Evaluating the quality of ground surfaces generated from Terrestrial Laser Scanning (TLS) data

Yanshen Sun

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Geography

Laurence W. (Bill) Carstensen

Todd Ogle

Thomas Pingel

May 10th, 2019

Blacksburg, VA, 24060

Keywords: point cloud, ground filter, classification

Evaluating the quality of ground surfaces generated from Terrestrial Laser Scanning (TLS) data

Yanshen Sun

ABSTRACT

Researchers and GIS analysts have used Aerial Laser Scanning (ALS) data to generate Digital Terrain Models (DTM) since the 1990s, and various algorithms developed for ground point extraction have been proposed based on the characteristics of ALS data. However, Terrestrial Laser Scanning (TLS) data, which might be a better indicator of ground morphological features under dense tree canopies and more accessible for small areas, have been long ignored. In this research, the aim was to evaluate if TLS data were as qualified as ALS to serve as a source of a DTM. To achieve this goal, there were three steps: acquiring and aligning ALS and TLS of the same region, applying ground filters on both of the data sets, and comparing the results.

Our research area was a 100m by 140m region of grass, weeds and small trees along Strouble's Creek on the Virginia Tech campus. Four popular ground filter tools (ArcGIS, LASTools, PDAL, MCC) were applied to both ALS and TLS data. The output ground point clouds were then compared with a DTM generated from ALS data of the same region. Among the four ground filter tools employed in this research, the distances from TLS ground points to the ALS ground surface were no more than 0.06m with standard deviations less than 0.3m. The results indicated that the differences between the ground

extracted from TLS and that extracted from ALS were subtle. The conclusion is that Digital Terrain Models (DTM) generated from TLS data are valid.

Evaluating the quality of ground surfaces generated from Terrestrial Laser Scanning (TLS) data

Yanshen Sun

GENERAL AUDIENCE ABSTRACT

Elevation is one of the most basic data for researches such as flood prediction and land planning in the field of geography, agriculture, forestry, etc. The most common elevation data that could be downloaded from the internet were acquired from field measurements or satellites. However, the finest grained of that kind of data is 1/3m and errors can be introduced by ground objects such as trees and buildings. To acquire more accurate and pure-ground elevation data (also called Digital Terrain Models (DTM)), Researchers and GIS analysts introduced laser scanners for small area geographical research. For land surface data collection, researchers usually fly a drone with laser scanner (ALS) to derive the data underneath, which could be blocked by ground objects. An alternative way is to place the laser scanner on a tripod on the ground (TLS), which provides more data for ground morphological features under dense tree canopies and better precision. As ALS and TLS collect data from different perspectives, the coverage of a ground area can be different. As most of the ground extraction algorithm were designed for ALS data, their performance on TLS data hasn't been fully tested yet.

Our research area was a 100m by 140m region of grass, weeds and small trees along Strouble's Creek on the Virginia Tech campus. Four popular ground filter tools (ArcGIS,

LASTools, PDAL, MCC) were applied to both ALS and TLS data. The output ground point clouds were then compared with a ground surface generated from ALS data of the same region. Among the four ground filter tools employed in this research, the distances from TLS ground points to the ALS ground surface were no more than 0.06m with standard deviations less than 0.3m. The results indicated that the differences between the ground extracted from TLS and that extracted from ALS were subtle. The conclusion is that Digital Terrain Models (DTM) generated from TLS data are valid.

Acknowledgement

I would like to first thank my advisor Dr. Bill Carstensen, who has been so kind and patient to me to provide me mental and professional support. Until recently I knew that not all advisors could be so nice to pick out all your grammar errors throughout the whole thesis paper. I would also thank Dr. Todd Ogle and Dr. Thomas Pingel for being my committee members and providing me valuable advice in perfecting my research.

I also want to express gratefulness to Dr. Cully Hession, Dr. Thomas Tucker and Dr. Lynn Resler, who provided me data or the required devices to collect them for the research in this thesis paper. Thanks to Peggy Layne, Abhay Joshi, and Daniel Hung, my supervisors in the Office of Academic Decision Support, for providing me financial support while allowing me work a flexible schedule so that I can spend more time on my research.

I also really appreciate the professional help from Dr. James Campbell, Dr. Yang Shao, Adrien Arles, Trevor White, and Zachary Duer. I thank Dr. Jianger Yu for helping me with my daily life, and all the other people for being so nice to me during these years in the United States.

Finally, I want to thank my parents Dr. Hong Wang and Mr. Zhifeng Sun, who provided me both mental and financial support. They love me so much and have been understanding all the time. I have been so lucky to have them as my parents.

Contents

Chapter 1 Introduction 1

1.1 Background 1

1.1.1 Remote Sensing and Laser Scanning 1

1.1.2 Aerial Laser Scanning (ALS) and Terrestrial Laser Scanning (TLS) 2

1.2 Goal and Objectives 6

Chapter 2 Literature Review 7

2.1 Pre-processing 7

2.1.1 Elimination of data noise 7

2.1.2 Registration 8

2.2 Ground filters 11

2.2.1 The adaptive TIN model (Axelsson, 2000) 13

2.2.2 The Multiscale Curvature Classification Model (MCC) (Jeffrey, 2007) 16

2.2.3 The Simple Morphological Filter (SMRF) (Pingel, 2013) 16

2.2.4 The Slope-based Model (Vosselman, 2000) 19

2.3 Result validation 22

Chapter 3 Methodology 25

3.1 Research area 25

3.2 Devices used in this study 25

3.3 Field work 27

3.3.1 TLS and GPS 27

3.3.2 ALS 29

3.4 Preprocessing 29

3.4.1 registration 31

3.4.2 Denoising 41

3.4.3 Tiling 58

3.5 Comparing Available Ground filters 62

3.5.1 ArcGIS 62

3.5.2 LASTools 63

3.5.3 MCC-LIDAR 69

3.5.4 PDAL 70

Chapter 4 Conclusion 75

4.1 Result 75

4.2 Future work 90

Reference 92

Table of Figures

Figure 1-1: Comparison between ALS and TLS 4

Figure 2-1: Paper target (https://goo.gl/images/j16PMk), pad target (https://goo.gl/images/7zY5zC), and sphere target (https://goo.gl/images/usCEsd) 10

Figure 2-2: Calculation of point's value during densification 14

Figure 2-3: Strategy processing discontinuous surface 15

Figure 3-1: pictures of research area 26

Figure 3-2: Six scanning spots of TLS showing blind spots under each scan 28

Figure 3-3: ALS data (false color: Elevation) 30

Figure 3-4: Clipped ALS Data (false color: elevation) 33

Figure 3-5: Comparison between ALS points (Red) and TLS points (White) 37

Figure 3-6: Cloud-to-cloud distance estimation 38

Figure 3-7: Histogram of elevation after registration 40

Figure 3-8: Removing duplicated points 42

Figure 3-9: Outliers before denoising 44

Figure 3-10: Example of ROR filter 46

Figure 3-11: Volume density after removing outliers with ROR filter 48

Figure 3-12: Neighbor of neighbors after removing outliers with ROR filter 49

Figure 3-13: Excessive low noise 51

Figure 3-14: Coarse surface (gray shadow) 55

Figure 3-15: Points 2 meters below the coarse ground surface (red) 57

Figure 3-16: Buffer of tile 60

Figure 3-17: Cutting Strategy 61

Figure 3-18: Difference between results of different parameter sets 68
Figure 3-19: Difference between results of different cell size (0.1m and 1m) 74
Figure 4-1: Source data of ALS and TLS 78
Figure 4-2: Histogram of Cloud-to-Mesh Distance (0.0~0.1m) 81
Figure 4-3: Penetrability for grass of ALS and TLS of different distance from TLS scanner 83
Figure 4-4: Difference between results of the four ground filters 87
Figure 4-5: Distances from the scan positions to the nearest outline of ArcGIS Pro ground mesh 89

Table of Tables

Table 3-1: Classification code of LAS point (version 1.4) 53

Table 3-2: Built-in parameter sets in lasground_new 65

Table 3-3: Features of the 15 test datasets provided by ISPRS (Pingel, 2013) 71

Table 3-4: examined parameter sets for different landscape types (Pingel, 2013) 73

Table 4-1: Basic statistics of ground filter results 77

Table 4-2: Cloud-to-Mesh Distance Summary 79

Chapter 1 Introduction

1.1 Background

1.1.1 Remote Sensing and Laser Scanning

Laser scanning is a type of Remote Sensing (RS). Generally, remote sensors can be divided into two types: passive and active (Wehr et al., 1999). One of the most common examples of passive sensors is a simple camera. It receives light reflected by objects and generates pictures from these optical signals. Landsat's Multi-Spectral Scanner and Thematic Mapper, space-borne multispectral scanners very familiar to geographers, are also passive sensors. An active sensor emits electromagnetic waves and then receives and records the reflectance of them. The most familiar active sensors for geographers are Synthetic Aperture Radar (SAR) and Light Detection and Ranging (LiDAR), in which SAR sends and receives microwaves and LiDAR utilizes laser beams.

While microwave radar performs better for change detection, LiDAR is outstanding in range measurement. The high energy pulse of the laser light beam guarantees little diffusion with distance and a precise location measurement within a very short time. However, as would be expected, the distance between the sensor and the target object can affect the accuracy and precision of measurements. Generally, sensors on satellites can only provide data up to 1m resolution (one point per m^2) because of their orbit altitudes. High resolution laser scanning is defined as "scanning that provides more than 20 points per m^2 (Pirotti et al., 2013), and thus can only be acquired from scanners on low-altitude aircraft, drones, or ground scanners.

A 3D laser scanner emits an eye-safe laser beam and calculates its distance to a scanned object based on either the phase difference between the emitted and return signal (phase-based) or based on light's round-trip time of flight (time-of-flight based). Phase-based (PB) scanners deliver data faster than time-of-flight (TOF) scanners, but TOF scanners are more often used in outdoor settings, as they are capable of scanning more distant objects. (Becerik-Gerber, 2011)

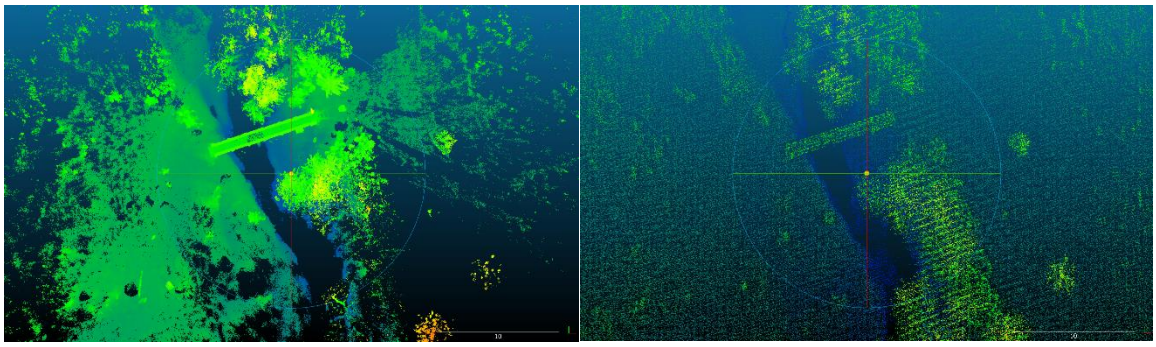
Laser scanning data have been used in solving various geographical problems, such as high resolution Digital Terrain (DTM) and Digital Surface Model (DSM) construction (Su et al., 2015), large scale forest parameter estimation (Riaño et al., 2004), soil erosion feature mapping (Perroy et al., 2010), and urban area modeling (Langerwisch et al., 2016). In addition, there are also "microscopic" applications, such as single tree model reconstruction (SHI et al., 2012), and Building surface modelling (Langerwisch et al., 2016), that can contribute to geo-visualization projects.

1.1.2 Aerial Laser Scanning (ALS) and Terrestrial Laser Scanning (TLS)

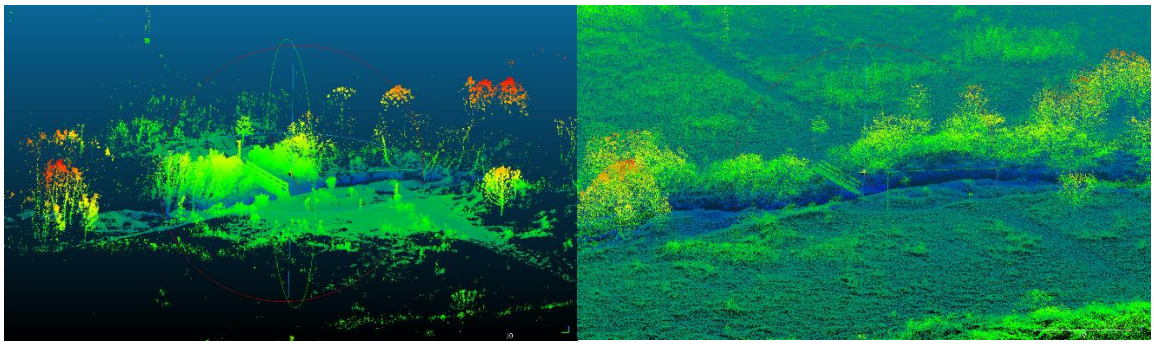
Aerial Laser Scanning (ALS) data have been used to model and identify geospatial characteristics of natural landscapes since the 1990s (Wehr et al., 1999). As ALS is able to acquire large area data on ground morphological features within a reasonable time, a large number of studies have been proposed to generate Digital Terrain Models (DTM) and Digital Surface Models (DSM) from ALS.

Terrestrial Laser Scanning (TLS) has been widely used for decades in 3D model construction for cultural relics, geological structures, architecture, individual plants, or even urban scenes, while it has rarely been used in the field of natural landscape model construction, especially not in terrain model construction. With the technological development of TLS equipment and techniques, however, terrestrial LiDAR can now precisely detect objects hundreds of meters away, and some ALS units are equipped with integrated GPS receivers just like airborne LiDAR devices, which enable TLS to contribute to applications focusing on larger area studies. While TLS cannot compete with ALS on regional coverage, many studies may merge TLS scans at strategic locations to improve understory or vertical coverage in conjunction with ALS.

Compared with ALS surveys, terrestrial laser scanning (TLS) has a relatively limited spatial coverage within one scan; the point density distribution of TLS is more irregular; and there are often obstructions causing gaps in data coverage (Pirotti et al., 2013). All of these disadvantages make it difficult to generate a complete terrain model from TLS. Nevertheless, TLS provides high-quality and very high density information at a much lower cost compared to ALS so it is worth investigation. While dealing with relatively small landscapes, it is proved to be more efficient than ALS in microscale forest inventory (Maas et al., 2008), erosion risk assessment (Schmid et al., 2004) and other applications requiring higher data density than generally possible in ALS. (De Agostino et al., 2012). TLS also generates data from a very different perspective from ALS, which complements information under dense foliage cover of the perpendicular surface (Fiocco et al., 2005) for airborne scanning data.



(a) TLS (left) to ALS (right) from top view: uneven distribution, data lost underneath scanner, blocked by ground objects



(b) TLS (left) to ALS (right) from side view: dense and detailed, full vertical features

Figure 1-1: Comparison between ALS and TLS

For the reasons mentioned above, the concept of a "millimeter scale terrain model" is proposed in this research. In the past few years, researchers have started to explore strategies for generating terrain models from TLS data, especially to support soil erosion and riverside roughness studies. Milenković et al. (2015) generated a millimeter scale terrain roughness model for a 2.6m by 3m seed bed using 10 TLS scans. In this research, the scans were arranged in a form of strips, similar to traditional spacecraft/aircraft laser scanning. (Bechet et al. 2015). Eltner et al. (2015) created a terrestrial model for a non-vegetated area during a rainfall event and examined the precipitation's erosive effect on a 40m by 50m area. All of the research above proved that it is very appropriate to generate a terrestrial model from TLS data for small and relatively unobstructed surfaces.

In our study, our TLS device did not support a GPS, so the TLS data would not be geo-referenced by the device. We compared three different methods to assign point cloud geospatial coordinates. Because our ALS unit did provide GPS geo-referencing, to evaluate quantitatively the accuracy of the registration, registered TLS ground point cloud data would be compared with a terrain model generated from geo-referenced ALS data. The result showed that, if the ground TLS unit does not provide the information (many do not), an external GPS device can be efficient in point cloud registration, making it possible to project TLS point cloud into geospatial coordinates by collecting GPS data asynchronously.

1.2 Goal and Objectives

In this research, the objective is to prove that terrestrial laser scanners can provide high quality natural landscape 3D data just like ALS. Meanwhile, TLS users do not need to explore new methods of processing TLS data, as traditional tools working on ALS data can also be used to process TLS data.

To achieve this goal, a site with many possible types of landcover features (artificial objects, high vegetation, medium vegetation, low vegetation, water, etc.) was chosen as the research area. By studying this area, we propose a pipeline for generating digital terrain models with several of the most popular point cloud classification tools. This pipeline can be useful for processing point clouds with any type of landcover, considering the rich types of objects that it can identify.

Chapter 2 Literature Review

2.1 Pre-processing

2.1.1 Elimination of data noise

The first step in processing point cloud data is denoising. Noise can come from different sources. For airborne LiDAR scanning, as the laser is cast directly from tens to hundreds of meters overhead, noise can be created by flying birds or other floating objects. To deal with this type of error, a height threshold is usually set to eliminate points that are too distant from the major cluster of the point cloud, for example, three standard deviations (Su et al., 2015).

For laser scanning of terrain, the distance from an object to the scanner can also lead to errors. Terrain laser devices usually scan surrounding objects from approximately 1-2m above the ground (Hopkinson et al. 2004; Clawges et al. 2007; Anderson et al. 2017). As the scanner is only 1m tall, TLS data from objects very far from the scanner can be acquired nearly horizontally. However, the further a point from the scanner, the less accurate it is. Thus, the concept of “range” is used to describe how far a reflectance point can be while guaranteeing an acceptable accuracy. In addition, when the distance from the scanner to a reflection point is extremely large, the power of the laser decays as the light goes further, and leads to a spread in the laser beam. The closer an object to the scanner, the smaller and denser reflection points the sensor receives. Points separated from the scanner beyond a determined effective distance may not be precise enough, because the footprint of an expanding light beam is too large to determine the accurate reflection point. This effective distance is defined as “range” for terrain laser scanners. For example, the FARO Focus3D

X series has three different types of devices with different ranges, 30m, 130m, and 330m. Points outside the range of devices should be eliminated, as they are not precise enough. A popular solution is the “Statistical Outlier Removal Filter” (Rusu, 2010). The method is based on the distances of each point and its nearest neighbor. First, the mean distance to one point’s k closest neighbors is computed. Then, assuming that the mean distances are normally distributed, the expectation and standard deviation are estimated. Finally, points with too large a mean distance to nearest neighbors (three standard deviations from the expectation) are considered as outliers and removed.

2.1.2 Registration

Registration methods can be divided into two categories: target-free and target-based (Becerik et al., 2011). Target-free registration uses Ground Control Points (GCPs) and identifiable features. High resolution GPS devices are set at recognizable positions (often where the TLS laser scanners are to be set) within the research area, so that both geospatial coordinates and scanner coordinates are available for a transformation matrix. Identifiable features can be segments, intersections, local planes, and other regular shapes, such as edge of roofs, corner of walls, and a section of a road. Fiocco et al. (2005) aligned terrestrial LIDAR data with aerial LIDAR data with facade/roof-top lines. Yang et al. (2016) extracted vertical feature lines from the point cloud and then took the intersection points of the lines and ground surface as registration references. These features are then employed to align raw laser scans with registered point clouds or photographs. Neither GCPs or identifiable features can be directly acquired from laser scanning points, and thus need to be geometrically calculated according to other points’ coordinates.

Target-based registration merges two or more point clouds by aligning “targets”. Generally, targets can be anything with high-contrast or highly reflective surfaces, the three most popular types of targets are paper, paddles, and spheres. Paper targets are paper printed with high contrast patterns, such as a chess board; a paddle target is a white pad with a tripod; while a sphere target is a white sphere with tiny supports. Among these types of targets, the sphere target has proved easiest to use (Becerik-Gerber et al., 2011). In contrast to a paper or paddle target, a sphere target does not require as much thought in target placement, because it looks the same from every perspective. On the other hand, the angle of a paper target requires it to be carefully placed, so that the scanner can capture the pattern from different orientations. Meanwhile, a paddle target requires rotation during the scanning procedure. We used the spherical option in our work mainly because there was no need to adjust the targets after placing them. We tried paper targets as well, but the area was too windy for paper targets to settle in place.

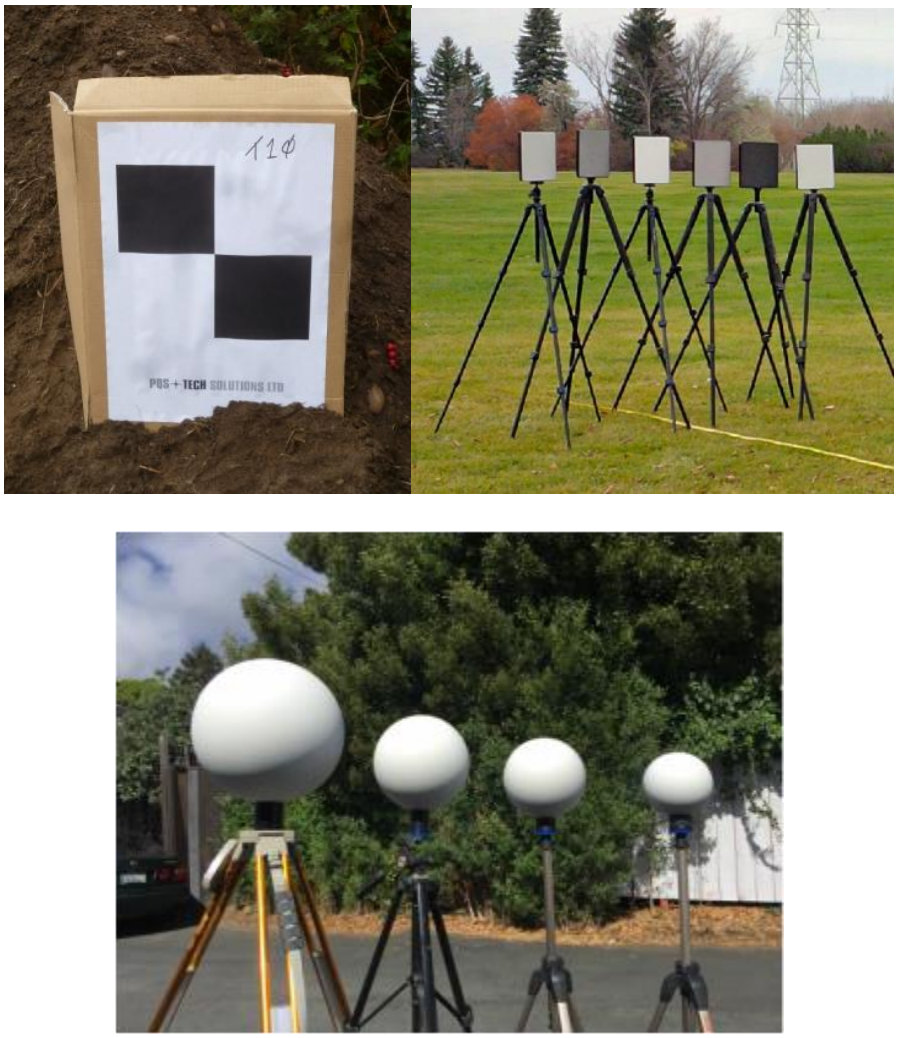


Figure 2-1: Paper target (https://goo.gl/images/j16PMk), pad target (https://goo.gl/images/7zY5zC), and sphere target (https://goo.gl/images/usCEsd)

Unlike GCP registration, target-free registration methods require an iterative least-squares algorithm to minimize error metrics. The most common algorithm is the Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992). This algorithm iteratively calculates the matrix of transformation to minimize the sum of distance between point pairs acquired from source point cloud and reference point cloud separately. Because of vertical obstructions in "real" environments, multiple scanning locations are required to create a complete point cloud for creation of a DTM or DSM. If enough scans are done to be effective, they are merged making a complete point cloud.

