Luís Miguel Ferreira Reis da Silva Carvalho

Licenciado em Ciências de Engenharia Eletrotécnica e de Computadores

Fuel Mass Mapping for Forested Areas

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador: José António Barata de Oliveira, Professor Associado,
Universidade NOVA de Lisboa

Co-orientador: Francisco Marques, Engenheiro de Investigação, UNINOVA-CTS

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro, 2020

Fuel Mass Mapping for Forested Areas

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

First, I would like to thank my adviser José António Barata and my Co-Adviser Francisco Marques for their support and guidance throughout the making of this dissertation. Despite all the current world affairs, with a global pandemic still raging, and sometimes personal problems, they always came out of their way to help me.

Secondly I would like to thank my group of friends, my partners in crime, and my closest companions on this trip. They always had my back and where always there for me to support me and motivate me.

Lastly but not least, I would like to thank my closest family. My mother, my father, my sister and my girlfriend for putting up with me in times of more stress and pressure. Their unconditional support means a lot to me and I love them for that.

Abstract

This dissertation aims to develop methods of forest analysis capable of estimating certain characteristics of trees, such as their height or the diameter they have at chest height. For that, algorithms were developed capable of dealing with 3D point clouds - originating from a handle sensor mounted on a UAV, more commonly called a drone - and from there the estimation of tree characteristics.

This implementation aims to facilitate the work of forest management and analysis, since a UAV can cover significantly greater distances than a human being and a computer can process more information than a human.

A wide set of algorithms were developed, the system being able to estimate parameters of total height and basal diameter for individual trees, as well as total height for trees inserted in sections of forest. Algorithms for detecting trunks in the forest section have also been developed, capable of detecting the presence of a tree trunk in a certain area of a forest section.

Keywords: Keywords: Biomass Estimation, Tree Height, Diameter at breast height (DBH), Grid Square, Tree Trunk Detection.

Resumo

Esta dissertação tem como objetivo desenvolver métodos de análise florestal capazes de estimar certas características das árvores, tais como a sua altura ou o diâmetro que possuem à altura do peito. Para isso foram desenvolvidos algoritmos capazes de lidar com nuvens de pontos 3D – com origem em sensor lidar montado num UAV, mais comummente designado drone – e daí fazer a estimação de características das árvores.

Esta implementação pretende facilitar o trabalho de gestão e análise florestal, visto que um UAV consegue percorrer distâncias significativamente maiores que um ser humano e que um computador consegue processar mais informação que um humano.

Um amplo conjunto de algoritmos foram desenvolvidos, sendo o sistema capaz de estimar parâmetros de altura total e de diâmetro basal para árvores individuais, bem como altura total para árvores inseridas em secções de floresta. Foram também desenvolvidos algoritmos para a deteção de troncos em seio florestal, capazes de detetar a presença de uma árvore numa determinada área de uma secção florestal.

Palavras-chave: Palavras-chave: Estimação de Biomassa, altura de árvore, diâmetro basal, quadrado de grelha, detecção de tronco.

Contents

List of Figures

List of Tables

Acronyms

3D Three Dimensions.

AGB Above Ground Biomass.

ALS Aerial Laser Scanning.

CBH Crown Base Height.

CHM Canopy Height Model.

DBH Diameter at breast height.

DSM Digital Surface Model.

DTM Digital Terrain Model.

EKF Extended Kalman Filter.

HD High Definition.

IMU Inertial Measurement Unit.

LiDAR Light Detection and Ranging.

PCD Point Cloud Data.

PCL Point Cloud Library.

RANSAC Random Sample Consensus.

RG Region Growing.

ROR Radius Outlier Removal.

ROS Robot Operating System.

SOR Statistical Outlier Removal.

TTH Total Tree Height.

UAV Unmanned Aerial Vehicle.

Chapter 1

Introduction

A reasonable estimate of Forest Biomass presents itself as a key tool in order to properly manage forest ecosystems around the globe. These ecosystems play a vital role in the global carbon cycle [1], as carbon is needed in order for a tree to grow, that way trapping it and avoiding its escape to the atmosphere. Forest Biomass estimation accounts, among other attributes, with CO2 stocks and is very important in the creation of adequate forest models for a proper forest planning. Forest management presents itself as a critical task in Portugal as the country presents a higher risk to suffer from big wildfires due to the effects of climate change.

Because of the threat that poorly managed forest systems present in increasing the consequences of wildfires, it is only obvious, that we should get to know and study our forest ecosystems. In this sense, we will exploit data gathered by an Unmanned Autonomous Vehicle, UAV, to which sensors will be attached, as as our data acquisition medium. The end-goal is to estimate Biomass levels, fundamental in order to assess the quantity of organic material present in a given area, as well being essential to quantify the amount of carbon trapped in a given plot of land , as forest ecosystems contain the vast majority of Carbon stored in terrestrial ecosystems[2]

1.1 Outline of this Thesis

This dissertation is organised as follows:
Chapter 1 provides an introduction to the Dissertation, the motivation behind it, the problem, the proposed solution

Chapter 2 will provide us with a survey and a review of the State-of-the-Art, as well with some useful concepts in the are of biomass estimation.
Chapter 3 presents a general overall explanation of what was intended, what has been done, and how it has been done.
Chapter 4 presents a more detailed explanation of the implementation of our algorithms.
Chapter 5 presents the implementation results.
Chapter 6 presents our conclusion to the job developed.

1.2 Problem

As mentioned earlier, in order to properly manage forest ecosystems we need to be able to quantify certain parameters about them. Forest Biomass presents as a key component in order to assess forests. Forest ecosystems are play a key role in the mitigation of climate change, and so, it an obligation that we look carefully after these ecosystems, so fundamental in the fixation of carbon. That way, science and technology need to come up with solutions that seek to make it easier to study these ecosystems, systems that can assist and make it easier to manage forests, providing data and tools that simplify the job.

In this sense, such a system must ensure certain requirements in order to properly assist in the task of proper forest management: first, it must be of easy usage, easy interaction and easy initiation, second, the system must be highly mobile, and third, but certainly not less important, the system must present the user with a solid estimate of Forest Biomass for the given plot of land at study, in a way that is easy to analyse.

In some literature we find several solutions on how to properly estimate Above Ground Biomass (AGB). One can find several approaches on how to solve this problem, some chose to use a mixed Satellite data, Sentinel-2, and LiDAR data (Light Detection and Ranging) approach [3], others prefer to use Multispectral data and imagery as their main support in order to do the same. Others such as Naesset et al. chose only to use LiDAR data in order to estimate AGB. [2]

It comes to no surprise that the choosing any given solution be it LiDAR only or Multispectral imagery for instance, changes the implementation of the solution, even though the end goal may be exactly the same, AGB estimation, the methods used and the solution path vary with the choosing of the data to be utilised.

There are cases where Satellite Multispectral data imagery is used to help estimate certain data relative to the problem in hands, namely biomass estimation. The specific satellite pointed in the article is equipped with a Multispectral working passively. As the Satellite moves along its orbit, data is continuously collected.[4]

This approach is inflexible because we cannot simply tell a satellite to point in a certain given direction, however due to the longevity of the mission, the Sentinel-2 presents itself as a fine tool to help scientists study earths ecosystems in the long run.

There is some literature pointing to Multispectral Cameras or even LiDARs attached to fixed wing aircrafts. This solution presents itself with the serious advantage of being

able to survey huge portions of ground on a single flyover, due to the great autonomy of airplanes, but it is rather obvious that this method is no solution when you are trying to estimate up to a medium plot of land, as such, fixed wing airplanes are not practical nor appropriate to be used in most scenarios.

Unmanned Aerial Vehicle (UAV), such as the one seen in 1.1, come forward as an adequate air vehicle to be of use in most situations [5], not coming, of course, without its downs, mainly a low flight autonomy, depending on the size of batteries compared to the size of the aircraft.

Figure 1.1: Example of a UAV equipped with a LiDAR sensor. [6]

There are some references pointing out to the usage of UAV-LiDAR systems applied for agricultural purposes, crops health evaluation, water stress levels and even biomass estimation in crops such as maize [7], but this area of application for this kinds of systems is of little interest for this dissertation. Mainly in, agricultural environments, we find much more commonly UAVs equipped with Multispectral cameras, as such application is of much use to this area of study. Autonomous flight systems equipped with Multispectral cameras are able to provide data, such as water stress in an easy and in a more intuitive way. An example of this type of usage presents itself with Tilling et al.[8]

Our main focus is to study and give a proper estimation of biomass presence in a forest environment. This does not mean that any findings may not be used in the field of agriculture, but as Li et. al[7] points out, biomass estimation applied to crops, namely maize, has it's own difficulties and peculiarities, usually related to the density and thickness of crops,characteristics that make it harder to have a proper lidar penetration, making it more difficult to extract certain needed parameters necessary to build a model [7].

1.3 Proposed Solution

Aerial LiDAR sweeping has come forward as a powerful tool capable to provide outstanding data about our forests, key if we want to increase the quality of our forest management. In this dissertation we come forward with a solution that may provide insights to the future of forest management and study, providing scientists with standard metrics,

consistent data, up to the individual tree level, that enlarge our knowledge about our forests.[9, 10] The idea is simple, we get an Unmanned Aerial Vehicle, and equip it with all sorts of technology, including, of course, a Lidar sensor. We propose to develop methods capable of estimating certain tree parameters, such as tree height and diameter at breast height, parameters that can then be used to estimate the presence of forest biomass. The system as a whole must be able to beat certain requirements:

1. The system must be easy to use.
2. The system must be able to analyse Lidar obtained data.
3. The system must be able to provide the user with a series of data relevant to the area of forest management.
4. The system must be able to provide a biomass estimation for the given plot of land at study.

For the purposes of this dissertation, only this last requirement will be the target of our attention. We are to use a UAV equipped with, a LiDAR, two antennas, an Inertial measurement unit, or imu, an Odometry unit, and a computer on board. The imu is important as it captures certain characteristics of flight such as Yaw, Roll, and Pitch as well as accelerations. This uav, will be accompanied with a ground RTK-GPS station, capable of providing GPS coordinates up to just centimetres of certainty. This is crucial if we want to have a proper Point Cloud derived from the LiDAR readings. Given the pandemic situation in 2020 we will resort also to public LiDAR dataset as there were no possibilities of acquiring the amount of data necessary using the RICS uav.

In a not so far away future we can we envision this vehicle being able to operate autonomously and capable of twenty four seven monitoring over forest areas. With a system such as this one, forest biomass estimation is facilitated and increases in scale, as a UAV can reach places places hard for a human being and can cover more territory than a human being can, offering more information up to study. With a system such as this we can study and parameterize large sections of forest, understand their level of development, and therefore better manage our forest ecosystems.

Chapter 2

State-of-the-Art Survey

2.1 Introductory Concepts on Biomass Estimation Systems

Biomass Estimation can be achieved in a multitude of ways. Mainly scientists resort to models and equations that describe relations of scale in the biological world. These types of equation go by the name of Allometric equations. Size and shape have a saying in how organisms capture resources [11]. In our case, the bigger the tree, the more, usually CO2 is trapped in it, and so, the larger is its Biomass. Biomass can be represented as a unit of mass, such as Kg, or as a unit of mass distributed along an area, a Density such as Tonnes/ha. This is the most recurring way to estimate biomass, found in several literature such as Wang et al.[3]

These allometric equations need to be fed with certain parameters, such as Crown Base Height (CBH), Diameter at breast height (DBH), Total Tree Height (TTH), and even more parameters could be pointed out. [11] Allometric equations are specific of each tree species, easy to see, as the relations between for example Tree Height and Crown Length vary from tree to tree. This presents us a difficulty, as we can only study trees for which we know such allometric equations, species for which we posses established relations and ratios from which we can derive and estimate Biomass.

And this bring us to yet another question that we need to bare in mind when we are dealing with allometric equations. They are not easily generalizable, as certain equations are only meant to be used in that kind of environment, i.e. a rain-forest or so on. We were able to find some recent literature on how to solve this question, ways to find a generalised way to calculate biomass by finding certain absolute relationships in trees, certain relations between the various parameters that can be used to build reasonable models capable of providing a fair estimate of biomass[12] .

Such parameters, as the one shown in 2.6, can be derived using Aerial Laser Scannings

(ALSs). LiDAR data an be used and worked on in a way that allows us to retrieve such parameters. Literature indicates how we can achieve Biomass Estimation using Laser Scanning data, or by other words, LiDAR data. For instance [13] uses an individual tree assessment. The method includes a delineation of individual trees, extraction of some data such as tree height and Crown width that are then fed to specific allometric equations that return us a model that gives us an estimate of biomass.

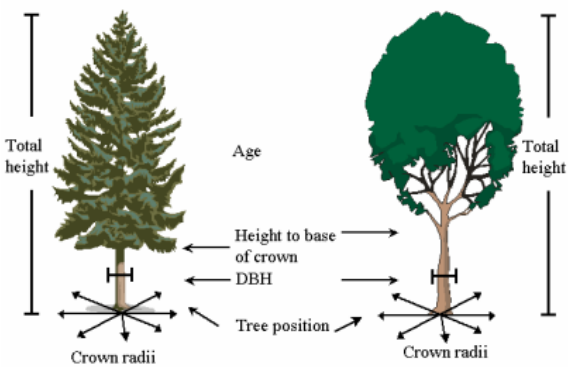


Figure 2.1: In this image we can observe some basic tree characteristics.[14]

2.2 Forest Data Acquisition Methods

There are several mediums we can use to obtain forest data. Technologically speaking, the most basic way we can do this is by using a High Definition (HD) Camera, taking pictures of the plot of land in study, and then analysing those same pictures, be it through processing software or other. Another resource we can use, is a Multispectral high definition camera. This device provides us with with high quality imagery being able to capture near infra-red radiation, invisible to the human eye. Finally, we can use LiDAR Technology to provide us with high quality datasets, from which we can, on a later stage extract detailed information, in a form of point clouds.

This section should serve to make it clear to the reader that there are several ways to collect forest data, each with it's own qualities and defects, each being fit for a different specific application that, we hope, should be clarified along this section.

2.2.1 High Resolution Camera Data Acquisition

The usage of High Resolution Camera Data Acquisition is not that common in forest studies that seek to obtain data related to Biomass. It is easy to see why. In order to obtain to compute the necessary parameters needed to estimate Biomass, such as Canopy Density, tree height parameters, diameter at tree base, and so on.

An example such usage of high definition imagery applied in a systematic way comes with Apostal et al [15]. In this paper, we can observe that the applicability of high definition forest images relates mainly with tree detection, identification, and delineation.

The researchers where able to find some relations between tree delineation and tree height, stem diameter, and biomass, suggesting some potential to estimate such values.

2.2.2 Multispectral Camera Data Acquisition

The purposes of multispectral technology are not that hard to comprehend. Multispectral cameras capture data on the reflection of light on objects, and this types of sensors are able to capture Near-infrared (NIR), short-wave infrared radiation (SWIR) and other wave-lenghts, invisible to the human eye.[16]

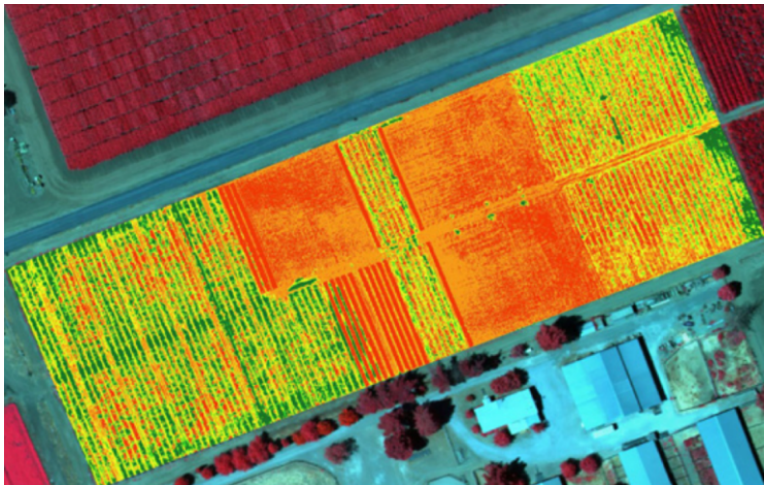


Figure 2.2: Example of a Multispectral image in agricultural purposes.[17]

From multispectral imagery and through the usage of algorithms we are able to obtain detailed information related to Vegetation Indexes, such as Normalised Difference Vegetation Index (NDVI) or even Photochemical Reflectance Index, that help us study the well being of the ecosystem at study. This information proves useful allowing for a better management of land or even when studying environmental protection measures.

As such, this technology finds its mains usage in the monitoring of plant health and management, because from multispectral imagery we can pinpoint diseases, pest damage, optimise fertilisation and assess water stress. AS such we can find many application of this technology in agriculture. One of such examples presents itself in [8], where the authors propose to study the level of water stress from multispectral imagery.

Examples of the usage of this technology in assessing forest ecosystems, biomass for instance, come with some kind of attachment with LiDAR technology [3]. This support seems only natural, as we cannot access 3D characteristics of trees, such as Tree height, supporting our analysis only with Multispectral imagery. Now, this characteristics prove key if we want to properly estimate Biomass.

In the case of this thesis, a multispectral approach is not best suited, as we pretend to estimate biomass and because of that fact, we need data from which we can extract certain 3D parameters, that we cannot calculate through multispectral data.

2.2.3 (ALS) LIDAR Acquisition

Light Detection and Ranging (LiDAR) is an active remote sensing system that can be used to measure vegetation height across wide areas. LiDAR is a useful tool for scientists as it allows to obtain complex datasets that permit scientists to estimate certain characteristics, such as Biomass, over large areas of forest.

LiDAR technology uses energy in the form of light. Light is at the core of this technology, the main principle being the same as sonar, but here applied to light. Rapidly firings laser provides the energy, the light, we then measure the time offset between the departure and arrival at source of such laser beam. This allows us to estimate distance, as the speed of light is a known variable. This is the main principle of this technology.

In order for a LiDAR System to properly work though, we still need a few more details. A LiDAR system must be used jointly with a GPS system. This is so, in order so that we are able to convert distances of travel to coordinates. An Inertial measurement unit is also necessary, providing with orientation, yaw, and pitch of the vehicle where the LiDAR device is set. [18]

Light Energy is an aggregation of photons. As these photons make their way to the ground, they may find obstacles like branches of a tree for example and bounce back their way to the sensor. The rest of the photons keep going their way down towards the soil, and them bouncing back to the sensor. This introduces an important concept. Multiple returns may be recorder from one light pulse. [18]

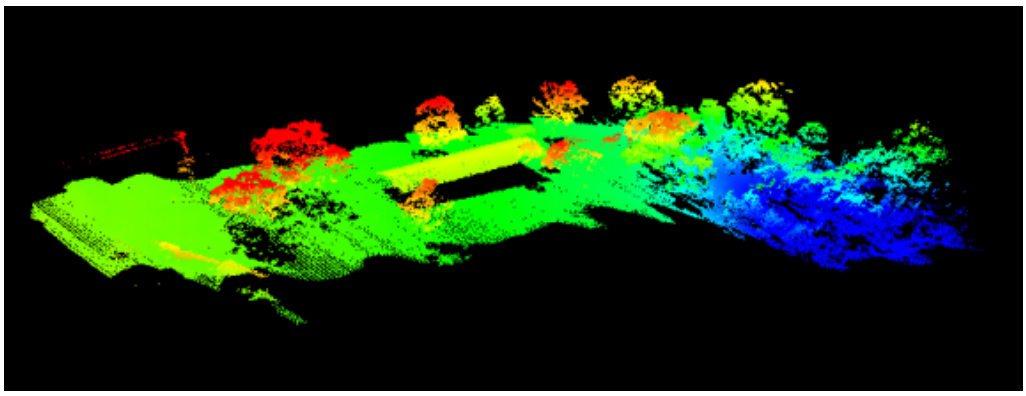


Figure 2.3: Here lays an example of a Lidar Pointcloud. In red we see points that present a higher elevation in the dataset, and in blue we see the points with the lowest elevation. We can also identify some trees in the dataset.[5]

The end-goal in LiDAR Technology is to obtain a Pointcloud from which we are able to extract useful information, namely, in our case, forest data.[16]. With today's technology we can expect yields of 100 up to 500 points per square meter with a vertical accuracy of 2-3 cm. A point in a point cloud is described by an (x,y,z) coordinate.

There are two main branches in LiDAR technology, discrete LiDAR, and Multi modal LiDAR, each with its strengths and utilities. Discrete return LiDAR systems allow for a few returns returns to be recorded for each light pulse, normally situated at the peaks

of energy of the returning wave. Full wave form systems record the amount of energy returned to the sensor on a very high frequency rate, imposing a small noise threshold, making it seem as if it were in a continuous fashion.[9] The differences between the two can be seen in fig. 2.4.

2.2.3.1 Discrete LiDAR

In literature we can find several examples of the usage of discrete LiDAR. Discrete return LiDAR are usually small footprint (typically 20–80 cm diameter) systems that record one to several returns through the canopy, in a vertically non-systematic manner. [19] Discrete Systems are able to provide the user with extremely dense point clouds and present a better resolution of ground and canopy surface. The ratio of ground to non-ground point volumes directly indicates light penetration through the canopy and has been shown to correlate strongly with forest structure. [20, 21]

Discrete LiDAR provides the user with an elevation component that allows for it to measure elevation with high precision, unlike other aerial images such as Multispectral imagery, [22]. For this reason discrete LiDAR is often used in the fields of topography, but it also finds its place in the study of forests, being highly suitable in the measurement of certain parameters such as Vertical canopy cove, angular canopy closure and leaf area index. [23] gives an example of the usage of discrete LiDAR, that supports that it hsa much use in the estimation of forest characteristics that that mainly only need the 3d points referring to the ground surface and the canopy surface.

