Characterizing Vegetation Structure on Anthropogenic Disturbances Features in Alberta’s Boreal Forest with Unmanned Aerial Vehicles

by

Shijuan Chen

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN GEOGRAPHY

CALGARY, ALBERTA

JUNE, 2017

© Shijuan Chen 2017
Abstract

Characterizing vegetation structure is an important component for understanding ecological recovery on non-permanent human footprint features in forests. However, current approaches to measuring vegetation structure rely on field protocols that are costly and difficult to scale. Compared to traditional field methods, UAV (unmanned aerial vehicle) photogrammetry has shown great promise in characterizing vegetation structure in a more cost-efficient way. In this research, I used a point-intercept sampling strategy to conduct a comparison of UAV-based estimates and field measurements at two scales: (i) point level and (ii) site (plot) level. I found that at the aggregated site level, UAV photogrammetry alone could replace traditional field-based vegetation surveys of mean vegetation height across the range of conditions assessed in this study, though significant differences remain between remote- and field-based vegetation surveys at point level. Cost analysis indicates that using UAV point clouds alone provides substantial cost-saving over traditional field vegetation surveys.
Acknowledgements

First, I would like to express my sincerest appreciation to my supervisors, Dr. Greg McDermid and Dr. Guillermo Castilla. Without your help, this thesis would have not been possible. I have been very lucky to have supervisors who offered me so much support, guidance and encouragements throughout my Master’s studies. Also, I would like to express heartfelt gratitude to Dr. Julia Linke for your help and advice on my research. Many thanks to all of you for your generous dedication of time and knowledge.

Also, I am very grateful to those who participated in field data collection of this research. Without your assistance of data collection, this research would have not been conducted successfully. Especially, I offer many thanks to Jennifer Hird for preparing a thorough, comprehensive and clear field protocol for the fieldwork, and Corey Feduck for leading the field crew, organizing fieldwork and collecting UAV data. Others who assisted on field data collection include Man Fai Wu, Jaren Koning, Vince Zafra, Natalie Sanchez, Michael Bodnar, Erica Graham, Tobias Tan, and Angelo Filicetti. Thank you.

My research is a subproject of BERA (Boreal Ecosystem and Assessment) project, which is funded by a Natural Sciences and Engineering Research Council of Canada Collaborative Research and Development Grant (CRDPJ 469943-14) in conjunction with Alberta-Pacific Forest Industries, Cenovus Energy, and ConocoPhillips Canada. Our collaborating agencies, Alberta Environment and Parks, Alberta Biodiversity Monitoring Institute (ABMI) and Canadian Forest Service also offered much support, such as providing GIS data. Many thanks to all of our partners.

I am also extremely grateful to all members of our lab: Julie Lovitt, Shannon Blackadder, Dr. Jordan Eamer, Man Fai Wu, Jennifer Hird, Dr. David Laskin, Dr.
Mustafizur Rahman, Taylor Mckeeman, Lucy Poley and Annie He. Your friendship and kindness create a nice and pleasant research environment and make me feel at home in Canada. I would like to extend my thankfulness to the entire Department of Geography at the University of Calgary.

Last but not least, I would like to express my deepest gratitude to my mother, Mrs. Weilin Zhao, and my father, Mr. Yunzhang Chen, for their endless love, support and encouragement.
Dedication

I dedicate this thesis to my mother: Mrs. Weilin Zhao.
## Table of Contents

Abstract ......................................................................................................................... i  
Acknowledgements .......................................................................................................... ii  
Dedication ......................................................................................................................... iv  
Table of Contents ............................................................................................................ v  
List of Tables ................................................................................................................... vii  
List of Figures and Illustrations .................................................................................... viii  
List of Symbols, Abbreviations and Nomenclature ........................................................ x

CHAPTER ONE: INTRODUCTION ..................................................................................1  
1.1 Background and motivation ....................................................................................... 1  
1.2 Organization of thesis .............................................................................................. 4

CHAPTER TWO: BACKGROUND: REMOTE SENSING POINT CLOUDS FOR  
CHARACTERIZING VEGETATION, A LITERATURE REVIEW ............................... 5  
2.1 Photogrammetric Point Clouds ....... ............................... ............................... 6  
2.1.1 Data Acquisition ............................................................................................... 7  
2.1.2 Generating and Processing Photogrammetric Point Clouds ............................. 11  
2.1.3 Characterizing Vegetation by Photogrammetric Point Clouds ......................... 12  
2.2 LiDAR Point Clouds for Characterizing Vegetation .............................................. 20  
2.3 Comparison of Photogrammetric and LiDAR Point Clouds ................................... 27

CHAPTER THREE: METHODS ......................................................................................30  
3.1 Study area ............................................................................................................... 30  
3.2 Field measurements ............................................................................................. 33  
3.3 UAV imagery acquisition ..................................................................................... 37  
3.4 Generating and geo-referencing point-clouds ...................................................... 38  
3.5 LiDAR data ............................................................................................................ 42  
3.6 Estimating vegetation height ............................................................................. 42  
3.7 Estimating fractional vegetation cover ............................................................. 44  
3.8 Comparison and accuracy assessment .......................................................... 45  
3.9 Cost analysis ....................................................................................................... 46

CHAPTER FOUR: RESULTS ..........................................................................................49  
4.1 Estimated vegetation height at point level ..................................................... 49  
4.2 Estimated mean vegetation height at site level ............................................... 52  
4.3 Profile comparison plot .................................................................................... 55  
4.4 Estimated Fractional Vegetation Cover at site level ....................................... 58  
4.5 Optimal search radius for the lowest and tallest UAV points around the sample points ................................................................. 61  
4.6 Cost analysis ....................................................................................................... 63

CHAPTER FIVE: DISCUSSION ......................................................................................64

CHAPTER SIX: CONCLUSION ......................................................................................67
**List of Tables**

Table 1.1 Operating range, flight time, altitude and payload size of UAVs with different sizes (Compiled from (Anderson & Gaston, 2013)) ........................................ 8

Table 1.2 the use of UAS for vegetation and forest inventory ........................................ 9

Table 1.3 Examples of studies of area-based approach with photogrammetric point clouds .................................................................................................................. 15

Table 1.4 Examples of studies using the area-based approach (LiDAR) ......................... 22

Table 1.5 Examples of studies using the individual tree detection approach (LiDAR) ... 25

Table 3.1 Vegetation height statistics for the 1743 sample points measured in this study. Measurements were obtained by point-intercept sampling on 30 seismic lines .......................................................................................................................... 35

Table 3.2 Field fractional vegetation cover by height strata (statistics) (%)1 ............... 45

Table 4.1 RMSE, normalized RMSE and bias of estimated vegetation heights........... 50

Table 4.2 RMSE, normalized RMSE, bias, Pearson’s r and p values of estimated vegetation heights by strata ......................................................................................... 51

Table 4.3 RMSE, normalized RMSE, bias, Pearson’s r and p value of estimated mean vegetation heights (n=30) ......................................................................................... 53

Table 4.4 Root mean square error, Paired-sample z tests, Bias and Pearson’s r of fractional vegetation cover (%) ................................................................................. 59

Table 4.5 Cost analysis of UAV_RTK, UAV_LiDAR, and UAV_UAV .......................... 63
List of Figures and Illustrations

Figure 3.1 Location of 30 sample sites, distributed among four study areas in the Boreal Forest of northeastern Alberta, Canada ................................................................. 31

Figure 3.2 Different Environments: (a) Poor - Xeric site, dominated by jack pine with an understory of lichen; (b) Medium - Mesic site, dominated by trembling aspen; (c) Medium - Hydric site, dominated by black spruce and willow; (d) Poor-Hydric site, dominated by black spruce and graminoid; (e) Poor-Mesic site, dominated by black spruce with understory of Labrador tea and feather moss; (f) Post-fire forest, dominated by trembling aspen ........................................................ 32

Figure 3.3 Vegetation height sampling using point-intercept strategy on a 150m × 6m plot on a seismic line ................................................................................................. 34

Figure 3.4 Vegetation height measurement at a sample point (Modified from (J. N. Hird, Nielsen, McDermid, & Tan, 2016)) ........................................................................ 35

Figure 3.5 Flight pattern and ground control points distribution ........................................ 38

Figure 3.6 A sample dense point clouds from one of the study sites................................ 41

Figure 3.7 Three methods to estimate vegetation height: UAV_RTK, UAV_UAV and UAV_LiDAR ............................................................................................................ 43

Figure 4.1 Estimated mean vegetation heights versus field measurements: (a) UAV_RTK; (b) UAV_LiDAR; (c) UAV_UAV ........................................................................ 54

Figure 4.2 Long-transect profile comparison plot example: (a) Long-transect profile comparison plot; (b) Locations of long-transect measurement stations are shown with red dots ........................................................ 56

Figure 4.3 Cross-transect profile comparison plot example: (a) 60-meter cross transect; (b) 75-meter cross transect; (c) 90-meter cross transect; (d) Locations of cross-transect measurement stations are shown with red dots......................... 57

Figure 4.4 Estimated fractional vegetation cover versus field fractional vegetation cover at 0.5-2.0m stratum (%): (a) UAV_RTK; (b) UAV_LiDAR; (c) UAV_UAV ........................................................................ 60

Figure 4.5 Bottom-search radius versus RMSE of vegetation height estimates using UAV_UAV method ................................................................................................. 62

Figure 4.6 Top-search radius versus RMSE of height estimates (Site level) .................. 62

Appendix A Figures:
Figure A.1 Dimensions of Base Receiver ................................................................. 88
Figure A.2 Configuration of Rinex profile .............................................................. 89
Figure A.3 Input file of base station ....................................................................... 92
Figure A.4 Output file of base station .................................................................... 93

Appendix C Figures:
Figure C.1 Dense point clouds screenshots for all study sites ............................... 122

Appendix D Figures:
Figure D.1 legend of profile comparison plots in Figure D.2 and Figure D.3 ....... 123
Figure D.2 Profile comparison plots of long transects ......................................... 135
Figure D.3 Profile comparison plots of cross transects at 60 meters (left), 75 meters (middle) and 90 meters (right), respectively .................................................. 143
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSR</td>
<td>Bottom Search Radius</td>
</tr>
<tr>
<td>BVLOS</td>
<td>Beyond Visual Line of Sight</td>
</tr>
<tr>
<td>CHM</td>
<td>Canopy Height Model</td>
</tr>
<tr>
<td>CSRS</td>
<td>Canadian Spatial Reference System</td>
</tr>
<tr>
<td>DSM</td>
<td>Digital Surface Model</td>
</tr>
<tr>
<td>DTM</td>
<td>Digital Terrain Model</td>
</tr>
<tr>
<td>FVC</td>
<td>Fractional Vegetation Cover</td>
</tr>
<tr>
<td>GCP</td>
<td>Ground Control Point</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>LiDAR</td>
<td>Light Detection and Ranging</td>
</tr>
<tr>
<td>PPP</td>
<td>Precise Point Positioning</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>RTK</td>
<td>Real Time Kinematic</td>
</tr>
<tr>
<td>SfM</td>
<td>Structure from Motion</td>
</tr>
<tr>
<td>TSR</td>
<td>Top Search Radius</td>
</tr>
<tr>
<td>UAS</td>
<td>Unmanned Aerial System</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>VLOS</td>
<td>Visual Line of Sight</td>
</tr>
</tbody>
</table>
Chapter One: Introduction

1.1 Background and motivation

The rapid development of natural resource extraction in Alberta, Canada, has created large numbers of anthropogenic disturbance features such as roads, seismic lines, pipelines, and well sites, which have cumulative negative effects on wildlife habitat and biodiversity. In the province’s Boreal regions, threatened populations of woodland caribou (Rangifer tarandus caribou) are facing extirpation due to industrial development (Festa-Bianchet, Ray, Boutin, Côté, & Gunn, 2011). Studies have shown that linear disturbances are both directly and indirectly linked to decline of the species (Athabasca Landscape Team, 2009; Ritchie & George, 2012). For example, linear features increase the spatial overlap between caribou and wolves, their main predator, and provide movement corridors for wolf packs to travel and hunt more efficiently (Pyper, Nishi, & Mcneil, 2014). Indirectly, linear features also provide forage and access for other prey species such as elk and deer that would otherwise not be present in caribou habitat, which in turn increases carrying capacity for wolves and other predators (A David M Latham, Latham, Knopff, Hebblewhite, & Boutin, 2013; Andrew David M Latham, Latham, & Boyce, 2015). Among the linear disturbances, seismic lines – narrow corridors created to transport seismic equipment for geophysical exploration – have the largest impact on caribou in Alberta, due to their very slow reforestation rates, high density, and wide distribution (Van Rensen, Nielsen, White, Vinge, & Lieffers, 2015), as well as a lack of reclamation regulations.

In order to mitigate these negative ecological effects, both regulators and industry are making efforts to assess and enhance vegetation recovery on seismic lines through active reclamation and monitoring. Measuring vegetation recovery status on seismic lines
is important, because certain vegetation metrics, such as height and percent cover, can be related to behavioral or population responses from caribou, which in turn can be used to indicate whether a seismic line is recovered or not. For example, a Golder Associates study found that caribou preferentially select seismic lines with vegetation higher than 1.5m (Golder Associates Ltd, 2009). Van Rensen et al. suggested using 3m vegetation height to define a recovered seismic line, based on the minimum regeneration height required by Alberta’s forestry ground rules (Van Rensen et al., 2015; Alberta Environment and Development Sustainable, 2012). In this study, we focused on assessing remote-sensing techniques for estimating vegetation height and percent cover.

Traditionally, vegetation height and percent cover are measured in the field by trained technicians using hand-held instruments such as hypsometers and densiometers: a time-consuming activity that is expensive and difficult to scale (Magnussen & Russo, 2012). Remote sensing provides an attractive alternative to field surveys, with a demonstrated capacity to measure vegetation attributes quickly and effectively at a variety of scales (Lefsky, 2010; McRoberts, Cohen, Næsset, Stehman, & Tomppo, 2010; Weber & Boss, 2009). In particular, three-dimensional (3D) remote sensing technologies such as light detection and ranging (LiDAR) can provide accurate estimation of horizontal and vertical vegetation structure (Pan, Birdsey, Phillips, & Jackson, 2013; Bouvier, Durrieu, Fournier, & Renaud, 2015). As an active remote-sensing technology, LiDAR has the ability to penetrate canopies, and can therefore capture a complete vertical profile of vegetation (J Breidenbach, Nothdurft, & Kändler, 2010; Estornell, Ruiz, Velázquez-Martí, & Fernández-Sarria, 2011; M Vastaranta et al., 2011). However, LiDAR acquisitions are expensive, and data sets with densities high enough to characterize fine-scale structural
attributes on small disturbance features, such as 5-8m wide seismic lines, have high associated costs. This factor is exasperated in vegetation-recovery monitoring programs, which require repeated observations. As a result, there is a strong interest to develop alternative remote-sensing techniques aimed at reducing costs and increasing efficiency.

Unmanned aerial vehicle (UAV) systems are capable of acquiring 3D data with high spatial and temporal resolutions (Whitehead et al., 2014b; Lisein, Pierrot-Deseilligny, Bonnet, & Lejeune, 2013; Hardin & Jensen, 2011). Using a sequence of overlapping images collected by UAV platforms, high-density point clouds akin to LiDAR can be generated through structure-from-motion (SfM) workflows (Dandois & Ellis, 2010), thus providing information on both the horizontal and vertical profile of vegetation (Järnstedt et al., 2012; Puliti, Olerka, Gobakken, & Næsset, 2015). Compared to LiDAR acquired from piloted aircraft, UAV point clouds have lower cost, more flexible flight planning, and much greater point densities (J. White et al., 2013; Dandois & Ellis, 2013; Whitehead et al., 2014a). While some studies have documented relatively lower accuracy of vegetation-structural variables estimated with this technique (Järnstedt et al., 2012; Mikko Vastaranta et al., 2013), UAV photogrammetry provides a promising platform for vegetation-recovery monitoring on small anthropogenic disturbance features, and a potential complement to traditional field surveys.

The goal of this research was to develop a method to characterize vegetation structure on linear disturbances in forests using photogrammetric point clouds derived from UAV imagery, and to assess the accuracy and cost of different application scenarios. We wanted to know: could photogrammetric point clouds from UAVs provide an effective means of complementing or even replacing traditional ground surveys of vegetation height
and fractional vegetation cover (FVC)? As a passive technology, the primary limitation of UAV photogrammetry involves reliably capturing terrain elevation in vegetated scenes. In order to examine this factor, we assessed three different data sets: (1) UAV_RTK: wherein photogrammetric point clouds were supplemented with terrain observations acquired in the field with survey-grade real-time kinematic (RTK) global navigation satellite (GNSS) surveys; (2) UAV_LiDAR: where photogrammetric data were supplemented with spatially coincident LiDAR data; and (3) UAV_UAV: where UAV photogrammetry data were used alone. We compared vegetation height and FVC measurements extracted from remote-sensing data to field measurements acquired by trained personnel on 30 seismic lines at various stages of recovery.