2.2 Ground filters

To extract features and objects from point clouds, the first step is to classify point cloud data into ground and non-ground points. Several methods have been developed for filtering non-ground data and they are the same regardless of the perspective of the source (ALS or TLS). The categories of slope and height-difference filters (Kilian, Haala, and Englich 1996; Kraus and Pfeifer 1998; Vosselman 2000), morphological filters (Zhang et al. 2003), moving-conicoid fitting filters (Zhang 2002), hierarchical filters (Briese and Pfeifer 2001; Hu 2003; Lin 2004; Huang 2006; Su et al. 2009), and progressive densification-based filters rank among the most popular approaches (Axelsson 2000).

Based on these methods, Meng et al. (2009) proposed the Multi-Directional Ground Filtering (MGF) algorithm to incorporate a two-dimensional neighborhood in directional scanning. Pingel (2013) simplified Zhang's Progressive Morphological Filter (PMF) (Zhang et al. 2003) and named his improved version Simple Morphological Filter (SMRF).

Błaszczak-Bąk et al. (2015) offered a method called M_{split} based on building functional models for ground surfaces. Su et al. (2015) developed a multi-scale weighted interpolation method with adaptive threshold selection for filter parameters.

Most researchers choose to retain only the first return of the points classified as ground. However, while dealing with natural landscapes in rural areas and forests, a combination of first and last returns is often employed (Briese et al., 2001; Kraus et al., 1999), because returns other than the first could record penetrable non-ground objects that are closer to the scanner. In our research, for TLS data, it would be better if we employed only the last return, as our research area was moist at the time we collected the data. The laser beam could hardly penetrate the fluffy soil/grass surface, which means that only the last return can be reflected by the ground. However, our Faro TLS points had no return attributes, and thus all of the points classified as ground were retained.

In this research, several popular ground filter algorithms are compared. These algorithms are implemented in commercial and open source point cloud processing tools (ArcGIS, LASTools, PDAL, MCC-LIDAR, and SAGA-GIS). Among these tools, ArcGIS is not an open-source software and thus the exact algorithm cannot be revealed. However, it is widely believed that it employs a modified algorithm of Axelsson's adaptive Triangulated Irregular Network (TIN) model (Axelsson, 2000), which is also employed by LASTools. The main idea of Axelsson's adaptive TIN model is to create a coarse TIN from a set of seed points, and then to densify the TIN with new points that meet a series of requirements (see Chapter 3 for detail). A Triangulated Irregular Network (TIN) represents a surface by

connecting measurement points with line segments forming a Delaunay triangulation. This technique differs from a raster grid, as a TIN describes the surface with triangles of varying sizes. Algorithms employed in PDAL, MCC-LIDAR, and SAGA-GIS employ grids for ground surface representation.

2.2.1 The adaptive TIN model (Axelsson, 2000)

This method first calculates initial values of TIN densification, distance threshold to the TIN facets, and the angle threshold to a point's closest TIN nodes based on user inputs and the median values of all points' surface normal angles and elevation differences. Then it creates a raster grid and extracts the lowest point within each grid cell as a seed point. These seed points are the nodes of the initial ground TIN. In each iteration, a point P is added to each TIN facet if it meets two conditions: (1) its distance to the closest TIN facet is shorter than a threshold value and (2) its angle to the closest TIN node is below threshold values. Specifically, to keep the cutting-off edges, a point is mirrored to its closest TIN node and it won't be added to the TIN if its deviation from the TIN facet is larger than the threshold value in (1). After each iteration, new threshold values of both distance and angle are recalculated based on the statistics of points within the TIN. The iteration stops until no new point added to the TIN (Figure 2-2 and 2-3, Axelsson, 2000).

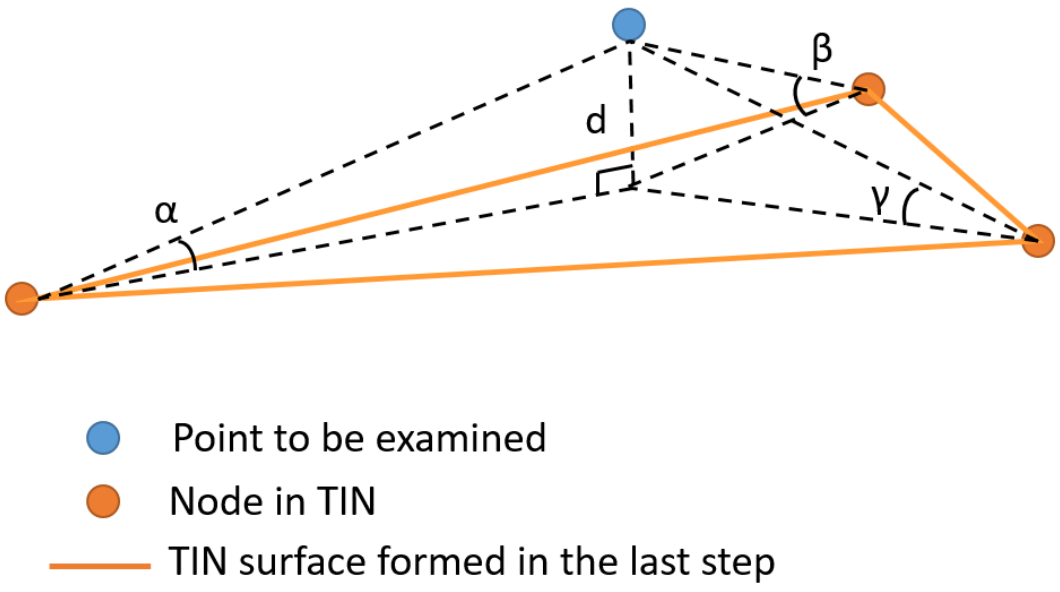


Figure 2-2: Calculation of point's value during densification

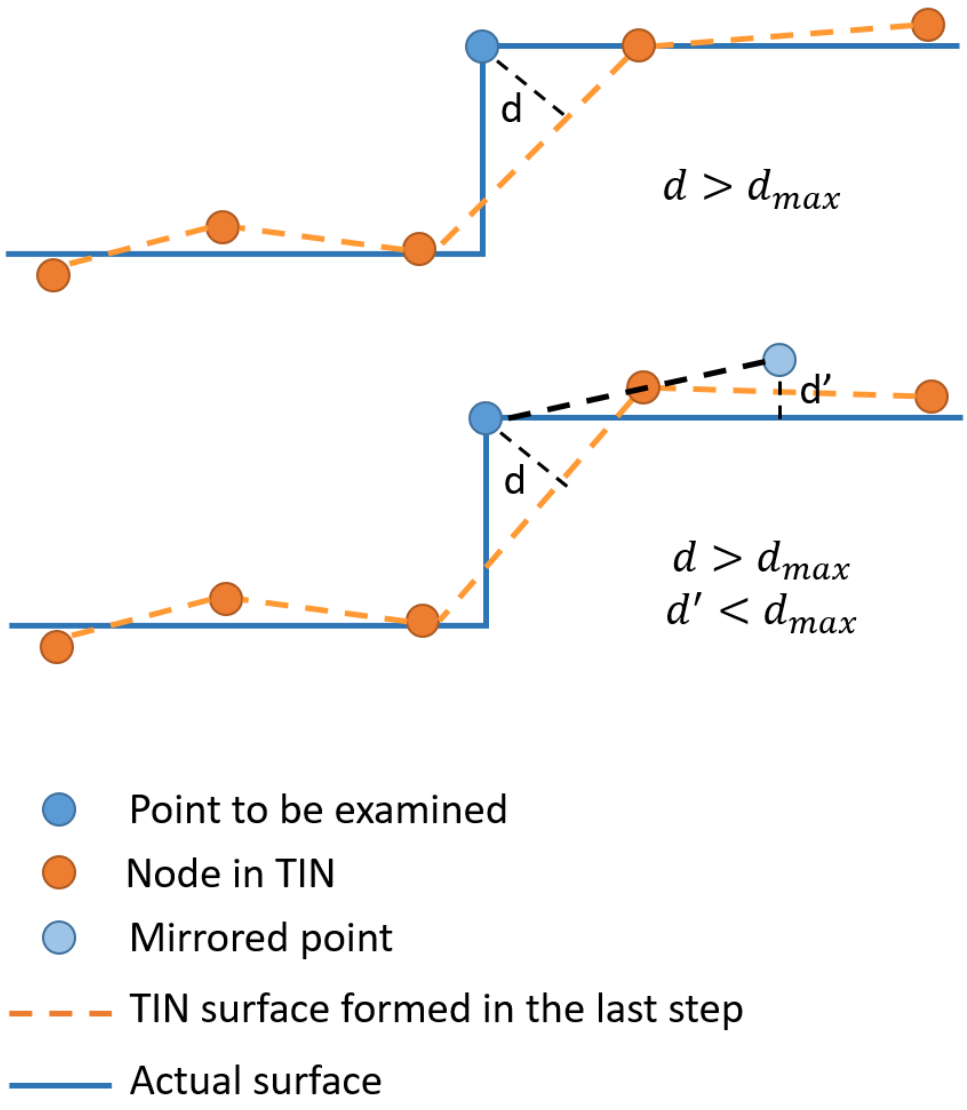


Figure 2-3: Strategy processing discontinuous surface

2.2.2 The Multiscale Curvature Classification Model (MCC) (Jeffrey, 2007)

Different from the adaptive TIN model, all the points are assumed to be ground points initially and then unqualified points are excluded from the point set. MCC first generates a raster with Thin Plate Spline (TPS) interpolation. The 12 nearest points to the cell center are used to calculate the pixel's z value. Each group of grid cells has a scale parameter s and an invariant tension parameter f ($f = 1.5$). Then a 3*3 mean kernel is used to smooth the grid. After interpolating a surface from the raster, points are eliminated if their z value is larger than the pixel value plus a curvature tolerance t. The remaining points are then used to refine the surface in the next iteration. As the relatively higher points are all eliminated, the smoothed surface tends to be lower and lower, until it is meets a "close enough" tolerance to "the lowest" surface.

In MCC, there are two hierarchies of iteration. The first level of iteration ranges from 1 to 3 and steps by 1 ($l = 1, 2, 3$). This iteration is mainly used to control the value of the scale parameter s and curvature tolerance t. While l equals to 1, s equals to $0.5s$ and t equals to t; when l equals to 2, s equals to s and t equals to $t + 0.1$; finally, when l equals to 3, s is changed to $1.5s$ and t becomes to $t + 0.2$. With each group of parameters, ground points are iteratively classified as non-ground points until less than "j" percent of the ground points changes class at a time, where j can be provided by users with a default value of 0.1.

2.2.3 The Simple Morphological Filter (SMRF) (Pingel, 2013)

SMRF is a modified version of PMF (Progressive Morphological Filter) added to the PDAL toolset in 2013. There are two major differences between SMRF and PMF: (1) the

window size increases linearly instead of exponentially and (2) an image painting technique is employed in empty cell interpolation instead of a nearest neighbor approach. These two changes reduce the number of user-defined parameters without increasing time complexity, which makes the algorithm more robust while dealing with different type of land cover.

The method first covers the research area with a grid, and then identifies points with the lowest elevation within each cell. Empty cells are filled with boundary cells of the empty area (spring-metaphor in painting technique (D'Errico, 2004)). Thus, a base grid terrain surface is generated. A "spring" metaphor is widely used in grid interpolation in the field of computer graphics. It assumes "springs" (with a nominal length of zero) connect each node with its eight neighbors in all the eight directions. Since each node tries to be like its neighbors, a function is set up to generate a compromise value for the empty cell according to its neighbors' value.

After initialization, the original grid surface is iteratively smoothed with an increasing window size. SMRF employs a disk-shaped structuring buffer with radius initially one cell, increasing by one cell until reaching the maximum window size or the bounding box edges. In the iteration, a method called "opening operation" is introduced. "Opening operation" consists of two stages, erosion and dilation. Given a window size w, erosion is a procedure setting cell values to the lowest cell value within w, while dilation is to turn cell values to the highest value within w after erosion. With these two operations, the surface can usually be simplified without changing the protruding objects' sizes, unless the window is so large

that the whole object is flattened during the erosion operation. At the same time, an elevation threshold (Et) is calculated by:

$$Et = St * c * N_{(W,c)}$$

$$St = \frac{dz}{dx}$$

in which "St" is the slope tolerance, "c" the cell size in meters, and "$N_{(W,c)}$" the number of cells within each window. "Et" is used to distinguish object cells from ground cells. The elevation difference between each cell's new value and its value of the last iteration is first calculated. Cells with values larger than the elevation threshold are all then classified as object cells.

As the window size grows, the grid surface becomes simpler and simpler. A max window size (Wk_{max}) is set by the user to stop the iteration. Wk_{max} should be larger than the largest object to smooth it down to the ground, but still reasonably small to keep the ground features. Different from other ground filters, SMRF can also eliminate low outliers by inverting the surface and filtering noises with a small window and large elevation threshold (e.g. $Slope = 500\%, Wk_{max} = 1$).

When the iteration is finished, a binary grid indicating if the cell is "ground" or "object" is generated. Therefore, ground cells can be extracted from the original base terrain surface with this binary grid. The result is a provisional DEM. With the provisional DEM, points can now be classified by calculating the z difference between the point and provisional

DEM. Instead of using a single value as the threshold of z difference, to keep steep slope features, "Et" for height differences is determined by interpolated (splined cubic interpolation) elevation and slope from the provisional DEM:

$$Et' = Et + scale * Slope_p$$

Where $Slope_p$ is the slope of the DEM at LIDAR point p.

2.2.4 The Slope-based Model (Vosselman, 2000)

The System for Automated Geoscientific Analyses (SAGA) employs a slope based ground filter developed by Vosselman in 2000. As the official document suggested, a workflow consisting of denoising and classification (Köthe, 2009) is supposed to be used.

The basic idea of Vosselman's method is that, the higher a point than its neighbors, the less possible it is a ground point. Based on this idea, there are two steps in the algorithm: generating an erosion surface and classifying points based on the possibility that they belong to ground or objects. The erosion surface is generated with a kernel operation. For each point P_i, the erosion value e_i equals the minimum value of a point P_j's height plus a height deviation variable acquired from the slope between P_i and P_j. ($e_{P_i} = [h_{P_j} + \Delta h_{max}(d(P_i, P_j))]$) P_j is any point in the cloud other than P_i. This function indicates that the further a point from P_i, the less it affects whether P_i is a ground point – the larger the corresponding erosion value. If P_i is lower or on the erosion surface ($h_i \leq e_i$), it is

considered as part of the ground points and therefore considered to be candidate nodes for final DTM generation.

More practicably, it is not necessary to compare $P_i$with all the other points. Instead, P_i can be rejected as long as it is higher than the local erosion surface generated by its neighbors. Therefore, in the modified version, a maximum search distance (MSD) is estimated from the largest height differences. For example, if the highest point on the terrain is 10m higher than the lowest point and $\Delta h_{max}(100m) = 10m$, the MSD is then set to 100m. In this case, a Delaunay triangulation is created to identify the distance from one point to another.

Considering that P_i is a regional lowest point, the algorithm is then able to verify all P_i's direct neighbors (nearest neighbors while dealing with point cloud) P_j with the equation mentioned above. Afterwards, if P_j is rejected (classified as non-ground point), its direct neighbors are then verified. The cluster of points is gradually enlarged until no point within MSD is added.

In the paper, Vosselman indicated that three types of functions could be used as $\Delta h_{max}(d(P_i, P_j))$ under different conditions. In the case that the maximum percentile slope (MPS) is known for a given distance d, $\Delta h_{max}(d(P_i, P_j))$ then equals to $MPS * d$. To account for noise in a point cloud dataset, a 5% confidence interval is taken and the equation becomes $\Delta h_{max} = MPS * d + 1.65 * 1.414 * \sigma$. If the MPS for d cannot be acquired directly, the distribution model $\Delta h_{max}(d(P_i, P_j))$ can be trained from a subset points. For a subset of N points, the probability density function can be:

$$f_{max}(\Delta h) = \frac{\partial F_{max}(\Delta h)}{\partial \Delta h} = NF(\Delta h)^{N-1}\frac{\partial F(\Delta h)}{\partial \Delta h} = NF(\Delta h)^{N-1}f(\Delta h)$$

For each distance d, the distribution model can then be obtained by taking integrals over this function. Different from the two functions mentioned above, the third method focuses on minimizing classification error. It connects the classification error measurement function to $\Delta h_{max}(d(P_i, P_j))$ and then turns it into an extremum problem.

Generally, there are two types of errors:

Type I error*: classify a ground point as a non-ground point (*$height\ error\ =\ h - h'$ *(projected point on the ground TIN))*

Type II error*: classify a non-ground point as a ground point (*$height\ error\ =\ h' - h$*)*

As a point P_i can best be classified as a ground point if and only if $P(P_i \in DEM) > P(P_i \notin DEM)$, the break-even point is $P(P_i \in DEM) = 0.5$. Assuming that $P(P_i \in DEM)$ is a function of Δh_{max}, d and if P_i's neighbor $P_j$belonging to the DEM, the equation can be written as:

$$P\left(\Delta h, d, P_j \in DEM\right) = \frac{P(P_i \in DEM, \Delta h, d, P_j \in DEM)}{P(\Delta h, d, P_j \in DEM)}$$

$$= \frac{P\left(d, P_j \in DEM, P_j \in DEM\right)P(P_i \in DEM, \Delta h, d, P_j \in DEM)}{P(\Delta h, d, P_j \in DEM)}$$

Practically, in the SAGA-GIS application, the "DTM filter" tool within the "Grid" tab is employed to classify the grid data. There are three parameters should be defined manually:

> ***search radius of the kernel in meters****: this parameter describes how far from the cell the height differences will be examined (should be at least half of the size of the largest object).*
>
> ***approximate terrain slope parameter in %****: used to modify the filter function to match the overall slope in the study area.*

Aside of the two parameters mentioned above, a confidence interval can be used to reject outliers, whose value is usually set to be 5%.

2.3 Result validation

There are three main types of accuracy assessment methods for ground and non-ground discrimination: visual validation, random sampling, and comparing to ground truth point cloud data. Visual validation is a qualitative evaluation method, and was commonly used in early research (Briese et al., 2001; Kraus et al., 1999; Kraus et al., 2001). This method is still appropriate when reference data are unavailable or obvious classification errors occur. However, with the improvement of scanning resolution and efficiency of ground

filter algorithms, misclassification is now usually caused by near ground objects such as bushes and grass, which are hard to recognize by human eyes.

The random sampling method randomly selects several ground and non-ground polygons from the filtered ground LiDAR data and compares them to manually labeled ground reference points. Shan et al. (2005) manually delineated polygons containing homogeneous ground points or building points, and then checked the number of points misclassified within those polygons; Zhang et al. (2003) visually examined 648 randomly sampled points; and Zhang et al. (2005) randomly selected 1600 points and visually examined them by comparing them with high-resolution aerial photographs. As for the Visual method, this method is not particularly convincing in areas with complex land cover (Meng et al., 2010).

The third method requires ground truth data, which, in certain locations or under certain conditions, may be collected from websites of authoritative Institutions or field surveys. Some institutions, for example, the International Society of Photogrammetry and Remote Sensing (ISPRS), provide accurately classified point cloud datasets, where the classification results were acquired by field surveys. These datasets cover typical landcover types such as urban, farmland, forest, barren land, etc., and thus are usually used by researchers reviewing different classification methods (Sithole, G., & Vosselman, G. 2003) or proposing innovative algorithms (Chen et al. 2012; Mongus et al. 2012; Thomas et al. 2013). Nevertheless, this method may be not reliable when the researchers try to estimate the accuracy on their own study area. Field surveys usually refer to sampling traditional

methods in person (Bechet et al. 2015; Eltner et al. 2015; Milenković et al. 2015), which is limited by the size of research area and precision of Lidar data.

Neither our ALS nor the TLS point clouds can be considered “correct”, so this study has no higher order dataset against which to compute accuracy. Therefore, to evaluate the quality of terrain point extraction from ground LiDAR data, we compared its vertical positions with those from the aerial terrain point cloud because the aerial techniques have long been used for this purpose and have been accepted as “reliable” in analytical work. Therefore, if the ground point set of TLS data is close enough to the ALS ground point set in the same area, a conclusion is drawn that the TLS ground point set is at least as “reliable” as the ALS ground point set. There are several ways for cloud comparison to be accomplished. The first and most straightforward method is to calculate the average Z-distance between a point within the target point cloud and its nearest neighbor point within the reference point cloud. However, error occurs when the resolution of reference point cloud (aerial) is far lower than that of the target point cloud, as not all TLS points have neighbors in the ALS point cloud. Other methods are possible to deal with the “sparse reference data” issue, such as to interpolate the reference point cloud and generate polygons from local points. The polygons can be least square best fitting plane of k nearest neighbors, Delaunay triangulations, or a quadratic height function (CloudCompare v2.10-alpha (64 bit), 2017).

Chapter 3 Methodology

3.1 Research area

Our study area is located in Blacksburg, VA. It is an approximately 142m*110m area mainly covered with grass and weeds. Stroubles Creek goes through the center of this area, and a wooden bridge crosses the creek in the center (Figure 3-1).