As for the estimation of biomass, discrete LiDAR presents us with some difficulties. As discrete LiDAR cannot capture the full returning waveform, some information is lost, mainly in the middle section of a tree, and mid to low canopy information may as well be lost, or presented with less detail. In order to present a more solid estimation of biomass of a given tree, we need those intermediate points, that allow for a proper biomass estimation. For this motive, discrete LiDAR is not well suited for biomass estimation.

2.2.3.2 Full Waveform LiDAR

When you record the whole of the returning energy as a continuous wave, that is what we call full-waveform LiDAR.[24] The full record of the whole returning energy, provides the end user with more information, allowing for a more open interpretation of the physical structure, and backscatering characterises of illuminated surfaces, such, as in the case of this current thesis, trees,[25]. In physics backskatter relates to the reflection of particles back to the direction of emission.

In this case, with full waveform LiDAR, we are not only sampling discrete periods of return signal, or just spikes in return, we are assessing the full waveform, spikes of return and the information in between.

As Full wave form LiDAR presents us with more comp lete datasets,with more information concerning to mid tree information, it allows for a more detailed study of tree

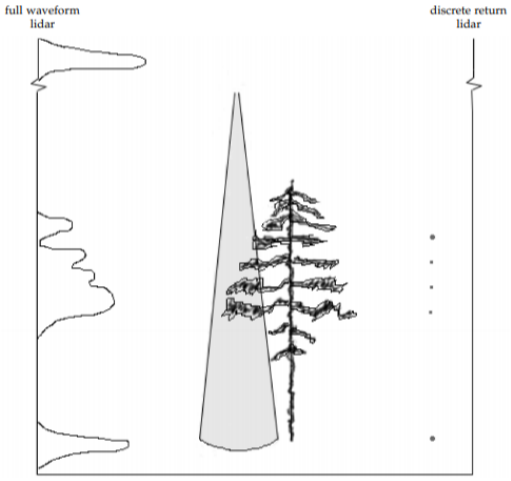


Figure 2.4: In full waveform Lidar the full returning wave is recorded as opposed to discrete Lidar where only a discrete number of pulses is recorded, usually four pulses.[9]

structures and for complex analysis of tree characteristics. We can find in literature several usages of full wave form LiDAR applied to forest study. Full wave form LiDAR is seen as the mostly complete tool to study forest inventories, and certain dendometric characteristics of trees.

This aptitude and versatility is seen in several papers where full wave form LiDAR is used among other things, to study stem volume and classification of trees [26], to estimate biomass using machine learning approaches [27] and estimate biomass with small footprint full waveform LiDAR [28], to predict certain stand characteristics of trees [21] like tree height Volume and Basal area with very good results. We can also find mixed approaches in the study of forest ecosystems, Wang et al.[3] use Sentinel 2 multispectral data jointly with full waveform LiDAR data.

Full waveform technology will remain a preferable choice for 3D vegetation mapping that requires the analysis of tall vegetation canopies (e.g., mature forest), as long as the number of multiple discrete returns provided by advanced DR LiDAR systems is limited by four. In these cases only four multiple discrete returns would leave wide gaps in the modelled vertical canopy profile.[22]

2.3 Tree Detection and Biomass Estimation Methods

In order to be able to estimate and calculate biomass, we first need to distinguish vegetation in the point cloud obtained from the LiDAR data obtained during the aerial survey of a given plot. More important than obtaining the LiDAR information in the form of a point cloud is sorting any sense out of it. There are several ways we can do this, including 3D Segmentation algorithms, Machine Learning classifiers and neural networks.

2.3.1 Tree Detection: 3D Segmentation

Being able to differentiate ground from tree and individual trees from each other may prove valuable in order to estimate biomass. That being said, there are several ways we can achieve this goal, the main way being using a technique called 3D Segmentation.

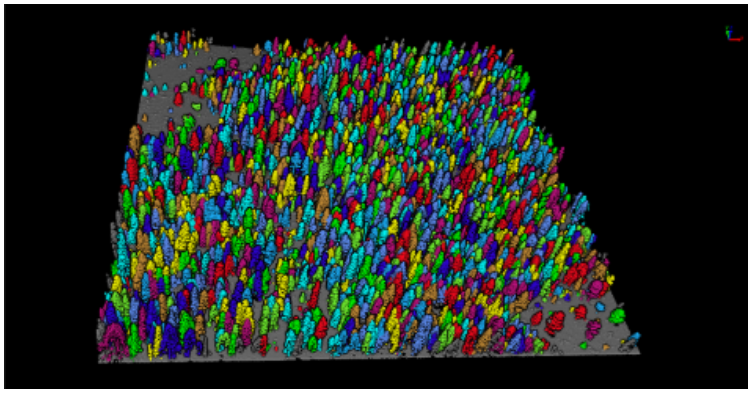


Figure 2.5: This image shows us the end goal in forest segmentation. To obtain individual trees in order to be able to extract forest metrics.[29]

3D Segmentation is one of the methods one can use to achieve point cloud comprehension. These kinds of algorithms have seen wide acceptance when dealing with 3D structures such as point clouds. These types of algorithms allow for the identification multiple structures namely buildings, vehicles, individuals and trees. There are a wide range of methods in this family of algorithms, Edge based methods, Region based methods, Attributes based, Model Based methods and even Graph Methods, [30].

2.3.1.1 Edge Based Methods

These kinds of methods work by detecting boundaries of several regions characterised by their intensity in the point cloud. Each point in the cloud is characterised by an x, y, z coordinate as well by an light intensity, that is bounded by the light reflect by the object that is to be detected. These kinds of algorithms are very sensitive to noise, but present some advantages in runtime [30]. The basic idea in play is to detect chances in intensity and thereby detecting an edge, a boundary region. This is advantageous in clear datasets, that is, datasets where it is relatively easy to distinguish different areas and where different boundaries are present, that is, different areas where a clear change of intensity is present. In the case of forest data these types of algorithms may present some difficulties as many times, with thick forest canopies, it is not easy to distinguish where the canopy of a tree ends and another begins, and, consequently, such areas will tend to present the same intensity, making it difficult to properly segment the trees canopies.

Bhanu et al. [31] proposed a slight change to the basic idea of these algorithms proposing the computation of a gradient detecting changes in the direction of normal vectors to the surface and fitting 3D lines while the gradient remains unchanged in direction. That is, for example, while light intensity goes down you can aggregate that point to

the segment, when else, light intensity starts to go up by more than a slight margin, the algorithm does not join that point to the segment, as it will certainly belong to an area of boundary, where another segment begins.

2.3.1.2 Region Based Methods

Region Based methods combine neighbouring points with similar properties. There are two subdivision of this family of methods, Seed - Region Growth and unseeded region growth. Seeded region algorithms start segmentation by choosing a random number of points, called seeds, and start growing out from there. If neighbouring points achieve certain criteria they are absorbed into the region. The initial seeded algorithm was proposed by Besl et al[32]. This approach is very sensitive to noise and time consuming, and is highly dependent on the selected seed points. Depending on that, we may achieve over segmentation or under segmentation. [33] Over segmentation occurs when the target point cloud is divided into way to many section. We get more segments that account for more objects than those that were initially present in the original scene. Under segmentation is exactly the opposite, we get sections that most times include more objects, in this case trees, and the result is that we get less segments than there are trees in the original scene.

In the unseeded region growth algorithm all points are grouped together. This single group is then subdivided into smaller regions based on a confidence level. This approach presents several difficulties, those being, where do we divide, how do we divide, and how do we establish the confidence calculation. The answer to these questions will affect the end result.

2.3.1.3 Attributes Based Methods

Attributes based methods employ the principle of clustering points of the 3D Point cloud based on certain similar characteristics. The attributes that the algorithm uses are extracted from patterns in the data that the algorithm finds. In a first stage these kinds of methods start by computing the attributes necessary to each point in the point cloud, secondly these types of segmentation algorithms aggregates the points based on the similarity between the attributes of each point. These patterns may be or may not be obvious and as such these types of algorithms may present behaviours out of the ordinary.

These methods are highly dependable on the quality of attributes derived from the pointcloud system, and for that reason, these attributes should be calculated in the most precise way possible, [33].

2.3.1.4 Model Based Methods

These kinds of algorithms are very intuitive to comprehend as they base their existence in mathematical models of certain geometric shapes known, be it a plane, be it a cylinder, a sphere,which they then use to group points that fit those standard shapes. In these types

of algorithms the points are agglomerated accordingly if the algorithm is able to find some correspondence between those points and models of geometric shapes the system knows. These types of algorithms are purely mathematical, and in that sense, they produce solid results behaving very well with outliers.

The state of the art for model fitting came with Fischer, [34],that introduced Random Sample Consensus (RANSAC) back in 1981. RANSAC roughly operates by creating two sets of points: inliers and outliers. Inliers are points that fit a given model, and outliers, points that don't fit that pattern, be it for example a plane surface. [35] These algorithms present some difficulties when dealing with data that is extremely irregular, presenting several different shapes or as the complexity of the point cloud increases. Their performance may also be affected by low point density.

2.3.2 Estimation of Biomass

We have spoken about how we can identify ground, trees and what methods can be used in order delineate such forest components. Another conversation is about the ways we estimate biomass. How can we estimate biomass from LiDAR data? There is no one way to achieve this goal, but usually the end goal is to build a model that can return a biomass estimation based on certain parameters. First of all, with biomass estimation it is important to try to know beforehand the main tree species present in the plot of land in study so that the best possible allometric equations can be found, as certain parameters in this equations may refer to specifics to each species, as an example wood density differs from species to species and it may be used as a parameter known beforehand. Another important issue is that in order to be able to build consistent models we need to posses solid data beforehand of the plot of land at study, allowing us to be able to have data upon which we can create a model and compare its results to.

It is important to notice that allometric equations establish relations between certain chosen parameters for a given species, and because of that, applying a model that was built to be of use in a temperate forest may prove deficient in a Nordic forest. Allometric equations mirror strong relations between parameters and the end goal to be quantified, in this case biomass, and an ability to quantify this relation, [36].

Some studies have shown up trying to find general relationships that may be applied to many different forest ecosystems. They try to assess a minimum set of meaningful parameters that can be used to find such general models.[12] We should aim for general ways to estimate biomass as they can then be easily used in a wide range of sets and forests.

Knapp et al. [12] seeks to find such general relationships between a general set of attributes in order to be able to build models that can be used in some different scenarios. They estimate total biomass as a sum of particular, individual tree biomass. They differentiate between above ground biomass, for which a definition as already been given, and below ground biomass, that accounts mainly for root systems biomass. From LiDAR data

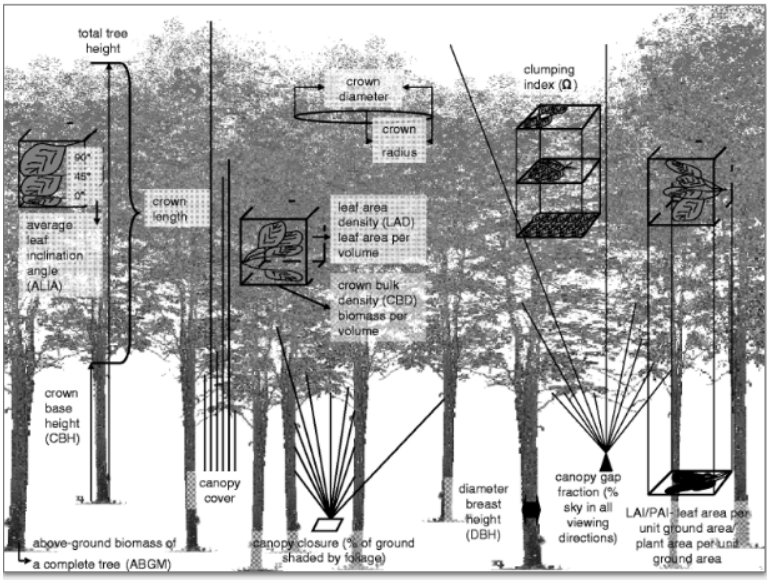


Figure 2.6: This image shows us the main forest metrics that we can use in order to estimate biomass.[37]

they calculated several tree parameters. Some parameters that Knapp et al. [12] chose to use included Mean canopy height, maximum tree height, vertical tree heterogeneity, average wood density (this parameter is the only one here that is not derived from LiDAR data) and maximum possible stand density. The later refers to a relation between basal area and total area.

In the case of Knap et al.[12] several parameters where calculated from LiDAR data, and several models where built from different aggregations of those parameters. Those models where then evaluated against pre-existing data of some of the plots they were studying with satisfactory results. We should notice that the parameters that they used for each model were chosen in the model building stage, not before. As for that we may end up with different models for different plots of land. Because of this, the best model for a given plot of land may differ from the best biomass estimation model overall as the parameters chosen and the metrics used may differ.

Bouvier et al.[38] tries a different approach, choosing beforehand some parameters and forest metrics in order to build the models used for biomass estimation. These metrics were obtained from LiDAR data and had to capture different aspects of forested areas. He obtained solid results, but those results were improved when the models were built taking into account site specific coefficients such as wood density of the most prevailing species in the plots at study.

A general method for biomass estimation can be understood by three steps. First you need to filter and process the LiDAR dataset, in order to make it useful. This includes filtering outlier points, next comes interpolation and segmentation of the pointcloud points and then aggregating such points into the creation of a Digital Terrain Model (DTM) and a Canopy Height Model (CHM). In Knapp et al, [12] in the filtering stage, we

can see that they choose willingly to ignore points below a certain base height set at 2 meters, as they only want to keep points that they are certain that belong to the canopy of a tree. This would prove useful ahead for the extraction of metrics.

On a second stage we are asked to extract forest characteristics which we can use to build our models of biomass estimation. These metrics most certainly will include tree height, and diameter at breast height, which is closely related to tree height. Tree characteristics extraction baser on LiDAR datasets is vital in the build of a solid biomass estimation model. [39]

A solid and widely accepted model to estimate biomass is introduced by [40] and is given as:

$$AGB = b * dbh^2 * height, \tag{2.1}$$

where b is a coefficient to be calculated and depends on the type of tree at study. This equation takes into account the whole tree, being able to estimate above ground biomass with good results. Diameter at breast height and tree height can be easily retrieved from LiDAR data.

Logarithmic transformed models have also been studied for biomass estimation. Fallah et al [41] used single tree biomass estimation based mainly on DBH. From LiDAR data they segmented a Digital Terrain Model (DTM) and a Digital Surface Model (DSM), from which they then calculated a Canopy Height Model (CHM). From here, single tree detection is obtained employing an algorithm that detects the tree top height. Diameter at breast height is extracted given the close relation between it an a tree height. this will be the main variable used and with fairly good results Fallah et al where able to estimate biomass.

2.4 State-of-the-Art Observations and Conclusions

This section will provide a conclusion and an opinion on which data acquisition and methods to use for the execution of this project.

2.4.1 Forest Data Acquisition: Discussion

The discussion in section 2.2 allowed us to see the several methods we can use to obtain data in the area of forest monitoring, management and analysis. Each method has it's advantages and it's weaknesses, being good at certain specific tasks and best suited for different application. In remembrance, we discussed high resolution camera data acquisition, Multispectral camera data acquisition, and the usage of LiDAR to capture forest characteristics and data.

High resolution cameras present as the simplest solution and most intuitive way to capture any type of visual data. Even though this is truth it does not present itself as a valuable way from which we can extract more complex data other than visual data. Using software we may be able to extract nothing other than limit bounds of individual trees

or even health related aspects given by coloration characteristics of a tree. Consequently, HD Camera's are not well suited for the purpose of biomass estimation, and, as such, shall not be used in such matter.

Multispectral camera imagery is a fine tool in forest studies. Multispectral Cameras give back more information than simple HD Cameras. The access to near infrared imagery, consisting of the several ranges of infrared, be it, near, short or thermal infrared allows for a more detailed and complex study of natural ecosystems. The kind of information obtained from this types of sensors allows for studies mainly in the area of plant health as shown in section 2.2. Usages of this type of equipment in the estimation of biomass, by itself, finds little support in literature. For such purposes multispectral cameras appear as supporters to complement data and to allow for a better extraction of parameters. As such this will not be the sensor to be used.

LiDAR sensors have full advantage for usage in biomass estimation, it allows for a complex study of forest ecosystems as it enables us to retrieve metrics unobtainable using other methods of data acquisition. We're talking about canopy density metrics, the ability to build forest stratus,and the advantage of easily retrieving tree height's and means. Because of these characteristics, and as explained before in an earlier section of this thesis, 2.2, LiDAR sensors are very well suited for the goal in sight. Between the choice of a discrete or a fullwave form system, we'll be choosing the latter, as it allows for more detail in the mid tree section, as the sensor records the full returning wave. For these reasons full wave form LiDAR will be the chosen tool in the development of this thesis.

2.4.2 Tree Detection and Biomass Estimation: Discussion

Tree Detection is very important in the process of biomass estimation. In section 2.2 we've discussed Segmentation methods found all over literature, and each has it's own weakness and strengths. Biomass Estimation usually follows a general method discussed in a previous section.

Edge based methods work out segments and agglomerate points using light intensity as it's main criteria. This may prove useful in situations where we can easily distinguish large planar areas where light reflectance may be the same, and consequently segmentation via intensity may prove easier. In a forest environment situation this may not be the case, except that is for the detection of a ground surface. Still it is easier if we can use the same method for the whole panorama, that is ground surface and tree structures.

Attributes based segmentation may not be of use to our end goal, that is, tree segmentation. This type of segmentation unites points that present similar attributes, these attributes need to be calculated beforehand and because of that this may increase the complexity of the general system. We have also discussed model based segmentation methods, and these kinds of methods may prove valuable in forest ecosystems, as they, in a simplified way present some rough geometric forms such as spheres as canopies,

and cylinders as trunks. As such these types of algorithms should be given a shot in the segmentation stage of our work.

As such region based based algorithms may prove to be the best solution in order to achieve the goal of a proper segmentation. Region based methods, and more specifically seeded ones may prove valuable in forest environments where the goal is to group neighbouring points that respect a given characteristic, this will probably be the case with tree canopies.

Biomass estimation methods usually include include the before talked segmentation stage, calculation of attributes and model construction. We have already discussed segmentation before.Using the segmentation step, a attribute that can be easily extracted from the point cloud is tree height and mean tree height, as such, these attributes, which are closely related to biomass will definitely be used in a model build for such intent. Working on top of the segmentation, we may as well easily calculate canopy densities, which are as well closely related with biomass, as seen in 2.3.2. As for a model we will aim to implement a general model as pointed out in [38]. This will allow for a system that can be used in different scenarios, allowing for a greater implementation and usage of our system.

Chapter 3

Biomass Estimation from Lidar Data: Proposed Model

3.1 Overview

In this section we will provide an overall explanation of the structure and organisation of the proposed system capable of estimating biomass from certain tree characteristics such as tree height, and diameter at breast height. In short, the system is composed of two main blocks, a mapping one, which deals with the raw data reconstruction, being responsible with the correct reproduction and also of rebuilding, in laboratory, the LiDAR scanning collected in the field.

The second major block of the system is responsible for Point Cloud Data (PCD) imagery manipulation and parameters estimation for forest landscapes - from hence forwards designated as forest sections- and this point will be the main focus of this dissertation, pointcloud manipulation. The main goal is by dividing each forest section up for analysis into smaller squares- from hence forward designated as grid-squares - adopting a grid like approach,then estimate forest parameters such as tree height, in order to produce an multi channel image image representing the information contained in the PCD file, down by a scale and viewed from above, where each colour channel represents a different parameter.

Figure 3.1 seeks to show a visual overall of the wanted system architecture and implementation. In terms of user control, the system is developed in such a way that the user is able to control some parameters of the system operation in order to influence and fine tune the end result.

