Novel algorithms for efficiently accumulating, analysing and visualising full-waveform LiDAR in a volumetric representation with applications to forestry

submitted by

Milto Miltiadou

for the degree of Doctor of Engineering

of the

University of Bath

Centre for Digital Entertainment

and of the

Plymouth Marine Laboratory

NERC Airborne Research Facility

December 2016

Abstract

no more than 300 words

NOTES:

Blue colour: additions according to Neill's feedback,

Red colour: notes

Gray colour: text that is going to be modified

To be added on top

Abstract

This study focuses on enhancing the visualisations and classifications of forested areas using coincident full-waveform (fw) LiDAR data and hyperspectral images. The ultimate aim is use both datasets to derive information about forests and show the results on a 3D virtual, interactive environment. Influenced by Persson et al (2005), voxelisation is an integral part of this research. The intensity profile of each full-waveform pulse is accumulated into a voxel array, building up a 3D density volume. The correlation between multiple pulses into a voxel representation produces a more accurate representation, which confers greater noise resistance and it further opens up possibilities of vertical interpretation of the data. The 3D density volume is then aligned with the hyperspectral images using a 2D grid similar to Warren et al (2014) and both datasets are used in visualisations and classifications.

Previous work in visualising fw LiDAR has used transparent objects and point clouds, while the output of this system is a coloured 3D-polygon representation, showing well-separated structures such as individual trees and greenhouses. The 3D density volume, generated from the fw LiDAR data, is polygonised using functional representation of object (FReps) and the marching cubes algorithm (Pasko and Savchenko, 1994) (Lorensen and Cline, 1987). Further, an optimisation algorithm is introduced that uses integral volumes (Crow, 1984) to speed up the process of polygonising the volume. This optimisation approach not only works on non-manifold object, but also a speed up of up to 51% was achieved. The polygon representation is also textured by projecting the hyperspectral images into the mesh. In addition, the output is suitable for direct rendering with commodity 3D-accelerated hardware, allowing smooth visualisation.

In future work, the effects of combining both hyperspectral imagery and fw LiDAR in classifications and visualisations are examined. At first, two pixel wise classifiers, a support vector machine and a Bayesian probabilistic model, will be used for testing the effects of the combination in generating tree coverage maps. Higher accuracy classification results are expected when metrics from both datasets are used together. Regarding the visualisations, the differences of applying surface reconstruction versus direct volumetric rendering will be discussed and an ordered tree structure with integral sums of the node values will be used for speeding up the ray-tracing of direct volumetric rendering and improving memory management of aforementioned optimisation algorithm with integral volumes. Further, deferred rendering is suggested for testing the visual human perception of projecting multiple bands of the hyperspectral images on the FW LiDAR

polygon representations. At the end of this project the combination of the datasets will be used along with the watershed algorithm for tree segmentation, which is useful for measuring the stem density of a forest and for tree species classifications.

from EDE:

Firstly, a new and fast way of aligning the FW LiDAR with Remotely Sensed Images has been developed in DASOS and by generating tree coverage maps it was shown that the combination of those datasets confers better remote survey results. This work was presented at the 36th ISRSE International Conference.

Secondly, automated detection of dead trees in native Australian forests has a significant role in protecting animals, which live in those trees and are close to extinction. DASOS allow the generation of 3D signatures characterising dead trees. A comparison between the discrete and FW LiDAR is performed to demonstrate the increased survey accuracy obtained when the FW LiDAR are used.

Finally, the last application is for improving visualisations for foresters. Foresters have a great knowledge about forests and can derive a wealth of information directly from visualisations of the remotely sensed data. This reduces the travelling time and cost of getting into the forests. This research optimises visualisations by using the new FW LiDAR representations and a speed of up to 51% has been achieved.

FW LiDAR has great potentials in forestry and this research has already started to have an impact in the FW LiDAR community by making those huge datasets easier to handle. DASOS is now used at Interpine Group Ltd, a world leading Forestry Company in New Zealand and it has been tested from a PhD student at Bournemouth University who looks into estimating bird distribution in the New Forest. In the future, it is expected that DASOS will be widely used in remote forest surveys (i.e. estimating the commercial value of a forest and detecting infected trees at early stages for treatment).

Acknowledgements

Above all, I would like to express my great gratitude to my industrial supervisors Dr. Michael Grant who had supported me continuously during my research and gave me the freedom to create a project of my own interest.

Then, I would like to thanks Dr. Matthew Brown, who helped me during my first years of my studies by giving me valuable and informative feedback. He was always there to keep me working on the right track.

Equally important is my current supervisor Dr. Neil D.F. Campbell and he is not to be missed from the acknowledgements.

Furthermore, special thanks are given to Dr. Mark Warren, Dr. Darren Cosker, MSc Susana Gonzalez Aracil and Dr. Ross Hill who occasionally advised me during my studies.

It further worth giving credits to my data providers, the Natural Environment Research Council's Airborne Research Facility (NERC ARF) and Interpine Group Ltd.

Last but not least, I am extremely grateful to my funding organisations, the Centre for Digital Entertainement and Plymouth Marine Laboratory, who supported financially and consequently made this research possible.

Abbreviations and Glossary

AGC	Automatic Gain Controller
ALS	Airborne Laser Scanning
APL	Airborne Processing Library
ARF	Airborne Research Facility
CG	Computer Graphics
CUDA	parallel computing platform available on nvidia graphic cards
DASOS	(=forest in Greek), the open source software implemented for managing FW LiDAR data
DEM	Digital Elevation Model
DTM	Digital Terrain Model (DTM)
FW	Full-Waveform
GB	Gigabyte
GPU	Graphics Processing Unit
LiDAR	Light Detection And Ranging
NASA	National Aeronautics and Space Administration
NDVI	Normalised Difference Vegetation Index
NERC	Natural Environment Research Council
NIR	Near-Infrared Region of the electromagnetic spectrum
TB	Terabyte
VIS	VIsual Spectrum
VLR	Variable Length Records
WPDF	Waveform Packet Descriptor Format
UK	United Kingdom

Publications

DASOS-User Guide, M. Miltiadou, N.D.F Campbell, M. Brown, S.C. Aracil, M.A. Warren, D. Clewley, D.Cosker, and M. Grant, Full-waveform LiDAR workshop at Interpine Group Ltd, Rotorua NZ, 2016

Alignment of Hyperspectral Imagery and Full-Waveform LiDAR data for visualisation and classification purposes, M. Miltiadou, M. A. Warren, M. Grant, and M. Brown, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 7, p. 1257, 2015.

Reconstruction of a 3D Polygon Representation from Full-Wavefrom LiDAR data, M. Miltiadou, M. Grant, M. Brown, M. Warren, and E. Carolan,*RSPSoc Annual Conference, New Sensors for a Changing World*, 2014.

Awards

EDE and Ravenscroft Prize - Finalist: Selected as one of the five finalists for this is a prestigious prize that recognises the work of best postgraduate researchers.

Student Poster Competition at Silvilaser.

Conference Presentations

ForestSAT Conference, Santiago, Chile, 2016 - Oral and Poster Presentation

Computer Graphics & Visual Computing (CGVC), Bournemouth, United Kingdom, 2016 - Poster Presentation

Silvilaser, La Grant Motte, France, 2015 - Oral Presentation

International Symposium of Remote Sensing of the Environment (ISRSE), 2015 - Oral Presentation

RSPSoc Conference, New Sensors for a Changing world, Aberystwyth, United Kingdom, 2014 - Oral Presentation

Contents

List of Figures

Chapter 1

Introduction

1.1 Forest Monitoring: Importance and Applications

Forest monitoring involves checking and observing the changes in the structure of the forests and their foliage over the years. It has a significant value in both sustainable and commercial forests, because it contributes into managing biodiversity, maintaining forest health and optimising wood trade procedures as explained below:

- **Biodiversity** plays a substantial role in ecosystem resilience [1] while various human activities affect biological communities by altering their composition and leading species to extinction [2]. For example, in Australian native forests many arboreal mammals and birds rely on hollow trees for shelters [3] . Hollow trees are trees that have hollows, which are semi-enclosed cavity on trunks and branches. They are formed by natural forces, like bacteria, fungi and insects and it takes hundreds of years to become suitable for animal/bird shelters. Unfortunately recent studies shown that there us likely to be a shortage of hollows available for colonisation in the near future [4] [5]. Therefore monitoring and protecting hollow trees have a positive impact in preserving biodiversity.

- **Forest Health**: Protecting vegetation from pests and diseases. An example of pests are the Brushtail Possums, which were initially brought to New Zealand for fur trade, but they have escaped and become a thread to native forests and vegetation [6]. In addition, anthropogenic factors have a negative impact to nature. For instance, acid rain is responsible for the freezing decease at red bruces because it reduces the membrane-associated calcium, which is important for tolerating cold [7]. Those changes in nature need to be monitor in order to preserve a healthy and resilience ecosystem.

- **Wood Trade**: Measuring stem volume and basal areas of trees contributes to forest planning and management [8]. For example, measuring stocking and wood quality would help into estimating the cost of harvesting the trees in relation to the stocking [9].

Traditionally, forest monitoring involves field work such as travelling into the area of interest and taking manual measurements. Regarding the need to monitor hollows, tree climbing with ladders and ropes gives very accurate results but it's dangerous, expensive, time consuming, and cannot easily scale into large forested areas [10] [11]. Therefore, automated ways of monitoring forests are essential and this is why Remote Sensing has a significantly positive impact in forestry.

1.2 Background Information about Remote Sensing and Airborne Laser Scanning Systems

Remote sensing refers to the acquisition of information about objects, for example vegetation and archaeological monuments, without physical contact and the interpretation of that information. The sensors used to capture the information are divided into passive and active. For example satellite photography is passive because information are collected from the reflected natural sun light, while Airborne Laser Scanners (ALS) are active because they emit laser beams and collects information from the backscattered laser energy [12].

According to Wanger et al, Airborne Laser Scanning (ALS) is a growing technology used in environmental research to collect information about the earth like vegetation and tree species. Comparing ALS with traditional photography, ALS is not influenced by light and it is therefore less dependant on weather conditions (ie. it collects information from below the clouds). The laser beam further penetrates the tree canopies allowing it to record information about the forest structure below the canopy, as well as the ground [13]. ALS methods are divided into pulse systems, which repeatedly emit pulses, and continuous wavelength systems that continuously emit light. They both acquire information from the backscattered laser intensity over time, but continuous wavelength systems are more complicated because they obtain one extra physical parameter, the frequency of the ranging signal. Further, according to Wehr and Lohr, continuous wavelength systems are 85 times less accurate than pulse systems [14].

LiDAR (Light Detection And Ranging) systems are passive and pulse laser scanning systems [14]. They are divided into two groups according to the diameter of the footprint left by the laser beam on the ground. This diameter depends on the beam divergence and the distance between the sensor and the target. The small-footprint systems have a 0.2-3m diameter, have been widely commercialised and are mostly carried on planes (ALS systems). In contrast, the large-footprint systems have a wider diameter (10-70m) and during experiments they were mostly adjusted on satellites. Small-footprint systems record at higher resolution but it cannot guarantee that every pulse will reach the ground due to the small diameter of their footprint, making topographic measurements difficult. In contrast, large-footprint scanners have wider diameters and can therefore scan wider areas with the likelihood of recording the ground to be higher [15] .

In addition, there are two types of LiDAR data, the discrete and the full-waveform (FW). The discrete LiDAR records a few peaks of the reflected laser intensity, while the FW LiDAR stores the entire backscattered signal. The discrete LiDAR has been widely used and a 40% reduction of fieldwork has been achieved at Interpine Ltd Group, New Zealand, with that technology. Regarding the FW LiDAR, scientists understand their concepts and potentials but due to the shortage of available tools able to handle these large datasets, there are very few uses of FW LiDAR [16].

The design of the first FW LiDAR system was introduced in 1980s, but the first operational system was developed by NASA in 1999 [17]. The increased amount of information recorded within the FW LiDAR suggests many new possibilities and problems from the point of view of image understanding, remote surveying and visualisation. As an indication, a 9.3GB discrete LiDAR from New Forest, UK, corresponds to 55.7GB of FW LiDAR.

This research is focused on the representations of the FW LiDAR data and contributes in both forestry visualisations and classifications. Two datasets are used for testing and evaluation: the New Forest and the RedGum dataset. An in depth explanation of LiDAR systems and the specifications, differences and challenges of the two dataset are given in and Section 2. An overview of the thesis along with its aims, objectives and contributions are then outlined at Section 3.

Chapter 2

Acquire Data

The aim of this section is to give a practical and scientific insight into the acquisition of data, because a good knowledge of these methods and their limitations is essential for understanding the related research undertaken. The relations between the two main datasets used in this project are depicted on Figure 2-1 and briefly explained here:

- The **New Forest dataset** from the UK was provided by the Natural Environment Research Council's Airborne Research Facility (NERC ARF). Measurements were collected simultaneously a Leica ALS50-II LIDAR and AISA Eagle/Hawk hyperspectral radiometers on the 8th of April in 2010. It contains Discrete LiDAR, FW LiDAR and hyperspectral images.

- The **RedGum dataset** was acquired in Australia using a Trimble AX60 integrated LIDAR/Camera instrument over the time period from the 6th of March in 2015 until the 31st of March in 2015. It was provided by the RPS Australia East Pty Ltd. Only the FW LiDAR data are used here.

The ALS data are explained first, because they are the main focus of this research, and hyperspectral imagery is towards the end of the chapter. In Section 2.1, an in-depth description of ALS systems and the differences between discrete and FW LiDAR data is given. Section 2.2 briefly discusses the binary file format of the acquired LiDAR data and Section 2.3 is a discussion on the limitations, the differences and the advantages of two LiDAR instruments; the Leica and Trimble. The essential information about the hyperspectral imagery, which is only associated with the New Forest dataset, is then covered in Section 2.4.

