compared to the 15% error in the remaining three models.

It was seen that the electrical conductivity of the soil (EC), was one of the contributors to the fruit yield. This was proven when the EC factor was removed from the analysis; the percentage error increased for all the prediction models. With EC data, the results showed a percentage error between 13-16%. The models gave a very low coefficient of determination for all the models, yet the percentage error was comparatively better. This was because of the varying yield produced in the different rows in the grove.

If there is an automated method for the fruit counting, the yield estimation could provide better results (Zaman et al., 2006). Yield prediction mainly depends on the health of the tree along with the site specific management. If the health of the grove is
incorporated in this model of yield prediction, the percentage error in the yield prediction can be comparatively lowered.

The algorithm was run on Java and Matlab environments. Though the results were the same, the time complexity showed a vast variation. The java code took an average of 1.5 seconds to execute for files of size 8.5 MB and the same file took about 1200 seconds in Matlab. This clearly showed that Java performed better than Matlab environment in terms of the processing time.

Conclusion

The volume, height and density data calculated from the algorithm produced satisfactory results. The algorithm gave a percent error of 2.5% for height when compared to the actual minibulk height. The volume data measured 3.5 m³ for individual mini-bulks which had an actual measurement of 3.23 m³ and gave a percent error of 8.2%. The density measurements calculated by the algorithm showed that the density decreased as the tree was defoliated. Figure 4-1 shows the graph for the density measurements.

The supervised models provided percentage error between 13-16% for PLS, PCR, SVM, MLR, CLS between the manually collected yield and the predicted yield. This study concentrated on analyzing the yield with just the physical characteristics of the tree and the soil condition.

Yield is dependent on many other factors such as climate, location, health and age of the tree. The yield in citrus groves is largely dependent upon the health conditions of the trees and this factor was not taken into consideration in this study.
**Future Work**

The health aspect of the groves has to be taken into consideration for the estimation of fruit yield. An approach to consider the health estimate of the groves using the Normalized Difference Vegetation Index (NDVI) data is in progress which could be used in the yield estimation to provide a better insight.
Figure 4-1. Density readings for a single tree

Figure 4-2. Relationship between the EC and yield
Figure 4-3. Relationship between the elevation and yield

Figure 4-4. Measured volume overlaid on the aerial image using Arc GIS software
### Table 4-1. Density output file

<table>
<thead>
<tr>
<th>Lat</th>
<th>Long</th>
<th>TD (mm/scan)</th>
<th>Zone</th>
<th>MD (mm/scan)</th>
<th>Zone</th>
<th>LD (mm/scan)</th>
<th>Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.385192</td>
<td>-81.143769</td>
<td>1221</td>
<td>2</td>
<td>551</td>
<td>1</td>
<td>1657</td>
<td>2</td>
</tr>
<tr>
<td>27.385190</td>
<td>-81.143768</td>
<td>1221</td>
<td>2</td>
<td>551</td>
<td>1</td>
<td>1657</td>
<td>2</td>
</tr>
<tr>
<td>27.385188</td>
<td>-81.143768</td>
<td>1749</td>
<td>2</td>
<td>582</td>
<td>1</td>
<td>1914</td>
<td>2</td>
</tr>
<tr>
<td>27.385186</td>
<td>-81.143767</td>
<td>1749</td>
<td>2</td>
<td>582</td>
<td>1</td>
<td>1914</td>
<td>2</td>
</tr>
<tr>
<td>27.385184</td>
<td>-81.143767</td>
<td>1749</td>
<td>2</td>
<td>582</td>
<td>1</td>
<td>1914</td>
<td>2</td>
</tr>
<tr>
<td>27.385183</td>
<td>-81.143766</td>
<td>1749</td>
<td>2</td>
<td>582</td>
<td>1</td>
<td>1914</td>
<td>2</td>
</tr>
<tr>
<td>27.385181</td>
<td>-81.143765</td>
<td>1770</td>
<td>2</td>
<td>623</td>
<td>1</td>
<td>2090</td>
<td>3</td>
</tr>
<tr>
<td>27.385179</td>
<td>-81.143765</td>
<td>1770</td>
<td>2</td>
<td>623</td>
<td>1</td>
<td>2090</td>
<td>3</td>
</tr>
</tbody>
</table>

### Table 4-2. Volume and height output file

<table>
<thead>
<tr>
<th>Lat</th>
<th>Long</th>
<th>Volume (m³)</th>
<th>Height (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.385192</td>
<td>-81.143769</td>
<td>0.204</td>
<td>3.93</td>
</tr>
<tr>
<td>27.385190</td>
<td>-81.143768</td>
<td>0.198</td>
<td>3.98</td>
</tr>
<tr>
<td>27.385188</td>
<td>-81.143768</td>
<td>0.154</td>
<td>3.23</td>
</tr>
<tr>
<td>27.385186</td>
<td>-81.143767</td>
<td>0.156</td>
<td>3.37</td>
</tr>
<tr>
<td>27.385184</td>
<td>-81.143767</td>
<td>0.214</td>
<td>4.09</td>
</tr>
<tr>
<td>27.385183</td>
<td>-81.143766</td>
<td>0.179</td>
<td>3.39</td>
</tr>
<tr>
<td>27.385181</td>
<td>-81.143765</td>
<td>0.151</td>
<td>3.16</td>
</tr>
<tr>
<td>27.385179</td>
<td>-81.143765</td>
<td>0.150</td>
<td>3.18</td>
</tr>
</tbody>
</table>

### Table 4-3. PLS statistics

<table>
<thead>
<tr>
<th>Measured yield (kg/ha)</th>
<th>Predicted yield (kg/ha)</th>
<th>% error</th>
<th>Mean % error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.37</td>
<td>14.07</td>
<td>14.03</td>
<td>15.0</td>
</tr>
<tr>
<td>12.59</td>
<td>13.35</td>
<td>6.04</td>
<td></td>
</tr>
<tr>
<td>12.21</td>
<td>14.62</td>
<td>19.77</td>
<td></td>
</tr>
<tr>
<td>12.38</td>
<td>14.99</td>
<td>21.08</td>
<td></td>
</tr>
<tr>
<td>11.96</td>
<td>13.09</td>
<td>9.47</td>
<td></td>
</tr>
<tr>
<td>15.92</td>
<td>12.81</td>
<td>19.53</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4-4. PCR statistics

<table>
<thead>
<tr>
<th>Measured yield (kg/ha)</th>
<th>Predicted yield (kg/ha)</th>
<th>% error</th>
<th>Mean % error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.37</td>
<td>14.07</td>
<td>14.03</td>
<td>15.0</td>
</tr>
<tr>
<td>12.59</td>
<td>13.35</td>
<td>6.04</td>
<td></td>
</tr>
<tr>
<td>12.21</td>
<td>14.62</td>
<td>19.77</td>
<td></td>
</tr>
<tr>
<td>12.38</td>
<td>14.99</td>
<td>21.08</td>
<td></td>
</tr>
<tr>
<td>11.96</td>
<td>13.09</td>
<td>9.47</td>
<td></td>
</tr>
<tr>
<td>15.92</td>
<td>12.81</td>
<td>19.53</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4-5. SVM statistics

<table>
<thead>
<tr>
<th>Measured yield (kg/ha)</th>
<th>Predicted yield (kg/ha)</th>
<th>% error</th>
<th>Mean % error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.37</td>
<td>14.26</td>
<td>12.88</td>
<td>13.8</td>
</tr>
<tr>
<td>12.59</td>
<td>14.18</td>
<td>12.58</td>
<td></td>
</tr>
<tr>
<td>12.21</td>
<td>14.13</td>
<td>15.73</td>
<td></td>
</tr>
<tr>
<td>12.38</td>
<td>14.23</td>
<td>14.92</td>
<td></td>
</tr>
<tr>
<td>11.96</td>
<td>13.86</td>
<td>15.86</td>
<td></td>
</tr>
<tr>
<td>15.92</td>
<td>14.16</td>
<td>11.08</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4-6. MLR statistics

<table>
<thead>
<tr>
<th>Measured yield (kg/ha)</th>
<th>Predicted yield (kg/ha)</th>
<th>% error</th>
<th>Mean % error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.37</td>
<td>14.07</td>
<td>14.03</td>
<td>15.0</td>
</tr>
<tr>
<td>12.59</td>
<td>13.35</td>
<td>6.04</td>
<td></td>
</tr>
<tr>
<td>12.21</td>
<td>14.62</td>
<td>19.77</td>
<td></td>
</tr>
<tr>
<td>12.38</td>
<td>14.99</td>
<td>21.08</td>
<td></td>
</tr>
<tr>
<td>11.96</td>
<td>13.09</td>
<td>9.47</td>
<td></td>
</tr>
<tr>
<td>15.92</td>
<td>12.81</td>
<td>19.53</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4-7. CLS statistics

<table>
<thead>
<tr>
<th>Measured yield (kg/ha)</th>
<th>Predicted yield (kg/ha)</th>
<th>% error</th>
<th>Mean % error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.37</td>
<td>12.37</td>
<td>24.42</td>
<td>15.5</td>
</tr>
<tr>
<td>12.59</td>
<td>11.74</td>
<td>6.80</td>
<td></td>
</tr>
<tr>
<td>12.21</td>
<td>11.46</td>
<td>6.12</td>
<td></td>
</tr>
<tr>
<td>12.38</td>
<td>12.07</td>
<td>2.49</td>
<td></td>
</tr>
<tr>
<td>11.96</td>
<td>9.24</td>
<td>22.76</td>
<td></td>
</tr>
<tr>
<td>15.92</td>
<td>11.10</td>
<td>30.29</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A
MATLAB CODE FOR HEIGHT, VOLUME, AND DENSITY CALCULATION

clc;
clear all;
umlines=0;

format long e;
areavarleft=0;
areavolleft=0;
areahtleft=0;
areavarright=0;
areavolright=0;
areahtright=0;

%%Open the file
inputfile = input('Enter File Name with extension: 
', 's');
 fid=fopen(inputfile);
 fid=fopen(inputfile);

%% get the number of lines in the file in numlines
for i=1:50000
 A=fgetl(fid);
 if A==-1
   break;
 else
   numlines=numlines+1;
 end
end
numlines=numlines-9;
numscans=numlines/26;

