DATA ANALYTICS AND SOFTWARE TO SUPPORT AVALANCHE FORECASTING DECISIONS

by

Peter Kenneth Ottsen

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2021

©COPYRIGHT

by

Peter Kenneth Ottsen

2021

All Rights Reserved

TABLE OF CONTENTS

1. INTRODUCTION 1

2. BACKGROUND 5

2.1 Lidar for Earth System Modeling 5
2.2 Snow and Avalanche Research 6
2.3 Snow and Avalanche Modeling 7
2.4 Snow and Avalanche Modeling with TLS Lidar 8
2.5 Scan Alignment 9

3. ALGORITHMS 11

3.1 Introduction 11
3.2 Snow Depth 11
3.2.1 Scan Comparison 13
3.3 Cliff and Vegetation Identification 16
3.4 Alignment 16

4. SOFTWARE 22

4.1 Overview 22
4.1.1 Python Packages 22
4.1.1.1 Laspy 22
4.1.1.2 VisPy 22
4.1.1.3 PyQt5 23
4.1.1.4 Numpy and SciPy 23
4.1.2 Software Design 23
4.1.2.1 Windows 23
4.1.2.2 Managers and File Object 23
4.1.2.3 Grid 25
4.1.2.4 Scene 25
4.1.3 Overall Windows Layout 25
4.2 Cropping Window 27
4.3 Alignment Window 28
4.4 Snow Depth Window 29

5. DISCUSSION AND EVALUATION 33

5.1 Grid Tuning 33
5.1.1 Cell Size 33
5.1.2 Equivalent Slope Angle Tuning 35

TABLE OF CONTENTS – CONTINUED

5.2 ICP Testing 36
5.2.1 Same Day Scan Comparison 37
5.2.2 Different Day Scan Comparison 38
5.2.3 Intensity Filtering 39
5.2.4 Alignment Without Reflectors 39
5.2.5 Point Selection on Man-Made Object 40
5.2.6 Point Selection on Natural Features 41
5.3 Snow Depth Evaluation and Workflow Testing 43

6. CONCLUSION 53

6.1 Summary and Best Practices 53
6.2 Future Work 54
6.2.1 Alignment 54
6.2.2 Intensity Information Gain 54
6.2.3 Avalanche Modeling 55

REFERENCES 56

LIST OF TABLES

Table Page

5.1 Computation times for creating the grid and adding scan points to it with differing cell size. ... 34

5.2 Error results for aligning December 5 scans of the cliffs on looker's right of the ridge and the lift shack on the looker's left of the ridge at Yellowstone Club. ... 38

5.3 Error results for aligning December 5 and 10 scans of the cliffs on looker's right of the ridge at Yellowstone Club. ... 39

5.4 Percentage error change for aligning December 5 scans of the cliffs on looker's right of the ridge and the lift shack on the looker's left of the ridge at Yellowstone Club with and without the intensity point filter. ... 40

5.5 Table of the error between the ground truth and the snow depth algorithm on January 11, 14, and 22. The error outliers are signified by "**". ... 47

5.6 Algorithm's calculated snow depth compared to actual snow depth. The error from the different sample days had no evidence of being statistically different. (2/2/2021 sample size = 6, 2/4/2021 samples size = 6, 2/9/2021 sample size = 5) ... 48

5.7 Comparison of two methods for calculating snow depth difference between two snow scans at four locations. ... 49

5.8 Minimum ground z-coordinate depth vs. average ground z-coordinate depth. ... 51

LIST OF FIGURES

Figure Page

1.1 Snow depth can be calculated by taking the difference of two co-referenced scans. 2

1.2 Workflow Diagram: Scan, process, then display the results. 4

3.1 Diagram showing snow depth relative to the slope of a mountain side. 12

3.2 Point-to-point snow depth calculation visualization. [7] 13

3.3 Point-to-grid snow depth calculation visualization. [7] 13

3.4 Grid-to-grid snow depth calculation visualization. [7] 13

3.5 Lidar scan of mountain side without grid. 15

3.6 Lidar scan of mountain side with grid visualization. 15

3.7 Tree height vs slope height. The maximum height change within the grid cell (red-dotted lines) is much less for the non-treed slope than treed slope. 17

3.8 Two scans offset by 1 unit. 18

4.1 Diagram showing data flow between components of the software. Red borders signify the Window components. Green borders signify the Managers and File Object components. Orange borders signify the Gridding components. Blue border signifies the Scene components. 24

4.2 Graphical User Interface (GUI) Layout. The orange box in the top left highlights the tabs for navigation between the different windows. The red box highlights the data analysis area of the windows. The green box highlights where the scans will be plotted. The blue box highlights where messages and information will be printed. 26

4.3 Lift pole surrounded by vegetation before cropping. 27

4.4 Lift pole after cropping. The vegetation has been removed. 27

4.5 Match of both lift poles after running the ICP alignment algorithm 29

LIST OF FIGURES – CONTINUED

Figure Page

4.6 Default coloring of scan. If depth has not been calculated the scan will have vegetation and cliffs colored green and the rest of the points colored gray. 30

4.7 Snow depth coloring of scan. Color scale is $blue - white - red$. Blue signifies a decrease in depth, white signifies no change, and red signifies an increase in depth. Green indicates identified cliffs and vegetation. 31

4.8 Intensity coloring of scan. Color scale is $blue - white - red$. Blue signifies the smallest intensity value, white the median value, and red signifies signifies the largest intensity value..... 32

4.9 Red box shows where the summary of the property value currently plotted will be printed for the selected points. 32

5.1 Green indicates identified vegetation or cliffs with a grid of size 5 meters. The circled areas show areas where we are not identifying trees that exist. 34

5.2 Green indicates identified vegetation or cliffs with a grid of size 0.5 meters. The circled areas show areas where we are identifying trees that exist with the 0.5 meter grid cell vs the 5 meter grid cell. 34

5.3 Avalanche probability by slope angle. [26] 35

5.4 Green indicates identified vegetation or cliffs with a 38° equivalent slope angle threshold. 36

5.5 Green indicates identified vegetation or cliffs with a 60° equivalent slope angle threshold. 36

5.6 Match of both lift poles after running the ICP alignment algorithm..... 41

5.7 Snow depth coloring at Bear Canyon before running ICP alignment algorithm. 42

5.8 Snow depth coloring at Bear Canyon after running ICP alignment algorithm. 42

5.9 Photo of the test slope at Bridger Bowl above the Alpine chair lift. 43

LIST OF FIGURES – CONTINUED

Figure Page

5.10 Lidar scan of the test slope at Bridger Bowl. The circled areas show the areas of missing points in the "Ground" scan due to shadowing. 45

5.11 Ground surface vegetation effects on depth average. 49

5.12 Snow surface effects on depth average. 49

5.13 Snow depth error with slope angle. (Sample size 17 measurements) 52

ABSTRACT

Avalanches are a very powerful force of nature and pose significant risk for ski areas and mountainous roads. Avalanche forecasting and mitigation are a very important part of keeping the public safe. Terrestrial laser scanning lidar systems have proven useful in more accurate forecasting and mitigation efforts, but utilizing them can be time consuming. The goal of this project is to operationalize a workflow and create algorithms and ultimately produce a software product that can rapidly analyze snow covered mountainous terrain, allowing avalanche forecasters to make informed decisions on where to focus their mitigation efforts. In this dissertation, I first present algorithms that were designed to align scans, identify trees and cliffs, grid scans, and calculate snow depth. I then introduce a software package that was implemented incorporating these algorithms with a point cloud visualization tool. This software package allows a user to control and visualize the analysis process to make more informed avalanche mitigation decisions. Algorithms were parameterized and validated with a field study consisting of data collection events at Bridger Bowl, Bear Canyon, and the Yellowstone Club in Montana. A Riegl VZ-6000 TLS lidar system was used for all data collection efforts. This dissertation documents the design of this analytics workflow by presenting the algorithms developed, discussing the software implemented, and presenting the data collection efforts that guided the design of the algorithms and served to validate their efficacy.

CHAPTER ONE

INTRODUCTION

Avalanches are a very powerful force of nature and can pose a danger to the public. One of the obvious places avalanches can occur is ski areas. Avalanches can injure or kill patrons and cause damage to buildings. Roads or buildings in mountainous areas can be susceptible to avalanches as well. If not properly mitigated, avalanches can cause road closures and be a danger to vehicles and passengers traveling on them or destroy buildings and property [15]. Avalanche forecasting and mitigation are a very important part of keeping the public and property safe. Forecasters need as much information as possible to make informed decisions. Terrestrial laser scanning (TLS) lidar systems have proven useful for enabling more accurate forecasting and mitigation efforts by allowing remote analysis of snow depth in complex mountainous terrain [8] [7] [24]. TLS lidar systems are portable and can be deployed and utilized in various types of terrain.

Lidar is short for *Light Detection and Ranging*. Pulsed light is used to capture and develop three-dimensional spatial information about an object or area of interest. It works similarly to sonar, but instead of using sound, it sends out pulses of light and then monitors and analyzes how the pulses reflect back. The light pulses are normally in the non-visible spectrum, meaning the human eye cannot see them. Depending on the size and strength of the lidar scanner, areas miles away can be accurately modeled [7]. Lidar was initially used in meteorology to measure cloud density and pollution [13]. Its application became more well known once it was used to map the surface of the moon in the 1971 Apollo 15 mission [28]. Since then, lidar has become the primary tool for terrain mapping. Early in its history, planes would fly over terrain of interest taking lidar readings, and based on the

scans, detailed topographical maps were made with relatively low degree of error [12]. Lidar has come a long way from these initial aerial flights. Modern lidar systems are smaller and can be deployed via a tripod or tension leg system. This portability makes lidar a natural choice for targeted snow evaluation. Lidar creates a three-dimensional point cloud which can then be analyzed for snow depth, snow volume, or other snow and terrain properties. The goal of our project is to operationalize a workflow and create a software product to rapidly analyze snow covered mountainous terrain using lidar so that avalanche forecasters can make informed decisions on where to focus their mitigation efforts.

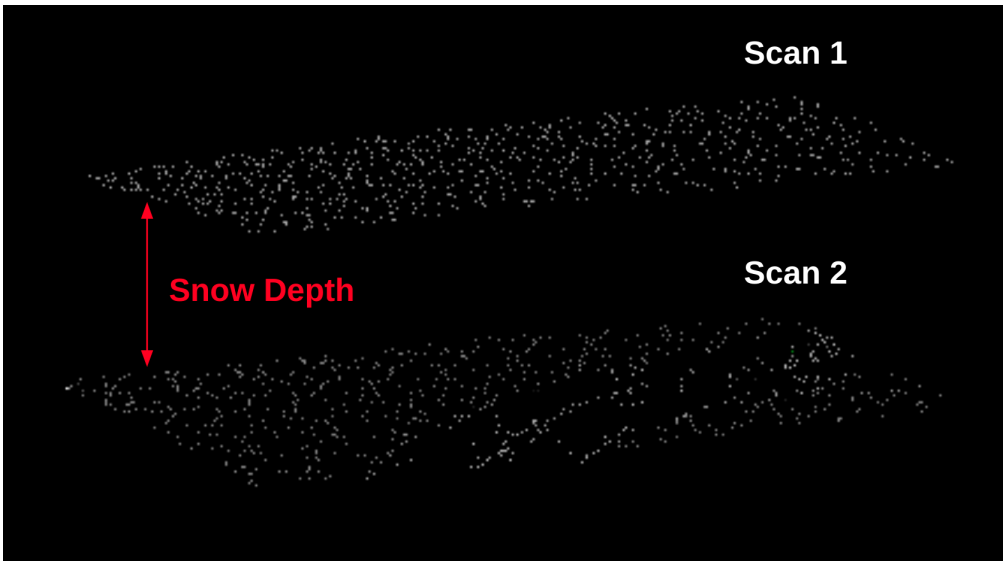


Figure 1.1: Snow depth can be calculated by taking the difference of two co-referenced scans.

The primary metric of interest in our work is snow depth because snow depth is the most direct metric to measure that will give insight into the destructive potential if an avalanche were to occur. Snow depth can be calculated by co-referencing scans and calculating the difference in the depth dimension as shown in Figure 1.1. Before accurately calculating snow depth, several technical challenges needed to be addressed:

1. Point clouds taken during different data collection events need to be aligned in a shared coordinate system so that corresponding points can be identified between the scans for

the depth calculation. If the scans are not accurately aligned, the misalignment will be falsely interpreted as snow depth.

2. Vegetation, cliffs, and man-made structures need to be identified in scans to exclude them from the snow depth calculations. Tiny variations in alignment can cause large inaccuracies in snow depth calculation when non-snow features are present. For example, if a point on a tree branch from one scan is misaligned to correspond to a point on the ground from another scan the calculated snow depth will be much greater than the actual depth.

3. The amount of point cloud data needs to be reduced to enable efficient computations. Terrain lidar scans can result in tens of millions of data points and analysis can be computationally expensive. Simplification techniques needed to be developed to optimize running time.

4. A user-friendly software package must be developed to manage the execution of these analysis algorithms, as avalanche forecasters will not have the expertise to run complex point cloud analysis.

Our workflow is illustrated in Figure 1.2 below.

Our workflow addresses the steps needed to analyze scans and make decisions: cropping (removing unneeded points), aligning scans, identifying vegetation and/or cliffs, calculating snow depth, and inspecting results. The lidar scans were taken at three locations surrounding Bozeman, MT: Yellowstone Club Ski Area, Bear Canyon Ski Area, and Bridger Bowl Ski Area. The Riegl VZ-6000 TLS lidar system was used for all the scans. The scanner was deployed on a tension leg tripod for all the scans.

The rest of this dissertation is organized as follows: Chapter 2 discusses background work. Chapter 3 details the development of our algorithms. Chapter 4 shows the set up and functionality of our software product. Chapter 5 provides insight into the different

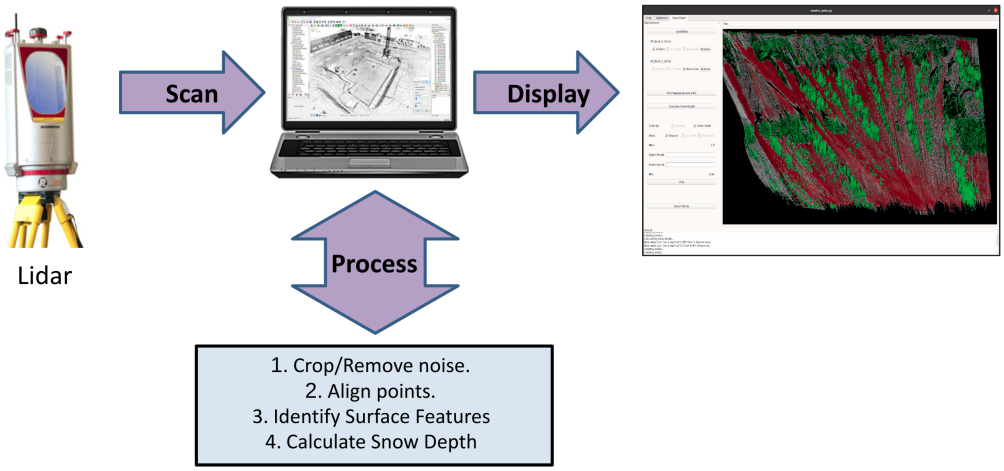


Figure 1.2: Workflow Diagram: Scan, process, then display the results.

parameters of our algorithm and the evaluation of our workflow, and the dissertation is concluded in Chapter 6.

