OUTLIER DETECTION AND DATA FILTERING IN LIDAR DATA WITH MULTIPLE ATTRIBUTES

GANG PANG

M.Phil

The Hong Kong Polytechnic University

2011

The Hong Kong Polytechnic University
Department of Land Surveying & Geo-Informatics

Outlier Detection and Data Filtering in LiDAR Data with Multiple Attributes

Gang Pang

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

December 2010

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

____Gang Pang____(Name of student)

ABSTRACT

Outlier detection and data filtering are two research topics in the area of LiDAR data processing, and have attracted lots of research attentions in recent years. The former one is considered as an essential preprocessing step for overall LiDAR data filtering and modeling, while, the latter one is necessarily required in the step of digital elevation model (DEM) generation. However, both the two issues always face great challenges in the automatic data processing. For outlier detection, it has proven to be surprisingly difficult to automatically remove low outliers in form of clusters. While, for data filtering, it has also suffered from great difficulties, especially in urban areas. Literature reviews demonstrate that most existing algorithms for both issues are mainly focusing on the analysis of single attribute: either the height or spatial neighborhood relationship information for outlier detection, and only geometrical information for data filtering. However, since the charactertics of outliers in LiDAR data are both single points and also clusters with elevations, either much higher or lower than the surrounding points, to effectively remove both types of outliers, it is necessary to analysis both the height and spatial neighborhood relationship information together. In parallel, since the LiDAR system simultaneously provides not only geometric information which mainly refers to height data but also radiometric information which mainly refers to intensity data, both of the two data describe the same features geometrically, to separate terrain points and off-terrain points, it is suggested that the comprehensive utilization of both two data may be advantageous over using either data individually. Thus, to fit the requirements of multiple attributes data processing for both issues, the minimum covariance determinant

(MCD) based multiple attributes model is proposed in this study which extends traditional data processing methods from single attribute to multiple attributes, from one dimension to multiple dimensions.

Specially, for outlier detection, we define the connectivity based outlier factor (COF) which indicates the spatial neighborhood relationship of a point to its neighbors as an attribute; then the COF attribute and the height attribute are extracted to organize a 2-D space, in the formed 2-D space, the proposed MCD-based multiple attributes model is conducted to identify outliers. Two typical experimental data are used of evaluating the performance of the proposed method. Comparative results by using the COF, Height, the proposed "COF + Height" and other eight representative algorithms are generated and analyzed. The result shows that the newly developed method can highly detect most outliers in both forms: individual and cluster. For data filtering, similar as removing outliers, both the height and intensity attribute are extracted to organize a 2-D space; in the formed 2-D space, the proposed model is conducted to separate terrain points and off-terrain points. Typical experimental data are utilized for checking the performance of the proposed method. Both quantitative and qualitative assessments of the results are carried out. By comparing with eight representative methods at the ISPRS filter test, it shows that our method is fair by minimizing the Type II error. In which, Type II error in our method ranks at about top 3 of every sample region with others, and simultaneity, Type I error and Total error ranks at a middle level.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the encouragement, support and help from many people in many different ways. Here, I would like to address my great appreciation to all of you:

First and foremost, particular thanks to my supervisor, Professor Wenzhong Shi, for his continued encouragement and invaluable suggestions during this work. Professor Wenzhong Shi supported me throughout my thesis with his patience and knowledge and provided great deal of valued and vital ideas and information to me. Without him, this thesis would not have been completed.

I am very grateful to Mr. Hua Zhang of China University of Mining and Technology, for his useful suggestions during his period of stay at the Department of Land Surveying and Geo-Informatics.

I would like to thank Mr. Fei Xiao, for the valuable discussions and help at programming.

I would like to thank Mr. Yiliang Wan for his patients of helping at thesis editing during his period of stay at the Department of Land Surveying and Geo-Informatics.

Many thanks to the staff, colleagues and students in the Department of Land Surveying and Geo-Informatics of the Hong Kong Polytechnic University especially for Professor Wu Chen and Bo Wu, for their concerning advice and comments in my confirmation presentation.

Finally, I am most indebted to my parents; my father and my mother, for their unfailing love, encouragement and support given me in all the years to enable me complete my studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 Introduction

1.1 Background

Airborne Light Detection and Ranging (LiDAR) has emerged as an active remote sensing system with rapid developments since 1990ies, especially in recent years. Aiming at producing various accurate digital terrain products, airborne LiDAR is typically defined as the integration of Global Positioning System (GPS), Inertial Navigation System (INS) and Laser into a single system mounted on certain airborne platforms capable of acquiring dense point measurements with three-dimensional coordinates. Laser pulses of light towards objects of interested are transmitted actively by a laser scanner sensor, the sensor then receives the light that is scattered and reflected by the objects (Liu, 2008). The time interval taken for the light to transmit from, and return to the sensor is recorded. And the distance between the LiDAR sensor and the object can be calculated by multiplying the speed of light by the time interval (Watkins, 2005; Weitkamp, 2005). In the post-treatment process of the LiDAR system, such distance combines with the attitude information recorded by INS and the aircraft flight path information provided by GPS, to give the three-dimensional spatial coordinates (x, y, z) of each laser point (see **Figure1.1**). Besides the geometric information generated above, simultaneously, the intensity data which refers to the radiometric information is provided as well. Being as the measurement of an object's reflectance, such data is regarded as the other major information offered by LiDAR.

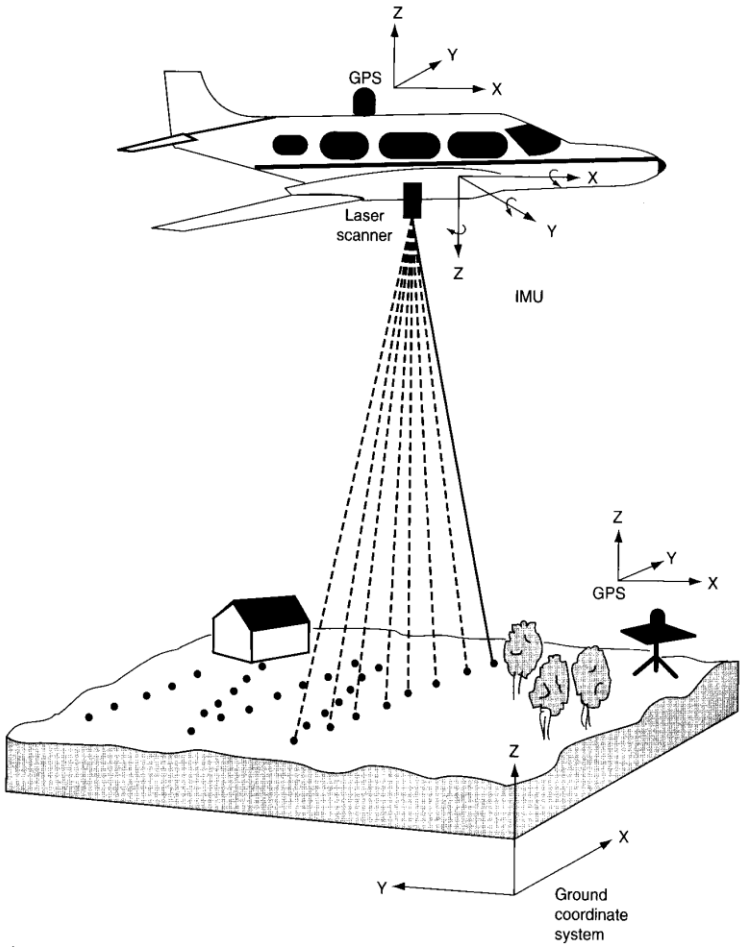


Figure 1.1 Components of an airborne LiDAR system (reproduced from Lillesand & Kiefer 2008: pp.717)

During recent years, with the advancements in commercially available GPS and Inertial Measurement Unit (IMU), LiDAR has performed as a robust technique for high accuracy in the survey of terrestrial landscapes (Bretar et al., 2003). By using LiDAR, the higher resolution, the more detailed surfaces generated. Comparing with traditional surveying and mapping systems, e.g., photogrammetric systems, LiDAR acts more directly, efficiently and accurately when measuring terrestrial landscapes (Shan and Sampath, 2005). The obtained measurements behave as 3-D point clouds, which include terrain

points for bare earth and off-terrain points for vegetation such as trees and objects such as buildings, bridges, power lines, and towers. Such measurements become a major source of digital terrain information (Raber et al., 2007), which are widely used to generate DEMs, produce land classification, achieve building extraction (3-D reconstruction) and city modeling. To the extent, LiDAR has even taken the place of traditional photogrammetric approaches, and is extensively used in many European countries (Vosselman, 2000; Schickler and Thorpe, 2001; Elmqvist et al., 2001). Among these applications, DEM generation is regarded as one of the most major usages of LiDAR data. LiDAR-derived DEMs offer advantages over ones based on traditional methods. The advantages mainly refer to high accuracy, high resolution, and more details. On the one hand, LiDAR data has a very high vertical accuracy, which enables the LiDAR system not only represent the earth surface with very high accuracy, but also has the potential to produce DEMs with acceptable horizontal resolution. On the other hand, LiDAR inherently scans the entire surface that whatever reflects the laser pulse to produce very detailed terrain surface information with mass points, by contrast, it would cost much more time to produce an equivalent amount of detail manually. In this study, we focus on two important steps for DEM generation, which are *outlier detection* and *data filtering*.