%% Interpolation of the GPS Points for both latitude and longitude

ld1= (floor(latdivide/100)*100);
ld= floor(latdivide/100);
gd1= (floor(longdivide/100)*100);
gd= floor(longdivide/100);
ldgga= (floor(latdividegga/100)*100);
ldgga= floor(latdividegga/100);
gd1gga= (floor(longdividegga/100)*100);
gdgga= floor(longdividegga/100);
eval(['data=xlsread(inputfile,1,''C10:GA'',num2str(numlines+9),'''');'])

%%% get the left laser data, right laser data and velocity of the tractor in
%%% variables leftdata, rightdata and veldata respectively (velocity in mm/s)

leftdata=zeros(numscans,12,181);
rightreading=zeros(numscans,12,181); %%% rightreading to leftreading
rightdata=zeros(numscans,12,181);
Arealeft=zeros(numsca
Arearight=zeros(numscans,12);

for i=1:numscans
    leftdata(i,:,:)=data((i-1)*26+3:(i-1)*26+14,:); %%%leftdata to rightdata
    rightreading(i,:,:)=data((i-1)*26+15:(i-1)*26+26,:); %%%changes made from
    rightreading to leftreading
end

%%% Input data from rightreading variable to rightdata variable from 181:1

for i=1:numscans
    for j=1:12
        m=181;
        for k=1:181
            rightdata(i,j,m)=rightreading(i,j,k);  % changes made from
            rightdata to leftdata
            m=m-1;
        end
    end
end

cellarrayleft= cell(numscans*12,7);
cellarrayright= cell(numscans*12,7);

%%% Distance between two rows of tree

distance= 8000 ; %%% changes made 3302
TD = distance/2;

%%% Height of the laser from the ground
SH = 500; %%% changed from 1400 to 500 to 1981.2

%%% Scan interval between 2 scans in seconds

dt= 0.0833;

%%% distance greater than 3151.78 mm is considered 0
for i=1:numscans
    for j=1:12
        for k=1:181
            if (leftdata(i,j,k)>TD) % previously 3151.78
                leftdata(i,j,k)=0;
            else
                leftdata(i,j,k)= leftdata(i,j,k);
            end
        end
    end
for i=1:numscans
    for j=1:12
        for k=1:181
            if (rightdata(i,j,k)>TD) % 2248 for 2
                rightdata(i,j,k)=0;
            else
                rightdata(i,j,k)= rightdata(i,j,k);
            end
        end
    end
end

ty=1;

%% Anything below 20 degree is considered 0 as it covers the ground
leftdata(:,:,1:20)=0;
rightdata(:,:,1:20)=0;
leftdata(:,:,171:181)=0;
rightdata(:,:,171:181)=0;

%% Transforming polar coordinates into Cartesian coordinates
for i=1:numscans
    for j=1:12
        for k=1:181
            tempyleft(ty)= leftdata(i,j,k);
            ty=ty+1;
            if (leftdata(i,j,k) == 0)
                xleft(i,j,k)=0;
                yleft(i,j,k)=0;
            else
                xleft(i,j,k)= abs(leftdata(i,j,k)*cos(degtorad(k-1))); % previously TD - (leftdata(i,j,k)*cos(degtorad(k-1)));
                yleft(i,j,k)= abs((leftdata(i,j,k)*sin(degtorad(k-1)))); % changes made SH + before
            end
            if (rightdata(i,j,k)== 0)
                xright(i,j,k)=0;
        end
    end
end
end
yright(i,j,k)=0;
    else
xright(i,j,k)=abs(rightdata(i,j,k)*cos(degtorad(k-1)));%%previously TD - (rightdata(i,j,k)*cos(degtorad(k-1)))
yright(i,j,k)= abs((rightdata(i,j,k)*sin(degtorad(k-1))));%%changes made SH + before
end
end
end

%%Total area calculation
Tleft=0;
Tright=0;

%% Putting the GPRS string and the volume in an excel file

% First 13 locations cover comma's until the last variable.
    outtopdenright=cell(1000);
outzetopright=cell(1000);
outmiddenright=cell(1000);
outzonenmidright=cell(1000);
outbotdenright=cell(1000);
outzonenlowright=cell(1000);
outtopdenleft=cell(1000);
outzetopleft=cell(1000);
outmiddenleft=cell(1000);
outzonenmidleft=cell(1000);
outbotdenleft=cell(1000);
outzonenlowleft=cell(1000);
sectionavgleft=cell(1000);
sectionavgright=cell(1000);
lineavgleft=cell(1000);
lineavgright=cell(1000);
topdenleft=cell(1000);
middenleft=cell(1000);
bodenleft=cell(1000);
topdenright=cell(1000);
middenright=cell(1000);
bodenright=cell(1000);
zenetopleft=cell(1000);
zenenmidleft=cell(1000);
zenelowleft=cell(1000);
zenetopright=cell(1000);
zenemidright=cell(1000);
zenelowright=cell(1000);

Vleft=0;
Vright=0;

eval(['[num text
raw]=xlsread(inputfile,1,''B10:B10'\',num2str(numlines),'''\']]);

fid=fopen(inputfile);
A=fgetl(fid);
fclose(fid);

locs=findstr(A,',');

t=1;

for i=1:numscans

GPSstring= text((i-1)*26+1,1);

GGA = strcmp(GPSstring,'$GPGGA');
RMC = strcmp(GPSstring,'$GPRMC');
GNRMC = strcmp(GPSstring,'$GNRMC');

if (GGA)

degreelat=data((i-1)*26+1,2);
veldata(i) = 178450.720224065;
    if ( i< numscans) temp1= data((i)*26+1,2); end
    deglat= str2double(num2str(((degreelat-ldegga)/60)+ ldgga),15));
    deglat1= str2double(num2str(((temp1-ldegga)/60)+ ldgga),15));
    if(deglat<0)
      deglatdiv= -abs(deglat1-deglat)/12;
    else
      deglatdiv=abs(deglat1-deglat)/12;
    end

degreelong=data((i-1)*26+1,4);
    if ( i< numscans) temp2= data((i)*26+1,4); end
    deglong= str2double(num2str(0-(((degreelong-gdegga)/60)+gdgga),15));
    deglong1= str2double(num2str(0-(((temp2-gdegga)/60)+gdgga),15));
    if(deglong<0)
      deglongdiv= -abs(deglong1-deglong)/12;
    else
      deglongdiv=abs(deglong1-deglong)/12;
    end

elseif (RMC || GNRMC)

degreelat=data((i-1)*26+1,3);
veldatanots=data((i-1)*26+1,8);
veldata(i) = veldatanots/0.00194384449;
    if ( i< numscans) temp1= data((i)*26+1,3); end
    deglat= str2double(num2str(((degreelat-ldegga)/60)+ ldgga),15));

\[
\text{deglat1} = \text{str2double(num2str(((temp1-ld1)/60) + ld),15));}
\]
\[
\text{if (deglat<0)}
\]
\[
\text{deglatdiv} = -\text{abs(deglat1-deglat)}/12;
\]
\[
\text{else}
\]
\[
\text{deglatdiv} = \text{abs(deglat1-deglat)}/12;
\]
\[
\text{end}
\]
\[
\text{degreelong} = \text{data((i)*26+1,5)};
\]
\[
\text{if (i< numscans) temp2 = \text{data((i)*26+1,5); end}
\]
\[
\text{deglng1} = \text{str2double(num2str(0-((degreelong-gd1)/60)+gd),15));}
\]
\[
\text{if (deglng<0)}
\]
\[
\text{deglngdiv} = -\text{abs(deglng1-deglng)}/12;
\]
\[
\text{else}
\]
\[
\text{deglngdiv} = \text{abs(deglng1-deglng)}/12;
\]
\[
\text{end}
\]
\[
\text{end}
\]
\[
\text{tu} = 1;
\]
\[
\text{for j = 1:12}
\]
\[
\% \text{tempxleft(i,j,21:170)} = \text{smooth(TD-xleft(i,j,21:170));}
\]
\[
\text{for ak=1:180}
\]
\[
\text{if (ak==180)}
\]
\[
\text{if(xleft(i,j,180)<xleft(i,j,179))}
\]
\[
\text{Arealeft(i,j) = Arealeft(i,j)+ ((abs((TD-xleft(i,j,179)* yleft(i,j,1)-(TD-xleft(i,j,1)* yleft(i,j,179))))/2);}
\]
\[
\text{else}
\]
\[
\text{Arealeft(i,j) = Arealeft(i,j)+ ((abs((TD-xleft(i,j,180)* yleft(i,j,1)-(TD-xleft(i,j,1)* yleft(i,j,180))))/2);}
\]
\[
\text{end}
\]
\[
\text{Arealeft(i,j) = Arealeft(i,j)+ ((abs((TD-xright(i,j,179)* yright(i,j,1)-(TD-xright(i,j,1)* yright(i,j,179))))/2);}
\]
\[
\text{else}
\]
\[
\text{Arealeft(i,j) = Arealeft(i,j)+ ((abs((TD-xright(i,j,180)* yright(i,j,1)-(TD-xright(i,j,1)* yright(i,j,180))))/2);}
\]
\[
\text{end}
\]
\[
\text{gf} = ak-1;
\]
\[
\text{if (ak==1) gf = 1; end}
\]
\[
\text{if(xleft(i,j,ak)<xleft(i,j,gf))}
\]
\[
\text{Arealeft(i,j) = Arealeft(i,j)+ ((abs((TD-xleft(i,j,ak)* yleft(i,j,ak+1)-(TD-xleft(i,j,ak+1)* yleft(i,j,gf))))/2);}
\]
\[
\text{else}
\]
\[
\text{Arealeft(i,j) = Arealeft(i,j)+ ((abs((TD-xleft(i,j,ak)* yleft(i,j,ak+1)-(TD-xleft(i,j,ak+1)* yleft(i,j,ak))))/2);}
\]
\[
\text{end}
\]
\[
\text{if (ak==1) gf = 1;}
\]
\[
\text{if(xright(i,j,ak)<xright(i,j,gf))}
\]
Arearight(i,j) = Arearight(i,j) + ((abs((TD-xright(i,j,gf)* yright(i,j,ak+1) 
- (TD-xright(i,j,ak+1)* yright(i,j,gf))))/2);
else
    Arearight(i,j) = Arearight(i,j) + ((abs((TD-
    xright(i,j,ak)* yright(i,j,ak+1)) 
- (TD-xright(i,j,ak+1)* yright(i,j,ak))))/2);
end