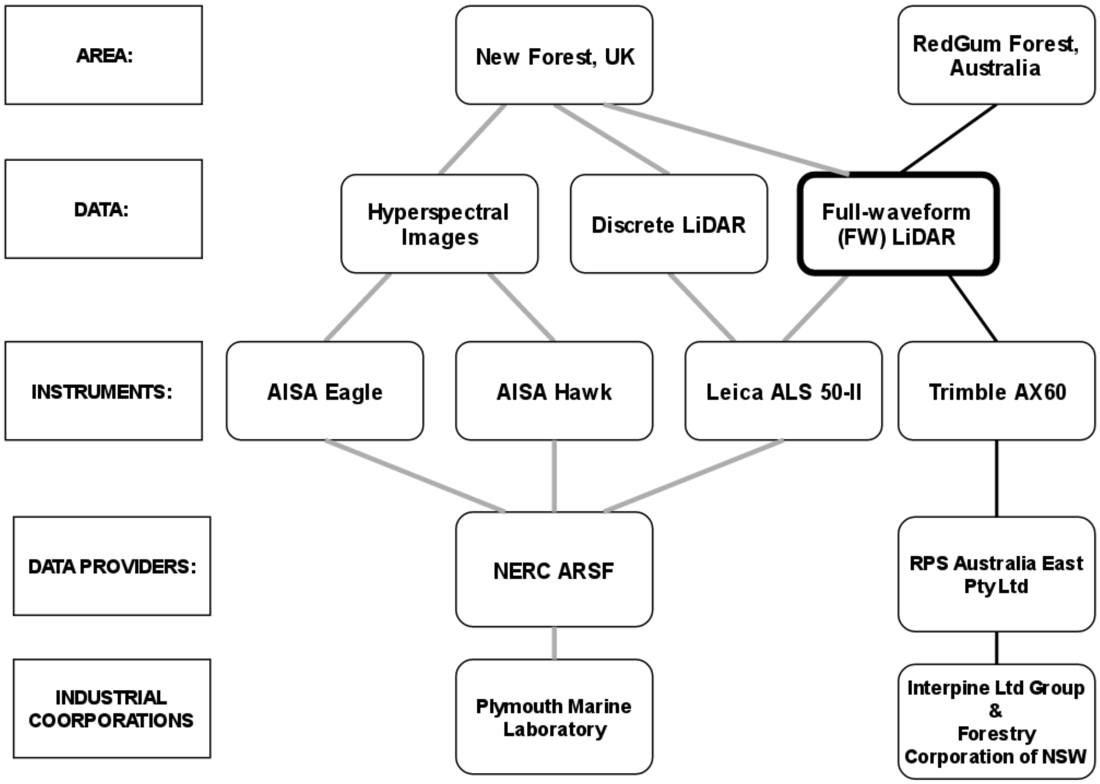
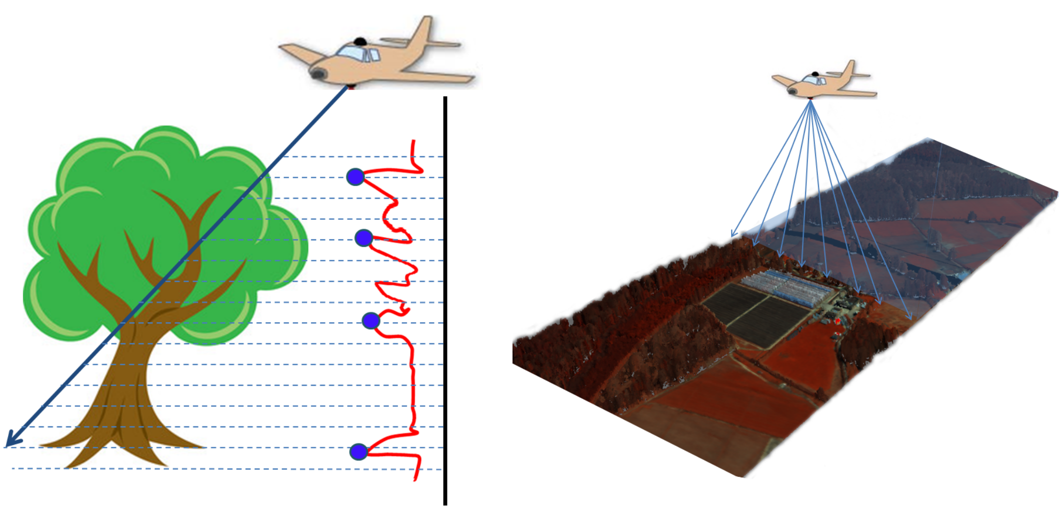


Figure 2-1: Data and Instruments



(a) A Pulse: Discrete Vs Full-Waveform

(b) A Flightline

Figure 2-2: Airborne Laser Scanning System

2.1 Airborne LiDAR systems: An in-depth Explanation

The ALS systems emit laser pulses from sensor mounted in a plane and collects information from the time-of-flight and the returned laser intensity. By the time the pulse has travelled the approximately 1-3km from the aircraft to the ground, it is roughly 20cm in width due to beam divergence. When the pulse hits an object (e.g. the forest canopy), then some of it reflects back while the rest penetrates through holes between leaves and branches. The laser pulse continues to hit structures, scattering and partially returning to the sensor until it reaches a solid barrier such as the ground and is fully blocked from further progress. The LiDAR systems record information from the backscattered laser pulse, measuring its round trip time and the returned intensity.

As mentioned at Section 1.2, there are two types of LiDAR data, discrete and FW. The discrete LiDAR observes the returned intensity signals and identifies and records a few peak intensity returns of the signal, while the FW LiDAR system digitises and stores the enitre backscattered signal into equally spaced time intervals (Figure 2-2a). The delivered data for the discrete LiDAR is a set of hit points ("returns"), which are associated with laser intensities. The world position of every return is calculated by measuring the round trip time of the laser return, giving a distance from the sensor, which is combined with the precisely known position and orientation of the aircraft/sensor (from GPS, an inertial measurement unit and precise shot direction of the laser pulse). The waveform recordings are triggered by and attached to first returns of discrete LiDAR data (to avoid sampling the uninteresting time period while the pulse travels through the atmosphere) and they are a list of intensities that correspond to the laser intensity returned over time. There is also an offset vector which defines the distance and direction between each wave sample (effectively a compression mechanism, by avoid recording the world position of every sample, replacing it with the location of the first return and this vector).

As shown in Figure 2-2b, the pulses are scanned back and forth across the landscape below (by a rotating mirror) as the plane travels forward. The scanned data has a limited maximum width according to the flight height and the field of view scan angle. During processing the track of the plane is divided into easier-to-handle pieces (flightlines) and saved into separate binary files. In this project the LAS1.3 file format is used for both datasets.

2.2 Brief Description of the LAS1.3 File Format

There are a few LiDAR file formats but the LAS1.3 was the first format to contain FW data and it is the one used to store the data for both New Forest and RedGum datasets. According to the LAS1.3 file specifications [18], a .LAS file contains information about both discrete and FW LiDAR data, with the waveform packets are attached to discrete returns and saved either internally at the end of the .LAS file or externally in a .WVS file.

As shown at (Figure 2-3) the .LAS file is divided into four sections and a brief explanation of each section is given here:

1. The **Header** contains general information about the entire flightline. For example, it includes the maximum scan angle used during the flight, whether the waveform packets are recorded internally or externally and the number of **Variable Length Records** (VLR).

2. Regarding the **VLR**, which contain arbitrary "extension" data blocks, the most important information given is the waveform packet descriptors that contain essential information on how to read the waveform packets (i.e. an ID, the number of wave samples and the size of each intensity in bits).

3. The **Point Data Records** are the discrete points and the waveforms are associated with first return discrete points. Each Point Data Record has a spatial location, an intensity and optionally a pointer to a waveform packet as well as the ID of the corresponding waveform packet descriptor.

4. The waveform packets is a list of intensities and they are either saved internally into the **Extended Variable Length Records** section of the .LAS file or inside an external .WVS file. Starting from the associated first return point, the spatial locations of the waveform packet (wave sample intensity) are calculated by adding an offset defined in the associated Point Data Record.

2.3 Leica Vs Trimble Instruments: Limitations, Differences and Advantages

As shown in Figure 2-1, the Leica ALS 50-II instrument was used to capture the LiDAR data of New Forest dataset and the Trimble AX60 for collecting the RedGum Forest FW LiDAR data. It is therefore important to clarify the differences, the limitations and the advantages of each instrument.

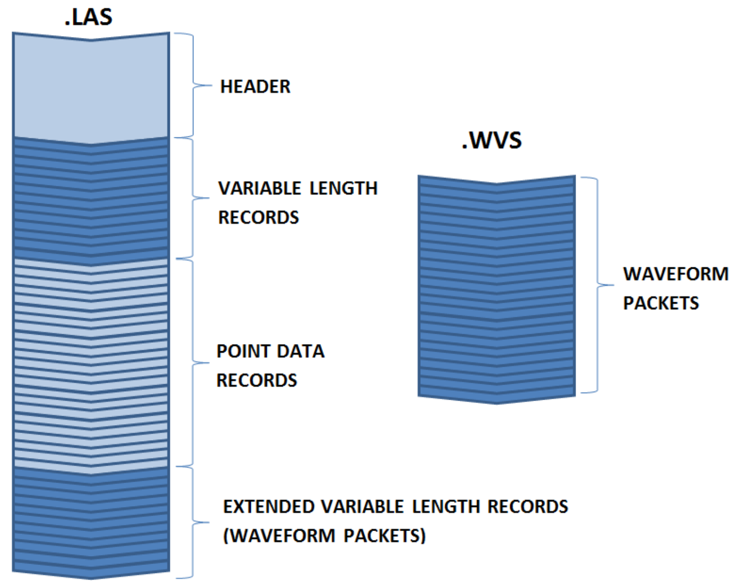


Figure 2-3: How the FW LiDAR data are stored into a binary file, according to the LAS1.3 file format specification

The Trimble performs at a pulse frequency of 400kHz, while the Leica's maximum pulse frequency is 120kHz. Nevertheless, during experiments there were occasions when the Leica discarded every other waveform due to I/O limitations despite being at or below the maximum pulse frequency [19]. The New Forest dataset has been affected by this and, on average, one third of the saved pulses only contain discrete data. We should therefore be extremely careful when comparing Discrete with FW LiDAR data. While [16] concludes that FW LiDAR data worth the extra processing because they have a better vertical profile, [20] states that extra information (the echo-width) from the FW LiDAR data are relatively unimportant. But the New Forest datasets were used for the comparison at [20] and there is no mention about the significantly less waveforms recorded in comparison to the discrete data. It is therefore suspected that their results has been affected by the missing waveforms.

Another problem with the Leica sysyem is the small dynamic range of intensities due to the number of bits used for recording them; the Leica system uses 8-bit integers (0-255 range) while the Trimble uses 16-bit integers (0-16385 range). For increased dynamic range and finer intensities without doubling storage costs, Leica introduced an Automatic Gain Controller (AGC). The AGC is an 8-bit number that defines how the recorded intensity range is shifted across a wider range of intensities. The AGC value is adjusted according to the reflected laser intensity of the previous 64 pulses and it therefore varies across a flightline. Consequently, the raw intensities are incomparable to each other and, since the relation between AGC and the intensities is not linear, the range normalisation is complicated [21] [22]. In this thesis, the intensities of the Leica system are used as boolean values (whether something existed or not, using a user-defined threshold) to quickly overcome that issue and focus on the major research objectives. Regarding the Trimble instrument, there is no AGC value because the intensities are saved into a 16bit integer and as long as the flight height is constant no normalisation is required. In a few words, the raw intensities recorded using the Leica system are not normalised and therefore not comparable to each other, while the intensities of the Trimble instrument are more meaningful.

The footprint of the laser on the ground depends on the scanning pattern of the instruments and the field of view. The sinusoidal scanning pattern of the Leica system results in a higher density of returns at the edges of the flightline. The footprint of the Trimble instrument is more equally spaced because they are scanned using a rotating polygon. The uneven density pattern of the Leica system is resolved by normalisation during the voxelisation process, but the Trimble's equally spaced pulse pattern is more prone to aliasing when voxelised. Regarding the field of view, the Leica is wider but both systems avoid large angles because otherwise data look deformed at edges of the flightlines.

Last but not least, the Trimble instrument is a native full-waveform sensor; the discrete LiDAR are produced by extracting peak points in post-processing. Therefore one of the purported advantages of a FW system, the concept of extracting a denser point clouds using Gaussian decomposition [13], does not apply in the Trimble's case. This was proven by extracting peak points from Trimble FW LiDAR data using the pulseextract from LAStools [23]; the number of points extracted was exactly the same as the number of points saved into the associated discrete LiDAR files. Therefore discrete data from the Trimble instrument are the same as those generated by echo decomposition and peak points extraction from the FW samples.

To sum up, the Trimble AX60 instrument is a newer sensor and therefore has less problems or design compromises in comparison to the Leica ALS50-II instrument. Table 2.1 summarises the differences between the two sensors.

2.4 Hyperspectral Imagery

Hyperspectral imagery has a positive impact in remote sensing because it contains information beyond human visibility. The human eye receives light from the visual spectrum into three bands (red, green and blue). The hyperspectral sensors captures a larger spectrum and divides its light components into hundreds of bands, recording

Table 2.1: Specifications of the LiDAR instruments used

Instrument Name:	**Leica ALS550-II**	**Trimble Ax60**
Scanned Area	New Forest, UK	RedGum, Australia
Year of Introduction:	Discrete LiDAR 2009 & FW LiDAR 2010	2013
Max Scan Frequency (kHz):	120	400
Recorded Intensity (bits):	8	16
AGC:	Yes	No
Scanning Pattern:	Sinusoidal	The footprints are more equally spaced on the ground
Max field of view (degrees):	75	60

this way more information than a human eye can receive [12].

Nevertheless, there are other compromises - for example, the time taken to integrate incoming light as the aircraft carrying the sensors moves. This means the raw airborne images appear deformed because the pixel length varies across the flightline. NERC-ARF geo-corrects the data using the Airborne Processing Library (APL) [24]. The processing levels are numbered. At 'level 3' (world coordinate system) the pixels are equally spaced and sized, which requires resampling and thus may look slightly blurred. The 'level 1' data (what the sensor saw) are non geo-corrected but they are associated with a file that defines the spatial location of each pixel. In this thesis, the 'level 1' data are used to preserve the highest possible quality.

In practise, the level 1 data are held in two files, the '.bil' and the '.igm'. The '.bil' file contains the hyperpsectral cube (Figure 2-4), all the pixel values at different wavelengths, and the .igm file gives the x, y, z coordinates of each pixel.

The number of bands and the spectrum range captured depends on the hyperspectral sensor. The data from New Forest were collected using the following instruments:

- the Eagle, which captures the visible and near infra-red spectrum (400-970nm)
- the Hawk, which covers short wave infra-red wavelengths (970-2450nm)

Both sensors divide their spectral range into 252 bands (programmable) and each band is a 2D vector as shown in Figure 2-4).

The hyperspectral images also come with a number of drawbacks. A few are mentioned here but since hyperpsectral imagery is not the main focused of the thesis there are not addressed:

- System faults sometimes occurs and the affected areas are masked out. This results in blank areas.

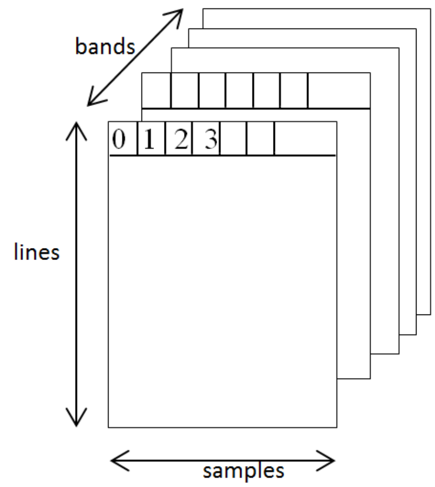


Figure 2-4: This figure shows the order of the hyperspectral pixels saved into the the binary .bil file.

- As a passive sensor, it is dependent on the sun for illumination and thus vulnerable to poor weather conditions
- Due to the high refraction of light at some wavelengths, some bands are highly influenced by humidity (i.e. wavelength 1898.33nm).

To sum up, hyperpsectral images contain information beyond the visible and they are delivered in two files, one contains the hyperspectral cube and the other one the geo-locations of each pixel. In this project, they are used in chapter (Chapter 7), where it is shown that the combination of Remote Sensing data confers better results for generating tree coverage maps.