1.2 **Problem statement**

To generate DEMs from raw LiDAR data, terrain points should be picked out from LiDAR point clouds. Such processing step to separate LiDAR points into terrain points and off-terrain points is recognized as LiDAR data filtering. Since LiDAR inherently scans the entire terrain surface that whatever reflects the laser pulse with mass points,

the work of removing all above-ground features is always challenging. Filtering and manual editing in raw LiDAR data to produce a clean bare-earth surface can be always time-consuming and labor-intensive. Therefore, automatically filtering is essentially required. However, in practice, it has been proven that automatically filtering is surprisingly crude. Outliers or large areas within various terrain features such as discontinuity or steep slopes may lead the filtering failure (Sithole and Vosselman, 2003).

1.2.1 Problems in outlier detection

Outliers appearing in LiDAR point clouds can be both single points and also clusters with elevations, either much higher or lower than the surrounding points. High outliers (see **Figure 1.2**) are points that normally do not belong to the landscape, in that they originate from hit off objects such as birds, low flying aircraft (Sithole and Vosselman, 2004). And the high outliers are always treated as positive outliers (Kobler et al., 2007; Höhle, 2009; Forlani et al., 2006). Like high outliers, low outliers (see **Figure 1.3**), in a different way, are not normal parts of the landscape. However, low outlier points have a different origin. They originate from multi-path errors and errors in the laser range finder (Sithole and Vosselman, 2004). And the low outliers are always treated as negative outliers (Kobler et al., 2007; Höhle, 2009; Forlani et al., 2006).

Since many of the filtering algorithms work on the assumption that a lowest point must belong to the terrain points, similarly, the relative higher points must belong to the off-terrain points, however, in cases where the lowest point is an outlier, the assumption is totally wrong. Same cases may happen where the higher points are outliers. Therefore, it is very important that the outliers are removed before DEM generation.

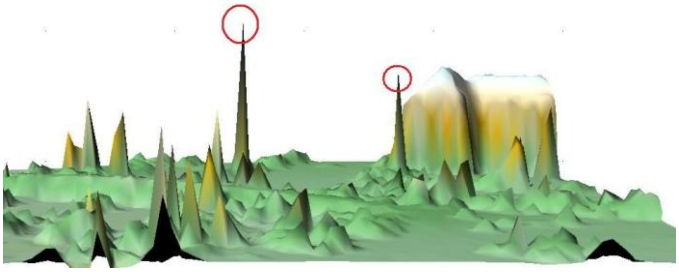


Figure 1.2 High outliers in laser data (marked in circles)

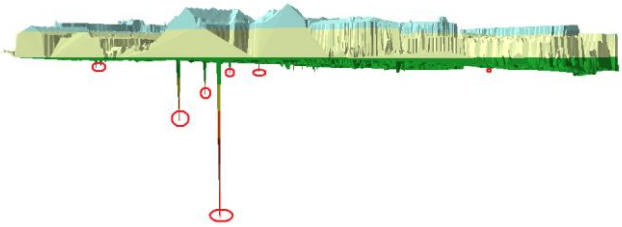


Figure 1.3 Low outliers in laser data (marked in circles)

Outlier detection issue has been frequently discussed in the LiDAR-drived DEM quality control and accuracy assessment (Höhle, 2009; Aguilar and Mills, 2008; Peng and Shih, 2006; Akca et al., 2009), and also attracted a lot of attentions in the process of automatic classification of raw LiDAR data (Forlani et al., 2006; Chehata et al., 2008). Besides, before the process of building extraction (3-D reconstruction) and city modeling based on raw LiDAR data, such as building roof reconstruction, building geometry fitting, roof segment identification, grouping of roof planes and generation of roof polygons (Charaniya, 2004; Rottensteiner and Briese, 2003; Verma et al., 2006; Bretar et al., 2009), outliers are as well need to be removed. Therefore, outlier detection becomes an essential preprocessing step for overall LiDAR data filtering and modeling (Amiri and

Sargent, 2007; Sotoodeh, 2006 & 2007; Eisenbeiss, 2009; Chen et al., 2007; Meng et al., 2009; Silván-Cárdenas and Wang, 2006; Wang et al., 2005; Kobler et al., 2007; Arefi et al., 2007; Sithole and Vosselman, 2004; Höhle, 2009).

There are many kinds of outlier detection approaches, generally, based on the classification of Papadimitriou et al., (2003), these approaches can be divided into five major categories, they are: distribution-based, depth-based, clustering-based, distance-based and density-based. Specially, according to the outlier characteristics in LiDAR data sets: "outliers appearing in LiDAR point data in the forms of both single points and also small clusters with elevations, either much higher or lower than the surrounding points", the *frequency distribution of elevation values method* (Meng et al., 2009; Silván-Cárdenas and Wang, 2006; Wang et al., 2005) which belongs to the distribution-based approach, the *mathematical morphology method* (Chen et al., 2007; Kobler et al., 2007), and the *density-based method* (Sotoodeh, 2006 & 2007) have widely proposed by researchers.

During the above three kinds of methods, the nature of both the *frequency distribution method* and the *mathematical morphology-based method* is to compare elevations locally or globally, by defining a threshold which is also the cut-off elevation range value, points beyond the threshold value are regarded as outliers. However, it is difficult to predefine the threshold value, and if the outliers appear in the form of clusters, such methods may suffer from great difficulties in determining a proper cut-off value. In the density-based method, outliers can be graphically viewed as objects or small groups of objects located in low density zones, contrasting with the denser intra-cluster structure (Almeida et al., 2007), to find outliers that are to identify low connectivity zones. Local

outlier factor (LOF) is a popular density-based method, it is the average of the ratios of the density of an example point and the density of its nearest neighbors. It depends on the local density of point's neighborhood, and it also indicates the spatial neighborhood relationship between point and its neighbors. By applying the LOF to LiDAR data, it is powerful to identify isolated points, however, since it lack of elevation information, and the global density of airborne LiDAR points include outlier points is not very, it could potentially misclassify normal objects as outliers and still cannot cope with outlier in the form of clusters which is a major drawback of this method.

1.2.2 Problems in data filtering

Most existing LiDAR data filtering methods are based on the assumption that there will be an abrupt change in height between an off-terrain (object) point and the neighboring terrain (ground) points (Shao and Chen, 2008), under which, with their performance characteristics, the processes of filtering can be divided into three groups: slope-based, morphological, and surface-based methods. Since many of these filtering algorithms are localized, object complexity such as very large and very small objects (see **Figure1.4**), complex shape or configuration (see **Figure1.5**), disconnected terrains (see **Figure1.6**), and steep slopes (see **Figure1.7**), may lead the filtering process failure (Sithole and Vosselman, 2003). Besides, it is also difficult to distinguish attached objects such as building on slopes, bridges and ramps. Moreover, the choice of appropriate data filtering techniques for different particular applications is also being investigated (Crosilla et al., 2004; Zheng et al., 2007).