CHAPTER TWO

BACKGROUND

Much research is being done on earth systems, such as meteorology, climate science, and glaciology, to better understand them individually as well as understand how they interact. Though we can not observe all of these systems up close, lidar can allow researchers to model the different systems and draw meaningful conclusions. One application is snow modeling and for us, more specifically, avalanche prediction, monitoring, and mitigation. We are particularly interested in how lidar is being used for mapping and monitoring different earth systems to get a broader view of how they interact.

2.1 Lidar for Earth System Modeling

In this section, we survey the the evolution of lidar and its effect on modeling Earth systems. Aerial lidar scanning revolutionized the ability of researchers to model these systems. With it, researchers were able to scan large areas of land then pull a diverse array of data from these scans and build models. For example, one application of aerial lidar scanning is vegetation analysis. Vegetation analysis is the identification and classification different trees and plants. Lidar has been used to estimate forest biomass and effects of logging in the Amazon rain forest [9] and to inventory different forest vegetation species in Oregon, USA [31]. An application more pertinent to our research is watershed mapping. In mountainous areas, the accumulated snow that feeds an area's water supply is known as a watershed. Many times the inhabitants rely on snow melt to supply the water they need [20]. Determining how much snow has accumulated in alpine areas above these inhabited areas is crucial for planning and preparing for the summer season and for monitoring long-term seasonal trends. Lidar can prove especially useful in this endeavor as it can penetrate

through the vegetation canopy of trees and accurately measure the quantity of snow on the ground which further improves the accuracy of the watershed estimation [20]. Two examples of lidar's successful use in watershed estimation are in the eastern Pyrenees mountians of Europe [1] and the Sierra Nevada Mountains of California, USA [22].

As lidar scanner systems have become more portable, scanners can now be mounted to a tension leg system or easily attached to a permanent mount if consistent scanning of an area will be needed. A tension leg system is a sturdy tripod that is portable and can be set up in different locations for scanning. This portability has further expanded lidar application. The earth processes which formerly could only be modeled by plane can now be modeled on a much smaller scale since the scanners can now be transported and deployed in a wider variety of settings. For example, researchers using a TLS lidar scanning system were able to test and monitor the rate of snow melt on a smaller scale than aerial lidar could to better understand watershed impacts [10]. In Greenland, researchers have permanently installed a lidar scanner system to continuously monitor the glaciers in the southeast part of the country[21]. These applications of the TLS or the permanently mounted lidar system highlight the ultimate goal of our research. We developed a workflow that integrates with a portable or permanent lidar system to monitor snow accumulation, redistribution, and melting to aid in avalanche prediction.

2.2 Snow and Avalanche Research

Every snow season, people are injured or killed in avalanches [15]. It is a very complex problem as there are many factors that influence avalanches. Because of this, much research is directed at understanding snow and mountainous terrain as it relates to understanding avalanches and predicting them. Mountainous terrain is a key factor influencing the probability of an avalanche [18]. Researchers found that slope angle and aspect, which is the compass direction that a slope faces, tie into avalanche probability, and these characteristics

can change rapidly across a small region. The distribution of snow and the changes that the snowpack will undergo throughout a day or over the course of an entire season are key factors in predicting avalanches [25]. The study by Guy and others in 2013, found that weak layers which form over the course of a winter act as trigger points for avalanches [18]. These weak layers form by the interaction of precipitation, wind, temperature, vegetation, and sun radiation that occur over the course of the winter. One possible result of this interaction is buried surface hoar which is a primary cause of avalanches [23]. Surface hoar is large snow crystals that can develop on the snow surface on cold clear nights, and then can get buried then next time it snows creating a weak layer. In addition, the overall depth of the snowpack has a significant effect on stability and destructive potential, but this can be hard to predict and shows great spacial variability across complex mountain terrain [17]. Birkeland and others further investigated the complicated relationship between snow stability, vegetation, and snow depth and found that snow overlaying rock is significantly less stable that snow overlaying vegetation [3]. Schweizer and others took a deeper look at the many different factors affecting avalanche potential such as slope angle, wind, temperature, and precipitation to show that trying to account for all these variables and make avalanche predictions is a very complicated endeavor [26]. The authors also evaluated the spacial variability of different snowpack properties and concluded that the high variability further complicates avalanche prediction [27].

2.3 Snow and Avalanche Modeling

Trying to model and predict avalanches is the next natural step to better understand them. Wide spread studies done in Iran using topographical data and machine learning techniques were able to identify areas of high avalanche probability [30] [6]. All these areas were on steep slopes at higher elevations on the mountains. These findings are something forecasters and practitioners already knew and in their discussion of the results the authors

express the need for better avalanche prediction techniques that can take weather, wind, and snowpack into account. A study conducted in the Rocky Mountains of Canada and the USA tried to better understand snow redistribution, due to wind or other weather, using satellite imagery to develop maps identifying where redistribution was likely to occur [29]. These maps were then verified using aerial lidar scans of the areas of interest.

While these studies in Iran and the Rocky Mountains of Canada and the USA provide insight into larger-scale avalanche trends and snow redistribution, their methods are not suited to providing targeted small scale analysis. With the TLS lidar system, scans can be taken at specific times (daily, before and/or after snow events, etc) and at specific locations and then analyzed to aid in decision making. The most significant work in this field has been done by Jeffrey Deems of the National Snow and Ice Data Center [**?**] [7].

2.4 Snow and Avalanche Modeling with TLS Lidar

In 2013, Deems and others did a study comparing lidar scanning methods and snow depth calculation methods [8]. They compare aerial lidar scanning to TLS lidar scanning and describe the best practices for both methods. They also discuss potential errors that can arise from terrain, vegetation, and post-processing. For snow depth, they discuss three different methods and considerations that come with using each. Lidar scans are commonly viewed as large 3-dimensional point clouds, so the depth can be calculated by comparing a ground scan to a co-aligned snow scan. The three different methods are: (1) point-to-point: where ground scan point elevation values are subtracted from the closest snow elevation point value, (2) point-to-grid: the ground scan is gridded and the grid cell elevation value is subtracted from the overlying snow elevation point value, (3) grid-to-grid: the ground scan and snow scan are gridded and the grid cell elevation value of the ground scan is subtracted from the grid cell elevation value of the snow scan. In our research, we started with these methods to assess their ability to determine snow depth for our application.

In 2013-2014, Deems and others conducted a study to test the ability of a TLS lidar system to integrate with ski area avalanche and control operations at Arapahoe Basin Ski Area in Colorado [7]. They used equipment and a workflow concept very similar to the one we operationalized with our research, by using a Riegl lidar scanner along with the Riegl RiSCAN software to run the scanner. In the summer, they took a scan of the mountain side without snow. In the winter they came back to scan the same mountain side at two different times in the winter. They used manually selected tie-points along with the Multi-Station Adjustment (MSA) tool in RiSCAN. The alignment step is a very important part of the workflow. If the alignment is off, the snow depth calculation will not be reliable as discussed below. In their paper, they calculate snow depth in RiSCAN using their Above Ground Level (AGL) tool. Using this workflow and comparing snow depth changes, they were able to identify areas of snow accumulation and decrease. The remnants of two slab avalanches were clearly identified and wind redistribution of snow can be seen throughout the scans. They were also able to identify areas of avalanche mitigation efforts as ski cuts and craters from explosives were evident in the depth changes. This pilot study led them to conclude that there is potential to use a TLS lidar system to supplement traditional avalanche mitigation techniques and that a natural next step is to use this technique and workflow for highway safety and avalanche control operations. This is where our research becomes relevant. We developed a fit for purpose, user friendly software application that can be used to expediently identify changes in snow depth and make informed avalanche mitigation decisions based on the analysis.

2.5 Scan Alignment

As mentioned above scan alignment is a key component in calculating an accurate snow depth between scans. Since the scans can be treated as 3-D point clouds, iterative closest point (ICP) was the most natural alignment algorithm to start with. ICP works to align the

points of the two scans so that the distance between corresponding points between the two scans, or error, is minimized. This method can be difficult to implement in complex terrain, so many variations of it have been developed. Zhu and others developed an algorithm to align complex terrain using ICP and surface normals [32], which sounded promising but could prove problematic since snow can cause terrain to constantly vary from scan to scan. In 2005, Chetverikov and others developed the 'Trimmed Iterative Closest Point' algorithm to align clouds that may only partially overlap [5]. In 2012, Eo and others were able to apply an ICP variation to buildings scanned from different views for construction and building purposes [11]. In 2013, Gressing and others developed a method of finding features by looking at the points surrounding each point to determine a shape then match those shapes between the scans [16]. Another ICP variation that could prove useful is the 'Sparse Iterative Closest Point' if a tight point spacing is unable to be achieved in the lidar scans [4]. A non-ICP option that could be used would be a point-to-plane registration when a point cloud would be aligned to a surface developed from the other point cloud [14]. Our approach was to first implement the normal ICP algorithm then adjust it as needed to improve alignment accuracy.

CHAPTER THREE

ALGORITHMS

3.1 Introduction

In this section, we present the algorithms developed for our workflow. Lidar scans pose some unique challenges due to their size. High density lidar scans can be tens to hundreds of millions of points. This means analysis can and will have a high computational cost. Our algorithms efficiently handle the data to keep run-time down and ensure the final product is usable.

3.2 Snow Depth

Snow depth is the primary snowpack property we constructed our software to determine. Visualizations of snow depth variations across the scans would help snow safety practitioners to make decisions regarding mitigation measures allowing them to better target their efforts and better determine the extent of measures needed. For this discussion, we will assume all the scans are geo-registered and aligned. We will discuss scan alignment in Section 5.2.

Snow depth is defined as the change in height from the ground to the top of a snow layer measured in the direction perpendicular to the center of the earth, as illustrated by Figure 3.1 below:

Most lidar systems have local leveling, meaning they automatically set the z-axis to be perpendicular to the center of the earth making snow depth a straightforward calculation of the difference in the z-axis. If this were not the case, finding the axis perpendicular to the center of the earth would prove very difficult as all terrain is unique. In comparing scans, there would need to be a transformation to global coordinates so that both scans are in the same reference system and the vertical direction specified. This would require survey-grade

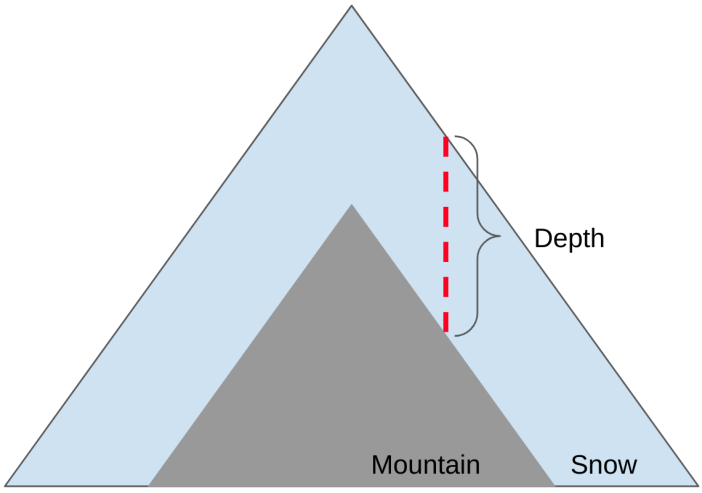


Figure 3.1: Diagram showing snow depth relative to the slope of a mountain side.

GPS units each time which would be time consuming and cumbersome. Our goal was to develop an accurate, easy to implement, and time efficient system and workflow that could be widely used on a daily basis in making mitigation decisions.

Another aspect we accounted for is feature identification and filtering. If the scan being used as the "ground" or "base" scan has vegetation, boulders, or cliffs, the snow depth calculation would be sensitive to this causing it to be noisy and also potentially invalid. Snow depth calculations would be invalid around trees as the branches will interfere with the lidar scan reaching the ground, and thus giving incorrect calculations. Cliffs would also pose a problem if a point at the top of a cliff in the "snow" scan is mapped to a point at the bottom of the cliff in the "base" scan causing the depth calculation to be incorrect. A 30 foot cliff could then show a snow depth of 30 feet or more. This presents an on-going problem since the terrain will constantly change with snow accumulation. Due to these identified error cases, a workflow that could recognize or identify features was needed.

3.2.1 Scan Comparison

Three different methods, point-to-point, point-to-grid, and grid-to-grid for calculating snow depth between scans were put forward by Deems and others [7] and are shown in Figures 3.2 - 3.4. The initial algorithm we tested to calculate snow depth was a point-to-

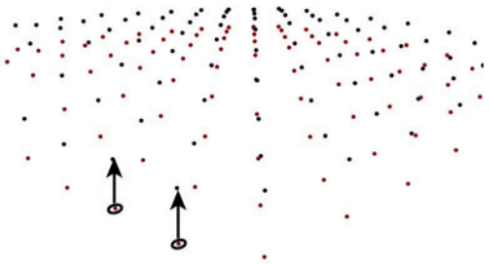


Figure 3.2: Point-to-point snow depth calculation visualization. [7]

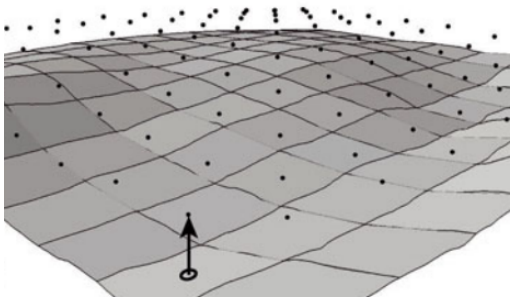


Figure 3.3: Point-to-grid snow depth calculation visualization. [7]

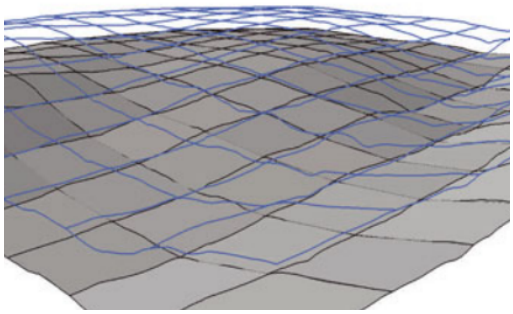


Figure 3.4: Grid-to-grid snow depth calculation visualization. [7]

point comparison between scans. Since the scans are aligned and the z-axis is perpendicular to the center of the earth, the points only needed to be compared in the x-axis and y-axis to find the closest corresponding point in the other scan. Once the closest corresponding point is found, the depth is calculated by subtracting the z-coordinates.

While this method worked, the computational complexity was costly. If each point in the first scan had to be compared to each point in the second scan, this approach has a complexity of: $O(n_1 * n_2)$ where n_1 and n_2 are the number of points in $scan_1$ and $scan_2$

respectively. A kd-tree was then implemented for searching and reduced the run time to: $O(n_1 * log(n_2))$ but this was still a time consuming step as high density lidar scans can be tens to hundreds of millions of points. Based on the computational cost of the point-to-point method, we decided that the point-to-grid method would also be too costly.