What has been implemented falls a little short of what we would like the whole system to be. Mainly, as it is, the system is capable of estimating height parameters for both individual trees and forest sections, and estimating diameter at breast height of trees that

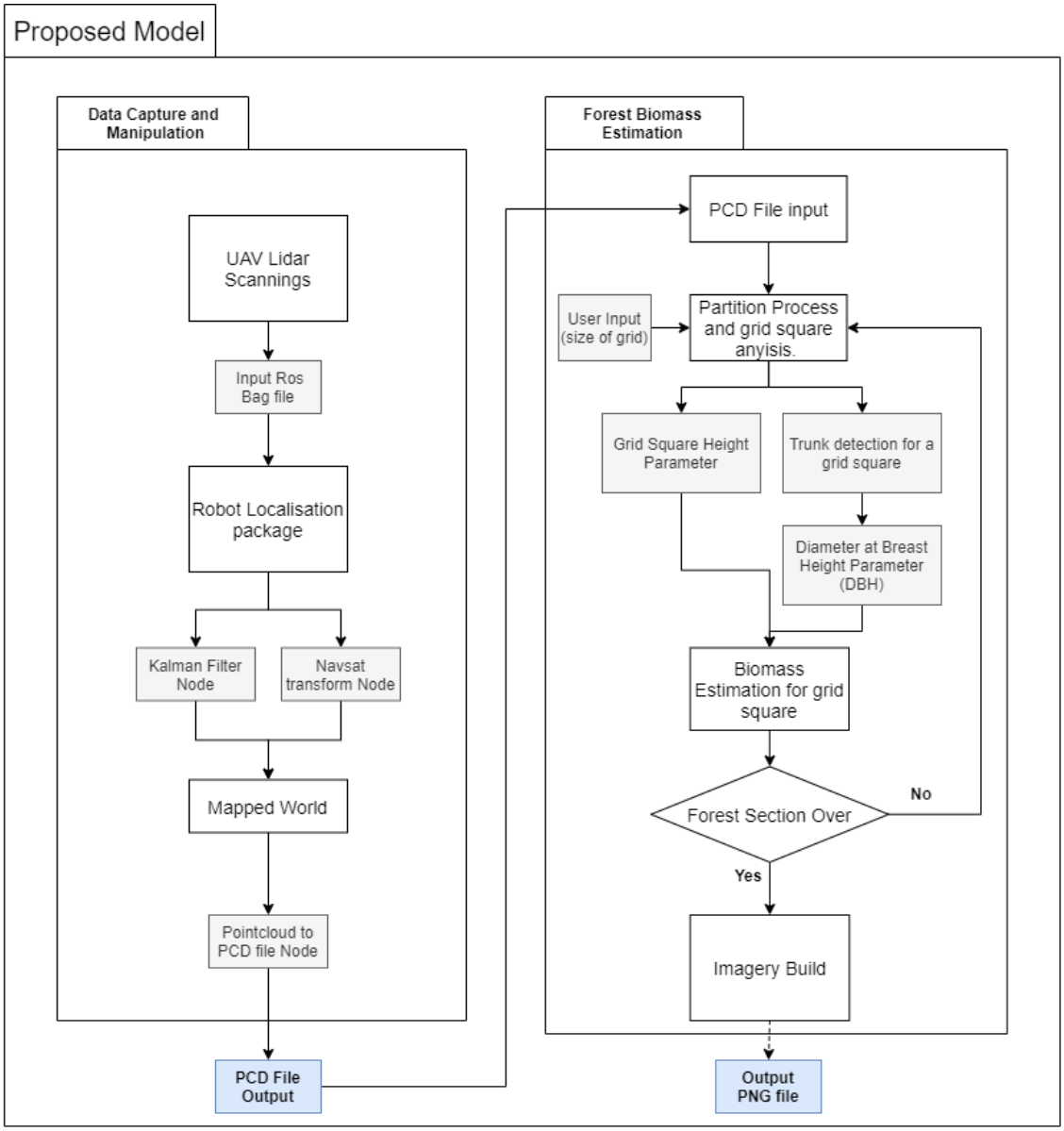


Figure 3.1: System architecture model.

have before been segmented, from the landscape, that is, individual trees. The system is also capable to deal with forest sections saves in a .pcd form file. The system divides the forest section to form a grid pattern and estimates heights for each grid square, the system is also capable of detecting tree trunks and if they are present in a given square.

Tree trunk detection is very important as it is fundamental if we then try to estimate diameters for tree trunks and as it also allows to ignore grid squares that may present height readings, but that don't have a tree present in it. The grid square structure presented itself as a more straightforward approach, as further down the road, if someone tries to implement the multi channel image file output in the future this structure is more intuitive when dealing with image build.

3.2 Data gathering and treatment

In this section we will try to provide a general idea, of how each of the system's blocks work. Firstly we will approach the matter of how to create a more workable form of data starting from the raw gathered data manipulation in order to make it workable in the form of a PCD file. Secondly, we will analyse the meaning through which we take a PCD image and transform it into an multi channel image file and then the block responsible by the parameters calculation and multi channel image image output.

3.2.1 BAG file manipulation

An UAV operation, running ROS, produces multiple data that needs to be saved and captured in order to allow the user to study the information posteriorly. The standard mean to save such information, in a ROS environment, is by using a ROSBag file. A ROS BAG file is the primary mechanism in ROS for data logging. Bags are typically created by a tool like Rosbag file, which subscribe to one or more ROS topics, and store the serialised message data in a file as it is received. These bag files can also be played back in ROS to the same topics they were recorded from, or even remapped to new topics [42].

We now have a standard medium to storage the data from our flights in the field, BAG files are built in into the ROS environment and as such there are plenty of ways to deal with such files in a laboratory environment, allowing us to study such information long after it has been produced. For us, this was the most desirable medium, as it allowed to capture more even data beyond theLiDAR readings. Bag files allowed us to store IMU and also Odometry readings that would prove fundamental in our work further ahead.

We will follow by describing the process by which we take a bag file and turn it into a PCD file. Raw data coming from a LiDAR sensor in motion, as attached to an UAV, does not translate properly the positions of objects nor their relations to to each other, making it impossible to determine where things situate in our world. Trying to visualise LiDAR data as it comes directly from the LiDAR sensor, as in the form of a BAG file is extremely counter-intuitive for human beings.

As it is, if we are to reproduce a bag file as it comes, we would find that it only presents us with a birds eye view, or in our case, drone's eye view (FIG XX), of the world. That is, what you see is the world going past you because you are the main system referential i.e. the world turns around you. In short, we first need to create a reference system upon which we can base our mapping and from which we can proceed to estimate our position in the environment, thus creating an understandable world that we can interpret and retrieve information from.

After building a coherent frame referential and thus being able to correctly reproduce the bag file, we can estimate a map of the entire scenario according to the real world. From here we can then proceed to into turning the information contained in the BAG file into a more suitable and friendlier way that allows us more liberty as regards to cloud manipulation.

3.2.2 PCD data file manipulation

In order to manipulate and better extract information from our point cloud, we need a better mean and a more friendly data file. the PCD data file came up as being the best solution to our problem as it is fully integrated with ROS, and has plenty of support with several code libraries, namely the PCL-ros library [43], Point Cloud Library (PCL). The PCD file format is not meant to reinvent the wheel, but rather to complement existing file formats that for one reason or another did not/do not support some of the extensions that PCL library brings to n-D point cloud processing.

PCD file characteristics that make it very useful:

1. The ability to store and process organised point cloud datasets – this is of extreme importance for real time applications, and research areas such as augmented reality, robotics, etc;

2. Storing different data types (all primitives supported: char, short, int, float, double) allows the point cloud data to be flexible and efficient with respect to storage and processing.

3. Invalid point dimensions are usually stored as NAN types.

3.3 Grid approach to forest sections and end goal output

In this body of work we decided to take a grid like approach to measure and estimate biomass presence in a given area. We call it grid like because the main goal is to divide a given territory into smaller parcels that are more easily manageable and workable, from which we can then use our tools in order to estimate tree characteristics from which we can then properly estimate biomass presence. That being said, the end goal is to present in a multi channel image file a visual representation that seeks to translate the presence, or not, of biomass in a certain are. To the subdivisions - grid structure - of a forest section

we have given the name grid squares and from here forward, this will be the term used to designate the smaller subdivision of a forest section. The grid square will be the core component when studying and approaching forest sections.

Dividing a given territory by grid showed up to an intuitive way to deal with large sums of pointcloud information, allowing at the same time for a smoother implementation of our algorithms. For each square, the system was thought out to be able to estimate a height value, evaluate if there are any trees present and finally to try to estimate a diameter for each tree. Finally with a height value and diameter value we would be able to estimate biomass presence for an individual grid square belonging to the grid.

To produce this body of work we had access to a fix number of bag files in order to develop the map building capacities of the system. To develop the PCD file manipulation we were able to retrieve from the previous stage a number of PCD files, corresponding to bits of terrain we were especially interested in. We also had access to a database compromising of several already segmented trees, and the section of forest composed by such trees.

As said earlier, our end goal is to produce an multi channel image image, which is intended to represent a top-down view point, that is, it represents the presence of biomass as if seen from above. We decided to use multi channel imagery mainly because it allows to insert at least three different types of information, one for each colour channel. We chose multi channel imagery as it is an intuitive mode of transmitting this type of information, and in our case a practical mode of storing information. All throughout literature we can find examples of biomass mapping, as if being a kind of heat map, giving indication about where biomass is present, or not, in a forested territory. Our approach even thought a bit different will allow to store information about biomass presence in one of the colour channels, and also about average tree height in a given square and Average DBH in a given grid square, for this using the remaining colour channels.

Chapter 4

Implementation

This chapter will dive into further detailing about the implementation stages of the methods described previously in a more general way, seeking to clarify the implementation of both the Bag file manipulation stages, and the PCD file manipulation block. Each of these blocks is composed of several parts that work in order to achieve the goal proposed, the first block deals with mapping and world reconstruction through manipulation of a BAG file. The goal here is to get a plausible world representation that allows us to obtain usable PCD files, in order to estimate biomass presence. For this one had access to some Bag recording file obtained both in the campus of our college site and several other bag files referring to field trips.

The second block goal is to deal with PCD files and from them obtain tree metrics and parameters. Focus was put on obtaining diameter at breast height and tree height as our main parameters, as these are the main parameters used to estimate biomass quantity. The implementation stage of this second block was divided into two parts. These methods were first tested in individual trees - fig. 4.1 shows the work developed in individual tree analysis - and from there one looked to adapt said algorithms in order to work for sections of forest - fig.4.2 shows the work developed related to Square grid analysis. To achieve this, one had access to a database consisting of 26 individual trees, mainly beech trees, belonging to a forest section that also was in our database. We will also approach the process of obtaining PCD files from a Bag recording file after it is rebuilt.

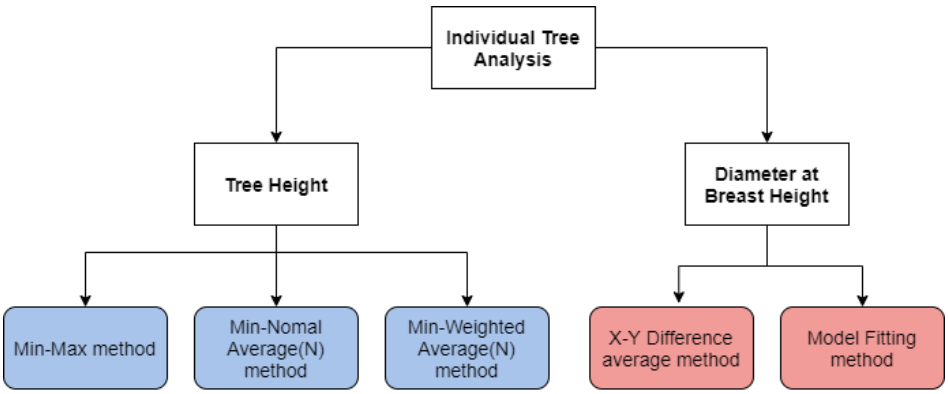


Figure 4.1: This schematic shows the work and algorithms developed in relation to individual trees analysis.

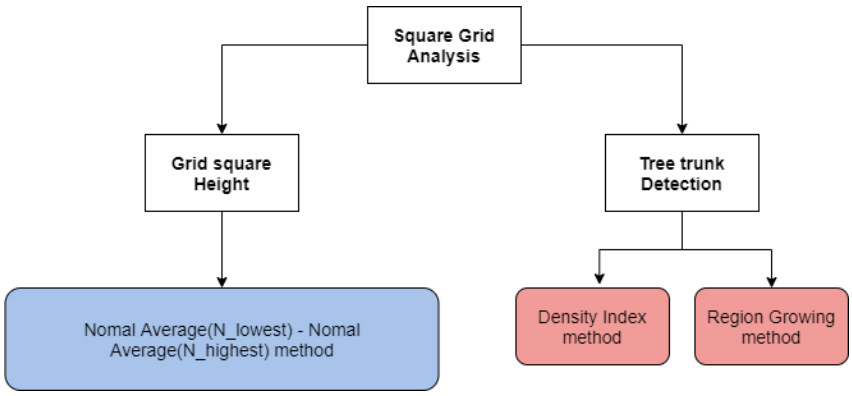


Figure 4.2: This schematic shows the work and algorithms developed in relation to Grid square analysis.

4.1 Reproduction and mapping of the bag files

Before being able to extract any forest data first one needed to grab the RosBag files and work them so that it becomes a palpable representation of the scene in the real world. RosBag files are the main medium to safe information in a ROS built system, it allows to record and save the messages from the several topics allowing to reproduce them at a later date. From field trips one was able to record some RosBags files representing some fly-overs of our UAV over different terrains.

When you open one of these files and try to reproduce them in a pointcloud compatible environment, RVIZ was used to visualise the data, and once you call in the right topic referring to the pointcloud information, it is noticeable that it does not make sense, as the pointcloud is just being reproduced with the UAV standing in the same place. In order to extract some order and certainty from the information contained in that file, and from the point cloud topic, one first needs to make some operations.

Firstly let us lay down the information contained inside our RosBag file.

- **RosBag file topics:**

- dji_sdk/imu: Transmits the accelarations that the UAV is sustaining at a given moment along each axis. Linear acceleration comes in m/s² and angular acceleration comes in rad/s². This data allows to estimate the UAV heading and next position over time
- /odom: Allows to determine the UAV's location in the world and relate sensor data to a static map, we need an odometry source to publish both a transform and a pose (yawn, pitch, roll) as well as a navMsgs/Odometry message over ROS, containing velocity information. The velocity information allows for the estimation of the UAV's current position starting from it's initial position.
- / pointCloud: Transmits the point cloud raw data, as viewed from the LiDAR viewpoint. It is completely disconnected from the world frame.

These are the main topics and the ones we will be focusing our attention upon, contained in our RosBag file. Being aware of these topics we can then proceed to solve the two main problems when dealing with map reconstruction, creating a valid referential structure, with the structure frames of both the UAV and the world rightly positioned in relation to each other, and then integrating the IMU data and the Odometry data to be able to correctly update the UAV position in relation to both the world frame and the place it spawned in the world, as in the beginning of time, moment 0 in our RosBag file, both the world frame, and the UAV frame correspond neatly.

In order to help us with the frame system and also in order to keep the relative position of our UAV correct throughout the time in relation to the world frame, we will utilise a ROS package called robot-localisation [44], that already implements and solves this problematic. This ROS package allows for global estimation of the position of our UAV in the world scene.

The robot-localisation[44] package has an implementation of an Extended Kalman Filter (EKF). It uses an omnidirectional motion model to project the state forward in time, and corrects that projected estimate using perceived sensor data.

So the Extended Kalman Filter helps us estimate the UAV position relative to the map frame, that is, the real world. In order to properly work we first need to set some parameters in some of the files responsible for tuning this filter. This makes it so that our filter will then listen for the user indicated RosBag topics, and also makes the filter aware of the name of all the frames of the system. Finally the filter then broadcasts its own topics, namely a topic relating to the correctly mapped pointcloud. The configuration needed in order for our system to work comes as follows.

- **Extended Kalman Filter Configuration:**
  - **Frame related initialisation:** Here we indicate to the EKF the frames of our UAV system.
    - map_frame: map; Defaults to "map"if unspecified

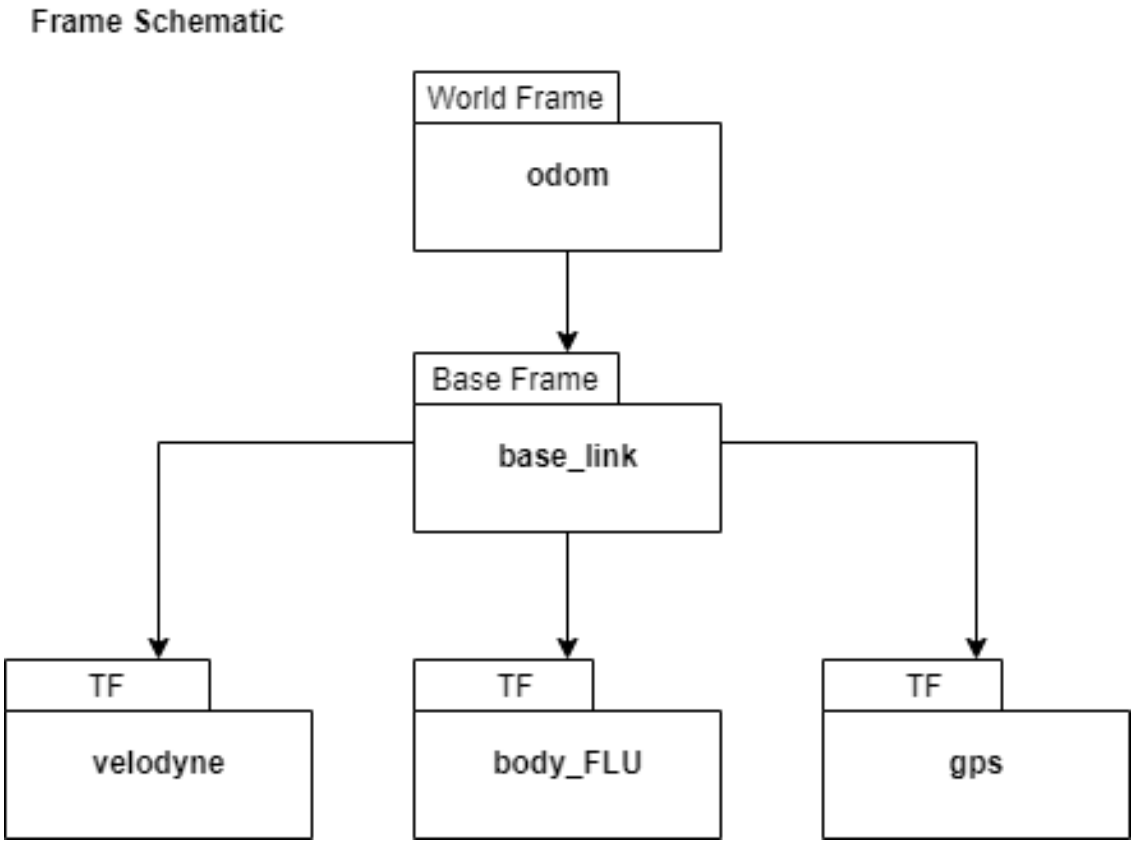


Figure 4.3: The Frame schematic. The world frame is connected to the odom data, base_-link relates with the UAV. The LiDAR pointcloud is connected to the Velodyne frame.

  - odom_frame: odom; Defaults to "odom"if unspecified
  - base_link_frame: base_link; Defaults to "base_link"if unspecified
  - world_frame: odom

- **Sensor information topic initialisation:** Here we indicate to the EKF which topics should he listen to, to obtain the data it needs.
  - odom0: odom; the topic name in the RosBag file transmitting odometry information
  - imu0: dji_sdk/imu; the topic in the RosBag file responsible with transmitting IMU data.

After correctly matching the topics and the UAV frames this package publishes a new PointCloud2 topic where the pointcloud appears as it should be. In order to visualise the RosBag pointcloud information we used RVIZ, which is a 3D visualiser for the Robot Operating System framework.

After correctly mapping and reconstructing the pointcloud information contained in the RosBag file, we can then use some specific tools that allow us to save that pointcloud information, be it all or certain chosen parts, under the file extension .pcd. This is most

necessary for the next stage of our work, parameter estimation and retrieval. It also serves as a bridge between the two main blocks of our system, the bag_file manipulation and the tree parameterization block, which deals primarily with PCD files.

4.1.1 Retrieving PCD-Files

In order to store the entirety of a pointCloud after its reconstruction, we used a PCL-ROS library Node purposefully built for this task. The pointcloud_to_pcd node subscribes to a ROS topic and saves point cloud messages to PCD files. Each message is saved to a separate file, names are composed of an optional prefix parameter, the ROS time of the message, and the .pcd extension. For instance in our particular case we'd subscribe to the reconstructed pointcloud topic. If this node was to be launched in the beginning of operation then we would have saved each pointCloud2 ROS message, and that would gives us an enormity of files, from start to finish of the point cloud reconstruction. What we did instead to solve this issue, was we only launched this node when the landscape reconstruction was done, that way we only get the last message sent by our reconstructed pointcloud topic, which corresponds to the entirety of the scene. File names look like 1285627014.833100319.pcd, the exact names depending on the message time stamps. [45]

The method above described only allows to extract the entirety of the scene. It doesn't allow the user to only retrieve certain parts of the scene. For this we found another way we found to retrieve .pcd files from our pointclouds. Using an RVIZ plugin /selected_points_publisher [46] that allows the user to select points from the pointcloud that he wants to save and publish them in a ROS pointcloud2 topic. Then using the above described pointcloud_to_pcd node we were able to save individual trees, from scenes where buildings where present for instance.

4.2 Tree height parameter

In this section we will approach the topic of how we implemented the block that deals with the estimation of tree height. In the process of development, we thought it necessary that we would part and distinguish individual tree setting from a forest section scenario. First, we built pieces of code and algorithm capable of dealing with an individual tree and only then, secondly, we took that same code, adapt it, so that it was capable of dealing with sections of forest tiles i.e. a grid square. In this section we will enter into detail as to the process of thought, as well as the reasoning and decision thought behind each implementation decision.

In this section we will mostly be dealing with manipulation of PCD files, and describing the operations needed and implemented in order to achieve the goal of estimating tree height, individually, and in group. We defined tree height as being the distance between base of an individual tree to the up most point in the z-axis.

4.2.1 Estimation of tree height for an individual tree

Tree height is one of the most used parameters in order to make estimates of the biomass weight for a given tree, that being said, we had to be able to obtain solid estimates that we could with the goal of biomass estimation. Fig. 4.4 shows the methods implemented as well as the resources used for their implementation.