1.2 Organization of thesis

Chapter one has introduced the background and motivation for characterizing vegetation structure on linear forest disturbances and the research objectives. Chapter two is a literature review of remote sensing point clouds for characterizing vegetation. Photogrammetric point clouds and LiDAR point clouds are introduced and compared. Chapter three describes the study area, data and methods of this research. Chapter four presents the results, including profile comparison plots, estimated vegetation at point level and site level, and estimated fractional vegetation cover at site level and cost analysis. Chapter five discusses the important findings of the work, comparison of this work to previous work, and limitations and error sources in this research. Chapter six summarizes the conclusions of the research, presents the significance and contributions of the research, and gives suggestions for future work.
Chapter Two: Background: Remote Sensing Point Clouds for Characterizing Vegetation, a Literature Review

Remote sensing point clouds have great capability to characterize vegetation structure. Point clouds provide three-dimensional (3D) information of ground targets, and are therefore capable of describing the horizontal and vertical profile of vegetation. Many studies have shown that remote-sensing point clouds can be used to estimate vegetation structural and biophysical parameters, such as tree height, diameter at breast height, volume (Ioki et al., 2010), biomass (García et al., 2010), and carbon density (Zolkos, Goetz, & Dubayah, 2013). There are two main types of remote-sensing point clouds: (i) photogrammetric point clouds and (ii) LiDAR point clouds. For these two types, there are two main approaches are used for estimating vegetation parameters from remote sensing point clouds: (i) the area-based statistical approach, and (ii) individual tree-detection approaches (Peuhkurinen et al., 2011; St-Onge et al., 2015; Nurminen et al., 2013). The area-based approach uses point cloud metrics of the entire plot, and creates statistical models to estimate vegetation parameters at the plot level. Compared to area-based approach, the individual tree detection approach requires detecting and delineating individual tree crowns first. Then, regression models or other statistical techniques are created for tree-segments in order to estimate tree-level parameters. This chapter first introduced data acquisition and processing workflow of photogrammetric point clouds. Then, this chapter compared photogrammetric point clouds and LiDAR in the use of characterizing vegetation structure.


2.1 Photogrammetric Point Clouds

Photogrammetric point clouds are generated from overlapping images acquired from different perspectives. The theoretical basis of photogrammetric point-cloud generation is stereophotogrammetry. Stereophotogrammetry involves determining the absolute and relative 3D location of an object from multi-view images of the object (Schenk, 2005). In the simplest case, two images of the same object from different views are acquired by a digital camera. Two common points, which refer to the exact same location on a given object, are identified in the two images. Then, two rays are constructed from the camera viewpoint and two common points, respectively. The 3D location of the point is derived from the intersection of the two rays (White et al., 2013). Repetition of the process enables generating many points forming a point cloud that represents the surface of the object (White et al., 2013).

Based on the theories from computer vision and stereophotogrammetry, the structure from motion (SfM) technique reconstructs the three-dimensional structure from a sequence of overlapping images. In this technique, the geometry of the scene, camera positions, and orientation can be solved automatically using a highly redundant bundle adjustment without knowing a-priori three dimensional positions (Westoby et al., 2012). Bundle adjustment is the problem of simultaneously refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameter estimates (Triggs et al., 2000). With SfM technique, three dimensional point clouds are generated in a relative image space coordinate system (Westoby et al., 2012). The transformation of the points from relative image space into absolute object-space coordinate system needs a small number of ground
control points with known absolute coordinates (Westoby et al., 2012). After that, the absolute coordinates of all points can be solved.

### 2.1.1 Data Acquisition

Images for generating photogrammetric point clouds can be acquired from digital cameras mounted on different remote-sensing platforms, such as piloted airplanes, unmanned aerial systems, and terrestrial platforms. Ginzler and Hobi (2015) presented a workflow to generate a surface model for Switzerland based on photogrammetric point clouds generated from digital airborne imagery. They also demonstrate the potential of using regular aerial images from national forest inventories. Ota et al. (2015) estimated aboveground biomass in seasonal tropical forest by using point clouds derived from digital airborne imagery. Granholm et al. (2015) used digital surface models based on aerial images for automated vegetation mapping. Aerial images cover a large extent, and are more appropriate for generating digital surface model over large areas. Typically, digital airborne imagery over large areas is less expensive to acquire than airborne LiDAR data due to the lower cost on sensors (Ginzler & Hobi, 2015; J. White et al., 2013). Terrestrial platforms have also been used to collect images for generating point clouds. Liang et al. (2014) utilized a hand-held consumer camera to generate point clouds and detect and model the individual tree stems. Unmanned aerial systems are very promising platforms for small-area inventory due to their advantages of low cost and flexibility. The utility of unmanned aerial systems for characterizing vegetation will be discussed in detail later in this section.

UAS (unmanned aerial systems) have a variety of environmental applications, such as agriculture (Zhang & Kovacs, 2012), wildlife studies (Sardà-Palomera et al., 2012),
hydrology (Rathinam et al., 2007), geomorphology (Hugenholtz et al., 2013), and forest inventory (Puliti et al., 2015). Compared to many other conventional airborne and satellite remote sensing, the most outstanding advantages of UAS are low cost and operational flexibility, which make them attractive to both commercial users and scientific researchers. Whitehead et al. (2014) summarized six advantages of UAS: (1) low cost; (2) ability to acquire data autonomously with minimized human operation; (3) manoeuvrability, especially for low-altitude flying and complex environments; (4) ability to operate in risky environments and unfavourable weather; and (5) reduced exposure risk to pilots. In addition, UASs are capable of acquiring data with high spatial and temporal resolution, indicating its great potential of timely and accurate measurements. All these advantages have enabled UAS become a promising remote sensing tool for small-area research or commercial applications.

A UAS typically consists of (1) an unmanned aerial vehicle (UAV); (2) a sensor payload (e.g. digital camera); (3) controller; (4) a navigational computer; (5) a ground-based pilot (UAV operator) and spotters if necessary (Whitehead et al., 2014b). The main differences between different designs for UAV platforms is their physical size and power, which limits their payload carry capacity, operating altitude and range (Anderson & Gaston, 2013). The size of UAVs can be classified as large, medium, small, mini, micro and nano. The characteristics of UAVs with different size is summarized in Table 1.1:

<table>
<thead>
<tr>
<th>Size</th>
<th>operating range</th>
<th>flight time</th>
<th>altitude</th>
<th>payload size</th>
</tr>
</thead>
</table>

Table 1.1 Operating range, flight time, altitude and payload size of UAVs with different sizes (Compiled from (Anderson & Gaston, 2013))
The lightweight UAV platforms are less expensive and more versatile than larger systems, and thus have a greater potential for environmental and ecological research (Anderson & Gaston, 2013; Lisein et al., 2013; Whitehead et al., 2014a). This section will only provide an overview of lightweight UAVs for characterizing vegetation. The small UAVs usually weigh less than 20 kilograms, have flight times of less than one hour, and have very limited ranges (Hardin & Jensen, 2011). Typically, there are two types of UAVs in the small size range: fixed and rotary wing platforms (Nex & Remondino, 2014). Fixed and rotary wings platforms are both used for forest and vegetation inventory. Table 1.2 gives some examples of the use of small UAVs for forest and vegetation inventory:

### Table 1.2 the use of UAS for vegetation and forest inventory

<table>
<thead>
<tr>
<th>UAS (name)</th>
<th>type</th>
<th>parameters</th>
<th>sensors</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large</td>
<td>Large (~ 500 km)</td>
<td>Long (up to 2 days)</td>
<td>Medium to high altitude (3-20km)</td>
<td>~200 kg internally and ~900 kg in under-wing pods</td>
</tr>
<tr>
<td>Medium</td>
<td>Large (~ 500 km)</td>
<td>Medium (~10 hours)</td>
<td>Medium (&lt; 4 km)</td>
<td>~50 kg</td>
</tr>
<tr>
<td>Small and mini</td>
<td>Small (&lt; 10 km)</td>
<td>Low (&lt; 2 hours)</td>
<td>low (&lt;1 km)</td>
<td>&lt; 30 kg (Small); up to 5 kg (mini)</td>
</tr>
<tr>
<td>Micro and nano</td>
<td>Small (&lt;10 km)</td>
<td>Very short (&lt;1 hour)</td>
<td>Very low (250 m)</td>
<td>Less than 5 kg</td>
</tr>
<tr>
<td>UAS Model</td>
<td>Type</td>
<td>Specifications</td>
<td>Camera Model</td>
<td>Reference</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Gatewing X100 small UAS</td>
<td>fixed wing</td>
<td>Wingspan: 100 cm; weight: 2 kg; cruise speed: 80 km/h; flight height: from 100 m to 750 m; maximum flight duration: 40 min</td>
<td>Ricoh GR3 still camera</td>
<td>(Lisein et al., 2013)</td>
</tr>
<tr>
<td>multirotor hexacopter UAV</td>
<td>rotary wing</td>
<td>Diameter 0.6 m; payload ~1.5 kg; maximum flight time of ~15 min</td>
<td>Canon ELPH 520 HS ‘point-and-shoot’ digital camera</td>
<td>(Zahawi et al., 2015)</td>
</tr>
<tr>
<td>SenseFly eBee UAV</td>
<td>fixed-wing</td>
<td>Weights approximately 537 g without payload; maximum flight time of 45 minutes</td>
<td>Canon S110 near infra-red (NIR) camera</td>
<td>(Puliti et al., 2015)</td>
</tr>
<tr>
<td>Gatewing X100 UAV</td>
<td>fixed-wing</td>
<td>Wingspan: 100 cm; take-off weight: 2.2 kg; flight duration: 45 min</td>
<td>RICOH GR Digital III camera</td>
<td>(Tuominen et al., 2015)</td>
</tr>
<tr>
<td>C-Astral Bramor Wing</td>
<td>fixed-wing</td>
<td>Wingspan: 230 cm; take-off weight: 4.2 kg; flight duration: 120 min</td>
<td>Olympus E-420 Sensor</td>
<td>(Tuominen et al., 2015)</td>
</tr>
</tbody>
</table>

An essential part of UAS for data acquisition is the flight planning and management subsystem (Colomina & Molina, 2014). Flight planning including a design of flying speed, direction, height, strip, waypoints, etc. When making a flight plan for photogrammetric
applications, the forward and cross overlap should be taken into careful consideration, since photogrammetry requires overlapping images, in order to create orthophotos or generate point clouds. Colomina & Molina (2014) points out that in order to compensate UAS instability, a requirement of flight plans for photogrammetry is large forward (80%) and cross (60-80%) overlap.

2.1.2 Generating and Processing Photogrammetric Point Clouds

There are three steps of reconstruction of three-dimensional scenes using the SfM technique (Verhoeven, 2010). The first step is alignment of photos (Verhoeven, 2010). The features in individual images are identified and then used for image correspondence (Westoby et al., 2012). Features are significant regions (buildings, ponds), lines (edge lines, roads, rivers) or points (corner points, line intersection) that are distinct and efficiently detectable in images, and expected to remain stationary (Zitová & Flusser, 2003). Feature points are detected and matched, so that the relative position of photographs can be computed. After feature detection, a sparse three-dimensional point cloud can be reconstructed based on the detected feature points. In addition, the estimation of the camera position and the internal calibration parameters are also main outputs of the step. The next step is the reconstruction of a dense three dimensional point cloud. A dense (pixel-based) multi-view reconstruction on aligned images is applied in order to build the details of the geometric scene (Verhoeven, 2010). Each pixel in one photograph is matched with another photograph. Thousands of points derived from the pixels form a dense point cloud (Bott, 2014). Finally, the point cloud can be presented as a mesh, and the mesh can be textured using the photographs (Verhoeven, 2010). Without geo-referencing, the point cloud does
not have absolute x, y and z coordinates. In order to relate the point cloud to real world, the ground control points surveyed in the field are used to transform the relative image-space coordinates into absolute object-space coordinates for each point.

In this procedure, feature detection and image matching are important components required for construction of dense point clouds and affect the quality of digital surface models (DSM). Many factors can affect the quality of image matching including: 1) little or no texture; 2) occlusion; 3) distinct object discontinuities; 4) repetitive objects; and 5) shadows caused by moving objects (Baltsavias et al., 2008). Baltsavias et al. (2008) also point out the difficulties of multi-temporal image matching when image scales, image qualities, illumination and atmospheric conditions are different. All these factors could affect the quality of image matching and the corresponding digital surface model.

2.1.3 Characterizing Vegetation by Photogrammetric Point Clouds

Most studies characterizing vegetation structure by photogrammetric point clouds are conducted in mature forests, both coniferous and deciduous. Only a few studies have investigated the feasibility of photogrammetric point clouds in a more variable vegetated environment. Zahawi et al. (2015) carried out a study at restoration sites of tropical forest by using area-based approach. Plots of three types of restoration treatments, “passive” (cattle excluded, no seedlings planted), “plantation” (mixed-species trees planted throughout plot), and “island” (same mixed-species trees planted but in patches with unplanted spaces between) were involved in their study. Their results show that the errors were substantial for passive treatment plots, due to the low vegetation heights. Zahawi et al. (2015) concluded that photogrammetric point clouds are less likely to be useful for
estimating heights of low stature vegetation. White et al. (2015) estimated the forest inventory attributes in a complex coastal forest environment by using the area-based approach. They evaluated the accuracy of estimated forest attributes across a series of strata with different slope (0-5°, 5-20°, 20-30°, or >30°) and canopy cover (0-10%, 10-50%, 50-90%, or 90-100%) by LiDAR and photogrammetric point clouds. They found that the similarity between metrics from the two data sources generally increased with increasing canopy cover, particularly for upper canopy metrics.

Creating canopy height models (CHMs) is useful for estimating vegetation parameters. The CHM is generated by subtracting a digital terrain model (DTM) from a digital surface model (DSM). The most common way is to use a DTM derived from LiDAR. This is because LiDAR has the capacity to penetrate the canopy and provide the information of the bare ground, while for photogrammetric point clouds, the ground information is obscured by the canopy, especially in the dense forest. Coregistration of the photogrammetric DSM with the LiDAR-DTM is a crucial step for generating CHMs, since the mismatch of DSM with the DTM leads to low accuracy of the canopy height model (Huang et al., 2009). However, the feasibility of this method may be affected when the LiDAR DTM has very low resolution or no existing LiDAR data is available. Zahawi et al. (2015) demonstrate two alternative methods to generate the digital terrain model, without using LiDAR-DTM: 1) they applied terrain filtering algorithms to the photogrammetric point clouds in order to generate DTM; and 2) differential GPS elevations were collected in the field and interpolated. Zahawi et al. (2015) concluded that canopy height models (CHMs) were strong predictors of field height regardless of whether a DTM was created directly from the photogrammetric point cloud or from differential GPS
measurements of the terrain. Therefore these authors demonstrated the potential of photogrammetric point clouds for characterizing vegetation in the region without existing LiDAR data.

The area-based approach is the most common method to estimate vegetation parameters by photogrammetric point clouds. White et al. (2013) describe the procedure of area-based approach for LiDAR point clouds. Since the photogrammetric point clouds share some common characteristics of LiDAR, the procedure of area-based approach is very similar. First, the images of the entire study area are acquired and point clouds are generated from the images. Point clouds are normalized by DTM. Field measurement of vegetation parameters is carried out in each sample plot. Then, the point clouds are clipped to the corresponding sample plots. In each plot, metrics (such as height percentiles) are calculated from the point clouds. Next, statistical regression models are built between point cloud-derived metrics and field measurements of vegetation parameters in each plot. Finally, the point cloud-derived metrics are calculated for the entire study area and the regression models are applied for wall to wall estimation of vegetation parameters at stand level.

The examples of studies that have applied area-based approach are summarized in Table 1.3. As I discussed before, most of these studies were conducted in mature forests. Only a few studies were carried out in a more complex forest environment. The area-based approach has been used for estimating forest attributes, including tree height, diameter, stem volume, basal area, stem number, canopy structure, above ground biomass, etc. Many studies show that the estimation of height usually has higher accuracy than other parameters (Vastaranta et al., 2013; Puliti et al., 2015; Järnstedt et al., 2012).
The point cloud-derived metrics often include height distribution metrics (height percentile, height standard deviation, etc.) and canopy cover (the proportion of points over given heights). Bohlin et al. (2012) added texture metrics as independent variables. Their results show that the best performing model is the combination of height percentile, canopy density and texture metrics, though the addition of texture metrics only improves the model modestly. Spectral metrics have been used in some studies. Puliti et al. (2015) added mean band values, band standard deviations, and band ratios of red, green and near-infrared as independent variables and evaluated the importance of these spectral variables. The comparison of different models revealed little improvements when spectral variables were added.

The most commonly used prediction model is linear regression. Logarithmically transformed linear regression is usually applied for estimating some forest inventory parameters, such as stem volume, basal area and Lorey’s height (Bohlin et al., 2012; Puliti et al., 2015). Some studies utilize non-parametric models. Random forest models are the most common non-parametric model (Vastaranta et al., 2013; White et al., 2015; Pitt et al., 2014).

