A Hybrid Model for DTM generation from LIDAR Data

Hyun Seung Lee

Follow this and additional works at: https://scholarsjunction.msstate.edu/td

Recommended Citation
https://scholarsjunction.msstate.edu/td/159

This Dissertation is brought to you for free and open access by the Theses and Dissertations at Scholars Junction. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholars Junction. For more information, please contact scholcomm@msstate.libanswers.com.
A HYBRID MODEL FOR DTM GENERATION FROM LIDAR DATA

By

Hyun Seung Lee

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2004
A HYBRID MODEL FOR DTM GENERATION FROM LIDAR DATA

By

Hyun Seung Lee

Approved:

Dr. Nicolas H. Younan
Professor of
Electrical and Computer Engineering
(Major Advisor, Dissertation Director, and
Graduate Coordinator)

Dr. Roger King
William L. Giles, Distinguished Professor of
Electrical and Computer Engineering
(Committee Member)

Dr. Lori Bruce
Associate Professor of
Electrical and Computer Engineering
(Committee Member)

Dr. David Evans
Professor of Forestry
(Committee Member)

Dr. Robert Taylor
Interim Dean, Bagley College of Engineering
This dissertation introduces an innovative technique to extract ground elevation models using small-footprint LIDAR data. This technique consists of a preprocessing step, ground modeling, and interpolation. In the preprocessing step, much of the non-terrain points are eliminated using a histogram-based clustering technique. Then, in the ground modeling stage, the information such as elevation and slope between nearest neighbor points is extracted. This step corresponds to an outlier detection process. In this stage, residuals and gradient indices for elevation and slope, are introduced. These indices are investigated for a constructed 95% confidence interval to discard the remaining non-terrain points. Finally, using spline interpolation, a smooth ground surface is generated. Simulation results show a significant improvement over existing techniques. Furthermore, the possibility of using a ground trend model developed from specific tree height measurements is investigated. For this analysis, a statistical regression analysis
model is used. Performing this analysis, a 0.63 R-squared value is obtained. This result indicates that the LIDAR ground surface obtained from the presented algorithm is related to the true ground surface.
DEDICATION

I would like to dedicate this research to my parents, my sister, and my lovely wife Heejin.
ACKNOWLEDGMENTS

I would like to thank my major advisor, Dr. Nicolas Younan, for his support and help that finally lead to the successful completion of this dissertation. I would like also to thank my committee members, Dr. Roger King, Dr. Lori Bruce, and Dr. David Evans for their encouragement and support.

All this would not have been possible without the constant encouragement and support provided by my parents and my sister throughout my studies. I would like to especially thank my wife for encouraging and supporting me.

I would like also to thank the department of Electrical and Computer Engineering - Mississippi State University and the Engineering Research Center – Computational GeoSpatial Technologies Center for supporting this research.

Finally, I would like to acknowledge EarthData for providing test data sets for this study.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Dissertation Contributions</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Structure of the Dissertation</td>
<td>8</td>
</tr>
<tr>
<td>II. RELATED WORKS</td>
<td>9</td>
</tr>
<tr>
<td>III. HISTOGRAM BASED CLUSTERING</td>
<td>14</td>
</tr>
<tr>
<td>3.1 The Purpose of the Clustering Stage</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Distance Based Clustering</td>
<td>15</td>
</tr>
<tr>
<td>3.3 Probabilistic Clustering</td>
<td>17</td>
</tr>
<tr>
<td>3.4 Neural Networks Approach</td>
<td>18</td>
</tr>
<tr>
<td>3.5 Histogram Based Clustering Algorithm</td>
<td>19</td>
</tr>
<tr>
<td>3.6 Summary</td>
<td>23</td>
</tr>
<tr>
<td>IV. GEOMETRIC MODELING</td>
<td>25</td>
</tr>
<tr>
<td>4.1 Outlier Detection</td>
<td>25</td>
</tr>
<tr>
<td>4.2 Confidence Intervals</td>
<td>32</td>
</tr>
<tr>
<td>4.3 DTM Error Correction Algorithm</td>
<td>34</td>
</tr>
<tr>
<td>4.3.1 Elevation Outlier Detection</td>
<td>34</td>
</tr>
<tr>
<td>4.3.2 Slope Outlier Detection Algorithm</td>
<td>39</td>
</tr>
<tr>
<td>4.3.3 Verification Stage</td>
<td>40</td>
</tr>
</tbody>
</table>
# Table of Contents

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4 Truncation Method for Outliers Detection</td>
<td>41</td>
</tr>
<tr>
<td>4.5 Summary</td>
<td>45</td>
</tr>
<tr>
<td>V. SPATIAL INTERPOLATION</td>
<td>46</td>
</tr>
<tr>
<td>5.1 The Purpose of Spatial Interpolation</td>
<td>46</td>
</tr>
<tr>
<td>5.2 Spatial Interpolation Techniques</td>
<td>47</td>
</tr>
<tr>
<td>5.2.1 Inverse Distance Weight Interpolation</td>
<td>48</td>
</tr>
<tr>
<td>5.2.2 Kriging</td>
<td>49</td>
</tr>
<tr>
<td>5.2.3 Spline Interpolation</td>
<td>54</td>
</tr>
<tr>
<td>5.3 Triangulated Irregular Networks</td>
<td>56</td>
</tr>
<tr>
<td>5.4 Summary</td>
<td>57</td>
</tr>
<tr>
<td>VI. SIMULATION RESULTS</td>
<td>58</td>
</tr>
<tr>
<td>6.1 Simulation Results</td>
<td>58</td>
</tr>
<tr>
<td>6.1.1 Test Data Description</td>
<td>58</td>
</tr>
<tr>
<td>6.1.2 Bare Earth Model Extraction</td>
<td>59</td>
</tr>
<tr>
<td>6.1.3 Statistical Analysis for Ground Trend Model</td>
<td>71</td>
</tr>
<tr>
<td>6.2 Accuracy Measure</td>
<td>73</td>
</tr>
<tr>
<td>6.3 Computational Complexity</td>
<td>76</td>
</tr>
<tr>
<td>6.3.1 Analytical Assessment</td>
<td>76</td>
</tr>
<tr>
<td>6.3.2 Experimental Benchmark</td>
<td>77</td>
</tr>
<tr>
<td>6.4 Summary</td>
<td>78</td>
</tr>
<tr>
<td>VII. CONCLUSION</td>
<td>79</td>
</tr>
<tr>
<td>7.1 Summary</td>
<td>79</td>
</tr>
<tr>
<td>7.2 Dissertation Contributions</td>
<td>83</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>85</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Quantitative quality comparisons of DTM\s for Test Data-1</td>
<td>75</td>
</tr>
<tr>
<td>2. Quantitative quality comparisons of DTM\s for Test Data-2</td>
<td>75</td>
</tr>
<tr>
<td>3. Computational Time Comparison</td>
<td>77</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 The Basic Scheme of the Proposed Algorithm</td>
<td>7</td>
</tr>
<tr>
<td>3.1 K-means Clustering for a LIDAR Data Set</td>
<td>16</td>
</tr>
<tr>
<td>3.2 Threshold Selection Procedure for Histogram Based Clustering</td>
<td>20</td>
</tr>
<tr>
<td>4.1 An Example of Eight Nearest Points for a Query Point</td>
<td>28</td>
</tr>
<tr>
<td>4.2 The Basic Steps Used in Outlier Detection of DTMs</td>
<td>34</td>
</tr>
<tr>
<td>4.3 A Pseudocode for the Elevation Outlier Detection Algorithm</td>
<td>37</td>
</tr>
<tr>
<td>4.4 Example that all $H_i$ are greater than Zero</td>
<td>37</td>
</tr>
<tr>
<td>4.5 Example that all $H_i$ are less than Zero</td>
<td>38</td>
</tr>
<tr>
<td>4.6 Example that a Query Point lie between Neighbor Points</td>
<td>38</td>
</tr>
<tr>
<td>4.7 A Pseudocode for the Gradient Outlier Detection Algorithm</td>
<td>40</td>
</tr>
<tr>
<td>4.8 A Pseudocode for the Verification Step</td>
<td>41</td>
</tr>
<tr>
<td>6.1 DTM of the last return data - Bellingham site, dense forest canopy area</td>
<td>60</td>
</tr>
<tr>
<td>6.2 EarthData Bare earth model- Bellingham site, dense forest canopy area</td>
<td>60</td>
</tr>
<tr>
<td>6.3 Histogram for the data in an x-y window (window size 4×4)</td>
<td>61</td>
</tr>
<tr>
<td>6.4 Threshold Selection of Kernel Density Estimation</td>
<td>62</td>
</tr>
<tr>
<td>6.5 LIDAR Points obtained from Histogram-based Clustering</td>
<td>63</td>
</tr>
<tr>
<td>6.6 LIDAR Points obtained from the Elevation Outlier Detection Algorithm</td>
<td>64</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>6.7 LIDAR Points obtained from the Gradient Outlier Detection Algorithm</td>
<td>64</td>
</tr>
<tr>
<td>6.8 The AIC Threshold Selection Technique</td>
<td>65</td>
</tr>
<tr>
<td>6.9 LIDAR Points obtained from Residuals Truncation</td>
<td>66</td>
</tr>
<tr>
<td>6.10 Terrain Image obtained from IDW Spatial Interpolation</td>
<td>67</td>
</tr>
<tr>
<td>6.11 Terrain Image obtained from Kriging Spatial Interpolation</td>
<td>67</td>
</tr>
<tr>
<td>6.12 Terrain Image obtained from Spline Spatial Interpolation</td>
<td>68</td>
</tr>
<tr>
<td>6.13 DTM Extraction from Linear Prediction</td>
<td>69</td>
</tr>
<tr>
<td>6.14 DTM Extraction from Modified Linear Prediction</td>
<td>69</td>
</tr>
<tr>
<td>6.15 DTM Extraction from Adaptive Smoothing, Delay = 10</td>
<td>70</td>
</tr>
<tr>
<td>6.16 DTM Extraction from Adaptive Smoothing, Delay=20</td>
<td>70</td>
</tr>
<tr>
<td>6.17 Modeling for Statistical Linear Regression Analysis</td>
<td>72</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

1.1 Background

Airborne LIDAR system technology represents a new and independent technology for the modeling of the complex three-dimensional structure, especially urban environments surrounded with objects of different natures and height and dense forests, etc., since data obtained from laser scanning have accuracy in height and range measurements and the system furnishes geometric results in terms of distance, position, attitude, and coordinates [1].

Two types of LIDAR systems, large-footprint and small-footprint LIDAR, are available. Both types of LIDAR systems utilize accurate timing of reflected laser pulses to determine the elevation of targets underneath the aircraft. Large-footprint systems record the continuous returned energy waveform from a few fairly large diameter footprints (10 to 25 m) across the path of the aircraft [2]. Small-footprint systems record multiple discrete returns (1 to 5) from footprints of often less than 1m diameter, but at pulse rates exceeding 33 KHz. LIDAR systems with pulse rates exceeding 80 KHz are expected soon. For each shot, the spatial vector from the laser platform to the point of reflection is established, thus providing the (x, y, z) coordinates of the footprint [1]. Although both LIDAR systems are available, large-footprint LIDAR systems are limited
to government operations. Small-footprint LIDAR systems have become widely available and LIDAR data obtained from such a system are used throughout this study. Since the aerial systems of LIDAR can generate sample points across the flight path as the aircraft moves across the terrain, the measurement data, combined with accurate determination of the aircraft location and altitude, are used to generate elevation models of the area under the aircraft path. The spacing of the post points is determined by the sampling rate, the sensor scan angle, the orientation of the aircraft axes, and the altitude and ground speed of the aircraft [3]. Generally, the operating altitude is limited up to 6000 m and the scan angle is usually less than $\pm 20^\circ$. Since the initial data capture for building elevation models is traditionally very expensive and time consuming, LIDAR technologies have emerged recently and have made it easier and less costly to collect larger amounts of accurate sample data. Furthermore, LIDAR data can be acquired during the day or night since the sun angle selection is not an issue.

Airborne LIDAR systems have several limitations. The interpretability of data is limited due to the fact that no object information is provided [4]. Certain features, such as water and moisture, including rain, clouds, and fog, often absorb laser pulses in the near-infrared region of the spectrum. In spite of these limitations, the applications of LIDAR systems have been widespread since LIDAR has the advantages over the traditional photogrammetric techniques, where the automation of the generation of 3-D city models, as required by many users of Geographic Information Systems (GIS), has become a major focus of photogrammetric research in the past few years [5]. The surveying and modeling of electrical power lines is another application suitable to laser scanning since
multiple return data can be used for the modeling [6]. Furthermore, the forestry community is interested in LIDAR technology, which has been used to predict the forest stand characteristics of height, basal area, and volume through elevation models because several thousands of returns can model trees and stands in three dimensions, and the characteristics of stands, important to forest industry, such as volume and height, are three-dimensional features [7].