This left the grid-to-grid comparison method or a fourth method, building a surface from the scan. A surface could easily calculate the depths as it would be based on a digital elevation model. Partitioning the scans into a grid and doing a grid-to-grid comparison gave a few distinct advantages over building a surface. If a surface is used, the digital elevation model would only reference the positional attributes of the points leaving other attributes, such as intensity or reluctance, not directly accessible. Partitioning the points into a grid allowed attributes to be aggregated on a grid cell level and specific points themselves are still directly accessible if needed.

After we determined that grid-to-grid comparisons would likely be best, the implementation details were determined. One approach was to grid it by x, y, and z. This meant that the grid either needed to follow the slope of the mountain or there needed to be grid cells stacked on top of each other in the z-direction. Upon further consideration, it was determined that gridding in the z-direction added unneeded complexity. By having a 2-dimensional grid in the x and y, the algorithm is able to aggregate data points in the grid cells, and more freely access the points to be used in calculations either individually or averaged across the cell.

Two implementation considerations were: 1) Do we make a grid for each scan? 2) Do we only use the points where the scans overlap? We decided that a single grid would work better due to less computer memory usage and complexity. Each grid cell would then store each scan's points separately within itself. For the second question, we decided to make the grid the size of the extents of combined scans. This means that the maximum distance in

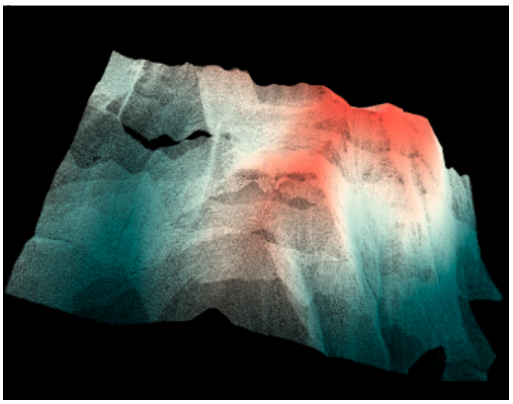


Figure 3.5: Lidar scan of mountain side without grid.

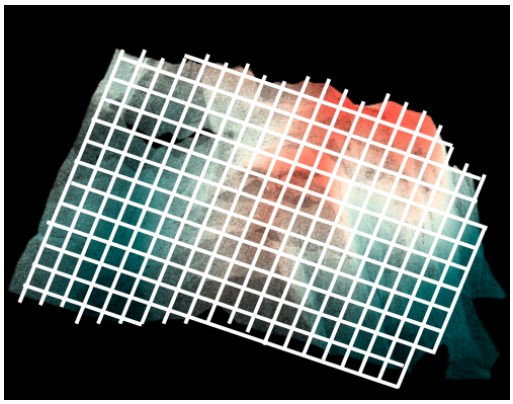


Figure 3.6: Lidar scan of mountain side with grid visualization.

the x-axis is the largest x seen in any of the scans and the minimum distance in the x-axis is the smallest x in any of the scans. The same is done for the y-axis. By defining the grids in this way, no potential data is lost by accidentally cutting points out the grid.

Once our approach to formatting and storing the data points was decided upon, we addressed optimization of the snow depth calculation. Deciding to grid the points allowed us to more efficiently do point-wise comparison between the scans, but would still be computationally expensive.

We determined the most computationally efficient way to compare the points within the grid cells was to aggregate the properties by scan. For snow depth, the calculation is the average height of the difference between the average height of the new scan, $\overline{z_1}$, compared to the old scan, $\overline{z_2}$:

$$\overline{z_1} = \frac{1}{n_1} \sum_{i=1}^{n_1} z_i$$
$$\overline{z_2} = \frac{1}{n_2} \sum_{i=1}^{n_2} z_i$$
$$Depth = \overline{z_1} - \overline{z_2}$$

This is much less computationally expensive as all that is needed is an average calculation followed by a subtraction. Two potential issues that arise with gridding the scans are: 1)

overlooking smaller terrain features in the scan and 2) ensuring depth calculations are not averaged over too large an area. Both of these are controlled through grid cell size. This was addressed by using a small enough grid cell size that a snow depth change is not diluted by averaging over too large an area. Choosing a small enough grid cell size would also ensure smaller terrain features are not overlooked and would have the added benefit of smoothing out micro features that could add noise to the snow depth calculation.

3.3 Cliff and Vegetation Identification

Gridding the scans also provides a more direct way to identify cliffs and trees. In order to get accurate snow depth measurements, trees and cliffs needed to be identified so that they can be eliminated from the calculation. Two approaches were discussed for achieving this: 1) point density and 2) height difference. The point density option worked well for trees. An amorphous cloud on a well spaced point scan would most likely indicate a tree, however, there was not a clear way for adapting this to cliffs. Using the maximum height difference within a grid cell would account for both cliffs and trees. A tree or cliff will have a drastic height change within each grid cell when compared to a normal slope, therefore comparing the maximum height difference for a scan within a grid cell can identified trees and cliffs, Figure 3.7. This is the method that was implemented. The height difference used for identification of trees and cliffs is determined based upon the slope of the mountain and is discussed further in Chapter 5.

3.4 Alignment

Alignment of the scans may be the most important step of the workflow. The snow depth calculation that is the central part of the project hinges on the scans being well aligned. Poor alignment will skew results. Consider, for example, two the scans in Figure

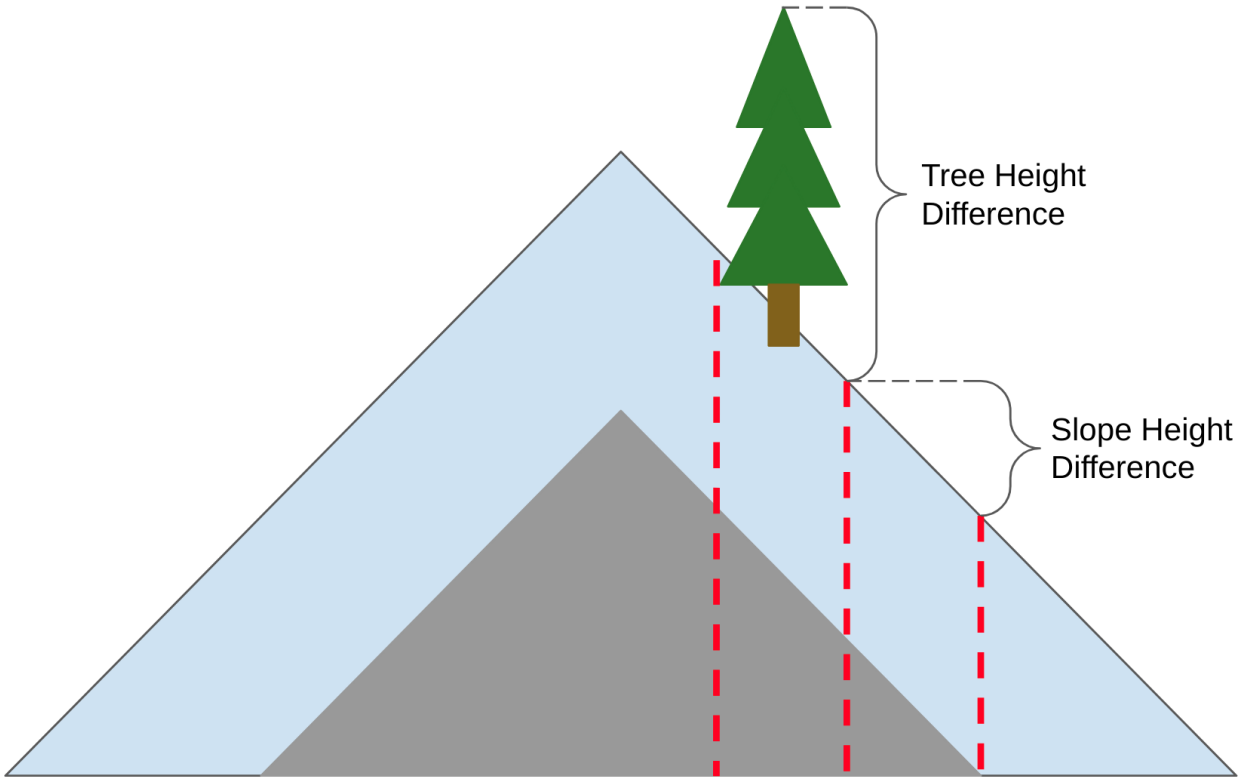


Figure 3.7: Tree height vs slope height. The maximum height change within the grid cell (red-dotted lines) is much less for the non-treed slope than treed slope.

3.8. If the correct alignment is for scan 1 to be vertically offset from scan 2 by 1 unit in the z-direction, and it is rotated or shifted, then what should be a uniform snow depth of 1 unit could range from 0.5 to 1.5 units or worse. Extrapolating this to a slope with snow, snow depths would be inaccurate, misleading forecasters and causing potential problem areas to be overlooked/ignored. The resulting unmitigated avalanche risk increases the risk to the public.

In the example above, the alignment of those planes is simple and straightforward. It becomes much more complicated when looking at the terrain of a mountain side. The features of a mountain side are constantly changing with snow (storm accumulation, wind effects, melting, etc). The lidar scanner can be used to mitigate much of this alignment

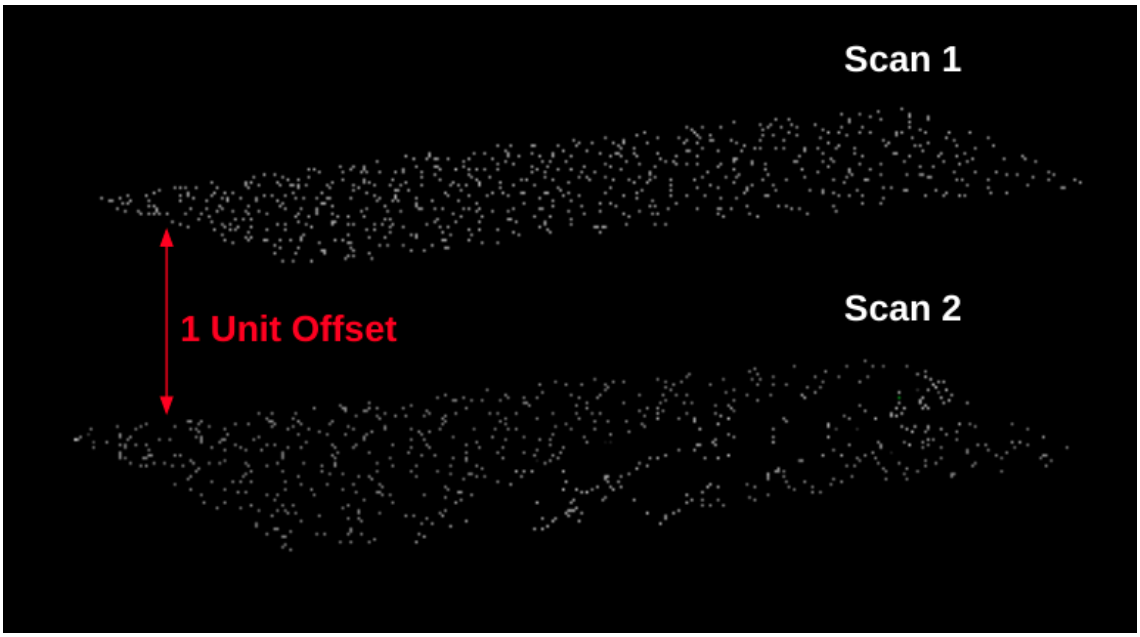


Figure 3.8: Two scans offset by 1 unit.

difficulty through the use of tie-points. Tie-points are permanent or semi-permanent objects installed at the scan sites used to align scans. There must be at least three tie-points at a given location to align scans. Each tie-point will be aligned to itself in other scans of the same area. For our project, we used small (5 centimeter in diameter) reflective disks as tie-points. The potential issues that arise with this are: 1) potential error at distance, 2) the ability to install tie-points in complex, dangerous, or changing terrain, 3) size of the tie-point object needed at distance to accurately align scans.

The potential issues all work together to complicate the problem of alignment. Small tie-points can easily be installed close to the scanner for alignment. The issue that arises with this is the error that can be introduced at distance. If there is a small error in alignment of a scan at 40m distance, this small error will be propagated and grow with distance, potentially causing it to be significant at a distance of 1 kilometer or greater.

This issue can be mitigated by installing tie-points at varying distances from the scanner, but another issue arises. All of the areas of interest are complex mountainous

terrain. This means that there may not be obvious locations to install reflectors. Installing tie-points on a cliff face means that bracketing or other anchoring means would be needed to hold them in place which in most cases may not be realistic. Trees do not make a good long term option as they can grow from season to season and snow accumulating around them may cause slight shifts which can cause alignment errors.

Even if tie-points can be installed farther away, another issue is the size of the tie-point needed. For our reflectors, at 1km the diameter of the reflector that is needed is approximately 1 meter. It can be difficult to find sufficient mount points for reflectors this large.

Due to the potential issues of using tie-points, other potential workflows for alignment were evaluated. To overcome this, we used an Iterative Closest Point (ICP) algorithm to help align scans [2] [19]. ICP algorithms can be used to align point clouds with corresponding features. They work by mapping a point cloud 1 onto point cloud 2. The angle between the mapping and the needed shift to align the scans based on that mapping is calculated to get a best match. This process is repeated until the error between the mapping is below the desired threshold. The ICP algorithm was developed and is detailed below.

The ICP algorithm iteratively calculates the rotation and translation need to align the points of $scan_1$ to the corresponding points in $scan_2$. This calculation is based on the equation:

$$scan_2 = R * scan_1 + t$$

where R is the rotation matrix to rotate $scan_1$ to $scan_2$ and t is the translation needed to move the two scenes together. The outline of the ICP algorithm implementation is:

The second step of the algorithm finds the closest points between $scan_1$ and $scan_2$. This can be a computationally costly step so a kd-tree is used to search the scans and assign the points. As mentioned earlier this has a run time of $O(n_1 * log(n_2))$ which is better than

Algorithm 1: ICP Algorithm Outline

Input: $scan_1$, $scan_2$, $Error_{Threshold}$

1. Initialize registration error, $Error = \infty$
2. For each point in $scan_1$, find the corresponding closest point in $scan_2$
3. Calculate parameters (R, t) from the point correspondence
   $R = RotationMatrix$
   $t = translation$
4. Let $scan_1 = R * scan_1 + t$
5. Update the registration error, $Error$, between the currently aligned $scan_1$ and $scan_2$
6. If $Error$ greater than $Error_{Threshold}$, return to step 2

Return: Aligned $scan_1$

searching point by point, $O(n_1 * n_2)$.