**Table 1.3 Examples of studies of area-based approach with photogrammetric point clouds**

<table>
<thead>
<tr>
<th>Vegetation metrics</th>
<th>forest type</th>
<th>sample sizes</th>
<th>prediction methods</th>
<th>literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree height, coniferous hemiboreal forest</td>
<td>344</td>
<td>Linear regression models</td>
<td>(Bohlin et al., 2012)</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Site 1: 30</td>
<td>Site 2: 40</td>
<td>Site 3: 100</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>Stem volume, and basal area</td>
<td>Mean forest biomass: coniferous and deciduous</td>
<td>154</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>deciduous forest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>heights of the five tallest trees, aboveground</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Biomass density and carbon density</td>
<td>Lorey’s mean height, dominant height, stem number, basal area, and stem volume</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Boreal forest with coniferous trees (main) and deciduous trees</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lorey’s mean height, dominant height, mean basal area diameter, stem number, basal area, and volume</td>
<td>coniferous and deciduous forest</td>
<td>Dataset1: 797 Dataset2: 140</td>
<td>Linear regression models based on principle component analysis (PCA)</td>
<td>(Gobakken, Bollandsås, &amp; Næsset, 2015)</td>
</tr>
<tr>
<td>Canopy height, Above-ground biomass, Canopy structure (openness and roughness)</td>
<td>tropical forest restoration sites</td>
<td>38</td>
<td>Linear regression models</td>
<td>(Zahawi et al., 2015)</td>
</tr>
<tr>
<td>height, diameter, basal area, stem volume, and biomass</td>
<td>coniferous forest</td>
<td>500</td>
<td>Random forest models</td>
<td>(Mikko Vastaranta et al., 2013)</td>
</tr>
<tr>
<td>Lorey’s mean height, basal area, and gross volume</td>
<td>complex coastal forest environment</td>
<td></td>
<td>Random forest models</td>
<td>(J. White et al., 2015)</td>
</tr>
<tr>
<td>timber volume, stem density, basal area and</td>
<td>coniferous forest</td>
<td>44</td>
<td>multivariate k nearest neighbor (kNN) model</td>
<td>(Rahlf, Breidenbach, Solberg, Astrup,</td>
</tr>
</tbody>
</table>
The area-based approach uses point-cloud metrics of the plot and creates statistic models to estimate vegetation parameters. Compared to area-based approach, the individual tree detection approach requires detecting and delineating individual tree crowns. Individual tree crowns are delineated from point clouds by using segmentation algorithms (Rahlf et al., 2015). Individual tree detection enables the estimation of the attributes of individual tree crowns. In addition, individual tree detection approaches theoretically provide more detailed information and enable species classification based on object-oriented analysis.
While several researchers have applied area-based approaches for estimating vegetation parameters with photogrammetric point clouds, only few studies have utilized individual-tree detection approaches. Wężyk et al., (2012) created an algorithm suitable for crown segmentation for forests of different types, species, and ages. They also compared LiDAR and photogrammetric canopy height models for individual tree crown segmentation. The single-tree detection accuracy of LiDAR and photogrammetric canopy height models was 90.1% and 74.1% respectively (Wężyk et al., 2012). St-Onge et al. (2015) used the individual tree crown approach for characterizing the height and species composition of a boreal forest with photogrammetric point clouds. Their results show that the accuracy of species classification based on photogrammetric point clouds is very similar to that performed by LiDAR. The accuracy of classification is 79% for photogrammetric point clouds, and 83% for LiDAR, when using only three dimensional structure metrics (St-Onge et al., 2015). When three dimensional metrics and multispectral or intensity data were used, the accuracy increased to 89% for photogrammetric point clouds, and 86% for LiDAR. Waser et al. (2011) developed an approach for classification of tree species by geometric and spectral information derived from canopy height model based on multi-resolution image segmentation. The overall accuracies of classification vary between 76% and 83%.

The individual-tree detection approach assumes that there is only a single tree in one segment (Bergseng et al., 2015). However, segmentation and detection errors usually exist in individual-tree detection approach since only the large trees are detected while smaller trees are omitted (Solberg et al., 2006). In order to addresses the segmentation errors caused by omission of suppressed trees, another approach called semi-individual tree...
crown approach has been applied in some research. The difference of semi-individual tree crown approach from individual tree crown approach is that the tree crown segments can involve more than one tree (Rahlf et al., 2015). Rahlf et al. (2015) applied the semi-individual tree crown approach to predict forest parameters and compared the results to area-based approach. Overall, the model with semi-individual tree crown approach performed better than area-based approach. Wallerman et al. (2012) used semi-individual tree crown approach estimate tree height at segment level. They reported that the root mean square error of estimating maximum tree height was 34%.

2.2 LiDAR Point Clouds for Characterizing Vegetation

LiDAR is an important technology for characterizing 3D vegetation structure. LiDAR provides a distance measurement between a target and sensor based on half of the elapsed time between the emission of a laser pulse and the detection of a returned signal (Baltsavias, 1999). Airborne Laser Scanning is based on LiDAR from an aircraft and the precise orientation of these measurements (Hyyppä et al., 2008). The position and orientation of the LiDAR system are determined by a real-time DGPS (differential global positioning system) and an IMU (inertia measurement unit). The positions of all recorded returns are computed from the position and orientation of LiDAR system. Then, georeferenced point clouds are produced. The digital terrain models (DTM), digital surface models (DSM) and 3D models of object can be calculated based on the georeferenced point clouds (Hyyppä et al., 2008). A variety of LiDAR systems have been used for forestry applications. Discrete-return or full-waveform LiDAR systems are two main categories of LiDAR systems. Discrete-return systems allow for one or more discrete returns to be
recorded for each pulse, while a full-waveform system records the amount of energy for a
sequences of time intervals (Lim et al., 2003). The footprints for most discrete-return
systems are about 0.2 to 0.9 m, while for full-waveform systems, the footprints vary from
8 to 70m (Lim et al., 2003).

Two major methods have been employed for estimation of vegetation
characteristics with small-footprint, discrete-return LiDAR systems: area-based approach
and individual tree-detection approach. The basic principles of area-based approach and
individual tree detection approach have been introduced before in this paper when
discussing photogrammetric point clouds. Since the photogrammetric point clouds share
many similar characteristics with LiDAR point clouds, the approaches for characterizing
vegetation also have some similarities. The area-based approach assumes that the height
distribution of the point clouds strongly correlate with the forest attributes in a fixed area.
In the area-based approach, the empirical models are created between the forests
parameters and the height distribution of LiDAR point clouds, sometimes with the addition
of the intensity information. The individual-tree detection approach is based the
assumption that the individual trees can be segmented and delineated by LiDAR point
clouds. The individual tree-detection approach focuses on detecting and identifying
individual trees in order to acquire tree-level estimation of forest attributes (Peuhkurinen
et al., 2011). The area-level or stand-level information can be acquired by aggregation of
tree-level estimations. Peuhkurinen et al. (2011) summarized the similarities and the
differences of area-based and individual-tree detection approach. First, the required
minimum pulse densities are different. The area-based approach usually requires a
minimum pulse density of 0.5 pulse per square meter (White et al., 2013; Magnusson et
Individual tree-detection approach typically requires a minimum pulse density from 5 to more than 10 pulses per square meter (Vauhkonen et al., 2008; Shrestha & Wynne, 2012). Second, the required types of ground-sample data are different. The area-based approach only requires plot sample data (Tselamichael et al., 2010), while individual-tree detection approach requires measurements at the individual-tree level (Shrestha & Wynne, 2012). Table 1.4 gives some examples of the studies with LiDAR-based forest inventory applied area-based approaches. Table 1.5 provide some examples of individual-tree detection approach.

### Table 1.4 Examples of studies using the area-based approach (LiDAR)

<table>
<thead>
<tr>
<th>Vegetation metrics</th>
<th>Vegetation type</th>
<th>Purpose</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>wood volume, stem volume,</td>
<td>Coniferous, mixed and deciduous stands (a wide range of canopy structures sites)</td>
<td>To develop a new generalized modeling approach that does not required stepwise regression using a series of airborne LiDAR height metrics.</td>
<td>(Bouvier et al., 2015)</td>
</tr>
<tr>
<td>aboveground biomass, and basal-area</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>timber volume by tree species, stem number</td>
<td>Coniferous and deciduous forests</td>
<td>To compare three nonparametric k nearest neighbor (kNN) approaches - most similar</td>
<td>(Breidenbach et al., 2010)</td>
</tr>
<tr>
<td>distributions over diameter classes</td>
<td>neighbor inference (MSN), random forests (RF) and random forests based on conditional inference trees (CF).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>biomass</td>
<td>Coniferous and deciduous forests To explore how sample size, data type and prediction method affect the accuracy of biomass estimations. (Fassnacht et al., 2014)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>volume, number of stems and Lorey’s mean height</td>
<td>Coniferous and deciduous forests To examine different plot selection strategies of ground plots. (Maltamo et al., 2010)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>basal area</td>
<td>tropical moist forest To estimate the basal area across tropical forest with a great diversity of forest structures. (Vincent et al., 2012)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>biomass</td>
<td>Shrub vegetation To estimate dry biomass of shrub vegetation. (Estornell et al., 2011)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean tree height, dominant height, mean diameter, stem</td>
<td>Coniferous forests To test a two-stage sampling procedure for (Næsset, 2002)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number, basal area, and timber volume</td>
<td>Deciduous forests</td>
<td>To estimate biophysical attributes of hardwood forests.</td>
<td>(Lim et al., 2003)</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------</td>
<td>--------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Maximum tree height, Lorey’s mean tree height, mean diameter at breast height, total basal area, percent canopy openness, leaf area index, ellipsoidal crown closure, total aboveground biomass, total wood volume and stem density</td>
<td>Conservation forest dominated by deciduous trees and commercial forest dominated by coniferous trees</td>
<td>To estimate Gini efficient in order to identify the differences between the structural complexity of forest with different management history.</td>
<td>(Valbuena et al., 2016)</td>
</tr>
</tbody>
</table>
Biomass vegetation To conduct a meta-analysis of reported biomass estimation using LiDAR and show the differences in accuracy between different types of LiDAR, forest types and plot size. (Zolkos et al., 2013)

<table>
<thead>
<tr>
<th>Vegetation metrics</th>
<th>Forest type</th>
<th>Purpose</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree height, diameter at breast height (DBH) and stem volume</td>
<td>Coniferous forests</td>
<td>To predict tree attributes based on a detection method together with random forest regression technique.</td>
<td>(Yu et al., 2011)</td>
</tr>
<tr>
<td>timber volume</td>
<td>Forests dominated by coniferous trees</td>
<td>To propose a semi-individual tree crown approach that overcomes the main problems related to ITC by imputing ground truth data within crown</td>
<td>(Breidenbach et al., 2010)</td>
</tr>
</tbody>
</table>

Table 1.5 Examples of studies using the individual tree detection approach (LiDAR)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Forest Type</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height and crown diameters</td>
<td>Deciduous and coniferous forests</td>
<td>To investigate an adaptive single tree method for delineating tree crown.</td>
<td>(Ene et al., 2012)</td>
</tr>
<tr>
<td>Stem volume, stem number and basal area</td>
<td>Coniferous forests</td>
<td>To compare the area-based approach and individual tree detection approach to estimate stem number and basal area; compare different regression models to estimate stem volume.</td>
<td>(Lindberg &amp; Hollaus, 2012)</td>
</tr>
<tr>
<td>Total volume, number of stems, diameter and height of the basal area, median tree, tree size distributions, number and volume of saw-log-sized trees, and dominant height</td>
<td>Coniferous forests</td>
<td>To compare the area-based approach and individual tree detection approach to estimate forest characteristics.</td>
<td>(Peuhkurinen et al., 2011)</td>
</tr>
</tbody>
</table>
2.3 Comparison of Photogrammetric and LiDAR Point Clouds

While photogrammetric and LiDAR point clouds share some common characteristics, they have some main differences in terms of data acquisition, processing time, cost, and products (White et al., 2013). With regards to data acquisition, airborne LiDAR can be flown more hours per day and is less sensitive to bad weather conditions (Gehrke et al., 2008). The photography used to generate SfM point clouds is strongly affected by shadow, while LiDAR is insensitive to shadows. Solar illumination and view angles influence the image quality and the success of image matching. In terms of processing, photogrammetric point clouds require more processing time than LiDAR. After data acquisition, the direct product of LiDAR is the three dimensional point cloud. However, generating photogrammetric point clouds from a large number of images needs a complicated photogrammetric workflow, which requires longer processing time.

Photogrammetric and LiDAR point clouds have different characteristics, and generate different products. First, photogrammetric point clouds normally have much higher point density than LiDAR with a given cost, since dense image matching have the ability to produce a large number of points. The high point density of photogrammetric point clouds theoretically has the advantage of being able to delineate individual tree characteristics. For LiDAR, if the point density is often low, and only the area-based approach is feasible for extracting vegetation-structure information. Second, photogrammetric point clouds generated from images have spectral information associated with each point, which may provide more species-specific information. However, LiDAR point clouds typically do not provide spectral information, although there are some LiDAR instruments that are multispectral, and those that are mono-spectral do at least provide
information about the intensity of the return. While in some studies, LiDAR has been combined with multispectral images for analysis of species composition, the co-registration of LiDAR and images might have an effect on the results. Third, LiDAR, as an active system, has the ability to penetrate the canopy and provide the information of the understory and bare ground, whereas photogrammetric point clouds only delineate the upper canopy envelop (White et al., 2013). Compared to LiDAR, photogrammetric point clouds derived from images cannot provide information under the canopy. Thus, photogrammetric point clouds can only provide digital surface model (DSM) without other auxiliary data. Therefore, the digital canopy height model (CHM) is usually generated from photogrammetric point clouds by subtracting the digital terrain model (DTM) from LiDAR data.

Several studies have compared the forest attributes estimated by photogrammetric and LiDAR point clouds. Most studies found that LiDAR point clouds provide more-accurate estimates of forest attributes. Some studies found that the model outputs of two types point clouds are comparable and aerial photogrammetric point clouds can be used as alternative technology for forest inventory updates with minor sacrifices in precision. For example, Gobakken et al. (2015) estimated biophysical forest characteristics in three strata (young forest and mature forest on poor sites, and mature forest on good sites). They found that photogrammetric point clouds performed better in estimating height in young forest and mature forest on poor sites. However, for other biophysical forest characteristics (Lorey’s height, dominant height, mean basal area diameter, stem number, basal area, volume, above ground biomass), LiDAR data yielded more-accurate estimations. White et al. (2015) compared LiDAR and photogrammetric point clouds metrics and estimated
forest inventory attributes (Lorey’s height, basal area, and gross volume) in a complex coastal forest environment. While they found significant differences between two types of point cloud metrics for different strata, model outcomes were comparable for two types of point clouds. Vastaranta et al. (2013) compared two types of point clouds for forest mapping and inventory update, and demonstrated the capability of photogrammetric point clouds for change detection. They also found that photogrammetric point clouds are suitable for estimating forest attributes modelled from height, whereas LiDAR is more suitable to capture stand density. Straub et al. (2013) compared photogrammetric and LiDAR point clouds in estimation of basal area and timber volume in a mixed forest in Germany, and demonstrated that stereo aerial images are an alternative technology to LiDAR for modeling key forest attributes even for complex forest. Rahlf et al. (2014) compared four different three-dimensional remote sensing data sets for estimating timber volume. LiDAR provided the most accurate estimation at plot level with RMSE = 19%, whereas the RMSE for aerial photogrammetry was 31%. Pitt et al. (2014) predicted forest attributes described by stem size and growing stock with airborne LiDAR and aerial image-based point clouds. Their results imply that the predictions based on these two techniques are comparable in accuracy, though LiDAR provides slightly more accurate estimation.

This chapter reviewed relevant literatures on photogrammetric point clouds and LiDAR for characterizing vegetation structure. Also, we made a comparison between photogrammetric point clouds and LiDAR point clouds in terms of data acquisition, data processing, cost, and products.
Chapter Three: Methods

3.1 Study area

The research was conducted across four study areas in northeastern Alberta, Canada (Figure 3.1). The four areas are representative of different Boreal environments (Figure 3.2) with a variety of nutrient-moisture regimes and contain a total of 30 field sites: one seismic line per site. The first study area is about 1000km² (central coordinate: 56° 21' 23.43" N, -111° 16' 54.1" W) located near the town of Anzac, and includes a series of seismic lines on poor-mesic, medium-mesic, and medium-hygric sites. The second study area is about 400km² (central coordinate: 55° 26' 56.07" N, -110° 47' 58.28" W) south of Conklin, and includes mainly poor-hydric, medium-hydric, medium-hygric, poor-mesic, and medium-mesic sites. The third study area is about 450km² (central coordinate: 54° 59' 55.35" N, 111° 52' 49.70'' W) near the town of Lac La Biche, and is comprised of medium-mesic and poor-xeric sites. The fourth study area is about 380 km² (central coordinate: 57° 31' 56.42" N, 111° 16' 14.75" W) north of Fort MacKay, and includes mainly poor-xeric sites. All four study areas are located in the Boreal Forest natural region, which is characterized by a mixture of upland and wetland vegetation communities on gently undulating terrain (Natural Regions Committee, 2006). Upland areas are generally forested with deciduous, coniferous, and mixedwood stands, while lowland areas are comprised of fens and bogs. Main vegetation species include trembling aspen (Populus tremuloides), black spruce (Picea mariana), green alder (Alnus crispa), paper birch (Betula papyrifera), jack pine (Pinus banksiana), and willow (Salix spp.). Although the seismic lines in the four areas may have different vegetation recovery status, most were found to have low vegetation canopies (less than 1.0 meter).
Figure 3.1 Location of 30 sample sites, distributed among four study areas in the Boreal Forest of northeastern Alberta, Canada
Figure 3.2 Different Environments: (a) Poor - Xeric site, dominated by jack pine with an understory of lichen; (b) Medium - Mesic site, dominated by trembling aspen; (c) Medium - Hydric site, dominated by black spruce and willow; (d) Poor-Hydric site, dominated by black spruce and graminoid; (e) Poor-Mesic site, dominated by black spruce with understory of Labrador tea and feather moss; (f) Post-fire forest, dominated by trembling aspen
3.2 Field measurements

We used a point-intercept sampling strategy to select the location of individual measurement stations within a field site (a seismic line; Figure 3.3). At each site, a 150m-long transect was set up along the centerline of the seismic line using measuring tapes. From the start of the long transect to the 60m point, we placed measurement stations every 10m. From the 75m point to the 90m point, we made measurements every 1m. From the 90m point to the end point, we returned to making measurements every 10m. For each seismic line in the first three study areas (16 sites), three additional 6m long perpendicular cross-line transects were also set up at the 60m, 75m, and 90m points, respectively. On these cross transects, we made measurements every 0.5m.