end

end

% Arealeft(i,j)=
polyarea(tempxleft(i,j,21:170),yleft(1,j,21:170));
%
% areavarleft=(areavarleft+Arealeft(i,j));
% tempxright(i,j,21:170) = smooth(TD-xright(i,j,21:170));
% Arearight(i,j)=
polyarea(tempxright(i,j,21:170),yright(1,j,21:170));
%
% hleft(j)=max(yleft(i,j,21:170));
hright(j)=max(yright(i,j,21:170));

areavarright=(areavarright+Arearight(i,j));

if((hleft(j)-SH) > 1923.5) % 1400 for speed 1; 1379 for 2 ; 1365 for 4 ; 1347 for 8
    cellarrayleft{t,6} = (hleft(j)+SH)/1000;
    areahtleft=areahtleft+cellarrayleft{t,6};
    cellarrayleft{t,5}=2*Arealeft(i,j)*veldata(i)*dt/1000000000000;
    areavolleft=areavolleft+cellarrayleft{t,5};
else
    cellarrayleft{t,6} = 0;
    cellarrayleft{t,5}=0;
end

if((hright(j)-SH) > 1923.5)
    cellarrayright{t,6} = (hright(j)+SH)/1000;
    areahtright=areahtright+cellarrayright{t,6};
    cellarrayright{t,5}=2*Arearight(i,j)*veldata(i)*dt/1000000000000;
    areavolright=areavolright+cellarrayright{t,5};
else
    cellarrayright{t,6} = 0;
    cellarrayright{t,5}=0;
end
if\( j==1 \)
cellarrayleft{t,1}=deglat;
cellarrayleft{t,3}=deglong;
cellarrayright{t,1}=deglat;
cellarrayright{t,3}=deglong;
else
  if\( \text{deglat} < \text{deglat1} \)
    if\( \text{deglatdiv} < 0 \)
      cellarrayleft{t,1}=cellarrayleft{t-1,1}-\text{deglatdiv};
      cellarrayright{t,1}=cellarrayright{t-1,1}-\text{deglatdiv};
      else
        cellarrayleft{t,1}=cellarrayleft{t-1,1}+\text{deglatdiv};
        cellarrayright{t,1}=cellarrayright{t-1,1}+\text{deglatdiv};
    end
  end
  else
    if\( \text{deglat} < 0 \)
      cellarrayleft{t,1}=cellarrayleft{t-1,1}+\text{deglatdiv};
      cellarrayright{t,1}=cellarrayright{t-1,1}+\text{deglatdiv};
      else
        cellarrayleft{t,1}=cellarrayleft{t-1,1}-\text{deglatdiv};
        cellarrayright{t,1}=cellarrayright{t-1,1}-\text{deglatdiv};
      end
  end
end
if\( \text{deglong} < \text{deglong1} \)
  if\( \text{deglongdiv} < 0 \)
    cellarrayleft{t,3}=cellarrayleft{t-1,3}-\text{deglongdiv};
    cellarrayright{t,3}=cellarrayright{t-1,3}-\text{deglongdiv};
    else
      cellarrayleft{t,3}=cellarrayleft{t-1,3}+\text{deglongdiv};
      cellarrayright{t,3}=cellarrayright{t-1,3}+\text{deglongdiv};
    end
  end
else
  if\( \text{deglong} < 0 \)
    cellarrayleft{t,3}=cellarrayleft{t-1,3}+\text{deglongdiv};
    cellarrayright{t,3}=cellarrayright{t-1,3}+\text{deglongdiv};
    else
      cellarrayleft{t,3}=cellarrayleft{t-1,3}-\text{deglongdiv};
      cellarrayright{t,3}=cellarrayright{t-1,3}-\text{deglongdiv};
    end
end
cellarrayleft{t,2}='N';
cellarrayleft{t,4}='W';
if((cellarrayleft{t,5}==0)||(cellarrayleft{t,6}==0))
cellarrayleft{t,6}=0;cellarrayleft{t,5}=0;
end

cellarrayright{t,2}='N';
cellarrayright{t,4}='W';
if((cellarrayright{t,5}==0)||(cellarrayright{t,6}==0))
cellarrayright{t,6}=0;cellarrayright{t,5}=0;
end

t=t+1;
end

g = (numscans*12);
for i=1:g
    if ((cellarrayleft{i,6})< 0.5)
cellarrayleft{i,7}=0;

    elseif ((cellarrayleft{i,6})< 1.2)
cellarrayleft{i,7}=1;

    elseif(cellarrayleft{i,6}<2.13 )
cellarrayleft{i,7}=2;

    elseif(cellarrayleft{i,6}<2.74)
cellarrayleft{i,7}=3;

    elseif(cellarrayleft{i,6}<3.35)
cellarrayleft{i,7}=4;

    elseif (cellarrayleft{i,6}>3.35)
cellarrayleft{i,7}=5;
end

cellarrayleft{i,5}= num2str(cellarrayleft{i,5});
cellarrayleft{i,6}= num2str(cellarrayleft{i,6});
cellarrayleft{i,7}= num2str(cellarrayleft{i,7});

    if (cellarrayright{i,6}<0.5)
cellarrayright{i,7}=0;

    elseif (cellarrayright{i,6}<1.2)  
      cellarrayright{i,7}=1;

    elseif(cellarrayright{i,6}<2.13)  
      cellarrayright{i,7}=2;

    elseif(cellarrayright{i,6}<2.74)  
      cellarrayright{i,7}=3;

    elseif(cellarrayright{i,6}<3.35)  
      cellarrayright{i,7}=4;

    elseif (cellarrayright{i,6}>3.35)  
      cellarrayright{i,7}=5;
    end

    cellarrayright{i,5}= num2str(cellarrayright{i,5});  
    cellarrayright{i,6}= num2str(cellarrayright{i,6});  
    cellarrayright{i,7}= num2str(cellarrayright{i,7});

end

count=1;
denvar=1;
  for itemp= 1:numscans  
    for divscan =1:3
      ktemp=1;
      if(divscan==1)  jtemp=1;  
      elseif(divscan==2)  jtemp=5;  
      elseif(divscan==3)  jtemp=9;  
      end

      for divdeg= 1:20

        alleft=[xleft(itemp,jtemp,ktemp),xleft(itemp,jtemp,ktemp+1),xleft(itemp,jtemp,ktemp+2),xleft(itemp,jtemp,ktemp+3),xleft(itemp,jtemp,ktemp+4),xleft(itemp,jtemp,ktemp+5),xleft(itemp,jtemp,ktemp+6),xleft(itemp,jtemp,ktemp+7),xleft(itemp,jtemp,ktemp+8)];

        alrigh=[xright(itemp,jtemp,ktemp),xright(itemp,jtemp,ktemp+1),xright(itemp,jtemp,ktemp+2),xright(itemp,jtemp,ktemp+3),xright(itemp,jtemp,ktemp+4),xright(itemp,jtemp,ktemp+5),xright(itemp,jtemp,ktemp+6),xright(itemp,jtemp,ktemp+7),xright(itemp,jtemp,ktemp+8)];

        alleft= alleft(alleft~= 0);
        alrigh= alrigh(alrigh~= 0);
lineavgleft{1}= nanmean(allleft); 
lineavgright{1}= nanmean(allright);
jtemp=jtemp+1;

a2left=[xleft(itemp,jtemp,ktemp),xleft(itemp,jtemp,ktemp+1),xleft(itemp,jtemp,ktemp+2),xleft(itemp,jtemp,ktemp+3),xleft(itemp,jtemp,ktemp+4),xleft(itemp,jtemp,ktemp+5),xleft(itemp,jtemp,ktemp+6),xleft(itemp,jtemp,ktemp+7),xleft(itemp,jtemp,ktemp+8)];
a2right=[xright(itemp,jtemp,ktemp),xright(itemp,jtemp,ktemp+1),xright(itemp,jtemp,ktemp+2),xright(itemp,jtemp,ktemp+3),xright(itemp,jtemp,ktemp+4),xright(itemp,jtemp,ktemp+5),xright(itemp,jtemp,ktemp+6),xright(itemp,jtemp,ktemp+7),xright(itemp,jtemp,ktemp+8)];

a2left= a2left(a2left~= 0); 
a2right= a2right(a2right~= 0);
lineavgleft{2}= nanmean(a2left);
lineavgright{2}= nanmean(a2right);

jtemp=jtemp+1;

a3left=[xleft(itemp,jtemp,ktemp),xleft(itemp,jtemp,ktemp+1),xleft(itemp,jtemp,ktemp+2),xleft(itemp,jtemp,ktemp+3),xleft(itemp,jtemp,ktemp+4),xleft(itemp,jtemp,ktemp+5),xleft(itemp,jtemp,ktemp+6),xleft(itemp,jtemp,ktemp+7),xleft(itemp,jtemp,ktemp+8)];
a3right=[xright(itemp,jtemp,ktemp),xright(itemp,jtemp,ktemp+1),xright(itemp,jtemp,ktemp+2),xright(itemp,jtemp,ktemp+3),xright(itemp,jtemp,ktemp+4),xright(itemp,jtemp,ktemp+5),xright(itemp,jtemp,ktemp+6),xright(itemp,jtemp,ktemp+7),xright(itemp,jtemp,ktemp+8)];

a3left= a3left(a3left ~= 0); 
a3right= a3right(a3right ~= 0);
lineavgleft{3}= nanmean(a3left);
lineavgright{3}= nanmean(a3right);

jtemp=jtemp+1;

a4left=[xleft(itemp,jtemp,ktemp),xleft(itemp,jtemp,ktemp+1),xleft(itemp,jtemp,ktemp+2),xleft(itemp,jtemp,ktemp+3),xleft(itemp,jtemp,ktemp+4),xleft(itemp,jtemp,ktemp+5),xleft(itemp,jtemp,ktemp+6),xleft(itemp,jtemp,ktemp+7),xleft(itemp,jtemp,ktemp+8)];
a4right=[xright(itemp,jtemp,ktemp),xright(itemp,jtemp,ktemp+1),xright(itemp,jtemp,ktemp+2),xright(itemp,jtemp,ktemp+3),xright(itemp,jtemp,ktemp+4),xright(itemp,jtemp,ktemp+5),xright(itemp,jtemp,ktemp+6),xright(itemp,jtemp,ktemp+7),xright(itemp,jtemp,ktemp+8)];

a4left= a4left(a4left~= 0); 
a4right= a4right(a4right~= 0);
lineavgleft{4}= nanmean(a4left);
lineavgright{4} = nanmean(a4right);

jtemp=jtemp+1;

if(divscan==1)
    jtemp=1;
else if(divscan==2)
    jtemp=5;
elseif(divscan==3)
    jtemp=9;
end

lineavgleft{1}=lineavgleft{1}(lineavgleft{1}~= '0');
lineavgleft{2}=lineavgleft{2}(lineavgleft{2}~= '0');
lineavgleft{3}=lineavgleft{3}(lineavgleft{3}~= '0');
lineavgleft{4}=lineavgleft{4}(lineavgleft{4}~= '0');
lineavgright{1}=lineavgright{1}(lineavgright{1}~= '0');
lineavgright{2}=lineavgright{2}(lineavgright{2}~= '0');
lineavgright{3}=lineavgright{3}(lineavgright{3}~= '0');
lineavgright{4}=lineavgright{4}(lineavgright{4}~= '0');

sectionavgleft(count)= nanmean([lineavgleft{1},lineavgleft{2},lineavgleft{3},lineavgleft{4}]);
sectionavgright(count)= nanmean([lineavgright{1},lineavgright{2},lineavgright{3},lineavgright{4}]);

count=count+1;
denvar=denvar+1;
ktemp=ktemp+9;
end
end

end
end

while( tintin ~= count)