Step 3 calculates the rotation matrix and translation needed to align $scan_1$ to its corresponding points in $scan_2$. To do this we first calculate the centroid of each scene:

$$\mu_i = \frac{1}{\|n_i\|} \sum_{j=1}^{n_i} p_j$$

where μ_i is the centroid of scene i, n_i is points in scene i, and p_i is the coordinates of point j in scan i. We then normalize the points of each scan about its centroid:

$$p'_j = p_j - \mu_i$$

which will cause the point sets to become zero mean. For each scans' normalized points, p'_j, we compute the nine possible products of the two vectors:

$$S_{xx} = \sum_{i=1}^{N_p} p'_{xj} * y'_{xi},\ S_{xy} = \sum_{i=1}^{N_p} p'_{xj} * y'_{yi},\ S_{xz} = \sum_{i=1}^{N_p} p'_{xj} * y'_{zi}$$
$$S_{yx} = \sum_{i=1}^{N_p} p'_{yj} * y'_{xi},\ S_{yy} = \sum_{i=1}^{N_p} p'_{yj} * y'_{yi},\ S_{yz} = \sum_{i=1}^{N_p} p'_{yj} * y'_{zi}$$
$$S_{zx} = \sum_{i=1}^{N_p} p'_{zj} * y'_{xi},\ S_{zy} = \sum_{i=1}^{N_p} p'_{zj} * y'_{yi},\ S_{zz} = \sum_{i=1}^{N_p} p'_{zj} * y'_{zi}$$

and compute the matrix, N:

$$N = \begin{bmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} - S_{zy} & -S_{xz} + S_{zx} & S_{xy} - S_{yz} \\ -S_{zy} + S_{yz} & S_{xx} - S_{zz} - S_{yy} & S_{xy} + S_{yx} & S_{xz} + S_{zx} \\ S_{zx} - S_{xz} & S_{yx} + S_{xy} & S_{yy} - S_{zz} - S_{xx} & S_{yz} + S_{zy} \\ -S_{yx} + S_{xy} & S_{zx} + S_{xz} & S_{zy} + S_{yz} & S_{zz} - S_{yy} - S_{xx} \end{bmatrix}$$

Once N is computed we calculate the eigenvectors and values. The quaternion representing the rotation, R, is the eigenvector corresponding to the largest positive eigenvalue of N. The translation, t, is then calculated by taking the difference between the centroids of $scan_1$ and $scan_2$. After the translation and rotation have been applied $scan_1$ is remapped to $scan_2$ and the error between the points is calculated. If the error is less than the $Error_{Threshold}$, the match is done, else repeat the process.

The ICP algorithm was implemented and first tested on synthetic objects, such as cubes and pyramids. This gave initial confidence in the algorithm, and the next step was to test on real scans. Applying this to complex mountain terrain is difficult as each scan will be slightly different due to changes in features due to snow accumulation. Strategies to account for this will be discussed in the Discussion section of the paper.

CHAPTER FOUR

SOFTWARE

4.1 Overview

The ultimate goal of this project was to build a software product that forecasters can use to make avalanche mitigation decisions. We built a software product that is streamlined for the workflow and user-friendly with the ability for future changes and functionality to be added. In this section we will discuss the Python libraries used, the structure of the software and how all the components interact. We then go into more detail about the layout, functionality, and purpose of each window. The software package can found here: https://github.com/pottsen/LidarViewer.

4.1.1 Python Packages

We used the following pre-built Python libraries to aid in the analysis of the lidar scans. These allowed added functionality without the time consuming step of building from the ground up.

4.1.1.1 Laspy Most lidar scans are in the *.las* file format. The software was built to use this file format. We chose LasPy to make handling *.las* files easier as this package provides a pre-built functionality for file reading, modification, creation, and saving.

4.1.1.2 VisPy We used VisPy for visualizing the 3D point clouds. It can leverage the computational power of modern graphics processing units (GPUs) power through OpenGL. This allowed us to have built in features with support and documentation making it easier to use and implement. Building this from the ground up or trying to use OpenGL directly

would have been time consuming with a steep learning curve. VisPy also gave us the basis for our point selection tool that we use to select data points within our software.

4.1.1.3 PyQt5 PyQt5 is a python library that allowed us to use Qt to build the software API (Application Programming Interface). This is the library used to make the application window and functional buttons to load in the data and run the algorithms.

4.1.1.4 Numpy and SciPy The computational python packages, SciPy and NumPy, are used in our software. They are powerful computational packages that help us to efficiently process and manipulate the lidar data.

4.1.2 Software Design

The software structure can be broken into four different groups of components: Windows, Managers with File Object and Data, Grid, and Scene. The full structure can be seen in Figure 4.1.

4.1.2.1 Windows The Window components are the visual containers for the software and are highlighted by red borders in Figure 4.1. The Master Window is used to hold and display the Cropping Window, Alignment Window, and Depth Window as tabs and initialize the File Manager. The Cropping, Alignment, and Depth windows handle the visual aspects of interacting with the software and pass along the user inputs to the other components.

4.1.2.2 Managers and File Object The Manager components handle and manipulate the scans and are highlighted by green borders in Figure 4.1. The File Manager stores scans (*.las* files) as the Las Data component and acts as the central source for the other managers. The Cropping, Alignment, and Depth Managers access scan data through the File Manager. This allows the file manager to push any changes made in one of the windows/managers to the other managers, and ensures the individual managers are always using the most updated

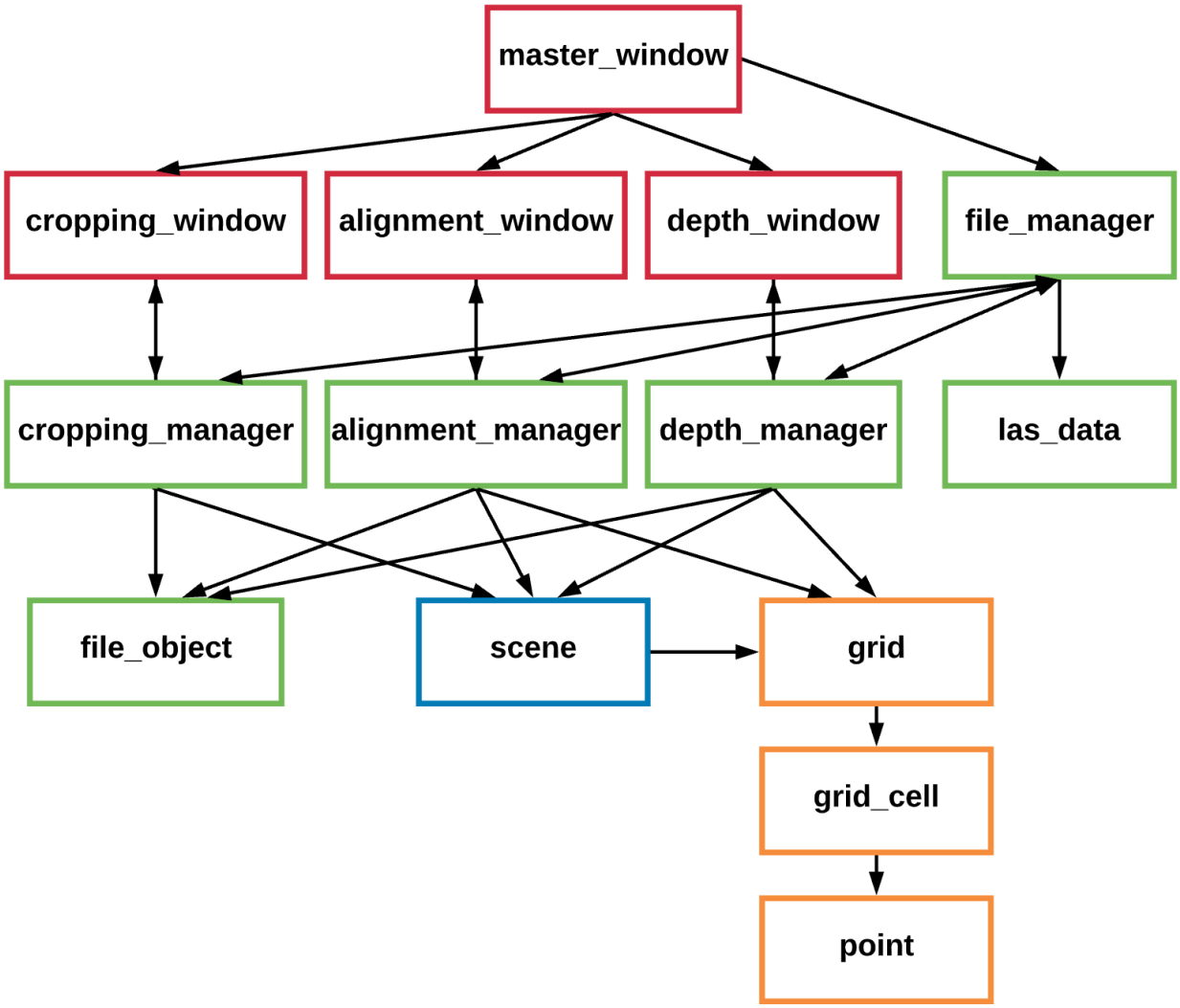


Figure 4.1: Diagram showing data flow between components of the software. Red borders signify the Window components. Green borders signify the Managers and File Object components. Orange borders signify the Gridding components. Blue border signifies the Scene components.

scans. The Cropping, Alignment, and Depth managers all have unique actions that allow them to manipulate the scan data to accomplish their specific functions. The File Object is used to make the unique file options that are displayed in each of the windows.

4.1.2.3 Grid The Gridding components are used to create the grid and grid cells, and store each point and its data from the scans. They are highlighted by orange borders in Figure 4.1. The Grid component contains all the needed functions to create the grid cells, identify trees, calculate depth, pull data, determine point colorings, and more. The Grid Cell component stores all the point data, whether the cell contains a tree or cliff, the snow depth value, and more. The Point component stores all the needed data of individual points, such as xyz-coordinates, intensity value, index location in the scan, and RGB coloring values for display.

4.1.2.4 Scene The Scene component is what displays the scans with the desired point colorings and is highlighted by the blue border in Figure 4.1. It allows the user to interact with and inspect the scans. It has a point selection functionality that allows the user to select specific areas and get data such as depth and intensity for that area.

4.1.3 Overall Windows Layout

Each of the windows has the same overall layout which can be seen in Figure 4.2. The left side, highlighted by the red box, contains the data analysis options for each window. The options will vary depending on which window is currently being used. The bottom of the window, highlighted by the blue box, is the message output. This is where alerts/messages or stats will be printed out for the user. These messages could include warnings or errors if something else needs to be specified for the analysis. The plot area, highlighted by the green box, is where the points will be plotted for viewing and manipulation.

Each window has unique data analysis options based on the purpose of the window. The only button they all have in common is the ‘Load Scan’ button. This button is used to load in the different scan files. A scan loaded in from any window will be available for selection in all the others. The buttons below are unique to each window. They perform the

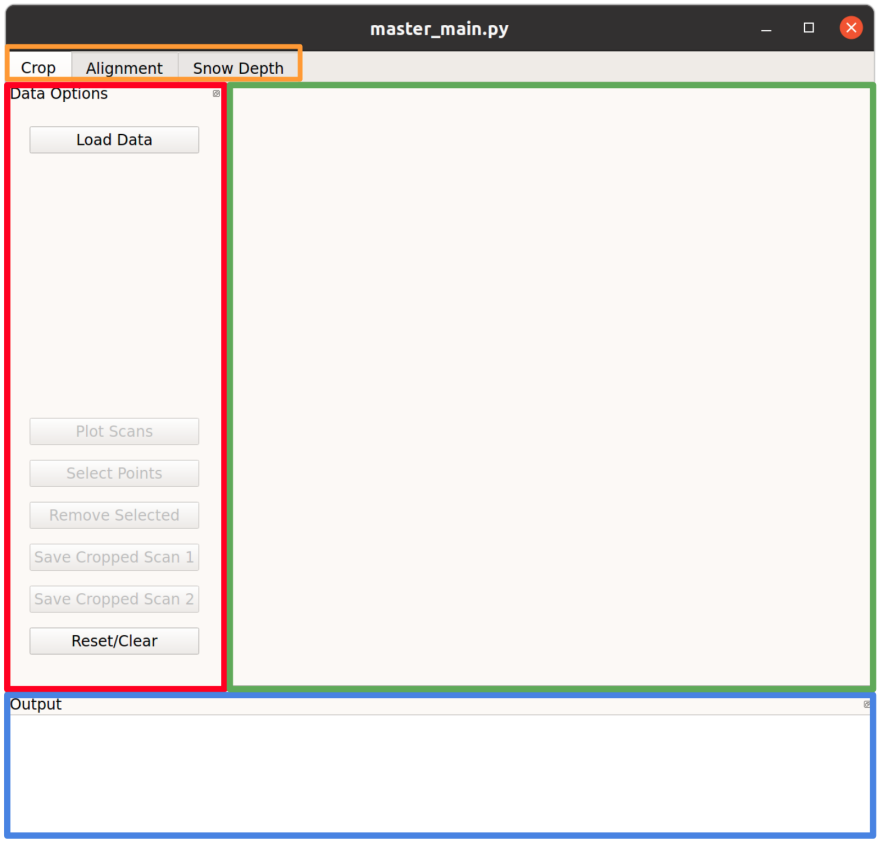


Figure 4.2: Graphical User Interface (GUI) Layout. The orange box in the top left highlights the tabs for navigation between the different windows. The red box highlights the data analysis area of the windows. The green box highlights where the scans will be plotted. The blue box highlights where messages and information will be printed.

specific actions for each window on the lidar scans.

There are tabs at the top left of the application, highlighted by the yellow box in Figure 4.2 that allow the user to switch among the different windows. The three different windows are labeled 'Crop', 'Alignment', and 'Snow Depth'. Each window represents a part of the workflow involved in preparing and adjusting scans to calculate snow depth. The 'Crop' window is used to remove points that are not important to the analysis which clutter the areas of interest. The 'Alignment' window is used to align the scans so that accurate snow depth calculations can be performed. The 'Snow Depth' window is the primary analysis window. It performs the snow depth analysis and can plot the scans based on depth, intensity,

and/or vegetation.

4.2 Cropping Window

The 'Crop' window is used to remove points that are unimportant to aligning the scans or calculating the snow depth. In the scans, there may be a lot of extra area that is not important to the analysis. For example, the scan may contain heavily vegetated areas on the edges of the scan that do not add anything to the analysis. These areas can be removed and an updated scan file saved. Another use is to remove excess points around an object or area that is being used to align two scans. For example, lift towers can be great alignment objects. They are sturdy and unaffected by wind or snowfall, but they can have trees around them which could affect the precision achievable in the point selection process of the 'Alignment' window. Using the 'Crop' window we can remove these tree points enabling precise selection of the lift tower points. Figure 4.3 shows a before shot and Figure 4.4 shows an after shot of a lift tower with trees around it.

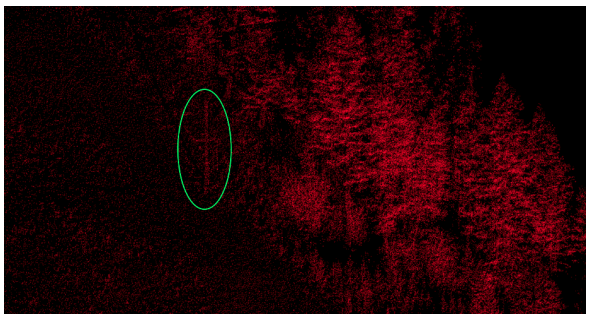


Figure 4.3: Lift pole surrounded by vegetation before cropping.

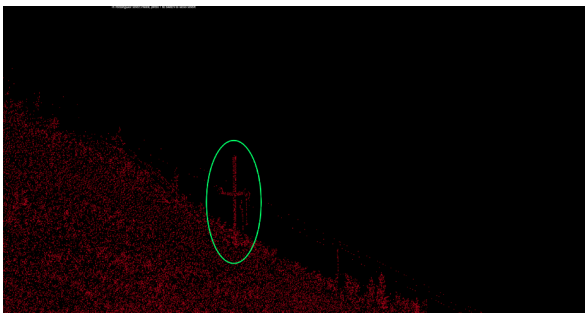


Figure 4.4: Lift pole after cropping. The vegetation has been removed.