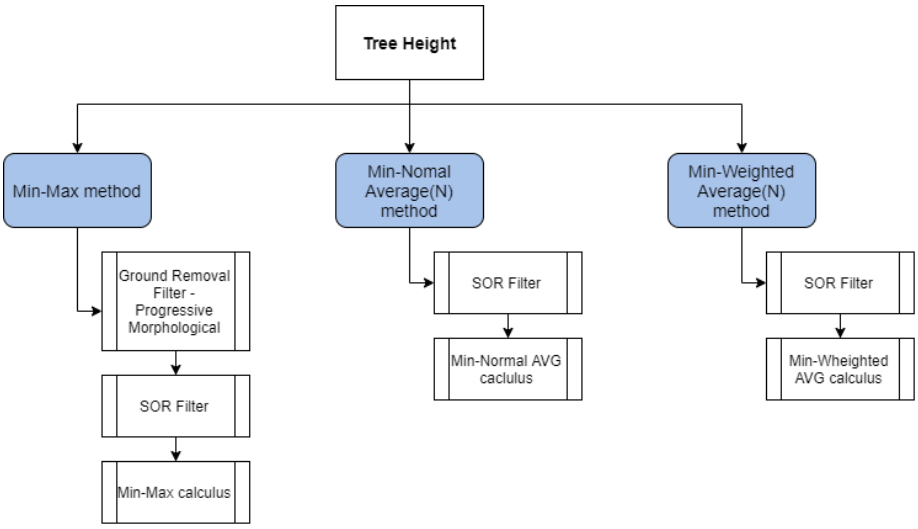


Figure 4.4: Individual Tree Height methods implementation.

The first challenge was, what should we consider the proper tree height and how should we implement the algorithm. Well firstly we decided, as we have stated previously, to consider individual tree height as being the distance between base of an individual tree to the up most point in the z-axis. This height is given by a method we designated Min-Max method, and this is the weakest method to estimate tree height, because in a pointcloud manipulation environment, this definition can lead us to some curious phenomenons.

For instance, if the PCD file for an individual tree presents some points at the canopy top that are clear outliers, this may give way to untruthful measurements or over estimates. This issue can be partially solved, by the implementation of a Statistical Outlier Removal (SOR) filter[47] that works the following way.

The Statistical Outlier Removal filter uses point neighbourhood statistics to filter outlier data. The algorithm iterates through the entire input cloud twice: The mean and standard deviation of the distance between each point and it's K-nearest neighbours are computed in order to determine a distance threshold. During the next iteration the points will be classified as inlier or outlier if their average neighbour distance is below or above this threshold respectively. Below are listed the parameters that we can change in regards to this filter.

- **Statistical Outlier Removal filter configuration:**

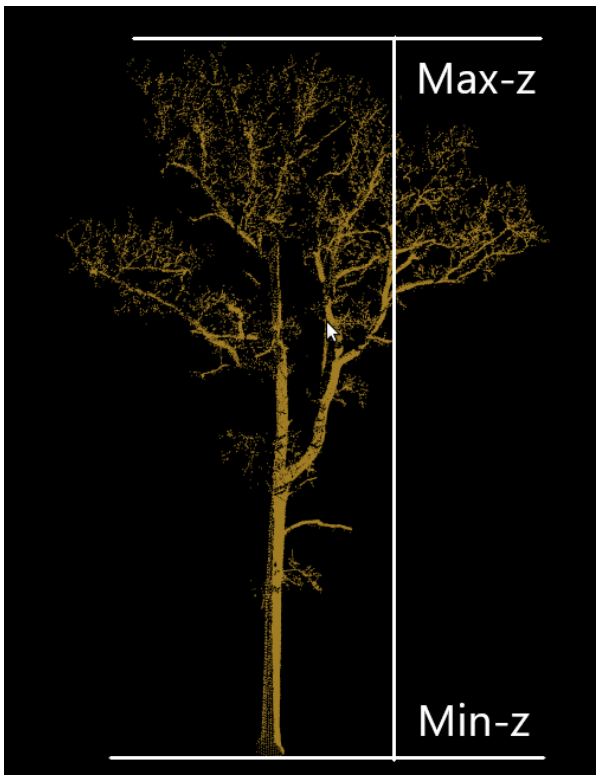


Figure 4.5: Min-Max tree height estimation method.

- setMeanK: Allows to set the number of neighbours taken into account when computing the average distance that each point has to its nearest neighbours.
- setStddevMulThreshold: : Sets the standard deviation multiplier, used to calculate the distance threshold, which in turn is calculated using the following formula: mean + stddevMult x stddev.

That being said, suppose a tree presents a branch that abnormally surges vertically, disrupting an otherwise regular tree canopy, it is our understanding that such branch should count in order to estimate the tree height. Of course this can lead down the road to an overestimation of a given tree biomass. Those considerations must be weighted in and properly evaluated, as the SOR filter can only help reduce this problem, before taking a decision about not considering such a tree particularity for tree height.

Parameter	Value
setMeanK	25
setStddevMulThresh	1.0

Table 4.1: Values used in the SOR filter.

If a PCD file representing an individual tree comes with the ground surface depicted in the point cloud, in other words the soil upon that given tree sits, and that ground surface depicts a slight elevation where upon the tree is atop, then, simply retrieving the

two points, one with the highest z-value and the other with the lowest z-value is simply not enough to get a proper estimation of a tree height as we would get a value that would be slightly overestimated, as the lowest z point would belong to the ground surface and not to the tree itself. We fix this by implementing a progressive morphological filter capable of estimating the points that belong to the ground surface, and those that do not belong to the ground surface. Usually points belonging to the ground surface belong to the last returning points in a pointcloud. Below are the parameter that we can change in regards to said filter.

- **Progressive Morphological Filter configuration:**
  - setMaxWindowSize: Set the maximum window size to be used in filtering ground returns.
  - setSlope: Set the slope value to be used in computing the height threshold.
  - setMaxDistance: Set the maximum height above the parameterized ground surface to be considered a ground return.
  - setInitialDistance: Set the initial height above the parameterized ground surface to be considered a ground return.
  - setCellSize: Set the cell size.
  - setBase: Set the base to be used in computing progressive window sizes.
  - setExponential: Set flag indicating whether or not to exponentially grow window sizes.

We can use this filter with each individual tree or, as we shall see further ahead, we can use it to larger extents of forest surface with solid results. Below we leave the values used for the progressive morphological filter that proved capable when filtering out the ground surface.

Parameter	Value
setMaxWindowSize	20
setSlope	1
setInitialDistance	0.5

Table 4.2: Values used in the morphological filter.

The Min-Max implementation to estimate tree height presents itself as being easy to fool, if some points of the ground surface remain in the pointcloud even after filtering, and computer resources exhaustive. As such we decided to implement other two slightly different approaches that needed no ground surface removal. In order to estimate tree height for an individual tree and in the spirit of tackling some of the issues appointed beforehand, instead of just calculating the two points which in the point cloud has the

highest and the lowest z-value, we decided to calculate the average value of the N points with the highest z-value.

We estimated heights using two different weight mechanisms, first we used a normal Arithmetic Average, where each point has the same weight and then we did a Weighted Arithmetic Average using the Mengoli series as weights, except for the last point to be weighted in. For the last point to be weighted in we use the weight $1 - \sum_{n=1}^{N-1} \frac{1}{n(n+1)}$, this is so that the total sum of the weights is equal to one. We use the Mengoli series, seen in equation 4.1, so that the point with the highest z-value counts more than the others.

$$\sum_{n=1}^{\infty} \frac{1}{n(n+1)} = 1$$

(4.1)

For instance, lets say we have a dataset composed by the natural numbers up until ten. Let say we use a N equal to 4, this means we will use the 4 highest values to obtain our average, that is, 10, 9, 8 and 7.

$$NormalAverage: \frac{10+9+8+7}{4} = 8.5 \tag{4.2}$$

$$WeightedAverage: 10 * \frac{1}{2} + 9 * \frac{1}{6} + 8 * \frac{1}{12} + 7 * \frac{3}{12} \approx 8.92 \tag{4.3}$$

We decided to use average point value as it gives us the ability to take into account the peculiarities that the top of a canopy can have, blending out the values of the N-points with the highest z-value allows for a safer and more confident estimation of the height of a tree that sits almost vertically. These algorithms where implemented for individual trees, as such, in order to use them to estimate heights in sections of forest, in a square of the grid, we need to make some decisions as well as some compromises as dealing with sections of forest gives way to a greater amount of uncertainty and complexity.

4.2.2 Estimation of medium tree height for a grid square

When we approach the topic of estimation of tree height values for sections of forest, multiple questions and problems tend to arise. Individual tree height estimation presents several challenges, namely and number one, building an algorithm capable of segmenting individual trees from sometimes complex sections of forest. In order to develop our approach we had a Pointcloud file consisting of a forest section of 20m by 40m.

Firstly, we subdivide our forest section into smaller portions that proved easier to deal with, this is what we call our grid approach. This grid approach allows us to analyse smaller pieces of forest section and work with them applying simpler methods. The grid square size is up to the user to decide, and further ahead we will be able to see that this has an impact on the results obtained.

Figure 4.6: Normal Average individual tree height estimation. All N points are given the same weight.

There are some challenges when presented by this grid system approach. The first one being that if the forest is very dense and the pointcloud density is rather low, the LiDAR scan may not be able to capture enough ground points, disabling proper height estimations. Another problem when working with forest sections as a whole and with grid squares in particular, has to do with ground surface slope. If the terrain is too steep this certainly has an effect over tree height section of forest. This particular problem can be aggravated by the chosen grid side size, if it is for instance 5m, this can lead (if the incline is lets say 6%) to an increase or decrease of the height measured for that grid square of 30cm at worse.

We dealt with the slope problem firstly filtering out the ground surface, this leaves us with a pointcloud consisting of the individual trees and some points that were not filtered out by the Morphological Removal Filter, and afterwards we calculate an average of the N points with the lowest z-axis value to obtain a grid tree base value. We also apply two different average methods, a normal Arithmetic Average and a Weighted Arithmetic Average using the Mengoli series as weights where the point with the highest z-value gets the highest weight. We did this in order to compensate for incline. We would like to underline that N should not amount to more than 2.5% of the total number of points of the pointcloud at hand, otherwise, in very dense pointclouds we would end up using way to may points for tree height estimation which would give very poor results.

After estimating the lower z-axis value we then followed to estimate the higher Z-axis

Figure 4.7: Weighted Average individual tree height estimation. Points higher up in the tree canopy are given a bigger weight.

values for the grid square. We did this three different ways, simply getting the point in the point cloud with the highest point, then a normal Arithmetic Average with N points and then a Weighted Arithmetic Average using the Mengoli with the point with the highest z_value given the highest weight.

Our approach does not take into account individual trees instead it looks at portions of forest sections at a time and analyses each section as a whole, estimating a height value utilising the points of that square for each square as it goes. But this leads us to a problem, what if in a particular grid square we don't have any tree? Imagine the following: we have a section of forest we are analysing, 2m by 2m, ground surface points still remain as we chose not to filter them away, and other points consisting of dead tree trunks sitting in the forest ground also still remain. There is no tree in that grid square, but still you get a reading of a tree height with a value high enough that cannot be ignored.

So far, from what as been stated this is a credible scenario, and in fact it did occur while developing our solution. This can happen if a tree canopy belonging to a tree on a adjacent square, is partly present in the square up for analysis. This lead us to develop the simplest trunk detection algorithm described further ahead in 4.4.1. It is so much a trunk detection algorithm as it allow us to know if there is a tree or any tree trunks in a given grid square. Knowing this allows us to ignores the squares where there is no presence of tree trunks giving us better results all around. In short we where able to develop an algorithm capable of differentiating grid squares without and with trees, and

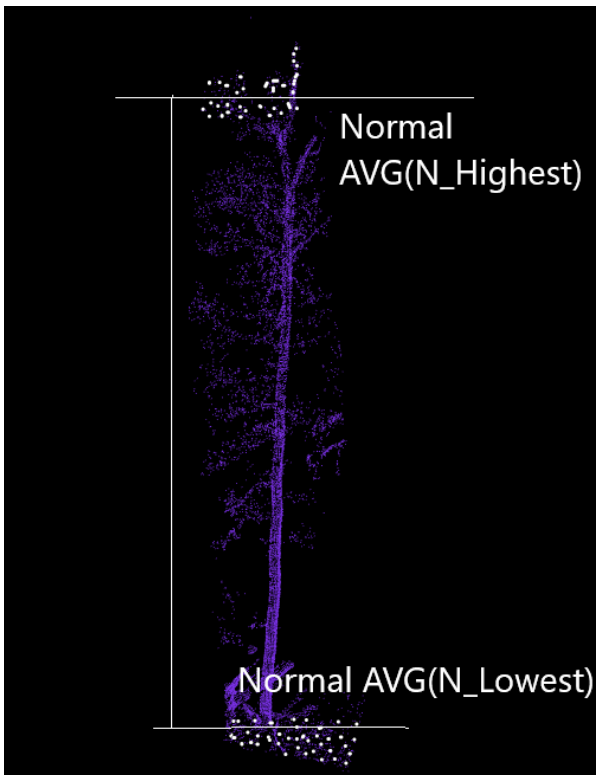


Figure 4.8: Grid square height estimation on a 4m by 4m grid square.

for those with tree estimate an average tree height associated with that grid square.

4.3 Diameter at breast height parameter

In this section we will approach the topic of how we implemented block that deals with the estimation of the diameter at breast height parameter. This is a key parameter in order to estimate tree biomass for a given tree, and we can say it was rather challenging to come up with ways to estimate it in a solid way. This section will place it's focus in the process of development as well as the decision making aspect related to the solution found.

In this section we will mostly be dealing with manipulation of PCD files, and describing the operations needed and implemented in order to achieve the middle goal of estimating diameter at breast height, this time only individually as we were not able to find in due time a solution solid enough that could be presented capable of estimating diameters at breast height for a section of forest with possible multiple trees. DBH is defined as being the diameter at 1.34 meters from the base of an individual tree.

4.3.1 Estimation of diameter at breast height for an individual tree

We have developed two different methods, see fig. 4.9, in order to estimate the DBH for an individual tree. First a method that uses the XY-Diference average between points, a

more classic approach to the problem at hand, a second method was achieved utilising Model Fitting approach to a target point cloud, that had been obtained from the original tree point cloud.

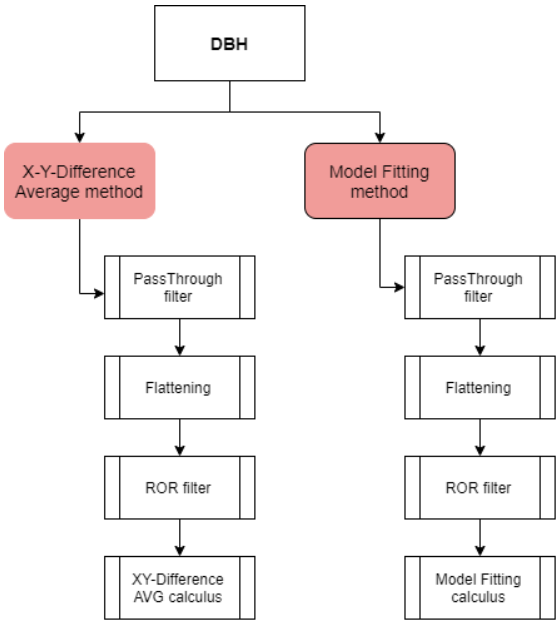


Figure 4.9: DBH methods implementation.

XY-Diference average method retrieves from the given pointcloud the maximum X-axis difference and the maximum Y-axis difference between points, and averages both values in order to estimate DBH. In order to work properly, it is fundamental that only the points belonging to the tree trunk remain. The Model fitting approach analyses a given pointcloud and tries to search for any similarities in the pointcloud to the model that the user indicates he's looking for, in our case circular shapes. If any pointcloud points form a pattern similar to a circumference, a model of a circumference is built and the algorithm gives the user the center X-Y coordinates of the circumference model, and the radius of said circumference.

Both X-Y-Difference-average method and Model fitting approach require the same previous pointcloud manipulation before being used to estimate DBH . In short, we will now follow with the idea and reasoning behind these solutions and its implementation. We first need to filter the pointcloud representing one individual tree as we are only interested in a particular section between the height of Z = 1,0m and Z= 1,37 m. In order to get an auxiliary pointcloud from the original one, we must utilise a Passthrough filter as implemented by the pclRos library.

This Passthrough filter allows us to iterate through the entire input cloud at once and automatically filter non-finite points and the points outside the interval specified by us, which applies only to the user specified axis, in our case, the z axis. this filter has the following parameters that can be adjusted:

- **Pass through filter configuration[48] :**
  - setFilterFieldName: Provide the name of the field to be used for filtering data. X axis, Y axis or Z axis.
  - setFilterLimits: Set the numerical limits for the field for filtering data.
  - setFilterLimitsNegative:Set to true if we want to return the data outside the interval specified by setFilterLimits (min, max) Default: false.

Parameter	Value
setFilterFieldName	Z
setFilterLimits	[1m ;1.37m]
setFilterLimitsNegative	false

Table 4.3: Values used for the DBH Passthrough filter.

After the filtering process all that remains are the points that obey z belonging to [1m , 1.37m], instead of only keeping a slice of the original tree pointcloud, see fig.4.11, we decided to keep a whole section of the tree. This has a reason behind it. Sometimes pointclouds may not have a consistent point density, that way, filtering and keeping points belonging to a certain interval by opposition to simply keeping a slice of that original point cloud, ensures us that we still have sufficient points that allow for a more solid estimate.

After applying the Passthrough filter, the point cloud enters a process we designated as flattening. Here we take all the points belonging to the pointcloud and attribute them the same z_quota, see fig 4.12, this allows us to concentrate our points in a simple planar surface which will prove useful further ahead. This flattening also serves the purpose of simulating a sectional slice of the tree, allowing us to calculate DBH estimates.

After the flattening process we follow with a filtering process, using a ror filter. The user specifies a number of neighbours which every point must have within a specified radius to remain in the PointCloud.[49] This filter is intended to remove points that don't belong to the tree trunk, but belong instead to the tree canopy. If such points are not filtered away and remain int the point cloud, results related to the X-Y-Difference-Average method will be severely affected.

After the needed point cloud manipulation we can proceed to use our XY-Diference average algorithm for DBH estimation. This method is particularly good and presents

Parameter	Value
setRadiusSearch	Variable(0.1m and 0.5m)
setMinNeighborsInRadius	6

Table 4.4: Values used for the DBH ROR filter.

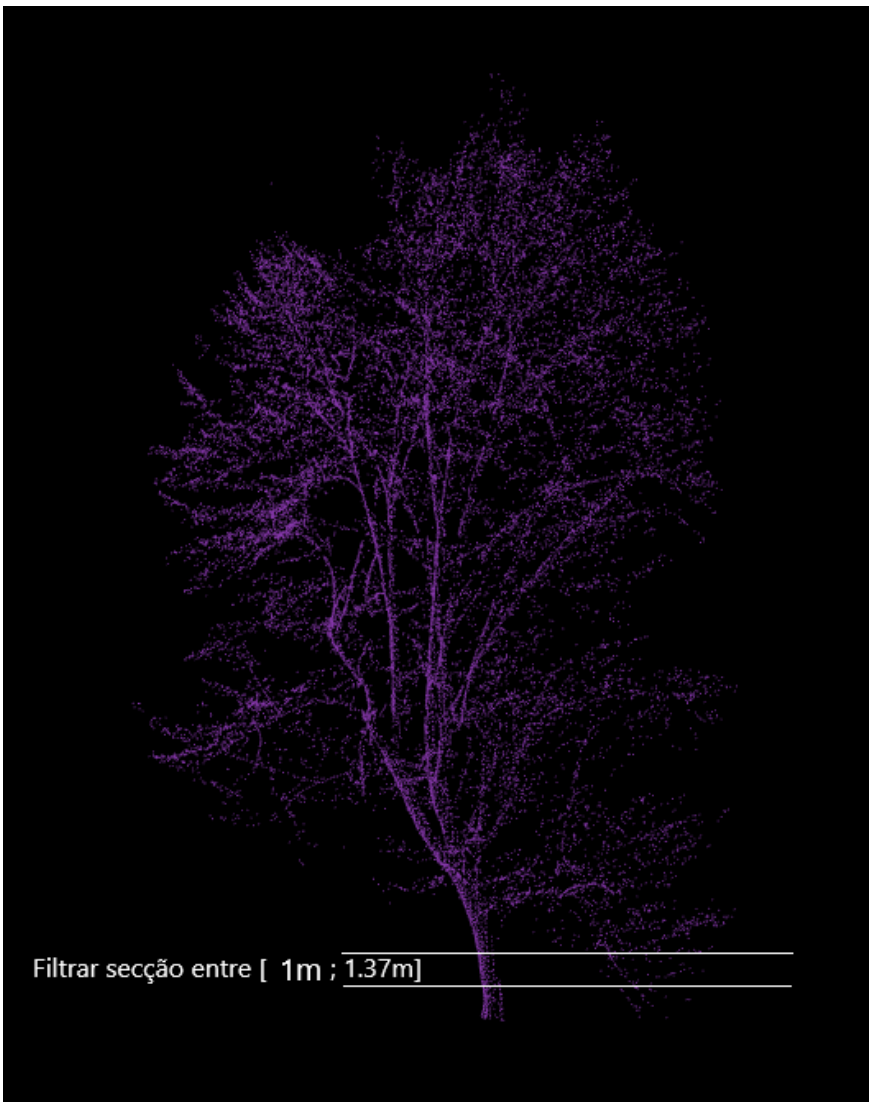


Figure 4.10: DBH stage one, filtering.

very solid results when it comes to individual tree DBH estimation if no outlier points remain after filtration.