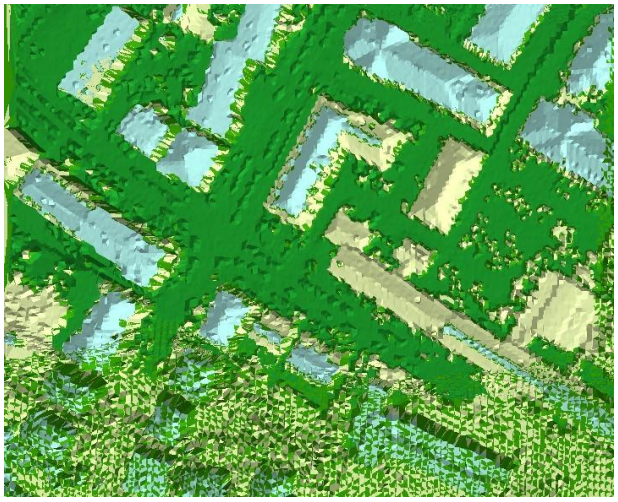


Figure 1.4 Very large and very small objects

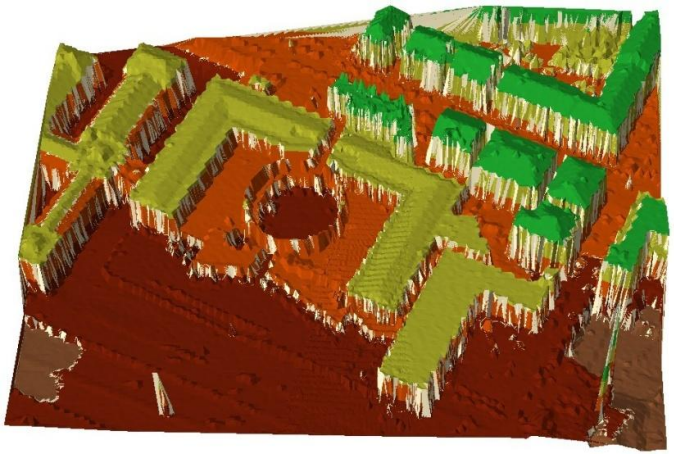


Figure 1.5 Complex shapes and configurations

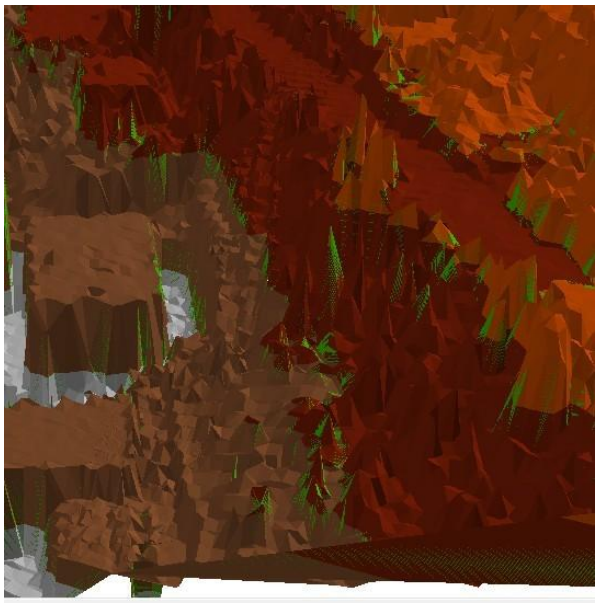


Figure 1.6 Disconnected terrains

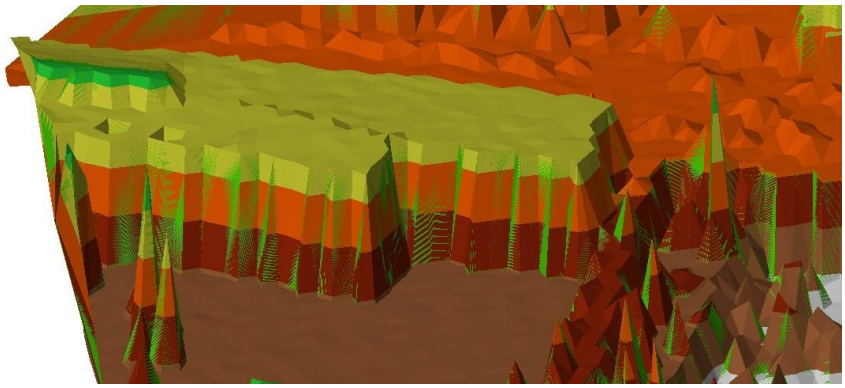


Figure 1.7 Steep slopes

Most of the methods discussed above are mainly based on the analysis of geometrical information of LiDAR points, while, intensity data is seldom used. The major reason for lacking using the intensity data is that it has speckle noise which usually leads a lower accuracy filtering (or classification) result by using such data compared to use geometrical information. The speckle noise usually comes from various sources such as

the atmospheric, the reflectance of the object and the relative position between the aircraft and the object, which becomes a major limitation of using the intensity data (Song et al., 2002; Yan and Shaker, 2010). Although suffering from the disturbance of noise, judging from supporting augments that: the intensity data has not only less influence on shadowing effect and relief displacement which are two of the major issues faced in high resolution optical remote sensing image but also high separability of surface reflectance in the spectrum range of the near infrared and short-wavelength near infrared spectrum (1064 nm or 1550 nm) under where commercial airborne LiDAR is operated, it still has the potential of using the intensity data to filter LiDAR data (Yan and Shaker, 2010). Song et al. (2002) explored the possibility of land-cover classification using LiDAR intensity data, by converting LiDAR point data to a grid and examining some resampling and filtering methods for noise removal, they assessed the separability of intensity data on four specified classes; they are asphalt road, grass, house roofs, and trees. Beasy et al. (2005) examined the application of LiDAR intensity data as well as height metrics to classify the nearshore materials at an ocean beach. Both supervised and unsupervised classifications are performed to validate high separability among the specified three features: bedrock, cobble and sand. Similarly, to classify coastal estuaries and beach habitat, intensity and elevation texture data were utilized by Goodale et al. (2007). Also, both supervised and unsupervised classifications are conducted, results show that it is effective to use these data under the consideration of tidal and seasonal factors. Yoon et al. (2008) examined the LiDAR intensity characteristics with a compressively analysis from both the radiometric perspective and geometric perspective. The results indicate that it does not have high separability

amongst land cover classes. However, just from the radiometric perspective, LiDAR intensity of vegetation is not higher than other targets with a large standard deviation which is regarded as noise due to the small footprint size of LiDAR. Similar findings appear in Bao et al.'s (2008) study, they proposed a so called "skewness change" algorithm to separate ground points and vegetation points in a forested area and finally find that in a small local area, the intensity of the vegetation is much less than that of the ground. The reason is that the area of each point target is much smaller than the footprint area. Wang and Glenn (2009) examined that it is effective to generate DTMs by integrating LiDAR intensity and elevation data in a forested area in similar terrain of relatively simple land-cover classes. Antonarakis et al. (2008) proposed a supervised object orientated approach by using the elevation and intensity data to classify forest and ground types, similar to Antonarakis' solution, the object orientated approach was also proposed by Brennan and Webster (2006) and Chen et al. (2009). And their results show that their methods achieved a high level accuracy.

Most of the above mentioned methods are mainly used in forestry areas. Since in the forestry areas, features are relatively few (probably most features are vegetation and bare earth), and vegetated structures are relatively simpler and much more likeness comparing with urban areas which has various features and more complex structures, it then has great potential to use intensity data to separate different features in such area. While, in urban areas, the potentialities of intensity are less obvious without data fusion such as remote sensing images, which attract few researchers to investigate on this study.

1.2.3 Summary

As analyzed in Section1.2.1, many researchers have developed various methods to remove outliers, and these methods can be summarized into two major categories: (1) Analysis of the elevation deviation; (2) Analysis of the spatial neighborhood relationship. Outlier detection schemes in both the two categories could only identify individual outliers, while potentially misclassify normal objects as outliers by analyzing single attribute: elevation or spatial neighborhood relationship (such as "LOF"). In this study, the spatial neighborhood relationship is considered as an attribute of every LiDAR point. Judging from the characteristics of outliers in LiDAR data: "Both individual points and cluster points with elevations much higher or lower than the surrounding points", it is easy to detect that there are two significant information of outliers: (1) the first one is the form issue: "individual or cluster"; (2) the second one is the height issue: "elevations much higher or lower". It is found that most of the existing approaches only focus on single information of the two. Therefore, to accurately detect both forms of outliers in LiDAR data, an outlier detection method by using both the two significant information (attributes) mentioned above will be introduced in this study.

While, for data filtering, as discussed in Section1.2.2, most of the existing algorithms only focus on the study of geometric information, the potential of using the intensity data (radiometric information) to filter LiDAR data in urban area is seldom analyzed. Since the geometric information and the radiometric information are simultaneously generated on the same platform, both the two information describe the same features geometrically, although it has challenges to calibrate the raw intensity data which always has speckle noise, the comprehensive utilization of both the height and intensity data simultaneously provided by LiDAR may be advantageous over using either data

individually (Wang and Glenn, 2009). Similar suggestions can be found in Clément Mallet (2009) and Vosserman's (2010) works, they also pointed out that intensity even more radiometric information (refers to the full-wave form data) could be utilized as additional information to improve the filter performances. Judging from the situations mentioned above, to investigate the potential of using both the two data for data filtering in urban areas, in this study, a filtering scheme by using both the geometric information and radiometric information (intensity data) to separate terrain points and off-terrain points is proposed.

1.3 **Research objectives**

This study aims to develop a multiple attributes model both for outlier detection and data filtering in raw airborne LiDAR data, which extends traditional data processing methods from single attribute to multiple attributes, from one dimension to multiple dimensions.

Specifically, to achieve this aim, the following objectives were set:

1. To propose the MCD-based multiple attributes model both for outlier detection and data filtering.
2. To develop an outlier detection method in LiDAR data by using the MCD-based model, and the method can automatically remove both single and cluster outliers.
3. To develop a data filtering method in LiDAR data by using the MCD-based model.
4. To conduct case studies and test the performances of the proposed outlier detection and data filtering methods.