Elevation models are often known as digital elevation models (DEMs), digital terrain models (DTMs), and digital surface models (DSMs). As used by the U.S. Geological Survey (USGS), DEMs are the digital cartographic representation of the elevation of the terrain at regularly spaced intervals in the x and y directions. DTMs are similar to DEMs, but they also incorporate the elevation of significant topographic features on the land, plus mass points and breaklines that are irregularly spaced so as to better characterize the true shape of the earth terrain. Also, DSMs are similar to DEMs or DTMs, except that DSMs depict the elevations of the top surfaces of buildings, trees, and other features elevated above the bare earth [8]. Since the forest characteristic, such as canopy height, can be well predicted if the exact DTMs can be extracted in forested areas, the forestry community is interested in the modeling of the earth ground using LIDAR systems, which is the topic of this dissertation.

Extracting DTMs without active sensors is time consuming and consequently very expensive. LIDAR data can be much less expensive to acquire and process than conventional data for wide area mapping. Traditional approaches for the development of precise elevation data sets have previously required the acquisition and analysis of many
stereo pairs of aerial photography, considerable ground surveying, and extensive post-collection analysis. LIDAR has taken this expensive, labor-intensive, and time-consuming task and reduced it to an affordable, flexible, and manageable process [6]. And since LIDAR systems can provide ground points beneath the canopy, even in coniferous forests, it can provide DTMs in these areas with accuracy equivalent to photogrammetric DTM in open areas. Although LIDAR systems can measure the earth ground beneath the canopy, the task for production of high quality DTMs can become difficult in dense forested areas because it is very difficult to determine if the laser beam is reflected from the top or the lower branches of a tree, or from the earth ground.

LIDAR systems are comparable in some ways to the photogrammetric method of automatic generation of DTMs by digital processing of image data. Both methods are highly automated, although photogrammetry is still to a lesser degree. Their results are geometric and can reach, in the application of high precision DTMs, similar accuracies. On the other hand, there are highly essential differences between both methods. LIDAR systems are active sensor systems. They provide ground points in a certain pattern that is primarily determined by the system design and only influenced to some extent by the geometry of the terrain surface and its cover. The photogrammetric points may be arranged in a pre-fixed rigid pattern, but quite often they are arbitrarily selected, depending on the image texture and features. In the case of vegetation, they would lie on the canopy, while laser points can be on or within and below the vegetation cover. However, the photogrammetric restriction to the visible canopy of vegetation can be a disadvantage depending on the purpose of the intended terrain model [1]. If field
sampling is sufficiently intense, a ground-based survey can produce very accurate results. However, the method is labor intensive, and it is not always possible to secure the services of well-trained field workers. Furthermore, the great demand for 3D data at a considerably high level of detail conflicts with the enormous costs of data acquisition.

To generate high quality DTMs using LIDAR data, the most important and difficult step is filtering, where non-terrain points and vegetation are eliminated while keeping terrain points for quality DTM extraction. However, this process is still in the development stages and some techniques depend on manual parameter setting. Therefore, in this dissertation, a filtering algorithm for LIDAR data is presented and then an optimal interpolation technique for terrain modeling is applied such that high quality DTMs are generated.

This dissertation introduces an innovative algorithm with high accuracy to extract DTMs from LIDAR data. Figure 1.1 shows the general configuration of this algorithm. The first step is to distinguish the data measured by LIDAR systems into terrain and non-terrain points. To perform the classification task, an unsupervised classification algorithm is applied since the training data pairs from the return LIDAR data sets cannot be obtained. This is similar to a two-class classification problem. However, the traditional unsupervised classification algorithm such as k-means clustering [9], which is based on the Euclidean distance, cannot catch the spatial variability inherited to the spatial data points. Therefore, it is necessary for a classification algorithm different from the Euclidean distance-based algorithms to be implemented to perform LIDAR data classification. In this stage, a histogram-based clustering algorithm is presented to
classify the data into two classes. However, exact classification is impossible since the true ground returns are not known. The approximated clustering result is enough to process the next stage of the algorithm. The purpose of this step is to approximately eliminate the non-terrain-like points while keeping as many of the terrain-like points as possible. Furthermore, it is actually a pre-processing step to reduce the data points so that the computational complexity is lessened. The detailed algorithm of the histogram-based classification is addressed in Chapter 3. The decision making step of selecting the most terrain-like points is implemented in the following “geometric modeling” step, in which the information such as elevation and surface gradient are used to determine the terrain points. To perform this step, the eight nearest neighbor points can be selected to investigate spatial continuity. The investigation of spatial continuity is the main idea of this step since spatial continuity exists in most earth data sets and two data points close to each other are more likely to have similar values than two data points that are far apart. Actually, extracting the terrain points and eliminating the non-terrain points can be considered as a statistical outlier detection problem. Therefore, in this step, statistical outlier detection algorithms for elevation and gradient are implemented. Then, as a final step of the proposed algorithm, the spatial interpolation technique is performed to generate and visualize the ground surface.
1.2 Dissertation Contributions

The major contribution of this dissertation is the development of an innovative technique with better accuracy than existing techniques to extract ground elevation models collected by small-footprint LIDAR systems. This technique is based on a preprocessing step, ground modeling, and interpolation. In the preprocessing step, a histogram-based clustering technique is introduced to eliminate much of the non-terrain points. Then, in the modeling stage, the information such as elevation and slope between nearest neighbor points is extracted. Since the non-terrain points over the terrain points can be considered as multiple outliers, this process corresponds to an outlier detection process. Accordingly, eight nearest neighbor points in terms of the Euclidean distance are selected to determine residual and gradient indexes for elevation and slope, respectively. These indexes are investigated for a constructed 95% confidence interval. Finally, several spatial interpolations, such as inverse distance weight (IDW), spline, and kriging, are applied and compared for generating a smooth ground surface.

Results shows that the proposed technique yields better results compared to existing techniques in terms of the root mean squared error, absolute mean, and absolute
standard deviation. Some of the existing techniques include linear prediction, modified linear prediction, and adaptive linear prediction techniques.

1.3 Structure of the Dissertation

This dissertation is organized so that Chapter 2 reviews the work related to DTM generation using LIDAR and Chapter 3 addresses a clustering technique based on histogram matching. Chapter 4 describes a technique for extracting terrain points based on the information of neighboring points. As a final step of the algorithm, spatial interpolation techniques to generate the ground surface are described in Chapter 5. Simulation results are given in Chapter 6 and the final remarks are concluded in Chapter 7.
CHAPTER II

RELATED WORKS

Prior to using LIDAR data, digital elevation models and/or digital terrain models have been generated using active remote sensing sensors. The generation of digital elevation models using synthetic aperture radar (SAR) has been very popular in recent years [10, 11, 12, 13, 14]. Davis et al. examined the spatial variability of elevation errors in high-resolution DEMs derived from stereo-image processing [15]. Error models were developed and evaluated by examining the correlation between various DEM parameters and the magnitude of the observed DEM vertical error. Pieraccini et al. proposed a ground-based interferometric synthetic aperture radar (ISAR) technique for terrain mapping [16]. The use of stereoscopic SAR images offered an alternative to ISAR for the generation of DEMs. The stereo radargrammetric method was robust and could generate DEMs of sufficient accuracy to geocode SAR images [17]. The accuracy of DEMs generated by the Jet Propulsion Laboratory (JPL) TOPSAR for extracting canopy height was also evaluated [18]. Wimmer et al. described how high-precision DEMs were obtained over the Wadden Sea using the interferometric radar [19]. In hydrological studies of the Little Washita River Watershed, OK, a SPOT-derived DEM was observed to contain systematic elevation errors, and a technique for minimizing these errors was developed [20]. Multibaseline SAR interferometry was exploited successfully for high-
quality DEMs reconstruction, provided that both noise and atmospheric effects are taken into account. A weighted combination of many uncorrelated topographic profiles strongly reduced the impact of phase artifacts on the final DEMs. The key issue was weights selection. In the present article, a wavelet domain approach was proposed [21]. Twenty-three 1-km advanced very high-resolution radiometer (AVHRR) images were composite by using a Bayesian statistical sampling technique to yield estimates of a simple terrain-based reflectivity model with 180-m resolution. The terrain values were determined from a 90-m resolution DEM [22].

Recently, the interest of forestry applications in the use of LIDAR technology for modeling DTMs has increased since researchers are interested in forest characterizations such as tree density, tree height, and canopy cover. The primary use of small-footprint LIDAR has focused on high-resolution terrain mapping [3]. LIDAR technology is especially used in forestry applications to determine the height of individual trees and for expressions of stand height in forest inventory and management. The total tree height has a direct influence on factors such as wood volume and suitability for wildlife habitat since useful correlations are often present between height and other characteristics. Therefore, height can be used as an estimator of basal area, stem diameter, and merchantable volume. For instance, the tree height has commonly been used with other tree size descriptors in a regression estimation technique in a multi-stage inventory of timber volume. Using LIDAR data, the automatic derivation of tree height, the extraction of other forest stand parameters, as well as the automatic derivation of the break line in
the data have been investigated [23, 24, 25], where few works have been published to
generate DTMs using small-footprint LIDAR data sets in the forestry area.

Blair et al. showed that the vertical structure information for vegetation contained
in a large footprint laser altimeter return waveform can be synthesized using small
footprint data sets [26]. Since small-footprint laser altimeters can be used to represent the
vertical distribution of intercepted surfaces within footprints, they can model the laser
altimeter return waveforms as the sum of the elementary pulses reflected from each
element in the footprint. Cobby et al. described a range image segmentation system for
data from LIDAR [27], measuring either the time of the last significant return, or
measuring the time of both the first and last returns. Petzold et al. used LIDAR data for
the derivation of DTMs and their work was performed using photogrammetric stereo
models or through comparison of the results with large-scale topographic maps [28].
However, these algorithms were not designed for LIDAR data in dense forested areas.
The derivation of high quality DTMs in forested areas with LIDAR data tried at the
Institute of Photogrammetry at the Stuttgart University where the so-called
morphological operator ‘opening’ has been applied for the task of evaluating laser
scanner data [29]. In this application, a window is moved over the data set. The lowest
point in the window is considered to be a ground point. All points within a certain height
range above these points are considered to be ground points as well and are given a
certain weight depending on the window size. The last step is the surface approximation
under the consideration of the weights.
The TerraScan, Inc. filtering method is described in [30]. A x-y grid is laid over the whole data set. The size of this grid has to be specified. The lowest point of each mesh is considered to be a ground point. These points are triangulated, which gives a first representation of the surface. The final surface is built iteratively by adding points to this triangulation. Points are accepted or rejected according to certain criteria. One criterion measures the height of a candidate point above the present surface, and another measures the angle between the surface with and without the candidate point. If certain threshold values are reached, a point is inserted into the triangulation.

Furthermore, the method of Kilian et al. [29] is transformed by using a linear prediction method to determine terrain models in wooded areas with airborne laser scanner data [31, 32]. In these papers, Kraus et al. introduced a linear prediction based algorithm designed for the generation of DTMs for LIDAR data in wooded areas. In this algorithm, a rough approximation of the surface is computed first. Next, the residuals, the distances from the surface to the measured points, are computed. Each z-measurement is given a weight according to its distance value, which is the parameter of a weight function. The surface is then recomputed, giving consideration to the weights of the z-measurements. A point with a high weight attracts the surface, whereas a point that has been assigned a low weight has little influence on the run of the surface. If the residual distance is above a certain value, the point is classified as a non-terrain point and eliminated from this surface interpolation. However, the algorithm relies on a ‘good mixture’ of ground and non-terrain points, which can also be seen as a high frequency of change from ground to non-ground points. Hyyppa et al. selected the minimum z values
within each pixel \((x, y)\) corresponding to minimum surfaces to generate DTMs [33]. Since many trees were still visible in the minimum surface, an 8 x 8 filter removing the existing crown hits was designed mainly looking for the minimum values. If the pixel value of the minimum surface deviates from the filtered minimum surface by less than a certain threshold, the pixel was assumed to be a ground hit. Using ground hits, other values in the surface were interpolated using Delaunary triangulation. However, all those techniques suffer some limitations, which have no ability to track deep valleys in the terrain.

Recently, Lee and Younan have proposed a modified linear prediction technique followed by adaptive smoothing to extract ground elevation models [34]. This technique is an extension to the algorithm proposed by Kraus et al. [23, 31, 32], where the ground points obtained from the linear prediction filtering are compared with the original measurement points to extract the matched points with the same locations as the measurements and then perform interpolation for refinement purposes. The aim of the adaptive filter is to perform a post-processing to eliminate some spurious peaks caused by the modified linear prediction step.