Chapter 3

Overview of Thesis

3.1 Problem

FW LiDAR systems have been available for a number of years but there still very few uses of FW LiDAR data. NERC-ARF has been acquiring airborne data for the UK and overseas since 2010 and it has more than 100 clients of new and archived data. Many clients request FW LiDAR data to be acquired, but despite the significant number of requests,the majority of research still only uses discrete LIDAR. Some of the factors regarding this slow intakes are:

- Typically FW datasets are 5 – 10 times larger than discrete data, with data sizes in the range of 50GB – 2.5TB GB for a single area of interest. NERC-ARF's datasets are up to 100GB each because most clients are research institutes but for commercial purposes each FW dataset is a couple of TB.
- Existing workflows are only able to work with the discrete data since the increased amount of information recorded within the FW LiDAR makes handling the quantity of data very challenging.

3.2 Aims and Objectives

This thesis explores visualisation and data-understanding for FW LIDAR systems and the overarching aim is to increase the accessibility FW LiDAR in remote forest surveying.

The objectives are the following: NEILL: Could perhaps provide more structure to these points by dividing into general aims and specific examples/applications? I would also number these so that you can refer to the objectives later on.

- Enable forestry experts with no computer science expertise to visualise and work with the FW LiDAR data.
- Enable forest understanding through 3D visualisations of FW LiDAR data.
- Improve and optimise visualisations of FW LiDAR data and hyperspectral images.
- Enable browsing of very large scale datasets and spectral bands in an efficient manner.
- Investigate data structures for faster iso-surface extraction of large volumetric datasets and efficient management of voxels.
- Estimate tree coverage and investigate the potential of integrating multiple remote sensing datasets in forestry.
- Dead tree detection in comparison to human detection and remote surveying with discrete LiDAR that will benefits biodiversity management.
- Research whether terrain classification can be improved by the inference of high quality 3D information, for example, using priors over the space of 3D elements.

3.3 Overview

*** the following text has been taken from the IAA2 funding application

To address the limitations of existing workflows for using with FW data we developed the open source software DASOS (from $\delta\alpha\sigma o\varsigma$ meaning forest in Greek) and novel algorithms that allow users, without computer science expertise, to work with and visualise large volumes of FW LiDAR data. Our open source software DASOS aims to remove the barriers preventing the use of FW LiDAR. Its contributions, and those of the new representations of the FW LiDAR, are demonstrated in three applications:

- Firstly, foresters can exploit their domain expertise to derive a wealth of information by observing the FW LiDAR data. We therefore improve visualisations for deriving information directly from the data, thus reducing travelling time and the associated expenses of getting into the forests. This cost includes appropriate cars and sometimes helicopters depending on the accessibility of the forests. While previous work on FW LiDAR visualisation talks about point cloud visualisation [25] and transparent voxels [26], DASOS is able to reconstruct the surfaces from the scanned area in 3D. This research further optimisesvisualisations by using the new FW LiDAR representations to accelerate this process by ****%. *** I will complete the percentage once related test are completed

- Secondly, a fast way of aligning the FW LiDAR with Remotely Sensed Images has been developed in DASOS. Subsequently, by generating tree coverage maps, it has been shown that the combination of these datasets confers better remote survey results [27].

- Finally, DASOS allows the generation of 3D priors. An example usage of this information is characterising dead standing Eucalyptuses, which as explained at Section **??** are extremely beneficial for managing biodiversity in native Australian forests. This is work in progress and a comparison between the discrete and FW LiDAR will be performed to demonstrate the increased survey accuracy obtained when the FW LiDAR is used.

In summary, FW LiDAR has great potential to improving automated surveying accuracy and consequently reduce the expensive fieldwork conducted in forestry and this research has already started to have an impact in the FW LiDAR community. DASOS is now used at Interpine Group Ltd, a world leading Forestry Company in New Zealand, and a PhD student at Bournemouth University is evaluating it for use in the estimation of bird distributions in the New Forest in the UK.

*** end of copied text

3.4 Thesis Structure

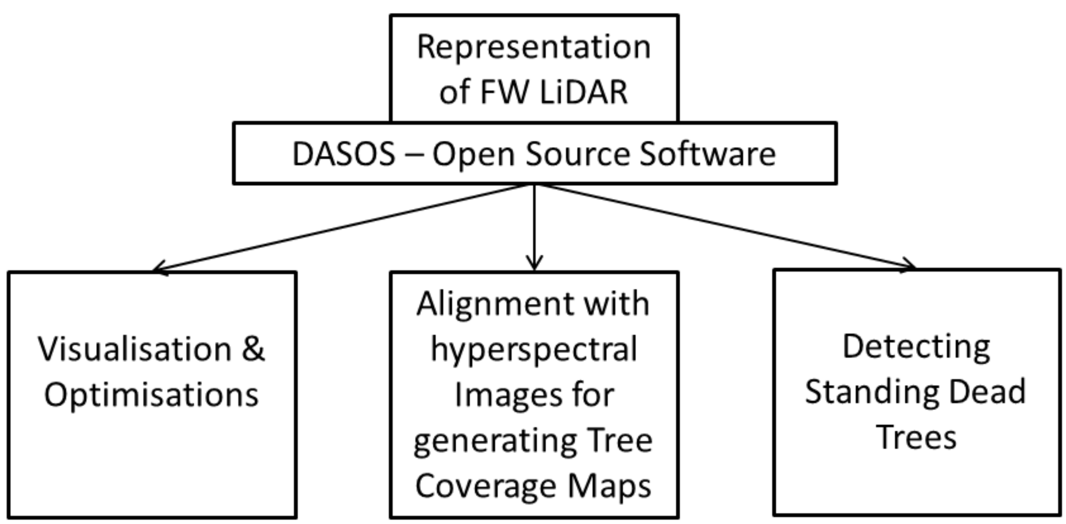


Figure 3-1: The pipeline of the thesis

Chapter 4

The open source software DASOS and the Voxelisation Approach

As mentioned at Section 3.1, there are very few uses of FW LiDAR data because of the quantity of the recorded information. For that reason, DASOS was developed DASOS (Section 4.3), as an open source software, to help foresters without computer science background to use FW LiDAR data. In this section:

- An overview of related software packages is given and we explain how DASOS differs from those packages (Section 4.1).
- The main method of interpreting the data within DASOS (the voxelisation approach) is described (Subsection 4.2).
- Then all the functionalities of DASOS are listed (Section 4.3)
- and finally a summary is provided (Section 4.4).

4.1 State-of-Art FW LiDAR Software Packages

The most common approach of interpreting the FW LiDAR is the Gaussian decomposition of the waveforms for peak points extraction. Each waveform is modelled as a set of Gaussian pulses and for every Guassian peak, a discrete LiDAR point is extracted [28]. Neunschwander et al used this approach for Landcover classification [29] while Reitberger et al applied it for distinguishing deciduous trees from coniferous trees [30]. Chauve et al further proposed an approach of improving the Gaussian model in order to increase the density of the points extracted from the data and consequently improve point based classifications of FW LiDAR data [17]. The following tools are able to extract discrete points from the waveforms and visualise small areas of interest:

- **Pulsewaves**: visualise a small number of waveforms using different transparencies according to the intensities of the wave-samples and are able to generate discrete point clouds [25].
  Link: <https://rapidlasso.com/pulsewaves/>

- **FullAnalyze**: supports echo decomposition. Regarding visualisations the user can select single trees from the Graphical User Interface (GUI) and for each wave-sample, a sphere with radius proportional to its amplitude is created and visualised [31].
  Link: <http://fullanalyze.sourceforge.net/>

- **SPDlib**: exports discrete LiDAR and visualises either the samples that are above a threshold level as points or the extracted discrete point cloud. It also colours them according to their intensity value [32].
  Link: <http://www.spdlib.org/>

Echo decomposition and extraction of peak points identifies significant features and further enables the interpretation of the data within existing workflows and software that support discrete LiDAR data. For example, the discrete LiDAR can be analysed using:

- **Lag**: a visualisation tool for analysing and inspecting discrete LiDAR point clouds.
  Link: <http://arsf.github.io/lag/>

- **Quick Terrain Modeller** : a 3D discrete LiDAR points visualiser, that can generate Digital Elevation Models (DEM) and Digital Terrain Models (DTM).
  Link: <http://appliedimagery.com/>

- **LAStools** : a tool set that classifies noise, visualises point clouds, clips data.
  Link <https://rapidlasso.com/lastools/>

DASOS approach of interpreting FW LiDAR data is fundamentally different from the aforementioned software packages. On the one hand, converting FW LiDAR into discrete, their usage is ease since existing overflows support discrete LiDAR, but on the other hand FW LiDAR contain information about pulse width that are not preserved after peak point extraction. Also the comparison of point clouds depends on the density of the emitted pulses; problems arise with the sinusoidal pattern of the Leica system. For that reason, in DASOS, this information is accumulated from multiple shots into a voxel array, building up a 3D density volume. The correlation between multiple pulses in a voxel representation produces a more accurate and complete representation, which confers greater noise resistance and it further opens up possibilities of vertical interpretation of the data. The idea of voxelising FW LiDAR data is explained in the following section 4.2.

4.2 Voxelisation for Interpreting FW LiDAR data

Voxelisation of FW LiDAR data was firstly introduced by Persson et al., who used it to visualise waveforms using different transparencies [26] and it has been adopted as the future of FW LiDAR data with the literature moving toward that direction. In 2016, Cao et al used it for tree species identification [33] and Sunmall et al characterised forest canopy from a voxelised vertical profile [20]. The innovative approach of voxelising the FW LiDAR data is an integral part of this thesis and it is used for both visualisations and classifications [34] [27].

The FW LiDAR data are voxelised by inserting the wave samples into a 3D regular grid and constructing a 3D discrete density volume. According to Persson et al, each wave sample is associated with the 3D cell, named voxel, that it lies inside. If multiple samples lie inside a voxel then the sample with the highest intensity is chosen [26]. In order to reduce noise, there are two differences between this approach and the way FW LiDAR data are voxelised in DASOS.

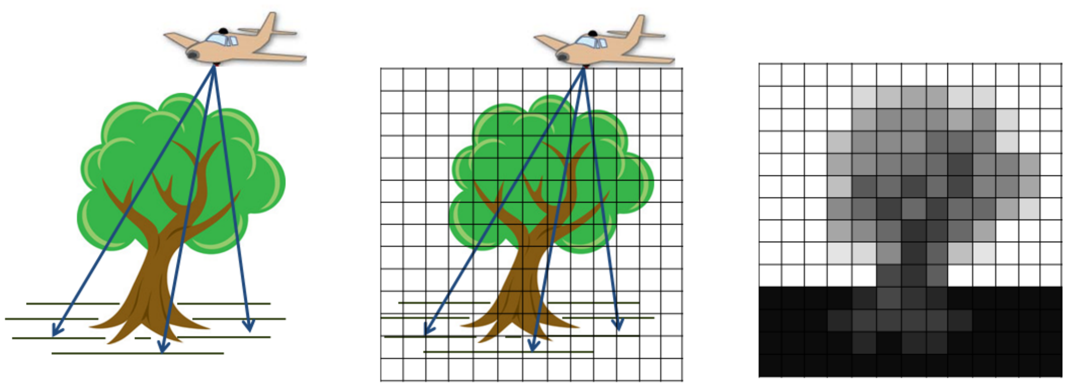
At first a threshold is used to remove low level noise, because when the width of a recorded waveform is longer that the distance between the first hit point and the ground, the system captures low signals, which are pure noise. For that reason, the samples whose intensity is lower than a user-defined noise level/threshold are discarded.

Then each wave sample is associated with the voxel that it lies inside and the second difference is how DASOS overcomes the uneven number of samples per voxels. The intensity of each sample is the laser intensity returned during the corresponding time interval. For example, if 5 samples are inside a voxel and the waveform is digitised at 2ns, then the laser intensity associated with that voxel corresponds to a 10ns waveform width. For comparison purposes, it's essential to keep the waveform width consistent across the voxels. For overcoming this issue in DASOS, the average intensity of the samples that lie inside each voxel is taken, instead of choosing the one with the highest intensity [26]. This way the likelihood of the 3D volume to be affected by outliers and high noise is reduced. The following equation shows how the intensity value of a voxel is calculated:

$$I_v = \frac{\sum_{i=1}^{n} I_i}{n} \tag{4.1}$$

where I_v is the accumulated intensity of voxel v, n is number of samples associated with that voxel and I_i is the intensity of the sample i.

To sum up, during voxelisation the area of interest is divided into voxels. The samples of the FW LiDAR data are inserted inside this 3D discrete density volume and normalised such that equally sized waveform width is saved inside each voxel. The result is a 3D discrete density volume of the scanned area. Figure 4-1 depicts this process in 2D.



(a) The sensor from the plane emits multiple pulses and collects information from the returned laser intensity.

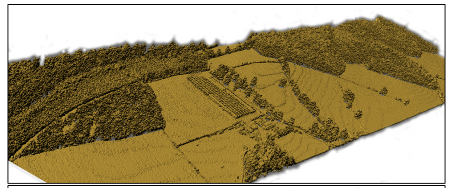
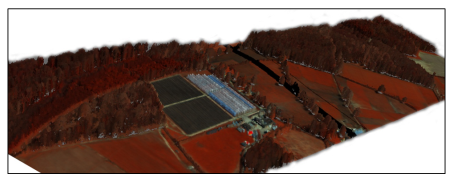
(b) The area of interest is divided into equally sized cubes, named voxels, generating this way a discrete volume.

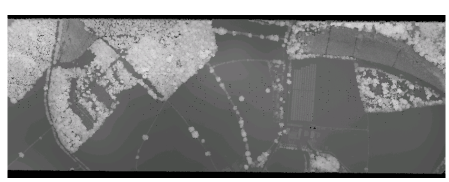
(c) The accumulated intensities of wave samples into the volume build up the voxelised representation of the scanned area.

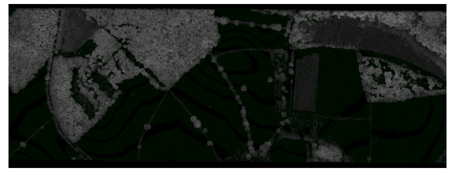
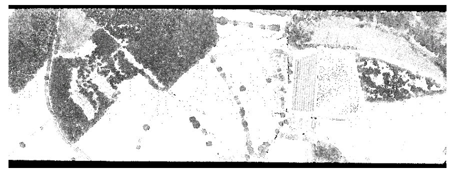
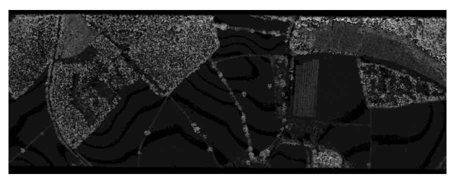
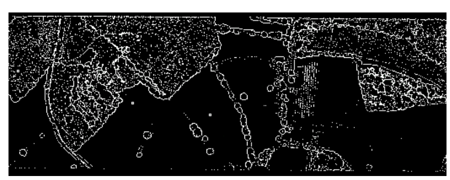
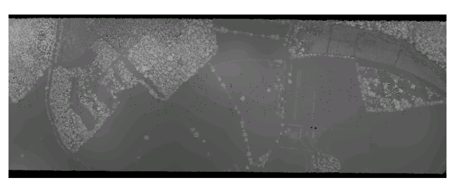
Figure 4-1: The above images depict the voxelisation process of the FW LiDAR data in 2D. Please note that the voxelisation output in Figure 4-1c shows how ideally the result would look. But in reality, a number of trees may be disconnected from the ground due to missing information about their trunk.