1.4 **An outlook of our approach**

Since this study aims to develop a multiple attributes data processing model, the minimum covariance determinant estimator (MCD) which is a super robust statistic estimator of location and scatter (Rousseeuw, 1984&1985) will be applied into LiDAR data. For outlier detection, we first define the connectivity-based outlier factor (COF), which indicates the spatial neighborhood relationship and is also a density-based outlier factor as an attribute of LiDAR points; then the COF attribute and the height attribute are extracted from LiDAR points to organize a 2-D space, in the formed 2-D space, the MCD-based model will be conducted to identify outliers. For data filtering, height attribute and intensity attribute are firstly extracted from LiDAR points to organize a 2-D space, also in the formed 2-D space, the MCD-based model will be conducted to separate terrain points and off-terrain points. Experimental case studies will be conducted on the ISPRS test data to demonstrate the effectiveness of the proposed methods.

1.5 **Thesis organization**

Based on the aims and objectives of this study and the adopted methods, the issues stated in this thesis are presented in eight chapters listed as follows:

Chapter 1 introduces the background of the study, research problems, the main aim and objectives, an outlook of our adopted approach and outline of the thesis.

Chapter 2 provides a comprehensive literature review in related works on both the outlier detection and data filtering in LiDAR data.

Chapter 3 introduces the study area and the data used in this study.

Chapter 4 presents details of the proposed MCD-based multiple attributes model.

Chapter 5 presents the process of applying the MCD-based multiple attributes model in a 2-D space formed by COF and height attributes to achieve outlier detection purpose.

Chapter 6 presents the process of applying the MCD-based multiple attributes model in a 2-D space formed by height and intensity attributes to do data filtering.

Chapter 7 illustrates the experimental results and discussions both for the outlier detection and data filtering issues.

Chapter 8 presents the main conclusions and future works of the study.

CHAPTER 2 Literature Review

As mentioned in Section1.3, this study aims to develop a multiple attributes model both for outlier detection and data filtering in raw airborne LiDAR data, which could extend traditional data processing methods from single attribute to multiple attributes, from one dimension to multiple dimensions. Meanwhile, four major objectives (see Section1.3) are also listed. To establish the theoretical background of this study, a great body of literature review has been undertaken. Since this study focus on approaches for outlier detection and LiDAR data filtering, all the related works on these areas will be reviewed and analyzed in this section.

2.1 A review and analysis of outlier detection in LiDAR data

As an important branch in the area of data mining, outlier detection with the major task is to discover the exceptional data in certain datasets has been conducted by many studies for large dataset such as LiDAR point clouds data. Hawkins (1980) defined outlier as: “An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism”. To find out the mechanisms, researchers have proposed various approaches for outlier detection. Specially, in LiDAR point clouds data, outlier detection is regarded an essential data preprocessing step of overall terrain modeling and has also attracted much more attentions in recently studies. Generally, Papadimitriou et al. (2003) classified these outlier detection approaches into five major categories, they are: distribution-based, depth-based, clustering-based, distance-based and density-based.

2.1.1 Distribution-based approach

Distribution-based approaches are commonly based on certain statistical distribution models such as the normal distribution model which are used to fit the dataset. The models are conducted into the given data, and then apply a statistical test, mostly in form of the statistical discordancy test to determine if an object belongs to such model or not. Due to such probability distribution, observations which deviate from the given distribution are treated as outliers (Yamanishi et al., 2000). For example, Yamanishi et al. (2000) used a Gaussian mixture model to present the normal behaviors, and each object is given a score on the basis of changes in the model. High scores indicate higher possibility of being an outlier. Normally, by adding some supervised knowledge to this approach to get general patterns for outliers. With these certain models, it becomes very clear to indicate how strong an outlier is and identify a certain percentage of the data. Since in many real applications, prior knowledge of the distribution of the dataset is unknown, distribution fitting is essentially needed to check which model fits enough (if any). After that, discordancy tests are commonly used to determine exceptional data. While, both the distribution fitting and the discordancy tests always perform expensive and time confusing (Hawkins, 1980; Rousseeuw and Leroy, 1987; Barnett and Lewis, 1994; Vanicek and Krakiwsky, 1982).

2.1.2 Depth-based approach

Depth-based approaches are commonly based on the computational geometry, in which, data will be organized into k-dimensional convex hulls, and then compute different layers of the k-d convex hulls (Preparata and Shamos, 1985; Ruts and Rousseeuw, 1996; Johnson et al., 1998). Depth represents how central of a point respect to a data set by

accounting the quantitative measurement. Based on the dense area of the data, an ellipsoid is generated, and outliers are outside such ellipsoid. Its basic processing steps can be summarized as follows: 1) all data are organized into a k-d data space; 2) each data represents as a point in the space with a given depth respect to the location of other points; 3) the smaller depth an object is, more 'exposed' it appears and are more likely to be an outlier as shown in **Figure 2.1**.

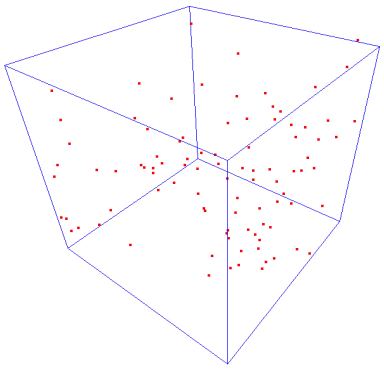


Figure 2.1 Depth-based outlier detection approach (Johnson et al., 1998)

There are many definitions of the depth, here we just take a popular one---the "Tukey Depth" which is proposed by Friedman and Tukey (1974) for example. Its definition is the minimum number of objects to be removed to expose the object. Assume there is a plane passing through the object, it may generate two datasets: one above and one below the plane (see **Figure 2.2**).

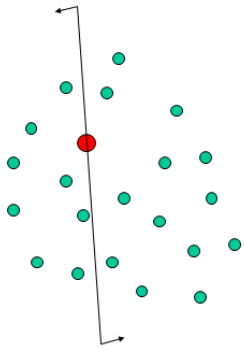


Figure 2.2 The definition of Tukey Depth (Friedman and Tukey, 1974)

Rotate line through the object in all possible angles and keep counting the number of objects of both datasets. The minimum numbers is the Division Number, and identify the objects with a small Tukey Depth as outliers (Friedman and Tukey, 1974) (see **Figure 2.3**).

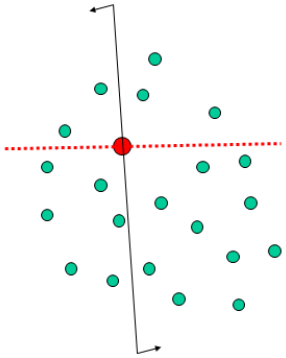


Figure 2.3 Minimum number of data points on any side of a line through point (Friedman and Tukey, 1974)

Although this approach can avoid distribution fitting, and in theory, it could conceptually work for multidimensional data objects to be processed (Mansur et al., 1999). However, in practice, if there have efficient algorithms for k = 2 or 3 (k for dimensions), such approaches become inefficient for large datasets for k > 3, and it is

well known that these algorithms suffer from the dimensionality limitation and are not sensitive for a large k. Both the distribution-based and depth-based approaches are based on statistics, and frequently appear in early work of outlier detection (Barnett, 1994; Ramaswamy et al., 2000).

2.1.3 Distance-based approach

Without distribution fitting, distance-based approaches are also widely used. This method was originally proposed by Knorr et al. (2000). Normally, in this method, outliers are detected as the following description: Given a distance measure on a feature space, suppose there is a point q in a data set is an outlier with respect to the parameters M and d, if there are less than M points within the distance d from q, where the values of M and d are predefined by the user. This outlier definition is based on a single but global criterion determined by the parameters. It also could be defined like this: a distance based outlier in a dataset D is a data object with a given percentage of the objects in D having a distance of more than dmin away from it. For example, if DB (p, D), it means object O is a distance-based outlier if at least p percent of the other objects are of a distance ≥ D from O (Knorr et al., 1998, 2000 & 2001; Ramaswamy et al., 2000).

A very popular algorithm based on this approach has been proposed by Knorr and Tucakov (2000), which named as the Nested-Loop (NL) algorithm. In NL, each data point in the data set is compared to every other point in the data set to determine its M nearest neighbors. Given the neighbors for each data point in the dataset, simply select the top n candidates according to the outlier definition. However, NL has quadratic complexity as we must make all pairwise distance computations between the data points. Although distance is an effective non-parametric approach to detecting outliers, the

drawback is the amount of computation time required. To improve this method, the use of spatial indexing structures such as R-trees and X-trees to find the nearest neighbors of each candidate point was suggested by Knorr et al. (2000). While, this suggestion may work well for low dimensional data sets but lead to poor performance with the dimensionality increases. Belal et al. (2007) proposed an algorithm to speed up NL. Test results were performed on different well-known data sets. The results show that their algorithm gave a reasonable amount of CPU time saving. However, the major problem of this approach is that it is difficult to determine the values of M and d, and these approaches are not sensitive to process cluster datasets with different densities (Chen et al., 2003).