The 'Crop' window allows the user to crop two scans at once, 'Crop 1' and 'Crop 2', and swap between the two using the tabs above the plot area. This way the two scans that are to be compared can be simultaneously cropped ensuring correspondence. It does not matter

which scan is designated ‘Crop 1’ or ‘Crop 2’ as this is merely an arbitrary designation for this window. If only one scan needs to be cropped, just the one scan needs to be selected.

4.3 Alignment Window

The ‘Alignment’ window is used to run the ICP algorithm discussed in the algorithms section. The ‘Alignment’ window plots two scans at once. The ‘Base’ scan is the scan that will remain the same, while the ‘Alignment’ scan will be rotated and shifted to find a best match with the base scan. The ICP algorithm can be run on ‘Cliffs/Veg’ or on user selected corresponding points between the two scans. The selection of points is additive. This means that you can keep selecting and adding points until you have selected all the points you would like to match on. The user can easily swap between the two scans using the tabs at the top of the plot area, so corresponding areas can be easily be selected and compared before alignment.

There are two coloring options for the scans. The default option just uses red for one scan and green for the other scan. The user can also choose to color by cliffs and vegetation. If this option is selected, the scan will have gray points with cliffs and trees colored in bright green.

When aligning the scans it is best to use lift towers or other immovable objects that will not change with snowfall. Cliffs can pose issues because each storm will slightly change how a cliff face looks making point correspondence harder. This issue and potential solutions will be discussed in the following discussion section. Figure **??** shows the match improvement from running the ICP algorithm on lift towers and Figure **??** shows a close up view of one of the towers.

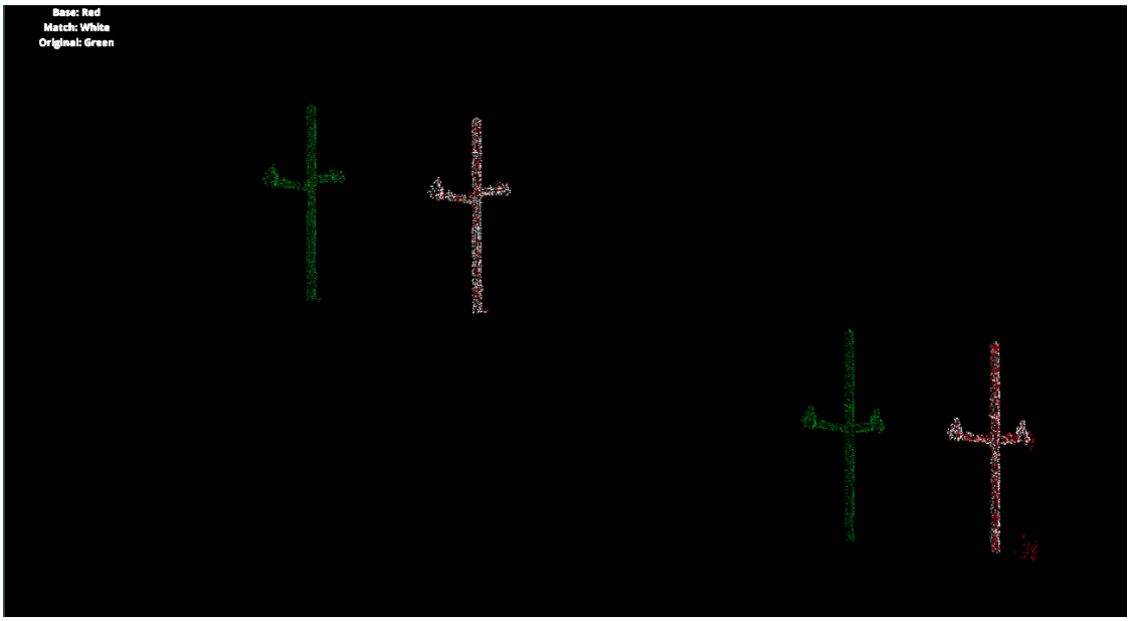


Figure 4.5: Match of both lift poles after running the ICP alignment algorithm.

4.4 Snow Depth Window

The 'Snow Depth' window is for calculating and visualizing snow depth between aligned scans. The loaded scans have three potential designations, 'Ground', 'Int. Snow', or 'New Snow'. 'Ground' should be either a ground scan with no snow or the oldest scan. 'Int. Snow' stands for Intermediate Snow and is intended to be an in-between scan. This scan could be could be used to track a snow event that the forecaster thinks may have formed a weak layer in the snowpack to check how much snow has accumulated on it. 'New Snow' is the most recent scan. The 'Find Vegetation and Cliffs' button grids the selected scans and flags cells where it finds vegetation and cliffs based on the criteria discussed in the algorithm sections. Then the user can plot the scan with the default coloring. Default coloring will color cliffs and vegetation green and all the other points will be gray, Figure 4.6.

The 'Calculate Snow Depth' button will calculate the average snow depth in each grid cell. Once this is done the scan can be colored by snow depth according to the basis chosen.

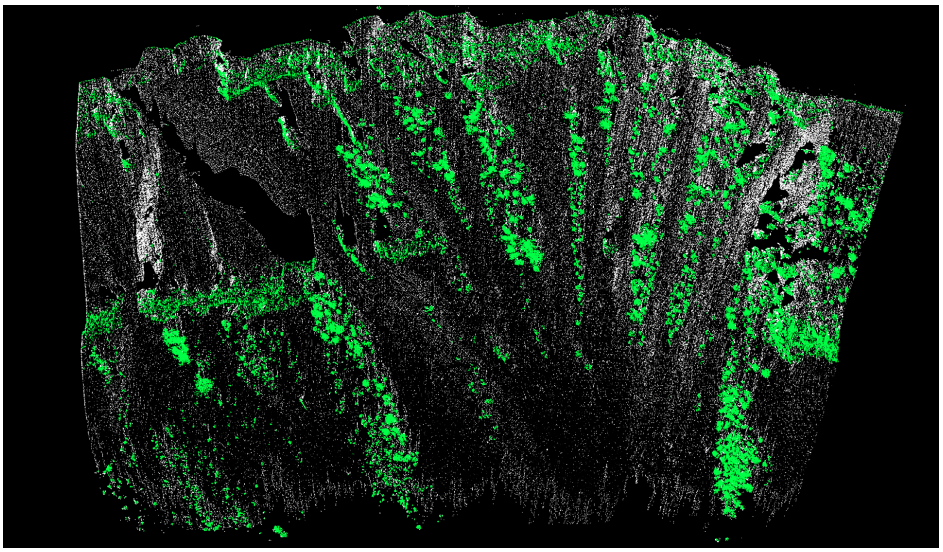


Figure 4.6: Default coloring of scan. If depth has not been calculated the scan will have vegetation and cliffs colored green and the rest of the points colored gray.

'Ground' will be based on the depth between the 'Ground' and 'New Snow' scan, while 'Int. Snow' will be based on the depth between the 'Int. Snow' and 'New Snow' scan. For the coloring, the default color scale with be *white* for a depth of zero, *red* for positive changes with darker *red* indicating greater change, and *blue* for negative changes with darker *blue* indicating greater change. These shadings can be adjusted using the input boxes to indicate the desired 'Upper Bound' and 'Lower Bound' of the shadings. Anything above the 'Upper Bound' will be full *red* and anything below the 'Lower Bound' will be full *blue* with anything in-between fading to white respectively. An example of this can be seen in Figure 5.6.

Intensity can be plotted once the selected scans have been gridded and cells with vegetation and cliffs flagged. The intensity will use the same $blue - white - red$ color scale as snow depth, but the default shading will have the 'Lower Bound' two standard deviations below the median intensity and the 'Upper Bound' two standard deviations above the median intensity. These shadings can be adjusted using the input boxes to indicate the desired 'Upper Bound' and 'Lower Bound' of the shadings similar to snow depth. An example of this can

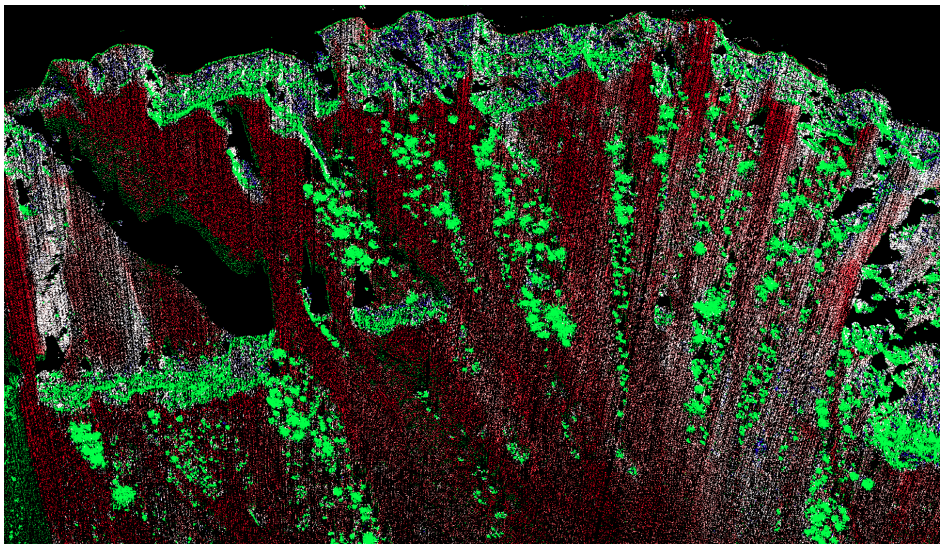


Figure 4.7: Snow depth coloring of scan. Color scale is $blue - white - red$. Blue signifies a decrease in depth, white signifies no change, and red signifies an increase in depth. Green indicates identified cliffs and vegetation.

be seen in Figure 4.8.

In order to do more detailed inspection, the user can select an area of points and a summary of the property value of the point plotted is printed to a text box on the plot window. This is meant to be useful for the user as they will be able to get detailed information on specific areas to aid in the decision making process.

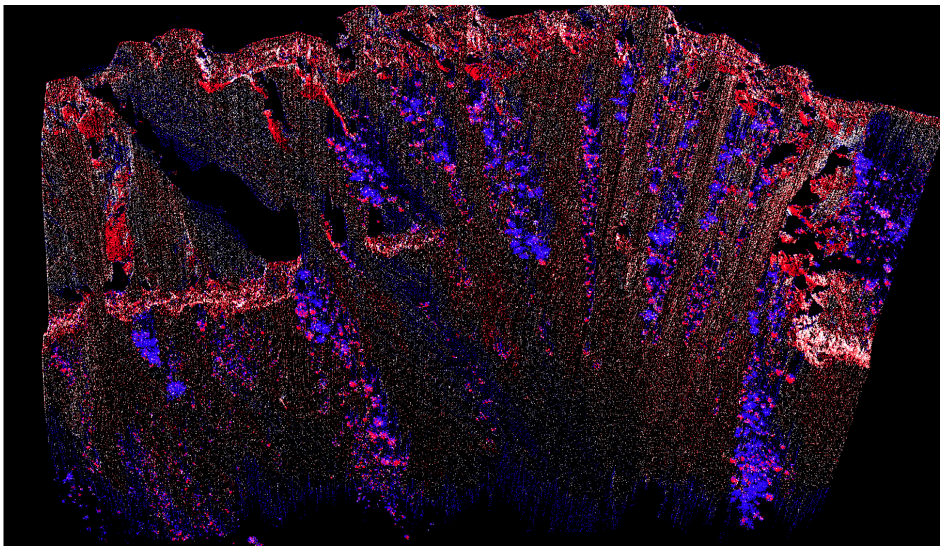


Figure 4.8: Intensity coloring of scan. Color scale is *blue* − *white* − *red*. Blue signifies the smallest intensity value, white the median value, and red signifies signifies the largest intensity value.

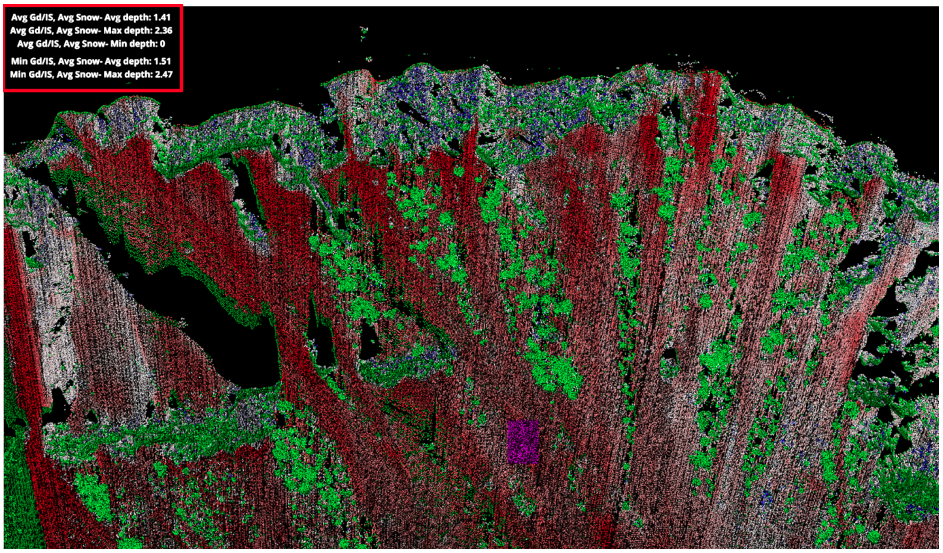


Figure 4.9: Red box shows where the summary of the property value currently plotted will be printed for the selected points.

CHAPTER FIVE

DISCUSSION AND EVALUATION

In this chapter we will discuss different considerations for our research. The goal is to give insight to the thought process that went into the algorithms and software development as well as the challenges faced.

5.1 Grid Tuning

The grid was optimized for cell size and equivalent slope angle for identifying cliffs and trees. Cell size affects computation time and the resolution of the algorithm to catch small features in the terrain. Equivalent slope angle affects the sensitivity of the algorithm to identify trees and/or cliffs. If it is too sensitive, normal slopes will be flagged as trees and/or cliffs thus overlooking important areas that could have avalanche potential. If it's not sensitive enough, trees and/or cliffs will be included in the snow depth calculation leading to inaccurate results.

5.1.1 Cell Size

Cell size refers to the length of the side of a grid cell within the algorithm. Grid cells are all square, so for example, a one meter cell is one meter by one meter. The run times for different cell sizes are different, but not significantly so. A cell size of 0.25 meters has the longest computation time while 2.5 meters has the shortest, Table 5.1.
The complexity of making the grid is given by:

$$O(max(n_1, n_2, |grid|))$$

where $|grid|$ is the number of grid cells, n_1 is the number of points in $scan_1$, and n_2 is the number of points in $scan_2$. The number of grid cells will increase with smaller cell size, and

Cell Size (meters)	**5**	**2**	**1**	**0.5**	**0.25**
Average Computation Time (seconds)	**14.86**	**14.86**	**15.09**	**15.07**	**16.97**

Table 5.1: Computation times for creating the grid and adding scan points to it with differing cell size.

could dominate, but it is not likely to exceed the the maximum number of points in the scans.