Of course there are some draw back to going with this X-Y-Difference approach, the first one being that it is time consuming to run through a given point cloud of N size, has the algorithms used to find the lowest and highest x an y values do. Has such, if a point cloud is very dense, this method may consume more computer resources and time. This would be a very severe draw back if we where to implement these algorithms in real time, but that is not the case with this dissertation. A second drawback relates to the needed precision in the ROR filtering stage, as if any points not belonging to the tree trunk remain, this algorithm is easily fooled and the results become compromised.

A third drawback, and this one is the most damming, - if we take into account what we are trying to achieve in the end, that is, a solid method that can estimate DBH for small to medium size sections of forest, that is squares of up to 4m by 4m - is that this

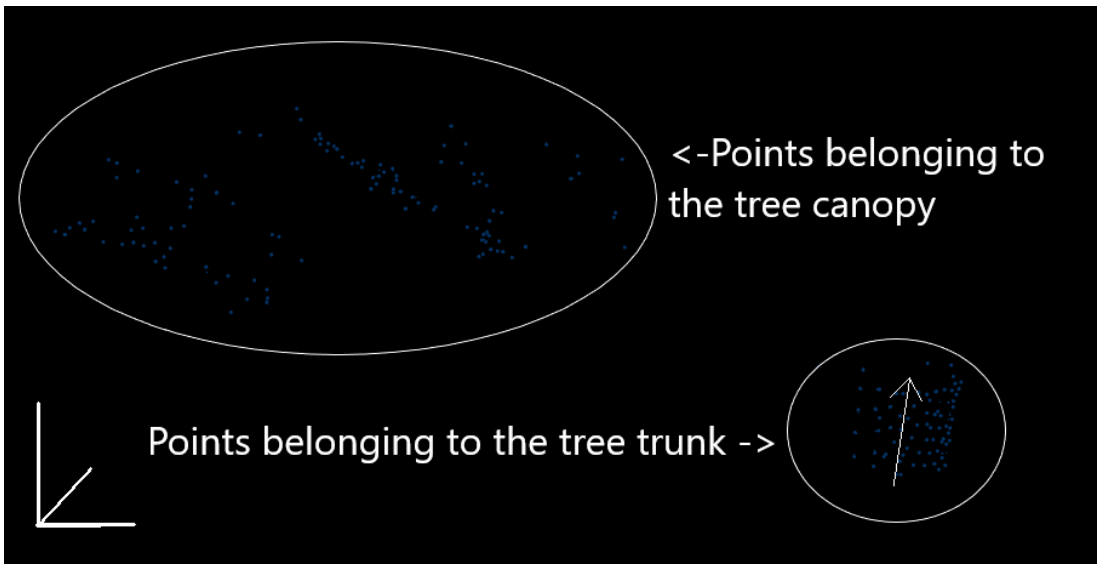


Figure 4.11: DBH, pointcloud after Passthrough process.

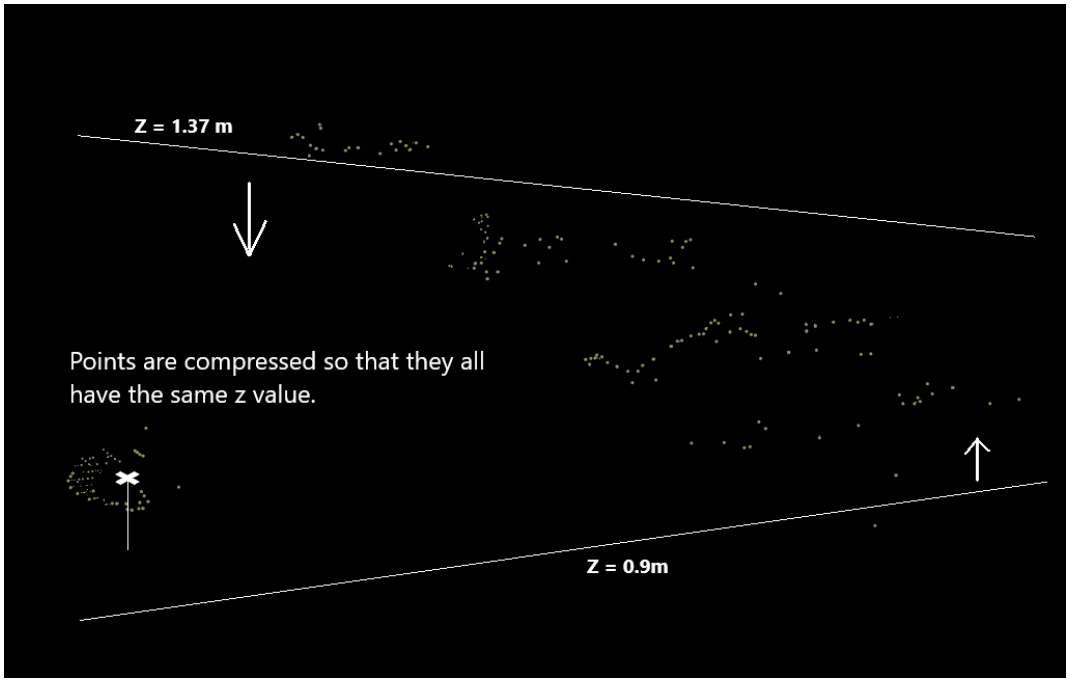


Figure 4.12: DBH, pointcloud after flattening process.

method proves to be simply to hard to scale up. With a forest section you may get several tree trunks into a simple square, that being said, if you where to filter the pointcloud in order to get the height section needed to calculate DBH, and right there after, you where to calculate the XY-Distance average between points, what you would get would be the largest distance between different tree trunk clusters and not the diameter of each tree trunk cluster.

Even though we were not able to implement a method capable of estimating DBH for trees belonging to a section of forest, we can still suggest a possible solution to the

Figure 4.13: DBH, pointcloud after ROR filtering process. Radius Search(0.1m)

problem previously indicated, whilst still using the same concept. We suggest that after filtering the section of forest that you need to make your calculation you try to cluster the different sections of tree trunks together and only then apply this method of calculus to each individual cluster. This should get you a solid estimate for DBH for each tree present in that particular forest section square.

Our second method for individual tree DBH estimation was able to estimate DBH with consistent results. As said before this second method uses a model fitting approach in order to obtain a diameter for a given tree. The pre-processing stage has been described before, and remains the same for this approach: 1º Passthrough Filter. 2º Flattening. 3º ROR filtering. After levelling every point to the same z-value quota, you will notice that the tree trunk resembles a circular shape as seen from above, as such we used Ransac SampleConsensusModelCircle2D which defines a model for 2D circle segmentation on the X-Y plane.

The model coefficients are defined as [50]:

- center.x : the Xcoordinate of the circle's center
- center.y : the Y coordinate of the circle's center
- radius : the circle's radius

Parameter	Value
setModelType	pcl::SACMODEL_CIRCLE2D
setMethodType	pcl::SAC_RANSAC
setDistanceThreshold	Variable(0.03m and 0.05m)
setMaxIterations	100
setRadiusLimits	[0.2m ; 0.5m]

Table 4.5: Values used for the RANSAC model fitting algorithm.

After model fitting you are left with the radius of the circumference model, one computation away from diameter. A model fitting approach presents some advantages such as being fully implemented into ROS through the PCL-ROS library, this guarantees that everything is as well optimised as can be, but perhaps the main advantage may lay in the scalabilility of the method if and when applied to entire sections of forest.

It presents some downs sides to, for instance, if the point cloud for an individual tree is not dense enough it may be that the model fitting cannot find any resemblance to a circle model, as such for less dense pointclouds, this method is not indicated, as it can make way for some blank results where it isn't capable of retrieving any result.

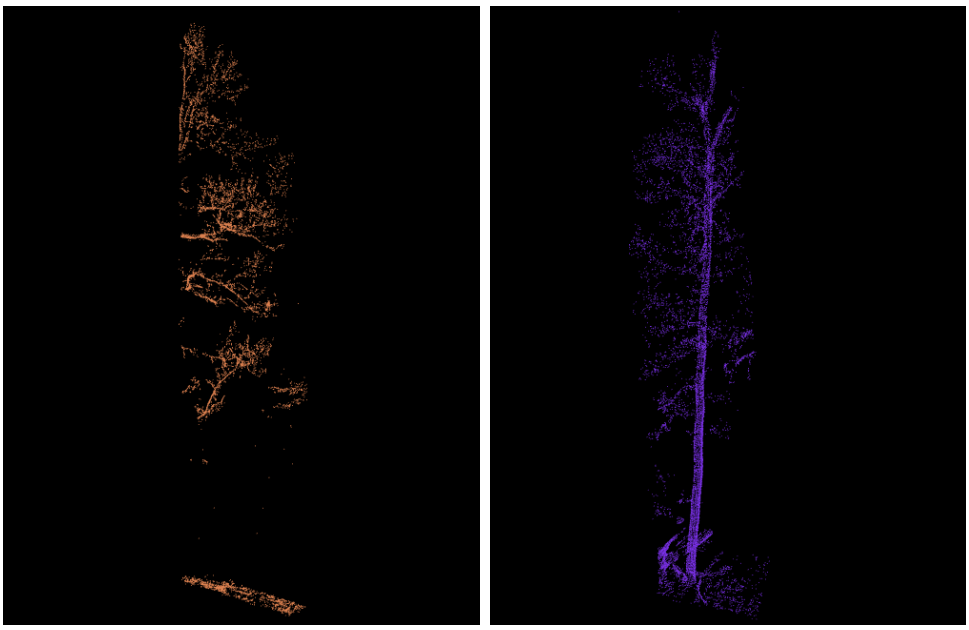
4.4 Tree Trunk Detection in grid squares

We have developed a method capable of detecting and counting tree trunks in a given section of forest. This comes as the middle step for DBH estimation in trees belonging to a section of forest, understandably it is most fundamental to single out each individual trunk in order to then estimate DBH. In this section we will be dealing with PCD files containing forest information, instead of individual trees.

4.4.1 Density Index

The first method implemented in order to detect the presence of tree trunks, based itself in the idea that if a tree trunk was present, then points representing that tree trunk would be present as well and as such we could count them and compare with the total number of tree points. This may prove useful in certain situations, for instance, when you slice a PCD file (pointcloud file) into several squares and the first thing you estimate for all forest squares is the Tree Height parameter you will end up with several squares that contain a height value that is not relevant, simply because there is no tree in such square. It might happen that, for some squares, tree height is estimated using ground surface points and points belonging to the canopy, see fig.4.14(a). This is not desired, and as such more work is needed to single out such instances where such problem occurs and eliminate them.

The Density index works by slicing horizontally the pointcloud at study along the z axis into sections of equal height - we used the Z-Min-Max value divided by 10 - and we calculate the density - the number of points that belong to that given section divided by the total number of points. With that density calculus done we can then say, looking to the second and third floors, I2 and I3 - if a point cloud has a z Min-Max difference of 10m, we will be looking at sections between 1m to 2m and also 2m to 3m -, if a tree trunk is present in that specific forest section or not simply. This is done by analysing the percentage of points present in those sections to the dispersion of the remaining points in the pointcloud. In order to find a suitable value as our threshold we ran our algorithm in individual trees to study point dispersion. We found out that using a threshold of 5%



(a) An invalid grid square with no tree trunk present coming from the ground. In order to form our tree height grid, these are the kind of squares we want to clear out.

(b) A valid grid square with a clear tree trunk present coming from the ground. In order to form our tree height grid, these are the kind of squares we want to count in.

Figure 4.14: The importance of Trunk detection for grid square height estimation.

of points as proven to be a solid value to detect individual tree trunks, and as such this is the value adopted when dealing with grid squares.

This method has it's limitations of course. As said earlier in the beginning of this section, first one being that for larger sections of forest this criteria for detecting the presence of a tree trunk simply does not hold.

The first problem is that for larger sections of forest area there may be that several tree trunk that need to be accounted. This method is simply not capable of detecting and account for individual tree trunks in a section of forest. Second problem faced is that for terrains with a high slope, simply sectioning along the z-axis and and calculating density may land highly inadequate results.

Thirdly and also based on the two points raised before, this approach would prove less useful if the goal is singling out those points belonging to tree trunks, in a given forest section, in order to estimate DBH, for that another approach is necessary.

Although proved able to indicate for individual trees and small sections of forest if a tree trunk is there or not, with its many faults Density calculation is an unsuitable

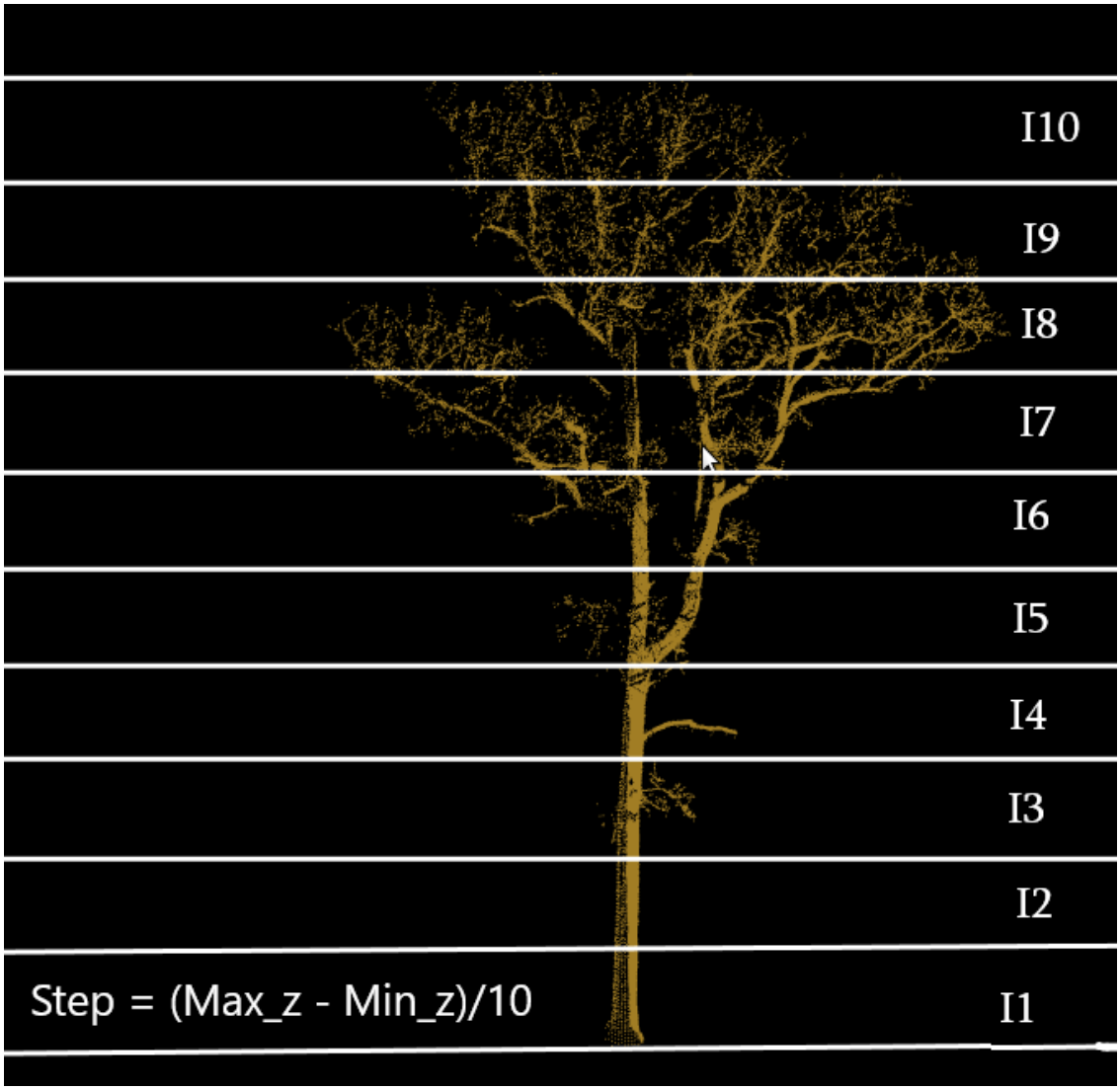


Figure 4.15: The Density Index concept.

approach for tree trunk detection in a section of forest if furthermore we want to then estimate DBH for that given tree trunk. We found this index most suitable applied in context of individual tree analysis, in figuring out the distribution of a tree, estimation of canopy base height and canopy shape and distribution, but these points do not fall under the theme of this thesis.

4.4.2 Region Growing approach

In order to be able to detect how many trees where in a given section of forest and at the same time segment those tree trunk section in order to single them out for further work ahead, one considered that a Region growing base algorithm would be most favourable for achieving the best results regarding detection of tree trunks in a grid square. First we need to apply a Passthrough filter, see tab.4.6, in order to select only the points that are of interest to us, those corresponding to the tree trunk area.

Parameter	Value
setFilterFieldName	Z
setFilterLimits	[0.9m ; 2m]

Table 4.6: Values used for the Passthrough filter needed to select the poinst of interest (trunk section) for the Region growing algorithm.

First of all it sorts the points by their curvature value. It needs to be done because the region begins its growth from the point that has the minimum curvature value. The reason for this is that the point with the minimum curvature is located in the flat area (growth from the flattest area allows to reduce the total number of segments). So one has a sorted cloud. Until there are no unlabelled points in the cloud, the algorithm picks up the point with minimum curvature value and starts the growth of the region [51]. This process occurs as follows:

1. The picked point is added to the set called seeds.
2. For every seed point, the algorithm finds its neighbouring points.
3. Every neighbour is tested for the angle between its normal and normal of the current seed point. If the angle is less than the threshold value then current point is added to the current region.
4. After that every neighbour is tested for the curvature value. If the curvature is less than the threshold value then this point is added to the seeds. Current seed
5. Current seed is removed from the seeds.
6. If the seeds set becomes empty this means that the algorithm has grown the region and the process is repeated from the beginning. [51]

As one can see the region growing algorithm suits well the need at hand. We were in need of a method capable of identifying and segmenting pointcloud points belonging to the tree trunk, so that, further ahead one can estimate a DBH for said tree trunk. For grid squares, this methods looks to be the most promising in achieving the goal of segmenting the different tree trunks, present in a same grid square. Below follows the parameters that one can change to set the region growing algorithm to our needs. In the testing stage of our Region growing algorithm two sets of values where tested, the values used for both runs are depicted in tabs. 4.7 and 4.8. In the second run, criteria for region growing is more tightened as to allow the algorithm a better chance at only selecting points belonging to the tree trunk, as such smoothness threshold was lowered to 12 degrees, meaning that points with a normal deviation of more than 12 degrees are not added to the same region.

- **Region Growing segmentation algorithm configuration:**

- setCurvatureTestFlag: Allows to turn on/off the curvature test. Note that at least one test (residual or curvature) must be turned on. If you are turning curvature test off then residual test will be turned on automatically.
- setCurvatureThreshold : Allows to set curvature threshold used for testing the points.
- setInputNormals: This method sets the normals. They are needed for the algorithm, so if no normals will be set, the algorithm would not be able to segment the points.
- setMaxClusterSize :Set the maximum number of points that a cluster needs to contain in order to be considered valid.
- setMinClusterSize:Set the minimum number of points that a cluster needs to contain in order to be considered valid.
- setInputNormals: This method sets the normals. They are needed for the algorithm, so if no normals will be set, the algorithm would not be able to segment the points.
- setNumberOfNeighbours: Allows to set the number of neighbours. For more information check the article.
- setResidualTestFlag: Allows to turn on/off the residual test. Note that at least one test (residual or curvature) must be turned on. If you are turning residual test off then curvature test will be turned on automatically.
- setResidualThreshold: Allows to set residual threshold used for testing the points.
- setSearchMethod: Allows to set search method that will be used for finding K-Nearest Neighbours.
- setSmoothModeFlag: This function allows to turn on/off the smoothness constraint.
- setSmoothModeThreshold: Allows to set smoothness threshold used for testing the points.

Parameter	Value
setMinClusterSize	40
setMaxClusterSize	1000
setSearchMethod	tree
setNumberOfNeighbours	8
setSmoothnessThreshold	(25.0 / 180.0 * M_PI) rad
setCurvatureThreshold	1.25

Table 4.7: Values used for the Region growing algorithm used for tree trunk detection, run 1.

Parameter	Value
setMinClusterSize	60
setMaxClusterSize	1000
setSearchMethod	tree
setNumberOfNeighbours	15
setSmoothnessThreshold	(12.0 / 180.0 * M_PI) rad
setCurvatureThreshold	1

Table 4.8: Values used for the Region growing algorithm used for tree trunk detection, run 2.

Chapter 5

Experimental Results

In this chapter we discuss the experimental results, and our findings. We will also try to explain results that do not fall in line to those that were expected, pointing to possible solutions for those that may follow. The discussion of experimental results is of upmost importance, as we need to critically evaluate if they make sense or not, and seeks to evaluate the repeatability of these results.

There were two sets of data used in this dissertation, first some field collected by the Velodyne LiDAR unit aboard RICS , and a dataset consisting of 26 individual trees, the great majority beech trees - *Fagus sylvatica*, see fig.5.1 - and a forest section of 20m by 40m made of those same individual trees. The data set can be found here [52] and was made available by Trochta et. al[53]. *Fagus sylvatica* is a large tree, capable of reaching heights of up to 50 m tall and 3 m trunk diameter, though more typically specimens stand at 25–35 m tall and up to 1.5 m in trunk diameter. [54]

One difficulty found was due to the fact that we did not have access to a plot of land of our own from where we could obtain our data and then validate our results for each individual tree and for the forest section as a whole. We certainly do believe this hinders the possible maximum outcome of this dissertation and as such is a point of improvement for future work.