In summary, several techniques have been used to extract DTMs from LIDAR data. They suffer from certain limitations. Thus, it is desired to develop a new algorithm for high quality DTMs extraction.
CHAPTER III
HISTOGRAM BASED CLUSTERING

As a preprocessing step, a clustering stage to reduce the computational burden and eliminate many of the non-terrain points of LIDAR data was applied. Actually, this is a two-class classification stage for which input-output pairs are used as training data for supervised classification problems. However, since one cannot have training data, the problem of unsupervised classification is called clustering. In this chapter, the method that classifies LIDAR data sets into two classes, terrain and non-terrain points, is presented.

3.1 The Purpose of the Clustering Stage

In many applications of classification, it is extremely difficult or even impossible to reliably label a training sample with its true category. In order to obtain the true value in the given data, the specific site associated with the data should be visited, which is a difficult task. Unsupervised classification refers to situations where the objective is to construct decision boundaries based on unlabeled training data. This concept is also known as clustering. The definition of a cluster should include patterns within a cluster that are more similar to each other than patterns belonging to different clusters. A cluster consists of a relatively high density of points separated from other clusters by a relativel
low density of points [9]. The classification of the LIDAR data is a main topic of this chapter. However, one cannot classify the data into two classes perfectly since one does not know where the points with assumed ground points come from. The purpose of this preprocessing step is to eliminate as many of the non-terrain points that existed in the raw LIDAR data prior to geometric modeling, where the points will be further removed.

3.2 Distance Based Clustering

To process data clustering, the measure of similarity needs to be defined because patterns are grouped into a class based on similarity. Unless a meaningful distance measure between pairs of data has been established, no meaningful cluster analysis is possible. The measure of similarity is in general application dependent. The most popular similarity measure is the Euclidean distance. By using the Euclidean distance as a measure of similarity, hyperspherical-shaped clusters of equal sizes are usually detected. The most common proximity index is the Minkowski metric [35], which measures dissimilarity. Given N patterns, \( x_i = (x_{i1}, ..., x_{in})^T, i=1, 2, ..., N \), the Minkowski metric for measuring the dissimilarity between the jth and kth patterns is defined by

\[
d(j,k) = \left( \sum_{i=1}^{n} |x_{ji} - x_{ki}|^r \right)^{1/r},
\]

(3.1)

where \( r \geq 1 \). The Euclidean distance \( (r = 2) \) is one of the most common Minkowski distance metrics. Since clusters are usually of arbitrary shapes and sizes, the Minkowski metrics seem to be a poor choice for situations where no a priori information about the geometric characteristics of the data set exists.
Clustering algorithms such as the k-means algorithm [36] and fuzzy C-means algorithm [37] are based on the “sum of intracluster distances” criterion. It is well known that this criterion is effective only when all clusters are roughly spheroidal with similar volume [38, 39]. Therefore, the k-means clustering algorithm fails to produce meaningful clusters for LIDAR data sets since the ground return LIDAR data points show arbitrary shapes as in Figure 3.1. Other algorithms have been developed to take into account clusters of different shapes and sizes. For example, the Gustafson-Kessel algorithm tries to accommodate ellipsoidal clusters by individualizing the norm matrix for each cluster [40]. This approach works better when the clusters are expected to be ellipsoidal rather than spherical, but still requires that the volumes of the clusters are roughly equal.

Figure 3.1. K-means Clustering for a LIDAR Data Set
3.3 Probabilistic Clustering

There is a long tradition in the statistical literature of using mixture models to perform probabilistic clustering [9]. A key feature of the mixture approach to clustering is the ability to handle uncertainty about cluster membership in a probabilistic manner by allowing overlap of the clusters. That is, methods belonging to this type of clustering method assume a statistical model instead of a predefined metric as before.

Let $X$ be a $d$-dimensional random variable and let $x$ represent a particular value of $X$, e.g., an observed data vector with $d$ components. A finite mixture probability density function for $X$ can be written as

$$f^{(k)}(x | \Phi^{(k)}) = \sum_{j=1}^{k} \alpha_j g_j(x | \theta_j), \quad (3.2)$$

where $k$ is the number of components in the model and each of the $g_j$ are the component density functions. The $\theta_j$ are the parameters associated with density component $g_j$ and the $\alpha_j$ are the relative weights for each component $j$, where $\Sigma_j \alpha_j = 1$ and $\alpha_j > 0$, $1 \leq j \leq k$. $\Phi^{(k)} = \{\alpha_1, \ldots, \alpha_k, \theta_1, \ldots, \theta_k\}$ denotes the set of parameters for the overall mixture model with $k$ components. Let $D^{\text{train}} = \{x_1, \ldots, x_N\}$ denote the training data from which the model parameters are estimated. Assuming independent observations from an underlying true density function $f(x)$, the log-likelihood of $\Phi^{(k)}$ is defined as

$$l(\Phi^{(k)} | D^{\text{train}}) = \log p(D^{\text{train}} | \Phi^{(k)}) = \sum_{i=1}^{N} \log \left( \sum_{j=1}^{k} \alpha_j g_j(x_i | \theta_j) \right). \quad (3.3)$$

Direct maximization of the mixture log-likelihood expression in Equation (3.3) is a difficult task. Thus, much of the popularity of mixture models in recent years is due to the
existence of efficient iterative estimation techniques for obtaining the maximum of this likelihood.

In particularly, the expected-maximization (EM) algorithm decomposes mixtures based on the maximum likelihood estimation and can be used for clustering [41]. That is, the algorithm can be split in two parts, the estimation of the probabilities for a data point to be in a cluster and the maximization of the mixture likelihood. The algorithm starts with any initial values and continues with its iterations as long as the parameter estimates differ by a certain amount from one iteration to the next. The result is the mean and covariance matrices for the clusters. However, the EM method is known to be highly sensitive to initialization [38]. A fuzzified version of the EM algorithm, as well as a special initialization procedure to reduce the EM sensitivity to initialization, was proposed in [42]. Another probabilistic clustering algorithm has been proposed by Rose et al. in [43].

3.4 Neural Networks Approach

As a clustering technique using neural networks, the popular self-organizing map (SOM) algorithm has an intimate connection with the classical k-means clustering algorithm [44]. Another similarity-based clustering algorithm based on neural networks is the adaptive resonance theory (ART) [45].

In the SOM, neurons are arranged in an m-dimensional grid, where m is usually 1, 2, or 3. Each neuron is connected to all the d input units. The weights on the connections for each neuron form a d-dimensional weight vector. During training,
patterns are presented to the network in random order. At each presentation, the winner whose weight vector is the closest to the input vector is first identified. Then, all the neurons in the neighborhood of the winner are updated such that their weight vectors move towards the input vector. Consequently, after training is done, the weight vectors of neighboring neurons in the grid are likely to represent input patterns, which are close in the original feature space. Thus, a “topology-preserving” map is formed.

3.5 Histogram Based Clustering Algorithm

In LIDAR applications, the elevation of the data can be a useful feature for classification purposes. That is, if the laser beam is reflected from the top or the lower branches of a tree, the measured points will lie too high with respect to the terrain. Otherwise, the points will lie relatively low corresponding to the terrain. Histogram outputs of the return LIDAR elevation data can be shown to be multi-modal, which indicates that the return data set is in fact composed of subclasses with slightly different characteristics. Or, the histogram can form a unimodal or uniform distribution according to the ground shape. In this dissertation, a histogram-clustering algorithm is presented. Based on this algorithm, the data is divided into x-y windows and a threshold is applied to the data within each window. The selection of threshold indexes is important for the implementation of this algorithm. A flowchart illustrating the threshold selection is shown in Figure 3.2. If the data samples inside an x-y window show a normal distribution function type, no thresholding is performed. Otherwise, the data can show a multi-modal, skewed, or uniform distribution. Kernel density estimation is applied to generate the
distribution. On the other hand, if two local maximum can be found, a local minimum between two maximum will be the threshold index that classifies the LIDAR data into ground and non-ground data. Otherwise, no thresholding is performed since the distribution will be a unimodal such as uniform or skewed.

Figure 3.2. Threshold Selection Procedure for Histogram Based Clustering

According to Figure 3.2, the input data are tested to assess its normality. The purpose of this test is to reduce the computational time. That is, if the given data satisfies the requirement of a normal distribution, no threshold processing is necessary. Therefore, the computational time will be reduced since further processing is not required. Of several normality tests, the Lilliefors test evaluates the hypothesis that the sample has a
normal distribution with unspecified mean and variance against the alternative hypothesis that the sample does not have a normal distribution [52]. The main difference from the well-known Kolmogorov-Smirnov test (KS test) is in the assumption about the mean and standard deviation of the normal distribution [52]. The KS test assumes that the mean and standard deviation of the normal distribution are known; the Lilliefors test does not make this assumption. For the Lilliefors test statistic, from a random sample of size n, the sample mean $\overline{X}$ and standard deviation, $\sigma$, are computed and the samples are converted to Z-scores according to:

$$Z_i = \frac{X_i - \overline{X}}{\sigma}, \ i = 1, 2, \ldots, n.$$  \hspace{1cm} (3.4)

Let the hypothesis set be

$H_0$: the distribution function of the $X_i$’s is normal

$H_1$: the distribution function of the $X_i$’s is not normal

The test statistics is the maximum vertical distance between the empirical distribution function of the Z-score and the distribution function of the standard normal distribution. The cumulative density function (CDF) of the standard normal distribution is plotted and called $F(x)$. A plot of the empirical CDF of the Z-scores is superimposed and called $S(x)$. Then, the test statistic is simply the maximum vertical distance between the two plots, or

$$T = \sup \left| F(x) - S(x) \right|.$$  \hspace{1cm} (3.5)

Therefore, the decision rule is that $H_0$ is rejected at the significant level $\alpha$ if $T$ exceeds the $1-\alpha$ quantile in a table of quantiles of the Lilliefors test statistic for normality.
After the normality test is completed, the distribution inherent to the data points is determined. However, histograms have potential limitations to determine if a data set produces a multi-modal, unimodal, or uniform distribution because the choices of the bin width of the histograms and the starting position of the first interval are arbitrary. Thus, the appearance of a histogram depends on both the choice of the origin and the bin width. Kernel Density Estimates (KDEs), which at their simplest can be thought of as smoothed histograms, can be used to avoid many of these problems.

Given \( n \) points \( X_1, X_2, \ldots, X_n \), a KDE can be thought of as being obtained by placing a “bump” at each point and then summing the height of each bump at each point on the x-axis. The shape of the bump is defined by a mathematical function – the kernel \( K(x) \) – that integrates to 1. The spread of the bump is determined by a bandwidth \( h \). Mathematically, this can be expressed as

\[
\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - X_i}{h} \right).
\]

Compared to the histogram, the shape of \( \hat{f}(x) \) does not depend upon the choice of the origin, but is affected by the bandwidth \( h \). In general, large values of \( h \) over-smooth the data, while small values of \( h \) under-smooth the data. However, the objective or data-driven choices of \( h \) can be made as follows.

The data can be thought of as a sample of \( n \) from an underlying and unknown true density \( f(x) \). It is possible to define a measure of closeness between the KDE and the true density, leading to an estimate of \( h \) that maximizes the closeness. If it is assumed that the true density is normal, it can be shown that an optimal choice of \( h \) is
\[ h = 1.06n^{-1/5} \hat{\sigma}, \]  

(3.7)

where \( \hat{\sigma} \) is an estimate of \( \sigma \), the standard deviation of the normal distribution. This is the normal scale rule and will over-smooth the data if the underlying density is not normal. The estimate of \( h \) depends on properties of the true density that are unknown. A related approach is the ‘solve the equation’ (STE) method, in which an equation that relates \( h \) to a function of the unknown density is defined. In essence, an initial estimate of \( h \) leads to an estimate of the density, that in turn leads to a new value for \( h \) and a new density estimate. The process continues until the estimate of \( h \) converges.

If two local maximum points can be found from a KDE, a local minimum can easily be obtained from these points, corresponding to a threshold index. This indicates that the observed data can contain heterogeneous samples and a threshold can divide the given data samples into two classes. Thus, if the elevation z-value is less than that of the threshold index, the points are considered as ground points. Otherwise, the points are considered to be non-ground points, which must be eliminated.

### 3.6 Summary

In this chapter, a preprocessing method that divides the last return LIDAR raw points into ground and non-ground points and then eliminates the non-ground points is described. This preprocessing step corresponds to data clustering or unsupervised classification since the training data pairs for the LIDAR data cannot be extracted. The basic clustering methods based on the Euclidean distance fail to divide the data properly
into two classes. Therefore, a clustering technique based on histograms is implemented. To process this algorithm, threshold selection is inevitable. For the data within an x-y window, a normality of the data is tested. If the data can pass the test, no threshold selection is necessary to save computational time. Otherwise, a distribution is estimated through the KDE method. From the estimated distribution, a local minimum index between two local maximum can be used as a threshold to divide the data into two classes.
CHAPTER IV
GEOMETRIC MODELING

Once a classification procedure finds a proper representation, further post-processing is inevitable since the data obtained in the clustering step is still likely to have non-terrain-like data. To further eliminate non-terrain-like points, the information between closest points can be useful since the closer data points tend to be more correlated for spatial data. This chapter describes the ground modeling technique, which uses distance and surface gradient between close points. Actually, since this modeling can be considered as an outlier detection problem, a statistical test based on a confidence interval is performed to determine outliers that exist in the measured data points.