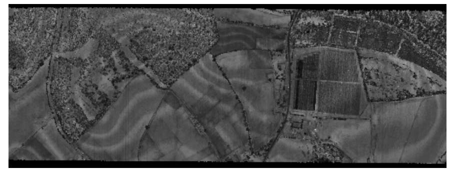
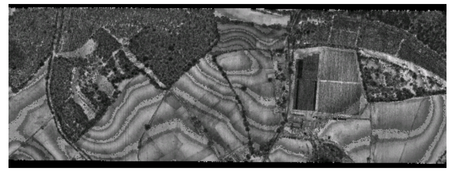
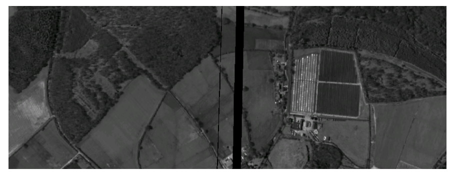
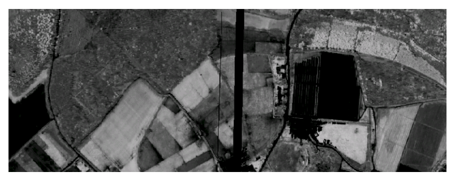
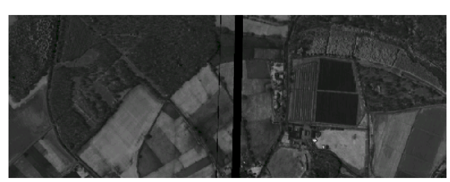
4.3 The functionalities of DASOS

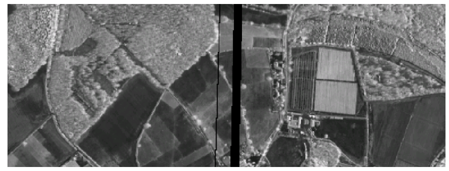
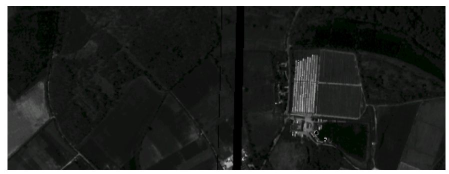
So far, an overview of existing software packages supporting FW LiDAR was given (Section 4.1) and it was explained how DASOS differs from them by voxelising the waveforms (Section 4.2). In this seciton, the three main the functionalities of DASOS are described in table 4.1.

1st Functionality: 3D Polygon Mesh Generation			
Input	**Description**	**Output Example**	**Output Format**
LAS1.3	**3D Polygon Mesh** Constructed from the volumetric representation (algorithms and user-defined parameters are explained in Section 5 while optimisation approaches are discussed in Section 6)		.obj
LAS1.3 and level 1 (.bil & .igm	**3D Coloured Polygon Mesh** Projecting 3 user-defined hyperspectral bands on the mesh (Section 7)		.obj & .png

2nd Functionality: Generation of 2D metrics aligned with hyperspectral imagery			
In Section 7 a selection of the following metrics are used for generating tree coverage maps			
Input	**Metric Description** (L for LiDAR metrics & H for hyperspectral metrics)	**Output Example**	**Output Format**
LAS1.3	**L0 - Height:** The distance between the top non-empty voxel and the lower boundaries of the volume.		.asc

LAS1.3	**L1 - Thickness:** The distance between the first and last non empty voxels in every column of the 3D volume.		.asc
LAS1.3	**L2 - Density:** Number of non-empty voxel over all voxels within the range from the first to last non-empty voxels.		.asc
LAS1.3	**L3 - First Patch:** The number of non-empty adjacent voxels, starting from the first/top non-empty voxel in that column.		.asc
LAS1.3	**L4: Last Patch:** The number of non-empty adjacent voxels, starting from the last/lower non-empty voxel in that column.		.asc
LAS1.3	**L5 - Edge detection:** The average height difference of neighbouring pixels.		.asc
LAS1.3	**L6: Lowest Return** The height of the lowest non empty voxel (the actual heights are very low and close to each but the example image has been scaled and the different seems bigger)		.asc

LAS1.3	**L7: Maximum Intensity** The maximum intensity of each column		.asc
LAS1.3	**L8: Average Intensity** The average intensity per column		.asc
LAS1.3 and level 1 (.bil & .igm)	**H0 : Mean** The mean of the hyperspectral spectrum.		.asc
LAS1.3 and level 1 (.bil & .igm	**H1: NDVI** The Normalised Difference Vegetation Index indicates whether green vegetation exists or not and it is derived from the electromagnetic spectrum as follow: $$NDVI = \frac{NIR - VIS}{NIR + VIS} \quad (4.2)$$ where the NIR is the near-infrared region of the spectrum (700-2500nm) and VIS is the Visible/Visual spectrum (430-770) [35].		.asc
LAS1.3 and level 1 (.bil & .igm	**H2: Standard Deviation** [1] The standard deviation of the hyperspectral spectrum at each pixel.		.asc

LAS1.3 and level 1 (.bil & .igm	**H3: Spectral Signature** [1] The squared spectral difference between each pixels' spectrum and the generalised vegetation signature retrieved from USGS Digital Spectral Library [36].		.asc
LAS1.3 and level 1 (.bil & .igm	**H4: Band** A single user defined hyperspectral band.	 	.asc .asc

3rd Functionality: 3D Priors / Signatures			
In Section 8, the 3D priors/templates are run over the volume for detecting dead standing trees			
Input	**Description**	**Output Example**	**Output Format**
LAS1.3	**3D Templates**	***	.csv
LAS1.3	**3D**	***	.csv

Table 4.1: The three functionalities of DASOS

In this sections an in depth overview of DASOS's functionalities was given (Table 4.1). Each functionality is linked to a number of Sections, which describes the algorithms implemented and related applications. In a few words, the 3D visualisations are useful in foresty for reducing fielwork and improving planning of field trips (i.e. checking

[1] Those two metrics were implemented specifically for the tree coverage maps [27] and they are not available on the released version of DASOS.

whether a road is passing through a fieldplot area). The 2D metrics allow simultaneous interepretation of FW LiDAR data and hyperspectral imagery. They could also be used in GIS softwares. In this thesis, they are used for generating tree coverage maps. Last but not least the priors enables 3D feature detection and they are used for detecting dead standing trees.

It further worth stating that the up-to-date information about DASOS are provided at: `<http://miltomiltiadou.blogspot.co.uk/2015/03/las13vis.html>` This link also indicates how to download DASOS, the complete user-guide and the source code, as well as where to seek for support while using it.

4.4 Summary and Discussion

Along with that thesis, the open source software DASOS was developed to encourage foresters to use the FW LiDAR data. The main way of interpreting FW LiDAR data in DASOS is fundamentally different from the state-of-art available software packages. In a few words, the FW LiDAR data are voxelised by inserting the wave samples into a 3D discrete density volume, which preserves an extra parameter (the pulse width) in comparison to point extraction algorithms. It also accumulates intensity values from multiple shots and stores them into a 3D regular grid, resolving this way the problem with the sinusoidal footprints pattern of the Leica system.

Furthermore, there are three main functionalities of DASOS: the construction of 3D polygon meshes, the generation of 2D metrics aligned with hyperspectral Images and characterisation of objects using 3D priors/signatures. The visualisation outputs are also state-of-art since previous visualisations talk about points [32] or spheres [31], while DASOS is able to create closed polygon representation. In addition, the integration of various sensors allows simultaneous interpretation of their data and in Section 7, it is shown that this confers better results for generating tree coverage maps. The 3D priors allows local inspection of data and Section 8 used them for dead standing tree detection in native Australian forests.

Finally, it worth mentioning that there a few individuals/organisation that showed interest in using DASOS and in the future, DASOS usage is expected to increased in remote forest surveys (i.e. for commercial forest's stocking estimation or for infected trees detection and treatment).

Chapter 5

Surface Reconstruction from Voxelised FW LiDAR Data

5.1 Introduction

To briefly sum up the previous sections, FW LiDAR data (Section 2) are laser scanning data particularly useful in Forestry, but the huge amount of information recorded make handling of the data difficult. The open source software DASOS (Section 4.3) was developed along with this thesis to ease the usage of the data. DASOS voxelises (Section 4.2) the data before interpretation and this approach is fundamentally different from the related and state-of-art software packages. The output of the voxelisation is a 3D discrete density Volume.

This chapter explains the process of reconstructing the surface of the scanned area from the 3D voxelised FW LiDAR. At first, volumetric rendering[1] approaches are briefly explained at Section5.2. Section 5.3 gives a mathematical definition to the voxelised data, while Section 5.4 describes the actual algorithm of extracting a surface. By the end the results are given in Section 5.5.

5.2 Rendering Approaches of Volumetric Data

Even though the concept of visualising 3D discrete density volumes (Volumetric Visualisations) is new in forestry and remote sensing, it has been widely research in medical imaging and visual effects. There are two approaches of visualising volumetric data.

The first approach is direct rendering, which repeatedly generates 2D images according to the view point (the camera). It is like "taking photos" from a camera and

[1] Volumetric rendering refers the process of visualising 3D Volumes.

putting them in a sequence to produce an interactive video. An example of direct rendering approach is ray-tracing. Ray-tracing generates images by "taking photos"; rays are casted from the view point passing through each pixel and intensity values are assigned to the pixels according to the nearest intersections [37]. Ray-tracing can be time expensive depending on the complexity of the scene and for that reason some of the literature focuses on parallellising the ray-casting process. By introducing parallelisation, real time rendering of small volumetric data (256^3) was achieved by Pfister et al in 1999 [38]. Also, after the release of the CUDA hardware (which is a parallel computing platform on recent nvidia graphics cards), Crassin et al achieved real-time rendering of billions of voxels in 2009 [39].

Even though it is possible to implement real-time interactive environments using direct rendering of the big voxel data, volumetric visualisations of FW LiDAR data is a new concept in remote sensing and for simplicity, this thesis experiments surface reconstruction. Surface reconstruction refers to the extraction of a polygonal mesh, which is a set of primitives like triangles, from the volumetric data. Constructing a surface may take several minutes, but real time visualisations of polygonal meshes are supported by free animation packages (like Blender and Meshlab) and it work on commodity hardware. This is achieved with rasterisation (the second rendering approach), which is a method that maps the primitives to pixels, it is widely used in computer games and it is significantly faster than ray-tracing. Furthermore, interactive operations (e.g. measuring the distance between two trees) are trivial calculations on primitives/polygonal meshes and they are easy to implement.

5.3 Algebraic Definition of the Volume

In computer graphics, polygonal meshes are a set of primitives usually triangles, but some objects are defined differently, for example using a function. Those objects are called either implicit or algebraic. Implicit representation of objects enables a mathematical definition of the 3D discrete density volume generated from the FW LiDAR data (Section 4.2).

Algebraic objects were firstly introduced in computer graphics by Blinn in 1982 [40] to enable the definition of complex objects without saving a large amount of primitives. Each object is defined by a function $f(X)$ and the iso-surface value α. The iso-surface value (iso-level) defines the boundaries of the object; for an object $[f(x), a]$ every n-dimensional point X that lies on the surface of the object satisfies the condition $f(X) = \alpha$. To be more accurate, the following rules apply according to Pasko et al [41]:

- f(X) $= \alpha$, when X lies on the surface of the algebraic object

- f(X) > α , when X lies inside the algebraic object and
- f(X) < α , when X lies outside the algebraic object

Regarding the algebraic representation of the 3D voxelised FW LiDAR data, X is a three dimensional point (x, y, z) representing the longitude, latitude and height respectively and $f(X)$ is a function that takes X as input and returns the accumulated intensity value of the voxel that X lies inside. Also, the iso-surface value α is a user defined parameter and it is noise dependant (Please look at figure 5-5 to understand how iso-level affects the output of the surface reconstruction of the voxelised FW LiDAR data).

5.4 Surface Reconstruction with the Marching Cubes Algorithm

Even though numerical implicitisation is beneficial in reducing storage memory, visualising implicit objects is not straight forward, since they contain no discrete values. This problem can either be address either by direct rendering or surface reconstruction (Section 5.2).

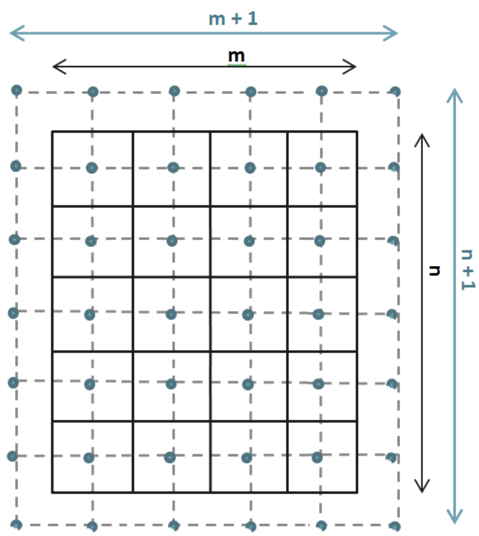
The Marching Cubes algorithm is an algorithm that polygonises implicit objects using a look up table. Let's assume that $f(X)$ defines an implicit object. At first the space is divided into cubes. Each cube is defined by eight corner points and each corner point lies either inside or outside the implicit object. By enumerating all the possible cases and linearly interpolating the intersections along the edges, the surface of the implicit object is constructed [42].

According to Lorensen and Cline [42], the normal[2] of each vertex is calculated by measuring the local gradient change. Even though this work well on smooth object (e.g. a sphere defined by its equation), because of the high gradient changes in the voxelised FW LiDAR data this algorithm results into normals pointing into inconsistent directions. This is a problem because when the normals are not consistent, the surface of the object appears rough. For that reason, in DASOS the normal of each vertex is derived by the average normal of its adjacent triangles.

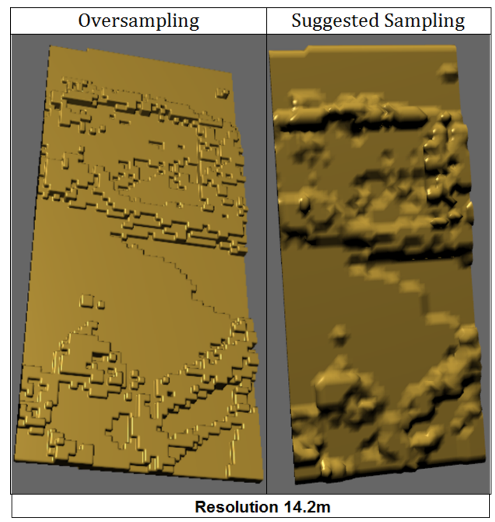
Additionally it worth highlighting that the sampling of the Marching cubes is independent from the sampling of the 3D density volume. But consistency between the two is required to avoid artifacts. Let's assume the discrete volume has (n * m * k) voxels,

[2] A normal is a vector that is perpendicular to the surface of a polygonal mesh. In graphics, the normals are important for calculating light illumination and each vertex is associated with one for smooth rendering of surfaces.

then the suggested Sampling of Marching Cubes is ((n+1) *(m+1) * (k+1)), as shown on Figure 5-1; the black grid represents a 2D density grid and the blue grid represents the suggested sampling of the polygonisation. Please note that every point that lies outside the volume is considered to be outside the implicit object. In addition, Figure 5-1b shows the effects of oversampling; the right image was oversampled and the second one was sampled as explained above.