3.2 Devices used in this study

For ground LiDAR data collection, we used a FARO Focus3D X terrestrial laser scanner. FARO Focus3D X is one of the Focus laser scanner series produced by FARO (https://www.faro.com/). It ranges with near-infrared (NIR) with wave length of 1550nm and takes HDR photos for color complementation of the points. The company also provides software called "FARO Scene" for pre-processing and basic analysis of point cloud data.

To collect aerial LiDAR data, we equipped a VAPOR 35 Unmanned Aerial Vehicle (UAV) (Pulse Aerospace, Inc.) with a LiDAR mapping system. Our VAPOR 35 is an electric powered, advanced helicopter. It weighs 30 lbs. and can handle payloads of up to 5 lbs. (above the weight of its batteries) with a 45-60 minutes' flight time. The YellowScan Mapper Core ALS System with a weight of 2.1 kg (with battery) includes a multi-echo LiDAR sensor, GNSS RTK+PPK receiver, bi-frequency L1/L2, calibrated IMU, embedded computer, 2 integrated batteries, power and data cables, and data pre-processing software to produce georeferenced point clouds. Pulse Aerospace, Inc. integrated this unit for our Vapor35 to assure easy connectivity. YellowScan employs near-infrared (NIR) with wave length of 905nm for ranging.

(a) Stroubles Creek (Red rectangle indicates the approximate research area)

(b) close views of the scene

Figure 3-1: pictures of research area

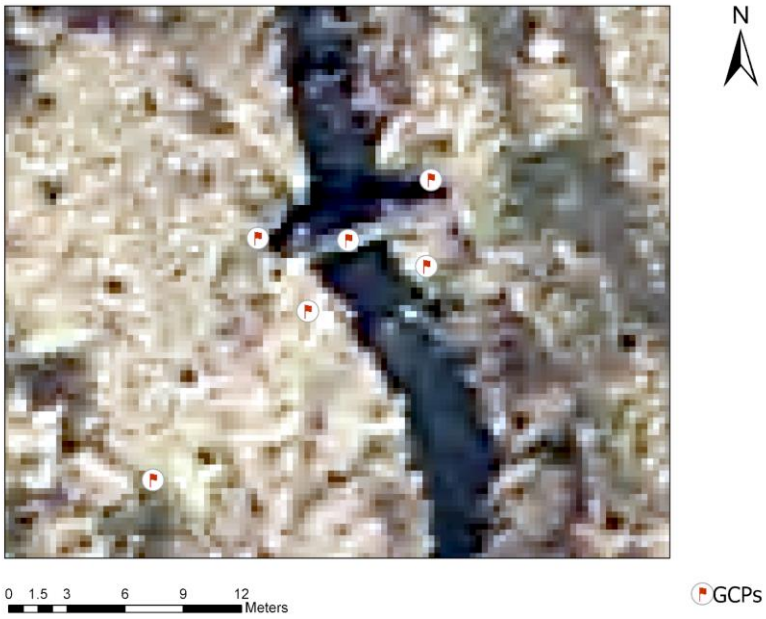
3.3 Field work

The data gathering procedure contains two parts: ground data collection and aerial data collection. In the first part, several overlapping terrestrial laser point clouds were taken individually, as well as the GPS location of each scanning spot. Nevertheless, the drone loading our ALS device also equipped with GPS, and thus the aerial data were all collected in one flight.

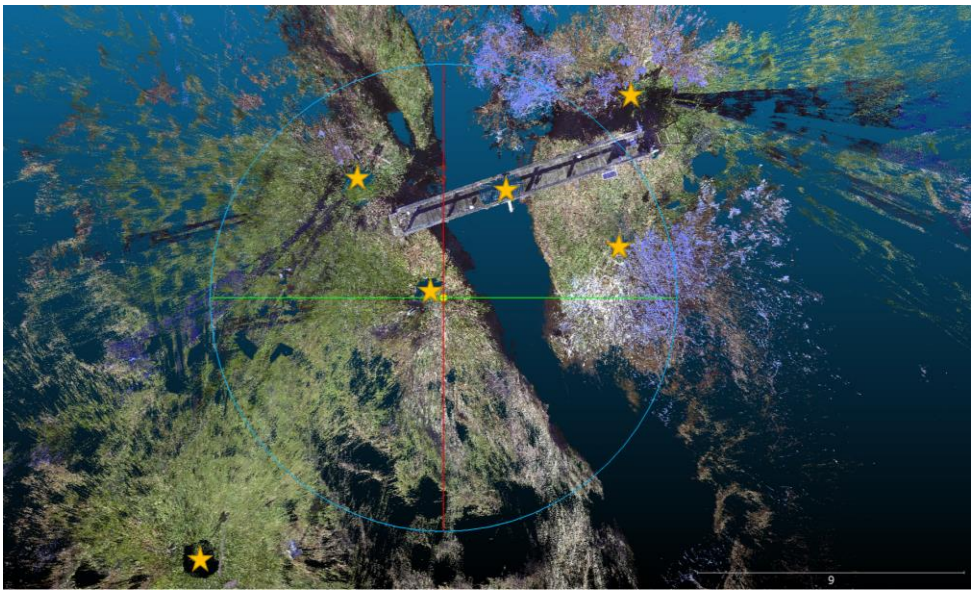
3.3.1 TLS and GPS

Our terrestrial laser scanning (TLS) data were collected by a FARO Focus3D X. The scanning covered the bridge and its surrounding area, and thus six scans were applied to the research area to cover as much of the research site as reasonable. We had scans from the four corners of the study area, one from the middle of the bridge, and one to supplement the others in areas we knew were hidden from the first five scan locations. Eight sphere targets were set on the bridge for registration of the scans.

The positions of scanning spots were selected visually in the field based on the objective to include all the typical landscape (creek banks, wetland, etc.) and objects (e.g. trees, artifacts, bushes, grasslands, bare land) in the area. We did not get 100% coverage for this research as that was not the focus of this work. Clearly the six scans provided valuable data for this research and additional scans would have been treated in the same manner as these six.



(a) Scan spots in a map



(b) Scan spots in our TLS data

Figure 3-2: Six scanning spots of TLS showing blind spots under each scan

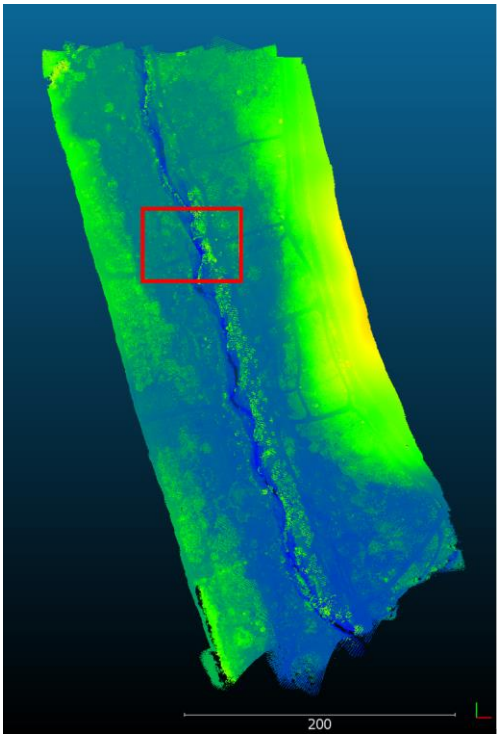
GPS data were taken with a Trimble Geo 7X. We set the handheld GPS device directly below the terrestrial laser scanner on sticks 10 centimeters above the ground. The geospatial coordinates were measured in GCS NAD-1983 (2011) and later projected to WGS84 UTM 17N. Twenty-three GPS and Glonass satellites participated in coordinate measurements. We took approximately 200 fixes for each scanner site and employed the mean value as the final measurement.

3.3.2 ALS

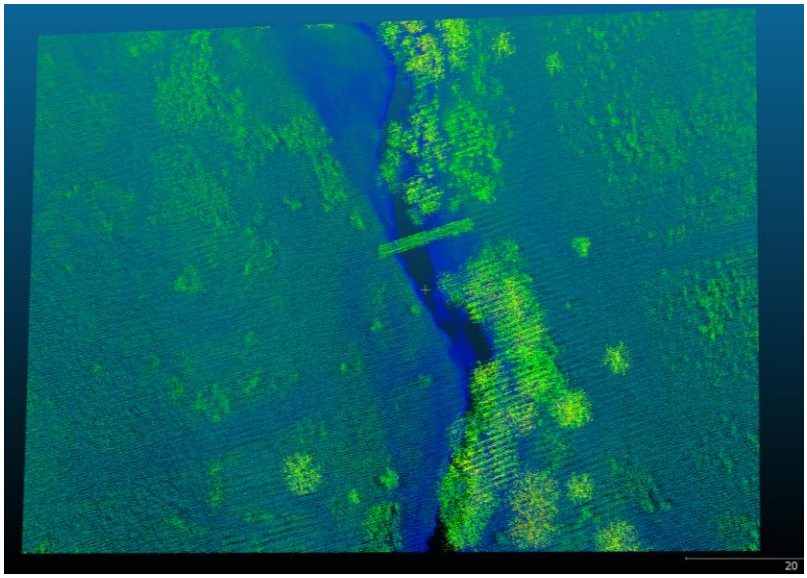
The aerial laser scanning (ALS) data were acquired by the YellowScan Mapper Core ALS System. A certificated drone pilot manipulated the Unmanned Aerial Vehicle (UAV) back and forth making several passes over the study area to densify the point cloud from an altitude of approximately 50m above the ground.

3.4 Preprocessing

The preprocessing step for both collection methods consisted of aligning the point clouds, eliminating noise and tiling the data into a reasonable portion for processing. During the alignment procedure, different scans were merged as a whole, which were done by the data providers. Then, the two point clouds (ALS and TLS) were registered to the same coordinate system to be compared.



(a) raw ALS data with the research area marked (red rectangle)



(b) ALS data clipped to the research area

Figure 3-3: ALS data (false color: Elevation)

For denoising of the TLS, four steps were taken to eliminate different types of noises. First, unexpected objects were removed, in our case these were mainly people (ourselves) caught by the scanner. Second, duplicate points were removed (there was only one of these in the dataset). The third step was to remove outliers, which are generally identified as data that are too sparsely spread to represent meaningful features. Finally, a series of operations were run to eliminate excessively low noise caused by water reflection. As for ALS, there were no obvious unexpected objects or excessively low points. Thus, only outlier removal was applied to the ALS data.

Considering of the limitations of the ground extraction algorithms' performances and the processing ability of our devices, after it was cleaned up, the point cloud was tiled into pieces with no more than 5 million points each for further procedures.

3.4.1 registration

1) TLS

Before registration, coordinates of each scan were assigned in their own polar coordinate system, whose origin point (0, 0, 0) was the scanner's position. All point coordinates represented the distance and orientation from the scanner's location. Therefore, the six scans had to be registered into the same coordinate system to acquire correct spatial relations to one another.

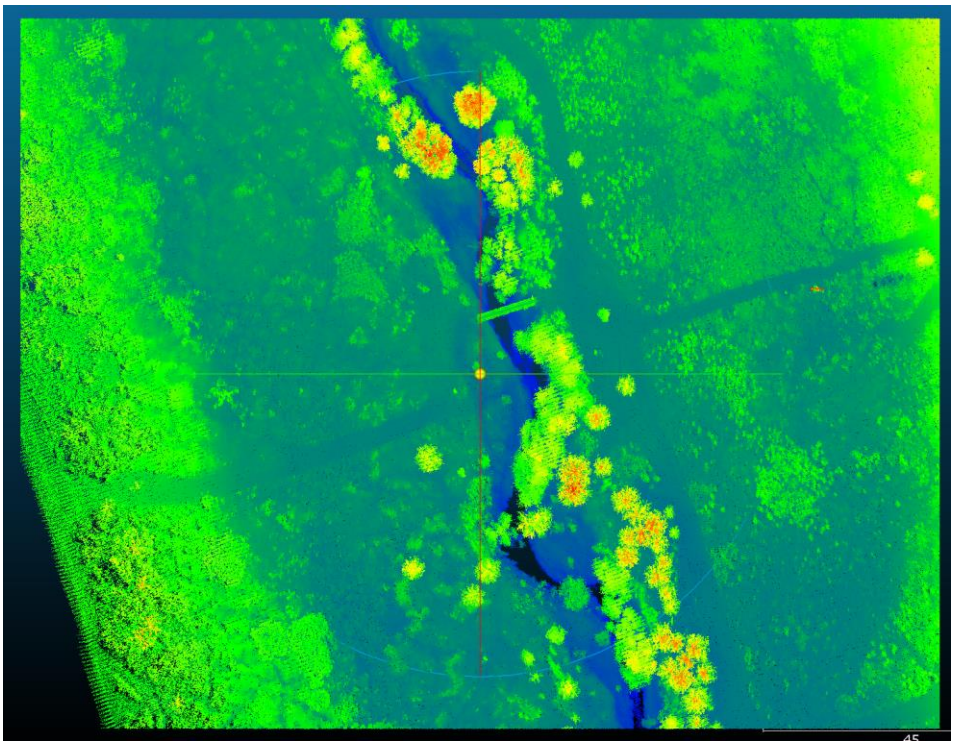
FARO Scene software was employed for registration. The basic idea was to put corresponding targets in different scans together. One of the six point clouds was first

assigned as the reference data. Afterwards the coordinates of the other point clouds were transferred into the reference. The user is first asked to depict the outlines of the same sphere target in different point clouds., The software would then calculate the accuracy of the target's center. When more than three targets were selected, a registration transformation matrix was generated. During the process, the error of each target after the transformation was calculated, so that the user could select targets that minimize the error. More than five reliable targets are usually required for precise registration of two point clouds. This work used six targets to assure that 3-5 would be visible in each scan.

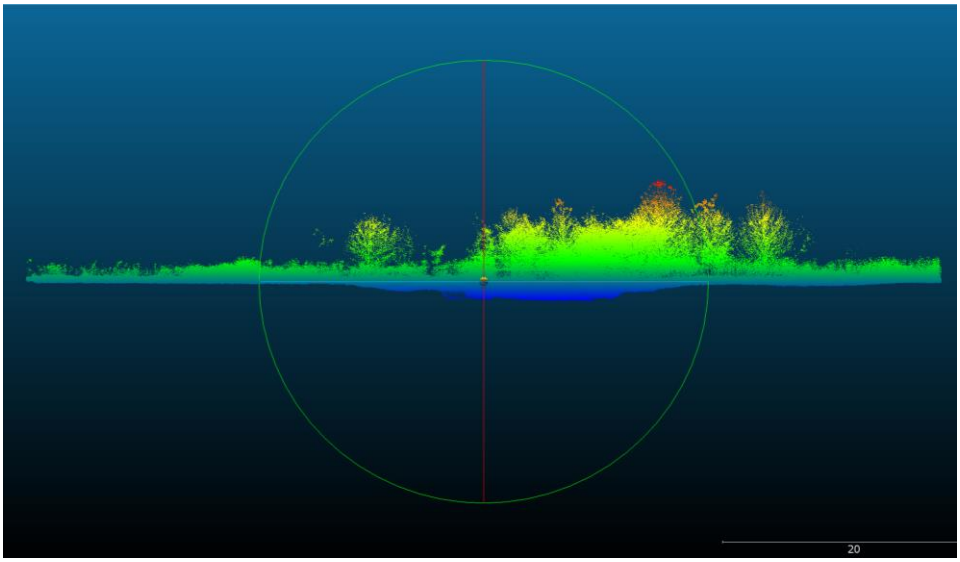
2) ALS

There were six flight lines for the ALS data, each of which contained a part of the point cloud. All these different aerial laser scans were registered through the UAV's trajectory. As there was a built-in GPS device on the UAV, the location of each scan was recorded in NAD83/UTM 17N while the scanner was collecting data. The point clouds were tied to the real-word coordinates by the flight lines. Afterwards, the six flight lines were merged together.

To reduce the data volume for the ALS data, the whole dataset was clipped to the extent of our research area (Figure 3-4) leaving 8,208,349 out of 42,582,800 points after clipping.



(a) clipped ALS data (top view)



(b) clipped ALS data (front view)

Figure 3-4: Clipped ALS Data (false color: elevation)

3) Register TLS to geographical space

In order to compare the TLS point cloud to the ALS cloud, we transformed the terrestrial scanning into NAD83/UTM 17N as well. As the TLS data and ALS data weren't taken at precisely the same time (the ALS was acquired on Nov 10th, 2017 and the TLS on Oct 31st, 2017), there were no artificial targets that could be used for registration, therefore, the TLS data were assigned geodesic coordinates as were the ALS points.

The major algorithm used in this process is called Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992). This algorithm was implemented in the software CloudCompare (CloudCompare v2.10-alpha (64 bit), 2017). Every point was taken into the calculation. To reduce errors caused by similar patterns of different objects, the TLS was first moved to a position close to the ALS by aligning the point 10-centimeters above the ground under the scanner to the GPS measurements. As indicated in Figure 3-2, each scan has a round "blind" area under the scanner's foot, whose centroid is the ground point just below the center of the scanner. Therefore, TLS data can be roughly registered by transforming these centroids into the GPS points' coordinate system. A small set of neighbor points was selected. Then, their mean value for each dimension was calculated at the centroids' coordinates. The coordinates were then added 0.1(meter) on the elevation value to be further used in estimating the transformation matrix from the coordinates of TLS to NAD 1983. Finally, the ICP tool was applied for more precise registration.

To evaluate the efficiency of our registration method in this step, the distance between the two point clouds must be estimated. "Distance" here generally refers to Hausdorff distance

– the longest distance from any point to its closest point in the other point set. However, as the two point clouds were produced neither by the same device nor at the same time, the mean distance could be a better indicator of the evaluation in this research.

Within CloudCompare, the “reference” and “compared” point clouds are assigned by the user. Then the distance from each point in the “compared” point cloud to the “reference” point cloud is calculated. There are generally two indicators of the distance between two point clouds: distance between the nearest points in different point clouds and distance between points from one dataset to a mesh generated from the other dataset. The second indicator is more frequently used when one of the point clouds is much sparser than the other one, because points in the denser dataset may not find corresponding returns in the other dataset.

In this research, the overall ALS’s density was 489.75 points per square meter with a spacing of 0.05m, while the density of our TLS was 9884.25 points per square meter and the spacing was 0.01m. By observing the registration result, we decided to thin both of the point clouds to a 0.05m spacing and keep only the highest points, because so many points in TLS didn’t have reference points in ALS. As displayed in Figure 3-5, because of its perspective from above, the ALS caught mainly the upper surface of the bridge, while TLS consisted of points from several roughly horizontal perspectives of the bridge. In this case, even if the bridge portrayed in both ALS and TLS had the same coordinates, the nearest ALS neighbor that a TLS point on the side surface of the bridge could find was on the top

surface. Therefore, although the registration worked fine, the average distance between the two point clouds would be too large.

As the thinned data was of the same resolution, both “cloud-to-cloud” distance and “cloud-to-mesh” distance could be considered reliable indicators of the precision of the registration. Specifically, a “local model” function was provided for calculating “cloud-to-cloud” distance, which based on the assumption that the “nearest neighbors” were actually from the same surface/object. Instead of the nearest neighbor, this mode evaluates the distance from a point to a local mesh formed by its k nearest neighbors or neighbors within a determined radius. Generally, six neighbors are enough for producing the local surface, while the radius is usually the average distance from one point to its sixth nearest neighbor.

In the “cloud-to-cloud distance” tool, three types of local surfaces can be generated: least square plane, 2.5D triangulation, and quadratic height function. The last one was employed here. As suggested in CloudCompare’s documentation, the estimation of “cloud-to-cloud” distance with local surface models was statistically more precise than that computed directly between points, but it might also lead to an underestimation of the distance, if the surface is generated by false points. Therefore, we calculated both. The distance with a quadratic surface model was 0.068m with a standard deviation of 0.163, while the result without a surface model was 0.082m with a standard deviation of 0.166.