DLT= [sectionavgleft(tintin),sectionavgleft(tintin+1),sectionavgleft(tintin+2),sectionavgleft(tintin+3),sectionavgleft(tintin+4),sectionavgleft(tintin+5)];
DLT=DLT(DLT~=0);
topdenleft{tmpl} = nanmean(DLT);
DLM =
[sectionavgleft{tintin+6}, sectionavgleft{tintin+7}, sectionavgleft{tintin+8}, sectionavgleft{tintin+9}, sectionavgleft{tintin+10}, sectionavgleft{tintin+11}];
DLM=DLM(DLM~=0);
middenleft{tmp1} = nanmean(DLM);

DBL =
[sectionavgleft{tintin+12}, sectionavgleft{tintin+13}, sectionavgleft{tintin+14}, sectionavgleft{tintin+15}, sectionavgleft{tintin+16}, sectionavgleft{tintin+17}, sectionavgleft{tintin+18}, sectionavgleft{tintin+19}];
DBL=DBL(DBL~=0);
botdenleft{tmp1} = nanmean(DBL);

DRT =
[sectionavgright{tintin}, sectionavgright{tintin+1}, sectionavgright{tintin+2}, sectionavgright{tintin+3}, sectionavgright{tintin+4}, sectionavgright{tintin+5}];
DRT=DRT(DRT~=0);
topdenright{tmp1} = nanmean(DRT);

DRM =
[sectionavgright{tintin+6}, sectionavgright{tintin+7}, sectionavgright{tintin+8}, sectionavgright{tintin+9}, sectionavgright{tintin+10}, sectionavgright{tintin+11}];
DRM=DRM(DRM~=0);
middenright{tmp1} = nanmean(DRM);

DRB =
[sectionavgright{tintin+12}, sectionavgright{tintin+13}, sectionavgright{tintin+14}, sectionavgright{tintin+15}, sectionavgright{tintin+16}, sectionavgright{tintin+17}, sectionavgright{tintin+18}, sectionavgright{tintin+19}];
DRB=DRB(DRB~=0);
botdenright{tmp1} = nanmean(DRB);

tmp1=tmp1+1;

tintin=tintin+20;
end

N = 'NORTHING';
NS = 'NS';
EW = 'EW';
E = 'EASTING';
vol = 'VOLUME (in meter)';
height = 'HEIGHT (in meter)';
cat = 'CATEGORY';
ar = 'AREA';

fileleft = sprintf('%s_left.xls', inputfile);
fileright = sprintf('%s_right.xls', inputfile);
denleft = sprintf('%s_left_Density.xls', inputfile);
denright = sprintf('%s_right_Density.xls', inputfile);

fid = fopen(fileleft, 'w');
fprintf(fid, '%s%s%s%s%s%s
', N, NS, E, EW, vol, height, cat);
fclose(fid);
fid = fopen(fileright,'w');
fprintf(fid, '%s\t%s\t%s\t%s\n', N, NS, E, EW, vol, height, cat);
fclose(fid);

fid = fopen(denleft,'w');
fprintf(fid, '%s\t%s\t%s\t%s\t%s\t%s\t%s\n', N, NS, E, EW, 'TOP_DENSITY', 'ZONE', 'MID_DENSITY', 'ZONE', 'LOW_DENSITY', 'ZONE');
fclose(fid);

fid = fopen(denright,'w');
fprintf(fid, '%s\t%s\t%s\t%s\t%s\t%s\t%s\n', N, NS, E, EW, 'TOP_DENSITY', 'ZONE', 'MID_DENSITY', 'ZONE', 'LOW_DENSITY', 'ZONE');
fclose(fid);

myformat= '%1.10e\t%s\t%1.10e\t%s\t%s\t%s\t%s\n';
denformat= '%1.10e\t%s\t%1.10e\t%s\t%d\t%d\t%d\t%d\t%d\n';

for i = 1:g

fid = fopen(fileleft,'a');
fprintf(fid, myformat, cellarrayleft{i,1}, cellarrayleft{i,2}, cellarrayleft{i,3}, cellarrayleft{i,4}, cellarrayleft{i,5}, cellarrayleft{i,6}, cellarrayleft{i,7});
fclose(fid);

fid = fopen(fileright,'a');
fprintf(fid, myformat, cellarrayright{i,1}, cellarrayright{i,2}, cellarrayright{i,3}, cellarrayright{i,4}, cellarrayright{i,5}, cellarrayright{i,6}, cellarrayright{i,7});
fclose(fid);

if((topdenleft{jh})<1000) & ((topdenleft{jh})~=0)
    zonetopleft{jh}=1;
elseif(((topdenleft{jh})>1000) & ((topdenleft{jh})<2000))
    zonetopleft{jh}=2;
elseif(((topdenleft{jh})>2000) & ((topdenleft{jh})<3000))
    zonetopleft{jh}=3;
else
    zonetopleft{jh}=0;
end

if((topdenright{jh})<1000) & ((topdenright{jh})~=0)
    zonetopright{jh}=1;
elseif(((topdenright{jh})>1000) & ((topdenright{jh})<2000))
    zonetopright{jh}=2;
elseif(((topdenright{jh})>2000) & ((topdenright{jh})<3000))
    zonetopright{jh}=3;
else
    zonetopright{jh}=0;
end
\[
\text{else}
\]
\[
zonetopright\{jh\} = 0;
\]
\end
\]
\[
\text{if} (((\text{middenleft}\{jh\}) < 1000) \& \& ((\text{middenleft}\{jh\}) \neq 0))
\]
\[
zonemidleft\{jh\} = 1;
\]
\[
\text{elseif} (((\text{middenleft}\{jh\}) > 1000) \& \& (\text{middenleft}\{jh\} < 2000))
\]
\[
zonemidleft\{jh\} = 2;
\]
\[
\text{elseif} (((\text{middenleft}\{jh\}) > 2000) \& \& (\text{middenleft}\{jh\} < 3000))
\]
\[
zonemidleft\{jh\} = 3;
\]
\[
\text{else}
\]
\[
zonemidleft\{jh\} = 0;
\]
\end
\]
\[
\text{if} (((\text{middenright}\{jh\}) < 1000) \& \& ((\text{middenright}\{jh\}) \neq 0))
\]
\[
zonemidright\{jh\} = 1;
\]
\[
\text{elseif} (((\text{middenright}\{jh\}) > 1000) \& \& (\text{middenright}\{jh\} < 2000))
\]
\[
zonemidright\{jh\} = 2;
\]
\[
\text{elseif} (((\text{middenright}\{jh\}) > 2000) \& \& (\text{middenright}\{jh\} < 3000))
\]
\[
zonemidright\{jh\} = 3;
\]
\[
\text{else}
\]
\[
zonemidright\{jh\} = 0;
\]
\end
\]
\[
\text{if} (((\text{botdenleft}\{jh\}) < 1000) \& \& ((\text{botdenleft}\{jh\}) \neq 0))
\]
\[
zonelowleft\{jh\} = 1;
\]
\[
\text{elseif} (((\text{botdenleft}\{jh\}) > 1000) \& \& (\text{botdenleft}\{jh\} < 2000))
\]
\[
zonelowleft\{jh\} = 2;
\]
\[
\text{elseif} (((\text{botdenleft}\{jh\}) > 2000) \& \& (\text{botdenleft}\{jh\} < 3000))
\]
\[
zonelowleft\{jh\} = 3;
\]
\[
\text{else}
\]
\[
zonelowleft\{jh\} = 0;
\]
\end
\]
\[
\text{if} (((\text{botdenright}\{jh\}) < 1000) \& \& ((\text{botdenright}\{jh\}) \neq 0))
\]
\[
zonelowright\{jh\} = 1;
\]
\[
\text{elseif} (((\text{botdenright}\{jh\}) > 1000) \& \& (\text{botdenright}\{jh\} < 2000))
\]
\[
zonelowright\{jh\} = 2;
\]
\[
\text{elseif} (((\text{botdenright}\{jh\}) > 2000) \& \& (\text{botdenright}\{jh\} < 3000))
\]
\[
zonelowright\{jh\} = 3;
\]
\[
\text{else}
\]
\[
zonelowright\{jh\} = 0;
\]
\end
\]
\[
\text{if} (\text{cellarrayleft(i,6)**='0'})
\]
\[
\text{outtopdenleft}\{jh\} = 0; \text{outzonetopleft}\{jh\} = 0;
\]
\[
\text{outmiddenleft}\{jh\} = 0; \text{outzonemidleft}\{jh\} = 0;
\]
\[
\text{outbotdenleft}\{jh\} = 0; \text{outzonelowleft}\{jh\} = 0;
\]
\[
\text{else}
\]
\[
\text{outtopdenleft}\{jh\} = \text{topdenleft}\{jh\};
\]
\[
\text{outzonetopleft}\{jh\} = \text{zonetopleft}\{jh\};
\]
\[
\text{outmiddenleft}\{jh\} = \text{middenleft}\{jh\};
\]
\[
\text{outzonemidleft}\{jh\} = \text{zonemidleft}\{jh\};
\]
\[
\text{outbotdenleft}\{jh\} = \text{botdenleft}\{jh\};
\]
\[
\text{outzonelowleft}\{jh\} = \text{zonelowleft}\{jh\};
\]
\end
\]
if (cellarrayright{i,6} == '0')