We cannot only use computation time to optimize grid size, because visual inspection shows that smaller grid cell sizes can better identify cliffs and trees. Consider the two scans below, the 0.5 meter cell size identifies the trees in the circled areas while the 5 meter cell size misses the trees in the circled areas:

So when choosing cell size we cannot base our decision on computation time alone. We

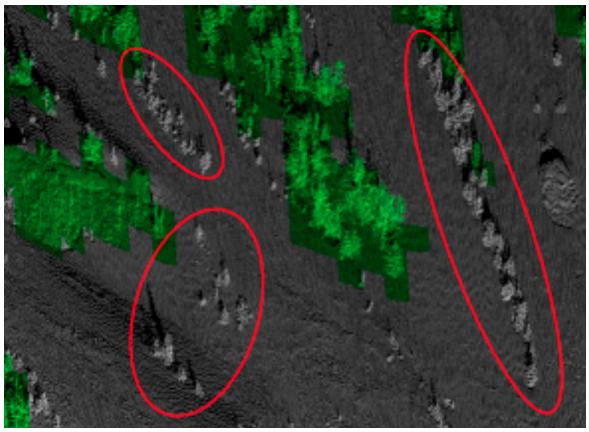


Figure 5.1: Green indicates identified vegetation or cliffs with a grid of size 5 meters. The circled areas show areas where we are not identifying trees that exist.

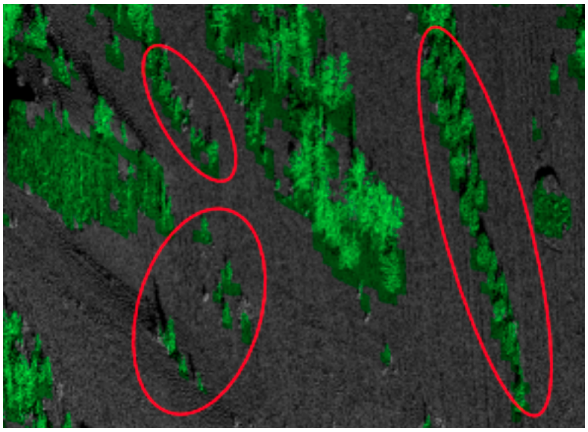


Figure 5.2: Green indicates identified vegetation or cliffs with a grid of size 0.5 meters. The circled areas show areas where we are identifying trees that exist with the 0.5 meter grid cell vs the 5 meter grid cell.

decided to go with a cell size of 0.5 meters, as it shows good identification properties of smaller features while being slightly less computationally expensive when compared to 0.25

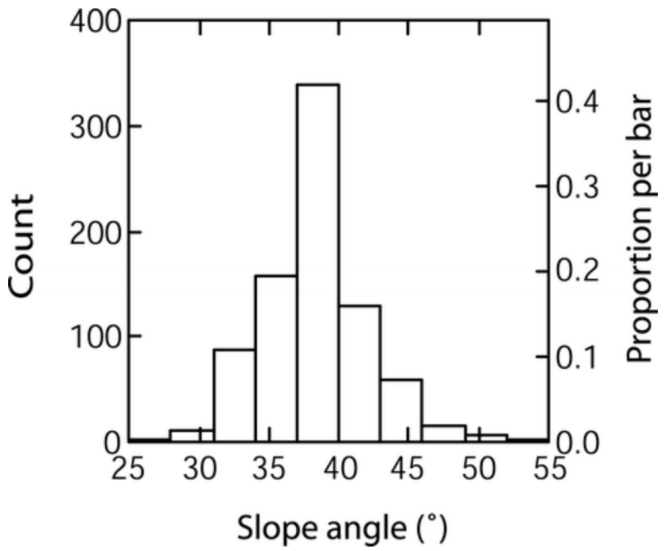


Figure 5.3: Avalanche probability by slope angle. [26]

meters.

5.1.2 Equivalent Slope Angle Tuning

The method for tree and/or cliff identification is dependent upon the height difference chosen for comparison. The key is to choose a height difference that will catch trees and/or cliffs, but not incorrectly flag slopes and vice-versa. To do this, slope angle was a key feature taken into consideration. Slope angles can be grouped into ranges based on their propensity to cause an avalanche

Based on this, we calculate the maximum height difference in a grid cell, h_{max}:

$$h_{max} = z_{max} - z_{min}.$$

where z_{max} is the maximum z-coordinate and z_{min} is the minimum z-coordinate, and then compare it to the height difference of a slope angle equivalent to 60 degrees:

$$h_{equiv} = l_{cell} * tan(60°).$$

where h_{equiv} is the equivalent height difference for a 60° slope, and l_{cell} is the grid cell size. If the maximum height difference, h_{max}, is greater than the equivalent height difference, h_{equiv}, the grid cell is considered a vegetation or a cliff and is flagged by the algorithm.

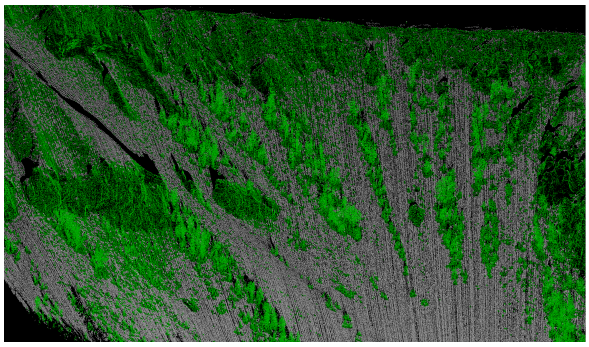


Figure 5.4: Green indicates identified vegetation or cliffs with a 38° equivalent slope angle threshold.

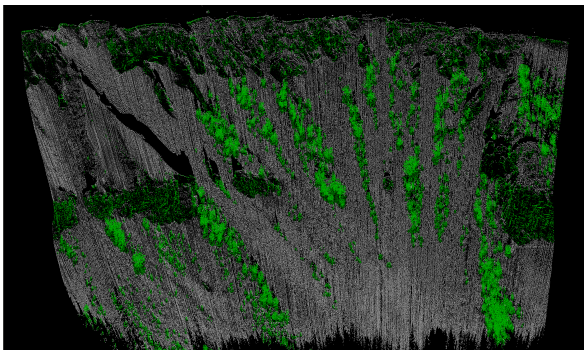


Figure 5.5: Green indicates identified vegetation or cliffs with a 60° equivalent slope angle threshold.

From Figure 5.4 and Figure 5.5, it can be seen that the slope degree has a significant effect on what is flagged as vegetation. In Figure 5.4, a 38° slope is used and much of the areas flagged are open slopes that could slide. While in Figure 5.5, 60° slope is used and only areas of cliffs or trees are flagged.

5.2 ICP Testing

Applying the ICP algorithm to complex mountain terrain can be a difficult task. Snow events will constantly change the landscape. Because of this, the best landmarks to use for the ICP algorithm for a given terrain area would be the cliffs and/or vegetation in the scan as they should change the least between scans. The cliff and vegetation algorithm discussed earlier is easily used to filter the scans down to only vegetation and/or cliffs. Also, the initial guess, or starting alignment, will be very important for the ICP algorithm match. If $scan_1$ and $scan_2$ are not overlapped to begin with the algorithm could be unable to find a match. The solution for this is to use tie-points when taking the scans. Tie-points are permanent

or semi-permanent objects installed at the scan sites used to align scans. There must be at least three tie-points at a given location to align scans. Each tie-point will be aligned to itself in other scans of the same area. This would get the scans as close to alignment as possible, then we could use the ICP algorithm to get the final alignment.

For our project, we used the Riegl VZ-6000 lidar scanner which comes with the Riegl RiSCAN software package. The Riegl RiSCAN software package has the built in ability to align scans into a shared coordinate system using tie-points. We used small reflectors 50 millimeters in size placed around the scan area at a distance of less than 40 meters for our tie-points. The Riegl RiSCAN software can recognize these reflectors then tie each of the reflectors to itself between the scans overtime.

This created a few aspects to be tested: 1)How closely will the tie-points/reflectors align the scans? 2) How well aligned do the scans need to be for the ICP algorithm to work? 3) Will cliffs and trees work for matching? 4) Would more specific point selection be needed to get corresponding areas? 5) Can man-made objects, such as lift towers, be used to match on?

5.2.1 Same Day Scan Comparison

The first field collected scans were from the Yellowstone Club Ski Resort on December 5, 2019. Two scans were taken of the ridge from different locations. The first location was on the ground next to a mid-mountain lift shack and the second location was approximately 200 feet away on the wooden deck of the same lift shack. These scans from the same day should have the least amount of difference in features as they were taken one after the other on a day with no precipitation and give a basis for the the accuracy of our alignment workflow. We used reflectors to align the scans into the same coordinate system and then used the ICP algorithm. Two sets of corresponding areas were tested from the different scan locations. We then compared the error of corresponding points before and after. The reflectors were

	Cliffs on Looker's Right Scans:		Lift Shack on Looker's Left Scans:	
	Avg. Error (cm)	(%) Change:	Avg. Error (cm)	(%) Change:
Initial:	3.29	64%	4.04	65%
Aligned:	1.18		1.41	

Table 5.2: Error results for aligning December 5 scans of the cliffs on looker's right of the ridge and the lift shack on the looker's left of the ridge at Yellowstone Club.

able to align the scans within four centimeters of error. We were able to decrease this error by more than 60% with our ICP alignment algorithm.

These results showed a lot of promise, but there were still other scenarios to test: How would the reflectors and alignment algorithms do on different days (with different snow accumulation)? Could scans be aligned without reflectors?

5.2.2 Different Day Scan Comparison

Testing of the ICP algorithm on scans from different days is the next step in the alignment workflow testing. Scans from different days will have more differences between corresponding features than scans from the same day. We took another scan of the Yellowstone Club ridge from the wooden deck of the lift shack on December 10, 2019 and aligned it with the December 5, 2019 scan taken from the same wooden deck. We used reflectors to align the scans into the same coordinate system and then used the ICP algorithm. We saw that the error was 5.5 cm using just the reflectors and decreased by 13% to 4.9 cm after running the alignment algorithm on the scans.

Based on these results we drew two conclusions: 1) the reflectors seem to do a very good job of getting the scans initially aligned and 2) scans from different days may need more than trees and cliffs to align on.

	Cliffs on Looker's Right Scans - Dec. 5 vs Dec.10:	
	Avg. Error (cm)	(%) Change:
Initial:	5.58	13%
Aligned:	4.86	

Table 5.3: Error results for aligning December 5 and 10 scans of the cliffs on looker's right of the ridge at Yellowstone Club.

5.2.3 Intensity Filtering

Another modification we tested with the ICP algorithm was running the ICP alignment algorithm on cliffs only. We hypothesized that including trees as part of the match could be introducing error since trees can show up in scans as amorphous objects due to the low density of their leaves and branches. For this reason, in aligning two scans, none of the tree points would actually correspond directly to each other. This was not the case for cliffs, since a cliff face is a solid wall. To isolate cliffs only, we filtered the scans based on the intensity value of a scan point. Trees have a low intensity value so the lower returns of the scan were filtered out to leave only the cliffs behind. This modification performed 2-3% worse than including both trees and cliffs, Figure 5.4. An undesired side effect of the intensity filter was that it also filtered out metal objects, such as lift towers and bomb lines, which were very good for matching. Also, the approach of filtering out trees and objects of lower intensity could be problematic for scans that have fewer cliff features leaving minimal points to align the scans.

5.2.4 Alignment Without Reflectors

We also evaluated the ICP algorithm without first running the initial tie-point/reflector alignment. If the reflectors proved unnecessary the process would be very versatile, and any

Table 5.4: Percentage error change for aligning December 5 scans of the cliffs on looker's right of the ridge and the lift shack on the looker's left of the ridge at Yellowstone Club with and without the intensity point filter.

December 5 Scans	Cliffs on Looker's Right Scans : (%) Change:	Lift Shack on Looker's Left Scans: (%) Change:
No Filter:	64%	65%
Intensity filter:	62%	62%

mountainous area could be scanned and then aligned for analysis. To test this, we used scans from Bear Canyon which is an old single lift ski area near Montatna State University. We did not use reflectors in these scans and instead tried to align the scans with just the ICP algorithm. We found the algorithm did not work well by itself without the initial tie-point/reflector alignment.

From these results, we saw the algorithm was highly sensitive to initial alignment. The scans needed to be rather closely aligned to begin with if a good match is to be found. This test also led us to think about other options for improving the ICP alignment. Bear Canyon has lift towers with a shape and appearance unaffected by snow accumulation over time. We concluded that if we could add in a point selection feature for matching, we might be able to get an accurate match without the initial tie-point/reflector alignment by selecting corresponding objects or areas between scans.

5.2.5 Point Selection on Man-Made Object

Being able to select exactly which points to match between the two scans should make it easier to match between scans. This functionality especially makes sense for scans with man-made objects in them such as lift towers, lift shacks, or bomb-lines because we know that these objects will directly correspond. This potentially would also provide a way to match scans that do not have reflectors in them.

We first tested this on the scans from Bear Canyon that did not have reflector alignment. These scans contained lift towers that are prominent in the scans and an obvious match point. These scans showed a 98% decrease in their alignment error. Figure **??** and Figure **??**, show how the match of the lift towers improved. Figure 5.7 and Figure 5.8 show the improvement of the snow depth map. In the before screenshot, Figure 5.7, the blue shading indicated a decrease in snow depth which is not possible given that one scan is bare ground in the summer and the other is from the winter when snow covered the slope. The consistent red color in the after picture, Figure 5.8 indicates a positive snow depth and corresponds to the snow that was covering the ground in the winter.

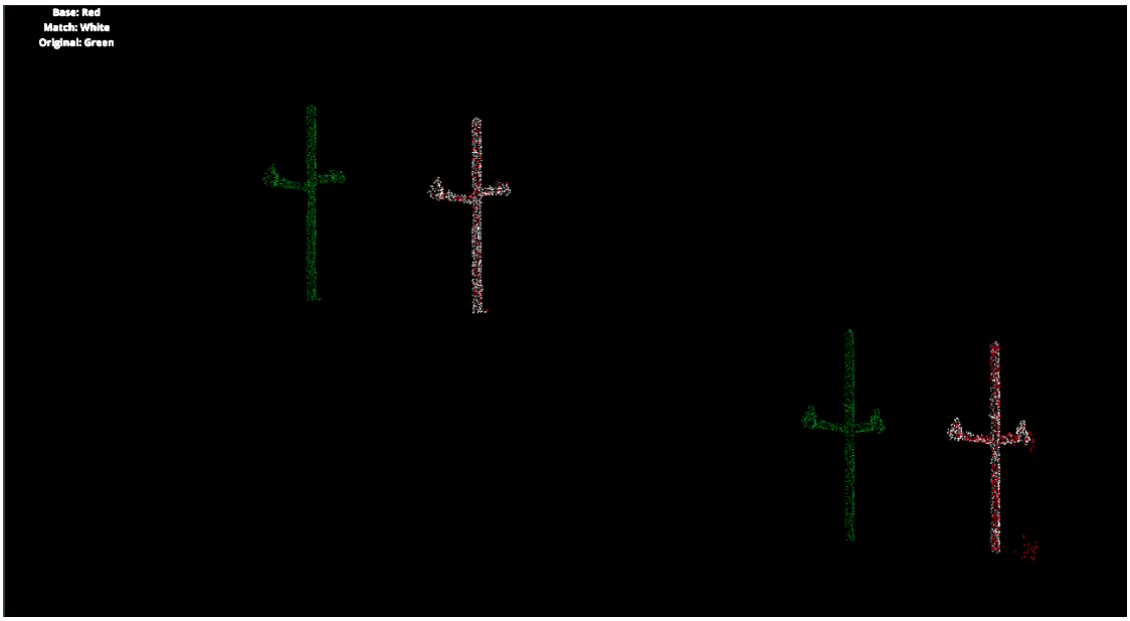


Figure 5.6: Match of both lift poles after running the ICP alignment algorithm.