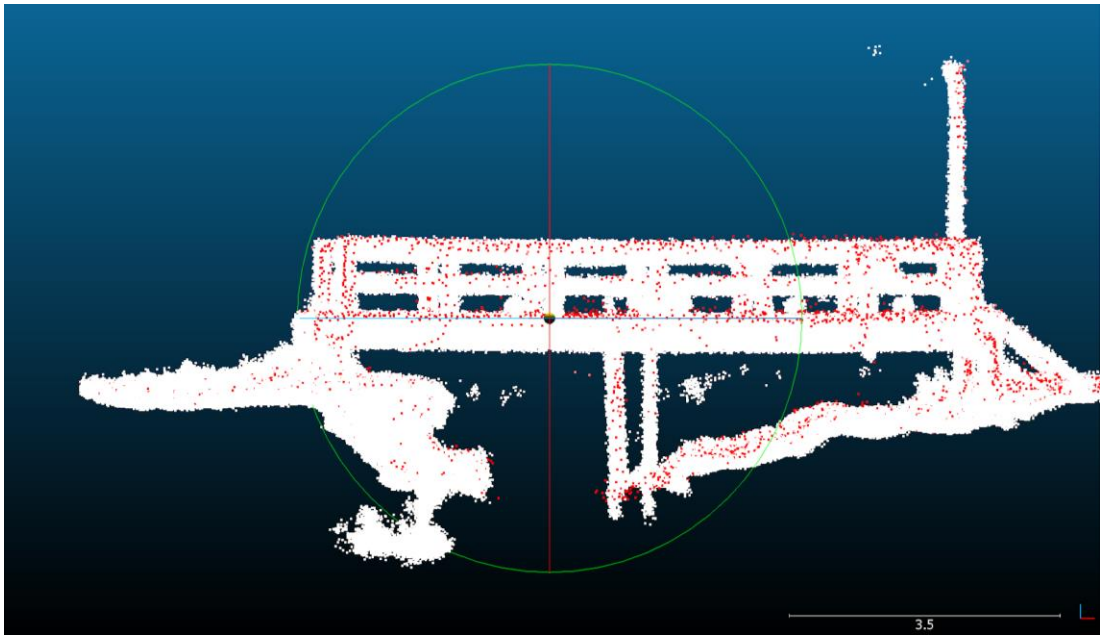
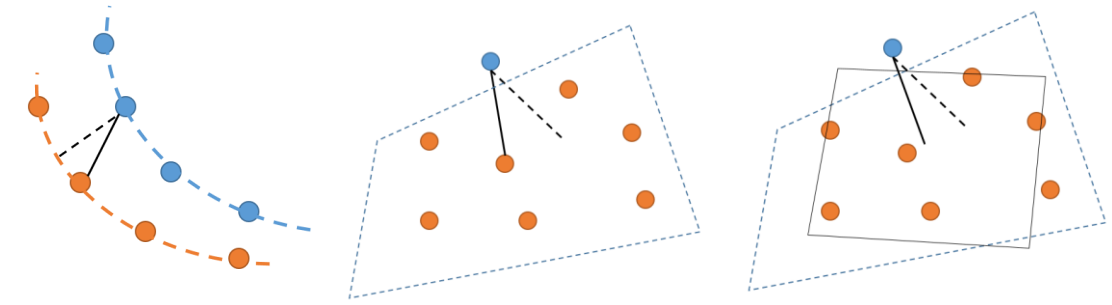
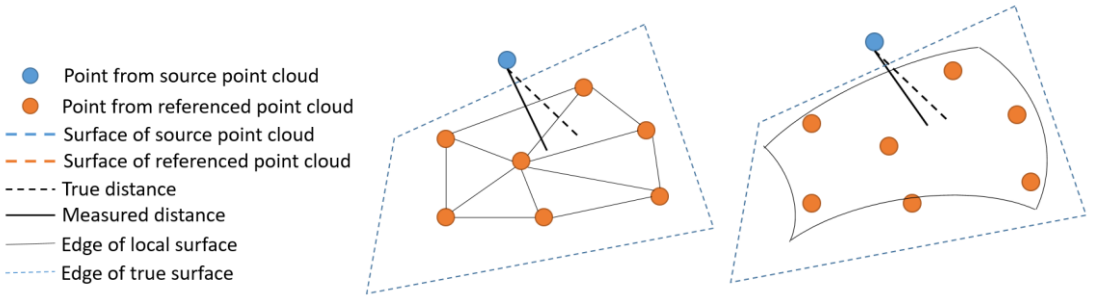


Figure 3-5: Comparison between ALS points (Red) and TLS points (White)



(a) cloud-to-cloud dist. **(b) nearest neighbor dist.** **(c) least-square-plane dist.**

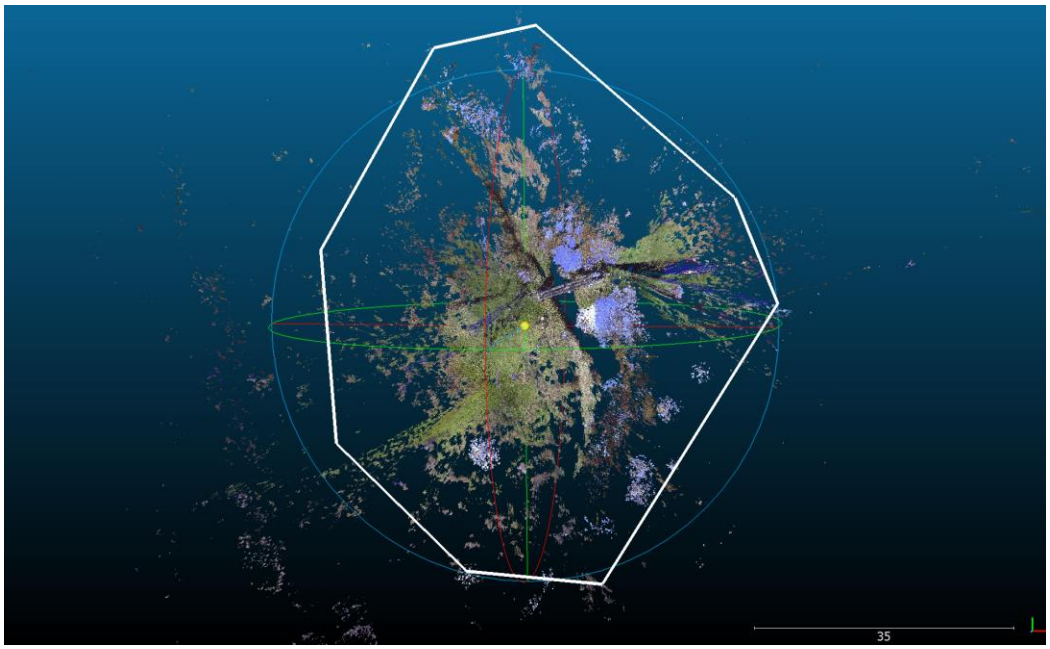


(d) 2.5D triangulation dist. **(e) quadratic-function dist.**

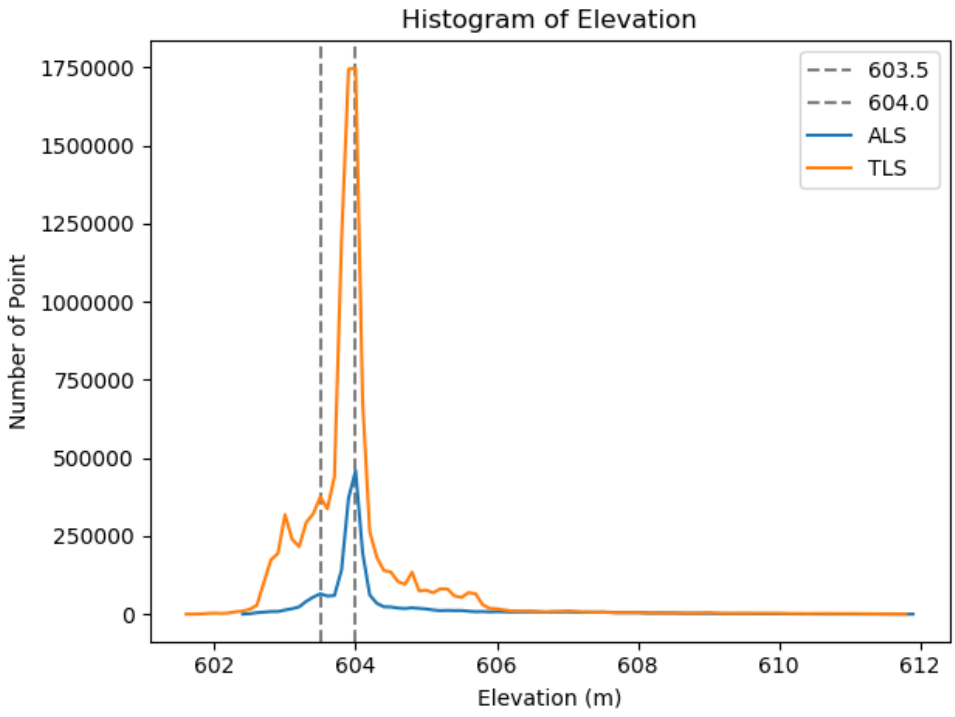
Figure 3-6: Cloud-to-cloud distance estimation

“Cloud-to-mesh” distance is usually employed in the condition that the reference point cloud is much sparser than the source data. The idea of this method is similar to Figure 3-6(d). As a result, the “cloud-to-mesh” distance much smaller than the other two measurements, with a distance of 0.016m and a standard deviation of 0.172. This concern is not really an issue though as the points will define a surface mesh as shown in Figure 3-6(d).

Furthermore, a histogram of elevation for both ALS and registered TLS is plotted, so that a comparison between the distribution pattern of elevations can be made. The central research area is segmented out for the histogram in order to reduce the difference in ground coverage between the two datasets. The boundary of segment is shown in Figure 3-7(a). Figure 3-7(b) clearly indicates that although the total number of points differs greatly, both of the point clouds show a very similar peak distribution. They both have a main peak at around 604.0m, which includes the majority of the ground points, and a secondary peak at around 603.5m, where points locate along the creek banks. Besides, the histogram of TLS shows a peak at elevation lower than 603.5m and several small peaks higher than 604.0m. The former peak represents excessively low noises (will be eliminated in 3.4.2), and the latter peaks represent points on the bridge. Overall, the TLS is accurately aligned to ALS’s coordinate system and ready for the next step.



(a) Boundary of segment



(b) Histogram of elevation within segment

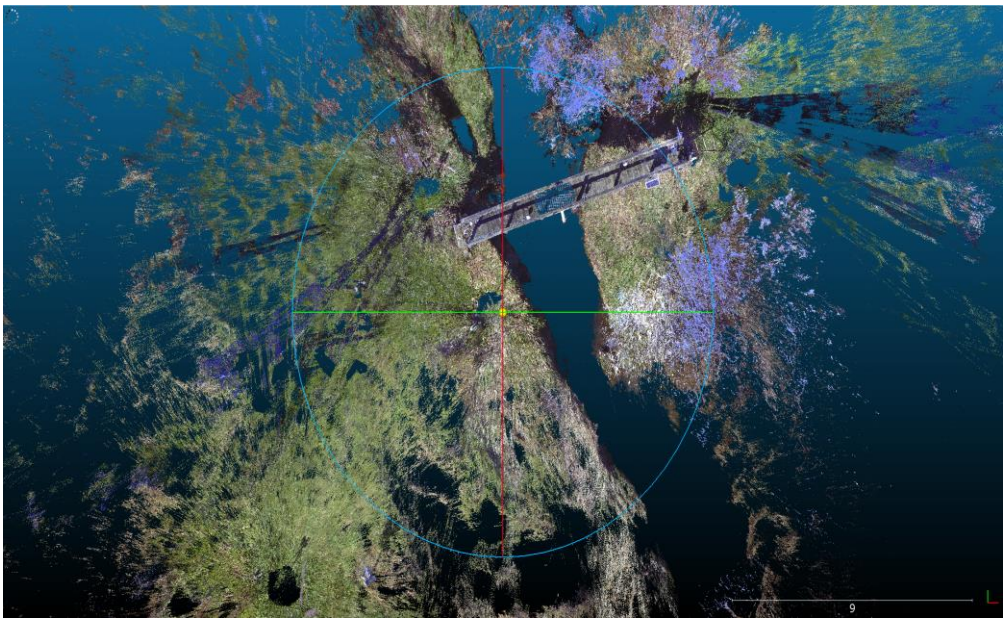
Figure 3-7: Histogram of elevation after registration

3.4.2 Denoising

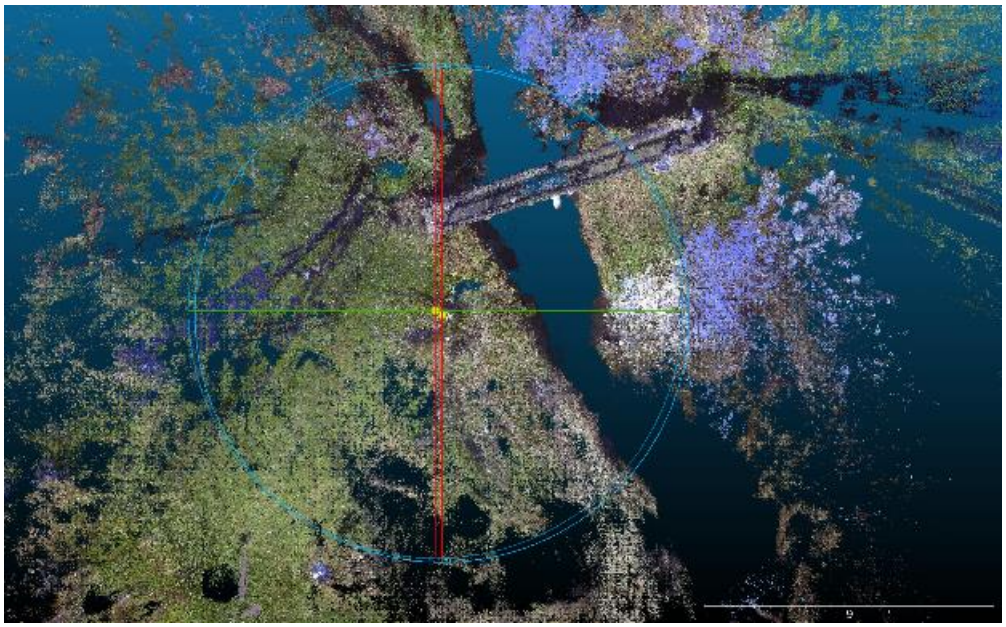
1) Subsampling and removing duplicate points

The original TLS point cloud contained 54,295,120 points. After manually deleting our personal portraits, 53,809,849 points remained. The average point density is 9884.25 per square meter.

Generally, producing point a cloud of such high density with different scans leads to plenty of duplicate points, as different scans catch the same features from different locations. However, what does “duplicate” mean? Obviously, there is no absolute “equal” or “not equal” in a decimal number – 1.00000000 can be considered the same as 0.99999999 if the required precision is lower than 10^{-8}. Therefore, the “remove duplicated points” tool in CloudCompare asks for a “cell size” from the user, which indicates that points are considered as “duplicated” if they are in the same cell. Our data were originally as precise as 0.0001 meter. Under the premise of keeping enough detail, but also to reduce the computation, we subsampled the data by setting the cell size as 0.001 meter. Only 11,570,685 points remained after this step, an average 2125.4 points per square meter (Figure 3-8).



(a) Before (53,809,849 points)



(b) After (11,570,685 points)

Figure 3-8: Removing duplicated points

2) Outlier filtering

As indicated in the height map (Figure 3-9), there are outliers both above and below the main cluster of the point cloud.

Based on the hypothesis that a point is outlier if it is further from its neighbors than most of the points (usually less than 6 neighbors within a 1-meter radius), two methods are typically employed to eliminate outliers, Statistical Outlier Removal (SOR) and Radius Outlier Removal (ROR).

SOR assumes that the point density is normally distributed all over the research area, and thus the distance from a point to its kth nearest neighbor should have a normal distribution. Therefore, if about 5% of the points need to be identified as outliers, just identify those points whose distance to its kth nearest neighbor is larger than 95% of all the points'.

The ROR is a very straight-forward noise filter. It eliminates points that have less than k neighbors within a sphere of radius r. For example, let $k = 6$ and $r = 1m$. In Figure 3-10, "Point" is not an outlier, because it has 6 neighbors within the radius of 1m. On the other hand, "Point j" only has 5 neighbors within 1m, which classifies it as an outlier.

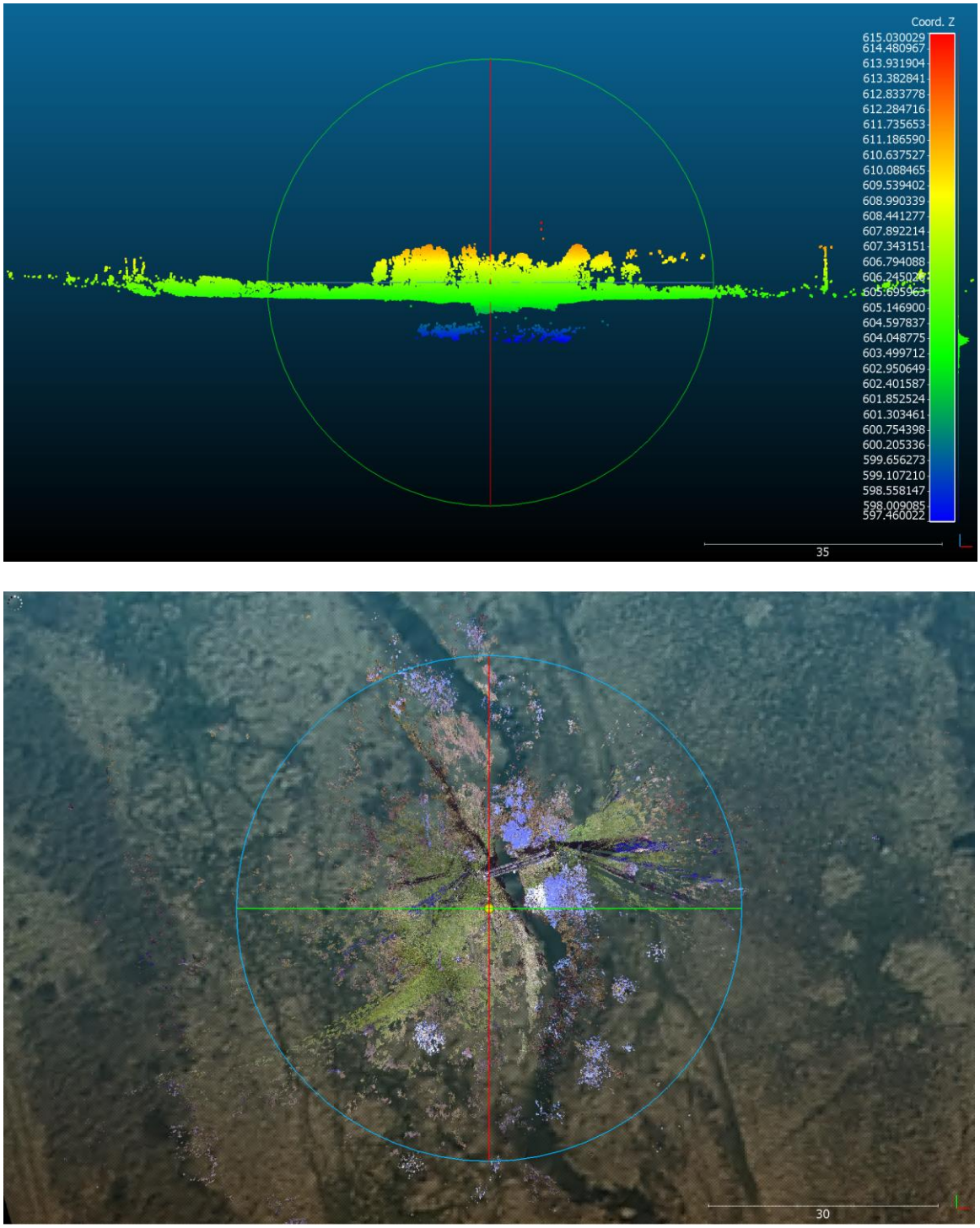


Figure 3-9: Outliers before denoising

The ROR method was chosen based on two features of our data. The first one is that, because of blockages by vegetation and the bridge in the TLS scans, the points are unevenly distributed in our research area, which leads to a non-normal distribution of point density and thus violates the assumptions of the SOR. The other reason is that relatively sparse points can still offer useful information. As our scan locations were set very close to one another, most of the points are crowded in the central area around the bridge, which makes the μ and σ in SOR small. However, even though a point is relatively far from the point cluster, the resolution around that point can still be high. For example, generally, 1% noise is a reasonable parameter for SOR. In our TLS cloud, assigning 1% of the total points as outliers means to remove 0.1 million points, which implies that points with less than 400 neighbors in 1m radius sphere will be eliminated. Nevertheless, in this application, 0.17m resolution is quite acceptable and those 1% points may still provide useful information, so should not be eliminated.

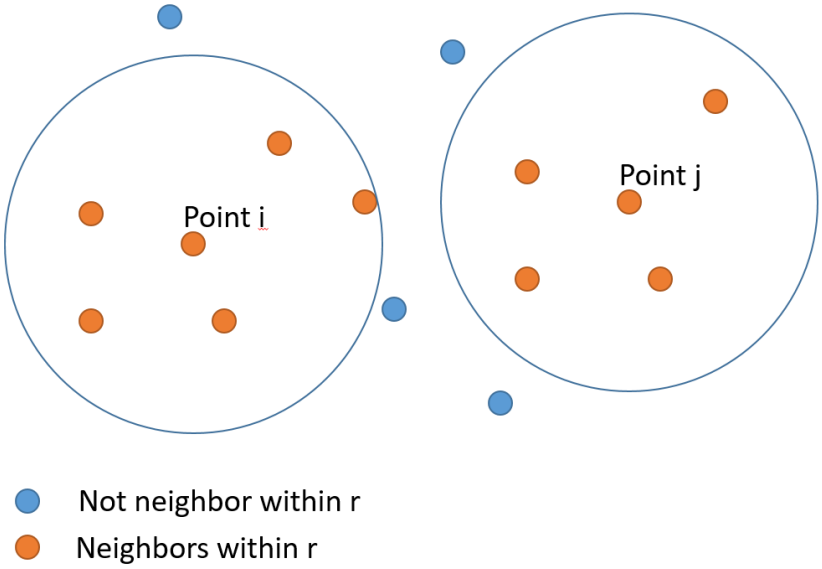


Figure 3-10: Example of ROR filter

To determine the proper parameters for ROR, the point density is first calculated. In CloudCompare, the "density" tool counts the number of points within each points' neighborhood. A radius parameter "r" is defined to generate a sphere "neighborhood" for each point. This parameter can either be input by the user or estimated from the distances between each point and its nearest neighbor. Three indicators of point density are provided: "number of neighbors", "surface density", and "volume density". "Surface density" equals the "number of neighbors" divided by πr^2 and "volume density" equals the "number of neighbors" divided by $\frac{4}{3}\pi r^3$. In this research, the radius "r" in density calculation is 1 meter. The result is as follows in Figure 3-10.

A series of number of neighbors values was visually tested on the dataset. A value of six neighbors was found to be a reasonable to both eliminate outliers and keep ground features appropriately. By applying the ROR filter, 11,569,889 points remained, with only 796 points classified as noise by the filter (Figure 3-11).