4.1 Outlier Detection

To further eliminate non-terrain points, the information such as slope and elevation between neighbor points and a query point can be used. The elevation of ground samples is an indication of the ground trend and the slope is also enhanced as a topographic feature of great importance in the terrain modeling [46]. If the slope and elevation of neighbor points are much different from that of a query point, the point is deviated from neighbor points since the closer the points are, the more correlated they are.
for ground surface models. Actually, the investigation of a query point for close neighbor points can be considered as an outlier detection problem.

There are several outlier detection methods for correcting errors of elevation models. Hannah used subjectively predefined, fixed thresholds by specifying the allowable slope and slope change [47]. Felicisimo employed a parametric statistical test based on global estimates for the central tendency and dispersion [46]. However, since most spatial data are non-stationary, and the resulting ground surfaces are often complex and heterogeneous, their approaches did not get reliable results. That is, since the magnitude and dispersion of interpolation residuals would be relatively small in a flat or slightly undulated area and large in a hilly area, global estimates based on the entire data set tend to hide the outliers. Lopez also applied a principal component analysis (PCA) to deal with errors in DEM extraction [48]. Recently, Liu et al. described an effective method for detecting outliers in an irregularly distributed spatial data set [49]. They devised both an outlier index, indicating the residual between an observed value and a predicted value, and a surface gradient-based index, indicating the surface shape and the local variation. The selection of the nearest points precedes the calculation of those indexes. To define neighborhood points, they used a super-block based method that partitions the bounding area of data into an array of blocks. The idea of the super-block based method is to partition the bounding rectangular area, defined by the minimum and maximum of the x and y coordinates, of the data set into an array of super blocks, and then assign each point to a block according to its location. However, this method performs well if only few outliers exist in the neighbor inside the super-blocks. For
LIDAR data, the return data can show multiple outliers over the near terrain points. Therefore, an algorithm for detecting multiple outliers in neighbor points is needed.

In this dissertation, the eight nearest neighbor points for a query point are selected as neighbor points to estimate the surface gradient and the elevation of one point over neighbor points. Other candidate values such as 4, 12, or 16 can be chosen for neighbor points, but eight points are expected to be enough to investigate the relationship of close points. In this case, “nearest” is measured in terms of the Euclidean distance. Figure 4.1 shows a situation where the eight nearest neighbor points, p1, ..., p8, exist around a query point q. In this figure, T1, T2, ..., T8 indicate triangles consisting of a query point q and the 8 neighbor points. Assuming continuity of the underlying surface, the elevation z of a data point can be well predicted through interpolation based on its surrounding neighboring points. The difference between the predicted value and the observed value indicates the extent to which the data point under examination is inconsistent with its neighbors. The idea of checking the observation value against the predicted value from its neighbor points is referred to as a cross-validation technique [50].
An inverse distance-weighting (IDW) algorithm, used as an interpolation technique, can be then applied to obtain a reliable prediction. In this algorithm, the predicted value is a linearly weighted function of its neighbors. That is,

\[ \hat{z}_q = \sum_{i=1}^{n} W_i z_i \]  \hspace{2cm} (4.1)

\[ W_i = \frac{d_i^{-b}}{\sum_{j=1}^{n} d_j^{-b}} \]  \hspace{2cm} (4.2)

where \( n \) is the number of neighbor points, \( \hat{z}_q \) is the predicted value for the query point \( q \), \( z_i \) is the elevation value of a neighbor point, \( d_i \) is the distance between the query point \( q \) and its neighbor points, \( w_i \) is the weight of neighbor points, and \( b \) is the global distance friction factor. However, if outliers exist in the selected neighbor points, the predicted
value gained by using Equation (4.1) and (4.2) will be contaminated. To make the prediction robust in the face of outliers in the neighbor points, a leave-one out technique can be used to identify the most influential neighbor points [50]. That is, one neighbor point at a time can be dropped and the remaining neighbor points can be used to make a prediction. Repeating this procedure for every neighbor point, additional predictions can be obtained by

\[ \hat{z}_q^{(k)} = \sum_{i \neq k} w_i z_i, \quad k = 1, \ldots, 8 \]  

(4.3)

\[ w_i = \frac{d_i^{-b}}{\sum_{j=1}^{n} d_j^{-b} - d_k^{-b}}, \quad k = 1, \ldots, 8, \]  

(4.4)

where \( \hat{z}_q^{(k)} \) is the predicted elevation value with the neighbor point \( p_k \) omitted, and the other parameters are the same as in Equation (4.1) and (4.2). To indicate the influence of the neighbor point \( p_k \), the absolute difference \( |\hat{z}_q^{(k)} - \hat{z}_q| \) can be set. Thus, in this algorithm, the most influential neighbor points having the large absolute difference can be dropped and then the weighted average of the remaining neighbor points can be taken as the robust estimate of the surface value at query point \( q \). Accordingly,

\[ \hat{z}_q = \sum_{i \neq k} w_i z_i \]  

(4.5)

\[ w_i = \frac{d_i^{-b}}{\sum_{j=1}^{n} d_j^{-b} - d_k^{-b}}, \]  

(4.6)
where \( \hat{z}_q \) is the robust prediction value calculated with the most influential neighbor points omitted and \( k_i \) is the \( i \)th point to be dropped. Furthermore, a statistical decision, based on the Akaike information theoretic criteria (AIC), is used so that the process of dropping the most influential neighbor points is not subjective [55]. Then, the residual between the observed elevation value and the predicted value at a query point \( q \) can be used as a residual index, \( \Delta z_q \),

\[
\Delta z_q = z_q - \hat{z}_q. \tag{4.7}
\]

If the query point is contaminated by serious error, the absolute value of \( \Delta z_q \) will be significantly larger.

The surface gradient around an individual data point can also be estimated from its adjacent neighbor points. A center point in conjunction with its surrounding neighbor points forms eight triangles, as in Figure 4.1. These triangles have various orientations in a three-dimensional space, depending on the relative positions and elevations of each pair of neighbors. The gradient for each triangle can be calculated by taking the cross product of any pair of sides from each triangle [51]. For example, for triangle \( T_1 \), the formula of the cross product vector can be expressed as [49]:

\[
X_1 = (y_2 - y_1)(z - z_1) - (y - y_1)(z_2 - z_1), \tag{4.8}
\]

\[
Y_1 = (z_2 - z_1)(x - x_1) - (z - z_1)(x_2 - x_1), \tag{4.9}
\]

\[
Z_1 = (x_2 - x_1)(y - y_1) - (x - x_1)(y_2 - y_1). \tag{4.10}
\]

This three-dimensional vector is perpendicular to the triangle plane \( T_1 \). For each triangle plane, the gradient can be calculated as:
\[ G_i = \sqrt{\left( \frac{X_i}{Z_i} \right)^2 + \left( \frac{Y_i}{Z_i} \right)^2}, \quad i = 1, 2, \ldots, n, \quad (4.11) \]

where \( n \) is the number of neighbor points. The weighted average of the gradients of the surrounding triangles reflects the overall gradient around the query point \( q \). If a triangle is large, its two neighbor points must be far away from the query point and the calculated gradient would be less reliable for representing the surface shape around the query point \( q \). Therefore, the larger the triangle is, the smaller the weight it has. Accordingly,

\[ \hat{G}_q = \sum_{i=1}^{n} G_i w_i \quad (4.12) \]

\[ w_i = \frac{a_i^{-1}}{\sum_{j=1}^{n} a_j^{-1}}, \quad (4.13) \]

where \( \hat{G}_q \) is the weighted average gradient, \( G_i \) is the gradient of triangle \( T_i \), \( a_i \) is the area of triangle \( T_i \), i.e.,

\[ a_i = \sqrt{\frac{X_i^2 + Y_i^2 + Z_i^2}{2}}, \quad i = 1, 2, \ldots, n, \quad (4.14) \]

and \( w_i \) is the weight of triangle \( T_i \). However, the surface gradient calculated in this way is subject to error if outliers exist in the neighbors. Like the case of the residual index, the most influential triangles having large gradient values can be dropped to make the gradient estimate robust to the presence of outliers in the neighbor points. The decision to drop the most influential triangles is also based on the AIC criterion. The weighted average of surface gradients of the remaining triangles is taken as the robust estimate for the surface gradient around the query point \( q \). That is,
\[ \hat{G} = \sum_{i \neq k_i} w_i G_i \]  

(4.15)

\[ w_i = \frac{a_i^{-1}}{\sum_{j=1}^{n} a_j^{-1} - a_{k_i}^{-1}} \]  

(4.16)

where \( \hat{G} \) is the robust weighted average gradient with the triangles dropped out and \( k_i \) is the point to be dropped.

### 4.2 Confidence Intervals

In this dissertation, statistical outlier tests for the two indexes, residual index and surface gradient index, can be established to determine outlier points. To process a statistical outlier test, a 95% confidence interval is set. That is, if the residuals of the elevation and the surface gradient of neighbor points over a query point are outside the confidence interval, they are considered to be outliers. To construct a confidence interval, the mean and standard deviation are the best unbiased measures for the central tendency and the spread assuming a normal distribution of the sample without contamination. However, according to the sample size, the distribution for a confidence interval is different. When the sample size is greater than 30, the sample distribution can be approximated to a normal distribution. On the other hand, when the sample size is small, less than 30, the statistic approximately follows the Student t distribution with \( n-1 \) degrees of freedom [52]. However, the measure of the mean and standard deviation in the possibility of outliers will not be reliable. Therefore, other statistics to measure the central tendency and dispersion considering outliers should be used. In this dissertation,
trimmed mean and winsorized standard deviation are applied to provide the estimate for the central tendency and the dispersion [53]. Accordingly,

\[
\Delta z_{trim} = \frac{\sum_{i=k+1}^{n-k} \Delta z_i}{n-2k} \quad (4.17)
\]

\[
\Delta z_{win} = \frac{k(\Delta z_{k+1} + \Delta z_{n-k}) + \sum_{i=k+1}^{n-k} \Delta z_i}{n} \quad (4.18)
\]

\[
s_{win} = \sqrt{\frac{k\left[(\Delta z_{k+1} - \Delta z_{win})^2 + (\Delta z_{n-k} - \Delta z_{win})^2\right] + \sum_{i=k+1}^{n-k} (\Delta z_i - \Delta z_{win})^2}{n-2k-1}}, \quad (4.19)
\]

where \(\Delta z_{trim}\) is the trimmed mean, \(\Delta z_{win}\) is the winsorized mean, \(s_{win}\) is the winsorized standard deviation, \(\Delta z_i\) denotes the \(i\)th ordered interpolation residual, \(k\) is the number of observation eliminated at each end of the distribution, and \(n\) is the total number of observations. As indicated in equations (4.17) – (4.19), winsorization replaces the lower and upper ends of the ordered interpolation residuals by their nearest adjacent values, while the trimming simply discards the lower and upper ends [53]. After trimming or winsorizing the extreme sample values at the two ends, the remaining data can be regarded as a clean subset, which is assumed to be free of outliers.

With these statistics, the deviation and spread statistic still satisfy a Student’s \(t\) distribution with \(n-2k-1\) degrees of freedom over a range of different possible distributions and in the presence of outliers [53]. As a result, a 95% confidence interval for the residual index can be set as

\[
\Delta z_{trim} - t_{\alpha/2, n-2k-1} \cdot s_{win} \leq \Delta z_j \leq \Delta z_{trim} + t_{\alpha/2, n-2k-1} \cdot s_{win} \quad (4.20)
\]
Also, a 95% confidence interval for the gradient index can be constructed in a similar way. However, since the surface gradient values are positive, both trimming and winsorizing consider only the upper end of the ordered gradient index.

4.3 DTM Error Correction Algorithm

The outlier detection algorithm for terrain modeling of LIDAR data can be summarized in three steps. The output data obtained by the histogram-based clustering technique can be used as input to an elevation outlier detection stage. Then, outliers can be eliminated further through a gradient outlier detection stage since elevations as well as gradients between neighbor points can affect the quality of the ground surface model. Finally, the final scattered points of a ground surface can be obtained through a “verification” stage, which investigates the outliers globally. Figure 4.2 shows the basic steps to detect existing outliers in the return LIDAR data such that a high quality DTM is generated.