(a) Suggested Sampling



(b) Effects of Oversampling

Figure 5-1: The suggested sampling during polygonisation using the Marching Cubes Algorithm

5.5 Results

The output of DASOS is a polygonal mesh exported into a .obj file, which is a standard graphics format. The .obj files can be loaded into various animation software tools like Maya and Meshlab (figure 5-2). Figure 5-3 shows polygonal meshes generated using NERC-ARF data from three different areas in UK. The region of interest is also user defined. The user defines whether an entire flightline or selected area is polygonised (figure 5-4).

Furthermore, there are three main user-defined parameters and figure 5-5 shows how the results are affected once modified:

1. The voxel Length controls the resolution of the output; the bigger the voxel length is the lower the resolution and the number of cubes are.

2. The iso-level is the boundary that defines whether a voxel is inside or outside the implicit object. While iso-level increases the number of voxels that are considered inside the implicit object decreases. For that reason, when it is too high most of the voxels are outside the boundary and the object seems to disappear.

3. The Noise level is the threshold of the low level filtering applied during Voxelisation (Section 4.2). If the noise level is too low, then the noise covers significant features of the data and when it is too high important information are discarded and the object seems to disappear again.

By the end, it is possible to 3D print the meshes using the MakerBot but it's tricky because the meshes are not manifold[3] (figure 5-6). Simplification of the mesh would have eased the processing of the .obj file in MakerBot.

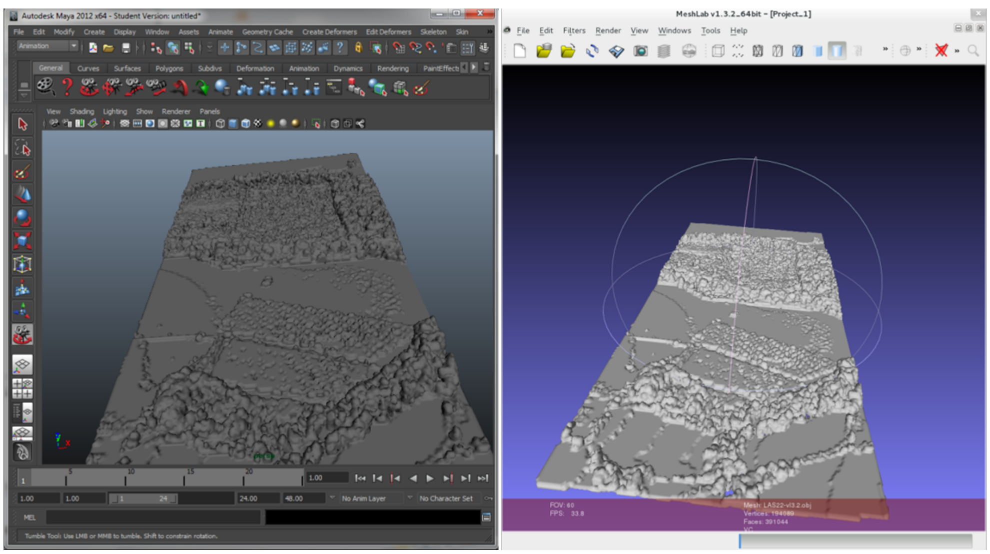


Figure 5-2: Visualising the output of DASOS into animation software packages (Maya and Meshlab)

[3]A non-manifold polygonal object may have triangle below the outside surface of the object

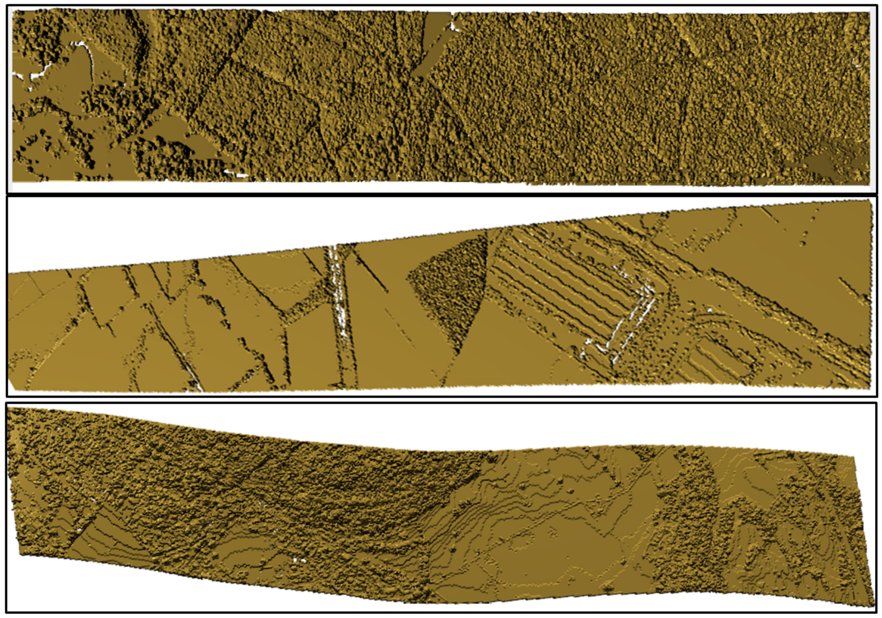


Figure 5-3: Polygonising NERC-ARF FW LiDAR data captured at different areas (New Forest, Milton Keynes and Eaves Wood)

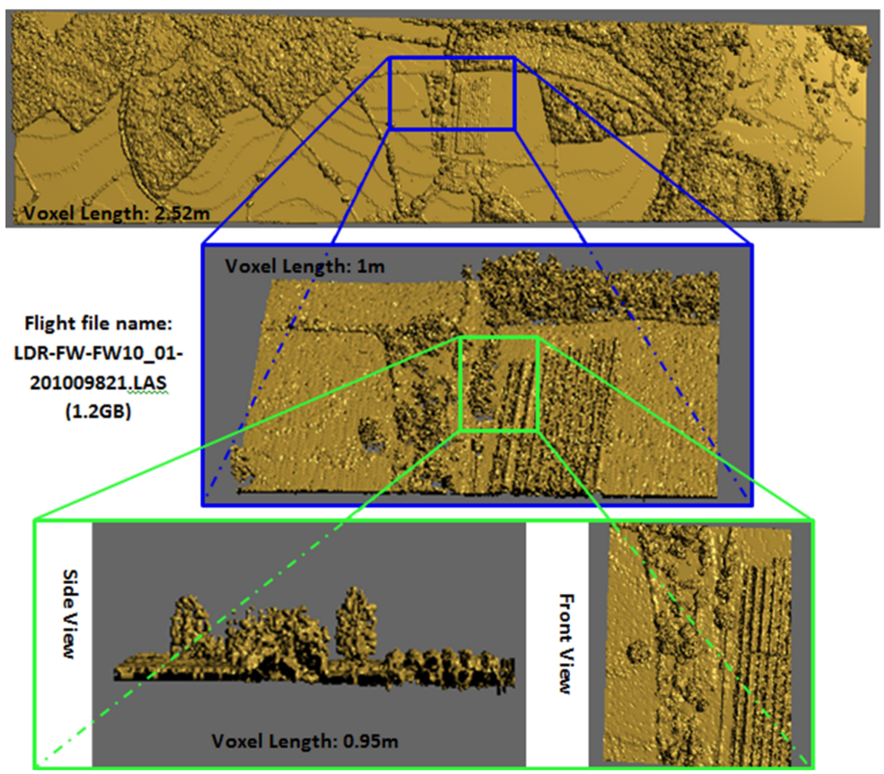


Figure 5-4: Selecting Region of Interest

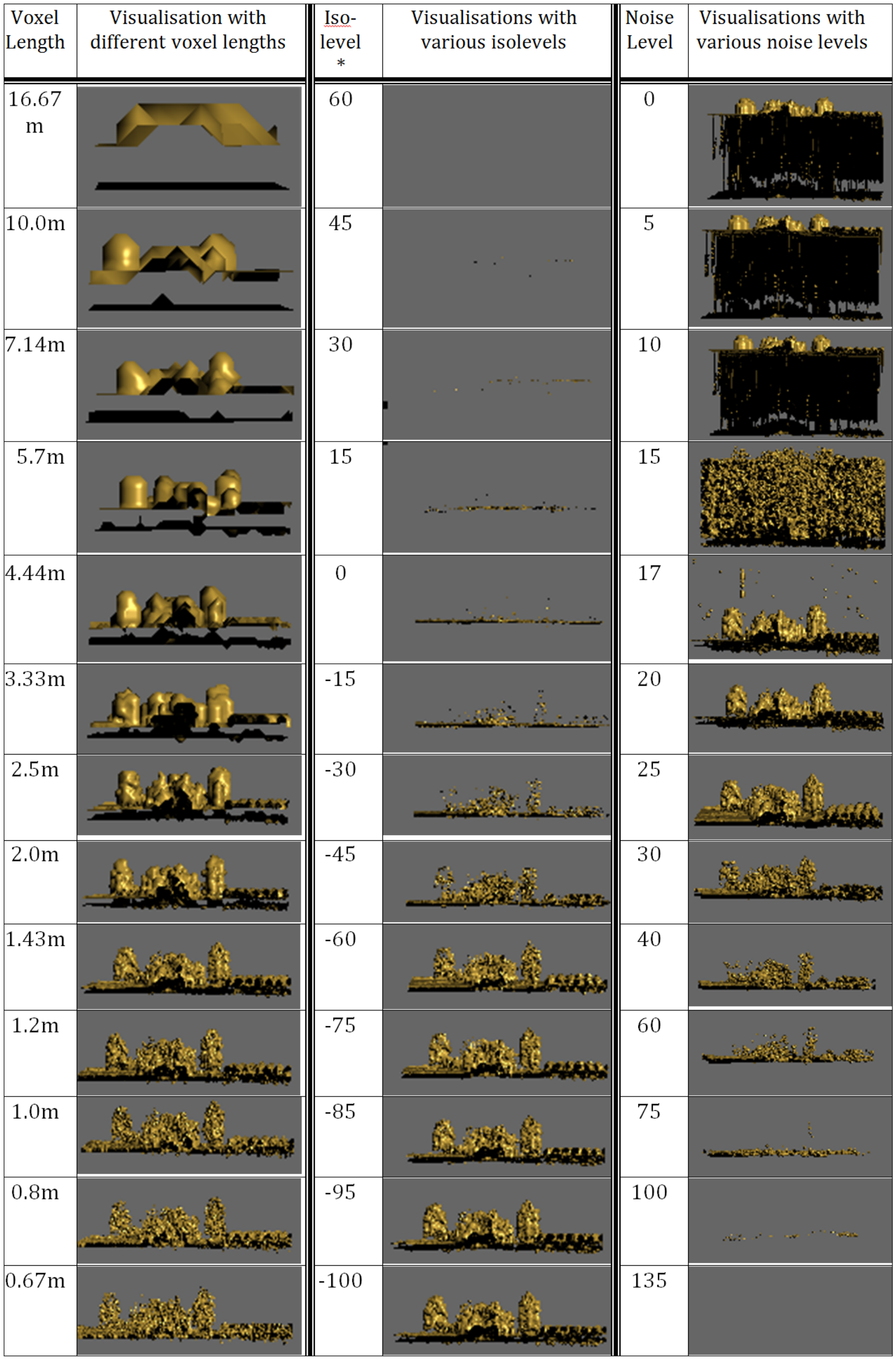


Figure 5-5: How the output polygon mesh is affected by modifying the user-defined parameters (voxel length, isolevel[4] and noise level). Please note that the intensities were scaled to be within the range [-100,100] and that the current released version of DASOS do not scale the intensities.

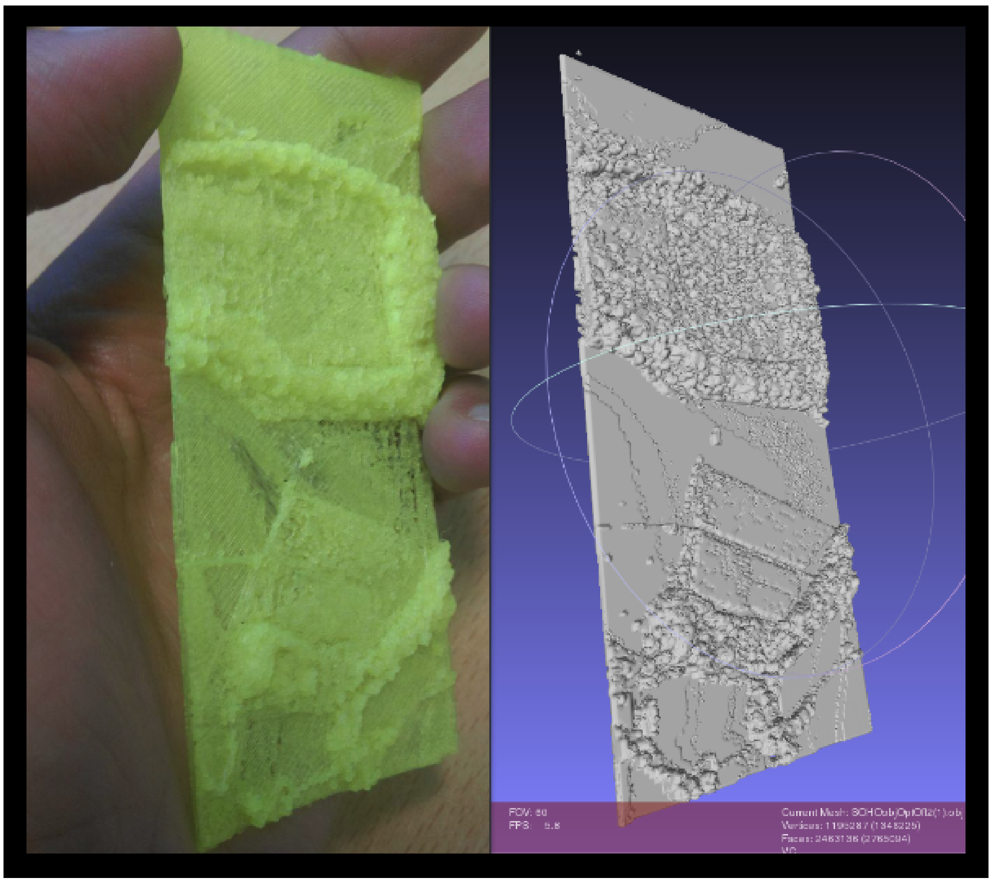


Figure 5-6: 3D printing of New Forest FW LiDAR data

Chapter 6

Optimisations of Polygon Meshes Extraction

the new data structures implemented and introduced for speeding up the surface reconstruction process.