2.1.4 Clustering-based approach

Clustering works as a popular technique to gather resembling data points or objects in groups or clusters, furthermore, it also performs an important tool for analysis of outlier detection (Jain and Dubes, 1988). There are certain popular clustering-based approaches such as CLARANS (Raymond et al., 1994), DBSCAN (Ester et al., 1996), BIRCH (Zhang et al., 1994) and CURE (Guha et al., 1998). The basic idea of these approaches can be summarized as the following three steps: (1) group data into clusters of different density; (2) points in small cluster are selected as candidate outliers; (3) then the distances between candidate points and non-candidate clusters are worked out (see **Figure 2.4**). If candidate points have large distances from all other non-candidate points, they are outliers.

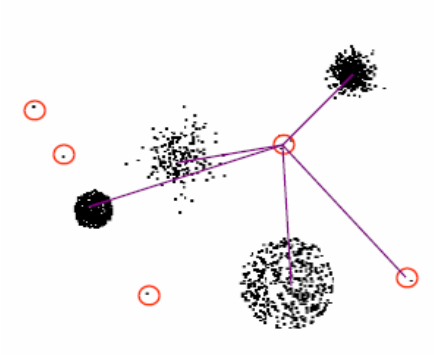


Figure 2.4 Clustering-based approach for outlier detection (Jain and Dubes, 1988)

Such clustering algorithms for outlier detection are recognized as by-products (Jain et al., 1999). In a clustering algorithm, outliers appear as objects not located in clusters of a dataset. Here, we just argue one of the most popular algorithms which named as "Partitioning Around Medoids (PAM)" algorithm (Kaufman and Rousseeuw, 1990). PAM strives to determine k separations for n objects. Instead of using the cluster mean, such algorithm uses the most centrally located object in a cluster (called medoid) which makes it more robust than the k-means algorithm when measuring outliers. By avoiding the influences of outliers and extreme values, the medoids produced by PAM are powerful representations of the cluster centers than the means (Laan et al., 2003; Kaufman and Rousseeuw, 1990; Dudoit and Fridlyand, 2002). Furthermore, PAM is a data-order independent algorithm (Hodge and Austin, 2004), and it also shows that the medoids offer better class partition than the k-means clustering algorithms (Lane and Bradley, 1999). Recently, Belal and Zoubi (2009) has improved the PAM algorithm. In his approach, a set of clusters and medoids (cluster centers) are firstly produced by the PAM algorithm; then they determine small clusters as outlier clusters. While, to define small clusters, they followed the solution of Loureiro et al. (2004) which are defined as

"a cluster with fewer points than half the average number of points in the k clusters". If there are any outliers in the remaining clusters, they can be removed by "calculating the absolute distances between the medoid of the current cluster and each of the points in the same cluster" (Belal and Zoubi, 2009). The test results show that their approach offers effective results when applied to different data sets. However, as addressed by Jain et al. (1999), since the major objective of a clustering algorithm is to find clusters, developments are mainly for optimizing clustering but outlier detection. General speaking, these algorithms treat outliers from a more global perspective (Breunig et al., 2000), and their efficiency and effectiveness are lack of optimization (Ng and Han, 1994).

2.1.5 Density-based approach

During recent years, many researchers have focused on the density-based approach (Breunig et al., 2000; Jin et al., 2001; Agyemang and Ezeife, 2004). This method was firstly proposed by Breunig et al. (2000) for Knowledge Discovery in Database (KDD) applications. Further, by calculating the Local Outlier Factor (LOF), which is "the average of the ratios of the density of example p and the density of its nearest neighbors", this method judges a point whether to be an outlier. The ratio indicates "how isolated the point is with respect to its nearest neighbors", and points with larger LOF value, it may have higher potential of being outliers. In Breunig et al.'s (2000) study, each object is assigned as a degree of being an outlier. Such degree is the so called "LOF" of an object which depends on "how isolated the object is with respect to the surrounding neighborhood". It is easy to see the LOF depends on the local density of its neighborhood. To define the neighborhood, "MinPts" is introduced and predefined

which refers to the minimum number of points of the nearest neighbors when calculating the density, while, the distance to the MinPts-th nearest neighbors is regarded as the neighborhood. **Figure 2.5** shows the concepts of density-based outliers.

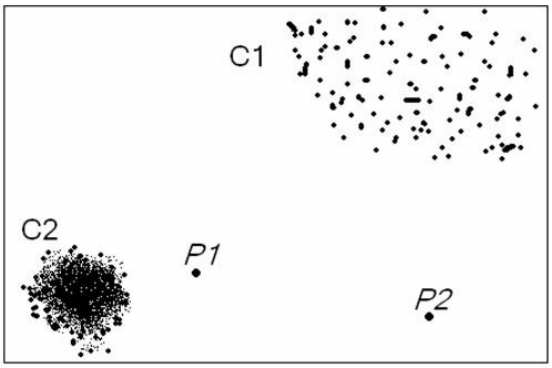


Figure 2.5 Concept of Density-based outliers (Breunig et al. 2000)

To some extent, this method can solve problems of cluster datasets with different densities (Breunig et al., 2000), however, the computation of the LOF value of each object in a dataset needs neighborhood queries which makes calculated amount very huge (Agyemang and Ezeife, 2004). Since outliers appear only small fraction of the entire dataset, Jin et al. (2001) improved the algorithm by assuming that the strongest n local outliers in a large dataset of size N. While, challenges still exist especially for large, arbitrary datasets.

2.1.6 Outlier detection in LiDAR point clouds data

The previous sections illustrate five major groups of outlier detection approaches in general, specifically, outliers appearing in LiDAR point clouds can be both single points and also small clusters with elevations, either much higher or lower than the surrounding points. High outliers are points that normally do not belong to the landscape, in that they

originate from hit off objects such as birds, low flying aircraft (Sithole and Vosselman, 2004). Random errors caused by sensor noises are also appearing as high outliers (Meng et al., 2009; Wang et al., 2005). And many researchers treated the high outliers as positive outliers (Kobler et al., 2007; Höhle, 2009; Forlani et al., 2006). Like high outliers, low outliers, in a different way, are not normal parts of the landscape. However, low outlier points have a different origin. They originate from multi-path errors and errors in the laser range finder (Sithole and Vosselman, 2004). And many researchers treated the high outliers as negative outliers (Kobler et al., 2007; Höhle, 2009; Forlani et al., 2006). Since many of the filtering algorithms work on the assumption that a lowest point must belong to the terrain points, however, in cases where the lowest point is an outlier, the assumption is totally wrong, same cases may happen where the highest point is an outlier, therefore, outlier detection becomes an essential preprocessing step for overall LiDAR data filtering and modeling (Amiri and Sargent, 2007; Sotoodeh, 2006 & 2007; Eisenbeiss, 2009; Chen et al., 2007; Meng et al., 2009; Silván-Cárdenas and Wang, 2006; Wang et al., 2005; Kobler et al., 2007; Arefi et al., 2007; Sithole and Vosselman, 2004; Höhle, 2009).

Frequency distribution of elevation values is commonly used as a very direct and popular way to identify these outliers (Meng et al., 2009; Silván-Cárdenas and Wang, 2006; Wang et al., 2005). In their works, an elevation histogram distribution was conducted to show the elevation range of ground and above-ground features, points beyond the range are regarded as outliers. Delaunay triangulation (Meng et al., 2009; Silván-Cárdenas and Wang, 2006) is then used to treat the remaining outliers by comparing the elevation range (or height difference) with respect to all their neighbors.

Furthermore, Höhle (2009) provided certain robust statistical methods such as median, normalized media absolute deviation as accuracy assessments of DEMs to analysis the influences of outliers. Quality control was then achieved by eliminating these outliers.

Mathematical morphology is widely used to filter LiDAR data. As a preprocessing step, Chen et al. (2007) also conducted the morphological operations to detect outliers. The higher outliers are removed in the morphological opened surface, while the lower outliers are removed by using the morphological erosion operator under the assumption that lower outliers are scattered. Kobler et al. (2007) followed the morphological operator to remove high outliers (positive outliers) and proposed a local elevation comparison algorithm to detect low outliers (negative outliers).

Sotoodeh (2006) followed the density-based theory, and conducted the LOF algorithm into laser scanner point cloud. Since neither it is constrained by the preliminary knowledge of the object nor suffers from the varying density of the points, this method is proposed in many cases such as (Amiri and Sargent, 2007; Eisenbeiss, 2009). However, cluster outlier is still a challenge. Sotoodeh (2007) then proposed a so called hierarchical outlier detection algorithm on the basis of the minimum spanning tree to solve the cluster outlier problems, while, it seems the airborne case is still not get anywhere.

2.2 A review and analysis of LiDAR data filtering

Most LiDAR data filtering methods are based on the assumption that there will be an abrupt change in height between an object point and the neighboring ground point (Shao and Chen, 2008), under which, with their performance characteristics, the processes of

filtering can be divided into three groups, they are slope-based, morphological, and surface-based methods (Sithole and Vosselman, 2004).