5.2.6 Point Selection on Natural Features

Point selection can also provide a way to select corresponding cliff bands to match giving better alignment overall. This was tested on scans from the Yellowstone Club taken on May 28, 2020 and June 30, 2020. However, using cliffs to align scans in this way increased

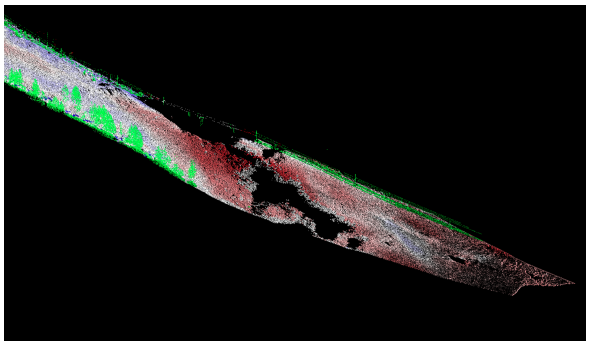


Figure 5.7: Snow depth coloring at Bear Canyon before running ICP alignment algorithm.

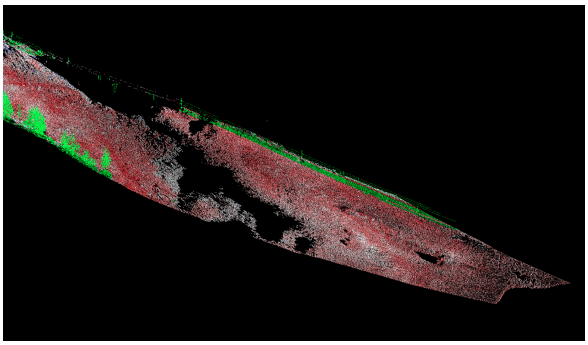


Figure 5.8: Snow depth coloring at Bear Canyon after running ICP alignment algorithm.

the error of the match by 165%.

The error increase is most likely due to the cliffs being complex terrain and differing slightly from scan to scan due to changes in snow coverage or scan angle. This has implications regarding when user specific point selection is appropriate. When using point selection, the user will most likely be aligning with fewer points. Because there are fewer points, the corresponding areas need to be very distinct as any small differences will be amplified by the ICP algorithm. This would explain why cliffs which can slightly change between scans would cause worse alignment than a lift tower or other man-made object which would not change between scans.

Summarizing these findings, we found that point selection for the ICP algorithm should be used on man-made objects if needed to help alignment between scans. Point selection should not be used on cliffs or trees unless a significant area is being matched so the small changes that occur between scans will be spread across the entire scan area.

The best practice for alignment is to use tie-points/reflectors to align scans taken at different times. If this initial alignment is not accurate enough, the ICP algorithm can be used and best results are obtained by using point selection on man-made objects.

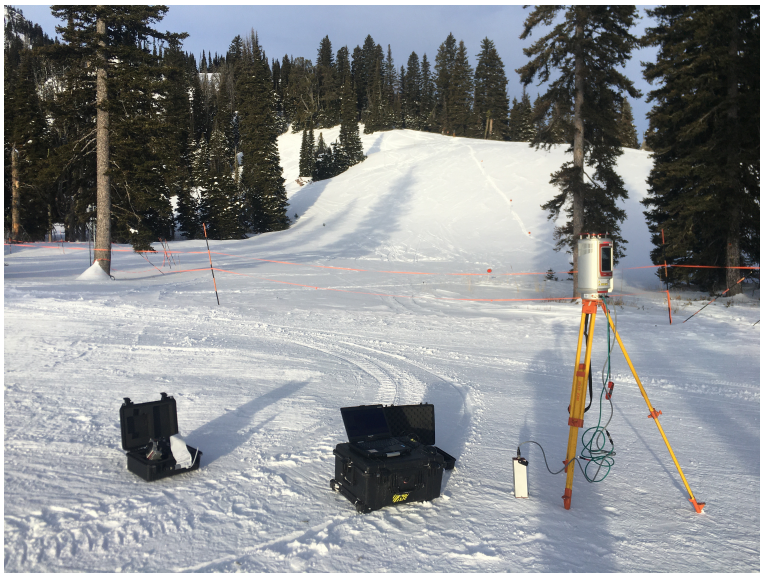


Figure 5.9: Photo of the test slope at Bridger Bowl above the Alpine chair lift.

5.3 Snow Depth Evaluation and Workflow Testing

During the winter 2019/2020 season, we took scans of the ridge at Yellowstone Club. This allowed us to test our alignment workflow and ICP algorithm as the ridge contains complex mountain terrain which we discussed above. We were unable to verify the snow depth algorithm at this site. We tried using survey grade GPS units to mark spots where we used probes to get actual snow depth data, but we were unable to reference the specific GPS locations on the ridge to the scan itself to compare the actual snow depths to algorithm predicted snow depths.

For the winter 2020/2021 season, we tested our workflow at Bridger Bowl Ski Area. The test location was a slope above their Alpine Lift, Figure 5.9. This area provided easy access allowing us to get multiple scans a week and Bridger Bowl let us store the scanner on site. This gave us the opportunity to verify our snow depth algorithm and the opportunity to run our full workflow in the expected operational environment. The Bridger Bowl test site was visited five times during the month of January 2021 and three times during the month

of February 2021.

The test slope at Bridger Bowl is immediately adjacent to the top of the lift. This proximity allowed us to develop a new strategy for verifying the snow depth algorithm. We were able to set out markers each time prior to scanning that would be visible in the scan. Once the scan was done, we would probe the snow directly in front of the marker to get a snow depth measurement. We would then navigate to the markers in the scan and use the snow depth selection tool to select the snow depth values directly in front of the marker and compare to the probed values to the algorithm calculated values.

In August of 2020, we obtained a summer/ground scan of the test slope. We ran the Riegl VZ-6000 scanner with the Riegl's RiSCAN software and aligned the summer scan to the snow scan using reflectors as tie-points with RiSCAN for the initial alignment of the scans. We found that the initial alignment was very good and no further alignment from the ICP algorithm was needed.

Analyzing the first January scans, the estimated algorithm depths were not lining up with the ground truth probing at the markers. Many times the depths of the selected points around the markers were reading as zero in areas where there should have been no zero depths. Initially we suspected the cause to be minor vegetation, tall grass, making the surface seem higher than actual. Following this reasoning, we inspected the ground scan and found that, while there was minor vegetation, it was not tall enough to cause the areas of zero depth we were seeing. We did notice that the ground scan had areas of lower point density and shadowing. Shadowing occurs if there is an object that was prominent enough to block the light from the lidar scanner from reaching the area directly behind it. We found that rocks, boulders, grass and small vegetation were causing shadowing in our ground scan. Figure 5.10 shows this phenomenon. This point density and shadowing effect will cause many cells to not contain points from the ground scan and snow scan to calculate a snow depth from. The cells missing points can be seen in Figure **??**. The light blue areas are

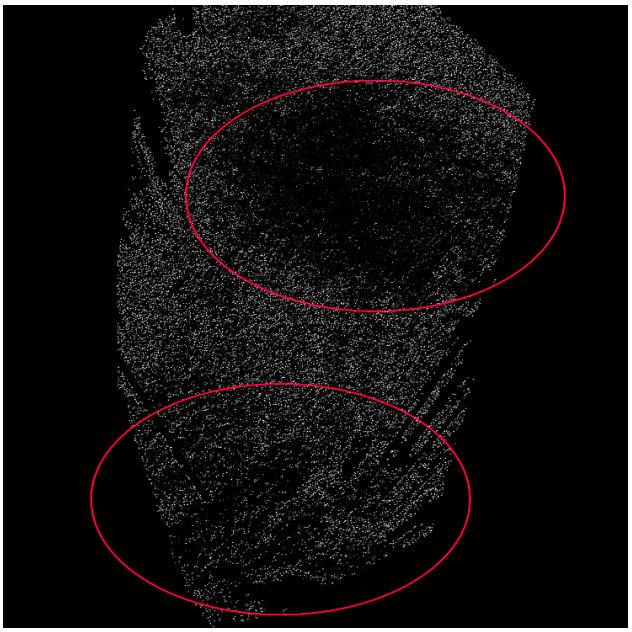


Figure 5.10: Lidar scan of the test slope at Bridger Bowl. The circled areas show the areas of missing points in the "Ground" scan due to shadowing.

cells that are missing points from one or more of the scans and therefore cannot calculate an accurate snow depth.

This led us to suspect that the algorithm was incorrectly accounting for the cells' missing points from one or more of the scans. Upon investigation, we determined that the default value for snow depth in a cell is zero, and if the cell had no points from one or more of the scans, this default depth was never updated and was incorrectly factored into the average depth calculation. A check was added to the snow depth algorithm to make sure that points from both scans are present within the grid cell before calculating the snow depth for that grid cell, and cells with no points are not included in the depth reporting. With the check added, the error of the algorithm prediction from the probed depth decreased significantly. In the first five visits to test site, the average error was -16.5 centimeters. (Note: The negative error implies that the algorithm was underestimating the depth by that much)

We then needed to answer a three questions regarding the error: 1) Is the error

consistent in offset or a percentage of snow depth and can we explain why? 2) Is there a better method to calculate snow depth within the cell besides simply averaging the points from each scan in each cell? 3) Does slope angle affect the error?

As we investigated the error to determine if it was consistent in offset or a percentage of snow depth, we noticed that there were some outliers. The Table 5.5 shows examples of this. On January 11th and January 22nd, there are multiple outliers of error -19 centimeters up to -38.2 centimeters. While on January 14th, the error is much more consistent and no more than -14.2 centimeters. This is a significant swing. Between these different scan days there was minimal snow accumulation, so it would be expected for the error to remain fairly consistent between these scans. It was hypothesized that when probing to get the actual snow depths the probe was penetrating into the ground if too much force was being used. For the final three February scans, less force was used when taking the probe depth and a pit was dug in the snow to the ground and the depth measured to verify the probed depths at each marker. The probe depths aligned closely with the pit depths giving confidence in our measurements and consistent accuracy.

Over the final three scans, the test site at Bridger Bowl accumulated over 100 centimeters of new snow. This provided us the opportunity to answer question 1 above. Table 5.6 gives the 'Actual Depth' on the days, and the algorithm calculated average depth, 'Cell's Average Calculated Depth' and the algorithms calculated maximum depth, 'Cell's Maximum Calculated Depth'. The error of the algorithms on each of these days is in parenthesis. Based on the 'Actual Depth' (this is the average depth found by digging at each marker on the specified date), snow had accumulated between each scan, but the average error of the 'Cell's Average Calculated Depth' is consistent around, -12.0 centimeters. This suggests that the error is consistent in offset or 'absolute' error regardless of snow accumulation. The consistent offset is most likely due to vegetation effects in the ground scan.

This explanation is reinforced by the 'Cell's Maximum Calculated Depth' (the maximum

Table 5.5: Table of the error between the ground truth and the snow depth algorithm on January 11, 14, and 22. The error outliers are signified by "**".

Date	**Avg Cell Error (cm)**
1/11/2021	-8.2
	-35**
	8
	-6.4
	-38.2**
1/14/2021	-0.8
	-14
	-2
	-14.2
	-11.8
1/22/2021	-6
	-15.8
	0.2
	-28.4**
	-19**
	-14.8

calculated cell depth in the selected area) underestimating the snow depth with an average error of -4.3 centimeters. If the maximum depth observed in the cells of the area where the ground truth was taken was equal to or greater than the ground truth depth, this would suggest the error was due to terrain fluctuations. Since the 'Cell's Maximum Calculated Depth' was consistently less than the ground truth, we were most likely not fully accounting for vegetation in the algorithm.

Table 5.6: Algorithm's calculated snow depth compared to actual snow depth. The error from the different sample days had no evidence of being statistically different. (2/2/2021 sample size = 6, 2/4/2021 samples size = 6, 2/9/2021 sample size = 5)

	Actual Depth (cm)	Cell's Average Calculated Depth (cm)	Cell's Maximum Calculated Depth (cm)
2/2/2021	79	67.4 (-11.6)	71.8 (-7.2)
2/4/2021	103	91.4 (-11.6)	100.4 (-2.6)
2/9/2021	187	174.3 (-12.7)	183.9 (-3.1)
Averages	123	111.0 (-12.0)	118.7 (-4.3)

Figures 5.11 and 5.12 below show how the error is most likely introduced from vegetation, grass or small shrubbery. The snow depth algorithm averages the z-coordinates of the points from each scan within a cell, and it is likely that this average is artificially raising the ground surface when taking the average z-coordinate value within the cell to calculate the depth. Figure 5.11 shows how this can affect the depth calculation, as true ground will be at the bottom while the averaged ground will be somewhere above it due to the vegetation. Figure 5.12 shows the snow surface. The snow surface is almost always smooth and consistent unless there is an avalanche debris pile or ski tracks.

We further investigated this to determine if the error is due to vegetation in the 'Ground' scan. We calculated the snow depth to the 'Ground' scan for the February 2nd scan and

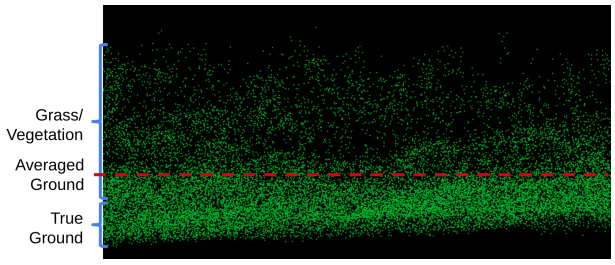


Figure 5.11: Ground surface vegetation effects on depth average.

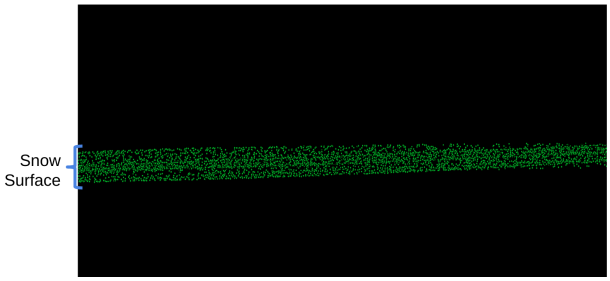


Figure 5.12: Snow surface effects on depth average.

Table 5.7: Comparison of two methods for calculating snow depth difference between two snow scans at four locations.

	Location			
	1	2	3	4
Difference of Total Depth (cm)	78	93	98	99
Direct Calculation (cm)	79	91	98	99

the February 4th scan, and then subtracted the February 2nd scan depth from the February 4th scan depth. This is given by 'Difference of Total Depth' in Table 5.7. If the error is from vegetation in the 'Ground' scan, taking the difference in depth will cause the error to cancel out. We then used the algorithm to directly calculate the snow depth between the February 4th scan and the February 2nd scan. This is given by 'Direct Calculation' in Table 5.7. We compared these depths at 4 corresponding locations. If the 'Difference of Total Depth' and 'Direct Calculation' are similar, then the error in the depth calculation will be from the vegetation in the 'Ground' Scan. The Table 5.7 below shows that the 'Difference of Total Depth' is almost exactly the same as the 'Direct Calculation'. This further supports the theory that vegetation in the ground scan is causing the error between the snow depth algorithm and the ground truthing. It also shows that if the scans being compared both have a relatively smooth surface, the algorithm gives an accurate result.