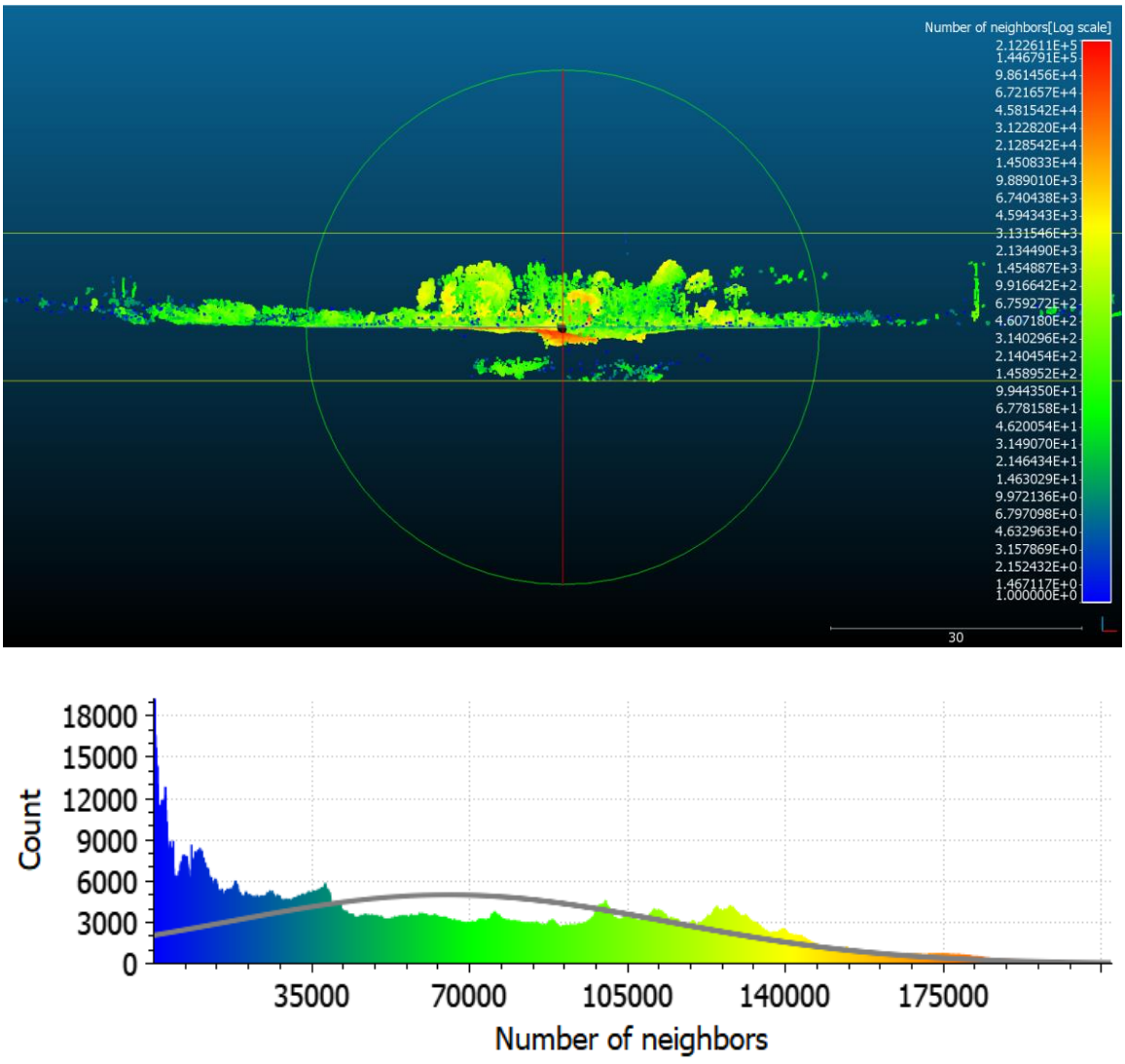


Figure 3-11: Volume density after removing outliers with ROR filter

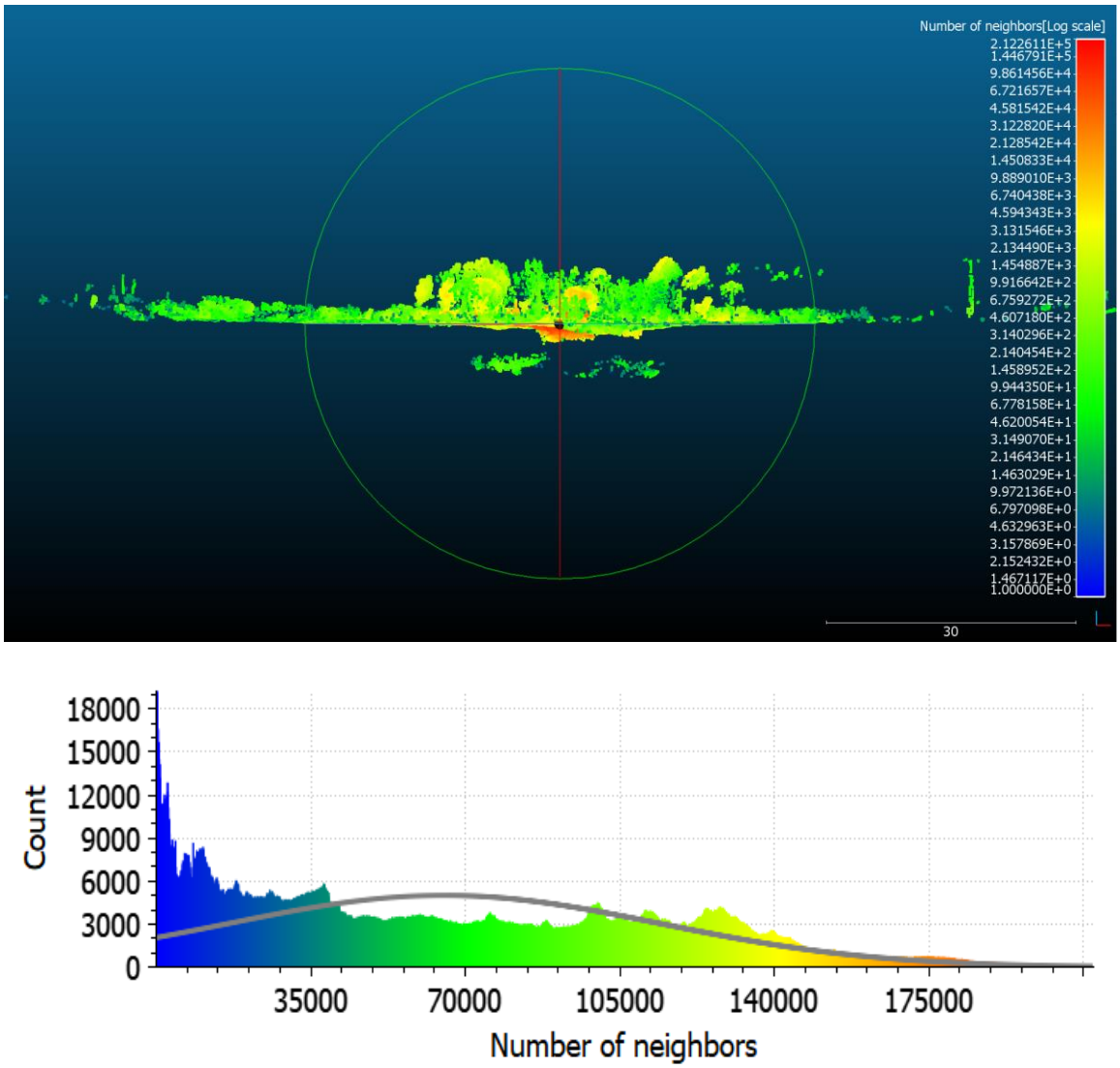
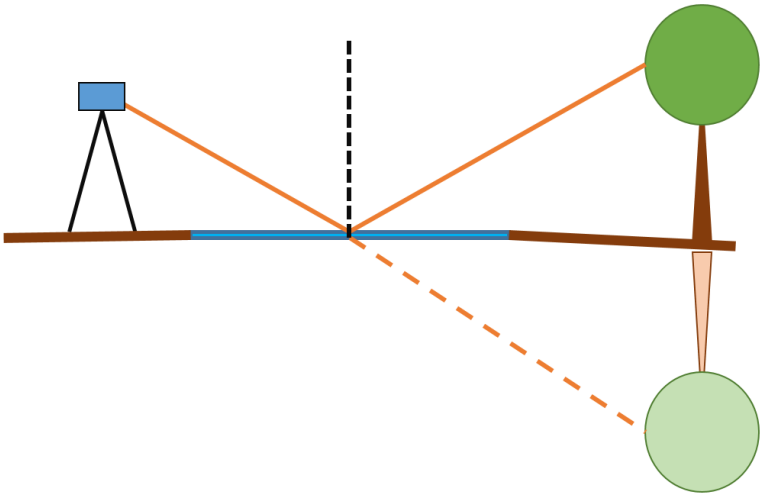


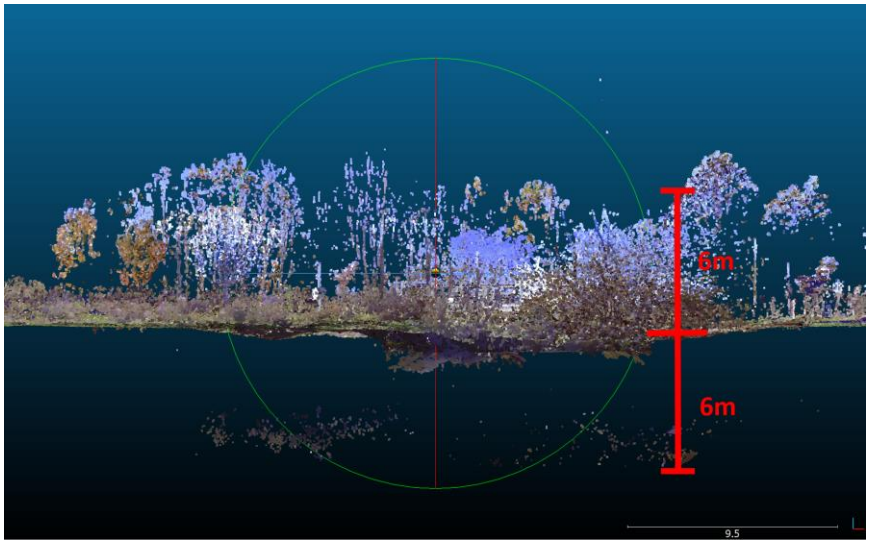
Figure 3-12: Neighbor of neighbors after removing outliers with ROR filter

3) removal of excessive low noise

In this research, the excessive low noise mainly came from water reflection. Although the definition of Lidar indicates that electromagnetic waves of different wavelengths, such as ultraviolet (10nm~380nm), visible (380nm~760nm), and near-infrared (760nm~2500nm), can be used for LiDAR, our ground and drone laser scanners employed near-infrared. According to the absorption spectrum of water, almost all energy of near-infrared is absorbed by water, especially those with wavelength of about 1.4, 1.9 and 2.7 μm, which means that hardly any near-infrared laser will penetrate water and reflect back. Therefore, signals captured by a near-infrared laser scanner are mainly reflections of a water surface instead of objects under water. When the incidence angle of the laser beam is not perpendicular to the water surface, the returned signal might come from the reflectance of an object (Figure 3-13).



(a) Schematic diagram how are excessive low noise produced



(b) Corresponding noise points and object points

Figure 3-13: Excessive low noise

To remove excessive low noise, the concept is to generate a rough ground surface and classify points below that surface as noise, so that any falsely caught underwater points would not affect the shape of ground surfaces generated in the next section. Multiple tools in LASTools (LASTools, 2014) were employed to solve this problem, such as lasthin, lasnoise, lasground_new, las2las, and lasheight. Among these tools, las2las is free for use, but, according to the documentation, lasthin, lasnoise, lasground_new's results would be "slightly distorted" if the number of points exceeds certain limits. As evaluating TLS as a cost savings in doing DTM construction is one purpose of this research and LASTools licenses are over $2000 for academic pricing, the loss of precision of "distortion" and "tiling" were also evaluated. The ground filters were applied on both the full dataset and a similiar tiled dataset to test if the effect of tiling is larger than the "distortion" on classification precision.

This section makes use of the class code of a point cloud, a value in the "classification" attribute for each point in a ".las" file. Table 3-2 lists all the classification code values of a point cloud (ASPRS, 2013).

Table 3-1: Classification code of LAS point (version 1.4)

Classification code	Meaning
0	Never classified
1	Unassigned
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Building
7	Low Point (noise)
8	Reserved *
9	Water
10	Rail
11	Road Surface
12	Reserved *
13	Wire - Guard (Shield)
14	Wire - Conductor (Phase)
15	Transmission Tower
16	Wire-Structure Connector (Insulator)
17	Bridge Deck
18	High Noise
19-63	Reserved
64-255	User Definable

Step 1: Generate a coarse ground surface

In this step, as the noise filter wasn't required to be precise, the point cloud was not divided into subsets. However, for section 3.5, ground filter, a strategy is proposed for tiling.

The highest points within each half-meter cell were extracted and temporarily classified as unused class "8" using the "lasthin" tool in LASTools. It first covers the xy extent with a uniform grid, and then picks out the highest, the lowest, or a random point from each grid cell. In this procedure, 10,358 points were identified.

Considering that the class 8 subset is effectively a new point cloud, we also need to eliminate outliers within the class. The "lasnoise" tool is designed for removing point outliers. Different from the SOR/ROR filters in CloudCompare, it divides the research area into cuboids (voxels) with user given length, width and height and identifies points within cells containing less than k points. 275 points out of class 8 were classified as class 7 (noise).

Finally, a rough ground was generated from the class 8 points with the tool "lasground". The rationale behind this tool will be discussed in section 3.5. 3,622 points out of 10,358 points in class 8 were classified as ground.

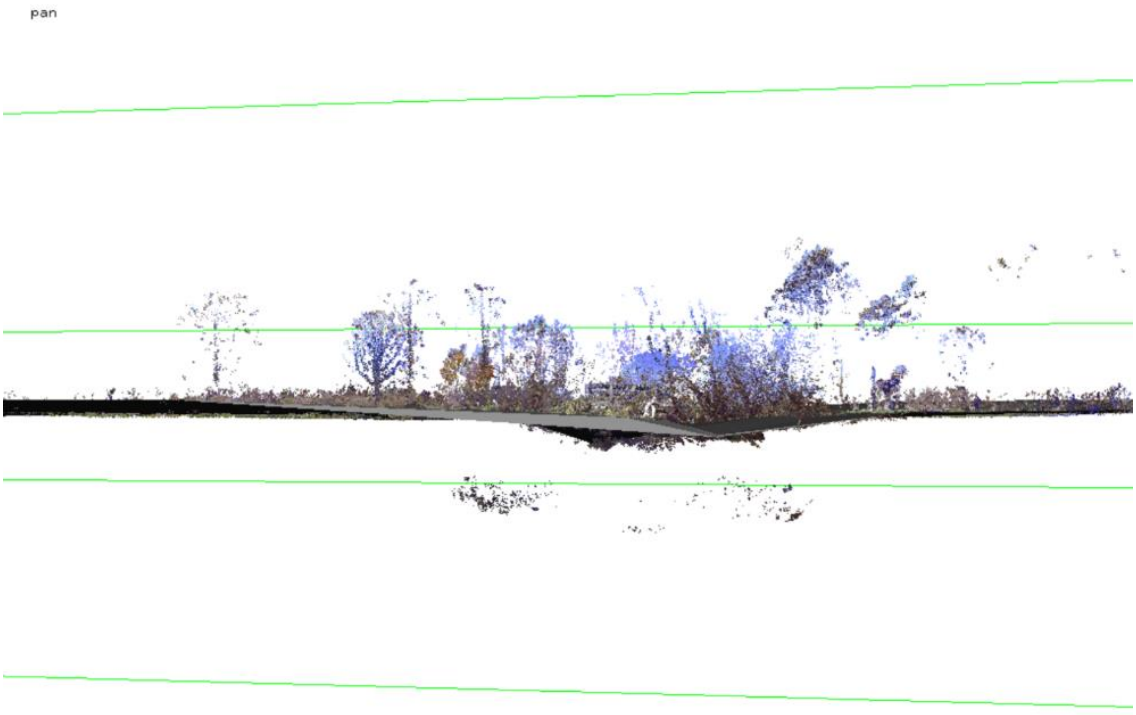


Figure 3-14: Coarse surface (gray shadow)

Step 2: Identify the points below the surface

The next step is to identify points much lower than the rough TIN generated from ground points acquired from step 1. The “lasheight” tool computes how far each point is from a certain class of points and enables users to classify points based on the range of those distances. In this step, points 2 meters or lower below the rough ground TIN were classified as class 12.

Step 3: Remove the points below the surface

Finally, to prepare for the ground classification algorithm, all the useful points need to be reclassified as 0 (unclassified) while all the noise needs to be deleted from the dataset. Therefore, the last step is to reclassify class 12 as class 7 and the other points back to 0 with the “las2las” tool. There were eventually 11,567,092 points that remained unclassified (0) while 2,797 points were classified as noise (7).

The noise was removed from the point cloud and “remove duplicate points” was applied again to the dataset to remove noise generated from LASTools. 10,463,811 points remained after cleaning up.

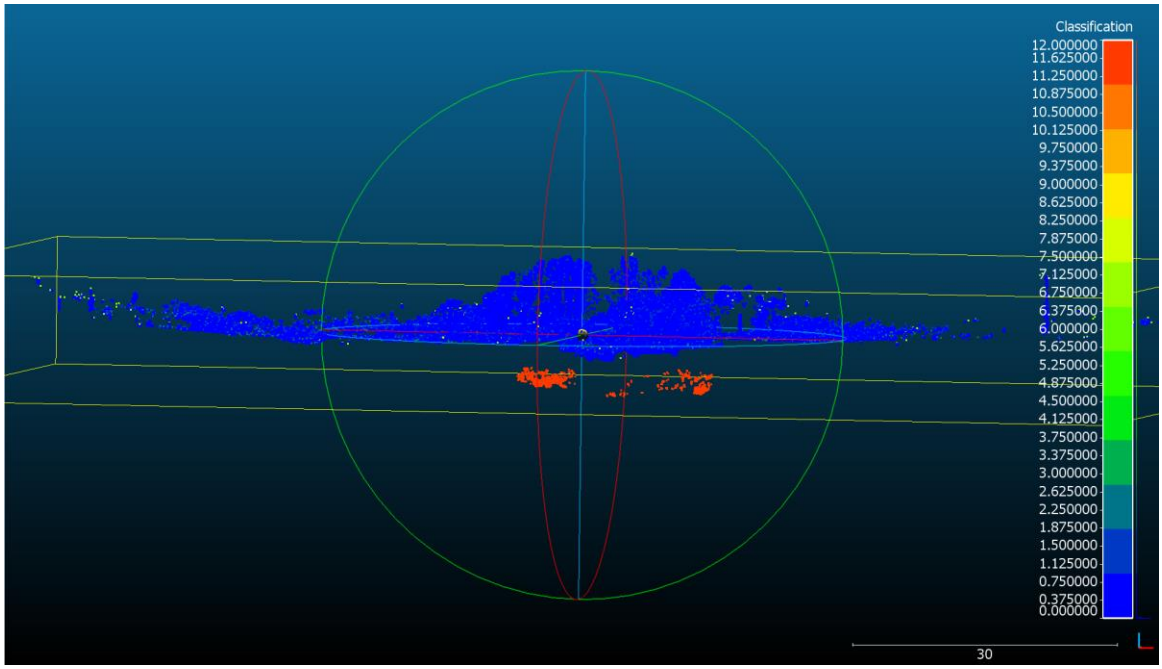


Figure 3-15: Points 2 meters below the coarse ground surface (red)

3.4.3 Tiling

There are usually ceilings restricting the number of points in ground filter tools. For example, the limitation of PDAL's (PDAL, 2018) "filter.ground" in Docker (https://www.docker.com/) is 5,000,000 points; the unlicensed ground filter "lasground_new" in LASTools add noise and distortion to results of point clouds with more than 1,500,000 points. We compared the performance of "lasground_new" on the whole TLS data with that on tiled data and found no obvious differences. Therefore, in this research, the data was only sliced into four pieces to meet the requirement of PDAL.

General tiling tools usually slice the point cloud with an evenly distributed voxel grid with fixed size. However, this strategy only works well on evenly distributed points. For TLS data, to achieve an appropriate volume of points at the scanning center, the cell size tends to be very small, which leads to a scarcity of information at the edge of scanned area. LASTools provides a "refine" option that enable users only subdivide tiles with more points than a threshold. The downside of this tool is that, users need to manually define the cell size and offsets of x and y in every subdividing operation. A tool called "chipper" in PDAL is able to slice point cloud into subsets with no more than "n" points, where "n" is a user given parameter. Nevertheless, "chipper" doesn't have a "buffer" option in its parameters. "Buffer" refers to extra areas beyond the true boundary of a tile that overlay with other tiles. For instance, in Figure 3-16, there are two tiles, whose true boundaries are the blue lines. However, the red and green rectangle are area covered by tile 1 and tile 2. The extra data points in the buffer area smoothens faults between ground

surfaces of adjacent tiles, which is pretty important in reducing edge effects in TIN based ground filters.

As none of these were useful in our datasets, we wrote a function to automatically tile point clouds into pieces with no more than 5,000,000 points with a 5% length span of the longer-edge buffer for each tile. To balance the number of points in each subset, the centroid xy coordinate of the tile was identified. Then the maximum distance from the centroid to edges was calculated and thus the length of buffer equals to the maximum distance times 5%. Finally, the tile was cut into 4 pieces by a cross centered at the centroid, while these pieces had a thick frame overlapped with one another.

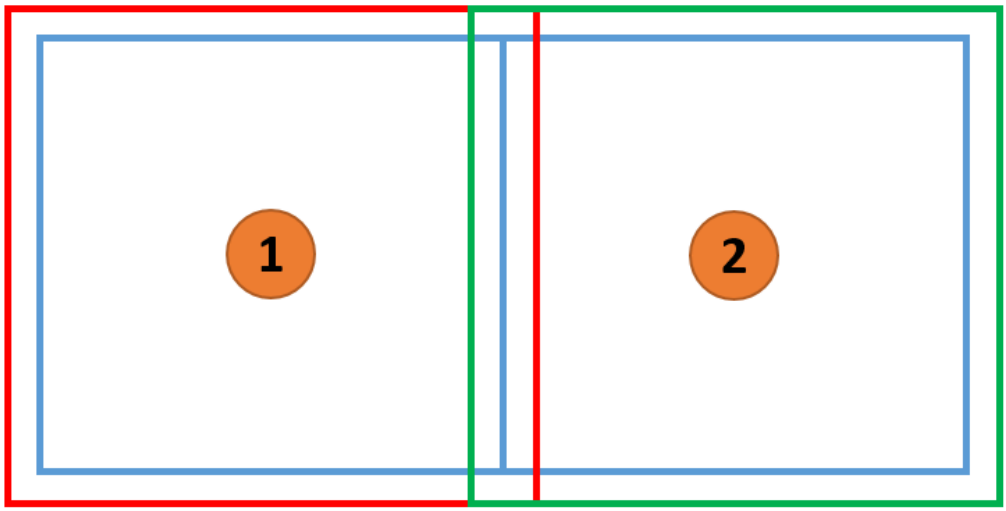


Figure 3-16: Buffer of tile

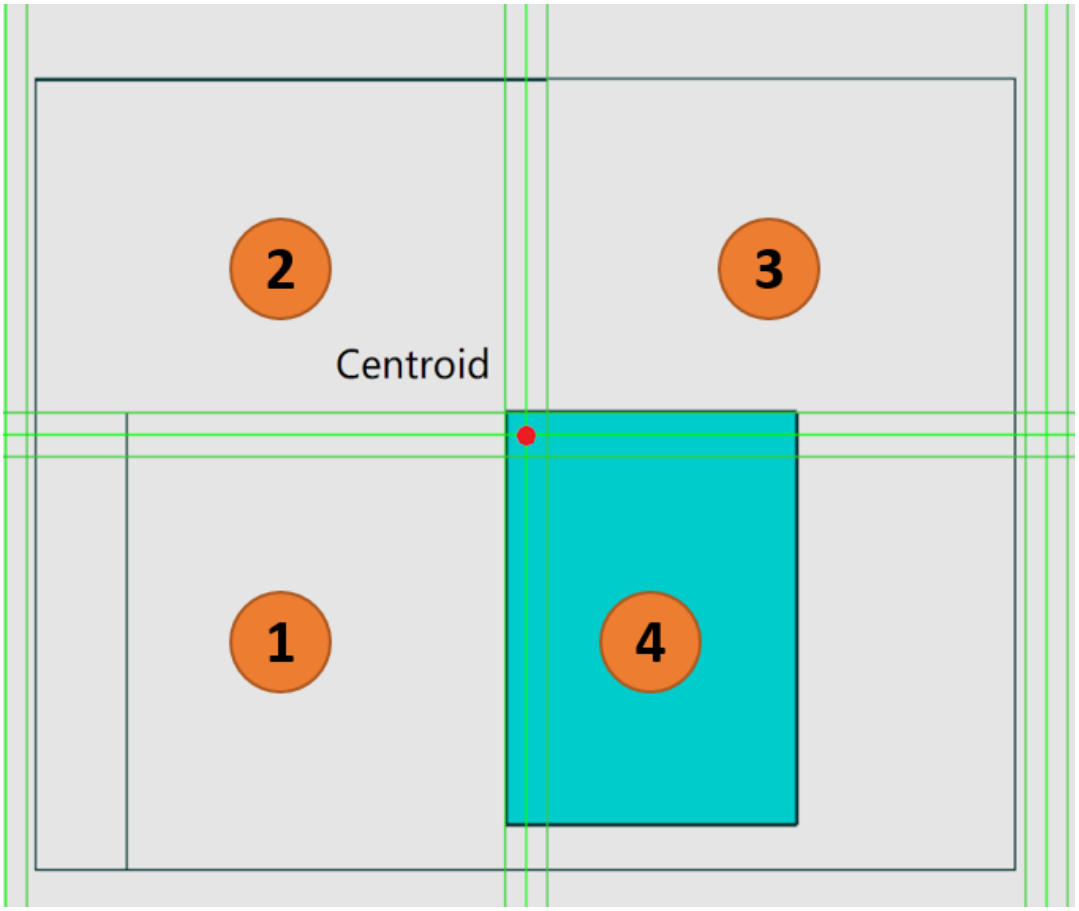


Figure 3-17: Cutting Strategy

3.5 Comparing Available Ground filters

After data preparation, we were ready to study the ability of four different systems to produce a DTM from our TLS datasets so they could each be compared with the more commonly used ALS version. As SAGA GIS is designed to process elevation raster rather than point clouds, we only discuss the performances of four ground filters mentioned in Chapter 2: ArcGIS, LASTools, PDAL, and MCC-LIDAR. For each tool, the parameters were chosen based on the performance on the TLS data. ALS data were then processed with the optimal parameters for each TLS for comparison in Chapter 4.