2.2.1 The slope-based method

The slope-based filter was first proposed by Vosselman (2000), they assume that terrain slopes rise under a certain threshold, if the features in the data that have slopes above this threshold, then they are treated as objects which belong to the off-terrain points, otherwise, they are regarded as the nature terrain surface which belong to the terrain points (Zhang et al., 2003). Obviously, the higher the predefined threshold, the less objects (off-terrain points) will be removed, thus, gentle slopes will be easily ignored by setting a certain higher value of the threshold. To solve this problem, Sithole (2001) developed such filter, the improvements makes the threshold varies with respect to the slope of the nature terrain surface. Besides, to determine an optimum threshold, it is always considerable to use prior knowledge about terrain surface in a study area. However, it is difficult to achieve good results unless the training datasets cover all types of ground features in a study area, which is not always practical (Zhang et al., 2003). In parallel, to improve the slope-based filter, Roggero (2001) applied a local operator to all the elements of the gridded network of raw LiDAR data to determine the local slope. Considering the slope-based filter characteristics, it is obvious that "good results will be obtained by applying the filter to areas which there are distinct differences between the slope of terrain and that of non-ground objects such as trees and buildings" (Zhang et al., 2003). By applying the filter to flat urban areas, such filter has achieved satisfactory results. While, in vegetated mountain areas, since they have a large slope variation, such filter performs poor.

2.2.2 Morphological method

The basic idea of this method is using certain morphological operations such as opening and closing to filter LiDAR data (Kilian et al., 1996; Lohmann et al., 2000). Zhang et al. (2003) converted the LiDAR points into a grayscale image in terms of elevation. Since elevation difference is a significant feature among terrain points and off-terrain points, therefore, by observing the obtained grey tone, LiDAR points could be identified.

In a given size window, point with lowest elevation is identified by conducting an opening operation, then for other points which stand above the identified point, if their elevations are lower than a predefined value, they are classified as terrain points. Iteratively, all LiDAR points are filtered by moving the window. Obviously, the selection of an optimal window size becomes a key issue. To find an optimal window size, Kilian et al. (1996) conducted the morphological operations iteratively from the smallest size to larger sizes. According the window size, point which has been identified as terrain point is given a weight and the weight is in direct proportion to the window size. Terrain points may have high weights, while off-terrain may have low weights. By comparing obtained weights, terrain points are picked out finally. A progressive morphological filter is proposed by Zhang et al. (2003) to discard off-terrain points. By increasing the window size and using elevation difference thresholds, preserving terrain points can be also gradually identified. Zhang and Whitman (2005) determined the threshold by the elevation difference and terrain slope. Zakšek and Pfeifer (2006) improved the classical morphological filter by incorporating trend surfaces extracted from raw LiDAR data, and makes it universal and attainable to use the filter effectively even in steep areas covered with vegetation. Chen (2007) also did some improvement in

the classical morphological filters. He presented a method to maintain the terrain features constant while using larger window sizes in process of filtering. Works like filling missing data and removing outliers are proposed in his method as well. However, this method demands a large number of parameters which makes it complex for average users to understand the algorithm. Therefore, Chen (2009) introduced an improved work based on his previous algorithm. The total number of parameters is reduced from 7 to 2, while, the average filtering error decreased slightly.

2.2.3 The surface-based method

The surface-based method is based on the point to surface concept, which is the classical parameterized surface fitting method. In this method, all points are assumed belong to the ground surface initially, after that, based on certain rules, points which do not fit the initial surface are removed. Kraus and Pfeifer (1998) and Pfeifer et al. (2001) presented an iterative linear least square interpolation which has been embedded in a hierarchical approach. The filter result was improved, and the computation was speeded up. An optimal ground surface was generated by reducing the weights of above-ground points and outliers as well. Krzystek (2003) proposed a surface fitting method which combines a pre-filter based on a convex hull and a subsequent finite element adjustment. This approach is based on a triangulated irregular network (TIN) model and has a good filtering result in forests areas with different forest structures. In addition, Elmqvist (2001) developed an active shape model method to evaluate the terrain surface from LiDAR data. By using this model points which are close enough to the surface are labeled as ground points, while, in some sparse data sets, problems may occur. Obviously, the initial surface in most approaches is assumed as an ideal continuous

surface, however, in practice, points may locate on break lines and discontinuous surface is normal, thus these assumptions may bring rounding errors into linear ground features. To reduce these errors, the TIN model is usually used to select ground points iteratively. Based on the TIN model, a progressive method was developed by Axelsson (2000) to densify the ground points. Haugerud and Harding (2001) developed a despike algorithm, by eliminating the spiking points from the TIN model, break lines can be recovered. An upward and downward densification method proposed by Sohn and Dowman (2002) is used to recursively segment the entire LiDAR data area into a set of piecewise planar surface models. By doing that, "underlying terrain slope variations will be regularized into homogeneous plane terrain". The operation is similar to the Axelsson's, but may suffer from the computational speed.

2.2.4 Additional information used for improvements of data filtering

Data filtering is the primary and essential step in the overall LiDAR data processing steps especially required for DEM generation. Various algorithms to separate terrain points and off-terrain points from airborne laser scanning data have been developed by many researchers. However, the problems addressed in the previous section (see Section1.2) are not completely worked out. Therefore, it needs further improvements to advance the current filtering results. As discussed before, almost all the existing filtering methods are mainly based on the analysis of geometrical information of LiDAR points, while, radiometric information such as intensity data is seldom used. Since the geometrical information and the radiometric information are simultaneously generated on the same platform, both the two data describe the same features geometrically, although it has challenges to calibrate the raw intensity data which always has speckle

noise, the comprehensive utilization of both the height and intensity data simultaneously provided by LiDAR may be advantageous over using either data individually (Wang and Glenn, 2009). Similar suggestions can be found in Mallet and Bretar, (2009) and Vosserman's (2010) works. Liu (2008) also pointed out that "using additional information such as intensity data has a potential for increasing the accuracy and reliability in the filtering process". Thus, the usage of additional information provides a direction for improving the filtering by including additional information and merging it into the current known filter algorithms.

CHAPTER 3 Study Area and Data Used

3.1 Overview of the study area and experimental data

The experimental data set used in this study both for verifying the proposed outlier detection and data filtering schemes are acquired from International Society for Photogrammetry and Remote Sensing (ISPRS) Commission III, WG III, which are available in the ISPRS web site. There are total eight test sites including nineteen sub-sites over four urban areas as well as four forest areas and the sub-sites' respective reference (ground-truth) data is also provided. The corresponding urban areas of the dataset are located in Vaihingen/Enz test field and Stuttgart city center (Germany). The intensity information as well as the first and last return information is recorded by an Optech ALTM scanner. Since in this study, we are focusing on the urban areas, then the basic information of the dataset in urban area is listed in **Table 3.1**. For a better understanding of the areas covered by the data, we use Google Earth to illustrate the urban sites with corresponding reference sub-sites (see **Figure 3.1**).

Table 3.1 Characteristics of the reference samples in urban area at the ISPRS filter test (Sithole and Vosselman, 2003).

Region	Sites	Ref.data (Sub-site)	Number of points	Terrain Features
Urban	Site1	Samp11	38010	Steep slopes, mixture of vegetation and buildings on hillside, buildings on hillside, data gaps
		Samp12	52119	
	Site2	Samp21	12960	Large buildings, irregularly shaped buildings, road with bridge and small tunnel, data gaps
		Samp22	32706	
		Samp23	25095	
		Samp24	7492	
	Site3	Samp31	28862	Densely packed buildings with vegetation between them, building with eccentric roof, open space with mixture of low and high features, data gaps
	Site4	Samp41	11231	Railway station with trains (low density of terrain points), data gaps
		Samp42	42470	