At each station, we measured top vegetation height and the 3D coordinates of the ground point (easting and northing in UTM zone 12, datum NAD83, plus elevation). Vegetation height was determined with a measuring pole. The pole, which was marked every 5cm, was placed vertically on the ground at the measurement station. If any vegetation touched the pole, the height of the point where it touched was recorded (Figure 3.4). If there were multiple touches, only the maximum height was recorded. In order to ensure accurate and reliable field measurements of vegetation heights, only sample stations where vegetation height was lower than 3.0m were retained for later processing. This height (3.0m) coincides with the minimum regeneration height required by Alberta’s forestry ground rules. Following this procedure, a total of 1743 measurement stations were visited by field personnel in the summer (leaf on season) of 2016. These stations comprise the sample points that provided the basis for forthcoming statistical analyses. Summary statistics of field vegetation heights recorded are presented in Table 3.1.
The coordinates of each measurement station was determined with RTK GNSS survey techniques, which can achieve decimeter-level precision (Minist, 2013). RTK surveys measure the position of a rover relative to the base station in real-time, with the latter setup at a fixed location with clear view of the sky. We used precise point positioning (PPP) post processing to improve accuracy of the original RTK measurements. First, raw observations of the base station were recorded during the survey. Then, the CSRS-PPP (Canadian Spatial Reference System-Precise Point Positioning) online application was used for to refine the absolute positions of base station. After that, the differences between the original and PPP coordinates of base station were computed. Finally, we applied a 3D shift based on the computed differences for each site. PPP post processing can achieve absolute positioning accuracies of 10cm or less. The complete PPP workflow we used for RTK GNSS survey is provided in Appendix A.

**Figure 3.3 Vegetation height sampling using point-intercept strategy on a 150m × 6m plot on a seismic line**
Figure 3.4 Vegetation height measurement at a sample point (Modified from (J. N. Hird, Nielsen, McDermid, & Tan, 2016))

Table 3.1 Vegetation height statistics for the 1743 sample points measured in this study. Measurements were obtained by point-intercept sampling on 30 seismic lines

<table>
<thead>
<tr>
<th>Study area</th>
<th>Site ID</th>
<th>Mean (m)</th>
<th>Maximum (m)</th>
<th>Minimum (m)</th>
<th>Range (m)</th>
<th>Standard deviation (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMK</td>
<td>201</td>
<td>0.15</td>
<td>0.6</td>
<td>0</td>
<td>0.6</td>
<td>0.18</td>
</tr>
<tr>
<td>FMK</td>
<td>202</td>
<td>0.03</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.09</td>
</tr>
<tr>
<td>FMK</td>
<td>203</td>
<td>0.10</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.14</td>
</tr>
<tr>
<td>Code</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>FMK</td>
<td>204</td>
<td>0.18</td>
<td>1.2</td>
<td>0</td>
<td>1.2</td>
<td>0.29</td>
</tr>
<tr>
<td>FMK</td>
<td>205</td>
<td>0.02</td>
<td>0.25</td>
<td>0</td>
<td>0.25</td>
<td>0.05</td>
</tr>
<tr>
<td>FMK</td>
<td>218</td>
<td>0.00</td>
<td>0.05</td>
<td>0</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td>FMK</td>
<td>219</td>
<td>0.01</td>
<td>0.1</td>
<td>0</td>
<td>0.1</td>
<td>0.02</td>
</tr>
<tr>
<td>FMK</td>
<td>227</td>
<td>0.01</td>
<td>0.1</td>
<td>0</td>
<td>0.1</td>
<td>0.02</td>
</tr>
<tr>
<td>FMK</td>
<td>235</td>
<td>0.24</td>
<td>1.85</td>
<td>0</td>
<td>1.85</td>
<td>0.38</td>
</tr>
<tr>
<td>FMK</td>
<td>239</td>
<td>0.16</td>
<td>0.7</td>
<td>0</td>
<td>0.7</td>
<td>0.18</td>
</tr>
<tr>
<td>FMK</td>
<td>246</td>
<td>0.14</td>
<td>0.7</td>
<td>0</td>
<td>0.7</td>
<td>0.22</td>
</tr>
<tr>
<td>FMK</td>
<td>247</td>
<td>0.20</td>
<td>2.3</td>
<td>0</td>
<td>2.3</td>
<td>0.39</td>
</tr>
<tr>
<td>FMK</td>
<td>248</td>
<td>0.26</td>
<td>1.2</td>
<td>0</td>
<td>1.2</td>
<td>0.23</td>
</tr>
<tr>
<td>FMK</td>
<td>298</td>
<td>0.02</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.09</td>
</tr>
<tr>
<td>LLB</td>
<td>301</td>
<td>1.44</td>
<td>2.9</td>
<td>0</td>
<td>2.9</td>
<td>0.99</td>
</tr>
<tr>
<td>LLB</td>
<td>305</td>
<td>0.22</td>
<td>2.95</td>
<td>0</td>
<td>2.95</td>
<td>0.63</td>
</tr>
<tr>
<td>ANZ</td>
<td>306</td>
<td>0.25</td>
<td>0.6</td>
<td>0</td>
<td>0.6</td>
<td>0.15</td>
</tr>
<tr>
<td>ANZ</td>
<td>307</td>
<td>0.47</td>
<td>1.4</td>
<td>0.05</td>
<td>1.35</td>
<td>0.25</td>
</tr>
<tr>
<td>ANZ</td>
<td>308</td>
<td>0.32</td>
<td>1.4</td>
<td>0</td>
<td>1.4</td>
<td>0.27</td>
</tr>
<tr>
<td>ANZ</td>
<td>309</td>
<td>0.67</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0.53</td>
</tr>
<tr>
<td>ANZ</td>
<td>310</td>
<td>0.38</td>
<td>1.9</td>
<td>0</td>
<td>1.9</td>
<td>0.39</td>
</tr>
<tr>
<td>ANZ</td>
<td>311</td>
<td>0.18</td>
<td>1.5</td>
<td>0</td>
<td>1.5</td>
<td>0.29</td>
</tr>
<tr>
<td>ANZ</td>
<td>312</td>
<td>0.19</td>
<td>1.2</td>
<td>0</td>
<td>1.2</td>
<td>0.18</td>
</tr>
<tr>
<td>CNK</td>
<td>314</td>
<td>0.15</td>
<td>0.8</td>
<td>0</td>
<td>0.8</td>
<td>0.13</td>
</tr>
<tr>
<td>CNK</td>
<td>315</td>
<td>0.33</td>
<td>2.9</td>
<td>0.05</td>
<td>2.85</td>
<td>0.32</td>
</tr>
<tr>
<td>CNK</td>
<td>316</td>
<td>0.56</td>
<td>2.4</td>
<td>0.05</td>
<td>2.35</td>
<td>0.46</td>
</tr>
<tr>
<td>CNK</td>
<td>317</td>
<td>0.26</td>
<td>0.75</td>
<td>0</td>
<td>0.75</td>
<td>0.16</td>
</tr>
</tbody>
</table>
3.3 UAV imagery acquisition

A light-weight UAV quadcopter platform, the 3D Robotics Iris, was used to carry out image acquisition. Mission Planner, a free, open-source software package, was used for flight planning. Flight specifications were as follows – altitude: 42m; speed: 3m/s; forward overlap: 90%; and side overlap: 71%. The flight pattern is shown in Figure 3.5, and was designed to maximize photographic overlap along the measurement stations. Each flight was comprised of 12 passes 64m in length and 15m apart, oriented perpendicular to the seismic lines. Each flight covered at least 180m x 64m on the ground, and took about 15 minutes to complete.

A consumer-grade Nikon COOLPIX A digital camera (RGB) was mounted under the UAV. Camera configurations were set as follows – mode: TAv; intervalometer: 1 frame per second; aperture: F2.8 or F3.5; shutter speed: 1/2500 or 1/3200; ISO: auto / NR disabled; white balance: manual, calibrated against field reference at each site; resolution: L / 3:2 (4928 x 3264 effective); and focus: infinity (4m to infinity).

\[\begin{array}{cccccc}
\text{CNK} & 318 & 0.34 & 0.95 & 0 & 0.95 & 0.19 \\
\text{CNK} & 320 & 0.40 & 2.35 & 0 & 2.35 & 0.33 \\
\text{CNK} & 321 & 0.51 & 1.2 & 0.1 & 1.1 & 0.20 \\
\text{all} & \text{all} & 0.30 & 2.95 & 0 & 2.95 & 0.42 \\
\end{array}\]

\(^1\)FMK represents the study area near Fort MacKay; LLB represents the study area near Lac La Biche; ANZ represents the study area near Anzac; CNK represents the study area near Conklin.
Each site had 9 ground control points (GCPs) established for positional reference. Three GCPs were laid out along the seismic lines, and an additional 6 were laid out in the adjacent forest in locations that were visible to the sky. GCPs were marked on the ground using high-visibility flagging tape, and the coordinates measured by RTK GNSS with PPP post-processing.

**Figure 3.5 Flight pattern and ground control points distribution**

### 3.4 Generating and geo-referencing point-clouds

Photogrammetric point clouds were generated from overlapping images acquired from different perspectives along the UAV flight paths. The theoretical basis of photogrammetric point-cloud generation is stereo photogrammetry, which enables determining the absolute and relative three-dimensional (3D) location of an object from multi-view images of the object (Schenk, 2005). In the simplest case, two images of the same object from different points of views can be acquired with a digital camera. Two common points, which correspond to the exact same location on a given object, are then identified in the two images. Then, two rays are constructed from the camera viewpoints and two common points, respectively. The 3D location of the point can then be derived
from the intersection of the two rays (J. White et al., 2013). Repetition of this process generates many points and finally produces a point cloud that represents the surface of the target (J. White et al., 2013).

Based on the theories from computer vision and stereo photogrammetry, the structure from motion (SfM) workflow reconstructs the 3D structure from a sequence of overlapping images. With this technique, the geometry of the scene, camera positions, and orientation can be solved automatically using a highly redundant bundle adjustment without a-priori knowledge of the camera’s 3D position and orientation (Westoby et al., 2012). Bundle adjustment is the problem of simultaneously refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameter estimates (Triggs et al., 2000). With the SfM technique, 3D point clouds are generated in a relative image-space coordinate system (Westoby et al., 2012). The transformation of the points from relative image space into an absolute object-space coordinate system requires a small number of ground control points with known absolute coordinates (Westoby et al., 2012). After that, the absolute coordinates of all points can be resolved.

We used a commercial software package, Agisoft PhotoScan Professional Edition v1.2.4, to generate dense point clouds with absolute coordinates matching those of the field surveys. First, we removed images captured during take-off and landing, and loaded the remaining images into PhotoScan. Second, the images were aligned using the following parameters – accuracy: high; pair preselection: disabled; key point limit: 4000; and tie point limit: 4000. After alignment of images was completed, the relative camera position was resolved and sparse point clouds were generated. Third, dense point clouds were generated using the following configuration: quality – medium; depth filtering method – aggressive.
Fourth, we used a guided approach for marker (GCP) projection in PhotoScan. Markers were automatically placed on individual photos and then refined manually. Then, we entered the geographic coordinates and elevation of the GCPs and re-generated the dense point clouds. Finally, we exported the geo-referenced point clouds to LAS format for further processing. A perspective view of a sample point cloud from one of the study sites is shown in Figure 3.6. Full graphics showing dense point clouds for each of the 30 study sites are provided in Appendix C.
Figure 3.6 A sample dense point clouds from one of the study sites
3.5 LiDAR data

LiDAR data for the Anzac study area was provided by the Alberta’s Oil Sands Data project. The data were acquired by fixed-wing aircraft in August 2013. The average point density for a single flight line was approximately 5 points/m². Due to the fact that the majority of the area is covered by overlapping flight lines, the average point density in the final point cloud was close to 10 point/m². LiDAR data for the other study areas were acquired between 2010 and 2012, and were provided by the Government of Alberta. The average point density of these data was about 10 point/m². The relative and absolute accuracy of these datasets is 15cm and 30cm, respectively.

3.6 Estimating vegetation height

In order to save time for later processing, the geo-referenced UAV-based point clouds were clipped with a 3m circular buffer around each sample point using LAStools. Then, we used three different methods to estimate vegetation height: (1) UAV_RTK, (2) UAV_LiDAR, and (3) UAV_UAV. (Figure 3.6). For each of the methods, vegetation height of a sample point was estimated as the top value minus an estimate of terrain elevation. The top values for the each of these three methods are the same, but our strategy for estimating terrain elevations were different. Once again, UAV photogrammetry is a passive remote-sensing technology, and accurate measures of vegetation height require reliable estimates of terrain elevation. For a sample point with RTK-measured ground coordinates \((x_r, y_r, z_r)\), the top value was defined as the 99 percentile of \(z\) values of the UAV-based points \((x_l, y_l, z_l)\), where \(d_l\) is less than 0.2 m. Given the positioning accuracy
of the RTK measurements, the true location of any given sample point will be within this circle 19 out of 20 times.

\[ d_i = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} \]  

(1)

Figure 3.7 Three methods to estimate vegetation height: UAV_RTK, UAV_UAV and UAV_LiDAR

For the UAV_RTK method, terrain elevation \( (z_r) \) at the sample point was simply \( z_r \), as measured in the field. For the UAV_LiDAR method, terrain elevation was estimated as the minimum of \( z \) values of the supplementary LiDAR points \( (x_i, y_i, z_i) \) within a bottom-search radius (BSR). For UAV_UAV method, terrain elevation was estimated as the minimum of \( z \) values of the UAV-based points \( (x_i, y_i, z_i) \) within a BSR. Terrain within the
surveyed sites was relatively flat and smooth, which allowed for a relatively wide BSR search radius to be used. To ascertain which radius would be most suitable for our study, we computed the site-level root mean square error (RMSE) of vegetation height estimates derived from different values of BSR, starting at 25cm and going up to 3m at steps of 25 cm. Once the best BSR was chosen, we also wanted to investigate whether or not our choice of 20cm for the top-search radius (TSR) could have been improved with a similar strategy. To establish this, we tested a series of top-search radius distances, starting at 5cm and proceeding to 30cm at 5cm intervals. These distances (5cm and 30cm) are equivalent to 0.5x and 3x the horizontal positional accuracy of the RTK measurements, respectively.

3.7 Estimating fractional vegetation cover

Fractional vegetation cover (FVC) is the vertical projection of vegetation elements at a certain height stratum (relative to the ground surface) expressed as percent of the reference area (Purevdorj, Tateishi, Ishiyama, & Honda, 1998). We estimated FVC by height strata as the proportion of sample points in a given site that fell within each of three height strata: (i) 0–0.5m; (ii) 0.51–2.0m; and (iii) > 2.0m. These strata are commonly used in field protocols for certification of reclaimed well sites in Alberta (Mcintosh, 2014). Thus, vegetation cover in each plot was estimated using the ratio of vegetation hits at each height stratum to the total number of sample points \( n \) in the plot, expressed as a percentage. Let’s call the fractional vegetation cover of stratum I \( v_{cI} \), and so forth. The number of sample points where vegetation heights are within stratum I is \( n_I \). Then:

\[
v_{cI} = \frac{n_I}{n} \times 100\%
\]

(2)
Points where vegetation height was 0 (for $v_c$) or not in the interval of the stratum (for $v_{c_I}$, $v_{c_{II}}$ and $v_{c_{III}}$) did not contribute to the numerator of equation (2). Summary statistics of field fractional vegetation cover by height strata for the 30 field sites are shown in Table 3.2.

Table 3.2 Field fractional vegetation cover by height strata (statistics) (%)\(^1\)

<table>
<thead>
<tr>
<th>variable</th>
<th>$v_c$</th>
<th>$v_{c_I}$</th>
<th>$v_{c_{II}}$</th>
<th>$v_{c_{III}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>67</td>
<td>49</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>max</td>
<td>100</td>
<td>91</td>
<td>59</td>
<td>37</td>
</tr>
<tr>
<td>min</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>range</td>
<td>98</td>
<td>89</td>
<td>59</td>
<td>37</td>
</tr>
<tr>
<td>standard deviation</td>
<td>34</td>
<td>26</td>
<td>17</td>
<td>7</td>
</tr>
</tbody>
</table>

\(^1v_c\): overall fractional vegetation cover (vegetation cover at all height strata); $v_{c_I}$: fractional vegetation cover below 0.5m; $v_{c_{II}}$: fractional vegetation cover between 0.51 m and 2m; $v_{c_{III}}$: fractional vegetation cover above 2.0m.

3.8 Comparison and accuracy assessment

In order to compare the three different methods evaluated in this study, UAV_RTK, UAV_LiDAR, and UAV_UAV, we created graphical profile plots for visual comparison. In addition, we computed RMSE, normalized RMSE, and bias of estimates using the following equations:
\[ RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \]  
\[ Normalized\ RMSE = \frac{RMSE}{y_{max} - y_{min}} \]  
\[ Bias = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)}{n} \]

RMSE and bias in the 30 plots were computed for FVC, both overall and by stratum. For height, we computed accuracy statistics at both the point (n=1743) and aggregated site (n=30) level. For FVC, statistics were calculated only at the site level.

3.9 Cost analysis

We estimated the cost that a company would incur for surveying a similar study area under three possible scenarios. Given that the UAV_RTK method was only used for research purposes and would not be used in practice, we did not include it in this analysis. The exercise tracks the cost of surveying 30 sites, each consisting of a 150m long plot on a seismic line. The estimated costs include equipment, ancillary data purchase, data collection, and data processing. We assumed that a two-person field crew is needed for data collection with the cost as follows: $400/day per field technician; $300/day of per person for meals and accommodation during the field, $100/day for truck rental, and 2 extra days of salary, per diem and truck for outbound and inbound travel from headquarters. We also assumed a working day of 10 hours max: 1 hour for travel from accommodation to 1st plot, 1 hour for travel from last plot of the day to the accommodation, and 1 hour of travel between plots (includes loading and unloading equipment from truck to plot). Cost
of data collection in the field would be $1500/day. We assumed that salary for technician for data processing in the office is $400/day.