Figure 4.2. The Basic Steps used in Outlier Detection of DTMs

4.3.1 Elevation Outlier Detection

Each point of the input data is investigated through an elevation outlier detection procedure to see if it is an outlier or not. This process continues until all the points are
visited. As a starting point, a point with a minimum elevation value is selected because the point is likely to be a “true” point of the return LIDAR data. From this point, the eight nearest neighbor points can be selected. In this case, the “nearest” neighbor points are measured in terms of the Euclidean distance of two dimensional x-y domains. From those neighbors, a query point is determined to see if the point is a ground point or not. However, since the minimum point is likely to be a “true” one, the investigation starts at the first nearest point to the minimum point.

To process this algorithm, the information such as the distance and height difference between the eight neighbors points and the query point can be extracted and used effectively with the residual index. That is, the vector distance between the points can be extracted and the height differences can also be obtained, and both can be used to process this algorithm. Let $p_0$ be an investing point and $p_1, \ldots, p_8$ be eight neighbors points. The height and the distance can be defined as:

$$H_i = p_{i,z} - p_{0,z}, \quad i = 1, 2, \ldots, 8$$

$$V_i = \sqrt{(p_i(x) - p_0(x))^2 + (p_i(y) - p_0(y))^2 + (p_i(z) - p_0(z))^2}, \quad i = 1, \ldots, 8$$

where $p_{i,z}$ is the height of the $i$th neighbor point, $p_{0,z}$ is the height of the investing point, $H_i$ is the difference between $p_{i,z}$ and $p_{0,z}$, and $V_i$ is the distance between $p_{i,z}$ and $p_{0,z}$. A pseudocode to implement this algorithm is shown in Figure 4.3. A graphical representation helps better in understanding the implementation of this algorithm. Figure 4.4 illustrates an example where all the $H_i$’s are greater than zero, considering the query point is not an outlier. Also, an example where all the $H_i$’s are less than zero is shown in Figure 4.5. In this case, the algorithm allows two choices: if the distance of the elevation
between a query point and a trimmean of neighbor points is greater than that of the elevation between a minimum point and a trimmean of neighbor points, the point is considered to be an outlier. In the pseudocode, \( D_1 \) is the distance of the elevation between a minimum point and a mean of neighbor points and \( D_2 \) is the distance of the elevation between a query point and a mean of neighbor points. For the third case, a query point lies between neighbor points, as illustrated in Figure 4.6. In this case, the algorithm performs a leave-out-out test and investigates a 95% confidence interval that is constructed through the test. That is, for residual indexes obtained from a leave-one-out test, the two most influential residual indexes are dropped and a query point is predicted using the remaining residual indexes. The residual index for the newly predicted point is investigated if it is inside the 95% confidence interval.
if all Hi are less than zero,
  if (D_2 > D_1),
    point is an outlier
  else
    point is not an outlier
  end
else if all Hi are greater than zero,
  point is not an outlier;
else
  for i = 1:n
    neighbors point is assigned to temp
    delete a data
    prediction of p_0,
    difference of prediction of p_0 and p_{0,z}
  end
  delete two points in the neighbors
  prediction of p_0 using the deleted neighbors
  calculate residual index
end
construct a confidence interval using (4.17) – (4.19)
if a residual index is inside the confidence interval
  point is not an outlier
else
  point is an outlier
end

Figure 4.3 A Pseudocode for the Elevation Outlier Detection Algorithm

Figure 4.4. Example that all H_i are greater than Zero
Figure 4.5. Example that all $H_i$ are less than Zero

Figure 4.6. Example of a Query Point that lies between Neighbor Points
4.3.2 Slope Outlier Detection Algorithm

The surface gradient, indicating the surface shape and the local variation, is a good feature of local surface continuity. Therefore, the gradient as well as elevation can be useful for investigating spatial continuity existed in the data set. The surface gradient around a query point can be estimated from its adjacent neighbor points. Likewise, surface gradient estimates can be obtained for each neighbor point. In the possibility of outliers, the two most influential gradients are dropped and surface gradients can be estimated using equations (4.15) and (4.16). A 95% confidence interval can be then constructed in the elevation outlier detection algorithm. To construct the confidence interval, only the upper ends of the ordered gradient index are considered because the gradient value is always positive. A pseudocode used for the gradient outlier detection algorithm is shown in Figure 4.7.
4.3.3 Verification Stage

A “verification” step that determines if outliers exist globally should be performed in the final stage since the results of the elevation and gradient outlier detection steps may contain some outliers. Those outliers must be eliminated since they affect the quality of the ground surface. In this verification step, the investigation of outliers is performed globally. That is, the data points with relatively high elevation over neighbors, if they exist, are eliminated globally. The pseudocode of this technique is shown in Figure 4.8. The idea of this technique is that the difference between an outlier and the prediction without it is large. An inverse distance-weighting algorithm can be used to obtain the prediction of a query point and a leave-one-out test can be performed to identify any outliers. Therefore, the residuals for all points can be obtained and
arranged in an increasing order. Then, data points with large residuals are discarded. For example, it can be assumed that 5% of the data points of the results are outliers. Therefore, the threshold \( k \) can be set as:

\[
k = \text{ceil} \left( \frac{0.95 \cdot n}{100} \right),
\]

(4.23)

where \( n \) is the size of the data points and “ceil” rounds the elements of (...) to the nearest integers towards infinity. However, assumptions should be avoided such that the performance is independent of the assumptions. Therefore, the AIC, described next in detail, can also be applied for the selection of the truncation number \( k \).

```
for i = 1:size of input
    assign input to temp data
    assign ith row of temp data to p
    delete ith row in temp
    prediction of p
    calculate residual
end

arrange residual in increasing order
set threshold k
discard points with k largest residual
```

Figure 4.8. A Pseudocode for the Verification Step

### 4.4 Truncation Method for Outliers Detection

In the previous section, the elevation and gradient outlier detection steps were estimated. The robustness of these estimates to the presence of outliers in the neighboring
points can be determined by how many of the most influential data points should be dropped. Furthermore, the problem to discard the points having large residual errors is also met in the “verification” step. This problem can be thought of selecting data points having independent residuals. That is, since spatial data is spatially correlated, the residual between the predicted value from neighbor points and the measured value will be generally very small, but the residual will be erratically large if outliers exist.

The obtained residuals can be described by the following model:

\[ y(n) = x(n) + r(n), \quad n = 1, \ldots, p, \quad (4.24) \]

where \( y(n) \) denotes the residual set, \( x(n) \) denotes a set of data containing large residuals, i.e., considered as a signal component, and \( r(n) \) denotes a set of data containing small residuals, i.e., considered as a noise component. Assume that \( r(n) \) is a Gaussian random process, independent of \( x(n) \), with zero mean and variance \( \sigma^2 \). A crucial problem associated with the model described in (4.24) is that of determining the number of independent data samples from a finite set of the residual \( y(n) \), most of the correlated sample sets. A promising approach to this problem is based on the structure of the covariance matrix of the residual set \( y(n) \). Because \( r(n) \) is zero mean and independent of the estimated data \( x(n) \), it follows that the covariance matrix of \( y(n) \) is given by

\[ R_y = R_x + \sigma^2 I \quad (4.25) \]

Assuming that the matrix \( R_x \) is of full rank, i.e., the column vectors are linearly independent, it follows that the rank of \( R_x \) is \( q \), or equivalently, the \( p-q \) smallest eigenvalues of \( R_x \) are equal to zero. Denoting the eigenvalues of \( R_y \) by \( \lambda_1 \geq \lambda_2 \cdots \geq \lambda_p \), it follows that the smallest \( p-q \) eigenvalues of \( R_y \) are all equal to \( \sigma^2 \), i.e.,
\[ \lambda_{q+1} = \lambda_{q+2} = \ldots = \lambda_p = \sigma^2. \quad (4.26) \]

The problem is that the covariance matrix is unknown in practice. When estimated from a finite sample size, the resulting eigenvalues are all different, thus making it difficult to determine the rank \( q \) merely by observing the eigenvalues. Therefore, the Akaike information theoretic criterion (AIC) is applied to select the rank \( q \). The AIC is defined by [55]:

\[
AIC(k) = -2 \log f(Y \mid \hat{\Theta}) + 2k, \quad (4.27)
\]

where \( \hat{\Theta} \) is the maximum likelihood estimate of the parameter vector \( \Theta \) and \( k \) is the number of free adjusted parameters in \( \Theta \). The first term is the log-likelihood of the maximum likelihood estimator of the parameters of the model. The second term is a bias correction term, inserted so as to make the AIC an unbiased estimate of the mean Kullback-Liebler distance between the density \( f(Y \mid \Theta) \) and the estimated density \( f(Y \mid \hat{\Theta}) \).

It considers the following family of covariance matrices

\[
R_{y}^{(k)} = R_{x}^{(k)} + \sigma^2 I, \quad (4.28)
\]

where \( R_{x}^{(k)} \) denotes a matrix of rank \( k \), \( \sigma \) denotes an unknown value, and \( k \in \{1, \ldots, p\} \) ranges over the set of all possible number. Using the well-known spectral representation theorem from linear algebra, \( R_{y}^{(k)} \) can be expressed as [55]

\[
R_{y}^{(k)} = \sum_{i=1}^{k} (\lambda_i - \sigma^2) V_i V_i^T + \sigma^2 I, \quad (4.29)
\]
where $\lambda_1, \ldots, \lambda_k$ and $V_1, \ldots, V_k$ are the eigenvalues and eigenvectors, respectively, of $R_y^{(k)}$. Let $\Theta^{(k)}$ be the parameter vector of the model, it follows that

$$\Theta^{(k)T} = (\lambda_1, \ldots, \lambda_k, \sigma^2, V_1^T, \ldots, V_k^T)$$

Since the observations are assumed to be Gaussian random vectors with zero mean, the probability density is given by

$$f(Y | \Theta^{(k)}) = \frac{1}{(2\pi)^{p/2}(\det R_y^{(k)})^{1/2}} e^{-\frac{1}{2}y^T [R_y^{(k)}]^{-1} y}.$$  \hfill (4.31)

Taking the logarithm and omitting terms that do not depend on the parameter vector, the log-likelihood function $L(\Theta^{(k)})$ is given by

$$L(\Theta^{(k)}) = -N \log(\det R_y^{(k)}) - tr[R_y^{(k)}]^{-1} \hat{R}_y.$$  \hfill (4.32)

The maximum-likelihood estimate is the value of $\Theta^{(k)}$ that maximize (4.32). These estimates are given by

$$\hat{\lambda}_i = l_i \quad i = 1, \ldots, k$$

$$\hat{\sigma}^2 = \frac{1}{p-k} \sum_{i=k+1}^{p} l_i$$

$$\hat{V}_i = C_i, \quad i = 1, \ldots, k,$$  \hfill (4.33)

where $l_1 > l_2, \ldots, l_p$ and $C_1, \ldots, C_p$ are the eigenvalues and eigenvectors of the sample covariance matrix. Substituting the maximum likelihood estimates (4.33)–(4.35) in the log-likelihood (4.32) yields,
\[
L(\Theta) = \log \left( \frac{\prod_{i=k+1}^{p} l_i^{1/(p-k)}}{\frac{1}{p-k} \sum_{i=k+1}^{p} l_i} \right)^{(p-k)N} .
\]

Therefore, the form of the AIC is given by

\[
AIC(k) = -2 \log \left( \frac{\prod_{i=k+1}^{p} l_i^{1/(p-k)}}{\frac{1}{p-k} \sum_{i=k+1}^{p} l_i} \right)^{(p-k)N} + 2k(2p-k). \tag{4.37}
\]

Thus, the number of truncation is determined as the value of \( k \in \{1, \ldots, p\} \) for which the AIC is minimized.

### 4.5 Summary

This chapter describes a technique to extract terrain data points from the output of the histogram-based clustering. This technique can be considered as an outlier detection problem for spatial data. To implement this technique, two indexes, the residual index and the gradient index for nearest eight neighbor points, are used effectively since the closer the points are, the more they are correlated. For outliers’ detection, statistical tests using a 95% confidence interval over these two indexes are performed. Finally, a verification stage that further removes the remaining outliers, if any exist, is implemented through the AIC criteria, which is a statistical order truncation method.
CHAPTER V

SPATIAL INTERPOLATION

Now that the data obtained from Chapter 4 contain the scattered terrain-like points, it is necessary to acquire a series of irregularly spaced elevation points from which uniformly spaced elevation points are interpolated since the topographic surface can be considered to be a mathematically continuous surface. In this chapter, spatial interpolation techniques can be used to construct a digital model of the surface by estimating heights at any location based on sample elevations.