We now answer the question: Is there a better method to calculate snow depth within the cell besides averaging the points from each scan? The results in the Table 5.7 indicate that, given smooth surfaces, the algorithm is accurate, but there could be a better way to account for depth if there is vegetation in the 'Ground' scan. One possible way to better account for vegetation is to use the minimum z-coordinate value in each cell from the ground scan. We tested this by calculating snow depths using the minimum z-coordinate value in the ground scan, and then calculating the snow depth using the average snow scan z-coordinate and the maximum snow scan z-coordinate. Table 5.8 below shows the results of this analysis.

The comparisons above show that for a 'Ground' scan with vegetation, taking the minimum z-coordinate value can reduce the error. Based on this, the best practice found was to use the depth calculated by the minimum ground z-coordinate if the 'Ground' scan has small vegetation in it and to use the depth calculated by the average ground z-coordinate if the 'Ground' scan contains a smooth surface.

We also investigated whether other factors, such as slope angle, can influence the error as well. Figure 5.13 shows how slope angle can affect the error. At higher slope angles the error decreases significantly. The errors were binned and averaged in intervals of 10 degrees such that, 0-10 degrees, 11-20 degrees, and 21-30 degrees were used. The 21-30 degree interval has one-third to one-fourth the error of the 0-10 degree and 11-20 degree intervals respectively. This analysis is encouraging as most slopes of interest will be 20 degrees or greater meaning the error due to slope angle should be relatively low.

The slope angle analysis was only done for a 'Ground' scan with vegetation. We did not have actual slope angle measurements for the analysis and the comparison done in Table 5.7, but the slope angle in the at least one of areas analyzed is from each of the bin intervals. Because there is little to no error between the two scans with a smooth snow surface, it suggests that error effects due to slope angle would not be significant if smooth surfaces are used in the algorithm.

Table 5.8: Minimum ground z-coordinate depth vs. average ground z-coordinate depth.

Date	Slope Angle	Avg G, Avg S - Avg (cm)	Min G, Avg S - Avg (cm)
2/2/2021	25	-5.8	-0.8
	27	-4.2	-0.2
	17	-15.8	-14.8
	6	-15.2	-17.2
	6	-12.8	-10.8
	3	-15.6	-10.6
2/4/2021	22	-5.4	1.6
	26	-10.8	-5.8
	24	-14.6	-16.6
	14	-15.6	-13.6
	7	-9.4	-7.4
	5	-13.6	-13.6
2/9/2021	26	-5.6	0.4
	23	-8.2	-4.2
	13	-20.8	-16.8
	13	-14.8	-13.8
	5	-14.2	-13.2
Average Error (cm):		**-11.90588235**	**-9.258823529**

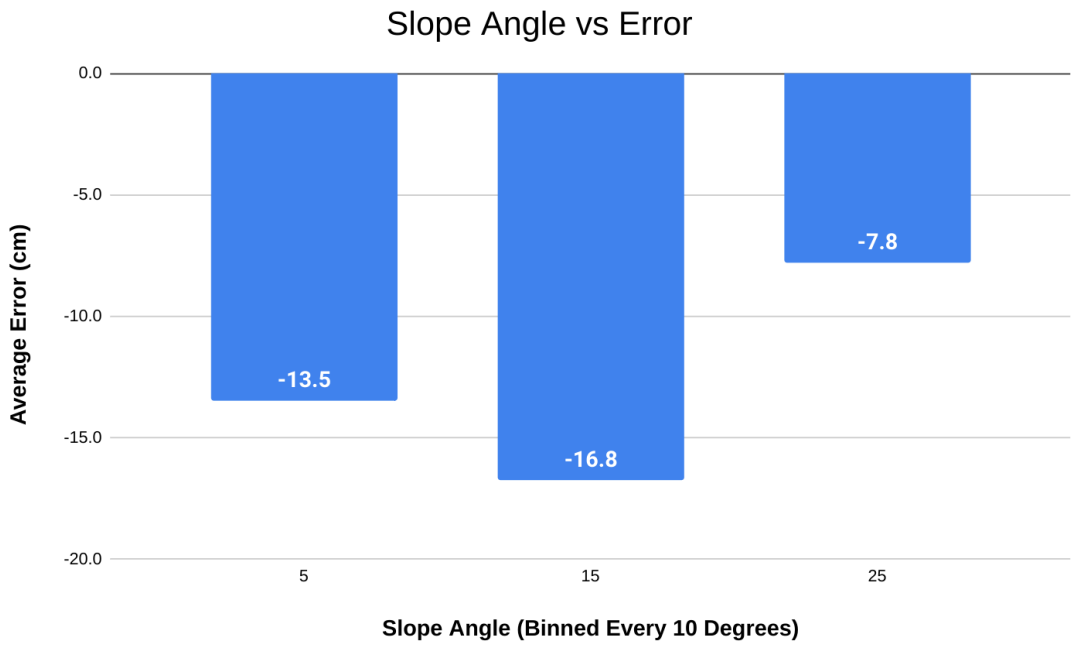


Figure 5.13: Snow depth error with slope angle. (Sample size 17 measurements)

In summary, we found that the best practices for snow depth calculations were to use the minimum z-coordinate depth when the 'Ground' scan has vegetation, and use the average z-coordinate depth when the 'Ground' scan is a smooth surface. This logic will hold true when calculating snow depth to the 'Intermediate Snow' scan as well.

CHAPTER SIX

CONCLUSION

6.1 Summary and Best Practices

Avalanches are a very powerful force of nature and can pose a danger to the public, roads, and buildings in mountainous terrain. Avalanche forecasting is an important part in mitigating this danger. We developed a software product and workflow for using a terrestrial scanning lidar system to aid avalanche forecasters in remotely analyzing snow depth in complex mountainous terrain. The software product allows the user to crop scans, align scans, then calculate and inspect snow depth distribution between scans. This is important in avalanche forecasting as snow depth is a key factor for avalanche formation and character [27] [25]. Through this development, best practices were determined for the workflow to ensure accurate analysis for avalanche forecasting.

Geo-registration and alignment of scans are imperative for accurate snow depth calculation between scans. Our workflow leverages tie-point alignment in Reigl's RiSCAN software as well as an Iterative Closes Point (ICP) algorithm that we wrote into our software. We found the best practice for alignment between scans is to use tie-points/reflectors to align scans taken at different times. If this initial alignment is not accurate enough, the ICP algorithm could be used and best results obtained by using point selection on man-made objects. Incorporating tie-points guarantees there will be objects to align on and this tie-point alignment should provide a robust initial guess for the ICP algorithm to improve upon. Using the ICP algorithm, scans from the same day showed a 65% improvement in alignment error and scans from different days showed a 13% improvement in alignment error. Man-made objects, such as lift towers or lift shacks, can be used for alignment if tie-points are infeasible or in order to try and improve on the initial tie-point alignment.

In testing alignment of two scans that did not have tie-points, we got a 98% improvement in our alignment error by using corresponding lift towers between scans. Further testing should be done to determine the efficacy of using distinct objects such as lift towers instead of tie-points/reflectors for alignment of scans.

When calculating snow depth we found that minor vegetation, such as grass or small shrubs, in the 'Ground' or 'Intermediate Snow' scan, can affect the accuracy of the snow depth calculation. The vegetation will cause the depth to have a consistent offset in the area when averaging of the z-coordinate. This is something to be aware of as the best practice is to use the depth calculated using the minimum z-coordinate when vegetation is present. If the two scans being compared have smooth surfaces, such as comparing an early season snow scan to a current late season scan, the best practice is to use the depth calculated to the averaged z-coordinate of the 'Ground' or 'Intermediate Snow' scan.

6.2 Future Work

6.2.1 Alignment

We ran into potential limitations with our alignment algorithm. Further testing should be done to determine the efficacy of using distinct objects such as lift towers instead of tie-points/reflectors for alignment of scans. Natural features, such as trees and cliffs, are constantly changing. Developing an algorithm to accurately align scans in spite of this would increase the versatility of our workflow.

6.2.2 Intensity Information Gain

My colleague, James Dillon, at Montana State University is doing research to determine what information can be inferred about the snowpack based on the snow surface reflectance value in the lidar scans. Incorporating his findings into the software could give forecasters more insight to make avalanche mitigation decisions.

6.2.3 Avalanche Modeling

There is potential to use our software to get input data for avalanche modeling. Based on the depth of the snowpack and surface characteristics of a buried layer (determined by the intensity return work James is doing), it could be possible to accurately model the destructive potential of avalanches.

REFERENCES

[1] I. Banos, A. Garcia, J. Alavedra, P. Figueras, J. Inglesias, C. Selles, P. Figueras, and J. Lopez. Snowpack depth modelling and water availability from lidar measurements in eastern pyrenees. *International Snow Science Workshop*, 2009.

[2] Paul Besl and H.D. McKay. A method for registration of 3-d shapes. ieee trans pattern anal mach intell. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14:239–256, 03 1992.

[3] K. W. Birkeland, K. J. Hansen, and R. L. Brown. The spatial variability of snow resistance on potential avalanche slopes. *Journal of Glaciology*, 41:467–479, 1995.

[4] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. *Computer Graphics Forum*, 32, August 2013.

[5] D. Chetverikov, D. Stepanov, and P. Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23:299–309, March 2005.

[6] Bahram Choubin, Moslem Borji, Amir Mosavi, Farzaneh Sajedi-Hosseini, Vijay P. Singh, and Shahaboddin Shamshirband. Snow avalanche hazard prediction using machine learning methods. *Journal of Hydrology*, 577:123929, October 2019.

[7] Jeffrey Deems, Peter Gadomski, Dominic Vellone, Ryan Evanczyk, Adam LeWinter, Karl Birkeland, and David Finnegan. Mapping starting zone snow depth with a ground-based lidar to improve avalanche control and forecasting. 2014.

[8] Jeffrey Deems, Thomas Painter, and David Finnegan. Lidar measurement of snow depth: a review. *Journal of Glaciology*, 59:467–479, 2013.

[9] Marcus V.N. d'Oliveira, Stephen E. Reutebuch, Robert J. McGaughey, and Hans-Erik Andersen. Estimating forest biomass and identifying low-intensity logging areas using airborne scanning lidar in antimary state forest, acre state, western brazilian amazon. *Remote Sensing of Environment*, 124:479 – 491, November 2012.

[10] L. Egli, T. Jonas, T. Grünewald, M. Schirmer, and P. Burlando. Dynamics of snow ablation in a small alpine catchment observed by repeated terrestrial laser scans. *Hydrological Processes*, 26:1574–1585, 2011.

[11] Y. D. Eo, M. Pyeon, S. Kim, J. Kim, and D. Han. Coregistration of terrestrial lidar points by adaptive scale-invariant feature transformation with constrained geometry. *Automation in Construction*, 25:49–58, August 2012.

[12] T. Goulden, C. Hopkinson, R. Jamieson, and S. Sterling. Sensitivity of dem, slope, aspect and watershed attributes to lidar measurement uncertainty. *Remote Sensing of Environment*, 179:23–25, June 2016.

[13] G. G. Goyer and R. Watson. The laser and its application to meteorology. *Bulletin of the American Meteorological Society*, 44(9):564 – 570, 01 Sep. 1963.

[14] D. Grant, J. Bethel, and M. Crawford. Point-to-plane registration of terrestrial laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 72:16–26, August 2012.

[15] E. Greene, Wiesinger T., K. Birkeland, C. Coléou, A. Jones, and G. Statham. Fatal avalanche accidents and forecasted danger levels: patterns in the united states, canada, switzerland and france. 2006.

[16] A. Gressin, C. Mallet, J. Demantke, and N. David. Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS Journal of*

Photogrammetry and Remote Sensing, 79:240–251, May 2013.

[17] T. Grünewald, M. Schirmer, R. Mott, and M. Lehning. Spatial and temporal variability of snow depth and ablation rates in a small mountain catchment. *The Cryosphere*, 4:215–225, 2010.

[18] Z. Guy and K. Birkeland. Relating complex terrain to potential avalanche trigger locations. *Cold Regions Science and Technology*, 86:1–13, 2013.

[19] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, Apr 1987.

[20] T. Kostadinov, R. Schumer, M. Hausner, K. Bormann, R. Gaffney, K. McGwire, T. Painter, S. Tyler, and A. Harpold. Watershed-scale mapping of fractional snow cover under conifer forest canopy using lidar. *Remote Sensing of Environment*, 222:34–49, March 2019.

[21] A.L. LeWinter, D.C. Finnegan, G.S. Hamilton, L.A. Stearns, and P.J. Gadomski. Continuous monitoring of greenland outlet glaciers using an autonomous terrestrial lidar scanning system: Design, development and testing at helheim glacier. *American Geophysical Union, Fall Meeting 2014*, 2014.

[22] C. Lowry, J. Deems, S. Leheide III, and J. Lundquist. Linking snowmelt-derived fluxes and groundwater flow in a high elevation meadow system, sierra nevada mountains, california. *Hydrological Processes*, 24:2821–2833, September 2010.

[23] Eric R. Lutz and Karl W. Birkeland. Spatial patterns of surface hoar properties and incoming radiation on an inclined forest opening. *Journal of Glaciology*, 57:355–366, 2011.

[24] Alexander Prokop. Assessing the applicability of terrestrial laser scanning for spatial snow depth measurements. *Journal of Glaciology*, 54(3):155–163, 2008.

[25] J. Schweizer and J. Jamieson. Snowpack properties for snow profile analysis. *Cold Regions Science and Technology*, 37:233–241, 2003.

[26] J. Schweizer, J. Jamieson, and M. Schneebeli. Snow avalanche formation. *Reviews of Geophysics*, 41, November 2003.

[27] J. Schweizer, K. Kronholm, J. Jamieson, and K. Birkeland. Review of spatial variability of snowpack properties and its importance for avalanche formation. *Cold Regions Science and Technology*, 51:253–272, February 2008.

[28] USGS. https://web.archive.org/web/20160219045753/http://lidar.cr.usgs.gov/. 2016.

[29] Nicholas E. Wayand, Christopher B. Marsh, Joseph M. Shea, and John W. Pomeroy. Globally scalable alpine snow metrics. *Remote Sensing of Environment*, 213:61 – 72, August 2018.

[30] Peyman Yariyan, Mohammadtaghi Avand, Rahim Ali Abbaspour, Mohammadreza Karami, and John P. Tiefenbacher. Gis-based spatial modeling of snow avalanches using four novel ensemble models. *Science of The Total Environment*, 745:141008, November 2020.

[31] H. Zald, J. Ohmann, H. Roberts, Henderson E. Gregory, M., R. McGauhey, and J. Braaten. Influence of lidar, landsat imagery, disturbance history, plot location accuracy, and plot size on accuracy of imputation maps of forest composition and structure. *Remote Sensing of Environment*, 143:26–38, March 2014.

[32] Q. Zhu, J. Wu, H. Hu, C. Xiao, and C. Wei. Lidar point cloud registration for sensing and reconstruction of unstructured terrain. *Applied Sciences*, 8, November 2018.