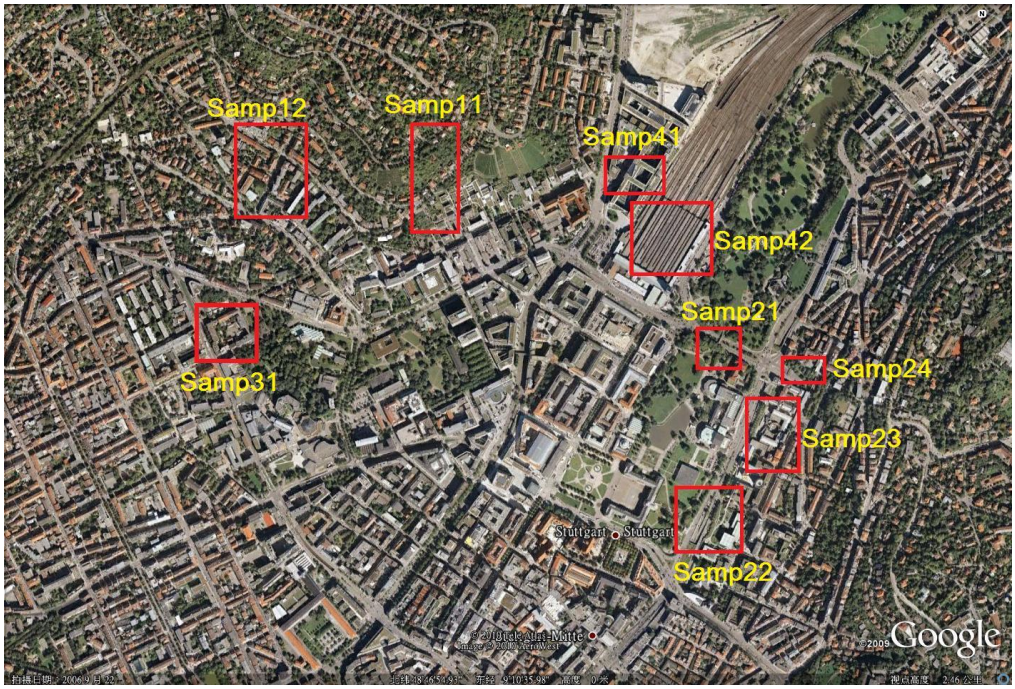
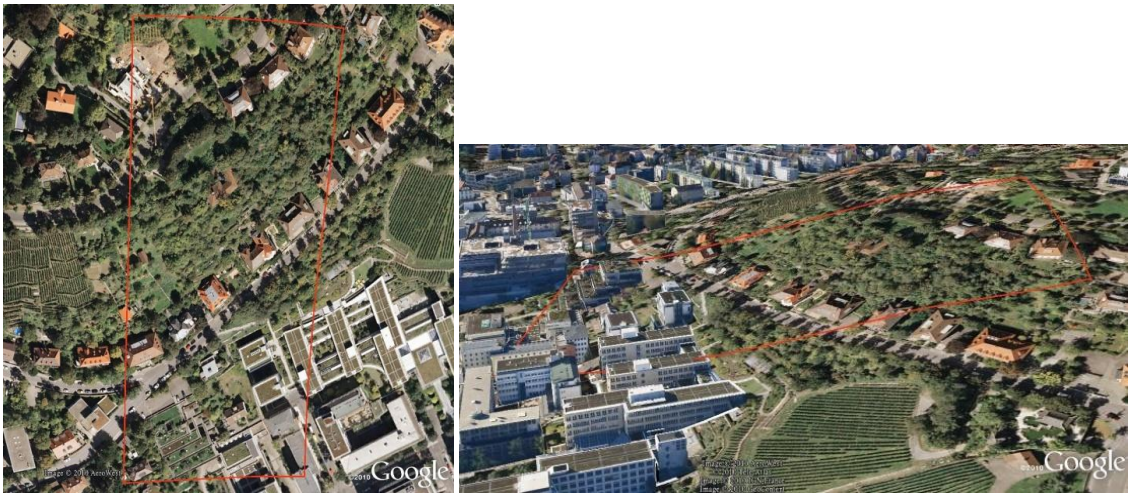


Figure 3.1 Urban sites with corresponding reference sub-sites on the imagery from Google Earth

3.2 Features viewed on the imagery from Google Earth

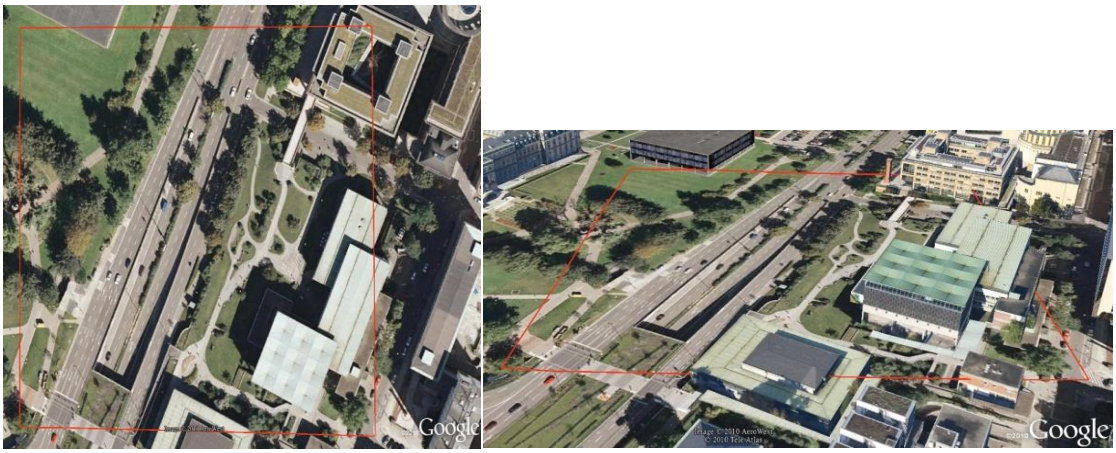
As a very popular virtual global product, Google Earth is used to visually display terrain features in this study, and it offers rough impressions on the target regions. After certain coordinates and formats transformation, the LiDAR points are plotted on the Google Earth in their reference positions. Then, features described in **Table3.1** could be found. Profile views of the target regions as well as the 3-D buildings layer in Google earth are also shown (see **Figure 3.2**).



(a) Features of Samp11 on the imagery from Google Earth

(b) Features of Samp12 on the imagery from Google Earth

(c) Features of Samp21 on the imagery from Google Earth

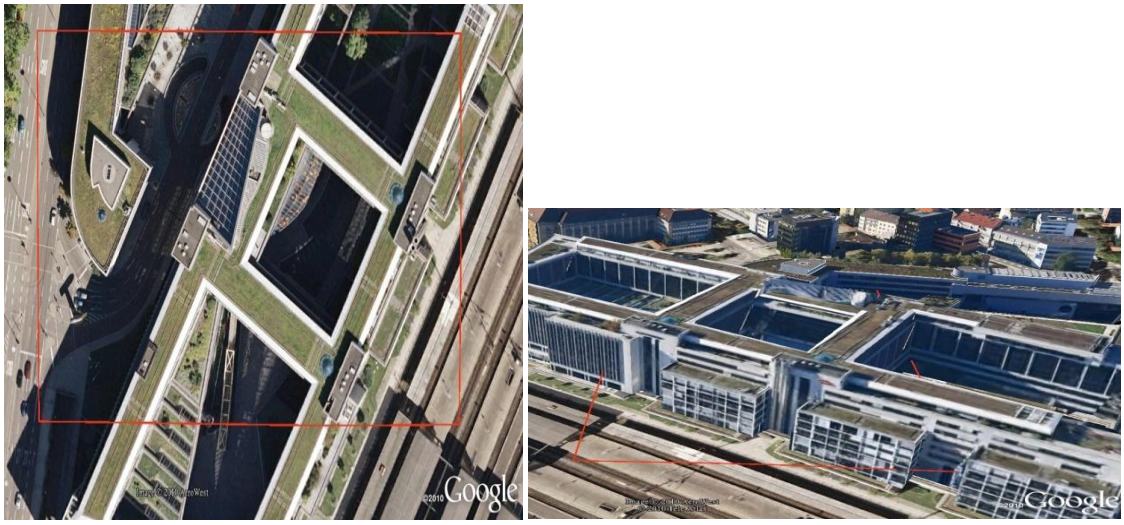


(d) Features of Samp22 on the imagery from Google Earth

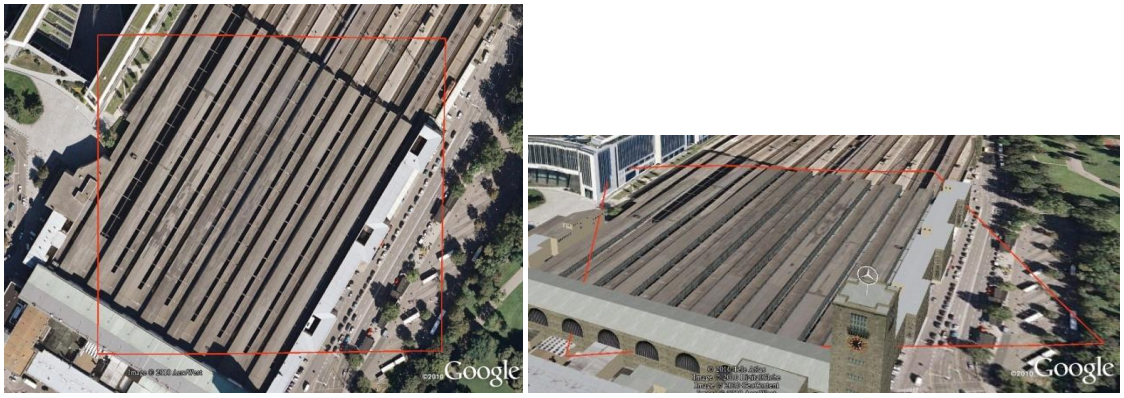
(e) Features of Samp23 on the imagery from Google Earth