5.1 The Purpose of Spatial Interpolation

Spatial interpolation provides a method for transforming values representing landscape phenomena measured at scattered points to a 3D grid that is suitable for modeling and visualization. Creating a grid surface from sample data requires the estimation of values for locations based on the sample data. There are several techniques for creating a grid surface from sample data, known also as spatial interpolation [54]. Generally, an interpolation problem finds a function that passes through the given set of discrete points. For the bivariate interpolation problem, interpolation is to find a surface: \[ z = f(x,y) \]. In real world, it is impossible to get exhaustive values of data at every desired point because of practical constraints. Thus, interpolation using only the measured data is
an ill-posed problem since it admits infinite solutions; any function passing through the observed data is a solution to the interpolation problem. As a result, it is usual to introduce a priori information about the behavior of the sample function to transform this ill-posed problem into a well-posed problem. Ground surfaces are instances of fields. The surface has one value of a variable at all points in space. They do have not only continuity properties, but also they may have cliffs and sharp breaks of slope. Therefore, the nature of the terrain surface should be considered in choosing the best interpolation technique.

Spatial interpolation techniques can be classified into exact interpolation such as inverse distance weight (IDW), spline, and kriging and approximate interpolation such as trend analysis. Exact interpolators honor the data points upon which the interpolation is based and the surface passes through all points whose values are known. On the other hand, approximate interpolators are used when there is some uncertainty about the given surface values. This utilizes the belief that, in many data sets, there are global trends that vary slowly overlain by local fluctuations, which vary rapidly and produce uncertainty in the recorded values. The effect of smoothing should therefore reduce the effects of error on the resulting surface [54]. However, in this dissertation, exact interpolation techniques are only considered to keep the original LIDAR returns. The choice of a spatial interpolator is especially important in mountainous regions where data collections are sparse and variables may change over short spatial scales.
5.2 Spatial Interpolation Techniques

5.2.1 Inverse Distance Weight Interpolation

One of the most commonly used techniques for interpolation of scatter points is the inverse distance weighted (IDW) interpolation [55]. IDW methods are based on the assumption that the interpolating surface should be influenced most by the nearby points and less by the more distant points. The interpolating surface is a weighted average of the scatter points, and the weight assigned to each scatter point diminishes as the distance from the interpolation point to the scatter point increases. The equation used is as follows:

\[
F(x, y) = \sum_{i=1}^{n} w_i f_i ,
\]

where \( n \) is the number of scatter points in the set, \( f_i \) is the prescribed function value at the scatter points, and \( w_i \) is the weight function assigned to each scatter point. The form of the weight function is:

\[
w_i = \frac{h_i^{-p}}{\sum_{j=1}^{n} h_j^{-p}} ,
\]

where \( p \) is an arbitrary positive real number, known as the power parameter, (typically, \( p=2 \)) and \( h_i \) is the distance from the scatter point to the interpolation point. The weight function varies from a value of unity at the scatter point to a value that approaches zero as the distance from the scatter point increases. The weight functions are normalized so that the weights sum to unity. The effect of the weight function is that the surface interpolates
each scatter point and is influenced most strongly, between scatter points, by the points closest to the point being interpolated. The best results from the IDW interpolation are obtained when sampling is sufficiently dense with regard to the local variation functions attempting to simulate. If the sampling of input points is sparse or very uneven, the results may not sufficiently represent the desired surface.

5.2.2 Kriging

Kriging is an interpolation method that originated in the earth sciences to fit geological data. The field of study from which it evolved is called geostatistics [56], so kriging involves statistics and random function theory. The basic idea of kriging is to use a weighted linear combination of values at the sample locations to interpolate the function. What kriging tries to do is to come up with the best linear estimator by minimizing the error of the estimation. Because the actual function is not known, the error is modeled with probability theory and then minimized, resulting in a linear system of equations for the weighting factors. Kriging uses a variogram, a measure of spatial correlation between two points, so the weights change according to the spatial arrangement of the samples.

The basic kriging theory can be described as follows. At every point having no sample, the unknown true value using a weighted linear combination of the available samples can be estimated by

$$\hat{\nu} = \sum_{j=1}^{n} w_j \cdot \nu.$$  (5.3)
The set of weights changes at different locations. The error of any particular estimated value, i.e., the difference between the estimated value and the true value, at that same location is defined as:

\[ r_i = \hat{v}_i - v_i, \quad (5.4) \]

where \( r_i \) is the error of the ith estimate. Then, the average error of a set of \( k \) estimates is

\[ m_r = \frac{1}{k} \sum_{i=1}^{k} r_i = \frac{1}{k} \sum_{i=1}^{k} (\hat{v}_i - v_i). \quad (5.5) \]

The error variance, \( \sigma^2_R \), of a set of \( k \) estimates can be written as

\[ \sigma^2_R = \frac{1}{k} \sum_{i=1}^{k} (r_i - m_r)^2 \]

\[ = \frac{1}{k} \sum_{i=1}^{k} \left[ \hat{v}_i - v_i - \frac{1}{k} \sum_{i=1}^{k} (\hat{v}_i - v_i) \right]^2, \quad (5.6) \]

where \( v_1, \ldots, v_n \) are the true values and \( \hat{v}_1, \ldots, \hat{v}_n \) are the corresponding estimates. Unfortunately, this equation cannot be used since it involves quantities of unknown values. In order to solve this problem, the model is assumed to be a stationary random function that consists of several random variables, one for the value at each of the sample locations \( V(x_1), \ldots, V(x_n) \) and one for the unknown value at the point to estimate, \( V(x_0) \). Each of these random variables has the same probability law at all locations. The expected value of the random variable is \( E\{V\} \). Any pair of random variables has a joint distribution that depends on the separation between the two points. This estimate is also a random variable at the available sample location:
\[ \hat{V}(x_0) = \sum_{i=1}^{n} w_i \cdot V(x_i). \]  

Similarly, the estimation error, defined as the difference between the estimate and the random variable, is:

\[ R(x_0) = \hat{V}(x_0) - V(x_0). \]  

By substituting Equation (5.7) into (5.8), R(x_o) can be expressed in terms of the original n + 1 random variables in this random function model. This yields the following:

\[ R(x_0) = \sum_{i=1}^{n} w_i \cdot V(x_i) - V(x_0). \]  

The estimation error of the unknown value at \( x_0 \) is an outcome of the random variable R(x_o). Though the variance of these actual errors cannot be minimized, it is possible to minimize the variance of the modeled error R(x_o).

The first task, then, is to find an expression for the variance of the model error. This error is a random variable, since it is a weighted linear combination of other random variables. The variance of a weighted linear combination can be expressed as:

\[ Var\left\{ \sum_{i=1}^{n} w_i \cdot V_i \right\} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i \cdot w_j \cdot Cov\{V_i V_j\}. \]  

Using this formula, the variance of the error can be expressed as:

\[ Var\{R(x_0)\} = Cov\{\hat{V}(x_0)\hat{V}(x_0)\} - Cov\{\hat{V}(x_0)V(x_0)\} \]

\[ - Cov\{V(x_0)\hat{V}(x_0)\} + Cov\{V(x_0)V(x_0)\} \]
\[= \text{Cov} \left[ \hat{V}(x_0) \hat{V}(x_0) \right] - 2 \text{Cov} \left[ \hat{V}(x_0) V(x_0) \right] + \text{Cov} \left[ V(x_0) V(x_0) \right] \]  \hspace{1cm} (5.11)\\

The first term of Equation (5.11), \(\text{Cov} \left[ \hat{V}(x_0) \hat{V}(x_0) \right]\), is equal to the variance of \(\hat{V}(x_0)\), which is a linear combination of another random variable; i.e.,

\[\text{Var} \left[ \hat{V}(x_0) \hat{V}(x_0) \right] = \text{Var} \left[ \sum_{i=1}^{n} w_i \cdot V_i \right] = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \bar{C}_{ij}. \]  \hspace{1cm} (5.12)\\

The third term in Equation (5.11), \(\text{Cov} \left[ V(x_0) V(x_0) \right]\), is also equal to the variance of \(V(x_0)\). If all of the random variables have the same variance, then this third term can be expressed as

\[\text{Cov} \left[ V(x_0) V(x_0) \right] = \sigma^2. \]  \hspace{1cm} (5.13)\\

The second term of Equation (5.11) can be written as

\[2 \text{Cov} \left[ \hat{V}(x_0) V(x_0) \right] = 2 \text{Cov} \left[ \sum_{i=1}^{n} w_i V_i \right] V_0 \]

\[= 2 \mathbb{E} \left[ \sum_{i=1}^{n} w_i V_i \cdot V_0 \right] - 2 \mathbb{E} \left[ \sum_{i=1}^{n} w_i V_i \right] \cdot \mathbb{E} \left[ V_0 \right] \]

\[= 2 \sum_{i=1}^{n} w_i \cdot \mathbb{E} \left[ V_i V_0 \right] - 2 \sum_{i=1}^{n} w_i \cdot \mathbb{E} \left[ V_i \right] \cdot \mathbb{E} \left[ V_0 \right] \]

\[= 2 \sum_{i=1}^{n} w_i \cdot \text{Cov} \left[ V_i V_0 \right] \]

\[= 2 \sum_{i=1}^{n} w_i \bar{C}_{ix}. \]  \hspace{1cm} (5.14)\\

Combining these three terms again, the error variance can be expressed as
\[
\tilde{\sigma}_{R}^2 = \tilde{\sigma}^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \tilde{C}_{ij} - 2 \sum_{i=1}^{n} w_i \tilde{C}_{i0}.
\]  
(5.15)

Equation (5.15) represents the error variance as a function of \( n \) variables, namely the weights \( w_1, \ldots, w_n \). The minimization of a function of \( n \) variables is usually accomplished by setting the \( n \) partial first derivatives to zero. This produces a system of \( n \) equations and \( n \) unknowns that can be solved by any one of several methods for solving a system of simultaneous linear equations. Unfortunately, this procedure is not quite correct for the minimization of \( \tilde{\sigma}_{R}^2 \) due to a constraint on the solution, i.e., any set of \( n \) weights that is considered as a solution to this minimization cannot be accepted. However, one must restrict possible solutions to those sets of weights that sum to 1. Such problems of constrained optimization can be solved by the technique of Lagrange parameters [57]. Therefore, the error variance can be expressed as

\[
\tilde{\sigma}_{R}^2 = \tilde{\sigma}^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \tilde{C}_{ij} - 2 \sum_{i=1}^{n} w_i \tilde{C}_{i0} + 2 \mu \left( \sum_{i=1}^{n} w_i - 1 \right),
\]  
(5.16)

where \( \mu \) is the Lagrange parameter that is used to convert the constrained minimization problem into an unconstrained one. In order to get the minimum variance of the error, the partial first derivatives of Equation (5.16) are taken with respect to \( w \) and setting the result to zero. Thus, all of the weight \( w_i \) can be represented by

\[
\sum_{j=1}^{n} w_j \tilde{C}_{ij} + \mu = \tilde{C}_{i0}, \text{ for each } i, 1 \leq i \leq n.
\]  
(5.17)

Now that each weight \( w_j \) can be obtained from Equation (5.17). After all the values are obtained, the value located at \( x_0 \) can be then estimated using Equation (5.7).
5.2.3 Spline Interpolation

Spline interpolation is a general-purpose method that fits a minimum curvature surface to the sample points [58, 59]. Because it generates smooth surfaces, the Spline method is better suited to the sample data that vary gently, i.e., it is not appropriate to use if there are large changes in the data values over short distances. Because the IDW method is a weighted average method, no cell receives a value that lies outside the range of the minimum and maximum sample point values. This means that extreme natural formations like ridges and valleys cannot be created unless they have been adequately sampled. Unlike the IDW method, a surface created with the spline method always has the exact value of a sample point at the corresponding surface location. However, it produces a larger range of values than the ones existed in the original dataset. It also produces a smooth surface because it minimizes the curvature.

The minimum curvature method is an old and ever-popular approach for constructing smooth surfaces from irregularly spaced data [59]. The minimum curvature surface corresponds to the minimum of the Laplacian power or, in an alternative formulation, satisfies the biharmonic differential equation. Physically, it models the behavior of an elastic plate. In most of the practical cases, the minimum curvature method produces a visually pleasing smooth surface. However, in cases of large changes in the surface gradient, the method can create strong artificial oscillations. Switching to lower order methods, such as minimizing the power of the gradient, solves the problem of extraneous inflections but also removes the smoothness constraint and leads to gradient discontinuities [51]. It is pointed out that the optimum properties of the spline fit can be
obtained by solving the following differential equation, which is equivalent to a third-order spline [59]:

\[
\begin{pmatrix}
\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \\
\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}
\end{pmatrix}
\begin{pmatrix}
\frac{\partial^2 u}{\partial x^2} \\
\frac{\partial^2 u}{\partial y^2}
\end{pmatrix}
= 0,
\] (5.18)

where \(u\) is the interpolating function. The boundary conditions impose that \(u(x_n, y_n) = g_n\), the \(n\)th observation, and that between the edge of the interpolating region and the observations, the interpolating function will tend to a plane as the interpolating region becomes larger. A function \(u(x, y)\) that minimizes the total squared curvature can be expressed as:

\[
C(u) = \iint \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)^2 \, dx \, dy.
\] (5.19)

This function obeys Equation (5.18), and conversely, if a function \(u\) obeys Equation (5.18), it minimizes \(C(u)\). Then, the discrete total squared curvature can be expressed as:

\[
C = \sum_{i=1}^{I} \sum_{j=1}^{J} \left( u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} \right)^2 / h^2,
\] (5.20)

where \(h\) is the grid spacing in both the \(x\) and \(y\) directions and \(u_{i,j}\) is the interpolated value at grid point \((i, j)\). From the necessary conditions to minimize \(C\), the following system of linear equations can be obtained for the observation points coinciding with a grid point:

\[
\begin{align*}
& u_{i+2,j} + u_{i+1,j} + u_{i,j+2} + u_{i,j+1} + 2(u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) \\
& - 8(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) + 20u_{i,j} = 0, i = 1, \ldots, I, \ j = 1, \ldots, J
\end{align*}
\] (5.21)

For observation points not on a grid point, an approximate expression for \(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\) at a given point \((x_o, y_o)\) is obtained by expanding \(u\) using Taylor’s series around
(x₀, y₀) and Equation (5.20). Another system of equations is then obtained from the necessary conditions to minimize C. Solving the above mentioned linear systems of equations leads to the interpolated values uᵢⱼ, for i = 1, …, I, and j = 1, …, J.