3.5.1 ArcGIS

A tool called “Classify LAS Ground” was employed in ArcGIS tools for ground point extraction. Unlike the other tools described above, this tool requires little parameter adjustment. There is only one parameter called “method” that may affect the accuracy of classification. There are three optional methods: standard, conservative, and aggressive. “Conservative” is designed for smooth terrain and “aggressive” for steep terrain. There are no obvious visual or statistical difference among the results of all the three parameter sets. However, the instruction of the tool suggests that among these three methods, “conservative” suits better in flat areas while “standard” and “aggressive” settings are more efficient in sharper-relief detection. As our research area is flat with smooth sloping stream banks, the “conservative” method was employed on both TLS data and ALS data.

3.5.2 LASTools

Lasground is one of the tools in LASTools, which was implemented and integrated into Lastools by Martin Isenburg (LASTools, 2014). The algorithm is based on the adaptive TIN model developed by Axelsson in 2000 (Axelsson, 2000). In this model, several parameters are required from the user. However, through experimentation, it appears that the user settings usually do not affect the accuracy of the result much, as they are automatically corrected by the statistical features of the points. The parameters are listed as below (https://github.com/LAStools/LAStools/blob/master/bin/lasground_new_README.txt):

Step: sample cell size of initial seed points; should be sufficiently large in order to remove large features

Sub: target intensity of triangulation; The larger the value the more detail and variation recorded in the TIN

Spike: a threshold of how far a ground point can protrude up or down from the plane generated from its neighbor nodes in the TIN.

Bulge: the initial threshold of the angle a candidate ground point to the closest TIN node from it; by default, is one tenth of the step size and then clamped to the interval [1.0, 2.0]

Stddev: describe how much a ground point can diverge from corresponding facet of the current ground TIN in centimeter. A non-zero value of this parameter enables the algorithm keep roads on the side of steep mountains instead of cutting it off by smoothing the mountain, but it can also lead to serious edge effects

Offset: a point can still be considered as ground point if its deviation along z axis from the current ground TIN is less than the offset value

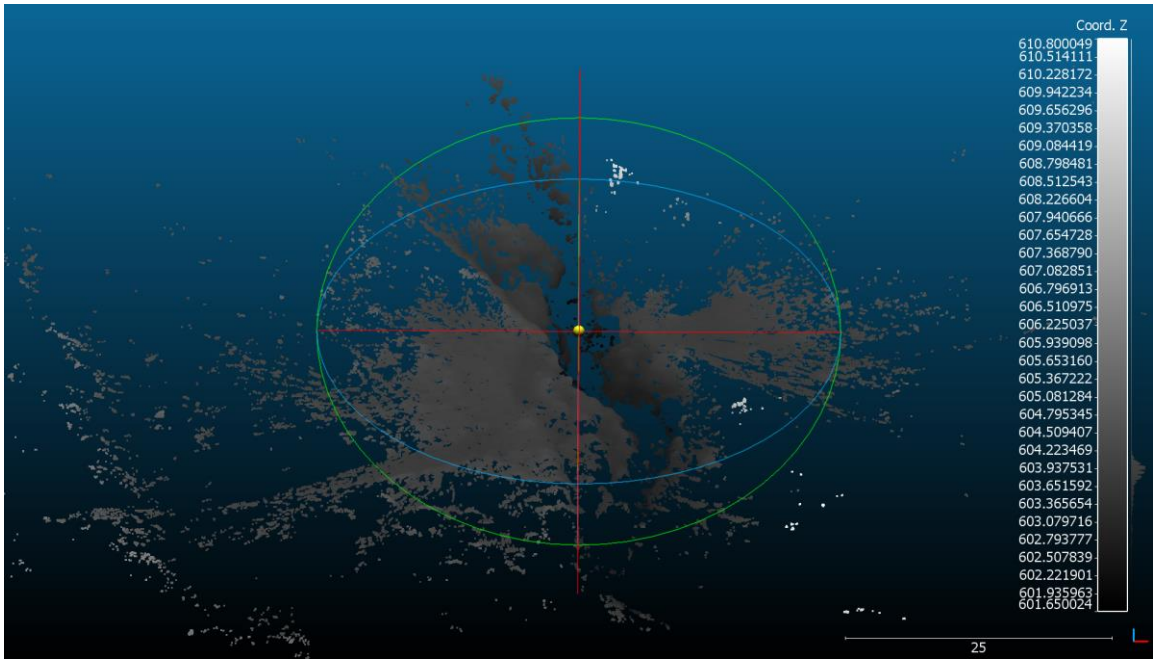
To simplify the procedure searching for suitable parameters for a specific dataset, several sets of pre-defined parameters, are provided by the software to assist users to customize the parameters suitable for their own research area's land cover types.

Table 3-2: Built-in parameter sets in lasground_new

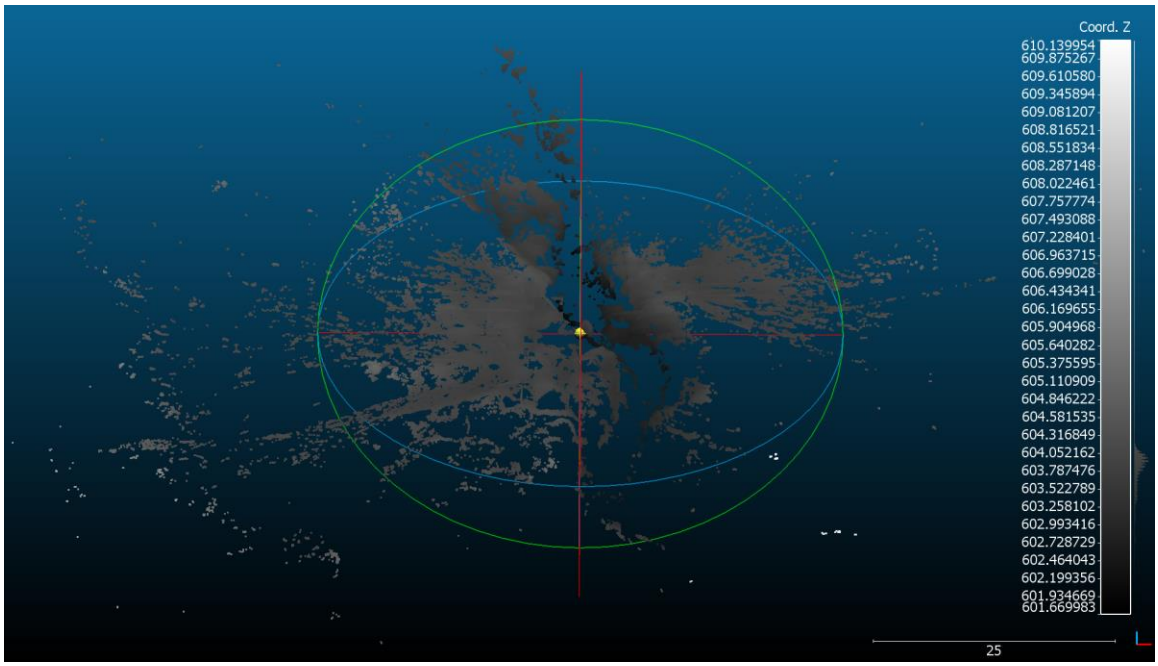
Parameter / Name	Step (m)	Sub (m)	Spike (m)	Bulge (m)	Stddev (cm)	Offset (m)
Archaeology	1.0	3.0	0.35/-3.5	0.2	1.0	0.02
Wilderness	3.0	4.0	1.0+10.0	0.6	10.0	0.05
Nature (default)	5.0	5.0	1.0+10.0	1.0	10.0	0.05
Town	10.0	6.0	1.0+1.5	1.0	10.0	0.05
City	25.0	7.0	1.0+1.5	2.5	10.0	0.05
Metro	50.0	8.0	1.0+1.5	5.0	10.0	0.05

Generally, the step should be a bit larger than the largest object in the research area, so that points on the objects will not be identified as seed points used to generate the initial ground TIN. At first, the Stroubles Creek was supposed to be the largest object, which is approximately 2.9 meters wide. However, the "wilderness" parameter set was found to be unable to identify trees at the edge of our research area, where ground data were rarely captured. To find the best parameter sets, "wilderness", "nature", "town", and "city" were all tested on the TLS data. The major difference between these parameter sets is an increment of "step size" and "sub" from "wilderness" to "city", which mainly determine if large objects are smoothed and set the tolerable surface variation. It can be visually identified that the ground generated from "town" and "city" are less completed than that of "nature" (Figure 3-18).

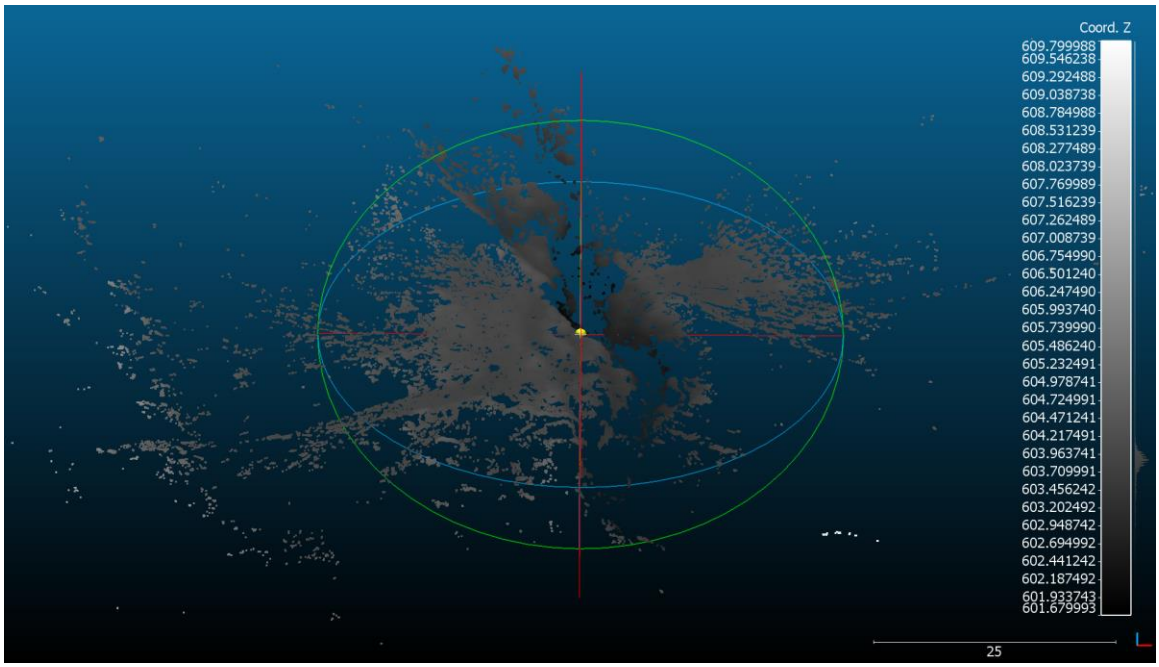
As LASTools claims that noise is added to point cloud with more than 1,500,000 points in unlicensed lasground_new, the algorithm can probably do better when sliced into pieces less than 1,500,000 points. Nevertheless, experiments showed that tiling the point cloud contributes nothing in improving the quality of classification result. According to all the tests listed above, the tiled dataset was discarded. Afterwards, different values of "Spike" and "Bulge" were tested based on the assumption that lower tolerance of variation leads to a smoother ground surface. The result showed no obvious increment or decrement of the quality of the classification. Therefore, default parameter values for "Nature" are suggested to be used in this case.



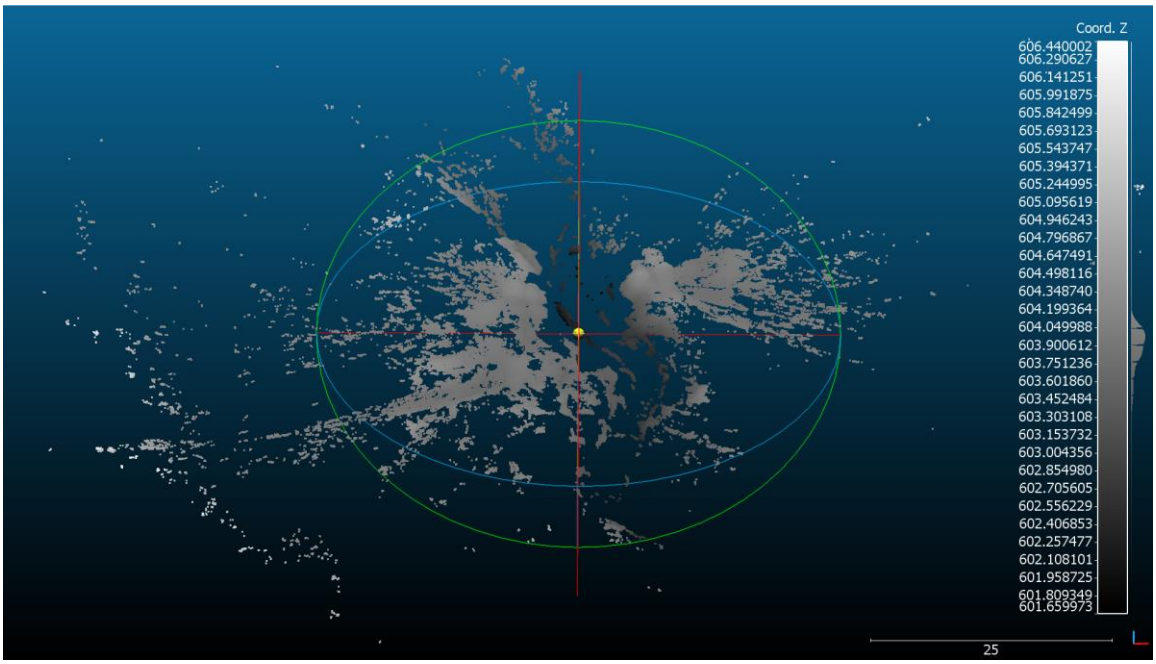
(a) wilderness (More high noises (white point))



(b) Nature (a few high noises (white point at the right bottom corner))



(c) Town (Fewer high noises (white point at right bottom corner); Detail lost on the bank (above the center point))



(d) City (No visible high noise; Detail lost on the banks)

Figure 3-18: Difference between results of different parameter sets

3.5.3 MCC-LIDAR

Multiscale curvature classification (MCC) was developed by Jeffrey in 2007 (Jeffrey, 2007) and employed in LiDAR data analytic tools, such as GRASS GIS (Geographic Resources Analysis Support System) and BCAL Lidar (Boise Center Aerospace Laboratory). In this research, a command-line application called MCC-LIDAR developed by the author's team (https://sourceforge.net/p/mcclidar/wiki/Home/) was employed.

The only two parameters necessary for executing MCC are the scale parameter "s" and curvature tolerance "t". For the scale parameter "s", the value is decided by both the size of objects and a formula base on the point spacing, approximately equal to the point spacing (sqrt(area/number of points)). For curvature tolerance "t", the official suggestion is to try 0.3 meters first and then vary it up or down by 0.1 meters for 1-foot resolution point cloud data.

In case of the TLS data, the point spacing increases from 0.002m to 0.4m with increasing distance from the scanning locations. According to the official suggestion, the value of "s" should fall between 0.002 and 0.4, but experiments showed that the application had trouble converging to a final result when "s" is too small (less than 0.03) or too big (greater than 0.4). Therefore, a series of values of "s" from 0.03 to 0.4 with an increment of 0.01 was tried. For parameter "t", the increment is hard coded as 0.1 in each iteration (see section 2.3.2), so the accuracy of the initial value does not need to be higher than 0. As our research area was pretty flat, 0.1, 0.2, and 0.3 for "t" were tested. The results indicated that a larger "s" (greater than or equal to 0.2) could filter more non-ground

points, while smaller “t” (t=0.1) could also improve the output. One explanation of these results is that a larger cell size can smooth down object points in sparser regions. On the other hand, from the equation used to calculate “t” in 2.3.2, the allowable slope is probably correlative to t/s, so that an s smaller than 0.1 can leads to a extremely large slope threshold.

3.5.4 PDAL

The Point Data Abstraction Library (PDAL) assembles tools in GDAL (Geospatial Data Abstraction Library) and other functions manipulating non-geographical point cloud data. The ground filter employed by PDAL is the Simple Morphological Filter (SMRF). This method, proposed by Pingel (Pingel, 2013), was tested on 15 datasets provided by the International Society of Photogrammetry and Remote Sensing (ISPRS).

Table 3-3: Features of the 15 test datasets provided by ISPRS (Pingel, 2013)

Site	Density (m^{-2})	Site	Sample	#	Features
Urban	0.67	1	1-1	1	Mixed vegetation and buildings on hillside
			1-2	2	Mixed vegetation and buildings
		2	2-1	3	Road and bridge
			2-2	4	Bridge and irregular ground surface
			2-3	5	Large, irregularly shaped buildings
			2-4	6	Steep slopes with vegetation
		3	3-1	7	Complex building
		4	4-1	8	Large gaps in data, irregularly shaped buildings
			4-2	9	Trains in railway yard
Rural	0.18	5	5-1	10	Data gaps, vegetation on moderate slopes
			5-2	11	Steep, terraced slopes
			5-3	12	Steep, terraced slopes
			5-4	13	Dense ground cover
		6	6-1	14	Large gap in data
		7	7-1	15	Underpass

Similar to Axelsson's method (Axelsson, 2000), SMRF requires more parameters from the users than the other ground filters such as the size of the grid cells, slope tolerance, maximum window size, elevation threshold for point classification, and scaling value. To assist using this tool, Pingel also recommend settings for different types of landcover. The cell size is suggested to be 1m regardless of the landcover type. For the other parameters, the combination that works best overall on all of the 15 datasets (the default settings in PDAL) is:

$slope\ tolerance\ =\ 15\%, maximum\ window\ radius\ (\mathrm{wk}_{max})\ =$

$18\ m, elevation\ threshold\ =\ 0.5\ m, and\ elevation\ scaling\ factor\ =\ 1.25m.$

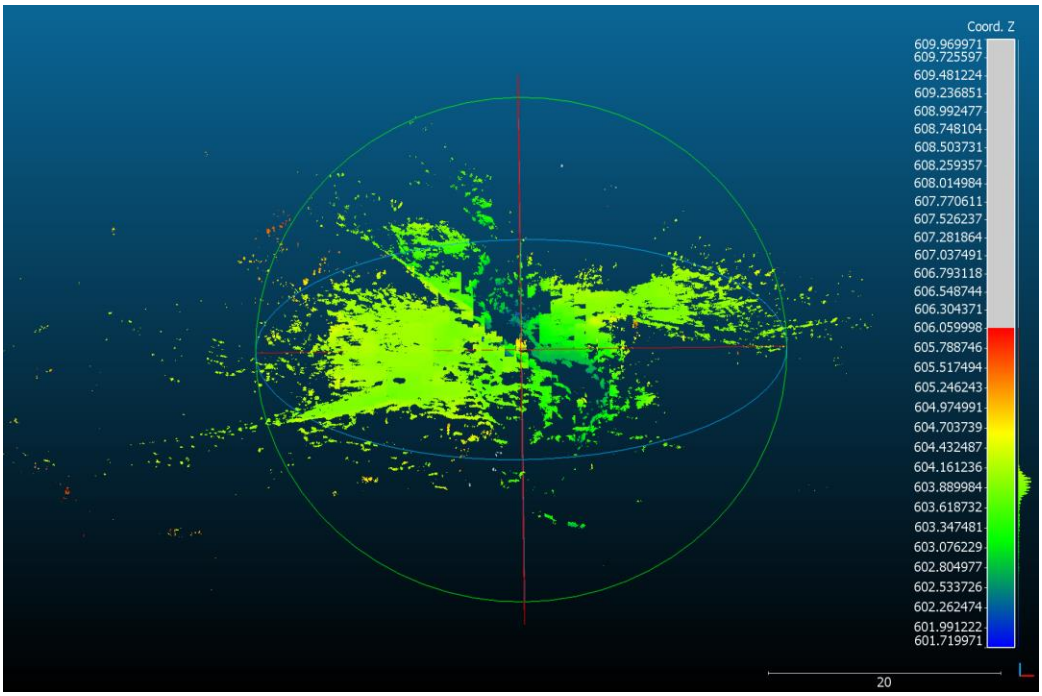
For individual dataset the best fitted parameters are listed in Table 3-4.

As our research area is of smooth terrain, the highlighted rows in Table 3-4 ("vegetation on moderate slopes" and "dense ground cover") are parameters that should do better on our data.