The three scenarios are:

1. **Traditional vegetation survey.** In this scenario, field crews perform traditional vegetation surveys in the field. Each plots takes 2 h for survey plus 1 h between plots. Thirty plots take 90 hours to complete: the equivalent of 9 days in the field. Adding the two additional travel days, the survey takes a total of 11 days.

2. **UAV photogrammetry surveys with supplementary LiDAR.** The UAV_LiDAR method is equivalent to having a LiDAR-derived digital terrain model (DTM) of 1-m pixel size, which would be a cheaper source of terrain elevation than the LiDAR point cloud used in our study. If we assume that the 30 sites are all located within a township, the project could purchase a DTM for this township for $100 within Alberta (http://www.altalis.com/products/terrain/lidar15_dem.html), though the costs would be exponentially higher if no such high-quality data existed. The total equipment cost in this scenario includes UAV: $1,500; RTK GNSS: $9,000; Camera: $500; and Accessories: $100. We assume that the equipment becomes obsolete in 5 years and is used in 40 field days per year. Then the equipment for a field-day is $55.5. Each plots takes 1 hour for UAV flight and GCP measurement, plus 1 hour between plots. 30 sites takes 60 hours, equivalent to 6 days for field work, plus 2 days for outbound and inbound travel. This scenario also involves post-processing through Agisoft PhotoScan Professional Edition software, with license cost of $5000/year. If we assume that this software will be used for five projects per year, then cost of software is $1000 per project. The data processing procedure including aligning photos, generating point clouds, refining GCP locations, re-
generating point clouds and estimating vegetation parameters, all of which except for refining GCP locations are automatic and do not produce labor costs. Manually refining GCP locations, loading data and organizing outputs would take about 2 days, with costs of $800.

(3) Stand-Alone UAV Photogrammetry Surveys. In practice, detailed geo-referencing of point clouds can be skipped in the UAV_UAV method, since vegetation parameters can be computed in a relative space. Therefore, we assume that no ground control points are required, and thus RTK GNSS equipment is not needed. The equipment cost includes UAV: $1,500; Camera: $500; and Accessories: $100. Given that the equipment is used in 40 field days/year for 5 years, equipment cost for a field-day is $ 10.5. Each plot takes 0.3 hour for UAV flight, plus 1 hour between plots. Thirty sites take 39 hours, equivalent to 4 field days, plus 2 days for travel. The survey takes a total 6 days, with costs of $ 9,000 for data collection and $ 63 for equipment. Cost of software is $1000. Regarding data processing, the difference between UAV_UAV and UAV_LiDAR is that the process of refining GCP locations and re-generating point clouds is unnecessary for UAV_UAV. Therefore, loading data and organizing outputs would take about 1 day, with cost of $ 400.
4.1 Estimated vegetation height at point level

Table 4.1 summarizes the statistics comparing each of the three remote-sensing derived estimates of vegetation height to field measurements at the point level (n=1743). The RMSE for UAV_RTK, UAV_LiDAR and UAV_UAV methods are 28cm, 31cm and 30cm, respectively; the normalized RMSE is 5% for all three; and the bias is 2cm, -2cm, and -3cm respectively. The correlation coefficients (Pearson’s r, n=1743) between field measurements and estimated vegetation heights were 0.76 for UAV_RTK, 0.72 for UAV_LiDAR, and 0.70 for UAV_UAV. The p values for paired-sample two-tailed z test were 0.001 for UAV_RTK, 0.020 for UAV_LiDAR and 0.000 for UAV_UAV, indicating that there are significant differences between field measurements and estimated vegetation heights at the point level.

Table 4.2 shows RMSE, normalized RMSE, bias, Pearson’s r and p values of estimated vegetation heights by height strata: stratum I (low): 0–0.5m; stratum II (medium): 0.51–2.0m; stratum III (high): > 2.0m). The RMSE of stratum I and II are low, whereas the RMSE of stratum III is high, suggesting that our methods works better in low to medium vegetation height strata than in high stratum. The three methods does not show large differences in low and medium strata, whereas the RMSE and bias of UAV_UAV in high strata are significantly larger than the other two methods. This indicates that UAV_UAV would underestimate vegetation height when vegetation is high. The correlation coefficients of all sample points is higher than that of each strata. Most of p values are lower than 0.05, suggesting that the differences between field measurements and estimated vegetation heights at point level are significant. Although there are two p values
is higher than 0.05 at stratum III, considering the small sample size, large RMSE and poor correlation, the estimated vegetation heights of stratum III at point level are still not reliable and accurate.

Our results suggest that at the point level, UAV photogrammetry cannot replace traditional field-based surveys of vegetation height across the range of conditions assessed in this study. The estimated vegetation height at point level can achieve the accuracy around 30 cm with or without supplemental terrain estimates. In addition, the performance of the three methods are very similar, which suggests that adding terrain data from field measurements or LiDAR does not significantly improves the accuracy in the conditions of our study. It also suggests that our methods perform better in low to medium vegetation than in high vegetation.

**Table 4.1 RMSE, normalized RMSE and bias of estimated vegetation heights**

<table>
<thead>
<tr>
<th>site ID</th>
<th>sample size</th>
<th>UAV_RTK</th>
<th></th>
<th>UAV_LiDAR</th>
<th></th>
<th>UAV_UAV</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE (m)</td>
<td>nRMSE (%)</td>
<td>bias (m)</td>
<td>RMSE (m)</td>
<td>nRMSE (%)</td>
<td>bias (m)</td>
</tr>
<tr>
<td>201</td>
<td>43</td>
<td>0.21</td>
<td>29</td>
<td>0.13</td>
<td>0.23</td>
<td>39</td>
<td>-0.15</td>
</tr>
<tr>
<td>202</td>
<td>43</td>
<td>0.15</td>
<td>22</td>
<td>0.11</td>
<td>0.1</td>
<td>19</td>
<td>-0.03</td>
</tr>
<tr>
<td>203</td>
<td>43</td>
<td>0.19</td>
<td>37</td>
<td>0.14</td>
<td>0.17</td>
<td>34</td>
<td>-0.1</td>
</tr>
<tr>
<td>204</td>
<td>42</td>
<td>0.23</td>
<td>22</td>
<td>-0.05</td>
<td>0.34</td>
<td>28</td>
<td>-0.18</td>
</tr>
<tr>
<td>205</td>
<td>43</td>
<td>0.06</td>
<td>31</td>
<td>0.05</td>
<td>0.05</td>
<td>21</td>
<td>-0.02</td>
</tr>
<tr>
<td>218</td>
<td>43</td>
<td>0.43</td>
<td>16</td>
<td>0.1</td>
<td>0.36</td>
<td>16</td>
<td>0.06</td>
</tr>
<tr>
<td>219</td>
<td>43</td>
<td>0.04</td>
<td>20</td>
<td>0.02</td>
<td>0.02</td>
<td>22</td>
<td>-0.01</td>
</tr>
<tr>
<td>227</td>
<td>43</td>
<td>0.08</td>
<td>40</td>
<td>0.07</td>
<td>0.03</td>
<td>26</td>
<td>-0.01</td>
</tr>
<tr>
<td>235</td>
<td>43</td>
<td>0.35</td>
<td>17</td>
<td>0.01</td>
<td>0.44</td>
<td>23</td>
<td>-0.23</td>
</tr>
<tr>
<td>239</td>
<td>43</td>
<td>0.17</td>
<td>27</td>
<td>0.09</td>
<td>0.24</td>
<td>33</td>
<td>-0.16</td>
</tr>
<tr>
<td>246</td>
<td>43</td>
<td>0.19</td>
<td>16</td>
<td>0.05</td>
<td>0.25</td>
<td>35</td>
<td>-0.13</td>
</tr>
<tr>
<td>247</td>
<td>43</td>
<td>0.19</td>
<td>17</td>
<td>-0.04</td>
<td>0.35</td>
<td>22</td>
<td>-0.18</td>
</tr>
<tr>
<td>248</td>
<td>43</td>
<td>0.17</td>
<td>21</td>
<td>-0.01</td>
<td>0.3</td>
<td>42</td>
<td>-0.24</td>
</tr>
<tr>
<td>298</td>
<td>43</td>
<td>0.08</td>
<td>17</td>
<td>-0.02</td>
<td>0.09</td>
<td>18</td>
<td>-0.02</td>
</tr>
</tbody>
</table>
Table 4.2 RMSE, normalized RMSE, bias, Pearson’s r and p values of estimated vegetation heights by strata

<table>
<thead>
<tr>
<th>Variable</th>
<th>UAV_RTK</th>
<th>UAV_LiDAR</th>
<th>UAV_UAV</th>
</tr>
</thead>
<tbody>
<tr>
<td>v̄h</td>
<td>0.28</td>
<td>0.22</td>
<td>0.3</td>
</tr>
<tr>
<td>v̄h_1</td>
<td>0.4</td>
<td>0.24</td>
<td>0.45</td>
</tr>
<tr>
<td>v̄h_II</td>
<td>0.81</td>
<td>0.24</td>
<td>0.87</td>
</tr>
<tr>
<td>v̄h_III</td>
<td>0.31</td>
<td>0.45</td>
<td>1.15</td>
</tr>
<tr>
<td>RMSE (m)</td>
<td>0.28</td>
<td>0.22</td>
<td>0.3</td>
</tr>
<tr>
<td>nRMSE (%)</td>
<td>5</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>Bias</td>
<td>0.02</td>
<td>-0.14</td>
<td>-0.3</td>
</tr>
<tr>
<td>Pearson’s r</td>
<td>0.76</td>
<td>0.55</td>
<td>0.09</td>
</tr>
<tr>
<td>p value</td>
<td>0.001</td>
<td>0.62</td>
<td>0.053</td>
</tr>
</tbody>
</table>

1 v̄h: overall vegetation height; v̄h_1: vegetation height below 0.5m; v̄h_II: vegetation height between 0.51 m and 2m; v̄h_III: vegetation height above 2.0m. Sample size of v̄h, v̄h_1, v̄h_II, and v̄h_III are 1743, 1402, 315 and 26 respectively.

51
p value: p value of two-tailed paired-sample z test between estimated vegetation height and field measurement. If Sig. < 0.05, there is a significant difference between the means of the two pairs.

4.2 Estimated mean vegetation height at site level

Table 4.3 summarizes the statistics comparing the three different methods for estimating mean vegetation height to field measurements at the site level (n=30). The RMSE for the UAV_RTK, UAV_LiDAR and UAV_UAV methods is 11cm, 15cm and 8cm, respectively; the normalized RMSE is 20%, 25% and 26%, respectively; and the bias is 0.03m, and -0.03m, and -0.02 m respectively. Paired-samples z tests (two-tailed) suggest that there are no significant differences between field-measured and remote-sensing measured estimates of mean vegetation height at plot level (p=0.09, 0.30, and 0.20 for UAV_RTK, UAV_LiDAR and UAV_UAV, respectively). In addition, correlation coefficients (Pearson’s r) shows that all three methods are highly associated with field measurements (r= 0.96, 0.91, and 0.95 for UAV_RTK, UAV_LiDAR and UAV_UAV, respectively). Figure 4.1 presents the estimated mean vegetation heights versus field measurements.

Our results suggest that at the aggregated site level, UAV photogrammetry could replace traditional field-based vegetation surveys of mean vegetation height across the range of conditions assessed in this study. In addition, UAV photogrammetry data alone (UAV_UAV) achieves the same level accuracy as those supplemented with field-(UAV_RTK) or LiDAR- (UAV-LiDAR) estimates of terrain, suggesting that no such supplements are necessary.
Table 4.3 RMSE, normalized RMSE, bias, Pearson’s r and p value of estimated mean vegetation heights (n=30)

<table>
<thead>
<tr>
<th>Statistics</th>
<th>UAV_RTK</th>
<th>UAV_LiDAR</th>
<th>UAV_UAV</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE (m)</td>
<td>0.11</td>
<td>0.15</td>
<td>0.08</td>
</tr>
<tr>
<td>nRMSE (%)</td>
<td>20</td>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>Bias (m)</td>
<td>0.03</td>
<td>-0.03</td>
<td>-0.02</td>
</tr>
<tr>
<td>Pearson’s r</td>
<td>0.96</td>
<td>0.91</td>
<td>0.95</td>
</tr>
<tr>
<td>p value</td>
<td>0.09</td>
<td>0.30</td>
<td>0.20</td>
</tr>
</tbody>
</table>

† p value: p value of two-tailed paired-sample z test between estimated fractional vegetation cover and field vegetation cover. If Sig. < 0.05, there is a significant difference between the means of the two pairs.

![Graph showing the relationship between UAV_RTK and Field Height](image_url)
Figure 4.1 Estimated mean vegetation heights versus field measurements: (a) UAV_RTK; (b) UAV_LiDAR; (c) UAV_UAV
4.3 Profile comparison plot

Profile comparisons for sample long- and cross-transect profiles are shown in Figure 4.2 and Figure 4.3, respectively. These visualizations reveal the generally strong agreement between field measurements at remote sensing estimates of height noted in the quantitative assessments, with only a few instances of mismatch. In areas of relatively tall vegetation – the top-most cross transect in Figure 4.3 (a), for example, and the 80-90m portion of the long transect in Figure 4.2 – we do observe some underestimation of vegetation height with UAV_UAV. Long- and cross-transect profiles for all 30 study sites are provided in Appendix D.
Figure 4.2 Long-transect profile comparison plot example: (a) Long-transect profile comparison plot; (b) Locations of long-transect measurement stations are shown with red dots.
Figure 4.3 Cross-transect profile comparison plot example: (a) 60-meter cross transect; (b) 75-meter cross transect; (c) 90-meter cross transect; (d) Locations of cross-transect measurement stations are shown with red dots.
4.4 Estimated Fractional Vegetation Cover at site level

Table 4.4 summarizes the statistics comparing the three different remote-sensing methods for estimating FVC to field measurements at the site level (n=30). The RMSE for the UAV_RTK, UAV_LiDAR and UAV_UAV methods is 36%, 30% and 47%, respectively; and Pearson’s r is 0.60, 0.85 and 0.48, respectively.

The RMSE of FVC for the UAV_RTK, UAV_LiDAR and UAV_UAV methods suggests that the estimated fractional vegetation covers from 0.5m to 2.0m and those above 2.0m have reasonable accuracy in the three methods. However, considering the fact that fractional vegetation cover above 2.0m for most sites (80%) is 0%, we cannot make a conclusion for this stratum. The RMSE of estimated fractional vegetation covers from 0.5m to 2.0m are: UAV_RTK: 7%, UAV_LiDAR: 14%, UAV_UAV:15%; the bias is -4.5%, -3.3% and -8.1%, respectively.

Figure 4.4 shows estimated fractional vegetation cover versus field fractional vegetation cover at 0.5-2.0m stratum (%). UAV_RTK estimates are highly correlated with field measurements (r=0.93); UAV_LiDAR estimates are moderately correlated with field measurements (r=0.69); UAV_UAV estimates are moderately correlated with field measurements (r=0.64).

Our results suggests that the three methods achieve accuracy of 15% or less for estimating fractional vegetation covers from 0.5m to 2.0m. The accuracy of UAV_RTK method is higher than the other two methods. However, the results of z tests suggests that there are significant differences between field measurements and estimated fractional cover.
Table 4.4 Root mean square error, Paired-sample z tests, Bias and Pearson’s r of fractional vegetation cover (%)

<table>
<thead>
<tr>
<th>Variable</th>
<th>UAV_RTK</th>
<th>UAV_LiDAR</th>
<th>UAV_UAV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>νc</td>
<td>νc₁</td>
<td>νc₂</td>
</tr>
<tr>
<td>RMSE (%)</td>
<td>36</td>
<td>38</td>
<td>7</td>
</tr>
<tr>
<td>Bias</td>
<td>24</td>
<td>28</td>
<td>-4.5</td>
</tr>
<tr>
<td>Pearson’s r</td>
<td>0.60</td>
<td>0.47</td>
<td>0.93</td>
</tr>
<tr>
<td>p value²</td>
<td>0.000</td>
<td>0.001</td>
<td>0.338</td>
</tr>
</tbody>
</table>

¹ νc: overall fractional vegetation cover (vegetation cover at all height strata); νc₁: fractional vegetation cover below 0.5m; νc₂: fractional vegetation cover between 0.51 m and 2m; νc₃: fractional vegetation cover above 2.0m.

² p value: p value of two-tailed paired-sample z test between estimated fractional vegetation cover and field vegetation cover. If Sig. < 0.05, there is a significant difference between the means of the two pairs.
Figure 4.4 Estimated fractional vegetation cover versus field fractional vegetation cover at 0.5-2.0m stratum (%): (a) UAV_RTK; (b) UAV_LiDAR; (c) UAV_UAV
4.5 Optimal search radius for the lowest and tallest UAV points around the sample points

Figure 4.5 shows the relationship between the bottom-search radius and RMSE of vegetation heights estimated using the UAV_UAV method. At the site level, the best results (0.08m RMSE) were obtained with a 1.0m bottom-search radius. This same radius also performed very well at the point level, though the RMSE is slightly higher (1cm) than that produced with a 1.25m radius. We elected to go with the smaller 1.0m bottom-search radius, since it places less pressure on our assumption that terrain elevation does not change significantly within our search radius. For the sake of consistency, we used the same radius for heights estimated with UAV_LiDAR.