5.3 Triangulated Irregular Networks

The topographic surface can be considered to be a mathematically continuous surface, but mass points interpolated using a spatial interpolation technique are not continuous. Therefore, mass points are the predominant input to a triangulated irregular network (TIN) and forms the overall shape of the surface such that the terrain surface can be visualized.

A TIN is a digital terrain model that is based on an irregular array of points which form a sheet of non-overlapping contiguous triangle facets [60]. The simplest interpretation of the model, in terms of defining a continuous surface, is a linear approach and is based on the principal that a flat plane can be fit to any three non-collinear points [61]. Thus, areas of consistent slope and aspect are represented in the TIN by individual triangles. Although TINs can be expensive to build because the accurate source data is expensive to collect, TINs have several advantages to use. The most important is that TIN models adapt to the variable complexity of the terrain. Where there is little variation in the surface, only a small amount of data are stored, and where there is much variation, more data are stored. TINs also maintain the original input data. Mass points, breaklines, or other features, used as input to TIN, are exactly maintained in the TIN. TINs are also
the native structure used for 3D graphics rendering, and, therefore, can be used to improve the performance of interactive 3D visualizations [8].

5.4 Summary

Spatial interpolation is a technique for creating a grid surface from sample data. There are two classes of spatial interpolation: exact and approximate. However, to improve the accuracy of DTMs, exact interpolators such as IDW, spline, and kriging honor the data points upon which the interpolation is based. The surface passes through all points whose values are known are considered. Since no single interpolation technique is the best for all situations, the characteristics of the input sample data and the surface being modeled should be considered. Since the ground surface is generally smooth, the spline method is better suited to sample data that varies gently and can generate a smooth surface. Furthermore, the interpolated data points are used as input to a TIN to visualize the ground surface.
The theory of the foregoing terrain model generation algorithm has been described in detail. In this chapter, the implementation and effect of this technique for extracting quality DTM is investigated using test data sets for different areas. Then, a comparison with some of the existing techniques is made to evaluate the performance of the presented technique. Both visual and quantitative comparison is performed.

6.1 Experimental Results

6.1.1 Test Data Description

Two sites corresponding to three different test data sets are used in this investigation. The first site represents commercial forestland owned by the Trillium Corporation, near Bellingham, in northwest Washington State. The second site represents the Starr Forest.

The first site has pronounced variable terrain with a range of canopy densities and heights. Collected by EarthData, two LIDAR data sets, representing two different areas, dense and less dense forestry areas, were provided. Each pulse shot may have one or multiple returns depending on the reflected surface. For instance, a first return is produced as a result of a reflection of an outgoing pulse from a top of a tree. The remainder of the pulse continues downwards. Part of it may then be reflected from
somewhere within the canopy and produces a second return. In a similar manner, multiple returns can be detected. Shots are then tagged to indicate that returns from 2, 3, or 4 reflecting surfaces are captured. The data associated with the last return are used since they are likely to contain terrain points.

The second site represents the Starr Forest. Collected data sets from this site consist of a LIDAR return data set and a validation tree location data set. Since a ground trend model can be extracted from a validation data set, the possibility of a ground trend model developed from specific tree height measurements is verified.

The DTM images presented here were obtained using the ERDAS Imagine Virtual GIS Tool. All the images are geo-referenced to a UTM projection system. The spheroid is GRS1980, zone number is 10, and datum is NAD83. Note that the viewing information is as follows: Field of View (FOV) is 50°, pitch is -29°, and azimuth angle is 2°.

6.1.2 Bare Earth Model Extraction

The applicability of the presented method of terrain model generation from a dense forestry area data set is illustrated first. Figures 6.1 and 6.2 show the images produced from the last return raw data of the dense area site as well as the bare earth image model generated by EarthData extracted from the last return using automated and manual editing. Note that the collection of the bare earth model was achieved by passing a search window over the field area to acquire LIDAR points from all returns (first, second, etc.). Some points were regarded as points associated with the terrain surface and labeled as being “ground” points. Others were labeled as “features” above the ground.
level. This process was repeated until all points in the field area were classified and a representation of the ground surface was obtained. Manual editing was then carried out to remove noise or unordinary high points in area of dense canopy where the last returned is resumed well above the true ground surface [62].

Figure 6.1. DTM of the last return data - Bellingham site, dense forest canopy area

Figure 6.2. EarthData Bare earth model- Bellingham site, dense forest canopy area
A step-by-step implementation of the foregoing algorithm was performed next. The corresponding results follow. As a first step of the algorithm, the histogram-based clustering technique is performed to separate the measured data into terrain and non-terrain points. A two-dimensional x-y window is used to process this task. Histograms of each window can be calculated. Figure 6.3 illustrates a histogram plot over a 4x4 window.

![Figure 6.3. Histogram for the data in an x-y window (window size 4x4)](image)

A kernel density estimation (KDE) is performed next to obtain an estimate of the probability density function through the histogram. The purpose of this step is to decide whether the density function of the data satisfies a multi-modal, skewed, normal, or uniform distribution. As stated in Chapter 3, a threshold index that classifies the LIDAR data into ground and non-ground data can be determined by a KDE. Figure 6.4 illustrates the KDE threshold selection based on the histogram of Figure 6.3. It indicates that a local
minimum point between the two local maximum points is a threshold index that classifies the LIDAR data into ground and non-ground data. Based on this information, the result from the histogram-based clustering technique is illustrated in Figure 6.5. This figure indicates that many of the non-ground points are eliminated, thus keeping most of the assumed ground points.

![Threshold Selection Method of Kernel Density Estimation](image)

*Figure 6.4. Threshold Selection Method of Kernel Density Estimation*
The next stage to the implementation of this method is the ground modeling stage. The purpose of this stage is to further eliminate non-ground points based on both the distance and surface gradient in conjunction with a statistical test to determine the existence of outliers in the measured data. Figures 6.6 and 6.7 show the results of the elevation and the gradient outlier detection steps. These figures indicate that this process does further eliminate non-ground and outlier-like points. Note that the elevation and gradient outlier detection steps are performed back to back. Accordingly, the output of the first stage (elevation) is used as input to the second one (gradient) and the input to the first stage is the resulting output obtained from the histogram clustering of Figure 6.5.
Figure 6.6. LIDAR Points obtained from the Elevation Outlier Detection Algorithm

Figure 6.7. LIDAR Points obtained from the Gradient Outlier Detection Algorithm
In addition, a statistical test is used in conjunction with the elevation and gradient detection steps for residual truncation. Figure 6.8 shows the use of the AIC criterion for residual truncation. According to this figure, the truncation number is determined to be around 40 since this value corresponds to a minimum AIC. Also, this technique is applied to determine the most influential neighbor points over eight neighbor points. The result of the “verification” stage using the AIC based truncation technique is shown in Figure 6.9.

![Figure 6.8. The AIC Threshold Selection Technique](image-url)
The final stage of this technique is spatial interpolation. As addressed in Chapter 4, the performance of several spatial interpolation techniques is investigated. Figures 6.10, 6.11, and 6.12 illustrate the bare earth images obtained from performing IDW, kriging, and spline interpolation techniques, respectively. Visually, these figures indicate that IDW interpolation produces rough terrain among all three techniques, whereas kriging interpolation causes a few spurious peaks. On the other hand, the image of spline interpolation shows a smoother ground-like image, close to the bare earth ground model of Figure 6.2.
Figure 6.10. Terrain Image obtained from IDW Spatial Interpolation

Figure 6.11. Terrain Image obtained from Kriging Spatial Interpolation
It is also of interest to compare the performance of the presented technique with some existing techniques such as linear prediction, modified linear prediction, and adaptive smoothing. The resulting DTM images obtained as a result of implementing these techniques are illustrated in Figures 6.13 - 6.16. Note that, for the adaptive implementation, the step size, the filter lengths are set to 0.9 and 50 with delays of 10 and 20, respectively.
Figure 6.13. DTM estimated from Linear Prediction

Figure 6.14. DTM Extraction from Modified Linear Prediction
Figure 6.15. DTM Extraction from Adaptive Smoothing, Delay = 10

Figure 6.16. DTM Extraction from Adaptive Smoothing, Delay=20
The terrain images of Figures 6.13 and 6.14, obtained from the linear prediction and the modified linear prediction, indicate that they cannot track valleys very well, whereas the terrain images of Figures 6.15 and 6.16, obtained from adaptive smoothing, are closer to the bare earth of Figure 6.2. However, they do also suffer from spurious peaks and valleys. Based on visual comparison, it can be seen that DTM extraction obtained from the presented technique are more robust than the ones obtained from linear prediction, modified linear prediction, and adaptive smoothing.

Similar processing has been performed on LIDAR data collected from the Bellingham site with less dense forestry area. However, only quantitative performance assessment is presented in section 6.2 of this chapter.

6.1.3 Statistical Analysis for Ground Trend Model

The second site that was investigated represents the Starr Forest. Collected data sets from this site consist of raw LIDAR points for an area of trees measured (spacing trials), tree heights from canopy model on validated ground trees, and tree locations and field heights. Through this simulation, the possibility of using a ground trend model developed from specific tree height measurements is discussed. The procedure of the test is to generate a ground surface with the presented algorithm and a canopy surface afterward. Validated tree locations can be used with the canopy surface to fix a ground trend by subtracting the field-measured heights from the canopy surface. At each known tree location, two heights are given: the one from the predicted ground model and the one derived from subtracting the tree height from the canopy surface. The two measurements
are paired observations that can be used in statistical analysis, regression analysis seems to be appropriate, to examine the relationship between them. If a regression analysis model is used, the strength of the relationship is given by the R-squared value in the model. A high R-squared value indicates a close similarity between the trend given by the ground surface and the trend given by the field validated tree heights. The reason for this approach is to be able to validate ground surface models in areas where there are little or no surveyed points that can be used (under tree cover, remote areas, etc.). The model of this approach is shown in Figure 6.17.

Figure 6.17. Modeling for Statistical Linear Regression Analysis
As stated in the previous paragraph, the two paired ground surface data sets are obtained from the presented algorithm and the given measurement data set. With the paired sets, statistical regression analysis is performed. Performing this analysis, a 0.63 R-squared value is obtained. This indicates that the LIDAR ground surface obtained from the presented algorithm is related to true ground surface.

In addition, assuming a linear regression model, \( Y = AX + B \), where \( Y \) is a linear ground surface (LGS), \( X \) is ground trend surface (GTS\(_{ht}\)), \( A \) is a slope, and \( B \) is an intercept from Figure 6.17, the values of \( A \) and \( B \) can be obtained from following equations.

\[
A = \frac{\sum_{i=1}^{n} X_i Y_i - \left(\sum_{i=1}^{n} X_i\right) \left(\sum_{i=1}^{n} Y_i\right)}{n \cdot \sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2} = 0.267
\]

\[
B = \bar{Y} - A \cdot \bar{X} = 51.80
\]

where \( n \) is the number of samples, 102.

### 6.2 Accuracy Measure

In general, error quantification on a digital elevation model can be obtained by comparing a set of known altitude values with the corresponding homologous points of the model using conventional statistical tests. Error quantification in the DTMs is normally carried out by calculating the root- mean-squared error or \( R_{MSE} \) through the following expression [8]:
where \(z'(i)\) is the altitude of the estimated point, \(z(i)\) is the "real" or reference altitude taken from an accurate data source, and \(N\) is the number of points used in this comparison.