Despite that, we can still compare our results between each other, using certain parameters as reference values for each individual tree calculated by using the available point cloud data. That way we can then validate our results, and check if the approach we took for each parameter makes sense, and outputs solid and sounding results. The machine used to obtain this results is equipped with an Intel(R) Core(TM) i7-5500U CPU @2.4Ghz chip with 8GB of RAM Memory running Ubuntu 18.04 LTS and ROS.

Figure 5.1: European Beech (*Fagus sylvatica*) in the Cansiglio Forest (Italy), June 2017. [55]

5.1 Data Gathering

The first data set was composed mainly of bags files and those were used to ascertain a way of how we could capture forest data and later reproduce it in a way useful if we seek to study those forest systems, be it estimate biomass or other fields of study. These first stage results, that is, bags manipulation and pointcloud reconstruction, are what we seek to show here. The LiDAR data obtained ranges from field experiments in the district of Castelo Branco, Portugal, to one LiDAR data acquisition of a section of our university campus. Much of the data was unusable for the later stage of this work, as it either lacked point density, or focus on the matter at study, for instance, one of the pointclouds, even though in a forest revolved mainly around a power-line cable.

5.2 Individual Tree Results

5.2.1 Tree Height Results

Relating to individual tree height estimations, work was done utilising the individual tree data set, found online. The algorithm was executed four times using different parameters in regards to the number of points used in order to estimate an average and in regards to doing those calculation with and without ground surface. We regard the control data as being the one presented in table 5.1, under Min-Max without floor. These numbers reflect a tree height estimation obtained from the subtraction of the highest z-value point in regards to the lowest z-value point, with the ground surface previously removed using the morphological ground removal filter.

Tree	Min-Max w/ ground surface	Min-Max w/out ground surface	Dif.
1	20,424	19,84	0,58
2	18,328	17,777	0,55
3	18,709	18,15	0,56
4	16,074	15,57	0,5
5	8,16602	7,633	0,53
6	20,783	19,853	0,93
7	17,009	16,49	0,52
8	19,848	19,317	0,53
9	17,781	17,203	0,58
10	20,2	19,7	0,5
11	17,974	17,473	0,5
12	20,536	19,951	0,59
13	25,185	24,485	0,7
14	20,727	20,215	0,51
15	20,989	20,486	0,5
16	16,783	16,214	0,57
17	19,685	19,144	0,54
18	22,65	22,089	0,56
19	18,545	18,025	0,52
20	21,356	20,5	0,86
21	23,613	23,096	0,52
22	12,164	11,019	1,15
23	16,761	16,216	0,54
24	21,276	20,679	0,6
25	24,505	23,772	0,73
26	22,023	21,431	0,59
		On average:	0,61

Table 5.1: Reference values for each individual tree (values in m). Table regarding height estimation values for individual trees, using the Min-Max method. The values on the right collum were calculated without ground surface and for that reason are the closest to reality.

As we can see from the data contained in table 5.1, there is a real danger of overestimating individual tree height if ground surface is taken into account, this has to do with slope of the terrain upon which the individual tree settles. That overestimation, when making individual tree height calculations, not removing ground surface, amounts, on average, to 61cm, see 5.1. As it has been explained in the previous chapter, methods were implemented in order to deal with such constraints. Results show that average methods are able to deal with ground surface slopes interference in the end result in a way that a simple min max methods cannot, that is of course if we don't remove the ground surface.

One may ask the following, if removing the ground surface and then following calculating Min-Max tree height value can serve as a reference, then why not stick with these results instead of doing further calculations. The answer has to do with time consumption, as morphological ground removal filter is very time consuming, see tab.5.2, but it has to do as well as with the focus of this dissertation, we are aiming to validate averaging methods to estimate tree heights not only for individual trees but as well, further ahead, for grid squares, depicting a forest section.

	Execution times (ms)
Min-Max w/out g.surf.	94665,4
N(50) w/ g.surf.	76,8238
N(100) w/ g.surf.	77,6015
N(2.5%) w/ g.surf.	79,6438

Table 5.2: Execution time for each set of algorithms (ms)

As we can see from tab. 5.2, morphological ground removal consumes a lot of time, of course it depends on the machine onto which you are running these algorithms, nevertheless one thing will remain , that is, the relative relations between said values, and a certainty that in fact doing height estimations with ground surface takes substantial fewer time than if otherwise.

What follows in the next set of pages is a series of result tables that depict results when estimating tree height values for individual trees, where the only variable changing is the number of points taken into account for the average calculus. These tables aim to show results accounting to simple Min-Max method with ground surface removal, that is, the values we took as reference for each individual tree height, and to the right, the remaining two columns show the average methods results.

Table regarding height estimation values for individual trees, using the Min-Max method. The values on the right column were calculated without ground surface and for that reason are the closest to reality.

First thing we can comment is that, as predicted the Min-Weighted Average method puts out more optimistic estimations than the Min-Normal Average method, as it takes into higher account the points with the highest z-values. Another thing we can clearly confirm is that the individual tree dataset is composed of trees that are more or less

Tree	Min-Max w/out g. surf.	MIN-Normal Average	MIN-weighted Average
1	19,84	19,706	19,918
2	17,777	17,16	17,866
3	18,15	17,852	18,374
4	15,57	15,022	15,329
5	7,633	7,17001	7,84302
6	19,853	19,591	20,057
7	16,49	16,326	16,354
8	19,317	18,707	19,306
9	17,203	17,477	17,448
10	19,7	18,79	19,41
11	17,473	17,622	17,4
12	19,951	19,258	19,725
13	24,485	24,614	24,64
14	20,215	20,242	20,187
15	20,486	19,712	19,652
16	16,214	15,341	15,928
17	19,144	18	18,209
18	22,089	21,304	22,055
19	18,025	17,812	17,834
20	20,5	19,567	20,362
21	23,096	22,185	22,842
22	11,019	11,264	11,175
23	16,216	15,751	15,608
24	20,679	19,634	20,341
25	23,772	23,214	23,904
26	21,431	20,787	21,354

Table 5.3: Individual tree height N= 50(values in m)

uniform, that is, trees that do not present preponderant and profile cutting characteristics such as branches that tear the normal profile of the individual tree. We can also comment that our SOR filter was also capable of eliminating any outlier points present in the point cloud. These conclusions come from the fact that the Min-Max values are not abnormally far away from the averaging methods.

Another aspect that we can comment is that very few exceptions are found of cases were the estimated average value for tree height surpasses that of the min max value. This is nonetheless more frequent in the case where the number of points taken into account amounts to fifty (N = 50, tab. 5.3). This is due to the fact that the more points you take into account in order to estimate a tree height value, the further down you go down the tree pointcloud, and therefore the lower will be the individual tree height value estimation.

With N=50, see tab. 5.3, we obtain solid results in relation to the reference values taken for each individual tree. The average difference between the Min-Normal average method and the REF-values amounts to 47 cm on average, that is, for each tree the Normal average method loses 47 cm, for an absolute total sum of losses of 13.86 m for this data

set composed of 26 individual trees. The average difference between the Min-weighted average method and the REF-values amounts to 12 cm, that is, for each tree the weighted average method loses 12 cm, for an absolute total sum of losses of 6.2 m for this data set composed of 26 individual trees.

Tree	Min-Max w/out g. surf.	MIN-Normal Average	MIN-weighted Average
1	19,84	19,706	19,797
2	17,777	17,16	17,724
3	18,15	17,852	18,248
4	15,57	15,022	15,139
5	7,633	7,17001	7,69803
6	19,853	19,591	19,676
7	16,49	16,326	15,841
8	19,317	18,707	19,085
9	17,203	16,477	17,242
10	19,7	18,79	19,153
11	17,473	16,622	17,072
12	19,951	19,258	19,424
13	24,485	24,614	24,448
14	20,215	19,242	20,017
15	20,486	18,712	18,228
16	16,214	15,341	15,737
17	19,144	18	17,807
18	22,089	21,304	21,904
19	18,025	17,812	17,668
20	20,5	19,567	20,148
21	23,096	22,185	22,329
22	11,019	11,264	10,942
23	16,216	14,751	15,266
24	20,679	19,634	20,081
25	23,772	23,214	23,668
26	21,431	20,787	20,917

Table 5.4: Individual tree height N= 100(values in m)

Table regarding height estimation values for individual trees, using the Min-Max method. The values on the right column were calculated without ground surface and for that reason are the closest to reality.

With N=100, tab. 5.4, we obtain sub-optimal results in relation to the reference values taken for each individual tree. The average difference between the Min-Normal average method and the REF-values amounts to 66 cm, that is, for each tree the Normal average method loses 66 cm, for an absolute total sum of losses of 17.96 m for this data set composed of 26 individual trees. The average difference between the Min-Weighted average method and the REF-values amounts to 43 cm, that is, for each tree the weighted average method loses 43 cm, for an absolute total sum of losses of 11.49 m for this data set composed of 26 individual trees.

Table regarding height estimation values for individual trees, using the Min-Max

Tree	Min-Max w/out g. surf.	MIN-Normal Average	MIN-weighted Average
1	19,84	18,706	18,952
2	17,777	17,16	16,634
3	18,15	17,852	17,435
4	15,57	14,022	13,864
5	7,633	7,17001	7,75403
6	19,853	18,591	18,614
7	16,49	15,326	15,413
8	19,317	18,707	18,847
9	17,203	16,477	16,907
10	19,7	18,79	18,492
11	17,473	16,622	17,074
12	19,951	19,258	18,962
13	24,485	23,614	23,901
14	20,215	19,242	19,693
15	20,486	19,712	19,209
16	16,214	15,341	15,046
17	19,144	17	17,113
18	22,089	21,304	21,006
19	18,025	16,812	17,216
20	20,5	19,567	19,923
21	23,096	22,185	22,099
22	11,019	10,264	10,489
23	16,216	14,751	14,863
24	20,679	19,634	19,741
25	23,772	23,214	23,427
26	21,431	20,787	20,357

Table 5.5: Individual tree height N= 2,5% of total points (values in m)

method. The values on the right column were calculated without ground surface and for that reason are the closest to reality.

With N= 2.5% of total points, 5.5 we obtain weak results in relation to the reference values taken for each individual tree. The average difference between the Min-Normal average method and the REF-values amounts to 0.93 m, that is, for each tree the Normal average method loses 0.93 m of height, for an absolute total sum of losses amounting to 24.21 m for this data set composed of 26 individual trees. The average difference between the Min-weighted average method and the REF-values amounts to 0.91 m, that is, for each tree the weighted average method loses 0.91 m, for an absolute total sum of losses of 23.55 m for this data set composed of 26 individual trees.

Consistently all across the results shown here, the Min-weighted Average methods give more optimistic estimates in relation to Normal Average methods, see tab. 5.6. One factor that can explain the poor results of our algorithms when N=2.5% of total points has to do with point cloud density, and that leading to a very large number of points. Overall, when N was given a fix number, tree height estimations were superior but this only holds if tree pointclouds are very dense with plenty of individual points.

Total added losses in comparison with Min-Max values.		
	Normal Average	Weighted Average
N(50) w/ g.surf.	13,86m	6,2m
N(100) w/ g.surf.	17,96m	11,49m
N(2.5%) w/ g.surf.	24,21m	23,55m

Table 5.6: Total added losses (absolute values), Sum(Ref_values - Height value individual tree.)

Both N=100 and N=50, surpass in results N=2.5%. We can calculate the size of pointclouds at which N=2,5% would present better results when estimating individual tree heights, and rapidly understand that using a percentage of points instead of a fixed value is only good for relatively small pointclouds.

$$N(100) = 100/0.025 = 4000, \tag{5.1}$$

A point cloud would have to have less than 4000 points in order to be more beneficial to use N(2.5%) instead of N(100).

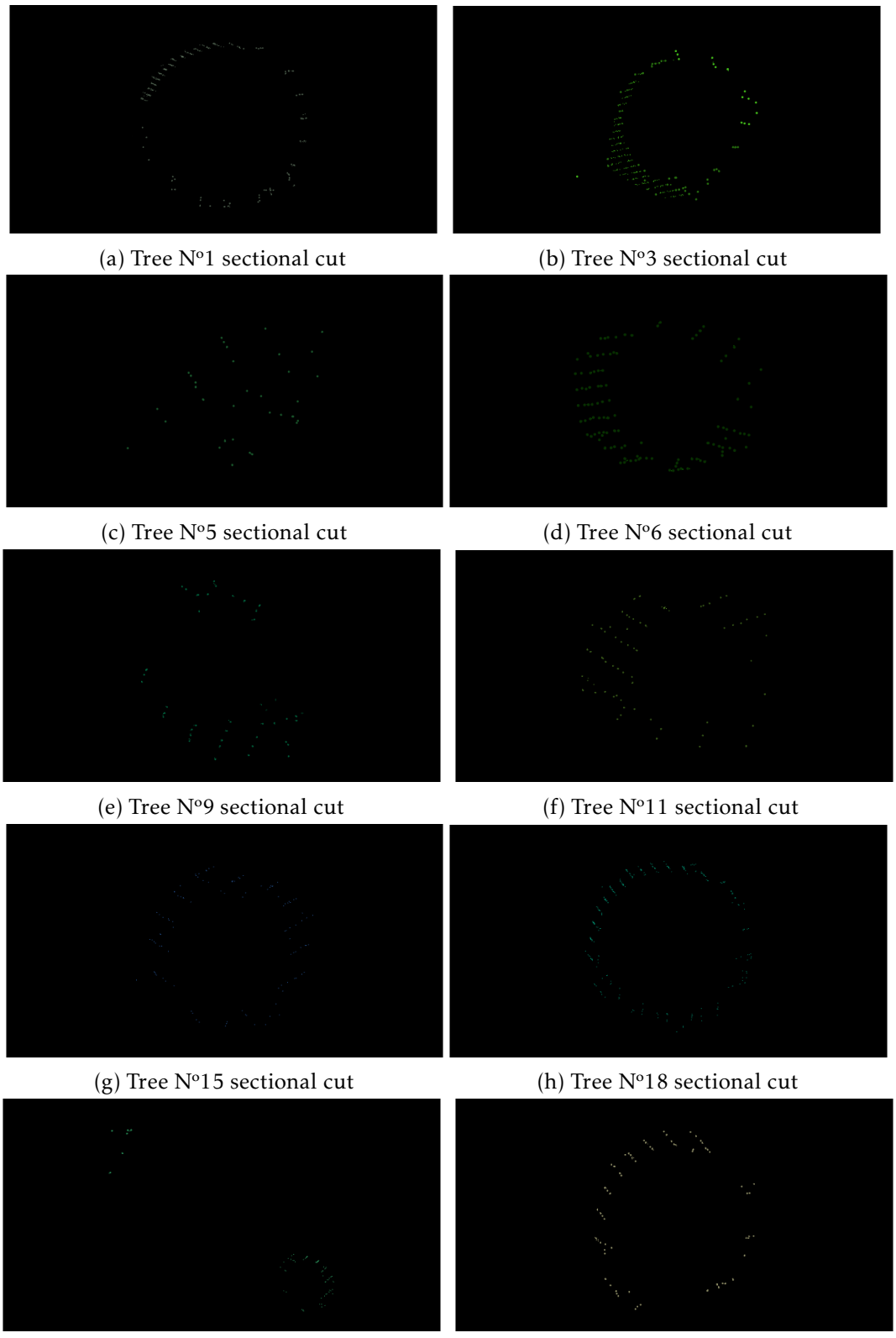
$$N(50) = 50/0.025 = 2000, \tag{5.2}$$

A point cloud would have to have less than 2000 points in order to be more beneficial to use N(2.5%) instead of N(50).

5.2.2 Diameter at Breast Height Results

When working with Diameter at breast height (DBH), we also used the same dataset used previously, consisting of 26 individual tree pointclouds. In this case, we were not able to set reference values of DBH for each tree. The result analysis will mainly consist of critical analysis of the results obtained, comparison between results, and analysis of any outlier value and trying to understand why it could have been so. Remembering, we implemented DBH in two ways, first a method we call XY-Difference and secondly a model Fitting approach. We also calculated the absolute difference between XY-Difference results and Model Fitting results, and then we added all of those differences in order to obtain a value that could evaluate quality. We executed three tests resulting in tables 5.7, 5.8 and 5.9, values come in cm.

Commenting on the results shown in tab.5.7, we can see that they are consistent, with only some trees registering disparities of more than 10cm, those are the alarming ones. For all but 8 trees, results seem very promising when comparing between the two different methods. Understanding inconsistencies coming from the XY-Difference method - like in the case of trees number 12, 16, 22 and 23, was faster to do as they amount to larger values of deviation. Being a rather simple algorithm, XY-Difference is unable to tell the difference between circumferences or other points in the point cloud, unlike its counterpart Model Fitting Algorithm, that behaved in an expected way for those trees.



(a) Tree Nº1 sectional cut

(b) Tree Nº3 sectional cut

(c) Tree Nº5 sectional cut

(d) Tree Nº6 sectional cut

(e) Tree Nº9 sectional cut

(f) Tree Nº11 sectional cut

(g) Tree Nº15 sectional cut

(h) Tree Nº18 sectional cut

(i) Tree Nº23 sectional cut

(j) Tree Nº26 sectional cut

Figure 5.2: Sectional cut of the some of the individual trees right after filtering and flattening.

Tree	XY-Difference	M.Fit.(Dist_th = 0.05)	Abs.Dif.
1	49,2983	49,1321	0,17
2	31,801	45,5962	13,8
3	46,2496	37,2789	8,97
4	45,9021	42,3682	3,53
5	27,849	47,5909	19,74
6	44,1505	37,8684	6,28
7	45,8496	42,7569	3,09
8	38,1001	36,3038	1,8
9	40,8512	39,8426	1,01
10	43,7502	45,4694	1,72
11	29,697	40,4988	10,8
12	389,398	36,8423	352,56
13	49,8009	46,202	3,6
14	46,8006	41,3725	5,43
15	35,0986	53,8967	18,8
16	421,449	34,4609	386,99
17	45,8973	38,8318	7,07
18	46,9488	43,5915	3,36
19	43,0008	40,3872	2,61
20	46,1498	43,6131	2,54
21	58,25	57,1701	1,08
22	357,398	77,5204	279,88
23	94,2986	23,9985	70,3
24	37,45	37,2395	0,21
25	56,2492	55,8085	0,44
26	32,449	36,827	4,38
		Sum of Differences:	1210,16

Table 5.7: Diameter at Breast Height for individual tree. (values in cm)

The question with trees number 12, 16, 22, and 23, has to do with the height at which the tree canopy begins, and for these trees in particular, it begins below 1.37m which is the height at which we filter points. That leads to some tree canopy points still remaining in the point cloud upon which we apply our XY-Difference algorithm, leading to overestimation of DBH for those individual trees. For a visual insight of what we mean refer to fig. 5.3. From here we can conclude that this method is not particularly good for trees whose canopy starts at a rather small height.

Disparities coming from the Model Fitting approach (meaning Model Fitting approach has the highest value), as in the case of trees number 5, 11 and 15 were harder to explain. One thing we could say is that disparities are comparatively of a smaller order than those from the XY-Difference method. As explained the XY-Difference method averages the larger distance between points along the X and Y axis , averaging them in order to produce a estimate. This way, if a tree trunk is not perfectly circular, and more like an elliptical shape, whilst the model fitting algorithm will try to get the largest circumference possible, thus, the diameter of said circumference will be the larger diameter of

Figure 5.3: We can see in dark red colour a circle surrounding the points belonging to the tree trunk, these points will be taken into account for the model fitting approach. In white we can see both the biggest X value difference and the biggest Y value difference. As seen, XY-Difference method takes into account points belonging to the tree canopy that were not filtered out, thus making an exaggerated diameter at breast height estimation. These points belong to tree number 12. Radius Outlier removal set at 0.5m.

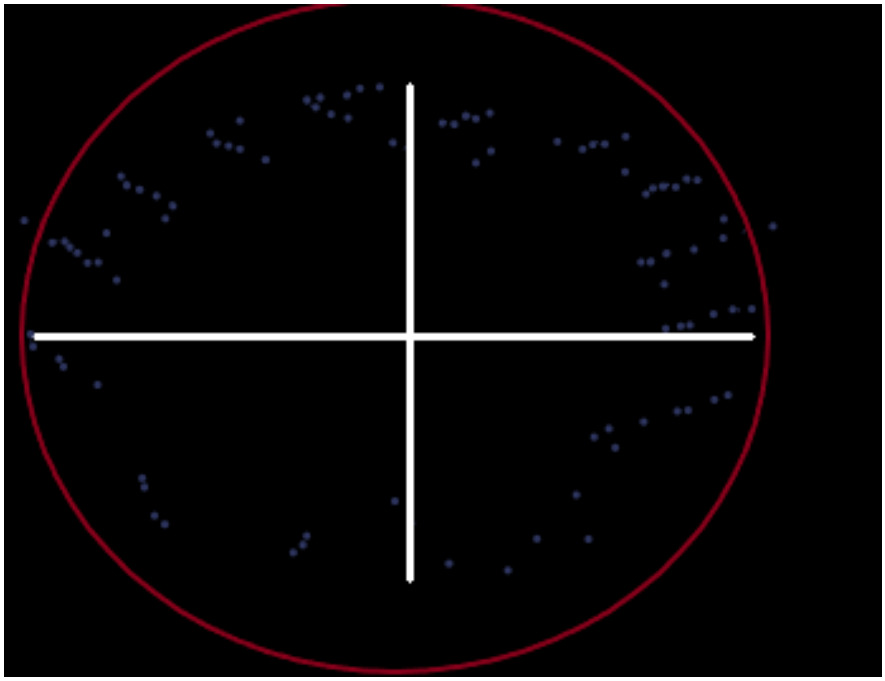


Figure 5.4: We can see in dark red colour an approximate Model fitting estimate for the tree trunk diameter. In white we can see both the biggest X value difference and the biggest Y difference, thus, when averaging we get a lower value for diameter estimation. These points belong to tree number 11 after the flattening stage.