    outtopdenright{jh}=0;
    outzonetopright{jh}=0;
    outmiddenright{jh}=0;
    outzonemidright{jh}=0;
    outbotdenright{jh}=0;
    outzonelowright{jh}=0;

else

    outtopdenright{jh}=topdenright{jh};
    outzonetopright{jh}=zonetopright{jh};
    outmiddenright{jh}=middenright{jh};
    outzonemidright{jh}=zonemidright{jh};
    outbotdenright{jh}=botdenright{jh};
    outzonelowright{jh}=zonelowright{jh};

end

if (isnan(outtopdenleft{jh}) | outtopdenleft{jh} == '')
    outtopdenleft{jh}=0 ;
end

if (isnan(outmiddenleft{jh}) | outmiddenleft{jh} == '')
    outmiddenleft{jh}=0 ;
end

if (isnan(outbotdenleft{jh}) | outbotdenleft{jh} == '')
    outbotdenleft{jh}=0 ;
end

if (isnan(outtopdenright{jh}) | outtopdenright{jh} == '')
    outtopdenright{jh}=0 ;
end

if (isnan(outmiddenright{jh}) | outmiddenright{jh} == '')
    outmiddenright{jh}=0 ;
end

if (isnan(outbotdenright{jh}) | outbotdenright{jh} == '')
    outbotdenright{jh}=0 ;
end

fid = fopen(denleft, 'a');
fprintf(fid, denformat,
    cellarrayleft{i,1},cellarrayleft{i,2},cellarrayleft{i,3},cellarrayleft{i,4},out
topdenleft{jh}, outzonetopleft{jh},outmiddenleft{jh},
outzonemidleft{jh},outbotdenleft{jh}, outzonelowleft{jh});
close(fid);

fid = fopen(denright, 'a');
fprintf(fid, denformat,
    cellarrayright{i,1},cellarrayright{i,2},cellarrayright{i,3},cellarrayright{i,4}
, outtopdenright{jh}, outzonetopright{jh},outmiddenright{jh},
outzonemidright{jh},outbotdenright{jh}, outzonelowright{jh});
close(fid);

if (mod(i,6) == 0)
    jh=jh+1;
end
APPENDIX B
JAVA CODE FOR HEIGHT, VOLUME, AND DENSITY CALCULATION

package Canopychar;
import java.io.*;
import java.util.*;

public class Canopychar {

    public static int numlines=0;/ number of lines in the csv file
    public static int numscans;/ number of scans
    public static String[][] completefiledata= new String[25000][190];// complete data of the
    file in an array
    public static String[][] data= new String[25000][190];// complete data of the file without
    the headings in an array
    public static double latdivide, ld1,ld gd1,gd,ld1gga, ldgga, gd1gga, gdgga;
    public static double longdivide;
    public static double latdividegga;
    public static double longdividegga;
    public static double[][][] leftdata = new double[4000][12][181];
    public static double[][][] rightreading= new double[4000][12][181];
    public static double[][][] leftreading= new double[4000][12][181];
    public static double[][][] rightdata= new double[4000][12][181];
    public static double[][][] cellarrayleft = new double[15000][8];
    public static double[][][] cellarrayright = new double[15000][8];
    public static double[][][] sdlowleft = new double[15000][12];
    public static double[][][] sdlowright = new double[15000][12];
    public static double[][][] sdtopleft = new double[15000][12];
    public static double[][][] sdtopright = new double[15000][12];
    public static double[][][] sdmidleft = new double[15000][12];
    public static double[][][] sdmidright = new double[15000][12];
    public static double distance=8000;
    public static double TD=distance/2;
    public static int SH = 500, gf;// laser height above the ground
    public static double[] xleft= new double[4000][12][181];
    public static double[] yleft= new double[4000][12][181];
    public static double[] xright= new double[4000][12][181];
    public static double[] yright= new double[4000][12][181];
    public static String GPSstring;
    public static double degreelat,temp1,deglat,temp2,deglong,deglat1,deglong1,deglatdiv,deglongdiv,degreelong, mean;
    public static double[] hleft= new double [4000][12];
    public static double[] hright= new double [4000][12];
    public static double[] Arealeft= new double [4000][12];

degreelat,temp1,deglat,temp2,deglong,deglat1,deglong1,deglatdiv,deglongdiv,degreelong, mean,
public static double[][] Arearight= new double [4000][12];
public static double[] khrigh= new double [12];
public static double[] khleft= new double [12];
public static ArrayList<Double> calcarealeft= new ArrayList<Double>();
public static ArrayList<Double> calcarearight= new ArrayList<Double>();
public static double[] veldata=new double[4000];
public static double veldataknots, sum, sd, sumsd;
public static double lhmax, rhmax, Areamedleft, Areamedright;
public static int t=0, k1;
public static long l=6000000000000L;
public static int g;
public static String N = "LATITUDE";
public static String NS= "NS";
public static String EW= "EW";
public static String E = "LONGITUDE";
public static String vol= "VOLUME (in cubic meter)";
public static String height= "HEIGHT (in meter)";
public static String cat= "CATEGORY";
public static String outputfileleft,outputfileright,densityfileleft, densityfileright;
public static String dlat, dlong;

//==========density variables==============
public static int denvar=0;
public static int divscan;
public static int jtemp;
public static double divdeg=0;
public static int ktemp;
public static int fin=102;

public static ArrayList<Double> heightright= new ArrayList<Double>();
public static ArrayList<Double> heightleft= new ArrayList<Double>();
public static ArrayList<Double> arrayhtopright= new ArrayList<Double>();
public static ArrayList<Double> arrayhmidright= new ArrayList<Double>();
public static ArrayList<Double> arrayhlowright= new ArrayList<Double>();
public static ArrayList<Double> arrayhtopleft= new ArrayList<Double>();
public static ArrayList<Double> arrayhmidleft= new ArrayList<Double>();
public static ArrayList<Double> arrayhlowleft= new ArrayList<Double>();
public static ArrayList<Double> Aleft= new ArrayList<Double>();
public static ArrayList<Double> Dleft= new ArrayList<Double>();
public static double[] lineavgleft = new double [5];
public static ArrayList<Double> a1left= new ArrayList<Double>();
public static ArrayList<Double> a2left= new ArrayList<Double>();
public static ArrayList<Double> a3left= new ArrayList<Double>();
public static ArrayList<Double> a4left= new ArrayList<Double>();
public static ArrayList<Double> Aleft= new ArrayList<Double>();
public static ArrayList<Double> Dleft= new ArrayList<Double>();
public static ArrayList<Double> a1right= new ArrayList<Double>();
public static ArrayList<Double> a2right= new ArrayList<Double>();
public static ArrayList<Double> a3right = new ArrayList<Double>();
public static ArrayList<Double> a4right = new ArrayList<Double>();
public static ArrayList<Double> Aright = new ArrayList<Double>();
public static ArrayList<Double> DR = new ArrayList<Double>();
public static ArrayList<Double> DL = new ArrayList<Double>();
public static double[] lineavgright = new double[5];
public static double[] sectionavgleft = new double[300000];
public static double[] sectionavgright = new double[300000];
public static double[] denleft = new double[5000];
public static double[] denright = new double[5000];
public static double[] topdenleft = new double[5000];
public static double[] topdenright = new double[5000];
public static double[] middenleft = new double[5000];
public static double[] middenright = new double[5000];
public static double[] botdenright = new double[5000];
public static double[] zonetopleft = new double[5000];
public static double[] zonetopright = new double[5000];
public static double[] zonemidleft = new double[5000];
public static double[] zonemidright = new double[5000];
public static double[] zonelowleft = new double[5000];
public static double[] zonelowright = new double[5000];
public static double[] botdenleft = new double[5000];
public static double[] outtopdenleft = new double[5000];
public static double[] outtopdenright = new double[5000];
public static double[] outmiddenright = new double[5000];
public static double[] outmiddenleft = new double[5000];
public static double[] outbotdenright = new double[5000];
public static double[] outzonetopleft = new double[5000];
public static double[] outzonetopright = new double[5000];
public static double[] outzonemidleft = new double[5000];
public static double[] outzonemidright = new double[5000];
public static double[] outzonelowleft = new double[5000];
public static double[] outzonelowright = new double[5000];
public static double[] tophleft = new double[5000];
public static double[] tophright = new double[5000];
public static double[] midhright = new double[5000];
public static double[] midhleft = new double[5000];
public static double[] bothleft = new double[5000];
public static int count = 0;
public static int tmp, hitheightdivleft, hitheightdivright;
public static void main(String[] args) throws FileNotFoundException, IOException {
    Scanner in = new Scanner(System.in);
    System.out.print("Enter input filename filename with extension: ");
    String filename = in.nextLine();
    File inputfile = new File(filename);
    InputStream is = new BufferedInputStream(new FileInputStream(inputfile));
    System.gc();
    BufferedReader br = new BufferedReader(new FileReader(inputfile));
    String line = null;
    int row = 0;
    int col = 182;
    int colcnt;
    while((line = br.readLine()) != null)
    {
        colcnt = 0;
        StringTokenizer st = new StringTokenizer(line, ",");
        while (st.hasMoreTokens())
        {
            //get next token and store it in the array
            completefiledata[row][colcnt] = st.nextToken();
            colcnt++;
        }
        row++;
    }
    numlines = row - 9;
    numscans = numlines / 26;