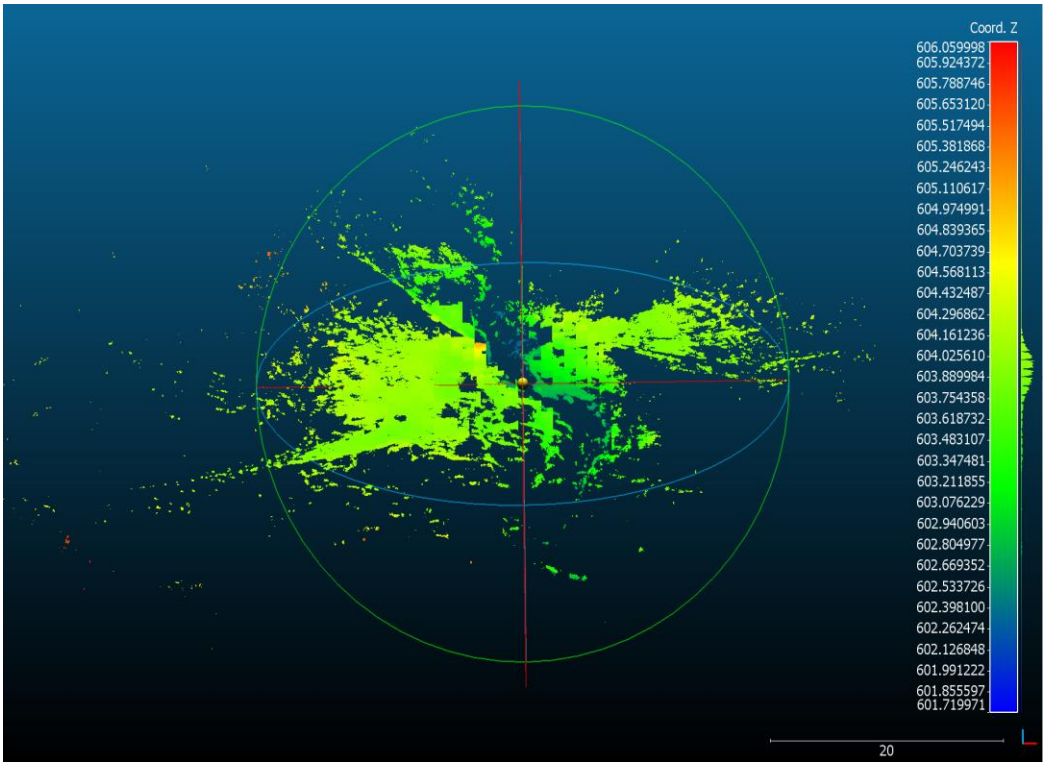
After several trials of different set of parameters, smaller "Et" and "Scalar" values were found suitable for flat terrain. Therefore, the one finally chosen was $Slope = 0.05, Wk_{max} = 18.0, Et = 0.15, and\ Scalar = 0.9$. Specifically, two different cells size were applied to the datasets: 1m and 0.1m, based on the assumption that smaller cells size could reveal more terrestrial details. By unifying the scale field of the results, it can be found that SMRF with 0.1m cell size performs worse in identifying marginal object points than 1m cell size, while 1m cell size SMRF is not able to distinguish a segment of bridge pillar from ground (Figure 3-19). 0.1m cell size was finally employed.

Table 3-4: examined parameter sets for different landscape types (Pingel, 2013)

Sample	Optimized			
	Slope	Wk_{max} (m)	Et (m)	$Scalar$
1 (1-1)	0.20	16	0.45	1.20
2 (1-2)	0.18	12	0.30	0.95
3 (2-1)	0.12	20	0.60	0.00
4 (2-2)	0.16	18	0.35	1.30
5 (2-3)	0.27	13	0.50	0.90
6 (2-4)	0.16	8	0.20	2.05
7 (3-1)	0.08	15	0.25	1.50
8 (4-1)	0.22	16	1.10	0.00
9 (4-2)	0.06	49	1.05	0.00
10 (5-1)	0.05	17	0.35	0.90
11 (5-2)	0.13	13	0.25	2.20
12 (5-3)	0.45	3	0.10	3.80
13 (5-4)	0.05	11	0.15	2.30
14 (6-1)	0.28	5	0.50	1.45
15 (7-1)	0.13	15	0.75	0.00



(a) PDAL cell size = 0.1m



(b) PDAL cell size = 1m

Figure 3-19: Difference between results of different cell size (0.1m and 1m)

Chapter 4 Conclusion

In this research, it is fairly hard to compare the accuracy of classification or different resolutions of point clouds. As discussed in section 2.3, there are three types of accuracy validation methods: visual validation, random sampling, and comparing to ground truth point cloud data. For visual validation, points from grass surfaces are very hard to distinguish from the true ground surface underneath because of the tiny size and low height of grass blades. Both random sampling and comparing to ground truth point cloud data require reference data which are very difficult to acquire for the resolution we desired. Although a coarser resolution standard reference point cloud, say that flown by the Virginia Geographic Information network (VGIN) can validate the overall curvature of TLS ground surface, it cannot prove that whether TLS provides higher resolution ground features or just low noise from grass.

To examine the accuracy of a millimeter resolution ground surface, millimeter resolution reference data are required, but not available. Therefore, this research chose to compare the TLS ground surface with a ground surface created from our ALS data, which is considered to be reliable by many previous research projects and is of far higher resolution than the VGIN imagery ranking second only to our TLS data.

4.1 Result

In Chapter 3, a set of parameters for each ground filter tool was determined based on the performance on TLS data. The same configurations were then applied to the ALS data to make fair comparisons. A summary of the results is listed in Table 4-1. A problem

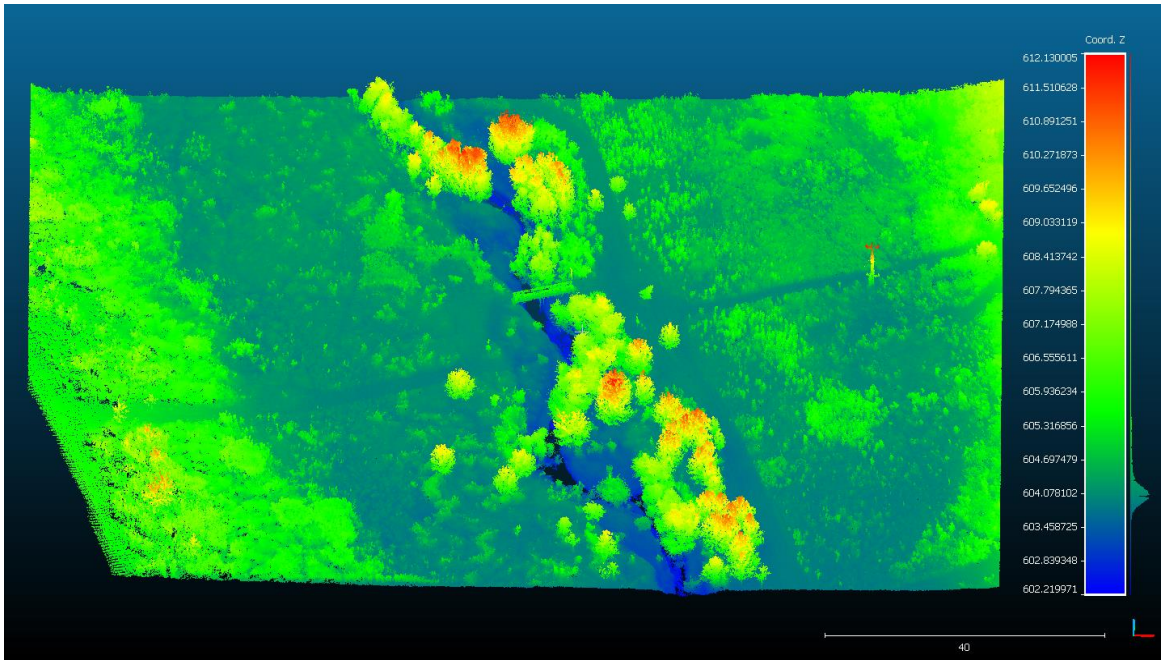
indicated by the average elevations in Table 4-1 is that there are approximately 0.3m ($|Avg_{elev_{TLS}} - Avg_{elev_{ALS}}|$) differences between TLS ground and ALS ground, which is much larger than the average "Cloud-to-Mesh Distance". The result is likely explained by the fact that most of the TLS points concentrate on the banks of Stroubles Creek which is the lowest elevation in the study area, lower than the areas further from the creek.

Although TLS has 2 million more points that ALS, it covers a rather smaller ground surface (Figure 4-1), which leads to a smaller number of points classified as ground points. The radial shape edges provide incomplete information of the ground surface, which can be improved a bit by some of the ground filters described in Chapter 2 and Chapter 3.

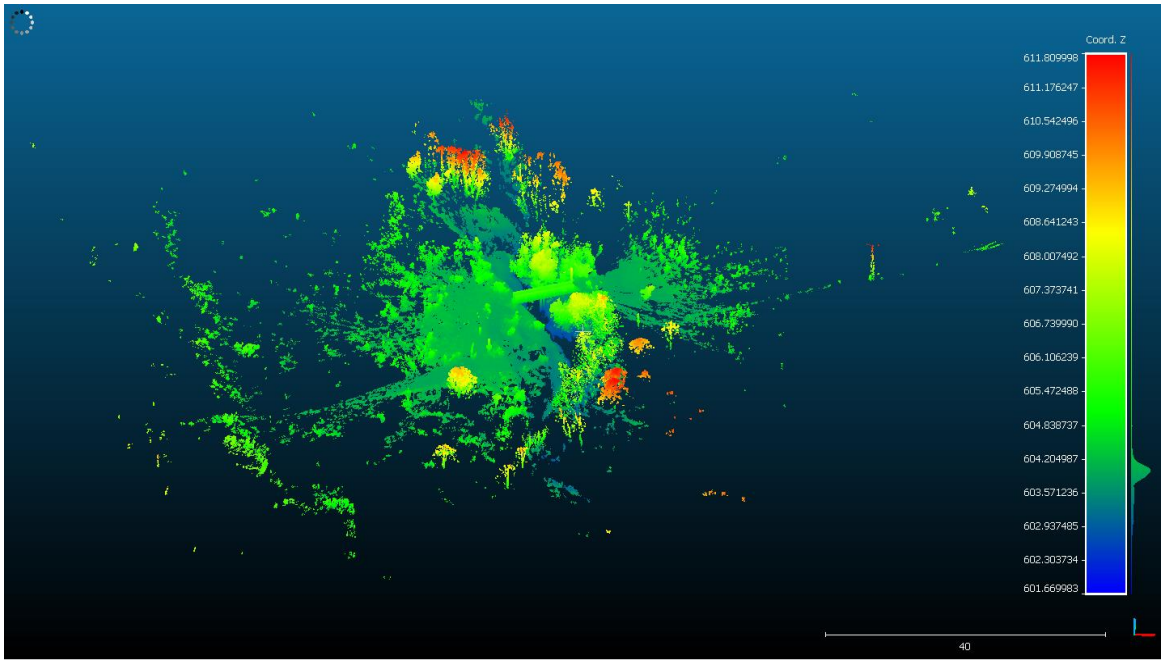
Visually comparing the ground surface generated from TLS and ALS, the TLS ground surface does provide more detail. However, as indicated at the beginning of this chapter, it is hard to identify whether those details come from high resolution or just noise caused by grass. What is to be discussed below is whether TLS can provide a ground surface that matches that from the ALS.

Table 4-1: Basic statistics of ground filter results

		# of Points	Elev Avg (m)	Elev Std (m)	Elev Min (m)	Elev Max (m)
Source Data	TLS	10,463,811	604.023	0.841	601.670	607.810
	ALS	8,208,349	604.472	0.961	602.220	612.130
ArcGIS	TLS	1,239,010	603.759	0.354	601.810	605.570
	ALS	2,489,486	604.094	0.446	602.220	607.880
LASTools	TLS	1,015,027	603.681	0.396	601.680	609.800
	ALS	1,295,851	604.075	0.521	602.200	607.180
PDAL	TLS	1,894,097	603.767	0.380	601.720	606.060
	ALS	3,680,706	604.088	0.464	602.280	607.220
MCC	TLS	1,946,720	603.815	0.483	601.670	610.950
	ALS	3,446,468	604.122	0.495	602.220	608.420



(a) Source data of ALS



(b) Source data of TLS

Figure 4-1: Source data of ALS and TLS

Table 4-2: Cloud-to-Mesh Distance Summary

	MCC	**PDAL**	**ArcGIS**	**LASTools**
min (m)	0.000	0.000	0.000	0.000
max (m)	6.993	1.724	1.491	6.099
mean (m)	0.059	0.056	0.056	0.060
stddev (m)	0.202	0.093	0.059	0.097
% point (d<0.1m)	92.5	88.7	89.1	89.9

To evaluate quantitatively the similarity between ALS ground and TLS ground for different ground filters, the "distance" between each pair of ground point clouds needed to be calculated. The traditional distance calculation method is "Cloud-to-Cloud Distance", which is the distance from a point to its nearest neighbor in the other point cloud. However, as our ALS is sparser than TLS, "Cloud-to-Cloud Distance" overestimates the vertical distance between the two point clouds. Therefore, "Cloud-to-Mesh Distance" was employed in this case. For each different ground filter, a mesh was first generated from the ALS, and then the vertical distance from each point in the TLS to the ALS mesh was calculated. A summary of the "Cloud-to-Mesh Distance" is listed in Table 4-2.

All of the ground filters are able to keep the mean distance between ALS ground and TLS ground under 6cm. On the other hand, the max distances are quite large (up to 7m) indicating that there is at least one non-ground point wrongly classified as a ground point. As all the standard deviations are excessively large compared to the corresponding mean values, it is reasonable to worry that many of the values diverge far from the mean value. However, the distances are not normally distributed in this case as absolute distances are calculated instead of signed distances. Therefore, we counted points diverging less than 0.1m from ALS mesh and divided them by the total number of points respectively. The result shows that approximately 90% of the points are closer than 0.1m to reference meshes (Table 4-2), which shows that the overall result is still quite good.

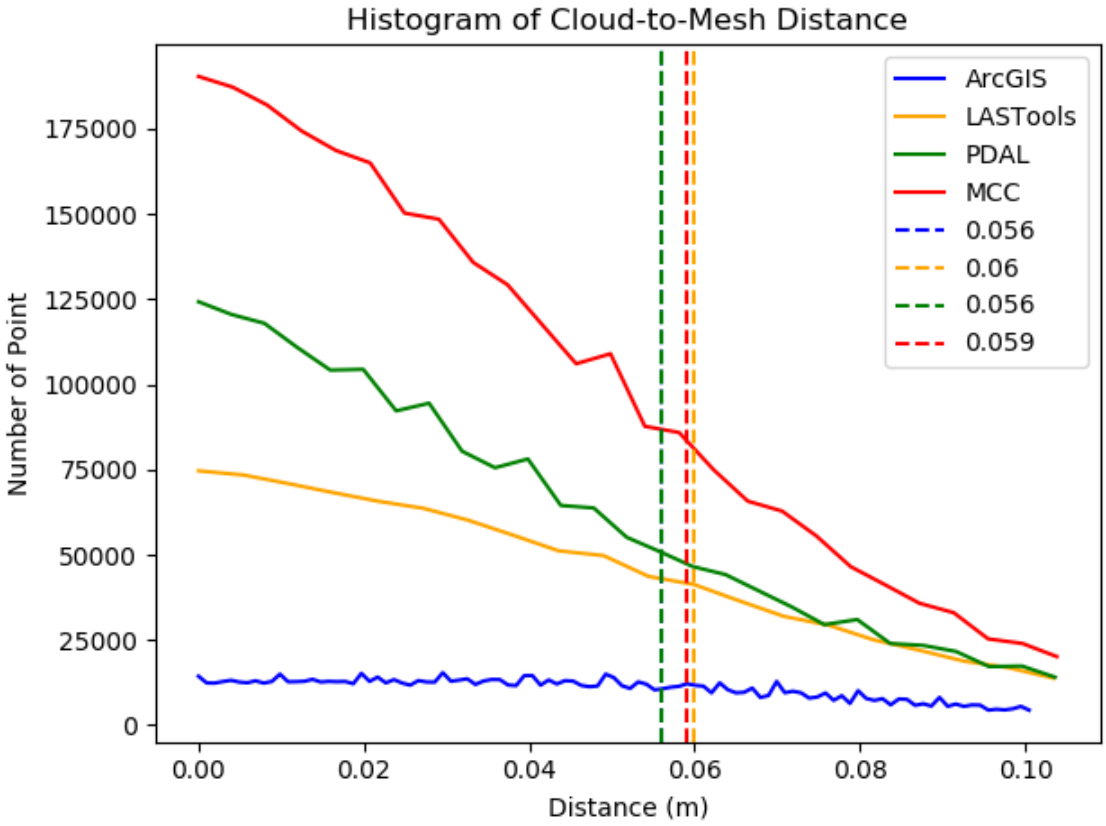
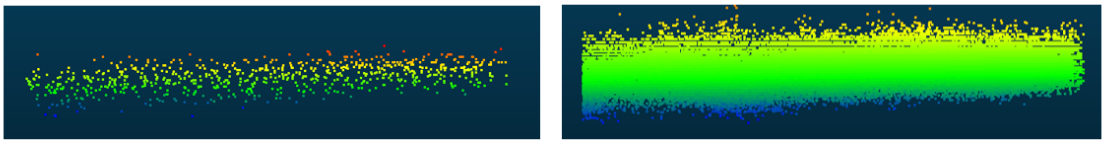


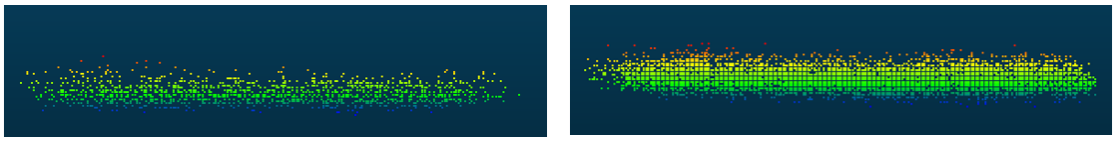
Figure 4-2: Histogram of Cloud-to-Mesh Distance (0.0~0.1m)

To better illustrate the distribution of the "Cloud-to-Mesh Distance" value and actual coverage of TLS ground surface, a mesh was generated from TLS ground, which is colored by "Cloud-to-Mesh Distance" in meters. The corresponding ALS ground is shown in white as reference.

Figure 4-4 indicates that the ground surface generated from the TLS fluctuates along the ALS ground surface. The TLS ground near the central research area tends to be lower than the ALS ground points, while higher than ALS in the outer research areas. This situation most likely results from variance of penetration of the TLS. For reflection points close to the TLS scanner, the laser beam is more powerful and able to penetrate more grass than the beam from the more distant ALS laser (Figure 4-3 (a)). As the distance to the TLS scanner increases, the penetrability tends to match the ALS (Figure 4-3 (b)). All of the ground classification algorithms employed in this research are based on the assumption that the lowest points within cells of a relatively sparse grid (cell size larger than the largest object) must belong to the ground and points can be classified as ground points if they are close enough to the surface formed by these lowest points. Because of this nature of these algorithms, it is hard for them to distinguish grass while identifying the ground curvature at the same time. As the algorithms got lower "lowest points" from the TLS than from the ALS in the central research area, the ground surface generated from the TLS also tends to be lower than ALS. That is the major cause of 6cm-error in Table 4-2.



(a) Penetrability for grass of ALS (left) and TLS (right) close to TLS scanner



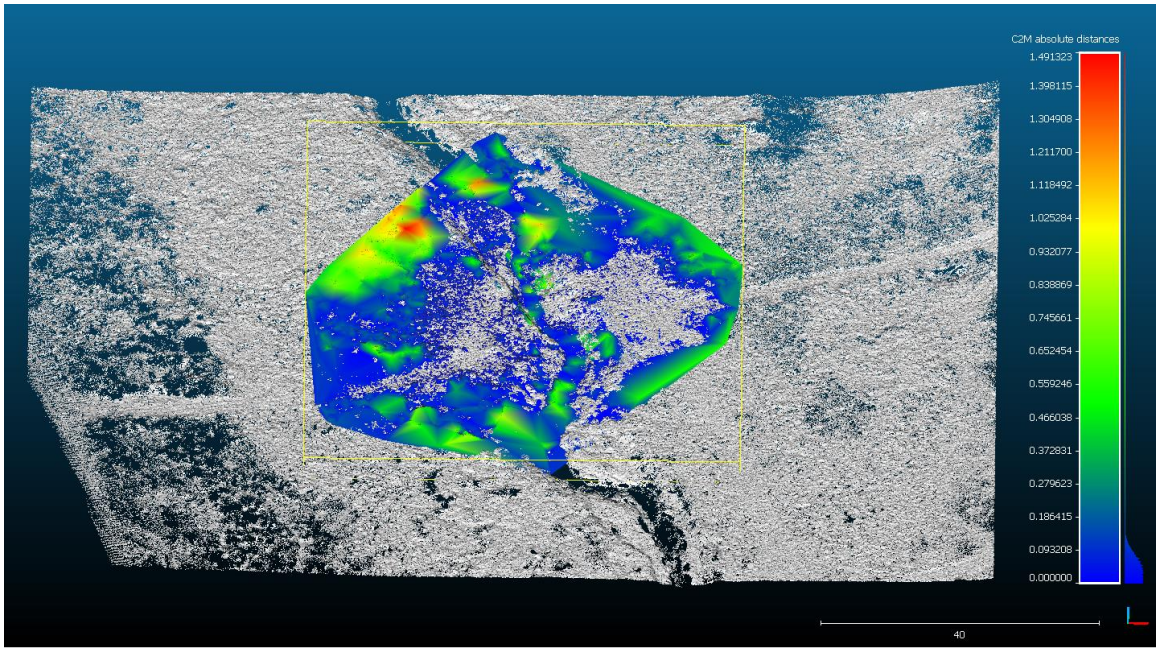
(b) Penetrability for grass of ALS (left) and TLS (right) far from TLS scanner

Figure 4-3: Penetrability for grass of ALS and TLS of different distance from TLS scanner

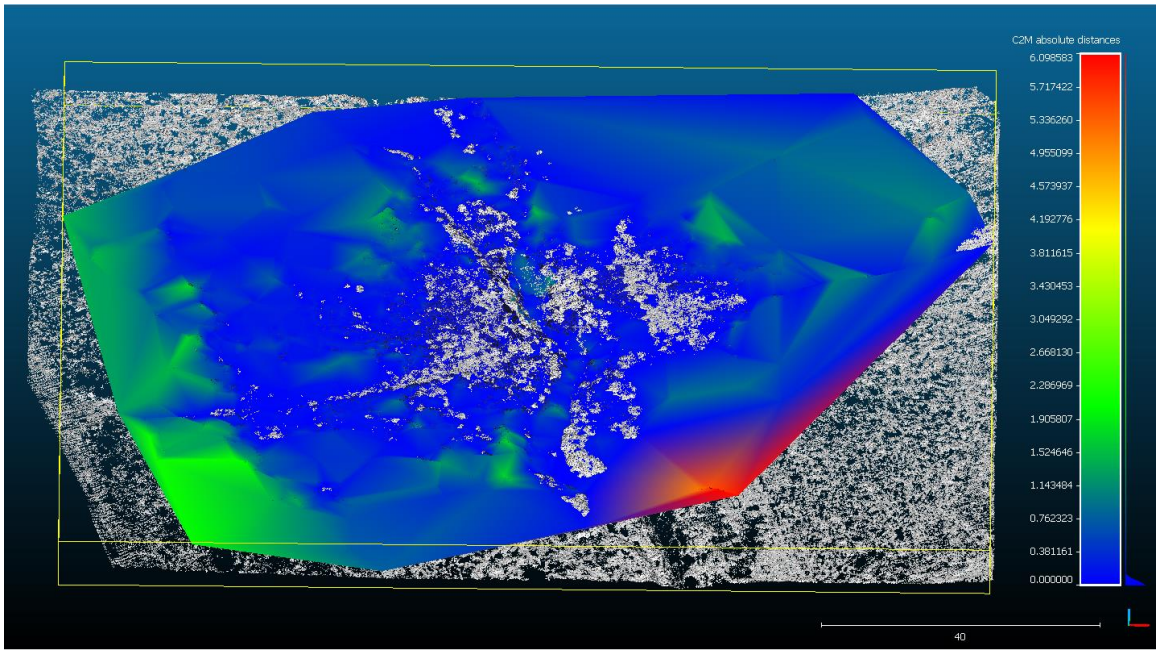
For quantitative validation, Table 4-2 shows that ArcGIS produces the smallest mean and standard deviation difference between the TLS ground and the ALS ground and PDAL is the second, while their ground point sets contain more low noise (absolute distance > 0.1m). On the other hand, MCC and LASTools were more likely to wrongly classify high object points in the sparse areas as ground points (Table 4-1).