Tree	XY-Difference	M.Fit.(Dist_th = 0.03)	Abs.Dif.
1	49,2983	49,3042	0,01
2	31,801	45,5962	13,8
3	46,2496	37,7237	8,53
4	45,9021	43,5682	2,33
5	27,849	44,9279	17,08
6	44,1505	44,1085	0,04
7	45,8496	51,9052	6,06
8	38,1001	37,6908	0,41
9	40,8512	40,9498	0,1
10	43,7502	48,7816	5,03
11	29,697	40,4988	10,8
12	389,398	38,3734	351,02
13	49,8009	46,2288	3,57
14	46,8006	40,6584	6,14
15	35,0986	44,4378	9,34
16	421,449	40,2045	381,24
17	45,8973	40,2404	5,66
18	46,9488	43,5406	3,41
19	43,0008	43,5132	0,51
20	46,1498	42,7942	3,36
21	58,25	57,9352	0,31
22	357,398	77,7665	279,63
23	94,2986	27,2957	67
24	37,45	42,0689	4,62
25	56,2492	55,5576	0,69
26	32,449	38,414	5,97
		Sum of Differences	1186,6589

Table 5.8: DBH Values for individual trees. Radius for ROR filter set at 0.5m. (values in cm)

such elliptical, tree trunk the XY-Difference method gets us a more conservative value for the diameter. This explains the disparities found on trees number 5, 11, and 15. For an example see fig.5.4.

When lowering the distance threshold related to the model fitting algorithm we can observe that the results, tend to get more similar, that is, disparities between algorithms reduce, as can be seen by the total sum of disparities. This can be explained by one reason, if we tighten the criteria in the model fitting algorithm for it to evaluate which points belong inside the circumference model or not, we tend to get smaller circumferences in diameter and that is what we can observe in table 5.8.

When restricting the Radius Outlier Removal filter from 0.5m to 0.1m we are able to eliminate more points belonging to the tree canopy thus improving results of the XY-Difference average method for trees number 12, 16, 22 and 23. This narrowing in the criteria of the ROR filter means though that also some points that belong to the trunk of tree get discarded and this worsens the results related to the Model Fitting approach,

Tree	XY-Difference	M.Fit.(Dist_th = 0.03)	Abs.Dif.
1	49,2983	51,5451	2,25
2	31,801	45,5962	13,80
3	41,9996	39,8127	2,19
4	45,9021	43,5682	2,33
5	24,0011	79,3261	55,33
6	42,3012	39,3706	2,93
7	41,9006	41,5689	0,33
8	36,75	47,8043	11,05
9	40,8512	41,168	0,32
10	43,7502	42,5097	1,24
11	29,697	51,8763	22,18
12	30,8992	39,5726	8,67
13	49,8009	46,2288	3,57
14	44,8505	40,9808	3,87
15	35,0986	44,4378	9,34
16	175,05	52,3221	122,73
17	32,2495	55,7924	23,54
18	46,9488	43,5406	3,41
19	39,6507	49,7532	10,10
20	46,1498	41,9593	4,19
21	52,3998	58,1393	5,74
22	283,1	156,242	126,86
23	25,6977	63,6402	37,94
24	37,45	44,4162	6,97
25	56,2492	56,7652	0,52
26	32,399	38,507	6,11
		Sum of Differences	487,51

Table 5.9: DBH Values for individual trees. Radius for ROR filter set at 0.1m. (values in cm)

particularly trees number 5 and 22. Overall results tend to improve, with results being more similar throughout the different trees. Fig. 5.5 and fig.5.6 show the practical results of this narrowing of belonging criteria by the ROR filter.

Figure 5.5: ROR radius set to 0.5m. In order for a point to remain in the point cloud, it needed 8 neighbours in a radius of 0.5m.

Figure 5.6: ROR radius set to 0.1m. In order for a point to remain in the point cloud, it needed 8 neighbours in a radius of 0.1m.

5.3 Forest Section Results

In this section we will proceed in developing the results from our tests concerning forest section analysis and parameter estimation. The data used in order to develop our algorithms consisted of a single pointcloud obtained from a free to use online dataset. This point cloud is composed by the individual trees used in the previous section.

One problem found when dealing with forest sections had to do with the partitioning of the forest section itself. When you set rigid values for the side your grid squares, you may end up with tree trunks cut in half. This presents itself as a difficulty, and no solution could be found to solve this problem while keeping the core fundamentals of our implementation.

5.3.1 Grid Height Results and Tree Trunk Detection

In relation to grid height, and estimation of a grid square height, we did four trials. The first two runs the only variable was the grid square size, 2m and 4m. The other two runs we used a grid square of two meters, but complemented our grid height estimation with our trunk detection methods previously described in section 4.4. This allow us to ignore squares with no presence of a tree trunk and thus making our estimation more precise. The grid height estimation results concerning a grid size of 2m can be seen in fig. 5.10, these values come in meters and were estimated with ground surface and with no tree trunk detection methods on.

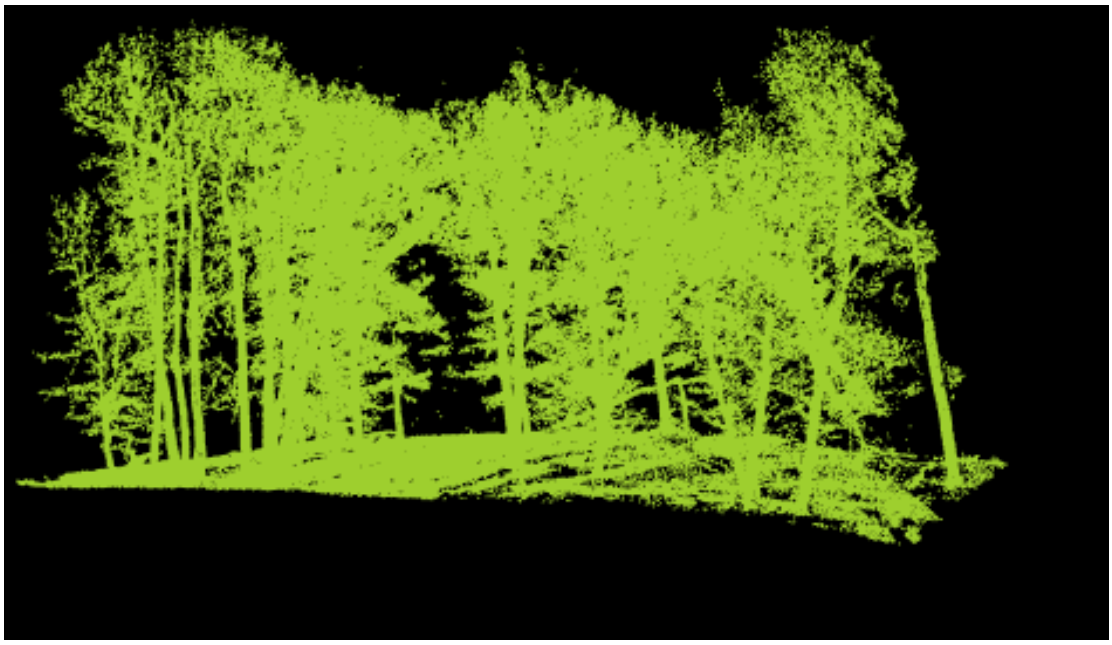


Figure 5.7: Side Profile of the section of forest used to estimate Grid Height.

In both figures 5.10 and fig. 5.11 we can see an overall birds eye view, of the elevation in relation to the ground of the entirety of the tree's canopies. We can also see that both

Index	Percentage of Points
I 10	1,74%
I 9	7,73%
I 8	14,73%
I 7	11,12%
I 6	7,72%
I 5	11,86%
I 4	9,95%
I 3	6,82%
I 2	9,07%
I 1	19,26%
	100,00%

Table 5.10: Density index for tree depicted in fig. 5.8

fig.5.10 and fig.5.11 roughly correspond all throughout the entirety of the forest section, that is, results don't differ all that much. When we raise the size of the grid square from 2m to 4m what we end up doing is we end up with 4 of the old grid squares, the 2m by 2m, fitting inside, a 4m by 4m.

That being said, it was expected that when we were to raise the size of the grid square, from 2m to 4m, what we would end up getting for each grid square would be a number slightly bigger than the highest value of the corresponding 2m by 2m squares, and that holds, true. This is due to the method we use in order to estimate grid square height, Min-Normal arithmetic average with N=50, and to the fact that, has you increase the size of the grid square, the slope of the terrain ends up affecting the end result much more, inflating it.

As said previously in the introduction to this subsection, one of the weakness of dealing with forest sections through a partitioning perspective has to do with information that can be lost, divided, or in this case, doubled. The possibility of having tree trunks cut in half is a weakness of this approach. As we know for sure that the forest section contains only 26, if we check fig. 5.14 we can ascertain that more than 26 grid squares present us with a value, that means that in our most restrictive method for tree trunk detection, still some anomalous squares remain. This can be due to the fact that we may sometimes, due to the grid system implemented, obtain vertically sliced up tree trunks as a consequence, that way ,these tree trunk in adjacent squares will be detected not as a single individual tree but as two different trees, as they are analysed separately.

Even thought the index trunk detection provides us the best results when it comes to filtering out grid squares with no useful information, it still has its weaknesses when it comes to identifying individual trees, and being able to count them. Also it is unable to differentiate how many tree trunks are in a given grid square, and as such it is not very flexible.

In order to evaluate the success of our tree trunk detection algorithms, we decided

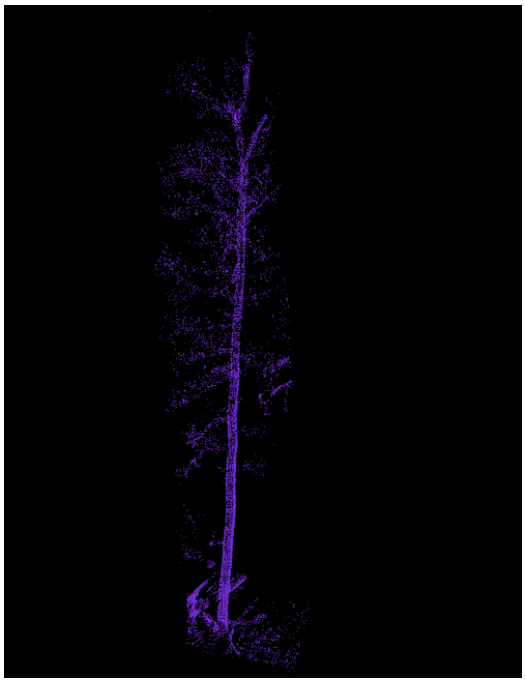


Figure 5.8: Example of a valid grid square (4m by 4m) that must be taken into account when estimating grid height.

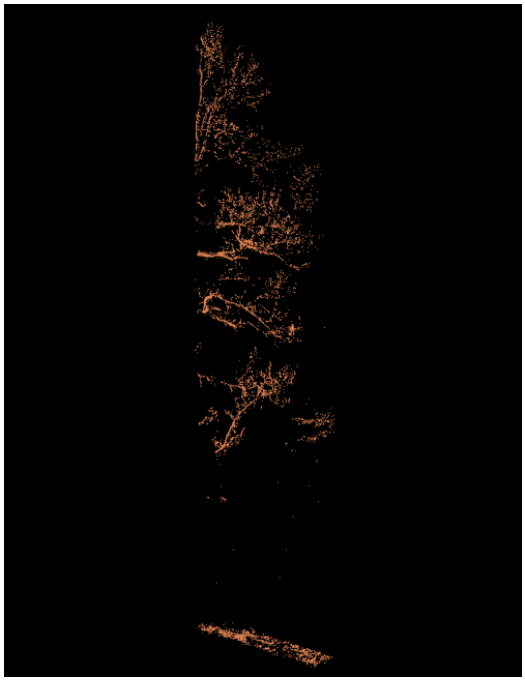


Figure 5.9: Example of a grid square (4m by 4m) that should not be taken into account when estimating grid height. As we an see, no tree trunk is born in this square, and as such, it can't be counted for grid square height assessment.

Index	Percentage of Points
I 10	8.35%
I 9	10.77%
I 8	13.55%
I 7	17.60%
I 6	13.33%
I 5	12.46%
I 4	6.67%
I 3	0.30%
I 2	0.03%
I 1	16,94%
	100,00%

Table 5.11: Density index for tree depicted in fig. 5.9

to divide our forest section, see fig.5.7, into grid squares of 4m by 4m. After this, we analysed each of the 4m by 4m grid squares pointclouds and manually we separated grid squares that had the presence of a tree trunk to one side, such as those like fig. 5.8, and those that didn't have the presence of a tree trunk, such as fig. 5.9, to the other. Right after this manual classification we had 19 grid squares with a valid tree trunk, and 36 grid squares that did not. We then used our algorithms to analyse said pointclouds separately, that is, first the ones with tree trunk, and then the ones without tree trunk. Like this, we were able to analyse the accuracy of our methods, both Density Index method, see tab.5.12, and the Region Growing method using the values set in tab.4.8, these results can be seen in tab.5.13.

		Index Algorithm Classification		
		W/ Trunk	W/out Trunk	Total
Manual Classification	W/ Trunk	18	1	19
	W/out trunk	4	32	36

Table 5.12: Density index algorithm success table. Manual classification vs. Algorithm classification. 19 grid squares were identified by us as having a tree trunk present, and the remaining 36 were classified as not having any tree trunk. Of the 19 grid squares with tree trunk, our Density index confirmed 18, and miss classified 1. Of 36 grid squares without tree trunk our Density index Algorithm confirmed 32 and miss classified 4.

		Region growing Algorithm Classification		
		W/ Trunk	W/out Trunk	Total
Manual Classification	W/ Trunk	15	4	19
	W/out trunk	30	6	36

Table 5.13: Region growing algorithm, using values for RG algorithm depicted in tab.4.8, success table. Manual classification vs. Algorithm classification. 19 grid squares were identified by us as having a tree trunk present, and the remaining 36 were classified as not having any tree trunk. Of the 19 grid squares with tree trunk, our region growing algorithm confirmed 15, and miss classified 4. Of 36 grid squares without tree trunk our Region growing Algorithm confirmed 6 and miss classified 30.

In relation to false positives results for the density index, grid squares that didn't have a tree trunk but the algorithm still identified one, we were able to identify the causes for such results. Three of the false positives were grid squares consisting only of ground surface points. Because we do not use a fix step in the density index - instead we divide the max_z-min_z by ten, see fig.4.15- and there are only ground surface points, all the points are uniformly distributed, miss-leading our algorithm. The remaining false positive has to with a grid square that contains a side tree trunk coming from an adjacent square, which tricks our algorithm into believing there is a tree trunk in that square. Overall results related to the region growing approach turned out to be disappointing, with the algorithm being able to consistently select grid squares with tree trunks, but being unable to clear out grid squares without a tree trunk.

0,00	0,00	0,00	0,93	19,39	19,70	20,21	20,51	18,86	18,27	18,68	19,11	18,62	19,02	17,64	17,25	15,83	13,51	12,17	8,77	7,24	6,12
0,00	0,00	0,00	0,00	19,97	21,40	21,72	22,11	19,49	18,71	20,08	20,45	18,89	17,57	19,12	18,81	17,30	15,92	12,65	10,98	8,88	5,54
0,00	0,00	0,00	0,00	19,39	21,93	21,29	21,68	19,96	20,34	19,54	19,93	18,50	19,09	19,67	19,23	15,75	11,35	12,07	13,74	12,39	10,88
0,00	0,00	0,00	15,49	19,06	21,44	22,90	21,22	18,54	15,81	18,26	16,71	19,07	18,52	17,19	15,64	15,29	12,88	14,48	15,16	15,96	17,57
0,00	0,00	0,00	9,10	20,52	21,02	22,33	20,70	20,13	19,50	17,90	18,28	17,67	18,21	4,68	11,19	14,74	13,39	15,01	15,71	16,35	19,02
0,00	0,00	3,14	10,62	19,03	23,36	23,69	23,14	19,61	20,09	19,38	18,91	14,00	15,85	17,38	17,93	16,40	16,12	15,56	16,17	16,80	17,67
0,00	6,14	18,62	20,11	21,53	23,72	24,11	24,60	20,18	20,56	19,95	18,39	12,99	16,38	20,04	20,36	19,04	16,67	17,15	15,73	17,16	18,95
0,00	11,59	20,07	20,56	20,70	24,17	24,72	23,11	18,53	20,00	19,54	16,06	10,62	16,09	18,49	19,84	19,65	17,05	16,66	16,21	17,73	19,27
0,00	10,97	13,42	21,85	23,17	22,60	23,18	23,59	22,07	18,61	17,14	14,72	4,12	15,34	17,10	15,44	14,07	13,69	12,17	15,94	15,47	19,24
0,83	11,28	21,71	23,20	23,67	23,99	23,53	24,00	22,45	20,99	21,58	17,04	10,53	6,81	4,37	2,83	2,16	5,87	11,48	14,02	12,86	14,60

Figure 5.10: Height quotas of the forest section as seen from above. A 2m grid square was used. No ground surface Removal. No tree trunk detection. Normal Average method for height estimation with N=50 used. Values in meters.

0,81	1,42	21,40	22,11	19,71	20,45	19,57	19,81	17,92	12,98	9,54
0,89	15,49	22,44	23,22	20,81	20,71	20,52	20,64	16,88	15,16	17,57
0,00	13,62	23,36	24,14	21,09	19,91	18,85	17,93	17,12	17,17	19,67
11,59	21,56	24,17	26,11	22,00	21,06	17,09	20,84	20,05	18,21	20,27
12,28	23,20	23,99	25,00	23,99	22,04	15,81	17,84	14,87	16,02	19,60

Figure 5.11: Height quotas of the forest section as seen from above. A 4m grid square was used. No ground surface Removal. No tree trunk detection. Normal Average method for height estimation with N=50 used. Values in meters.

0,00	0,00	0,00	0,93	19,39	19,70	20,21	20,51	18,86	18,27	18,68	19,11	18,62	19,02	17,64	0,00	15,83	13,51	0,00	8,77	0,00	0,00
0,00	0,00	0,00	0,00	19,97	21,40	21,72	22,11	0,00	18,71	20,08	20,45	18,89	0,00	19,12	0,00	17,30	15,92	0,00	10,98	0,00	0,00
0,00	0,00	0,00	0,00	19,39	21,93	0,00	21,68	0,00	0,00	19,54	19,93	18,50	0,00	19,67	19,23	0,00	11,35	12,07	13,74	12,39	0,00
0,00	0,00	0,00	15,49	19,06	21,44	22,90	21,22	18,54	0,00	18,26	16,71	19,07	18,52	0,00	0,00	0,00	12,88	0,00	15,16	0,00	0,00
0,00	0,00	0,00	9,10	0,00	21,02	0,00	20,70	0,00	19,50	17,90	18,28	17,67	18,21	4,68	0,00	14,74	0,00	15,01	15,71	0,00	0,00
0,00	0,00	3,14	10,62	19,03	23,36	23,69	23,14	0,00	20,09	19,38	18,91	0,00	15,85	17,38	17,93	0,00	0,00	15,56	0,00	0,00	17,67
0,00	6,14	18,62	0,00	0,00	23,72	24,11	24,60	20,18	20,56	19,95	18,39	0,00	16,38	0,00	20,36	0,00	16,67	0,00	15,73	17,16	18,95
0,00	11,59	0,00	20,56	0,00	24,17	0,00	23,11	18,53	20,00	0,00	16,06	0,00	16,09	18,49	19,84	0,00	17,05	16,66	16,21	0,00	0,00
0,00	0,00	0,00	21,85	0,00	22,60	23,18	23,59	22,07	18,61	17,14	14,72	4,12	0,00	17,10	15,44	14,07	13,69	12,17	15,94	15,47	19,24
0,83	11,28	0,00	23,20	23,67	0,00	0,00	24,00	22,45	20,99	21,58	17,04	10,53	0,00	0,00	2,83	2,16	0,00	0,00	14,02	12,86	0,00

Figure 5.12: Height quotas of the forest section as seen from above. A 2m grid square was used. No ground surface Removal. Region Growing Method for tree trunk Detection, values used in tab. 4.7. Normal Average method for height estimation with N=50 used. Values in meters.