    //close the file
    br.close();
    int m = 0;
    int n = 0;
    for (int i = 9; i < row; i++)
    {
        n = 0;
        for (int j = 2; j <= col; j++)
        {
            data[m][n] = completefiledata[i][j];
        }
    }
n=n+1;
} m=m+1;

int ldataindex=1; int rdataindex=13;
for(int i=0; i<numscans; i++)
{
    for(int j=0; j<12; j++)
    {
        ldataindex=ldataindex+1;
        rdataindex=rdataindex+1;
        for(int k=0; k<181; k++)
        {
            leftdata[i][j][k]= Double.parseDouble(data[ldataindex][k]);
            rightreading[i][j][k]= Integer.parseInt(data[rdataindex][k]);
        }
        if(j ==11)
        {
            ldataindex= ldataindex+14;
            rdataindex=rdataindex+14;
        }
    }
    for(int i=0; i<numscans; i++)
{ 
  for(int j=0;j<12;j++)
  {
    m=180;

    for(int k=0; k<181; k++)
    {
      rightdata[i][j][k] = rightreading[i][j][m];
      m=m-1;
    }
  }
}

for (int i=0; i<numscans;i++)
{
  for (int j=0;j<12;j++)
  {
    for (int k=0;k<181;k++)
    {
      if(leftdata[i][j][k] > TD)
      {
        leftdata[i][j][k]=0;
      }

      if(rightdata[i][j][k] > TD)
      {
        rightdata[i][j][k]=0;
      }
    }
  }
}
for (int i=0; i<numscans;i++)
{
    for (int j=0;j<12;j++)
    {
        for (int k=0;k<10;k++)
        {
            leftdata[i][j][k]=0;
            rightdata[i][j][k]=0;
        }
        for (int k=175;k<181;k++)
        {
            leftdata[i][j][k]=0;
            rightdata[i][j][k]=0;
        }
    }
}

for (int i=0; i<numscans;i++)
{
    for (int j=0;j<12;j++)
    {
        lhmax=0;
        rhmax=0;
        for (int k=0;k<181;k++)
        {
            if(leftdata[i][j][k]==0)
            {
                xleft[i][j][k]= 0;
                yleft[i][j][k]=0;
            }
            else
            {
                xleft[i][j][k]= Math.abs(leftdata[i][j][k])*Math.cos(Math.toRadians(k));
                yleft[i][j][k]= Math.abs(leftdata[i][j][k])*Math.sin(Math.toRadians(k));
            }
            if(rightdata[i][j][k]==0)
            {
                xright[i][j][k]= 0; yright[i][j][k]=0;
            }
        }
    }
}
else {
    xright[i][j][k] = Math.abs(rightdata[i][j][k]*Math.cos(Math.toRadians(k)));
    yright[i][j][k] = Math.abs(rightdata[i][j][k]*Math.sin(Math.toRadians(k)));
}

Arrays.sort(yleft[i][j]);
Arrays.sort(yright[i][j]);

hleft[i][j] = yleft[i][j][180];
hright[i][j] = yright[i][j][180];

}
}

int f=9;
int d=0;

for (int i=0;i<numscans;i++)
{
    GPSstring = completefiledata[f][1];
    f=f+26;
    //============================================GGA==============
    if(GPSstring.equals("$GPGGA"))
    {
        veldata[i]=178458.720224065;
        degreelat= Double.parseDouble(data[d][1]);
        latdividegga= Double.parseDouble(data[0][1]);
        longdividegga= Double.parseDouble(data[0][3]);
        ld1gga = (Math.floor(latdividegga/100)*100);
        ldgga = (Math.floor(latdividegga/100));
gd1gga = (Math.floor(longdividegga/100)*100);
gdgga = (Math.floor(longdividegga/100));

if(i<numscans-1)
{
    temp1 = Double.parseDouble(data[d+26][1]);
}
deglat = -((degrelat - ld1gga)/60)+ldgga;
deglat1 = -((temp1 - ld1gga)/60)+ldgga;

degreelong = Double.parseDouble(data[d][3]);
if(i<numscans-1)
{
    temp2 = Double.parseDouble(data[d+26][3]);
}
degl = ((degrelong - gd1gga)/60)+gdgga;
degl1 = ((temp2 - gd1gga)/60)+gdgga;

d = d+26;
}

//=====================================================================================================
else if(GPSstring.equals("$GPRMC")||GPSstring.equals("$GNRMC"))
{
    degrelat = Double.parseDouble(data[d][2]);
    veldatalknots = Double.parseDouble(data[d][7]);
    veldat[i] = veldatalknots/0.00194384449;
    latdivide = Double.parseDouble(data[0][2]);
    longdivide = Double.parseDouble(data[0][4]);
    ld1 = (Math.floor(latdivide/100)*100);
    ld = (Math.floor(latdivide/100));
    gd1 = (Math.floor(longdivide/100)*100);
    gd = (Math.floor(longdivide/100));

    if(i<numscans-1)
\[
\begin{align*}
\text{temp1} &= \text{Double.parseDouble(data[d+26][2])}; \\
\text{deglat} &= -((\text{degreelat} - \text{ld1})/60)+\text{ld}; \\
\text{deglat1} &= -(((\text{temp1}-\text{ld1})/60)+\text{ld}); \\
\text{degreelong} &= \text{Double.parseDouble(data[d][4])}; \\
\text{if}(i<\text{numscans}-1) \\
\{ \\
\quad \text{temp2} &= \text{Double.parseDouble(data[d+26][4])}; \\
\} \\
\text{deglong} &= ((\text{degreelong} - \text{gd1})/60)+\text{gd}; \\
\text{deglong1} &= ((\text{temp2}-\text{gd1})/60)+\text{gd}); \\
\text{d} &= d+26; \\
\}
\]
{  
  getit=0;
}
for(int ak=0;ak< 180;ak++)
{
  if (ak==179)
  {
    if(xleft[i][j][179]<xleft[i][j][178])
    {
      Arealeft[i][j]= (Arealeft[i][j]+(Math.abs(((TD-xleft[i][j][178])*(yleft[i][j][0]))- ((TD-xleft[i][j][0])*(yleft[i][j][178]))))/2;
    }
    else
    {
      Arealeft[i][j]= (Arealeft[i][j]+(Math.abs(((TD-xleft[i][j][179])*(yleft[i][j][0]))- ((TD-xleft[i][j][0])*(yleft[i][j][179]))))/2;
    }
  }
  if(xright[i][j][179]<xright[i][j][178])
  {
    Arearight[i][j]= (Arearight[i][j]+(Math.abs(((TD-xright[i][j][178])*(yright[i][j][0]))- ((TD-xright[i][j][0])*(yright[i][j][178]))))/2;
  }
  else
  {
    Arearight[i][j]= (Arearight[i][j]+(Math.abs(((TD-xright[i][j][179])*(yright[i][j][0]))- ((TD-xright[i][j][0])*(yright[i][j][179]))))/2;
  }
}
else
{
  gf= ak-1;
  if (ak==0) gf=0;
  if(xleft[i][j][ak]<xleft[i][j][gf])
  {
    Arealeft[i][j]= Arealeft[i][j]+ (Math.abs( ((TD-xleft[i][j][gf])*(yleft[i][j][ak+1]))- ((TD-xleft[i][j][ak+1])*(yleft[i][j][gf]))));
  }
  else
  {

Area_{left[i][j]} = Area_{left[i][j]} + \text{Math.abs}\left( (TD - x_{left[i][j][ak]})(y_{left[i][j][ak + 1]]) - ((TD - x_{left[i][j][ak + 1]})(y_{left[i][j][ak]})) \right);

\}
if((x_{right[i][j][ak]} < x_{right[i][j][gf]}) & (ak != 0))
{

Area_{right[i][j]} = Area_{right[i][j]} + \text{Math.abs}\left( (TD - x_{right[i][j][gf]})(y_{right[i][j][ak + 1]]) - ((TD - x_{right[i][j][ak + 1]})(y_{right[i][j][gf]})) \right);

\}
else
{

Area_{right[i][j]} = Area_{right[i][j]} + \text{Math.abs}\left( (TD - x_{right[i][j][ak]})(y_{right[i][j][ak + 1]]) - ((TD - x_{right[i][j][ak + 1]})(y_{right[i][j][ak]})) \right);

\}

}

g = g + 1;
}

//=================================Getting the Volume and Height data in cellarray=======================================

if((h_{left[i][j]} - SH) > 1923.5)// shoot height=600
{

cellarray_{left[i][j][5]} = (h_{left[i][j]} + SH) / 1000;
cellarray_{left[i][j][4]} = 2 * Area_{left[i][j]} * \text{veldata[i]} * dt / l;

}
else
{

cellarray_{left[i][j][5]} = 0;
cellarray_{left[i][j][4]} = 0;
}

if((h_{right[i][j]} - SH) > 1923.5)
{

cellarray_{right[i][j][5]} = (h_{right[i][j]} + SH) / 1000;
cellarray_{right[i][j][4]} = 2 * Area_{right[i][j]} * \text{veldata[i]} * dt / l;

}
} 
else 
{
    cellarrayright[t][5]= 0;
    cellarrayright[t][4]= 0;
}

if(j==0) 
{
    cellarrayleft[t][0]=Math.abs(deglat);
    cellarrayleft[t][2]=Math.abs(deglong);
    cellarrayright[t][0]=Math.abs(deglat);
    cellarrayright[t][2]=Math.abs(deglong);
}
else 
{
    cellarrayleft[t][0]= cellarrayleft[t-1][0]+deglatdiv;
    cellarrayright[t][0]= cellarrayright[t-1][0]+deglatdiv;
    cellarrayleft[t][2]= cellarrayleft[t-1][2]+deglongdiv;
    cellarrayright[t][2]= cellarrayright[t-1][2]+deglongdiv;
}

if((cellarrayleft[t][4]==0)||(cellarrayleft[t][5]==0)) 
{
    cellarrayleft[t][5]=0;cellarrayleft[t][4]=0;
}
if((cellarrayright[t][4]==0)||(cellarrayright[t][5]==0)) 
{
    cellarrayright[t][5]=0;cellarrayright[t][4]=0;
}

if(GPSstring.equals("$GPGGA"))
{
    dlat= data[0][2];
    dlong=data[0][4];
}
else if(GPSstring.equals("$GPRMC")||GPSstring.equals("$GNRMC")) 
{
    dlat= data[0][3];
    dlong=data[0][5];
}
if (dlat.equals("N"))
{
    cellarrayleft[t][0] = Math.abs(cellarrayleft[t][0]);
    cellarrayright[t][0] = Math.abs(cellarrayright[t][0]);
}
else
{
    cellarrayleft[t][0] = - Math.abs(cellarrayleft[t][0]);
    cellarrayright[t][0] = - Math.abs(cellarrayright[t][0]);
}
if (dlon.equals("W"))
{
    cellarrayleft[t][2] = - Math.abs(cellarrayleft[t][2]);
    cellarrayright[t][2] = - Math.abs(cellarrayright[t][2]);
}
else
{
    cellarrayleft[t][2] = Math.abs(cellarrayleft[t][2]);
    cellarrayright[t][2] = Math.abs(cellarrayright[t][2]);
}

    t = t + 1;