Figure 4.6 shows the relationship between top-search radius and RMSE of height estimates at the site level for the all three methods. The best results for UAV_UAV (0.08m RMSE) were found using 0.2 m as the top-search radius. For UAV_LiDAR the best results (0.15m RMSE) came from search radii 0.2m or smaller. With the goal in an operational scenario being to minimize the RMSE when no ancillary data (LiDAR or RTK) are available, we elected to use 0.2m for the top-search radius.
Figure 4.5 Bottom-search radius versus RMSE of vegetation height estimates using UAV_UAV method

Figure 4.6 Top-search radius versus RMSE of height estimates (Site level)
4.6 Cost analysis

The costs of the three scenarios are summarized in Table 4.5. Based on this analysis, using UAV photogrammetry alone to produce vegetation parameters ($10,463) saves significant costs compared to traditional field surveys ($16,900). Supplementing photogrammetry data with a high-quality DTM increases costs to $14,344, largely on account of the increased data-collection costs associated with ground control and RTK GNSS surveys.

<table>
<thead>
<tr>
<th>Method</th>
<th>Data purchase</th>
<th>Equipment</th>
<th>Data collection</th>
<th>Software</th>
<th>Data processing</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>$0</td>
<td>$0</td>
<td>$16,500</td>
<td>$0</td>
<td>$400</td>
<td>$16,900</td>
</tr>
<tr>
<td>UAV_LiDAR</td>
<td>$100</td>
<td>$444</td>
<td>$12,000</td>
<td>$1,000</td>
<td>$800</td>
<td>$14,344</td>
</tr>
<tr>
<td>UAV_UAV</td>
<td>$0</td>
<td>$63</td>
<td>$9,000</td>
<td>$1,000</td>
<td>$400</td>
<td>$10,463</td>
</tr>
</tbody>
</table>
Chapter Five: Discussion

Our results show that at the aggregated site level, UAV photogrammetry alone (UAV_UAV) could replace traditional field-based vegetation surveys of mean vegetation height across the range of conditions assessed in this study, though significant differences remain between remote- and field-based vegetation surveys at point level. In addition, for estimation of vegetation height both at point level and site level, UAV photogrammetry alone (UAV_UAV method) achieves the same level of accuracy as those supplemented with field- (UAV_RTK) or LiDAR- (UAV-LiDAR) estimates of terrain. In addition, UAV photogrammetry data alone achieves the same level of accuracy as data supplemented with LiDAR in the estimation of fractional vegetation cover at 0.5-2.0m stratum, though its accuracy is a bit lower than data supplemented with field-surveyed terrain (UAV_RTK). The RMSE of estimated fractional vegetation covers in the 0.5-2.0m stratum were all within 15% of field-measured values, showing the efficiency of UAV photogrammetry for characterizing the structure of low-lying vegetation. Cost analysis indicates that using UAV point clouds alone provides substantial cost-saving over traditional field vegetation surveys.

In earlier studies on characterizing vegetation with UAVs, LiDAR data is often used to estimate terrain elevation and normalize UAV-based point clouds (e.g. Puliti et al., 2015; J. White et al., 2015). In this research, we used three different methods to estimate terrain elevation for normalization. Previous studies have tended to estimate vegetation parameters at the plot or stand level (e.g., Puliti et al., 2015; Maltamo et al., 2010). In this research, we used a point-intercept strategy and conducted a comparison of UAV-based estimates and field measurements at both point level and plot level. In addition, many
earlier studies have used LiDAR data to characterize vegetation in forests (e.g. Weber & Boss, 2009; Estornell et al., 2011), and new other works have started to explore the use of UAVs in mature forests. However, only few studies have characterized vegetation in forest disturbances (e.g., Zahawi et al., 2015; Jennifer N Hird et al., 2017). In this respect, the current study is relatively unique.

Although UAV_UAV methods produced good estimation of vegetation parameters in places when vegetation is not too dense, it is likely that UAV photogrammetry alone would tend to underestimate vegetation height in dense vegetation. This is because UAV images cannot penetrate the canopy and provide accurate terrain elevation in dense vegetation (J. White et al., 2015). Under these conditions, considering accuracy and cost, UAV photogrammetry supplemented with existing LiDAR (UAV_LiDAR) may be a better choice.

There are several error sources in this research. First, RTK measurements, though accurate in most cases, may have relative large errors when vegetation is high and dense. RTK GNSS survey technique relies on good communication between base station and rover, as well as good reception of satellite signals. Both of these two aspects require a clear sky. When the receiver is covered by dense vegetation, the solution of coordinates cannot be fixed and low accuracy may result. This problem exists both in measuring coordinates of ground control points and the sample points on transects. In order to reduce this error, we only recorded measurements under conditions where the solution of RTK GNSS could be fixed. Vertical errors caused by RTK GNSS only affect UAV_RTK method. However, horizontal errors of RTK GNSS would affect all three methods explored, particularly at the point level. This may have contributed to the significant
differences noted between the field- and remote sensing-derived estimates of height at the point level. Second, UAV photogrammetry supplemented with LiDAR (UAV_LiDAR) requires accurate co-registration between the two datasets. In our case, the absolute accuracy of LiDAR dataset was reported at 30cm, which could have contributed some error. We don’t believe this to be a significant problem, though, since the terrain on the sampled seismic lines was relatively flat, and LiDAR was only used to supplement terrain data.

Further research should explore the factors that affect accuracy of vegetation parameter estimates, and how to improve accuracy. The possible factors include image quality, flight conditions, vegetation density, presence of shadows on the lines, and phenological conditions. It would be valuable to analyze how those factors affect accuracy and how to eliminate errors by controlling some factors, such as avoiding unfavorable flight conditions. In addition, further research should investigate how to reduce processing time of generating point clouds. Producing high-density point clouds from images took large amounts of computational time for each site of our study.
Chapter Six: Conclusion

In Alberta’s boreal forests, anthropogenic disturbance features have caused serious ecological problems, including the decline of threatened species such as woodland caribou. To reduce this, governments and industry are making efforts to promote vegetation recovery on linear disturbances, especially seismic lines. Traditionally, field inventory is commonly used for monitoring vegetation recovery on human footprint features, which is costly, time-consuming, and labor-intensive. Recent studies have shown that UAVs have great promise for characterizing vegetation in a cost-efficient way. However, methodological analysis and cost assessment are rare in earlier studies. Prior to this research, a robust methodology for characterizing vegetation structure on linear disturbances with UAVs had not yet been developed. In addition, the relative cost of UAV photogrammetry to traditional methods was unknown.

The goal of this research was to develop a method to characterize vegetation structure on linear disturbances in forests using photogrammetric point clouds derived from UAV imagery, and to assess the accuracy and cost of different application scenarios. We wanted to know: could photogrammetric point clouds from UAVs provide an effective means of complementing or even replacing traditional ground surveys of vegetation height and fractional vegetation cover (FVC)? Is supplementary terrain data necessary for characterizing vegetation structure with UAV-based photogrammetry? How much cost could UAV-based photogrammetry save compared to traditional field vegetation survey?

Before conducting our experiment, a literature review was completed after organizing and summarizing relevant literature on remote sensing point clouds for characterizing vegetation. There are two main approaches for estimating vegetation
parameters from remote sensing point clouds: (i) the area-based statistical approach, and (ii) individual tree-detection approaches (Peukurinen et al., 2011; St-Onge et al., 2015; Nurminen et al., 2013). We summarized the examples of using these two approaches for photogrammetric point clouds and LiDAR, respectively. Also, we made a comparison between photogrammetric point clouds and LiDAR point clouds in terms of data acquisition, data processing, cost, and products.

Our experiment addressed the research objectives successfully. We conducted our study on 30 seismic line segments in Alberta’s Boreal forest. We used a point intercept sampling strategy to collect vegetation parameters in the field. A lightweight UAV loaded with a consumer-grade RGB camera was used to acquire imagery. Then, dense point clouds were generated in PhotoScan software. Then, three methods were used to estimate UAV-based vegetation height, where the difference between each method was how to estimate terrain elevation at the point: (1) UAV_RTK: wherein photogrammetric point clouds were supplemented with terrain observations acquired in the field with survey-grade real-time kinematic (RTK) global navigation satellite (GNSS) surveys; (2) UAV_LiDAR: where photogrammetric data were supplemented with spatially coincident LiDAR data; and (3) UAV_UAV: where UAV photogrammetry data were used alone. We found that at the aggregated site level, UAV photogrammetry alone (UAV_UAV) could replace traditional field-based vegetation surveys of mean vegetation height across the range of conditions assessed in this study, though it could not replace field-based vegetation surveys at point level. UAV_UAV achieves the same level of accuracy with UAV_RTK and UAV_LiDAR in estimation of vegetation height both at point level and site level. In addition, UAV_UAV
achieves the same level of accuracy with UAV_LiDAR in estimation of fractional vegetation cover at 0.5-2.0m stratum.

Costs of three application scenarios (traditional vegetation survey, UAV_LiDAR and UAV_UAV) with the same task of vegetation survey were assessed based on reasonable assumptions of time and labor needed. The result indicates that using UAV-based point clouds is much more cost-efficient than traditional field vegetation survey. In addition, UAV_UAV is more cost-efficient than UAV_LiDAR method.

In conclusion, we developed a reliable and cost-effective way to characterize vegetation structure on linear disturbances with UAV-based point clouds. We found that UAV-based photogrammetric point clouds alone are capable to characterize vegetation structure on linear forest disturbances when vegetation is low and open in a cost-efficient way.

6.1 Research contributions

This research contributes to both the field of unmanned aerial system and the field of ecological conservation. One the one hand, the methods, results and suggestions provided by this research helps UAV users to perform better practice of UAV photogrammetry in, but not limited to, vegetation and forest applications. I presented this research in the Unmanned Systems Canada Conference held on November, 2016 in Edmonton, Alberta titled “Characterizing Vegetation Structure on Anthropogenic Features in Alberta’s Boreal Forest with UAV (unmanned aerial vehicle)”

A second presentation at the Canadian Remote Sensing Society’s Earth Observation Summit entitled
Estimating Vegetation Parameters on Seismic Lines with UAV-based Point Clouds is planned for June 2017 in Montreal.

In addition, this contributes to ecological issues of seismic-line restoration and caribou conservation. Monitoring vegetation recovery status is an essential part of restoration of caribou habitat, as it is the most direct and reliable way to evaluate whether a restoration protocol is effective or not. Our research would contribute to establishing repeatable, cost-effective, and final scale vegetation and ecological monitoring strategies on seismic lines. I did a poster presentation in a Canadian Institute of Forestry Technical Conference on Seismic Line Restoration, titled “Characterizing Vegetation Structure on Seismic Lines Using UAV Point Clouds” in Edmonton, Alberta in December, 2016.

Last but not least, this research also contributes to the BERA (Boreal Ecosystem Recovery and Assessment) collaborative research project. For example, ecological researchers in BERA project use vegetation parameters as important independent variables in animal-response prediction models. Our procedure of UAV-based photogrammetry would largely improve the efficiency and reduce the cost of collecting vegetation data. Another example is that a Ph.D. student in BERA project is developing automatic detecting and measuring coniferous seedlings in active restoration seismic lines that builds on my methods.

6.2 Recommendation for future research

Although this research made contributions to the fields of both remote sensing and ecological conservation, several issues that are relevant but out of the scope of this study
deserve future research. This section gives some recommendations for the direction of future research.

First, future research should explore the factors that affect accuracy of vegetation parameter estimates and how to improve the accuracy. The possible factors can be image quality, flight condition, vegetation density, etc. It would be valuable to analyze how those factors affect accuracy and how to eliminate errors by controlling some factors, for example, avoiding unfavorable flight conditions. It is expected that collecting data in a more favorable weather and site condition would eliminate errors and further improve the accuracy of our method. In addition, our method works well when vegetation is open and low, but the performance of our methodology in dense vegetation is unknown due to the characteristics of our study site. Future research should test our methodology throughout sites with a variety of vegetation status, especially in dense vegetation.

Second, further research should investigate how to reduce processing time of generating point clouds from images. Producing high dense point clouds from images took several hours of computational time for each site of our study. Compared to LiDAR, generating point clouds from images takes much more processing time, especially for large study areas. Further researcher should explore how to reduce the processing time by improving software and hardware.

Third, currently, one of the main limitations of UAV is that it is always flown under visual line of sight (VLOS) operations, with a short flight range. Under VLOS operations, monitoring vegetation recovery on long linear features would need several missions of UAV, which increases the cost and difficulty in data collection and processing. UAS experts are now exploring best practice of beyond visual line of sight (BVLOS) operations,
which would largely expand the flight range of UAVs and thus enhance the capability of UAV in different applications. It would have a great value to explore how to apply UAV under BVLOS operations for monitoring vegetation recovery on seismic lines. Such projects would require the development of highly efficient workflows and processing strategies that would be capable of handling the variable conditions and large data volumes that would be encountered.
References


Hardin, P. J., & Jensen, R. R. (2011). Small-Scale Unmanned Aerial Vehicles in


http://doi.org/10.1191/0309133303pp360ra


http://doi.org/10.1093/forestry/cpq039


detection and the area-based statistical approach for the retrieval of forest stand
characteristics using airborne laser scanning in Scots pine stands. Canadian Journal

stereo imagery and airborne laser scanning for the area-based estimation of forest

http://doi.org/10.3390/rs70809632

percent vegetation cover and vegetation indices. International Journal of Remote
Sensing, 3519–3535(February). http://doi.org/10.1080/014311698213795

Pyper, M., Nishi, J., & Mcneil, L. (2014). Linear Feature Restoration in Caribou Habitat:
A summary of current practices and a roadmap for future programs.

Parameter Prediction Using an Image-Based Point Cloud: A Comparison of Semi-

four types of 3D data for timber volume estimation. Remote Sensing of Environment,


geometric variables derived from Airborne Digital Sensor (ADS40) and RC30 data.


http://doi.org/10.1016/j.rse.2010.08.006


http://doi.org/10.1016/j.geomorph.2012.08.021


http://doi.org/10.3390/f6103704


APPENDIX A: PPP (Precise Point Positioning) workflow for RTK GNSS survey

1. Rename binary files (bin.) for each site so that you know the site ID of each binary file.

2. Use RinexDesktop software to convert binary files to Rinex obs. files. Click “File” then “Load Binary file”. In configuration, edit Antenna Delta height. Note that Antenna Delta height is the height measured from marker to bottom of the antenna surface. However, in BERA survey, we measured from the upper edge of the base receiver to the marker as antenna height (AH). Thus, Antenna Delta height (ADH) should be compute use equation (1).

\[
ADH = AH - 0.0742m \tag{1}
\]

0.0742m is height from the upper edge to the bottom of the receiver (Figure A.1).

Figure A.1 Dimensions of Base Receiver
Leave Antenna Type blank. Click “Process Rinex” to convert binary file to Rinex observation file (.obs)

3. Put all obs. files into one folder and compress the folder to a zip file.

4. Open the PPP website: [https://webapp.geod.nrcan.gc.ca/geod/tools-](https://webapp.geod.nrcan.gc.ca/geod/tools-putils/ppp.php?locale=en) to conduct post processed PPP (precise point positioning). Choose “Static”, “2002.0” as epoch and “CGDV28(HT2_0)” as vertical datum. Then upload the zip file and click “Submit to PPP”.

5. After a while, you will receive an email of PPP results. The results contains a lot of details of the process, most of which is not needed for our purposes. The most important information is the coordinates of the base stations. Summarize the Northing, Easting
and Orthometric Height of each base station in an excel file. Now, we get the coordinates of each base station after post-processing.

6. In each survey, positions of rover were computed based on the coordinates of the base station before post-processing. In order to correct the positions of rover, we need to compute the differences between the coordinates of base station before and after post-processing, and then shift the entire survey.

7. Based on the idea described in step 6, first organize coordinates of base stations before post-processing. Base station coordinates were recorded in REF file for each job. Open REF file as txt and you will see latitude, longitude and ellipsoidal height of base station. Summarize them into an excel file. Then, convert latitude and longitude into UTM coordinates. You can use ArcGIS for this conversion.

8. Note that the height recorded in REF file is ellipsoidal height which is height above reference ellipsoid. However, the exported txt files of other points recorded orthometric height which is height above geoid. Ellipsoidal heights of base stations should be converted to orthometric height using GPS_H tool:


   In order to do a batch process by GPS_H online tool, organize data as “latitude” (1st column), longitude” (2nd column), and “ellipsoidal height” (3rd column) and save it as .csv file. Make sure it has a header.

   Check “Batch Processing”. Choose “HT2_0” as Geoid, “NAD83(CSRS)” as reference frame and type in “2002” as epoch. Then, upload the csv file. You will get orthometric height of each base station.
9. The next step is to conduct a 3D shift for each survey. A python script (3D_shift_v1.12.py) was written in order to do this task for all sites in a convenient way. Please see the Python script and its description:

**Script name:** 3D_shift v1.12

**Author:** Shijuan Chen

**Date:** 2016/10/12

**Function Description:** The script computes the difference between coordinates of the base station before and after post-processing and then applies a 3D shift to the entire survey to correct coordinates of all points in each survey. Example input and output files are also provided.

**Input file descriptions:**

**Input data 1:** csv. file of base station information.

Base station information should be organized as follow (Figure A.3):
The 1st row is header. You can change the text of header, but you should organize the data of each row in order. The 1st column is the sequence number (1, 2, ..., n). The 2nd column is Site ID. The 3rd column is X (Easting) coordinates of base station before post-processing. The 4th column is Y (Northing) coordinates of base station before post-processing. The 5th column is orthometric height of base station before post-processing. The 6th column is X (Easting) coordinates of base station after post-processing. The 7th
column is Y (Northing) coordinates of base station after post-processing. The 8th column is orthometric height of base station after post-processing.

Please note that the file should be saved as csv. format.