0,00	0,00	0,00	0,00	19,39	19,70	0,00	0,00	0,00	0,00	0,00	0,00	0,00	19,02	17,64	0,00	0,00	0,00	0,00	8,77	0,00	0,00
0,00	0,00	0,00	0,00	19,97	21,40	0,00	0,00	0,00	0,00	20,08	20,45	0,00	17,57	19,12	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00	19,39	21,93	0,00	0,00	0,00	0,00	0,00	0,00	18,50	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00	19,06	21,44	0,00	0,00	0,00	0,00	0,00	16,71	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	9,10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	18,28	0,00	18,21	4,68	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	10,62	0,00	23,36	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,00	23,72	24,11	0,00	0,00	0,00	0,00	0,00	0,00	16,38	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	11,59	20,07	0,00	0,00	0,00	24,72	0,00	0,00	0,00	0,00	16,06	10,62	16,09	0,00	19,84	0,00	0,00	0,00	16,21	0,00	0,00
0,00	0,00	0,00	0,00	23,17	22,60	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	12,17	15,94	15,47	19,24
0,00	0,00	0,00	23,20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	14,02	12,86	0,00

Figure 5.13: Height quotas of the forest section as seen from above. A 2m grid square was used. No ground surface Removal. Region Growing Method for tree trunk Detection, values used in tab. 4.8. Normal Average method for height estimation with N=50 used. Values in meters.

Figure 5.14: Height quotas of the forest section as seen from above. A 2m grid square was used. No ground surface Removal. Density Index for tree trunk detection. Normal Average method for height estimation with N=50 used. Values in meters.

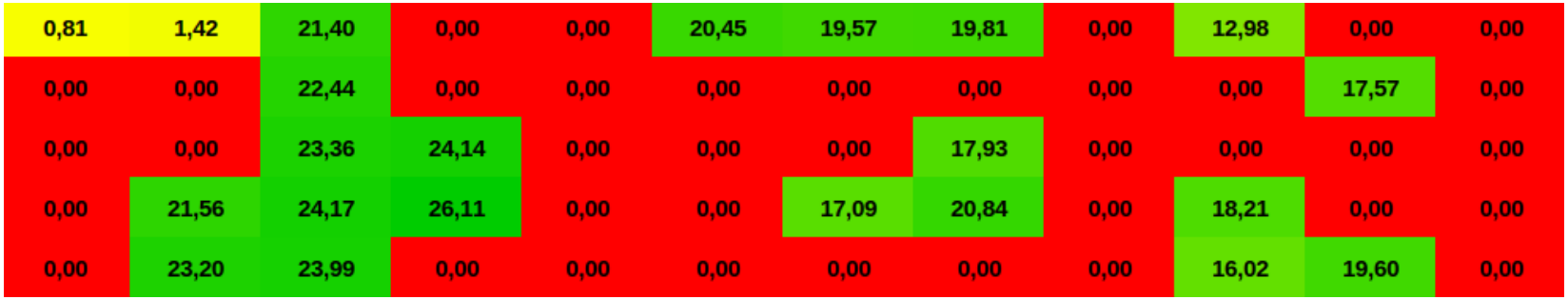


Figure 5.15: Height quotas of the forest section as seen from above. A 4m grid square was used. No ground surface removal. Density Index for tree trunk detection. Normal Average method for height estimation with N=50 used. Values in meters.

Chapter 6

Conclusions and Future work

In form of conclusion one shall proceed with analysing the work and approach of this dissertation. We do think of the problematic of forest management has a very pressing concern as it as to do with ecosystem that are responsible for the fixation of CO2. This characteristic helps to keep global temperatures well balanced, and as such, looking after forest ecosystems and properly managing them proves to be fundamental in today's world. The more information we can obtain that will help to preserve such habitats, the better.

As such, a Fuel Mass Mapping for Forested Areas was presented. During the experimental testing stage the system showed the ability to correctly estimate metrics such as individual tree Diameter at breast height (DBH), individual tree height, and grid square forest height. We were also able to develop a solid method allowing us to understand if a tree was present in a given grid square or not. Metrics such as grid square forest height are then presented to the user in a grid-like format with different granularity. While this may not allow a complete estimation of fuel biomass, it is usable in real world scenarios, namely in assessing the age of a certain section of forest based on it's height, estimating biomass presence based on models that only take height into account and based on our tree trunk detection capabilities, this system is also capable of assessing tree dispersion on a given plot of land.

This is very appealing for forest management and of great importance worldwide and namely in Portugal at the current time. During the summer season Portugal is relentlessly affected by wildfires. With global warming, the severity of said wildfires is certain to increase. Another negative contribution comes from the fact that Portugal has an interior landscape that is much less populated than it's coastal area, leaving vast swathes of forested territory abandoned or unattended. As such, using a Unmanned Aerial Vehicle (UAV) to collect large amounts of data, combined with tools such as the one developed, can give insight and useful data about our forested ecosystems, allowing for a better

monitoring of fuel presence and better knowledge about the stage of development of large portions of forested territory, which in turn, can lead to better forest management practices mitigating the effect of wildfires.

Biomass estimation for single trees was possible, as we had a way of estimating both DBH and Tree Height for individual trees. Many models could be found in literature, and as such it would have been an easy next step. The reason we chose not to do it has got to do with the fact that the goal we had established in the beginning had in mind the build of our own model, but for that, we needed our own easy access forest section, and outside expert knowledge in real world biomass estimation, so that we could then build our models.

The choice of building the system upon ROS [56] was to create a modular architecture. Constructing modules capable of executing a minor task, but easy to integrate with each other. ROS is also a commonly used system worldwide and this allows that the work developed in this dissertation can be used and worked on by other individuals without a need for a great curve of learning, this is, if the user is minimally familiar with ROS.

Future Work:

- Fully implement DBH estimation for trees in grid squares. Development of a tree trunk detection algorithm, capable, at the same time, of segmentation the points belonging to the tree trunk.

- Implementation of multi channel imagery output in regards to forest sections. The grid structure allows for implementation in multi channel imagery, as for each pixel you can then correspond a grid square. A different channel can be used for each of metric, and another channel can be designated for biomass value.

- Arranging for a plot of forest ground to be allocated to this study and for further development. Having easy access to a forest ground would allow for a stack up of future work results against the real world. It will also allow for the building of our own models to estimate biomass presence, as for that you need to fit your model to the real world data.

- Study the impact of this grid square approach in regards to cutting trees in half. Analyse and seek a way to implement full scale forest section biomass estimation.

- Development of algorithms capable of identifying different tree species based on tree height attributes, diameter at breast height and density index. Development of new metrics if needed.

- Development of algorithms capable of measuring below canopy shrubs and plants, most times responsible for fire propagation. This is useful to identify forest areas that may need human intervention, that is, need to be trimmed.

- Implementation of an algorithm capable of estimating the height at which the tree canopy begins.

Bibliography

[1] R Watson, I. Noble, B Bolin, N Ravindranath, D Verardo, and D Dokken. *Land Use, Land-Use Change and Forestry. A Special Report of the International Panel on Climate Change*. Jan. 2000, pp. –.

[2] E. Næsset and T. Gobakken. "Estimation of above- and below-ground biomass across regions of the boreal forest zone using airborne laser." In: *Remote Sensing of Environment*. Vol. 112. 6. 2008, pp. 3079–3090. DOI: 10.1016/j.rse.2008.03.004.

[3] D. Wang, B. Wan, J. Liu, Y. Su, Q. Guo, P. Qiu, and X. Wu. "Estimating aboveground biomass of the mangrove forests on northeast Hainan Island in China using an upscaling method from field plots, UAV-LiDAR data and Sentinel-2 imagery." In: *International Journal of Applied Earth Observation and Geoinformation*. Vol. 85. September 2019. Elsevier, 2020, p. 101986. DOI: 10.1016/j.jag.2019.101986. URL: https://doi.org/10.1016/j.jag.2019.101986.

[4] *Sentinel-2 - Satellite Description - Sentinel Online*. URL: https://sentinel.esa.int/web/sentinel/missions/sentinel-2/satellite-description (visited on 02/05/2020).

[5] L. Wallace, A. Lucieer, C. Watson, and D. Turner. "Development of a UAV-LiDAR system with application to forest inventory." In: *Remote Sensing* 4.6 (2012), pp. 1519–1543. ISSN: 20724292. DOI: 10.3390/rs4061519.

[6] *Utilização da Tecnologia Lidar em Drones - PixForce*. URL: https://pixforce.com.br/utilizacao-da-tecnologia-lidar-em-drones/ (visited on 02/07/2020).

[7] W. Li, Z. Niu, N. Huang, C. Wang, S. Gao, and C. Wu. "Airborne LiDAR technique for estimating biomass components of maize: A case study in Zhangye City, Northwest China." In: *Ecological Indicators*. Vol. 57. Elsevier Ltd, 2015, pp. 486–496. DOI: 10.1016/j.ecolind.2015.04.016. URL: http://dx.doi.org/10.1016/j.ecolind.2015.04.016.

[8] A. K. Tilling, G. J. O'Leary, J. G. Ferwerda, S. D. Jones, G. J. Fitzgerald, D. Rodriguez, and R. Belford. "Remote sensing of nitrogen and water stress in wheat." In: *Field Crops Research* 104.1-3 (2007), pp. 77–85. ISSN: 03784290. DOI: 10.1016/j.fcr.2007.03.023.

[9] K. Lim, P. Treitz, M. Wulder, and M. Flood. "LiDAR remote sensing of forest structure." In: 1 (2003), pp. 88–106.

[10] F. Morsdorf, C. Nichol, T. Malthus, and I. H. Woodhouse. "Assessing forest structural and physiological information content of multi-spectral LiDAR waveforms by radiative transfer modelling." In: *Remote Sensing of Environment* 113.10 (2009), pp. 2152–2163. ISSN: 0034-4257. DOI: 10.1016/J.RSE.2009.05.019. URL: https://www.sciencedirect.com/science/article/pii/S003442570900176X.

[11] J. Gayon. "History of the Concept of Allometry." In: *American Zoologist* 40.5 (2000), pp. 748–758. ISSN: 0003-1569. DOI: 10.1093/icb/40.5.748.

[12] N. Knapp, R. Fischer, V. Cazcarra-Bes, and A. Huth. "Structure metrics to generalize biomass estimation from lidar across forest types from different continents." In: *Remote Sensing of Environment*. Vol. 237. December 2019. Elsevier, 2020, p. 111597. DOI: 10.1016/j.rse.2019.111597. URL: https://doi.org/10.1016/j.rse.2019.111597.

[13] T. Jucker, J. Caspersen, J. Chave, C. Antin, N. Barbier, F. Bongers, M. Dalponte, K. Y. van Ewijk, D. I. Forrester, M. Haeni, S. I. Higgins, R. J. Holdaway, Y. Iida, C. Lorimer, P. L. Marshall, S. Momo, G. R. Moncrieff, P. Ploton, L. Poorter, K. A. Rahman, M. Schlund, B. Sonké, F. J. Sterck, A. T. Trugman, V. A. Usoltsev, M. C. Vanderwel, P. Waldner, B. M. Wedeux, C. Wirth, H. Wöll, M. Woods, W. Xiang, N. E. Zimmermann, and D. A. Coomes. "Allometric equations for integrating remote sensing imagery into forest monitoring programmes." In: *Global Change Biology* 23.1 (2017), pp. 177–190. ISSN: 13652486. DOI: 10.1111/gcb.13388.

[14] A. Pommerening. "Basic tree variables , forestry summary characteristics and biodiversity measures." In: (2007).

[15] B. Apostol, M. Petrila, A. Lorenţ, A. Ciceu, V. Gancz, and O. Badea. "Species discrimination and individual tree detection for predicting main dendrometric characteristics in mixed temperate forests by use of airborne laser scanning and ultra-high-resolution imagery." In: *Science of the Total Environment*. Vol. 698. 2020. DOI: 10.1016/j.scitotenv.2019.134074.

[16] *Sensors 101: The Basics of LiDAR, Thermal, Hyperspectral, and Multispectral Technology*. URL: https://www.precisionhawk.com/blog/media/topic/sensors-101-basics-lidar-thermal-hyperspectral-multispectral-technology (visited on 01/31/2020).

[17] *Drone Multispectral Imaging for Agriculture Training Course*. URL: https://abjacademy.global/drone-training-courses/multispectral-imaging-agriculture/ (visited on 02/07/2020).

[18] *The Basics of LiDAR - Light Detection and Ranging - Remote Sensing | NSF NEON | Open Data to Understand our Ecosystems*. URL: https://www.neonscience.org/lidar-basics (visited on 01/31/2020).

[19] J. S. Evans, A. T. Hudak, R. Faux, and A. M. Smith. "Discrete return lidar in natural resources: Recommendations for project planning, data processing, and deliverables." In: *Remote Sensing* 1.4 (2009), pp. 776–794. ISSN: 20724292. DOI: 10.3390/rs1040776.

[20] A. T. Hudak, N. L. Crookston, J. S. Evans, M. J. Falkowski, A. M. Smith, P. E. Gessler, and P. Morgan. "Regression modeling and mapping of coniferous forest basal area and tree density from discrete-return lidar and multispectral satellite data." In: *Canadian Journal of Remote Sensing* 32.2 (2006), pp. 126–138. ISSN: 17127971. DOI: 10.5589/m06-007.

[21] J. E. Means, S. A. Acker, B. J. Fltt, M. Renslow, and C. J. Hendrix. "Means_et_al_-2000." In: (2000), pp. 1367–1371.

[22] V. Ussyshkin and L. Theriault. "Airborne lidar: Advances in discrete return technology for 3D vegetation mapping." In: *Remote Sensing*. Vol. 3. 3. 2011, pp. 416–434. DOI: 10.3390/rs3030416.

[23] L. Korhonen, I. Korpela, J. Heiskanen, and M. Maltamo. "Airborne discrete-return LIDAR data in the estimation of vertical canopy cover, angular canopy closure and leaf area index." In: *Remote Sensing of Environment* 115.4 (2011), pp. 1065–1080. ISSN: 00344257. DOI: 10.1016/j.rse.2010.12.011. URL: http://dx.doi.org/10.1016/j.rse.2010.12.011.

[24] *A Complete Guide to LiDAR: Light Detection and Ranging - GIS Geography*. URL: https://gisgeography.com/lidar-light-detection-and-ranging/ (visited on 01/31/2020).

[25] C. Mallet and F. Bretar. "Full-waveform topographic lidar: State-of-the-art." In: *ISPRS Journal of Photogrammetry and Remote Sensing*. 2009. DOI: 10.1016/j.isprsjprs.2008.09.007.

[26] W. Yao, P. Krzystek, and M. Heurich. "Tree species classification and estimation of stem volume and DBH based on single tree extraction by exploiting airborne full-waveform LiDAR data." In: *Remote Sensing of Environment*. Vol. 123. Elsevier Inc., 2012, pp. 368–380. DOI: 10.1016/j.rse.2012.03.027. URL: http://dx.doi.org/10.1016/j.rse.2012.03.027.

[27] C. J. Gleason and J. Im. "Forest biomass estimation from airborne LiDAR data using machine learning approaches." In: *Remote Sensing of Environment* 125 (2012), pp. 80–91. ISSN: 00344257. DOI: 10.1016/j.rse.2012.07.006. URL: http://dx.doi.org/10.1016/j.rse.2012.07.006.

[28] S. Luo, C. Wang, X. Xi, S. Nie, X. Fan, H. Chen, D. Ma, J. Liu, J. Zou, Y. Lin, and G. Zhou. "Estimating forest aboveground biomass using small-footprint full-waveform airborne LiDAR data." In: *International Journal of Applied Earth Observation and Geoinformation*. Vol. 83. July. Elsevier, 2019, p. 101922. DOI: 10.1016/j.jag.2019.101922. URL: https://doi.org/10.1016/j.jag.2019.101922.

[29] *View the Point Cloud Segmentation Results · LiDAR360 User Guide*. URL: https://greenvalleyintl.com/wp-content/lidar360{_}en/ToolReference/ALSForest/ViewthePointCloudSegmentationResults.html (visited on 02/07/2020).

[30] A. Nguyen and B. Le. "3D point cloud segmentation: A survey." In: *IEEE Conference on Robotics, Automation and Mechatronics, RAM - Proceedings* November (2013), pp. 225–230. ISSN: 2158219X. DOI: 10.1109/RAM.2013.6758588.

[31] B. Bhanu, S. Lee, C. C. Ho, and T. Henderson. "Range Data Processing: Representation of Surfaces By Edges." In: *Proceedings - International Conference on Pattern Recognition* (1986), pp. 236–238.

[32] P. J. Besl and R. C. Jain. "Segmentation Through Variable-Order Surface Fitting." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.2 (1988), pp. 167–192. ISSN: 01628828. DOI: 10.1109/34.3881.

[33] J. Liu, T. Wang, A. K. Skidmore, S. Jones, M. Heurich, B. Beudert, and J. Premier. "Comparison of terrestrial LiDAR and digital hemispherical photography for estimating leaf angle distribution in European broadleaf beech forests." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 158.September (2019), pp. 76–89. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2019.09.015. URL: https://doi.org/10.1016/j.isprsjprs.2019.09.015.

[34] M. a. Fischler and R. C. Bolles. "Paradigm for Model." In: *Communications of the ACM* 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692.

[35] *pcl/Handbook/RANSAC - ROS Wiki*. URL: http://wiki.ros.org/pcl/Handbook/RANSAC (visited on 02/07/2020).

[36] B. W. Smith and G. J. Brand. "Allometric biomass equations for 98 species of herbs, shrubs, and small trees." In: *USDA Forest Service* Research N.3 (1983), pp. 1–8.

[37] D. Seidel, S. Fleck, C. Leuschner, and T. Hammett. "Review of ground-based methods to measure the distribution of biomass in forest canopies." In: *Annals of Forest Science* 68.2 (2011), pp. 225–244. ISSN: 1297-966X. DOI: 10.1007/s13595-011-0040-z. URL: https://doi.org/10.1007/s13595-011-0040-z.

[38] M. Bouvier, S. Durrieu, R. A. Fournier, and J. P. Renaud. "Generalizing predictive models of forest inventory attributes using an area-based approach with airborne LiDAR data." In: *Remote Sensing of Environment* 156 (2015), pp. 322–334. ISSN: 00344257. DOI: 10.1016/j.rse.2014.10.004. URL: http://dx.doi.org/10.1016/j.rse.2014.10.004.

[39] *Lidar for Biomass Estimation | IntechOpen*. URL: https://www.intechopen.com/books/biomass-detection-production-and-usage/lidar-for-biomass-estimation (visited on 01/31/2020).

[40] V. Thomas, P. Treitz, J. H. McCaughey, and I. Morrison. "Mapping stand-level forest biophysical variables for a mixedwood boreal forest using lidar: An examination of scanning density." In: *Canadian Journal of Forest Research* 36.1 (2006), pp. 34–47. ISSN: 00455067. DOI: 10.1139/x05-230.

[41] Y. Fallah and M. Onur. "Lidar for Biomass Estimation." In: *Biomass - Detection, Production and Usage* (2011). DOI: 10.5772/16919.

[42] *Bags - ROS Wiki*. URL: http://wiki.ros.org/Bags (visited on 10/22/2020).

[43] *pcl - ROS Wiki*. URL: http://wiki.ros.org/pcl (visited on 02/07/2020).

[44] *GitHub - cra-ros-pkg/robot$_l$ocalization : robot$_l$ocalizationisapackageof nonlinearstateestimationnodes.* URL: https://github.com/cra-ros-pkg/robot_localization.

[45] *pcl_ros - ROS Wiki*. URL: http://wiki.ros.org/pcl{_}ros{\#}bag{_}to{_}pcd (visited on 11/30/2020).

[46] *pcl_ros - ROS Wiki*. URL: http://wiki.ros.org/pcl{_}ros{\#}bag{_}to{_}pcd (visited on 11/30/2020).

[47] *StatisticalOutlierRemoval*. URL: https://pointclouds.org/documentation/classpcl_1_1_statistical_outlier_removal.html.

[48] *Removing outliers using a Conditional or RadiusOutlier removal — Point Cloud Library 0.0 documentation*. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/remove{_}outliers.html?highlight=outlierremoval{\#}removing-outliers-using-a-conditional-or-radiusoutlier-removal (visited on 11/29/2020).

[49] *Removing outliers using a Conditional or RadiusOutlier removal — Point Cloud Library 0.0 documentation*. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/remove{_}outliers.html?highlight=outlierremoval{\#}removing-outliers-using-a-conditional-or-radiusoutlier-removal (visited on 11/29/2020).

[50] *Region growing segmentation — Point Cloud Library 0.0 documentation*. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/region{_}growing{_}segmentation.html{\#} (visited on 11/29/2020).

[51] *Region growing segmentation — Point Cloud Library 0.0 documentation*. URL: https://pcl.readthedocs.io/projects/tutorials/en/latest/region{_}growing{_}segmentation.html{\#} (visited on 11/29/2020).

[52] *pcl_ros - ROS Wiki*. URL: http://wiki.ros.org/pcl{_}ros{\#}bag{_}to{_}pcd (visited on 11/30/2020).

[53] J. Trochta, M. Kruček, and K. Kraâl. *3D Forest - Terrestrial lidar data processing tool*. URL: https://www.3dforest.eu/ (visited on 11/30/2020).

[54] Georg von Wühlisch. "Technical Guidelines for genetic conservation and use for European beech (Fagus sylvatica)." In: *Euforgen* (2008), p. 6. URL: http://www.euforgen.org/species/fagus-sylvatica/.

[55] *Creative Commons — Attribution-ShareAlike 4.0 International — CC BY-SA 4.0*. URL: https://creativecommons.org/licenses/by-sa/4.0/ (visited on 11/28/2020).

[56] *ROS/Introduction - ROS Wiki*. URL: http://wiki.ros.org/ROS/Introduction (visited on 02/04/2020).