In 1984, Crow proposed an image representation where each pixel value is replaced by the sum of all the pixels that belong to the rectangle defined by the lower left corner of the image and the pixel of our interest (Crow, 1984). Even though more storage space is required to save the image, due to the larger numbers, the sum of every rectangle in the image can be calculated in constant time, once the table is constructed. The integral image can also be constructed in linear time O(n), where n is the number of pixels in the image. One iteration through the entire image is enough to replace the pixel values of the image.

image

6.1 Optimising Surface Reconstruction

The original Marching Cubes algorithm uses a scan line approach and processes all the cubes sequentially. The scan line approach sometimes implies looping through large amounts of empty cubes. To optimise the process, in this research, it is proposed an algorithm that repeatedly divides the volume and utilities Integral Volumes to discard empty volumes during the polygonisation. Using this optimisation algorithm it was achieved an up to 51This optimisation algorithm is explained in the following sections, which are structured as follow: - Background information, including previous related work. - How integral images are extended into 3D. - The proposed algorithm of using Integral Volumes to speed up the process of polygonising an implicit object using the Marching Cubes algorithm - Implementation details that contribute to the efficiency and speed up of the algorithm - Results

6.1.1 Background

Much research has been done so far on optimising and improving the polygonisation of implicit surfaces. But most research is based on closed and manifold object. A polygon mesh is closed when it has no holes and it is manifold when it can be unfolded into a 2D continues surface. In order to guarantee consistent topology, an optimised approach with an enhanced look up table was proposed by Lewiner et al in 2003.

Further surface-tracking was discussed in both papers of (Rodrigues de Araujo and Pires Jorge, 2005) (Hartmann, 1998). Starting from a seed point, the surface is expanded according to the local curvature of the implicit object. Self-intersections and collisions are avoided using heuristic edge length. This method is considered to be faster and more efficient, in comparison with the Marching Cubes algorithm. Because, by tracking the surface, huge empty spaces are not searched and it is also possible to create smaller triangles on places with high gradient changes and bigger triangles on surfaces with low variance. Nevertheless, surface-tracking Algorithms cannot be applied in our case though, because the output of on my program is a non-manifold objects. The 3D Volume generated from FW LiDAR data is not consistent, since small footprint Leica FW systems cannot guarantee that the last return is from the ground. For that reason, it is possible that some trees may be separated from the ground. Surface-tracking algorithms converge once the object is closed. Therefore, there is a possibility that they may converge after polygonising a single tree if the seed point lies on a tree that is separated from the ground.

In 1992, Hansen and Hinker proposed parallelising the polygonisation process of BlobTree trees on Single Instruction, Multiple Data (SIMD) machines. BlobTree trees represent implicit objects as a combination of primitives and operations likes union and blends (Galbraith, MacMurchy, and Wyvill, 2004). While the depth of the tree increases, the time required to get the value that defines whether a point is inside or outside the object increases as well. On SIMD machines greater speed up is achieved at longer instruction due to the less communication cost. Therefore parallelising the process of BlobTree trees with long depth is beneficial, but in our case the value returned for a given point is calculated in constant time. Further, according to the C++ Coding

Standards by Sutter and Alexandrescu, when optimisation is required is better to seek an algorithmic approach first because it is simpler to maintain and the possibility of being bug free is higher (Sutter and Alexandrescu, 2004).

OpenVDB library manages volumetric data with octrees. My program uses 1D arrays allowing constant time access of voxel values and by importing OpenVDB library, its speed was significantly decreased; while the number of voxels is increasing the time required to get the value of a voxel is also increasing. Further according to the documentation, the VolumeToMesh class "meshes any scalar grid that has a continuous isosurface", while the surface of the FW LiDAR volumes is not continuous; there are triangles that represent leafs inside the trees and some of the trees may be disconnected from the ground because the last return do not always reach the earth (OpenVDB 2.3.0).

In this report, it is introduced a new method of optimising the marching cubes algorithm. This method utilises Integral Volumes (an extension of Integral Images) to discard chunks of empty cubes during polygonisation. Its effectiveness stands at the ability of integral volumes to find the sum of any sub-volume into constant time and it is important because it works effectively non-manifold or non-closed objects.

6.2 ? Summary

Then an algebraic representation of that volume is defined by the function $f(X)$ and the iso-surface value α. The function $f(X)$ takes as input a point X and returns the associated intensity of the voxel that X lies inside. If the returned intensity is greater than the value α then X lies inside the algebraic object, if it is is equal to α then it is on the boundary otherwise it lies outside.

Chapter 7

Alignment with Hyperspectral Imagery

7.1 Previous Work

Regarding the integration of FW LiDAR and hyperspectral data in remote forest surveying, there are diverse opinions on whether the integration of multi-sensor data improves remote forest surveying. Clark et al attempted to estimate forest biomass but no better results were observed after the integration [43], while the outcomes of Aderson et al for observing tree species abundances structures were improved after the integration of the data [44].

Buddenbaum et al [45], and Heinzel and Koch [46], used a combination of multi-sensor data for tree classifications. Buddenbaum et al use fusion of data to generate RGB images from a combination of FW LiDAR and hyperspectral features, although the fusion reduces the dimensionality of a classifier [45]. In that study, three different classifiers were implemented and the Support Vector Machines (SVMs) returned the best results. SVMs were also used in [46] to handle the high dimensionality of the metrics (464 metrics). In that research a combination of FW LiDAR, discrete LiDAR, hyperspectral and colour infrared (CIR) images are used. Each of the 125 hyperspectral bands was directly used as a feature in the classifier, contributing to the high dimensionality.

In this chapter, the hyperspectral images are introduced to improve the visual output of the polygonal meshes derived from the FW LiDAR data (Chapter 5) and it is also investigated how the combination of NERC-ARF data from New Forest (Figure 2-1) performs for generating tree coverage maps.

7.2 Spatial Representation of Hyperspectral Pixels for Quick Search

For the New Forest Dataset (Figure 2-1), there are both FW LiDAR and hyperspectral data, but since the data are collected from different instruments they are not aligned. To integrate the data geo-spatially, aligning the data is required. In order to preserve the highest possible quality and reduce blurring that occurs during geo-rectification, data in original sense of geometry (level 1) are used. More Information about the hyperpsectral Imagery are available in Section 2.4.

In Anderson et al [44], an inverse distance weighted algorithm is used to raster the hyperspectral images and the pixel size is constant, 15.8m, while in this study an approach similar to Warren el [24] is used and the resolution is changeable. The main concept of our geo-rectification algorithm is to be able to find the nearest hyperspectral pixel to a point in the fastest possible way. For the reason, a spatial representation of the hyperspectral pixels is created by importing the pixels into a 2D grid, similar to [24]. The dimensions of the grid in meters are constant, but the dimensions of the grid in number of squares (n_x, n_y) is modifiable according to the chosen average number of pixels per square (A_{ps}):

$$n_x = \sqrt{\frac{n_s^2}{A_{ps}}} \qquad n_y = \sqrt{\frac{n_l^2}{A_{ps}}} \tag{7.1}$$

where n_s is the number of samples and n_l is the number of lines of the hyperspectral cube (figure 2-4).

Furthermore, while Warren et al use a tree-like structure, here a structure similar to hash tables is used for speeding up searching. We utilise the unordered_multimap available with the standard library of the $C{+}{+}$ programming language, where for every key there is a bucket with many values stored inside. Each square (x_s, y_s) has a unique key and each pixel is associated with the square it lies inside. In other words, every key correspond to a bucket and a single square from the spatial grid. Every bucket contains all the pixels that lie inside the related square. Also, the key is equal to $(x_s + y_s * n_x)$ where n_x is the number of pixels in the x-axis. Figure 7-1 illustrates how the hash table works for the spatial representation of pixels.

The next step is for a point (x_v, y_v, z_v) to find the pixel whose geolocation is the closest to it. First we project the point into 2D by dropping the z coordinate and then we find the square (x_s, y_s) that the projected point $\mathbf{v}(x_v, y_v)$ lies inside, as follow:

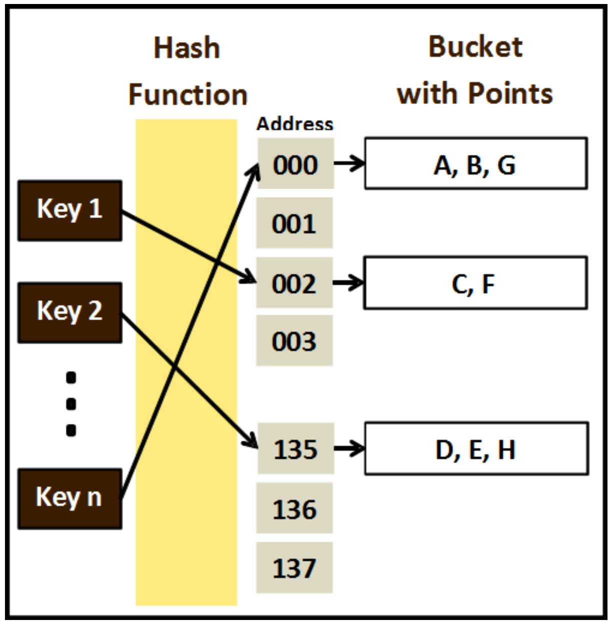


Figure 7-1: The hash table of the spatial representation of the hyperspectral pixels; each bucket contains all the pixels of a square and has a unique key derived from the location of its square. The hash function takes as input the key and returns the address in memory of the corresponding bucket.

$$x_s = \frac{x_v - X_{min}}{X_{max} - X_{min}} * n_x \tag{7.2}$$

$$x_s = \frac{y_v - Y_{min}}{Y_{max} - Y_{min}} * n_y \tag{7.3}$$

where X_{max}, X_{min}, Y_{max}, Y_{min} are the geospatial boundaries of all the hyperspectral image and n_x, n_x are the number of pixels in the x and y axis accordingly.

From the square (x_s, y_s) we can get the set of pixels that lie inside the same square with the point of our interest. Let's assume that the geospatial locations of these pixels are the vectors $\mathbf{g_1}, \mathbf{g_2}, \mathbf{g_3}, ..., \mathbf{g_n}$ respectively. Then, by looping through that set of pixels, we can find the pixel i that is most likely to be the closest pixel to the point $\mathbf{v}(x_v, y_v)$:

$$i = argmin|\mathbf{v} - \mathbf{g_i}|^2 \tag{7.4}$$

By the end, there is the case of the closest point to be within an adjacent square and this occurs when the point is very close to the edges of the square. Even though this was not implemented in DASOS when the paper [27] was published, it can be done by checking the distance between the edges of the square and the point. If this distance is smaller than the distance between pixel i then we could loop through the points of the corresponding adjacent square and check whether there is another pixel closer to point $\mathbf{v}$ than pixel i. Similarly, by checking the distance between the point $\mathbf{v}$ and the corners of the square (x_s, y_s), the case of the closest point to exist inside a diagonal adjacent square is also covered.

7.3 Projecting Hyperspectral Images into Polygon Meshes generated using FW LiDAR data

This section focuses on projecting the (level 1) hyperspectral Images onto the polygonal meshes reconstructed from the FW LiDAR data as explained in Section 5.4. As shown the at Figure 7-2 the result is a coloured polygon mesh. That mesh is saved into two files:

- the .obj file that contains the 3D geometry and
- the .png file that contains the 2D texture image

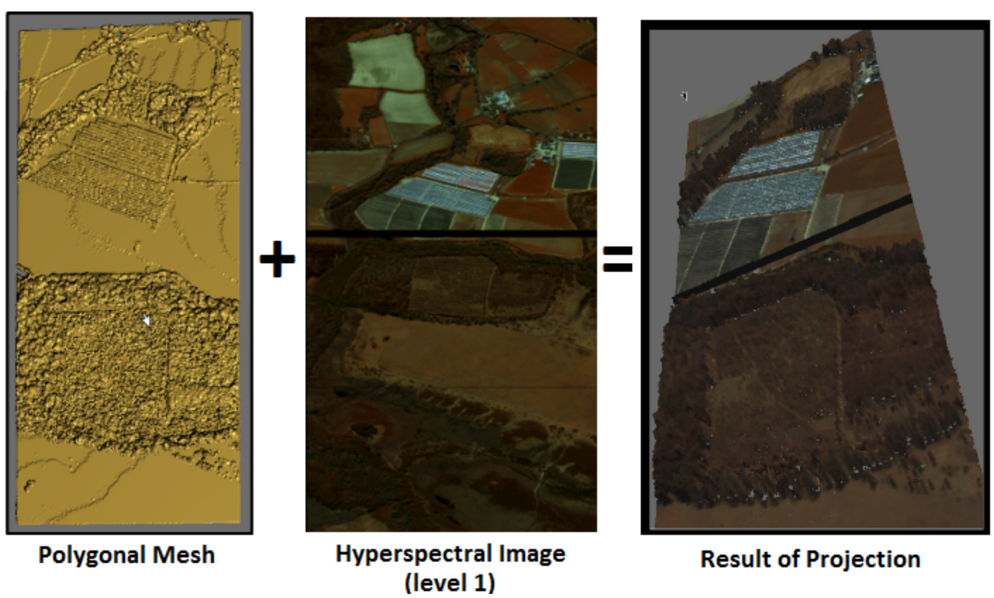


Figure 7-2: Projecting hyperspectral images into the polygonal meshes

The (level 1) hyperspectral images look deformed because the pixel size is not consistent (figure 7-2) and DASOS resolves this problem by adjusting the texture coordinates of the polygonal mesh according to the geolocation of the pixels. The texture coordinates (u, v) of each vertex lies inside the range $[0, 1]$ and if they are multiplied by the height/width of the texture, then the position of the corresponding pixel of the texture is given. In order to calculate the texture coordinates of each vertex (x_v, y_v, z_v), the spatial representation of the hyperspectral pixels (explained at Section 7.2) is used for quickly detecting the pixel (x_p, y_p), whose geolocation is the closest to a vertex. By dividing the pixel position with the number of samples (n_s) and lines (n_l) in the hyperspectral image, the texture coordinates (u, v) of each vertex $v(x_v, y_v)$ are calculated:

$$u = \frac{x_p}{n_s} \qquad v = \frac{y_p}{n_l} \tag{7.5}$$

Regarding the outputs the texture coordinates of the polygonal mesh are added into the .obj file, while the 2D texture is simply an image generated from three user-selected bands for the RGB colours. The width of the image is equal to the number of hyperspectral samples per line while its height is equal to the number of lines.

7.3.1 Results

The results of the projection are coloured polygonal meshes. Each coloured polygonal mesh is exported into two files:

1. the .obj file contains the 3D geometry with all the information about the vertices, edges, faces, normals and texture coordinates, and
2. the .png is the 2D texture (an RGB image) and it is aligned with the texture coordinates of the polygonal mesh.

Figure 7-3 shows how the visual output is affected by projecting hyperspectral data from different sensors and by changing the selected bands.