<table>
<thead>
<tr>
<th>No</th>
<th>Site_ID</th>
<th>EX_BF</th>
<th>NY_BF</th>
<th>OH_BF</th>
<th>EX_AF</th>
<th>NY_AF</th>
<th>OH_AF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>301</td>
<td>451128.2</td>
<td>6094418</td>
<td>590.676</td>
<td>451129.6</td>
<td>6094418</td>
<td>590.9792</td>
</tr>
<tr>
<td>2</td>
<td>304</td>
<td>456863.8</td>
<td>6078254</td>
<td>648.352</td>
<td>456864.4</td>
<td>6078254</td>
<td>648.2762</td>
</tr>
</tbody>
</table>

**Figure A.3 Input file of base station**

**Input data 2:** txt. files of RTK points coordinates exported from SurvCE software. Please note that all txt files should be saved in a common folder and the file name of each txt. file should be the Site ID of the study site. Otherwise, the script is not able to find correct txt. file.

**Output file descriptions:**

**Output data 1:** csv. file of base station info.

The first 8 columns are the same as input file. The 9th column is the differences between X (Easting) coordinates of base station before and after post-processing. The 10th column is the differences between Y (Northing) coordinates of base station before and after post-processing. The 11th column is the differences between orthometric height of base station before and after post-processing.
Output data: txt. files of RTK points coordinates after three dimensional shift. Please note that the txt. file only records Point ID, Y coordinates, X coordinates, and orthometric height. The descriptions of points are not recorded.

Python Script:

```python
# This script conducts a 3D shift for RTK points.
import csv
import os

def ThreeD_shift(base_data_path, rtk_folder_path, output_folder_path):
    class site_base_crd(object):
        def __init__(self, no, site_id, ex_bf, ny_bf, oh_bf, ex_af, ny_af, oh_af, dx, dy, dh):
            self.no = no
            self.site_id = site_id
            self.ex_bf = ex_bf
            self.ny_bf = ny_bf
            self.oh_bf = oh_bf
            self.ex_af = ex_af
            self.ny_af = ny_af
            self.oh_af = oh_af
            self.dx = dx
            self.dy = dy
            self.dh = dh

    site_base_list = []

    with open(base_data_path, 'rb') as fb:
        base_reader = csv.reader(fb)
        row1 = next(base_reader)
        b_list = list(base_reader)
        site_num = base_reader.line_num - 1
        print "number of sites: ", site_num

        for i in range(0, site_num):
```
site_base_list.append(site_base_crd(0,0,0,0,0,0,0,0,0,0,0))

    for i in range(0, site_num):
        site_base_list[i].no = b_list[i][0]
        site_base_list[i].site_id = b_list[i][1]
        site_base_list[i].ex_bf = float(b_list[i][2])
        site_base_list[i].ny_bf = float(b_list[i][3])
        site_base_list[i].oh_bf = float(b_list[i][4])
        site_base_list[i].ex_af = float(b_list[i][5])
        site_base_list[i].ny_af = float(b_list[i][6])
        site_base_list[i].oh_af = float(b_list[i][7])
        site_base_list[i].dx = site_base_list[i].ex_af - site_base_list[i].ex_bf
        site_base_list[i].dy = site_base_list[i].ny_af - site_base_list[i].ny_bf
        site_base_list[i].dh = site_base_list[i].oh_af - site_base_list[i].oh_bf

    fb.close()

    if not os.path.exists(output_folder_path):
        os.makedirs(output_folder_path)
        base_newfile_path = output_folder_path + r"\Base_info.csv"

    with open(base_newfile_path, 'wb') as fb_w:
        base_writer = csv.writer(fb_w)
        base_writer.writerow(['No', 'Site_ID', 'EX_BF', 'NY_BF', 'OH_BF', 'EX_AF', 'NY_AF', 'OH_AF', 'DX', 'DY', 'DH'])

            for i in range(0, site_num):
                base_writer.writerow([site_base_list[i].no, site_base_list[i].site_id, site_base_list[i].ex_bf, site_base_list[i].ny_bf, site_base_list[i].oh_bf, site_base_list[i].ex_af, site_base_list[i].ny_af, site_base_list[i].oh_af, site_base_list[i].dx, site_base_list[i].dy, site_base_list[i].dh])

        print 'Base info was saved in', base_newfile_path

    fb.close()

class rtk_crd(object):
    def __init__(self, site_id, point_id, x, y, h):
        self.site_id = site_id
        self.point_id = point_id
        self.x = x
        self.y = y
        self.h = h
rtk_newfolder_path = output_folder_path + '\'

for k in range(0, site_num):
    rtk_file_path = rtk_folder_path + '\' + site_base_list[k].site_id + '.txt'
    with open(rtk_file_path) as fr:
        rtk_point_list = []
        rtk_point_newlist = []
        rtk_point_num = 0
        for line in fr:
            rtk_row = line.split(',')
            rtk_point_list.append(rtk_crd(site_base_list[k].site_id, rtk_row[0], rtk_row[2], rtk_row[1], rtk_row[3]))
            rtk_point_num += 1
        print "Number of RTK points in site ", site_base_list[k].site_id, ' is ', rtk_point_num, '.

        for i in range (0, rtk_point_num):
            rtk_af_x = float(rtk_point_list[i].x) + site_base_list[k].dx
            rtk_af_y = float(rtk_point_list[i].y) + site_base_list[k].dy
            rtk_af_h = float(rtk_point_list[i].h) + site_base_list[k].dh

            rtk_point_newlist.append(rtk_crd(rtk_point_list[i].site_id, rtk_point_list[i].point_id, rtk_af_x, rtk_af_y, rtk_af_h))

        rtk_newfile_path = rtk_newfolder_path + site_base_list[k].site_id + '_new.txt'
        with open(rtk_newfile_path,'w') as fw:
            for i in range(0, rtk_point_num):
                fw_line = rtk_point_list[i].point_id + ',' + '%.4f' % rtk_point_newlist[i].y + ',' + '%.4f' % rtk_point_newlist[i].x + ',' + '%.4f' % rtk_point_newlist[i].h + '
'
                fw.write(fw_line)
        fw.close()

    print 'New coordinates were saved in ', rtk_newfile_path
fr.close()

# base_data_path = 'E:\BERA\Post Processing\3D_shift\data\Base_BF_AF.csv'
# rtk_folder_path = 'E:\BERA\Post Processing\3D_shift\data'
# output_folder_path = "E:\BERA\Post Processing\3D_shift\output"

print "\n-----------------------------------------------------------
----------------
"
"Script name: 3D shift V1.12  Author: Shijuan Chen  Date: 2016/10/12"
"Description: This python script makes a three dimensional shift of the RTK points based on "
"base station coordinates before and after post-processing of each survey job. \n"
"Please see the specific requirements of input data format in '3D_shift_help.doc' file. \n"
"-------------------------------------------------------------
-------------------
base_data_path = raw_input("Please input base station data file path:")
rtk_folder_path = raw_input("Please input rtk txt folder path:")
output_folder_path = raw_input("Please input output folder path:")

ThreeD_shift(base_data_path, rtk_folder_path, output_folder_path)
raw_input("Please press any key to exit.")
APPENDIX B: Core Python Scripts

The core python functions used in this research program are as follow:

```python
# -----------------------------------------------------------------------------------------------------------#
# Script name: BERA_core19
# Description: This script includes core classes and functions used in the Master's thesis.
# Author: Shijuan Chen
# Latest version Date: 05/12/2017
# -----------------------------------------------------------------------------------------------------------#
import math

class rtk_crd(object):
    def __init__(self, site_id, point_id, x, y, ter_h):
        self.site_id = site_id
        self.point_id = point_id
        self.x = x
        self.y = y
        self.ter_h = ter_h

class veg(object):
    def __init__(self, site_id, point_id, gr_veg_h):
        self.site_id = site_id
        self.point_id = point_id
        self.gr_veg_h = gr_veg_h

class rtk_veg(rtk_crd):
    def __init__(self, site_id, point_id, x, y, ter_h, gr_veg_h):
        super(rtk_veg, self).__init__(site_id, point_id, x, y, ter_h)
        self.gr_veg_h = gr_veg_h

class rtk_veg_cmp(rtk_veg):
    def __init__(self, site_id, point_id, x, y, ter_h, gr_veg_h, pt_veg_h):
        super(rtk_veg, self).__init__(site_id, point_id, x, y, ter_h, gr_veg_h)
        self.pt_veg_h = pt_veg_h

    def cal_delta_h(self, pt_veg_h, gr_veg_h):
        # cal_delta_h calculates the differences between point cloud measurement and ground veg height
        delta_h = pt_veg_h - gr_veg_h


class pt_2d(object):
    def __init__(self, x, y):
        self.x = x
```

97
def _init_(self, x, y, h):
    super(pt_3d, self)._init_(x, y)
    self.h = h

#--------------------------------------------------#
def hlist_inBuffer(pt_2d_x, pt_2d_y, pt_3d_list_x, pt_3d_list_y,
pt_3d_list_h, bufferDistance):
    num = len(pt_3d_list_x)
    if not (num == len(pt_3d_list_y)):
        if not (len(pt_3d_list_y) == len(pt_3d_list_h)):
            return
    return

    hlist = []
    for i in range(0, num):
        dist = (pt_2d_x - pt_3d_list_x[i]) ** 2 + (pt_2d_y -
        pt_3d_list_y[i]) ** 2
        if math.sqrt(dist) < bufferDistance:
            hlist.append(pt_3d_list_h[i])
    return hlist

#--------------------------------------------------#
import glob
import os
from liblas import file
import numpy
import BERA_classes as bc

def CmpVegHtCyl_UAV_RTK(las_ifolder_path, txt_ifolder_path,
    txt_ofolder_path, top_radius = 0.2):
    if not os.path.exists(las_ifolder_path):
        print "Could not find ", las_ifolder_path, "!"
        if not os.path.exists(txt_ifolder_path):
            print "Could not find ", txt_ifolder_path, "!"
            return
    return
    if not os.path.exists(txt_ofolder_path):
        os.makedirs(txt_ofolder_path)

    las_file_list = glob.glob(os.path.join(las_ifolder_path,
        "*.las"))
    txt_file_list = glob.glob(os.path.join(txt_ifolder_path,
        "*.txt"))

    las_file_num = len(las_file_list)
txt_file_num = len(txt_file_list)
if not (las_file_num == txt_file_num):
    print "Number of las and txt files does not match!"
    return

for k in range(0, las_file_num):
    print las_file_num
    with open(txt_file_list[k]) as fr_txt:
        with open(las_file_list[k]) as fr_las:
            # read las
            pt_3d_las = []
            fr_las = file.File(las_file_list[k], mode='r')

            pt_list_x = []
            pt_list_y = []
            pt_list_h = []

            print len(fr_las)
            for f in fr_las:
                pt_list_x.append(f.x)
                pt_list_y.append(f.y)
                pt_list_h.append(f.z)

            print "Reading las file ", las_file_list[k]

            fr_txt.readline()
            site_id = fr_txt.readline().split(',')[0]
            fr_txt.seek(0)
            output_file_name = txt_ofolder_path + '\\' + site_id + '_cmp UR.txt'

            with open(output_file_name, 'wb') as fw:
                fw.write('site_id,point_id,x,y,ter_h,gr_veg_h,las_x,las_y,las_h,pt_veg_h,delta_h\n')

                # read txt, search, and write to txt file
                fr_txt.readline()
                for line in fr_txt:
                    row = line.split(',')
                    txt_pt = bc.rtk_veg(str(row[0]),
                                        str(row[1]),
                                        float(row[2]),
                                        float(row[3]),
                                        float(row[4]),
                                        float(row[5]))

                    top_hlist = bc.hlist_inBuffer(txt_pt.x,
                                                  txt_pt.y,
                                                  pt_list_x,
                                                  pt_list_y,
                                                  pt_list_h,
                                                  top_radius)
                    if top_hlist:
                        top_h = numpy.percentile(top_hlist, 99)
                    else:
                        top_h = 999
\[
\begin{align*}
\text{pt_veg_h} &= \text{top_h} - \text{txt_pt.ter_h} \\
\delta_h &= \text{pt_veg_h} - \text{txt_pt.gr_veg_h} \\
\text{txt_pt.x}, \text{txt_pt.y}, \text{txt_pt.ter_h}, & \backslash \text{n}
\end{align*}
\]
# print \text{txt_pt.site_id}, \text{txt_pt.point_id},
\text{min_dis}, \text{pt_veg_h}, \delta_h
# print '-----------------------------------

\text{w_line} =
'\%s,\%s,\%0.3f,\%0.3f,\%0.3f,\%0.3f,\%0.3f,\%0.3f,\%0.3f,\%0.3f,\%0.3f\text{n}' \% \\
(str(\text{txt_pt.point_id}), \text{txt_pt.x}, \text{txt_pt.y}, \text{txt_pt.ter_h}, \\
\text{txt_pt.gr_veg_h}\backslash \text{n}, \text{txt_pt.x}, \text{txt_pt.y}, \text{top_h}, \\
\text{pt_veg_h}, \delta_h)

\text{fw.write}(\text{w_line})
\text{print 'Output file saved in ', output_file_name}
\text{fw.close()}
\text{fr_las.close()}
\text{fr_txt.close()}

#---------------------------------------------------------------

\text{def CmpVeghtCyl_UAV_LDA(Ulas_ifolder_path, Llas_ifolder_path,}
\text{txt_ifolder_path, txt_ofolder_path, top_radius = 0.2, bottom_radius = 1.0):}
\text{if not os.path.exists(Ulas_ifolder_path):}
\text{print "Could not find ", Ulas_ifolder_path, "!"}
\text{if not os.path.exists(Llas_ifolder_path):}
\text{print "Could not find ", Llas_ifolder_path, "!"}
\text{if not os.path.exists(txt_ifolder_path):}
\text{print "Could not find ", txt_ifolder_path, "!"}
\text{return}
\text{return}
\text{return}
\text{if not os.path.exists(txt_ofolder_path):}
\text{os.makedirs(txt_ofolder_path)}

\text{Ulas_file_list} = \text{glob.glob(os.path.join(Ulas_ifolder_path,}
\text{\'*.las\')})
\text{Llas_file_list} = \text{glob.glob(os.path.join(Llas_ifolder_path,}
\text{\'*.las\')})
\text{txt_file_list} = \text{glob.glob(os.path.join(txt_ifolder_path,}
\text{\'*.txt\')})
\text{Ulas_file_num} = \text{len(Ulas_file_list)}
\text{Llas_file_num} = \text{len(Llas_file_list)}
\text{txt_file_num} = \text{len(txt_file_list)}
if (Ulas_file_num <> txt_file_num) or (Llas_file_num <>
txt_file_num) or (Ulas_file_num <> Llas_file_num):
    print "Number of las and txt files does not match!"
    return

for k in range(0, Ulas_file_num):
    print Ulas_file_num
    with open(txt_file_list[k]) as fr_txt:
        with open(Ulas_file_list[k]) as fr_Ulas:
            # read Ulas
            fr_Ulas = file.File(Ulas_file_list[k], mode='r')
            Upt_list_x = []
            Upt_list_y = []
            Upt_list_h = []

            print len(fr_Ulas)
            for f in fr_Ulas:
                Upt_list_x.append(f.x)
                Upt_list_y.append(f.y)
                Upt_list_h.append(f.z)

            print "Reading las file ", Ulas_file_list[k]

    with open(Llas_file_list[k]) as fr_Llas:
        # read Llas
        fr_Llas = file.File(Llas_file_list[k], mode='r')
        Lpt_list_x = []
        Lpt_list_y = []
        Lpt_list_h = []

        print len(fr_Llas)
        for f in fr_Llas:
            Lpt_list_x.append(f.x)
            Lpt_list_y.append(f.y)
            Lpt_list_h.append(f.z)

        print "Reading las file ", Llas_file_list[k]

        fr_txt.readline()
        site_id = fr_txt.readline().split(',')[0]
        fr_txt.seek(0)

        output_file_name = txt_ofolder_path + '\' + site_id + '_cmp_UL.txt'

        with open(output_file_name, 'wb') as fw:
            fw.write('site_id,point_id,x,y,ter_h,gr_veg_h,las_x,las_y,las_h,pt_v
eg_h,delta_h\n')

        fr_Ulas.close()
        fr_Llas.close()
        fr_txt.close()
# read txt, search, and write to txt file
fr_txt.readline()
for line in fr_txt:
    row = line.split(',')
    txt_pt = bc.rtk_veg(str(row[0]),
                        str(row[1]), float(row[2]), float(row[3]), float(row[4]),
                        float(row[5]))
    top_hlist = bc.hlist_inBuffer(txt_pt.x,
                                 txt_pt.y, Upt_list_x, Upt_list_y, Upt_list_h, top_radius)
    if top_hlist:
        top_h = numpy.percentile(top_hlist, 99)
    else:
        top_h = 999
    bottom_hlist =
    bc.hlist_inBuffer(txt_pt.x, txt_pt.y, Lpt_list_x, Lpt_list_y, Lpt_list_h, bottom_radius)
    if bottom_hlist:
        bottom_h = min(bottom_hlist)
    else:
        bottom_h = -999
    pt_veg_h = top_h - bottom_h
    delta_h = pt_veg_h - txt_pt.gr_veg_h
    # print txt_pt.site_id, txt_pt.point_id, txt_pt.x, txt_pt.y, txt_pt.ter_h, '
    # print min_pt_x, min_pt_y, min_pt_h, min_dis, pt_veg_h, delta_h
    # print '-------------------------------
    w_line =
    '%s,%s,%0.3f,%0.3f,%0.3f,%0.3f,%0.3f,%0.3f,%0.3f,%0.3f,%0.3f