}

// ===============================================Density
Calculation===============================================

for (int itemp = 0; itemp < numscans; itemp++)
{
    for (divscan = 0; divscan < 3; divscan++)
    {
        ktemp = 0;
        if (divscan == 0) jtemp = 0;
        else if (divscan == 1) jtemp = 4;
        else if (divscan == 2) jtemp = 8;

        for (divdeg = 0; divdeg < 20; divdeg++)
        {

        }  
}
a1left.add(xleft[itemp][jtemp][ktemp]); a1left.add(xleft[itemp][jtemp][ktemp+1]);
a1left.add(xleft[itemp][jtemp][ktemp+2]); a1left.add(xleft[itemp][jtemp][ktemp+3]);
a1left.add(xleft[itemp][jtemp][ktemp+4]); a1left.add(xleft[itemp][jtemp][ktemp+5]);
a1left.add(xleft[itemp][jtemp][ktemp+6]); a1left.add(xleft[itemp][jtemp][ktemp+7]);
a1left.add(xleft[itemp][jtemp][ktemp+8]);

a1right.add(xright[itemp][jtemp][ktemp]); a1right.add(xright[itemp][jtemp][ktemp+1]);
a1right.add(xright[itemp][jtemp][ktemp+2]); a1right.add(xright[itemp][jtemp][ktemp+3]);
a1right.add(xright[itemp][jtemp][ktemp+4]); a1right.add(xright[itemp][jtemp][ktemp+5]);
a1right.add(xright[itemp][jtemp][ktemp+6]); a1right.add(xright[itemp][jtemp][ktemp+7]);
a1right.add(xright[itemp][jtemp][ktemp+8]);

jtemp=jtemp+1;
lineavgleft[0]= mean(a1left);
lineavgright[0]= mean(a1right);

a2left.add(xleft[itemp][jtemp][ktemp]); a2left.add(xleft[itemp][jtemp][ktemp+1]);
a2left.add(xleft[itemp][jtemp][ktemp+2]); a2left.add(xleft[itemp][jtemp][ktemp+3]);
a2left.add(xleft[itemp][jtemp][ktemp+4]); a2left.add(xleft[itemp][jtemp][ktemp+5]);
a2left.add(xleft[itemp][jtemp][ktemp+6]); a2left.add(xleft[itemp][jtemp][ktemp+7]);
a2left.add(xleft[itemp][jtemp][ktemp+8]);

a2right.add(xright[itemp][jtemp][ktemp]); a2right.add(xright[itemp][jtemp][ktemp+1]);
a2right.add(xright[itemp][jtemp][ktemp+2]); a2right.add(xright[itemp][jtemp][ktemp+3]);
a2right.add(xright[itemp][jtemp][ktemp+4]); a2right.add(xright[itemp][jtemp][ktemp+5]);
a2right.add(xright[itemp][jtemp][ktemp+6]); a2right.add(xright[itemp][jtemp][ktemp+7]);
a2right.add(xright[itemp][jtemp][ktemp+8]);

jtemp=jtemp+1;
lineavgleft[1] = mean(a2left);
lineavgright[1] = mean(a2right);

a3left.add(xleft[itemp][jtemp][ktemp]); a3left.add(xleft[itemp][jtemp][ktemp][ktemp+1]);
a3left.add(xleft[itemp][jtemp][ktemp][ktemp+2]); a3left.add(xleft[itemp][jtemp][ktemp][ktemp+3]);
a3left.add(xleft[itemp][jtemp][ktemp][ktemp+4]); a3left.add(xleft[itemp][jtemp][ktemp][ktemp+5]);
a3left.add(xleft[itemp][jtemp][ktemp][ktemp+6]); a3left.add(xleft[itemp][jtemp][ktemp][ktemp+7]);
a3left.add(xleft[itemp][jtemp][ktemp][ktemp+8]);
a3right.add(xright[itemp][jtemp][ktemp]); a3right.add(xright[itemp][jtemp][ktemp][ktemp+1]);
a3right.add(xright[itemp][jtemp][ktemp][ktemp+2]); a3right.add(xright[itemp][jtemp][ktemp][ktemp+3]);
a3right.add(xright[itemp][jtemp][ktemp][ktemp+4]); a3right.add(xright[itemp][jtemp][ktemp][ktemp+5]);
a3right.add(xright[itemp][jtemp][ktemp][ktemp+6]); a3right.add(xright[itemp][jtemp][ktemp][ktemp+7]);
a3right.add(xright[itemp][jtemp][ktemp][ktemp+8]);

jtemp = jtemp + 1;
lineavgleft[2] = mean(a3left);
lineavgright[2] = mean(a3right);

a4left.add(xleft[itemp][jtemp][ktemp]); a4left.add(xleft[itemp][jtemp][ktemp][ktemp+1]);
a4left.add(xleft[itemp][jtemp][ktemp][ktemp+2]); a4left.add(xleft[itemp][jtemp][ktemp][ktemp+3]);
a4left.add(xleft[itemp][jtemp][ktemp][ktemp+4]); a4left.add(xleft[itemp][jtemp][ktemp][ktemp+5]);
a4left.add(xleft[itemp][jtemp][ktemp][ktemp+6]); a4left.add(xleft[itemp][jtemp][ktemp][ktemp+7]);
a4left.add(xleft[itemp][jtemp][ktemp][ktemp+8]);
a4right.add(xright[itemp][jtemp][ktemp]); a4right.add(xright[itemp][jtemp][ktemp][ktemp+1]);
a4right.add(xright[itemp][jtemp][ktemp][ktemp+2]); a4right.add(xright[itemp][jtemp][ktemp][ktemp+3]);
a4right.add(xright[itemp][jtemp][ktemp][ktemp+4]); a4right.add(xright[itemp][jtemp][ktemp][ktemp+5]);
a4right.add(xright[itemp][jtemp][ktemp+6]); a4right.add(xright[itemp][jtemp][ktemp+7]);
a4right.add(xright[itemp][jtemp][ktemp+8]);

jtemp=jtemp+1;
lineavgleft[3]= mean(a4left);
lineavgright[3]= mean(a4right);

if(divscan==0) jtemp=0;
else if(divscan==1) jtemp=4;
else if(divscan==2) jtemp=8;

Aleft.add(lineavgleft[0]); Aleft.add(lineavgleft[1]); Aleft.add(lineavgleft[2]); Aleft.add(lineavgleft[3]);
sectionavgleft[count]= mean(Aleft);

Aright.add(lineavgright[0]); Aright.add(lineavgright[1]); Aright.add(lineavgright[2]); Aright.add(lineavgright[3]);
sectionavgright[count]= mean(Aright);

count=count+1;
denvar=denvar+1;
a1left.clear(); a2left.clear(); a3left.clear(); a4left.clear(); Aleft.clear();
a1right.clear(); a2right.clear(); a3right.clear(); a4right.clear(); Aright.clear();

ktemp=ktemp+9;
} }
jtemp=0;

}

int tintin=0;
tmp=0;
int tmp1=0;
while( tintin != count) {

Dleft.add(sectionavgleft[tintin]); Dleft.add(sectionavgleft[tintin+1]); Dleft.add(sectionavgleft[tintin+2]); Dleft.add(sectionavgleft[tintin+3]);

Dleft.add(sectionavgleft[tintin+4]); Dleft.add(sectionavgleft[tintin+5]); Dleft.add(sectionavgleft[tintin+6]); Dleft.add(sectionavgleft[tintin+7]);
Dleft.add(sectionavgleft[tintin+8]); Dleft.add(sectionavgleft[tintin+9]); Dleft.add(sectionavgleft[tintin+10]); Dleft.add(sectionavgleft[tintin+11]);

Dleft.add(sectionavgleft[tintin+12]); Dleft.add(sectionavgleft[tintin+13]); Dleft.add(sectionavgleft[tintin+14]); Dleft.add(sectionavgleft[tintin+15]);

Dleft.add(sectionavgleft[tintin+16]); Dleft.add(sectionavgleft[tintin+17]); Dleft.add(sectionavgleft[tintin+18]); Dleft.add(sectionavgleft[tintin+19]);

    DL.add(Dleft.get(0)); DL.add(Dleft.get(1)); DL.add(Dleft.get(2));
    DL.add(Dleft.get(3)); DL.add(Dleft.get(4)); DL.add(Dleft.get(5));
    topdenleft[tmp1]= mean(DL);
    DL.clear();
    DL.add(Dleft.get(6)); DL.add(Dleft.get(7)); DL.add(Dleft.get(8));
    DL.add(Dleft.get(9)); DL.add(Dleft.get(10)); DL.add(Dleft.get(11));
    middenleft[tmp1]= mean(DL);
    DL.clear();
    DL.add(Dleft.get(12)); DL.add(Dleft.get(13)); DL.add(Dleft.get(14));
    DL.add(Dleft.get(15)); DL.add(Dleft.get(16)); DL.add(Dleft.get(17));
    DL.add(Dleft.get(18)); DL.add(Dleft.get(19));
    botdenleft[tmp1]= mean(DL);
    DL.clear();

denleft[tmp]= mean(Dleft);
Dleft.clear();

Dright.add(sectionavgright[tintin]); Dright.add(sectionavgright[tintin+1]); Dright.add(sectionavgright[tintin+2]); Dright.add(sectionavgright[tintin+3]);

Dright.add(sectionavgright[tintin+4]); Dright.add(sectionavgright[tintin+5]); Dright.add(sectionavgright[tintin+6]); Dright.add(sectionavgright[tintin+7]);

Dright.add(sectionavgright[tintin+8]); Dright.add(sectionavgright[tintin+9]); Dright.add(sectionavgright[tintin+10]); Dright.add(sectionavgright[tintin+11]);

Dright.add(sectionavgright[tintin+12]); Dright.add(sectionavgright[tintin+13]); Dright.add(sectionavgright[tintin+14]); Dright.add(sectionavgright[tintin+15]);

Dright.add(sectionavgright[tintin+16]); Dright.add(sectionavgright[tintin+17]); Dright.add(sectionavgright[tintin+18]); Dright.add(sectionavgright[tintin+19]);

    DR.add(Dright.get(0)); DR.add(Dright.get(1)); DR.add(Dright.get(2));
    DR.add(Dright.get(3)); DR.add(Dright.get(4)); DR.add(Dright.get(5));
topdenright[tmp1] = mean(DR);
DR.clear();
DR.add(Dright.get(6)); DR.add(Dright.get(7)); DR.add(Dright.get(8));
DR.add(Dright.get(9)); DR.add(Dright.get(10)); DR.add(Dright.get(11));
middlenright[tmp1] = mean(DR);
DR.clear();
DR.add(Dright.get(12)); DR.add(Dright.get(13)); DR.add(Dright.get(14));
DR.add(Dright.get(15)); DR.add(Dright.get(16)); DR.add(Dright.get(17));
DR.add(Dright.get(18)); DR.add(Dright.get(19));
botdenright[tmp1] = mean(DR);