For visual validation, according to Figure 4-4, all of the ground filters could generate a reliable ground surface from reasonably dense TLS data. ArcGIS and PDAL could automatically eliminate the sparse parts of the TLS to avoid large deviations, while MCC and LASTools were able to identify ground points from sparse area in the point clouds, which introduce high offsets from the true surface. Based on Figure 4-4, it can be concluded that the major bias from the ALS ground surface is caused by sparsity and incompleteness of point cloud data at the edges of the research area. However, MCC produces relatively high noise even in the area with sufficient point density (along the creek), and thus may not be a good indicator of the TLS ground surface. As for the other three tools, ArcGIS, PDAL and LASTools each has its advantages. ArcGIS and PDAL produce a TLS ground that is close to the ALS ground surface, while ArcGIS is too sensitive to the decrement of point density (generated the smallest effective ground coverage) and PDAL shows a discontinuous TLS ground pattern in Figure 3-19. LASTools provides the largest effective ground coverage among the three algorithms, while it tends to include more points in the sparse area, which potentially provides false information for geomorphology research. Overall, traditional ground filters designed for ALS processing can also perform well on TLS data, but to acquire a reliable ground

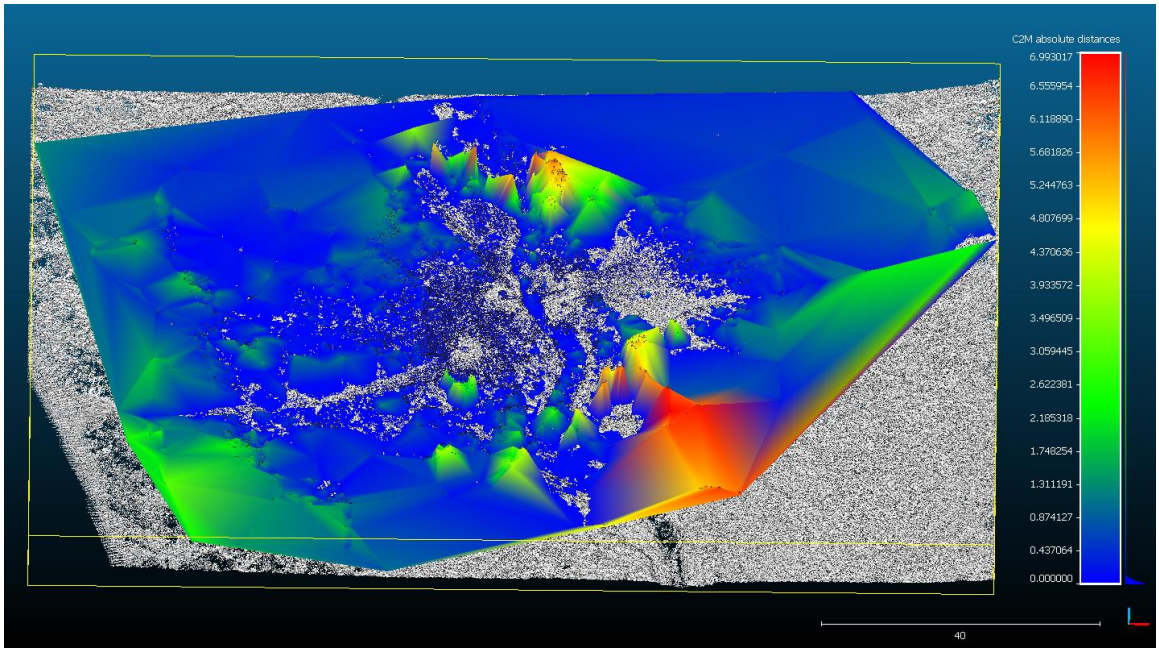
surface, a reasonable point density is necessary is all portions of a study area. However, for one who's not very confident of the point density of the TLS data, ArcGIS and PDAL might be better choices as they won't produce large morphological errors and could be a good indicator of how to complete the coverage in the next step.



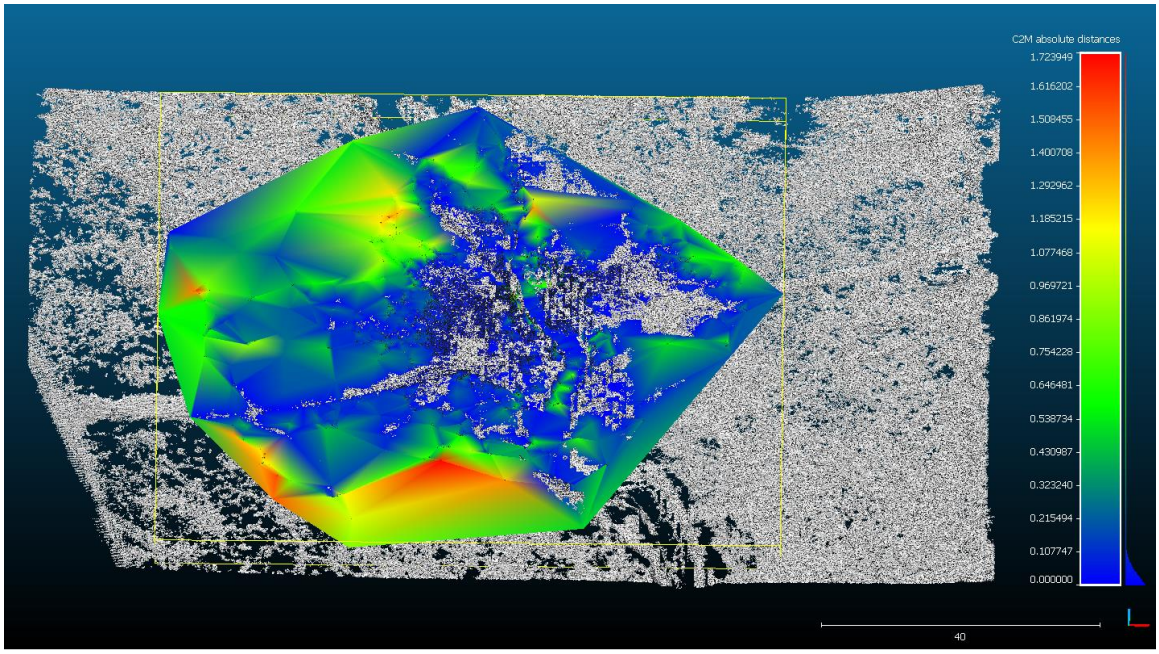
(a) ArcGIS



(b) LASTools



(c) MCC



(d) PDAL

Figure 4-4: Difference between results of the four ground filters

As complete coverage was not the major concern in this research, six scans were used to generate reasonable data for the intent of this study. To estimate the approximate number of scans to cover the enough ground in the research area, the distance from each scan to the nearest edge of the ground mesh outline was calculated. ArcGIS Pro ground mesh was employed in this case, as it provided the smallest ground coverage. Figure 4-4 indicates that the effective distances range from 20 to 27m. Considering the research area is flat without many objects higher than 1m, about 12 total scans would have been needed to complete the coverage in this $100\text{m} \times 140\text{m}$ area. As each ground scan requires about 10 minutes, 2 hours would have been sufficient to do a more complete coverage. Our results in the areas that did have good TLS density show that this technique is quite promising for small study areas and also for enhancing ALS surveys in areas with over story blockages.

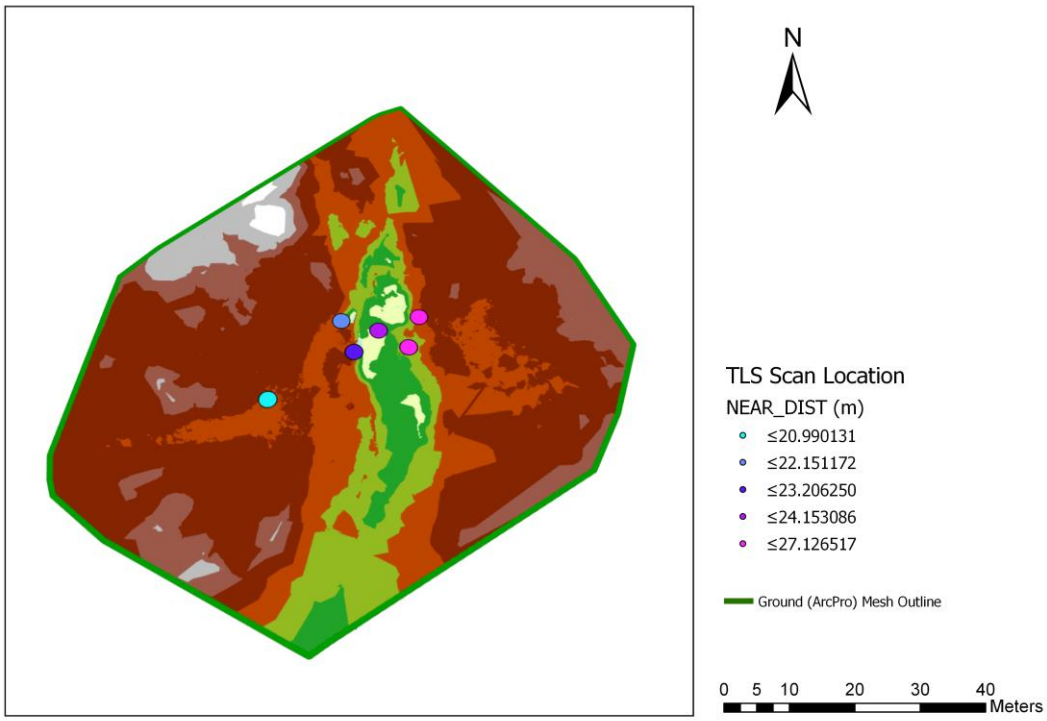


Figure 4-5: Distances from the scan positions to the nearest outline of ArcGIS Pro ground mesh

4.2 Future work

With the limitation of time and devices, we could not test all the possible parameter sets, which might lead to an underestimate of each algorithm's performance. For example, in LASTools, only the default parameter sets were tested on our TLS and ALS data. We tried to change several parameters, but the classification results showed either no obvious changes or quality declined. A systematic summary of how each parameter affect the result could be useful for understanding the algorithms and the data.

The validation method was not perfect either. With the lack of standard reference data, we examined the quality of a TLS ground point set with a standard similar to Type II error, showing how many non-ground points were classified as ground points. A method should be added to indicate "false positive" (Type I error) as well, that is, how many ground points were falsely classified as non-ground points. One possible complementarity to our validation method is to take a fixed number of points from the "best" part (points closest to ALS ground surface) of the result of each algorithm, so that the percentage and coverage of the TLS ground points can be comparable.

In the future, more ground filter algorithms should be evaluated and improvements should be made to the algorithms tested in this research. As indicated in the last section, despite the prevalence of low noise, all these algorithms are based on a surface generated from the lowest points. The methods have proved effective and the assumption that the TLS is quite comparable to the TLS is true in most of the situations. However, there can

also be innovative methods based on information other than only the morphological features of points.

Furthermore, an inter-visibility study of the initial scans would lead to a GIS model of optimal scanning locations to generate complete coverage at a point density found to make a promising TLS ground surface. As indicated in the former section, the effective ground coverage is actually rather smaller than the theoretical ranging distances of TLS scanners even for flat terrain. A good estimation of the location and coverage of each scanning site is really helpful in avoiding revisiting the research area to fill in gaps.

We are also interested in the relationship between the features of ground surface produced by point cloud and penetration variations of TLS. Although the penetration issue was used to interpret the 6cm-error in the results, evidence needs to be provided to prove this assumption in our research.

Reference

[1] Anderson, K. E., Glenn, N. F., Spaete, L. P., Shinneman, D. J., Pilliod, D. S., Arkle, R. S., ... & Derryberry, D. R. (2017). Methodological considerations of terrestrial laser scanning for vegetation monitoring in the sagebrush steppe. Environmental monitoring and assessment, 189(11), 578.

[2] Axelsson, P. (1999). Processing of laser scanner data—algorithms and applications. ISPRS Journal of Photogrammetry and Remote Sensing, 54(2-3), 138-147.

[3] Bechet, J., Duc, J., Jaboyedoff, M., Loye, A., & Mathys, N. (2015). Erosion processes in black marl soils at the millimetre scale: preliminary insights from an analogous model. Hydrology and Earth System Sciences, 19(4), 1849.

[4] Becerik-Gerber, B., Jazizadeh, F., Kavulya, G., & Calis, G. (2011). Assessment of target types and layouts in 3D laser scanning for registration accuracy. Automation in Construction, 20(5), 649-658.

[5] Besl, P. J., & McKay, N. D. (1992, April). Method for registration of 3-D shapes. In Sensor Fusion IV: Control Paradigms and Data Structures (Vol. 1611, pp. 586-607). International Society for Optics and Photonics.

[6] Błaszczak-Bąk, W., Janowski, A., Kamiński, W., & Rapiński, J. (2015). Application of the Msplit method for filtering airborne laser scanning data-sets to estimate digital terrain models. International Journal of Remote Sensing, 36(9), 2421-2437.

[7] Briese, C.; Pfeifer. N. Airborne laser scanning and derivation of digital terrain models. In Proceedings of Fifth Conference on Optical 3-D Measurement Techniques, Vienna, Austria, 2001.

[8] Clawges, R., Vierling, L., Calhoon, M., & Toomey, M. (2007). Use of a ground-based scanning lidar for estimation of biophysical properties of western larch (Larix occidentalis). International Journal of Remote Sensing, 28(19), 4331-4344.

[9] CloudCompare (version 2.10) [GPL software]. (2017). Retrieved from http://www.cloudcompare.org/

[10] Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V., and Böhner, J. (2015): System for Automated Geoscientific Analyses (SAGA) v. 2.1.4, Geosci. Model Dev., 8, 1991-2007, doi:10.5194/gmd-8-1991-2015.

[11] De Agostino, M., Lingua, A., & Piras, M. (2012). Rock face surveys using a LiDAR MMS. Italian Journal of Remote Sensing/Rivista Italiana di Telerilevamento, 44(1).

Evans, Jeffrey S.; Hudak, Andrew T. 2007. A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments. Geoscience and Remote Sensing. 45(4): 1029-1038.

[12] Eltner, A., & Baumgart, P. (2015). Accuracy constraints of terrestrial Lidar data for soil erosion measurement: Application to a Mediterranean field plot. Geomorphology, 245, 243-254.

[13] Fiocco, M., Bostrom, G., Gonçalves, J. G., & Sequeira, V. (2005, June). Multisensor fusion for volumetric reconstruction of large outdoor areas. In 3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on (pp. 47-54). IEEE.

[14] Hopkinson, C., Chasmer, L., Young-Pow, C., & Treitz, P. (2004). Assessing forest metrics with a ground-based scanning lidar. Canadian Journal of Forest Research, 34(3), 573-583.

[15] Isenburg, M, lasground_new_readme, https://www.cs.unc.edu/~isenburg/lastools/download/lasground_new_README.txt

[16] Kraus, K.; Rieger, W. Processing of laser scanning data for wooded areas. In Photogrammetric Week‟99, Fritsch, D., Spiller, R., Eds.; Wichmann Verlag: Stuttgart, Germany, 1999; pp. 221-231.

[17] Kraus, K.; Pfeifer, N. Advanced DTM generation from LiDAR data. In Proceedings of the ISPRS Workshop on Land Surface Mapping and Characterization Using Laser Altimetry; Hofton, M.A., Ed.; Annapolis, MD, USA, 2001.

[18] Langerwisch, M., Krämer, M. S., Kuhnert, K. D., & Wagner, B. (2016). Construction of 3D Environment Models by Fusing Ground and Aerial Lidar Point Cloud Data. In Intelligent Autonomous Systems 13 (pp. 473-485). Springer, Cham.

[19] LAS Specification, Jul. 2013, [online] Available: http://asprs.org/a/society/committees/standards/LAS_1_4_r13.pdf.

[20] Maas, H. G., Bienert, A., Scheller, S., & Keane, E. (2008). Automatic forest inventory parameter determination from terrestrial laser scanner data. International journal of remote sensing, 29(5), 1579-1593.

[21] Meng, X., Wang, L., Silván-Cárdenas, J. L., & Currit, N. (2009). A multi-directional ground filtering algorithm for airborne LIDAR. ISPRS Journal of Photogrammetry and Remote Sensing, 64(1), 117-124.

[22] Milenković, M., Pfeifer, N., & Glira, P. (2015). Applying terrestrial laser scanning for soil surface roughness assessment. Remote Sensing, 7(2), 2007-2045.

[23] M. Isenburg, "LAStools - efficient LiDAR processing software" (version 141017, unlicensed)

[24] PDAL contributors (2018). PDAL: The Point Data Abstraction Library URL https://pdal.io/.

Mongus, D., & Žalik, B. (2012). Parameter-free ground filtering of LiDAR data for automatic DTM generation. ISPRS Journal of Photogrammetry and Remote Sensing, 67, 1-12.

[25] Shan, J.; Sampath, A. (2005). Urban DEM generation from raw LiDAR data: a labeling algorithm and its performance. Photogramm. Eng. Remote Sens. 71, 217-226.

[26] Pingel, T. J., Clarke, K. C., & McBride, W. A. (2013). An improved simple morphological filter for the terrain classification of airborne LIDAR data. ISPRS Journal of Photogrammetry and Remote Sensing, 77, 21-30.

[27] Perroy, R. L., Bookhagen, B., Asner, G. P., & Chadwick, O. A. (2010). Comparison of gully erosion estimates using airborne and ground-based LiDAR on Santa Cruz Island, California. Geomorphology, 118(3-4), 288-300.

[28] Pirotti, F., Guarnieri, A., & Vettore, A. (2013). State of the art of ground and aerial laser scanning technologies for high-resolution topography of the earth surface. European Journal of Remote Sensing, 46(1), 66-78.

[29] Riaño, D., Valladares, F., Condés, S., & Chuvieco, E. (2004). Estimation of leaf area index and covered ground from airborne laser scanner (Lidar) in two contrasting forests. Agricultural and Forest Meteorology, 124(3-4), 269-275.

[30] Rusu, R. B. (2010). Semantic 3D object maps for everyday manipulation in human living environments. KI-Künstliche Intelligenz, 24(4), 345-348.

[31] Schmid, T., Schack-Kirchner, H., & Hildebrand, E. (2004, October). A case study of terrestrial laser scanning in erosion research: calculation of roughness and volume

balance at a logged forest site. In Proceedings of the ISPRS Working Group VIII/2: Laser-Scanners for Forest and Landscape Assessment (pp. 03-06).

[32] SHI, Y., CHENG, X., & JIA, D. (2012). Applications of 3D Laser Scanning Based Tree Model to Forestry [J]. Bulletin of Surveying and Mapping, 3.

[33] Sithole, G., & Vosselman, G. (2003). Report: ISPRS comparison of filters. ISPRS commission III, working group, 3.

[34] Su, W., Sun, Z., Zhong, R., Huang, J., Li, M., Zhu, J., ... & Zhu, D. (2015). A new hierarchical moving curve-fitting algorithm for filtering lidar data for automatic DTM generation. International Journal of Remote Sensing, 36(14), 3616-3635.

[35] Tulldahl, H. M., Bissmarck, F., Larsson, H., Grönwall, C., & Tolt, G. (2015, October). Accuracy evaluation of 3D lidar data from small UAV. In Electro-Optical Remote Sensing, Photonic Technologies, and Applications IX (Vol. 9649, p. 964903). International Society for Optics and Photonics.

[36] Wehr, A., & Lohr, U. (1999). Airborne laser scanning—an introduction and overview. ISPRS Journal of photogrammetry and remote sensing, 54(2-3), 68-82.

[37] Wei, L., Yang, B., Jiang, J., Cao, G., & Wu, M. (2017). Vegetation filtering algorithm for UAV-borne LiDAR point clouds: A case study in the middle-lower Yangtze River riparian zone. International Journal of Remote Sensing, 38(8-10), 2991-3002.

[38] Yang, B., Dong, Z., Liang, F., & Liu, Y. (2016). Automatic registration of large-scale urban scene point clouds based on semantic feature points. ISPRS Journal of Photogrammetry and Remote Sensing, 113, 43-58.

[39] Zhang, K.; Whitman, D. Comparison of three algorithms for filtering airborne LiDAR data. Photogramm. Eng. Remote Sens. 2005, 71, 313-324. 39.

[40] Zhang, K.; Chen, S.; Whitman, D.; Shyu, M. A progressive morphological filter for removing nonground measurements from airborne LiDAR data. IEEE Trans. Geosci. Remote Sens. 2003, 41, 872-882.