In addition to the \(R_{MSE}\), the absolute mean deviation, \(d_{\text{mean}}\), and the error standard deviation, \(d_{\text{std}}\), are also used as accuracy measures. The altitude error on a point \(i\), \(d(i)\), is defined as the absolute difference between \(z(i)\) and \(z'(i)\), i.e., \(d(i) = |z(i) - z'(i)|\). Accordingly, the absolute mean error or mean deviation, \(d_{\text{mean}}\), for a set of data is defined as:

\[
d_{\text{mean}}(i) = \frac{1}{N} \sum_{i=1}^{N} d(i),
\]

and the error standard deviation, \(d_{\text{std}}\), is calculated via the following expression:

\[
d_{\text{std}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (d(i) - d_{\text{mean}})^2}.
\]

In general, a visual comparison alone is not sufficient. A better understanding of DTM extraction and their suitability requires testing of their accuracy. Accordingly, it is also of interest to perform a quantitative comparison as a measure of accuracy. For the comparison, the RMS errors are calculated. In addition, the height differences between the two DTMs for the surveyed-check points can be analyzed using statistical absolute mean and standard deviation to determine the error as well. For the data set corresponding to the dense forestry area, the results of the quantitative comparison for all
DTMs generated by various techniques are presented in Table 1. In this table, “Delay 10” and “Delay 20” indicate the delay values used in the implementation of the adaptive linear prediction technique, and “MLP” and “LP” denote an abbreviation for the modified linear prediction and linear prediction techniques, respectively. Also, in terms of accuracy measurement, $A_m$ and $A_{std}$ denote the absolute mean and absolute standard deviation, respectively. Careful examination of the tabulated results reveal that adaptive smoothing and the presented method yield close $R_{MSE}$ values. However, the overall quantitative performance assessment results indicate that the presented technique produces superior quality DTMs over the existing method.

<table>
<thead>
<tr>
<th></th>
<th>Delay 10</th>
<th>Delay 20</th>
<th>MLP</th>
<th>LP</th>
<th>IDW</th>
<th>Kriging</th>
<th>Spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>5.34</td>
<td>6.98</td>
<td>8.06</td>
<td>10.41</td>
<td>5.84</td>
<td>5.46</td>
<td>5.05</td>
</tr>
<tr>
<td>$A_m$</td>
<td>4.55</td>
<td>5.46</td>
<td>6.09</td>
<td>8.21</td>
<td>1.04</td>
<td>1.03</td>
<td>0.96</td>
</tr>
<tr>
<td>$A_{std}$</td>
<td>2.88</td>
<td>4.48</td>
<td>5.44</td>
<td>6.59</td>
<td>1.01</td>
<td>0.85</td>
<td>0.78</td>
</tr>
</tbody>
</table>

(unit: cm)

A similar type analysis is performed on LIDAR data collected from the Bellingham site with less dense forestry area. Table 2 shows the results obtained from the presented method with different interpolation schemes.

<table>
<thead>
<tr>
<th></th>
<th>IDW</th>
<th>Kriging</th>
<th>Spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>5.84</td>
<td>5.46</td>
<td>5.05</td>
</tr>
<tr>
<td>$A_m$</td>
<td>1.04</td>
<td>1.03</td>
<td>0.96</td>
</tr>
<tr>
<td>$A_{std}$</td>
<td>1.01</td>
<td>0.85</td>
<td>0.78</td>
</tr>
</tbody>
</table>

(unit: cm)
6.3 Computational Complexity

Although the presented method produces DTMs with better accuracy, it is, in general, computationally extensive when compared to the linear prediction based method. The computational costs are analyzed and compared in terms of: 1) analytical assessment and 2) experimental benchmarking.

6.3.1 Analytical Assessment

The presented method requires two distinct stages where histogram-based clustering and geometric modeling steps are performed. Let \( N \) be the size of the input data. At the histogram-based clustering, the operations required are \( N/M \) per an \( x \)-\( y \) window, where \( M \) is the window size. Therefore, the total number of operations for the clustering is \( O(N) \). At the input of the geometric modeling, the data size will be reduced to \( N_1 \), where \( N_1 < N \). The operations required for the elevation outlier detection step are the selection of the eight nearest neighbor data points, sorting those data, and detecting existing outliers per a sample point. That is, since the operation order per a sample is \( O(2N_1 + 2N_1 \log N_1) \), the total number of operation for this step is \( O(N_1^2 \log N_1) \). Also, at the next gradient outlier detection step, the data size \( N_1 \) will be further reduced to \( N_2 \), where \( N_1 \leq N_2 \). Similar to the elevation outlier detection step, the total number of operation has the same order with sample size \( N_2 \), i.e., \( O(N_2^2 \log N_2) \). A further reduction at the output of the gradient outlier detection step is obtained. Thus, the input data size of the final “verification” step will be \( N_3 \), where \( N_3 \leq N_2 \). The order of the operation for the
The computational order for an initial ground approximation is $O(N)$, where $N$ is the size of the data samples. For the following steps, such as residual calculation, filtering using a special weighting function, and extracting matched points, the required order is $O(K_1)$ since the iteration depends on a predefined constant $K_1$. Finally, the number of iterations of the algorithm depends on a penetration rate $K_2$. Therefore, the overall order of complexity is:

$$O_{LP} = O(K_2(N+K_1)) = O(N).$$  \hspace{1cm} (6.6)

### 6.3.2 Experimental Benchmarking

An experimental assessment of the computational cost of the two methods is performed. The data size of the assessment has a length of $N = 6294$ and this experiment was performed on a 333MHz Pentium II CPU. Table 3 shows the experimental results of the computational expense.

<table>
<thead>
<tr>
<th>Table 3. Computational Time Comparison</th>
<th>Linear Prediction based algorithm</th>
<th>Presented Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational Time</td>
<td>20.39 [s]</td>
<td>218.35 [s]</td>
</tr>
</tbody>
</table>
The results presented in Table 3 indicates clearly that linear prediction based methods are, in general, faster to execute. However, in terms of accuracy, the presented method is more robust in extracting digital elevation models from LIDAR returns. Thus, there is a trade-off between computational time and accuracy.

6.4 Summary

In this chapter, results for DTM extraction obtained from the foregoing algorithm are presented. Both visual and quantitative comparison with existing methods, such as linear prediction, modified linear prediction, and adaptive smoothing, are also methods to illustrate the validity of the presented method. It is clearly seen that the presented method is more robust in terms of the accuracy measures presented. However, there is a trade-off between accuracy and computational complexity, i.e., linear prediction based methods are faster to execute.

In addition, a statistical regression analysis is performed to measure LIDAR ground surface relationship to true ground surface assuming that the ground trend surface is an approximation of the true ground surface. The result indicates that the predicted ground surface is closely related to the true ground surface.
CHAPTER VII

CONCLUSIONS

7.1 Summary

This dissertation has addressed the generation of high quality digital terrain models (DTMs) using light detection and ranging (LIDAR) systems. An algorithm that consists of several processing steps has been developed to the generation of DTMs. The simulation results for test sites are compared visually and quantitatively to illustrate the applicability and validity of the algorithm. Also, the computational complexity of the proposed algorithm is addressed. A significant contribution of this dissertation is the accuracy of DTMs that uses information such as elevation and slope between nearest neighbor data sets.

This technique is based on a preprocessing step, ground modeling, and interpolation. In the preprocessing step, a histogram-based clustering technique is introduced to eliminate much of the non-terrain points. In this step, a clustering technique based on unsupervised classification is introduced. That is, a histogram-based clustering technique using the height of data points has been implemented since traditional Euclidian distance-based clustering techniques such as k-means clustering and self-organizing feature map (SOM) fail to produce the desired classification result. The main point of this clustering algorithm is a selection of a threshold index, which can divide the returned LIDAR data samples into ground points and non-ground points. The idea of the
threshold selection method is to use the estimation of a distribution function inherent to the given data. If return surfaces contain several distinct features such as trees, building, and ground, the data are likely to show multi-modal distribution. But, if the surface lies in a plain area, the return data samples are likely to show uniform distribution. Therefore, the threshold selection method considers the possibility of several distributions. In this dissertation, distributions are estimated through kernel density estimation (KDE) since histograms have several limitations such as the dependency of bin-width and a starting point. Actually, KDE is a smoothed histogram. However, since the performance of KDE generally depends on bin-width, an adaptive selection technique for the bin-width is successfully applied to estimate distributions, from which a threshold index can be chosen. That is, a local minimum between two local maximum can be found from the estimated density function. Thus, the procedure of the threshold selection performed a normality test, KDE, and finds a local minimum index. The reason for the normality test is to reduce computational complexity of the method. If data samples pass the normality test, the data can produce a normal distribution, which is a unimodal distribution. In this case, further processing is not needed since no threshold selection is necessary. Otherwise, the remaining procedures are performed. The data that fail the normality test can show a multi-modal distribution, uniform distribution, or skewed distribution. Since uniform and skewed distributions are a class of unimodal distribution, no threshold selection procedure is processed for the distributions. In this dissertation, if a local minimum point between local maximum points can be found from KDE, the data samples contain a
multi-modal distribution. The local minimum index corresponds to a threshold index. Otherwise, the data samples show a skewed or uniform distribution.

In the geometric modeling step, information such as elevation and slope between nearest neighbor points is extracted. Since the non-terrain points over the terrain points can be considered as multiple outliers, the algorithm corresponds to outlier detection process. In this algorithm, eight nearest neighbor points in terms of the Euclidean distance are selected to determine residual and gradient indexes for elevation and slope, respectively. Those indexes are investigated for a constructed 95% confidence interval. That is, if those indexes for each data point are outside the confidence interval, it is processed as an outlier. Otherwise, it can be a ground point. This procedure is continued until all data points are visited. As a final stage of this processing, a “verification” stage is implemented. This step can further eliminate the remaining outliers that may occur in the previous stages. The idea of this technique is that each point is investigated for its deviation from a global trend. That is, a leave-one-out test is performed for all points. During leave-one-out tests, residuals – the difference between a query point and a predicted point – can be calculated for which largest residuals are considered as the remaining outliers. At this point, an important issue is to determine how many large residuals can be considered as outliers in order to be discarded. An approach is to assume the percentage of the outliers, but the subjective selection is not desired. To solve this issue, an information theoretic criterion (AIC) based on statistical selection is effectively used. Thus, using the AIC technique, the automatic selection of the number of outliers over neighbor points is possible.
Finally, spatial interpolation is applied to acquire the topographic surface. The interpolation techniques such as IDW, kriging, and spline are investigated and the DTM images obtained from these interpolation techniques are compared with existing techniques such as linear prediction, modified linear prediction, and adaptive linear prediction.

Experiments are performed for several test data sets. The results indicate that the proposed technique can improve the quality of DTMs over existing techniques. The resulting DTM images obtained as a result of implementing the linear prediction, modified linear prediction, as well as adaptive filtering do suffer from spurious peaks and valleys. On the other hand, the images obtained from the proposed technique indicate that the images are not significantly different from the true DTM. In general, since only visual comparison is not sufficient and a better understanding of DTM extraction and their suitability requires testing of their accuracy, a quantitative comparison is also performed as a measure of accuracy. For this comparison, the RMS errors are calculated and the height differences between the two DTMs for the estimated-check points are analyzed using absolute mean and absolute standard deviation to determine the error as well. The results of the quantitative comparison for all DTMs generated by various techniques are presented and these results indicate that the proposed technique produces superior quality DTMs over the existing methods. Furthermore, the possibility of using a ground trend model, developed from specific tree height measurements, is also investigated. For this analysis, a statistical regression analysis model is used. Performing this analysis, a 0.63
R-squared value is obtained. This result indicates that the LIDAR ground surface obtained from the presented algorithm is related to the true ground surface.

In this dissertation, the computational cost of the proposed algorithm is also analyzed in terms of analytical assessment and experimental benchmarking. It is shown that the order of complexity of the proposed method is \( O_{\text{proposed}} = O(N + N_1^2 \log N_1 + N_2^2 \log N_2 + N_3) \), compared to \( O(N) \) of the linear prediction. Although the proposed technique is shown to produce DTMs with better accuracy, it is computationally expensive in comparison with linear prediction methods. Note that using high-level software languages such as C and C++ that provide faster processing speed can compensate for the computational time limitation of this method.

7.2 Dissertation Contributions

The major contribution of this dissertation is the development of an innovative technique to extract ground elevation models collected by small-footprint LIDAR systems. The basic idea of this technique is that the closer data points tend to be, the more correlated they are for spatial data modeling like terrain modeling. In general, existing algorithms are shown to approximate ground surfaces while keeping the outliers that affect the quality of DTMs. However, most of the outliers can be eliminated with this method. Thus, robustness in the accuracy can be reached. Simulation results have shown an improvement over existing techniques and the proposed technique can generate high quality DTMs in even dense forestry area.
Also, the performance of all of existing techniques depends on several predefined
variables. But, there is a trade-off between accuracy and computational complexity when
this technique is compared directly with the linear prediction method. Although the linear
prediction method is faster, the proposed technique has shown approximately 51% improvement in the RMSE and 88% in both mean and standard deviation.
REFERENCES


pp. 277-288.