DR.clear();
tmp1 = tmp1 + 1;
denright[tmp] = mean(Dright);
Dright.clear();
tmp = tmp + 1;
tintin = tintin + 20;
}

// Yield estimation manual calculation, get the denleft and denright in top, mid, bottom, 6 sections each for top mid bottom

//=--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
//=--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
g = numscans * 12;

for(int i=0; i<g; i++)
{
    if (cellarrayleft[i][5] < 0.5)
        cellarrayleft[i][6] = 0;
    else if (cellarrayleft[i][5] < 1.2)
        cellarrayleft[i][6] = 1;
    else if (cellarrayleft[i][5] < 2.13)
        cellarrayleft[i][6] = 2;
else if (cellarrayleft[i][5] < 2.74)
    cellarrayleft[i][6] = 3;

else if (cellarrayleft[i][5] < 3.35)
    cellarrayleft[i][6] = 4;

else if (cellarrayleft[i][5] > 3.35)
    cellarrayleft[i][6] = 5;

if (cellarrayright[i][5] < 0.5)
    cellarrayright[i][6] = 0;

else if (cellarrayright[i][5] < 1.2)
    cellarrayright[i][6] = 1;

else if (cellarrayright[i][5] < 2.13)
    cellarrayright[i][6] = 2;

else if (cellarrayright[i][5] < 2.74)
    cellarrayright[i][6] = 3;

else if (cellarrayright[i][5] < 3.35)
    cellarrayright[i][6] = 4;

else if (cellarrayright[i][5] > 3.35)
    cellarrayright[i][6] = 5;

}

outputfileleft = filename + "_left.txt";
outputfileright = filename + "_right.txt";
densityfileleft = filename + "_DENSITY_LEFT.txt";
densityfileright = filename + "_DENSITY_RIGHT.txt";

File fileleft = new File(outputfileleft);
File fileright = new File(outputfileright);
File dencalcleft = new File(densityfileleft);
File dencalcright = new File(densityfileright);

PrintWriter outleft = new PrintWriter(new FileWriter(fileleft));
PrintWriter outright = new PrintWriter(new FileWriter(fileright));
PrintWriter outdensityleft = new PrintWriter(new FileWriter(dencalcleft));
PrintWriter outdensityright = new PrintWriter(new FileWriter(dencalcright));

outleft.println(N + "t" + NS + "t" + E + "t" + EW + "t" + vol + "t" + height + "t" + cat + "n\n" + "n\n");

outright.println(N + "t" + NS + "t" + E + "t" + EW + "t" + vol + "t" + height + "t" + cat + "n\n" + "n\n");
for(int i=0;i<g;i++)
{
    outleft.print(cellarrayleft[i][0]+"\t"+dlat+"\t"+cellarrayleft[i][2]+"\t"+dlong+"\t"+cellarrayleft[i][4]+"\t"+cellarrayleft[i][5]+"\t"+cellarrayleft[i][6]+"\n");
    outright.print(cellarrayright[i][0]+"\t"+dlat+"\t"+cellarrayright[i][2]+"\t"+dlong+"\t"+cellarrayright[i][4]+"\t"+cellarrayright[i][5]+"\t"+cellarrayright[i][6]+"\n");
}

outright.close();
outleft.close();

outdensityleft.print(N+"\t"+ NS+ "\t" + E + "\t"+ EW + "\t"+
    "LEFT_TOP_DENSITY"+ "\t"+ "ZONE"+ "\t"+"LEFT_MID_DENSITY"+ "\t"+"ZONE"+ "\t"+
    "LEFT_LOW_DENSITY"+"\t"+"ZONE"+"\n\n");
outdensityright.print(N+"\t"+ NS+ "\t" + E + "\t"+ EW + "\t"
    +"RIGHT_TOP_DENSITY"+ "\t"+"ZONE"+ "\t"+"RIGHT_MID_DENSITY"+ "\t"+"ZONE"+
    "\t"+"RIGHT_LOW_DENSITY"+"\t"+"ZONE"+"\n\n");

int jh=0;
for(int i=0;i<g;i++)
{
    if(((topdenleft[jh]<1000)& (topdenleft[jh])!=0)) zonetopleft[jh]=1;
    else if(((topdenleft[jh]>1000) & (topdenleft[jh])<2000)) zonetopleft[jh]=2;
    else if(((topdenleft[jh])>2000) & ((topdenleft[jh])<3000)) zonetopleft[jh]=3;
    if(((topdenright[jh]<1000)& (topdenright[jh])!=0)) zonetopright[jh]=1;
    else if(((topdenright[jh])>1000) & ((topdenright[jh])<2000)) zonetopright[jh]=2;
    else if(((topdenright[jh])>2000) & ((topdenright[jh])<3000)) zonetopright[jh]=3;
    if(((middenleft[jh]<1000)& (middenleft[jh])!=0)) zonemidleft[jh]=1;
    else if(((middenleft[jh])>1000) & ((middenleft[jh])<2000)) zonemidleft[jh]=2;
    else if(((middenleft[jh])>2000) & ((middenleft[jh])<3000)) zonemidleft[jh]=3;
    if(((middenright[jh]<1000)& (middenright[jh])!=0)) zonemidright[jh]=1;
    else if(((middenright[jh])>1000) & ((middenright[jh])<2000)) zonemidright[jh]=2;
    else if(((middenright[jh])>2000) & ((middenright[jh])<3000)) zonemidright[jh]=3;
    if(((botdenleft[jh]<1000)& (botdenleft[jh])!=0)) zonelowleft[jh]=1;
    else if(((botdenleft[jh])>1000) & ((botdenleft[jh])<2000)) zonelowleft[jh]=2;
    else if(((botdenleft[jh])>2000) & ((botdenleft[jh])<3000)) zonelowleft[jh]=3;
    if(((botdenright[jh]<1000)& (botdenright[jh])!=0)) zonelowright[jh]=1;
    else if(((botdenright[jh])>1000) & ((botdenright[jh])<2000)) zonelowright[jh]=2;
    else if(((botdenright[jh])>2000) & ((botdenright[jh])<3000)) zonelowright[jh]=3;
}
else if(((botdenright[jh]>1000) & (botdenright[jh]<2000)))
zonelowright[jh]=2;
else if(((botdenright[jh]>2000) & (botdenright[jh]<3000)))
zonelowright[jh]=3;

if(cellarrayleft[i][4]==0)
{
    outtopdenleft[jh]=0; outzonetopleft[jh]=0;
    outmiddenleft[jh]=0; outzonemidleft[jh]=0;
    outbotdenleft[jh]=0; outzonelowleft[jh]=0;
}
else
{
    outtopdenleft[jh]=topdenleft[jh]; outzonetopleft[jh]=zonetopleft[jh];
    outmiddenleft[jh]=middenleft[jh]; outzonemidleft[jh]=zonemidleft[jh];
    outbotdenleft[jh]=botdenleft[jh]; outzonelowleft[jh]=zonelowleft[jh];
}
if(cellarrayright[i][4]==0)
{
    outtopdenright[jh]=0; outzonetopright[jh]=0;
    outmiddenright[jh]=0; outzonemidright[jh]=0;
    outbotdenright[jh]=0; outzonelowright[jh]=0;
}
else
{
    outtopdenright[jh]=topdenright[jh]; outzonetopright[jh]=zonetopright[jh];
    outmiddenright[jh]=middenright[jh]; outzonemidright[jh]=zonemidright[jh];
    outbotdenright[jh]=botdenright[jh]; outzonelowright[jh]=zonelowright[jh];
}

outdensityleft.print(cellarrayleft[i][0]+"\t"+dlat+"\t"+cellarrayleft[i][2]+"\t"+dlong+"\t"+
    outtopdenleft[jh]+"\t"+ outzonetopleft[jh]+"\t"+outmiddenleft[jh]+"\t"+outzonemidleft[jh]+"\t"+
    outbotdenleft[jh]+"\t"+outzonelowleft[jh]+"\t"
    outzonelowright[jh]);

outdensityright.print(cellarrayright[i][0]+"\t"+dlat+"\t"+cellarrayright[i][2]+"\t"+dlong+"\t"+
    outopdenright[jh]+"\t"+ outzonetopright[jh]+"\t"+outmiddenright[jh]+"\t"+outzonemidright[jh]+"\t"+
    outbotdenright[jh]+"\t"+outzonelowright[jh]+"\t"
    outzonelowright[jh]);
if(i%4==0)
{
    jh=jh+1;
}
}

outdensityleft.close();
outdensityright.close();
public static double mean(ArrayList<Double> list) {
    ArrayList<Double> templist = new ArrayList<Double>();
    double arrayMean = 0;
    for (int rt = 0; rt < list.size(); rt++) {
        if (list.get(rt) != 0) {
            templist.add(list.get(rt));
        }
    }
    int Length = templist.size();
    for (int rt = 0; rt < templist.size(); rt++) {
        if (Length == 0) {
            arrayMean = 0;
        } else if (Length == 1) {
            arrayMean = templist.get(0);
        } else {
            arrayMean = arrayMean + templist.get(rt);
        }
    }
    arrayMean = arrayMean / templist.size();
    return arrayMean;
}

public static double stddev(ArrayList<Double> list) {
    //
sum=0; sumsd=0;

for(int h=0; h<list.size(); h++)
{
    sum = sum + list.get(h);
}

mean = sum / list.size();
for(int h=0; h<list.size(); h++)
{
    sumsd += Math.pow((list.get(h) - mean), 2);
}

sd = Math.sqrt(sumsd / list.size());
return sd;
LIST OF REFERENCES


BIOGRAPHICAL SKETCH

Akshatha R. Shenoy was born to Mr. K. Ramesh Shenoy and Mrs. Asha R. Shenoy in India. She studied at the Amrita Vishwavidyapeetham University, Amrita School of Engineering Bangalore Campus from 2004-2008. She was awarded the degree in Bachelor of Technology, computer science in April 2008. She worked with Tata Consultancy Services (TCS) as a Software Systems Engineer from 2008 to 2010 in Bangalore, India. She quit the job at TCS in December 2010 to pursue her higher education at the University of Florida. She pursued a concurrent degree in agricultural and biological engineering and computer science and engineering at the University of Florida from January 2011- August 2013.