    (str(txt_pt.site_id), str(txt_pt.point_id),
    str(txt_pt.x), txt_pt.y, txt_pt.ter_h, 
    txt_pt.gr_veg_h, pt_veg_h, delta_h)
    fw.write(w_line)
    print 'Output file saved in ',
output_file_name
    fw.close()
fr_Llas.close()
fr_Ulas.close()
fr_txt.close()
def CmpVegHtCyl_UAV_UAV(las_ifolder_path, txt_ifolder_path, txt_ofolder_path, top_radius = 0.2, bottom_radius = 1.0):
    if not os.path.exists(las_ifolder_path):
        print "Could not find ", las_ifolder_path, "!"
        if not os.path.exists(txt_ifolder_path):
            print "Could not find ", txt_ifolder_path, "!"
            return
        return
    if not os.path.exists(txt_ofolder_path):
        os.makedirs(txt_ofolder_path)

    las_file_list = glob.glob(os.path.join(las_ifolder_path, '*.las'))
    txt_file_list = glob.glob(os.path.join(txt_ifolder_path, '*.txt'))

    las_file_num = len(las_file_list)
    txt_file_num = len(txt_file_list)
    if not (las_file_num == txt_file_num):
        print "Number of las and txt files does not match!"
        return

    for k in range(0, las_file_num):
        print las_file_num
        with open(txt_file_list[k]) as fr_txt:
            with open(las_file_list[k]) as fr_las:
                # read las
                pt_3d_las = []
                fr_las = file.File(las_file_list[k], mode='r')

                pt_list_x = []
                pt_list_y = []
                pt_list_h = []

                print len(fr_las)
                for f in fr_las:
                    pt_list_x.append(f.x)
                    pt_list_y.append(f.y)
                    pt_list_h.append(f.z)

                print "Reading las file ", las_file_list[k]

                fr_txt.readline()
                site_id = fr_txt.readline().split(','[0]
                fr_txt.seek(0)

                output_file_name = txt_ofolder_path + '"' + site_id + '_cmp_UU.txt'

                with open(output_file_name, 'wb') as fw:
fw.write('site_id,point_id,x,y,ter_h,gr_veg_h,las_x,las_y,las_h,pt_veg_h,delta_h\n')

#read txt, search, and write to txt file
fr_txt.readline()
for line in fr_txt:
    row = line.split(',
    txt_pt = bc.rtk_veg(str(row[0]),
                        str(row[1]), float(row[2]), float(row[3]), float(row[4]),
                        float(row[5]))
    top_hlist = bc.hlist_inBuffer(txt_pt.x,
                                 txt_pt.y, pt_list_x, pt_list_y, pt_list_h, top_radius)
    if top_hlist:
    top_h = numpy.percentile(top_hlist, 99)
    else:
    top_h = 999
    bottom_hlist = bc.hlist_inBuffer(txt_pt.x,
                                 txt_pt.y, pt_list_x, pt_list_y, pt_list_h, bottom_radius)
    if bottom_hlist:
    bottom_h = min(bottom_hlist)
    else:
    bottom_h = -999
    pt_veg_h = top_h - bottom_h
    delta_h = pt_veg_h - txt_pt.gr_veg_h
    # print txt_pt.site_id, txt_pt.point_id, 
    # print min_pt_x, min_pt_y, min_pt_h, 
    # print '-----------------------------------'
    w_line =
    '%s,%s,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f\n' % 
    (str(txt_pt.site_id),
     str(txt_pt.point_id), txt_pt.x, txt_pt.y, txt_pt.ter_h,
     txt_pt.gr_veg_h, 
     txt_pt.x, txt_pt.y, top_h, pt_veg_h, delta_h)
    fw.write(w_line)
    print 'Output file saved in ', output_file_name
fw.close()
fr_las.close()
fr_txt.close()

#---------------------------------------------
import matplotlib.pyplot as plt
def plot_4p(txt_folder_path, output_folder_path):
    if not os.path.exists(txt_folder_path):
        print "Could not find ", txt_folder_path
        return

    if not os.path.exists(output_folder_path):
        os.makedirs(output_folder_path)

    txt_file_list = glob.glob(os.path.join(txt_folder_path, '*\.txt'))

    fL = output_folder_path + '\\Ltran'
    f6 = output_folder_path + '\\6Ctran'
    f7 = output_folder_path + '\\7Ctran'
    f9 = output_folder_path + '\\9Ctran'

    if not os.path.exists(fL):
        os.makedirs(fL)
    if not os.path.exists(f6):
        os.makedirs(f6)
    if not os.path.exists(f7):
        os.makedirs(f7)
    if not os.path.exists(f9):
        os.makedirs(f9)

    for k in range(0, len(txt_file_list)):
        with open(txt_file_list[k], 'rb') as fr:
            fr.readline()
            site_id = fr.readline().split(',')
            print "Processing site", site_id
            fr.seek(0)
            fr.readline()

            L_output_file_path = fL + '\\' + site_id + '.png'
            C6_output_file_path = f6 + '\\' + site_id + '.png'
            C7_output_file_path = f7 + '\\' + site_id + '.png'
            C9_output_file_path = f9 + '\\' + site_id + '.png'

            L_xlist, L_y1list, L_y2list, L_y3list, L_y4list = [], [], [], [], []
            C6_xlist, C6_y1list, C6_y2list, C6_y3list, C6_y4list = [], [], [], [], []
            C7_xlist, C7_y1list, C7_y2list, C7_y3list, C7_y4list = [], [], [], [], []
            C9_xlist, C9_y1list, C9_y2list, C9_y3list, C9_y4list = [], [], [], [], []

            L_dis = C6_dis = C7_dis = C9_dis = 0.0

            for line in fr:
                row = line.split(',,')
                point_id = row[1]
indct = row[1].split('_')[1]
y1 = float(row[4])
y2 = float(row[5])
y3 = float(row[6])
y4 = float(row[7])

if ('L' in indct) and ('.' not in indct):
    # create long transect list
    L_y1list.append(y1)
    L_y2list.append(y2)
    L_y3list.append(y3)
    L_y4list.append(y4)
    L_dis = float(indct.split('L')[1])
    L_xlist.append(L_dis)

if ('.' in indct) and (indct[0] == '6'):
    # create 60-m cross transect list
    C6_dis = float(indct[2:5])
    C6_y1list.append(y1)
    C6_y2list.append(y2)
    C6_y3list.append(y3)
    C6_y4list.append(y4)
    C6_xlist.append(C6_dis)

if ('.' in indct) and (indct[0] == '7'):
    # create 75-m cross transect list
    C7_dis = float(indct[2:5])
    C7_y1list.append(y1)
    C7_y2list.append(y2)
    C7_y3list.append(y3)
    C7_y4list.append(y4)
    C7_xlist.append(C7_dis)

if ('.' in indct) and (indct[0] == '9'):
    # create 90-m cross transect list
    C9_dis = float(indct[2:5])
    C9_y1list.append(y1)
    C9_y2list.append(y2)
    C9_y3list.append(y3)
    C9_y4list.append(y4)
    C9_xlist.append(C9_dis)

L_xlistnew, L_y1listnew, L_y2listnew, L_y3listnew, L_y4listnew = resort4(L_xlist, L_y1list, L_y2list, L_y3list, L_y4list)
C6_xlistnew, C6_y1listnew, C6_y2listnew, C6_y3listnew, C6_y4listnew = resort4(C6_xlist, C6_y1list, C6_y2list, C6_y3list, C6_y4list)
C7_xlistnew, C7_y1listnew, C7_y2listnew, C7_y3listnew, C7_y4listnew = resort4(C7_xlist, C7_y1list, C7_y2list, C7_y3list, C7_y4list)
= resort4(C7_xlist, C7_y1list, C7_y2list, C7_y3list, C7_y4list)
C9_xlistnew, C9_y1listnew, C9_y2listnew, C9_y3listnew, C9_y4listnew \
= resort4(C9_xlist, C9_y1list, C9_y2list, C9_y3list, C9_y4list)

draw_4Lplt(site_id, L_xlistnew, L_y1listnew, L_y2listnew, L_y3listnew, L_y4listnew, L_output_file_path)
draw_4Cplt(site_id, C6_xlistnew, C6_y1listnew, C6_y2listnew, C6_y3listnew, C6_y4listnew, C6_output_file_path)
draw_4Cplt(site_id, C7_xlistnew, C7_y1listnew, C7_y2listnew, C7_y3listnew, C7_y4listnew, C7_output_file_path)
draw_4Cplt(site_id, C9_xlistnew, C9_y1listnew, C9_y2listnew, C9_y3listnew, C9_y4listnew, C9_output_file_path)

def draw_4Lplt(site_id, xlist, y1list, y2list, y3list, y4list, output_file_path):
    if not xlist:
        return
    fig = plt.figure(figsize=(150, 20))
    fig.canvas.set_window_title(site_id)
    plt.ylim(0, 3)
    plt.xlim(0, 150)
    plot_gr_veg, = plt.plot(xlist, y1list, '-bs', linewidth=8.0, markersize=35)
    plot_pt2_veg, = plt.plot(xlist, y2list, '-ro', linewidth=8.0, markersize=35)
    plot_pt3_veg, = plt.plot(xlist, y3list, '-gd', linewidth=8.0, markersize=35)
    plot_pt4_veg, = plt.plot(xlist, y4list, '-y^', linewidth=8.0, markersize=35)
    plt.legend([plot_gr_veg, plot_pt2_veg, plot_pt3_veg, plot_pt4_veg], ['Field', 'UAV_RTK', 'UAV_UAV', 'UAV_LiDAR'], fontsize=100)
    plt.xlabel('distance (m)', fontsize=140)
    plt.ylabel('veg height (m)', fontsize=140)
    xticks_list = range(10, 160, 10)
    plt.xticks(xticks_list, fontsize=140)
    plt.yticks(fontsize=140)
    plt.savefig(output_file_path, bbox_inches='tight')
    plt.close()

def draw_4Cplt(site_id, xlist, y1list, y2list, y3list, y4list, output_file_path):
    if not xlist:
        return
fig = plt.figure(figsize=(35, 25))
fig.canvas.set_window_title(site_id)
plt.ylim(0, 3)
plt.xlim(0, 6)
plot_gr_veg, = plt.plot(xlist, y1list, '-bs', linewidth=8.0, markersize=30)
plot_pt2_veg, = plt.plot(xlist, y2list, '-ro', linewidth=8.0, markersize=30)
plot_pt3_veg, = plt.plot(xlist, y3list, '-gd', linewidth=8.0, markersize=30)
plot_pt4_veg, = plt.plot(xlist, y4list, '-y^', linewidth=8.0, markersize=30)
plt.legend([plot_gr_veg, plot_pt2_veg, plot_pt3_veg, plot_pt4_veg], ["Field", "UAV_RTK", "UAV_UAV", "UAV_LiDAR"], fontsize=100)
plt.xlabel('distance (m)', fontsize=120)
plt.ylabel('veg height (m)', fontsize=120)
xticks_list = range(1, 7, 1)
plt.xticks(xticks_list, fontsize=120)
plt.yticks(fontsize=120)
plt.savefig(output_file_path, bbox_inches='tight')
#    plt.show()
plt.close()

def resort4(xlist, y1list, y2list, y3list, y4list):
    newxlist, newy1list, newy2list, newy3list, newy4list = [], [], [], [], []
    newxindex = numpy.argsort(xlist)
    for i in range(0, len(xlist)):
        newxlist.append(xlist[newxindex[i]])
        newy1list.append(y1list[newxindex[i]])
        newy2list.append(y2list[newxindex[i]])
        newy3list.append(y3list[newxindex[i]])
        newy4list.append(y4list[newxindex[i]])
    return newxlist, newy1list, newy2list, newy3list, newy4list

#------------------------------------------------------------------------------------------
#------------------------------------------------------------------------------------------

def VegCover(smy3_folder_path, output_folder_path):
    if not os.path.exists(smy3_folder_path):
        print 'Cannot find', smy3_folder_path, '!' 
        return
    if not os.path.exists(output_folder_path):
        os.makedirs(output_folder_path)

    smy3_file_list = glob.glob(os.path.join(smy3_folder_path, '*/.txt'))
UR_file = output_folder_path + '//' + 'UAV_RTK.csv'
fur = open(UR_file, 'w')
UL_file = output_folder_path + '//' + 'UAV_LiDAR.csv'
ful = open(UL_file, 'w')
UU_file = output_folder_path + '//' + 'UAV_UAV.csv'
fuu = open(UU_file, 'w')
var_name_list = ['vc', 'vc_00_05', 'vc_05_20', 'vc_20up']
wline = 'site_id,pt' + ',pt'.join(var_name_list) + ',fd' + 
    ',fd'.join(var_name_list) + ',delta_' + ',delta_'.join(var_name_list) + '
'
fur.write(wline)
ful.write(wline)
fuu.write(wline)

for file in smy3_file_list:
    with open(file) as fr:
        fr.readline()
        fdvc_list = [0, 0, 0, 0]
        UR_ptvc_list, UR_delta_list = [0, 0, 0, 0], [0, 0, 0, 0]
        UL_ptvc_list, UL_delta_list = [0, 0, 0, 0], [0, 0, 0, 0]
        UU_ptvc_list, UU_delta_list = [0, 0, 0, 0], [0, 0, 0, 0]
        index = 0
        for line in fr:
            index += 1
            row = line.split(',,)
            site_id = str(row[0])
            fd_veg_h = float(row[4])
            UAV_RTK_vh = float(row[5])
            UAV_UAV_vh = float(row[6])
            UAV_LDA_vh = float(row[7])

            vc_condition(fd_veg_h, fdvc_list)
            vc_condition(UAV_RTK_vh, UR_ptvc_list)
            vc_condition(UAV_LDA_vh, UL_ptvc_list)
            vc_condition(UAV_UAV_vh, UU_ptvc_list)

fr.close()

for i in range(0, 4):
    fdvc_list[i] = '%d' %
        (100*float(fdvc_list[i])/float(index))
    UR_ptvc_list[i] = '%d' %
        (100*float(UR_ptvc_list[i])/float(index))
    UR_delta_list[i] = '%d' % (float(UR_ptvc_list[i]) -
        float(fdvc_list[i]))
    UL_ptvc_list[i] = '%d' % (100 * 
        float(UL_ptvc_list[i]) / float(index))
    UL_delta_list[i] = '%d' % (float(UL_ptvc_list[i]) -
        float(fdvc_list[i]))
    UU_ptvc_list[i] = '%d' % (100 * 
        float(UU_ptvc_list[i]) / float(index))
    UU_delta_list[i] = '%d' % (float(UU_ptvc_list[i]) -
        float(fdvc_list[i]))
ur_line = site_id + ',' + ','.join(UR_ptvc_list) + ',' + ','.join(fdvc_list) + ',' + ','.join(UR_delta_list) + '
ul_line = site_id + ',' + ','.join(UL_ptvc_list) + ',' + ','.join(fdvc_list) + ',' + ','.join(UL_delta_list) + '
uu_line = site_id + ',' + ','.join(UU_ptvc_list) + ',' + ','.join(fdvc_list) + ',' + ','.join(UU_delta_list) + '
fur.write(ur_line)
ful.write(ul_line)
fuu.write(uu_line)
fur.close()
ful.close()
fuu.close()

def vc_condition(veg_h, vc_list):
    if veg_h > 0.02:
        vc_list[0] += 1
    if veg_h > 0.02 and veg_h < 0.5:
        vc_list[1] += 1
    if veg_h >= 0.5 and veg_h <= 2.0:
        vc_list[2] += 1
    if veg_h > 2.0:
        vc_list[3] += 1

#--------------------------------------------------
#--------------------------------------------------
# import numpy as np

def VegMeanHeight(sym3_folder_path, output_file_path):
    if not os.path.exists(sym3_folder_path):
        print "Could not find ", sym3_folder_path
        return

    sym3_file_list = glob.glob(os.path.join(sym3_folder_path, '*.txt'))
    with open(output_file_path, 'w') as fw:
        fw.write('site_id,gr_veg_mean,UAV_RTK_mean,UAV_LDA_mean,UAV_UAV_mean,delta_UAV_RTK,delta_UAV_UAV,delta_UAV_LDA\n')
        for file in sym3_file_list:
            with open(file) as fr:
                fr.readline()
                gr_veg_h, UAV_RTK_vh, UAV_UAV_vh, UAV_LDA_vh = [], [], [], []
                site_id = 0
                for line in fr:
                    row = line.split(',')
                    site_id = row[0]
                    gr_veg_h.append(float(row[4]))
                    UAV_RTK_vh.append(float(row[5]))
                    UAV_UAV_vh.append(float(row[6]))
                    UAV_LDA_vh.append(float(row[7]))
gr_veg_avg = np.average(gr_veg_h)
UAV_RTK_avg = np.average(UAV_RTK_vh)
delta_UAV_RTK = UAV_RTK_avg - gr_veg_avg
UAV_UAV_avg = np.average(UAV_UAV_vh)
delta_UAV_UAV = UAV_UAV_avg - gr_veg_avg
UAV_LDA_avg = np.average(UAV_LDA_vh)
delta_UAV_LDA = UAV_LDA_avg - gr_veg_avg
wline = '%s,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f
' %
(site_id, gr_veg_avg, UAV_RTK_avg, UAV_UAV_avg, UAV_LDA_avg, delta_UAV_RTK, delta_UAV_UAV, delta_UAV_LDA)
fw.write(wline)

delta_UAV_RTK, delta_UAV_UAV, delta_UAV_LDA)
fw.write(wline)

#--------------------------------------------------
APPENDIX C: Dense point clouds screenshots for all study sites

(1) 201

(2) 202

(3) 203
Figure C.1 Dense point clouds screenshots for all study sites
APPENDIX D: Profile Comparison Plots Results for all Study Sites

Figure D.1 legend of profile comparison plots in Figure D.2 and Figure D.3
(10) 239

(11) 246

127
Figure D.2 Profile comparison plots of long transects
Figure D.3 Profile comparison plots of cross transects at 60 meters (left), 75 meters (middle) and 90 meters (right), respectively.