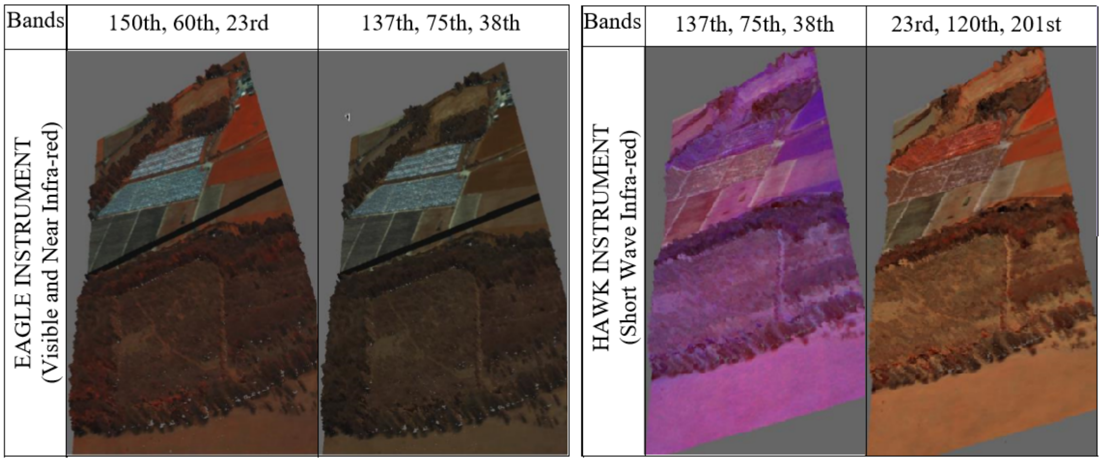


Figure 7-3: Results of Alignment; the left table shows the results of projecting hyperspectral images from the Eagle Instrument onto the polygonal meshes generated using FW LiDAR data and the right hand side table shows results using the Hawk instrument.

7.4 Tree Coverage Maps

As mentioned before, there are diverse opinions on whether integration of remotely sensed data improves forest monitoring [43] [44]. For that reason, a simple pixelwise classifier was implemented to test how the integration of NERC-ARF data, using metrics generated from DASOS, performs for generating tree coverage maps.

The metrics generated from both hyperspectral and FW LiDAR data are 2D aligned images (Table 4.1). In other words, the pixel (x, y) has the same geospatial coordinates in every metric. Further the resolution of the metrics depends on the resolution of the 3D voxelised FW LiDAR data (Section 4.2). If the dimensions of the volume are (x, y, z) then the dimensions of the metrics are (x, y). For the LiDAR metrics, each pixel is coloured according to the information derived from the corresponding column. Regarding the Hyperspectral metrics, (level 1) data are used to preserve the highest possible quality. The same method as Section 7.2 is used for finding the pixel from the hyperspectral data that its geospatial location is the closest to the centre of each column of the 3D voxelised FW LiDAR.

The metrics used for generating tree coverage maps are grouped into two categories (FW LiDAR and hyperspectral metrics):

- FW LiDAR: Height (L0), Thickness (L1), Density (L2) and First Patch (L3)
- Hyperspectral: Mean (H0), NDVI (H1), Standard Deviation (H2) and Spectral Signature (H3)

For more descriptive information and examples of the metrics please look at Table 4.1, where all the functionalities of DASOS are listed.

7.4.1 Testing and Results

In this case, the total accuracy was increased with the integration of FW LiDAR data and hyperspectral images. A Naïve Bayesian classifier using a multi-variance Gaussian model is applied for distinguishing tree covered areas from the ground. The main idea is for each pixel/column to find the class that is more likely to belong to Tree or Ground. Ground truth data were hand painted using 3D models generated with DASOS and they were divided into training and testing data. Further there are three test cases and for each test case the following metrics are used:

- 1st test case uses the L0-L3 metrics that are generated from the FW LiDAR data.
- 2nd test case uses the H0-H3 metrics that are generated from the hyperspectral imagery.
- 3rd test case uses L0-L3 & H1-H4 which is a combination of metrics generated from either FW LiDAR data or hyperspectral imagery.

For each test case, an error matrix is generated to indicate the accuracy of the classification results as verified on the ground truth data (Tables 7.1, 7.2 and 7.3) [47]. Each row shows the number of pixels assigned to each class relative to their actual class. For example, the first row of Table 7.1 shows that 130445 pixels were classified as trees, where 125375 were actual trees and the rest 5070 were ground. From the error matrices the classification accuracy of each test case was calculated and it is presented in Table 7.4.

Figure 7-4 depicts the coverage maps generated for each test case. Three areas are marked for comparison. In Area 1 there is low vegetation, in Area 2 there are short trees and in Area 3 warehouses. Area 1 has been wrongly classified when only the hyperspectral data were used; nevertheless when the height information of the LiDAR data was included into the classifier, area 1 was correctly classified. Similarly, Area 2 was wrongly classified when using the only FW LiDAR metrics because the height of the trees was less than the average training samples. But since features from the hyperspectral data are not height dependant, the classification results of test case 1 (with hyperspectral metrics) was better at Area 2. By the end Area 3 seems to confuse the first two classifiers in different ways, while the combination improved the results.

By the end, to demonstrate the usefulness of DASOS's polygonal meshes, the results of the tree coverage maps were projected into the polygon representations as shown in the following Figure 7-5.

Results		Ground truth data		
		Tree	Ground	Row Total
	Tree	125375	5070	130445
	Ground	45093	228495	273588
	Total	170468	233565	404033

Table 7.1: Error Matrix for 1st test case (Hyperspectral)

Results		Ground truth data		
		Tree	Ground	Row Total
	Tree	154768	39504	194272
	Ground	15700	194061	209761
	Total	170468	233565	404033

Table 7.2: Error Matrix for 2nd test case (FW LiDAR)

Results		Ground truth data		
		Tree	Ground	Row Total
	Tree	152597	10548	163145
	Ground	17871	223017	240888
	Total	170468	233565	404033

Table 7.3: Error Matrix for 3rd test case (Both)

	FW LiDAR	**Hyperspectral Imagery**	**Both**
Tree	73.55%	90.79%	89.52%
Ground	97.83%	83.09%	95.48%
Total	87.58%	86.34%	92.97%

Table 7.4: Classification accuracy of each test case

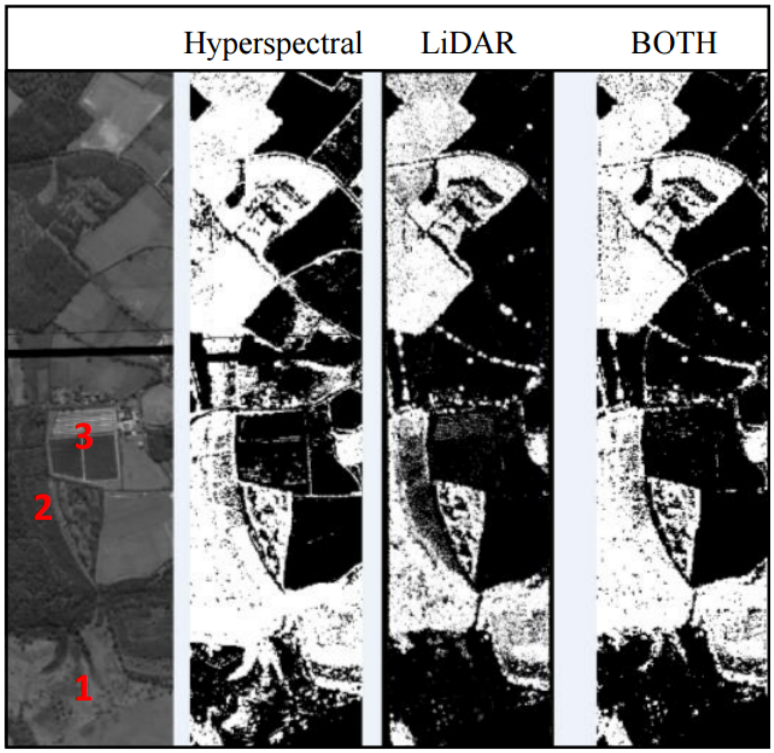


Figure 7-4: Visual Comparison of the Results of the Coverage Maps

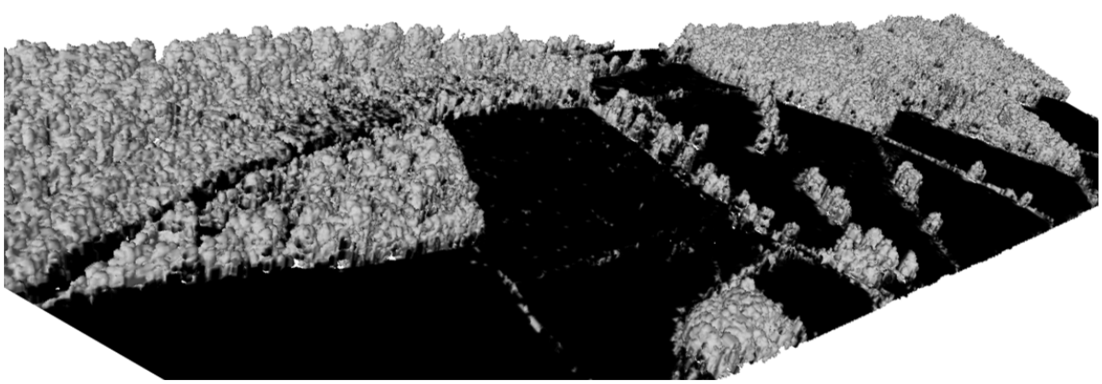


Figure 7-5: 3D Coverage Model, generated by Projecting the Results of the Tree Coverage Classification into a Polygonal Mesh.

7.5 Summary and Conclusions

To sum up, this chapter describes an efficient way of aligning the FW LiDAR data and hyperspectral images using a spatial representation of hyperspectral pixels. The voxelisation of the FW LiDAR data also eases the generation of aligned metrics from both datasets. Furthermore, the resolution of the metrics is changeable and depends on the user-defined resolution of the voxelised FW LiDAR data. Additionally, since the closest pixel is always selected, regardless the distance from the point of interest, the problem of having data of different resolution is automatically resolved.

Regarding the results, coloured polygonal meshes are generated using the alignment and it was also showed that the integration of this specific data has great potentials in remote forest surveying. This was demonstrated using a simple classifier for generating tree coverage maps. The results were positive; the classification accuracy was improved by 5.39% when both datasets were used.

Chapter 8

Classifications using 3D Prior Models

This talk presents the new features of DASOS, which is an open source software for managing full-waveform LiDAR data and those features are used for detecting dead standing Eucalypt.

The value of dead standing Eucalypt trees from a biodiversity management perspective is large. In Australia, many arboreal mammals and birds that are close to extinct inhabit hollows [5]. Nevertheless, studies predict shortage of hollows in the near future due to tree harvesting and the decades required for a tree to be mature enough to develop a hollow [3] [4]. Dead standing eucalypt trees are more likely to be aged and have hollows, therefore automated detection of them plays a significant role in protecting animals that rely on hollows.

DASOS (= $\delta \acute{\alpha} \sigma o \varsigma$) means forest in Greek and it is an open source software aiming to ease the way of handling FW LiDAR data in forestry [27]. Traditional ways of interpreting FW LiDAR data, suggests extraction of a denser points cloud using Gaussian decomposition [29] [30]. Nevertheless DASOS was influenced by Persson et al, 2005, who used voxelisation to visualise the waveforms [26]. But, DASOS do not only uses voxelisation for visualisations but also for extracting metrics useful in classification. It further normalises the intensities so that equal pulse length exists inside each voxel, making intensities more meaningful. It is further seems that the literature is moving towards voxelisation with the good results obtained at recent publication on tree species classification [33].

The new features of DASOS: New features of DASOS which enables observation at tree level: i.e. distribution of intensities at specific area

The data, provided by RPS Australia East Pty Ltd, were collected in March 2015 from the Riegl (LMS-Q780 or LMS-Q680i?) sensor at an Australian native Forest with eucalyptus. The fieldplots has been provided by (Interprine Group Ltd or Forest Corporation?).

examined with Random Forest

The new features of DASOS are presented and used for generating 3D signatures characterising dead standing trees and a comparison between the discrete and FW LiDAR data is performed to demonstrate the increased survey accuracy obtained with the FW LiDAR.

This paper presents a new feature of DASOS, which is an open source software for managing full-waveform (FW) LiDAR data and that feature is used for detecting dead standing Eucalypt trees in native Australian forests.

The value of dead standing Eucalypt trees from a biodiversity management perspective is large. In Australia, many arboreal mammals and birds, which are close to extinct, inhabit hollows [5]. Nevertheless, studies predict shortage of hollows in the near future due to tree harvesting and the decades required for a tree to develop a hollow [3] [4]. Dead standing eucalypt trees are more likely to be aged and have hollows, therefore automated detection of them plays a significant role in protecting animals that rely on hollows.

The LiDAR data used for this project are provided by RPS Australia East Pty Ltd and they were collected in March 2015 using the Riegl (LMS-Q780 or LMS-Q680i?) sensor. The Riegl LMS-Q??? is a native full-waveform sensor and the LiDAR point clouds were generated from the waveform instrument data during post processing. In addition, the field plots used for the classifications are provided by (Interprine Group Ltd or Forest Corporation?) and contain around 1000 Eucalypt trees while 10% of them are dead.

The new feature of DASOS calculates forestry metrics within a radius relevant to canopy height and exports all metrics into a single vector for fast interpretation in advanced statistical tools. Traditional ways of interpreting FW LiDAR data, suggests extraction of a denser points cloud [29] [30], but as mentioned before with the Riegl system this is done at post processing. Nevertheless DASOS was influenced by Persson et al, 2005, who used voxelisation to visualise the waveforms [26], but DASOS also uses it for generating metrics. It further normalises the intensities so that equal pulse length exists inside each voxel, making intensities more meaningful. Further, recent publication on tree species classification showed that voxelisation could confer good results while interpreting FW LiDAR data [33].

Previous work on dead standing trees detection, suggests single tree segmentation before dead trees identification [48] [49] but in case of Eucalypt trees single tree detection is a challenge on its own due to their irregular structure and multiple trunk splits.

In this project, the new feature of DASOS is used for generating 3D signatures characterising dead standing Eucalypt trees and a comparison between the LiDAR point cloud and FW LiDAR data is performed using Random Forest to demonstrate the increased survey accuracy obtained with voxelisation.

Chapter 9

Comparison with Discrete Data

Furthermore, DASOS allows the user to choose whether the waveform samples or the discrete returns are inserted into the 3D density volume. Each sample or each return has a hit point and an intensity value. So, in both case the space is divided into 3D voxels and the intensity of each return or sample is inserted into the voxel it lies inside.

In general the results of discrete returns contain less information compared to the results from the FW LiDAR, even though the FW LiDAR contain information from about half of the emitted pulses (Section 3). As shown on the 1st example of table 3 the polygon mesh generated from the FW LiDAR contains more details comparing to the one created from the discrete LiDAR. The forest on the top is more detailed, the warehouses in the middle have a clearer shape and the fence on the right lower corner is continuous while in the discrete data it is disconnected and merged with the aliasing.

FW LiDAR polygons, compared to the discrete LiDAR ones, contain more geometry below the outlined surface of the trees. On the one hand this is positive because they include much information about the tree branches but on the other hand the complexity of the objects generated is high. A potential use of the polygon representations is in movie productions: instead of creating a 3D virtual city or forest from scratch, the area of interest can be scanned and then polygonised using our system.

But for efficiency purposes in both animation and rendering, polygonal objects should be closed and their faces should be connected. Hence, in movie productions, polygons generated from the FW LiDAR will require more post-processing in comparison with object generated from the discrete LiDAR.

Example 2 in table 3 shows the differences in the geometry complexity of the discrete and FW polygons using the x-ray shader of Meshlab. The brighter the surface appears the more geometry exists below the top surface. The brightness difference between area 1 and area 2 appears less in the discrete polygon.

Nevertheless, the trees in area 2 are much taller than in area 1, therefore more geometry should have existed in area 2 and sequentially be brighter. But the two areas are only well-distinguished in the FW LiDAR. On average the FW polygon is brighter than the discrete polygon, which implies higher geometry complexity in the FW polygon.

The comparison example 3 is rendered using the Radiance Scaling shader of Meshlab (Vergne et al, 2010). The shader highlights the details of the mesh, making the comparison easier. Not only the FW polygons are more detailed but also holes appear on the discrete polygons. The resolution of the voxels of those examples is 1.7m3 is, the bigger the holes are, while the full-waveform can be polygonised at a resolution of 1m3 without any significant holes. Figure 4 shows an example of rendering the same flightline of examples 3 at the resolution of 1m3 LiDAR data.

The last two examples (4 and 5) compare the side views of small regions. On the one hand the top of the trees are better-shaped in the discrete data. This may occur either because the discrete data contain information from double pulses than the FW data (Section 3) or because the noise threshold of the waveforms is not accurate and the top of the trees appear noisier on the FW LiDAR data. On the other hand more details appear close to the ground on the FW LiDAR data.

*** left during copying :s (and the higher the resolution, using FW)

Chapter 10

Overall Results

Chapter 11

Conclusions

11.1 Contributions

Bibliography

[1] T. Elmqvist, C. Folke, M. Nyström, G. Peterson, J. Bengtsson, B. Walker, and J. Norberg, “Response diversity, ecosystem change, and resilience,” *Frontiers in Ecology and the Environment*, vol. 1, no. 9, pp. 488–494, 2003.

[2] D. U. Hooper, F. S. Chapin Iii, J. J. Ewel, A. Hector, P. Inchausti, S. Lavorel, and B. Schmid, “Effects of biodiversity on ecosystem functioning: a consensus of current knowledge,” *Ecological monographs*, vol. 75, no. 1, pp. 3–35, 2005.

[3] D. B. Lindenmayer and J. T. Wood, “Long-term patterns in the decay, collapse, and abundance of trees with hollows in the mountain ash (eucalyptus regnans) forests of victoria, southeastern australia,” *Canadian Journal of Forest Research*, vol. 40, no. 1, pp. 48–54, 2010.

[4] R. L. Goldingay, “Characteristics of tree hollows used by australian birds and bats,” *Wildlife Research*, vol. 36, no. 5, pp. 394–409, 2009.

[5] P. Gibbons and D. Lindenmayer, *Tree Hollows and Wildlife Conservation in Australia.* CSIRO Publishing, 2002.

[6] “Animal pests: Poss.”

[7] D. H. DeHayes, P. G. Schaberg, G. J. Hawley, and G. R. Strimbeck, “Acid rain impacts on calcium nutrition and forest health alteration of membrane-associated calcium leads to membrane destabilization and foliar injury in red spruce,” *BioScience*, vol. 49, no. 10, pp. 789–800, 1999.

[8] J. Holmgren, “Prediction of tree height, basal area and stem volume in forest stands using airborne laser scanningce,” *Scandinavian Journal of Forest Researcht*, vol. 19, no. 6, pp. 543–553, 2004.

[9] S. G. Aracil and R. B. A. Herries, D.L, “Evaluation of an additional lidar metric in forest inventory,” *Proceedings of Silvilaser*, 2015.

[10] M. J. Harper, M. A. McCarthy, R. Van Der Ree, and J. C. Fox, "Overcoming bias in ground-based surveys of hollow-bearing trees using double-sampling," *Forest Ecology and Management*, vol. 190, no. 2, pp. 291–300, 2004.

[11] L. Rayner, M. Ellis, and J. E. Taylor, "Double sampling to assess the accuracy of ground-based surveys of tree hollows in eucalypt woodlands," *Forest Ecology and Management*, vol. 36, no. 3, pp. 252–260, 2011.

[12] R. B. Smith, *Introduction to Hyperspectral Imaging.* MicroImages, 2014.

[13] W. Wanger, A. Ullrich, T. Melzer, C. Briese, and K. Kraus, "From single-pulse to ful-waveform airborne laser scanners," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, pp. 100–112, 2004.

[14] A. Wehr and U. Lohr, "Airborne laser scanning - an introduction and overview," *ISPRS Journal of Photogrammerty and Remote Sensing*, vol. 54, pp. 68–82, 1999.

[15] C. Mallet and F. Bretar, "Full-waveform topographic lidar: State-of-the-art," *ISPRS Journal of Photogrametry and Remote Sensing*, vol. 64, pp. 1–16, 2009.

[16] K. Anderson, S. Hancock, M. Disney, and K. Gaston, "Is waveform worth it? a comparison of lidar approaches for vegetation and landscape characterization," *Remote Sensing in Ecology and Conservation*, 2015.

[17] A. Chauve, C. Mallet, F. Bretar, S. Durrieu, M. Deseilligny, and W. Puech, "Processing full-waveform lidar data: Modelling raw signals," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2007.

[18] *LAS Specification version 1.3-R1.* Bethesda, Maryland: American Society for Photogrammetry and Remote Sensing, 2010.

[19] M. Warren, *Full Waveform Upgrade.* NERC ARSF wiki, 2012.

[20] M. J. Sumnall, R. A. Hill, and S. A. Hinsley, "Comparison of small-footprint discrete return and full waveform airborne lidar data for estimating multiple forest variables," *Remote Sensing of Environment*, vol. 173, pp. 214–223, 2016.

[21] K. H. R. A. . Z. A. Lehner, H., "Consideration of laser pulse fluctuations and automatic gain control in radiometric calibration of airborne laser scanning data.," *Proceedings of 6th ISPRS Student Consortium and WG VI/5 Summer School*, 2011.

[22] I. Korpela, H. O. Ørka, H. V. Hyyppä, J., and T. Tokola, "Range and agc normalization in airborne discrete-return lidar intensity data for forest canopies," vol. 65, no. 4, pp. 369–379, 2010.

[23] M. Isenburg, *LAStools - efficient tools for LiDAR processing.* rapidlasso.

[24] M. Warren, B. Taylor, M. Grant, and J. D. Shutler, "Data processing of remorely sensed airborne hyperspectral data using the airborne processing library (apl)," *ScienceDirect, Computers and Geosciences*, vol. 64, 2014.

[25] M. Isenburg, "Pulsewaves: An open, vendor-neutral, stand-alone, las-compatible full waveform lidar standard.," 2012.

[26] A. Persson, U. Soderman, J. Topel, and S. Ahlberg, *Visualisation and Analysis of full-waveform airborne laser scanner data.* V/3 Workshop, Laser scanning 2005, 2005.

[27] M. Miltiadou, M. A. Warren, M. Grant, and M. Brown, "Alignment of hyperspectral imagery and full-waveform lidar data for visualisation and classification purposes," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 7, p. 1257, 2015.

[28] W. Wanger, A. Ullrich, V. Ducic, T. Maizer, and N. Studnicka, "Gaussian decompositions and calibration of a novel small-footprint full-waveform digitising airborne laser scanner," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, pp. 100–112, 2006.

[29] A. Neuenschwander, L. Magruder, and M. Tyler, "Landcover classification of small-footprint full-waveform lidar data," *Jounal of Applied Remote Sensing*, vol. 3, no. 1, pp. 033544–033544.

[30] J. Reitberger, P. Krzystek, and U. Stilla, "Analysis of full waveform LiDAR data for tree species classification," *International Journal of Remote Sensing*, vol. 29, no. 5, pp. 1407–1431, 2008.

[31] A. Chauve, F. Bretar, S. Durrieu, M. Pierrot-Deseilligny, and W. Puech, "Fullanalyze: A research tool for handling, processing and analysing full-waveform lidar data," *IEEE International Geoscience and Remote Sensing Symposium*, 2009.

[32] P. Bunting, J. Armston, D. Clewley, and R. M. Lucas, "Sorted pulse data (spd) library—part ii: A processing framework for lidar data from pulsed laser systems in terrestrial environments," *Computers & Geosciences*, vol. 56, pp. 207–215, 2013.

[33] L. Cao, N. Coops, L. Innes, J. Dai, and H. Ruan, "Tree species classification in subtropical forests using small-footprint full-waveform lidar data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 49, pp. 39–51, 2016.

[34] M. Miltiadou, M. Grant, M. Brown, M. Warren, and E. Carolan, "Reconstruction of a 3d polygon representation from full-wavefrom lidar data," *RSPSoc Annual Conference 2014, New Sensors for a Changing World*, 2014.

[35] R. Crippen, "Calculating the vegetation index faster," *Remote Sensing of Environment*, vol. 34, no. 1, pp. 71–73, 1990.

[36] R. N. Clark, G. A. Swayze, R. Wise, K. E. Livo, T. Hoefen, R. F. Kokaly, and S. J. Sutley, "Usgs digital spectral library splib06a," *US Geological Survey, Digital Data Series*, vol. 231, 2007.

[37] P. Hanrahan, "Ray tracing algebraic surfaces," *ACM SIGGRAPH Computer Graphics*, vol. 17, no. 3., 1983.

[38] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, "The volumepro real-time ray-casting system," *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 251–260, 1999.

[39] C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann, "Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering," *Proceedings of the 2009 symosium on Interactive 3D graphics and games*, pp. 15–22, 2009.

[40] J. F. Blinn, *A Generalization of Algebraic Surface Drawing*, vol. 1. ACM Transactions on Graphics (TOG).

[41] A. Pasko and V. Savchenko, *Blending operations for the functionally based constructive geometry.* 1994.

[42] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM Siggraph Computer Graphics*, vol. 21, pp. 163–169, 1987.

[43] M. L. Clark, D. A. Roberts, J. J. Ewel, and D. B. Clark, "Estimation of tropical rain forest aboveground biomas with small-foorprint lidar and hyperspectral sensors," *ScienceDirect, Remote Sensing of Enviroment*, vol. 115.

[44] J. E. Anderson, L. C. Plourde, M. E. Martin, B. H. Braswell, M. L. Smith, R. O. Dubayah, M. A. H. Dubayah, and J. B. Blair, "Integrating waveform lidar with hyperspectral imagery for inventory of a northern temperate forest," *Remote Sensing of Environment*, vol. 112, no. 4, pp. 1856–1870, 2008.

[45] H. Buddenbaum, S. Seeling, and J. Hill, "Fusion of full-waveform lidar and imaging spectroscopy remote sensing data for the characterization of forest stands," *International Journal of Remote Sensing*, vol. 32, no. 13, pp. 4511–4524, 2013.

[46] J. Heinzel and B. Koch, "Investigating multiple data sources for tree species classification in temperate forest and use for single tree delineation," *International Journal of Applied Earth Observation and Geoinformation*, vol. 18, pp. 101–110, 2012.

[47] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sensing of Enviroment*, vol. 37, no. 1.

[48] W. Yao, P. Krzystek, and M. Heurich, "Identifying standing dead trees in forest areas based on 3d single tree detection from full-waveform lidar data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-7, pp. 359–364, 2012.

[49] P. Polewski, W. Yao, M. Heurich, P. Krzystek, and U. Stilla, "Active learning approach to detecting standing dead trees from als point clouds combined with aerial infrared imagery," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 10–18, 2015.

Chapter 12

Appendices

12.1 Birds and Mammals Catalogue

12.1.0.1 Introduction

12.1.0.2 Australian arboreal Mammals

12.1.0.3 Australian Birds

The Forestry Corporation, Australia, provided a list of bird species that rely on hollows. But species are not limited to that list and more species rely uses hollows for shelters.

The provided list of the birds is divided into three groups:

1. Categorised as threatened species according to the Environement Protection and Biodiversity Conservation Act, 1999

Corella Eastern Rosella Superb Parrot Barking Owl Masked Owl

2. All the above species are included to the Action Plan for Australian Birds, 2000, as well as the following once:

Powerful Owl Sooty Owl

3. The rest:

Kookaburra Sulphur Crested Cockatoo Crimson Rosella Rainbow Lorikeet Musk Lorikeet Little Lorikeet Red-winged Parrot Cockatiel Australian Ringneck (Parrot) Red-rumped Parrot Powerful Owl Sooty Owl Barn Owl White-throated Treecreeper

12.1.0.4 Web-links of Photos

Mammals · Brush-tailed Possum - protected wildlife (Hollow: http://www.cavershamwildlife.com.au/comm
brushtail-possum/) (`http://www.rymich.com/girraween/photos/animals/mammals/possums/trichosu`

Birds · Kookaburra (`http://tenrandomfacts.com/blue-winged-kookaburra/`) ·
Sulphur Crested Cockatoo (`http://aussiegal7.deviantart.com/art/Sulphur-Crested-Cockatoo-08-`

· Corella (http://www.theparrotplace.co.nz/all-about-parrots/long-billed-corella/) · Crimson Rosella (http://25.media.tumblr.com/tumblr_m3mo89c40r1r4t9h1o1_1280.jpg) · Eastern Rosella (http://2.bp.blogspot.com/-pYxw51WjSOY/UB-LEFgd2KI/AAAAAAAAAWg/9z60PUWE6TE/s1600/_GJS6601-as-Smart-Object-1.jpg) · Galah (https://www.pinterest.com/pin/537546905498955709/) · Rainbow Lorikeet (https://www.reddit.com/r/pics/comments/328fvc/a_rainbow_lorikeet_found_in_coastal_regions/) · Musk Lorikeet ((http://www.rymich.com/girraween/photos/animals/birds/medium/glossopsitta_concinna/glossopsitta_concinna_001.jpg) · Little Lorikeet (http://www.pbase.com/sjmurray/psittacidae) · Red-winged Parrot (https://www.pinterest.com/pin/395894623469889727/) · Superb Parrot (http://www.davidkphotography.com/?showimage=637) · Cockatiel (http://up.parsipet.ir/uploads/Cockatiels-for-sale.jpg) · Australian Ringneck (Parrot) (http://ontheroadmagazine.com.au/wp-content/uploads/2015/09/Twenty-eight-parrot-2jpg) · Red-rumped Parrot (http://parrotfacts.net/wp-content/uploads/Red-Rumped-Parrot-on-a-jpg) · Powerful Owl (http://farm1.staticflickr.com/219/495796536_f78dac04c1.jpg) · Sooty Owl (hollow: http://www.mariewinn.com/marieblog/uploaded_images/screech2-738532.jpg) (http://www.owlpages.com/owls/species/images/greater_sooty_owl_richard_jackson-1.jpg) · Barking Owl (http://www.pcpimages.com/Nature-and-Wildlif Birds/i-7JKSTp5/1/L/owl%20%281%20of%201%29-L.jpg) · Masked Owl (http://www.survival.org.au/images/birds/masked_owl_2_600.jpg) · Barn Owl (Hollow: http://www.barnowltrust.org.uk/wp-content/uploads/Barn_Owl_hollow_tree-wallpaper.jpg) (https://upload.wikimedia.org/wikipedia/commons/c/c6/Tyto_alba_-British_Wildlife_Centre,_Surrey,_England-8a_%281%29.jpg)

· White-throated Treecreeper (http://www.birdlifemelbourne.org.au/bird-lists/47-Treecreepers/White-throated-Treecreeper/White-throated%20Treecreeper%202%20JB.jpg) (hollow: https://geoffpark.files.wordpress.com/2011/09/female-white-throated-tre jpg)

Hollow Owl : http://www.mariewinn.com/marieblog/uploaded_images/screech2-738532.jpg