(f) Features of Samp24 on the imagery from Google Earth

(g) Features of Samp31 on the imagery from Google Earth



(h) Features of Samp41 on the imagery from Google Earth



(i) Features of Samp42 on the imagery from Google Earth

Figure 3.2 Features viewed on the imagery from Google Earth: (a) to (i) represents Samp11 to Samp41 respectively, the left ones are the target regions in Google Earth; and the right ones are the profile views of the target regions with 3-D buildings

3.3 Data for outlier detection

For outlier detection, as described by Sithole and Vosselman (2003) in the report of "ISPRS test on extracting DEMs from point clouds: A comparison of existing automatic filters", although the number of outliers (both single and cluster) are relatively small,

even for a single outlier, the influence on filtering in its neighborhoods can be considerable, and experiments on certain filters such as (Hubert and Debruyne, 2009; Axelsson, 1999&2000) and Sithole and Vosselman (2004) have proven this issue. In their experiments, it shows that most filters can detect single outlier easily, because they are so far elevated above neighboring points. However, for many low outliers (outliers in the form of both single and cluster), it may cause problems for many filters such as (Brovelli et al., 2002; Hubert and Debruyne, 2009; Axelsson, 1999&2000; Sithole and Vosselman, 2004). To verify the proposed outlier detection scheme in this study, we are looking for experimental data which can contain outliers both in the single form and cluster form, especially for low outliers in a local area. Two pieces of data in Sithole and Vosselman's (2004) experiments， "Samp41" and "Samp31" are appropriate and adopted. "Samp41" which contain low outliers both in the single form and cluster form, is an ideal experimental data set. Since for many current algorithms, they work on the assumption that points neighboring a low point must belong to an object, in cases where the lowest point is an outlier, the assumption may result in erosion of points in the neighborhood of the low outlier, and "Samp31" is such a case although it only has several single outliers.

3.4 Data for data filtering

Reasons for the selection of all these nine sample data (Samp11-Samp42) to do data filtering can be summarized into the following four aspects: (1) Areas covered in all these samples are located in the urban areas which meet the requirements of this study; (2) These samples have diverse feature contents such as open fields, vegetation, outliers,

buildings, roads, railroads, rivers, bridges, power lines, water surfaces, etc. And the representative of different environments provides expected difficulties to check the performance of the proposed methods; (3) All of the selected samples have similar resolutions. Since in the data filtering steps (see Section6.2), we will use Samp31 as a training dataset to estimate the parameters of the proposed filter, and then the filter is tested on the other datasets referred as the test datasets. Such process follows a very important assumption is that the distribution of features is similar between the test dataset and the train dataset, obviously, the resolutions between the test dataset and the train dataset also need to be similar; (4) All points in the sample data sets were labeled Bare Earth or Object manually, and they are very suitable for quantitative analysis (Sithole and Vosselman, 2003).

CHAPTER 4 MCD-Based Multiple Attributes Model

As discussed in Section1.2, for outlier detection, most of the existing outlier detection schemes could only identify single outliers, and potentially misclassify normal objects as outliers by using single attribute: elevation or spatial neighborhood relationship (such as "LOF"); for data filtering, it suggested that the comprehensive utilization of both height and intensity data simultaneously provided by LiDAR may be advantageous over using either data individually and the radiometric information could be utilized as an additional information to improve the filter or classifier performances .Therefore, to fit the requirements of multiples attributes data processing both for outlier detection and data filtering, in this section, we will introduce the MCD-based multiple attributes model in detail.

4.1 MCD-based multiple attributes model

With the ever-increasing volume of spatial data, there may be always multiple attributes associated with each spatial location, and such attributes represent the data in different views. Here, we take a spatial data set with multiple attributes for instance: In **Figure 4.1-(1)**, objects are located in the X-Y plane with their attribute values, while in **Figure 4.1-(2)**, same objects are located in the X-Y plane with their another attribute values. Two types of attributes may represent the same object in different views by accounting their attribute values. Therefore, theoretically, when we use these data to meet applications, all of the attributes need to be taken into account. However, in practice, the attributes which play as key roles for the application are always extracted out, and then a

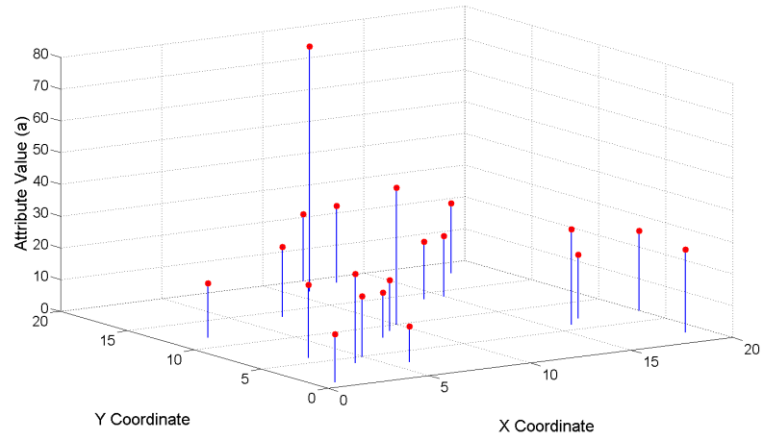
multiple dimensional space is generated by using the extracted attributes (see **Figure 4.2**). In the space, certain operations are conducted to do corresponding data process to meet the application.

Specially, in LiDAR data, each point is located in the "X-Y" plane with its different attributes such as geometric information (height) and radiometric information (intensity). When we use the data to meet certain application, we need to exact the attributes which act as key roles in this application. In this study, we are focusing on the two applications: outlier detection, and data filtering. For outlier detection, as discussed in the previous sections, the major problem exists in current methods is that they only consider only one attributes: height or spatial neighborhood relationship. Results show that they could only identify single outliers, and potentially misclassify normal objects as outliers. Therefore, the two key attributes: "height" which refers to the outlier charactertics of "with elevations much higher or lower than the surrounding points" and "spatial neighborhood relationship" which refers to the outlier charactertics of "in the form of single points or clusters" are need to be extracted out. Then a 2-D space is generated by using the extracted two attributes. In such space, certain operations need to be done to detect outliers. Normally, a standard solution to investigate whether a multivariate data set contains outliers is to calculate the Mahalanobis distance of the observations (objects). The process of the calculation of the M-distance (Mahalanobis distance) can be summarized as the following two steps: (1) the sample mean and sample covariance which used to represent the "center" of the sample are calculated first; (2) then we calculate the M-distance of every sample (observation) to the "center "of the data set. Obviously, a higher M-distance indicates a higher possibility of an outlier. Since the

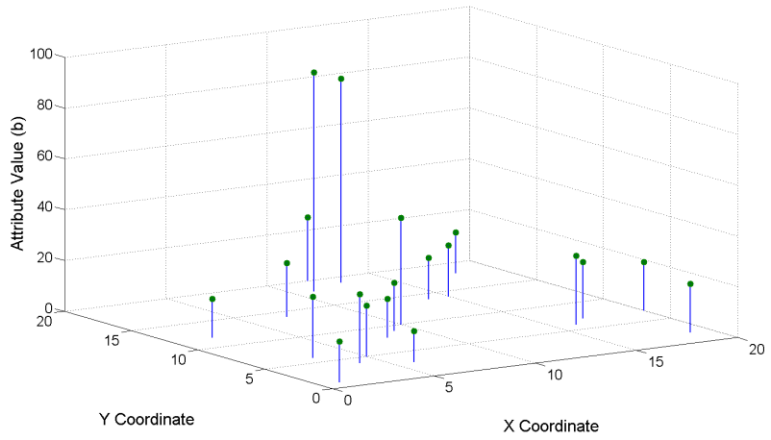
data set contains outliers, thus, the sample mean and sample covariance calculated before is not precise. Then the Minimum Covariance Determinant (MCD) estimator is proposed to estimate a precise sample mean and sample covariance, that is a precise sample "center", and the robust M-distance of every sample can be obtained. However, how to make a cut-off value for how higher of the M-distance of an object be an outlier? It is well known that if the data follows a multivariate normal distribution, then the squared M-distance approximately follow a chi-square distribution, that is $\chi_q^2(\alpha)$, where q is the degrees of freedom, it equals to the dimensionality of the data set, and (100α) is the percentile of the distribution. Then we determine a "α", samples with the squared M-distance fall into the black tail of the chi-square distribution (see **Figure 4.3,** take q=2 for example) are flagged as outliers. The overall process mentioned above is the principle of the proposed MCD-based multiple attributes model, and such model extends traditional data processing methods in LiDAR data from single attribute to multiple attributes, from one dimension to multiple dimensions.

For data filtering, the nature of this application is to separate terrain points and off-terrain points from LiDAR point clouds. Under certain conditions, which will be further detailedly discussed in Chapter 6 (the description of the conditions can be summarized as "for a small local area, on the one hand, from a radiometric perspective, the intensity of the vegetation is much less than that of the ground; on the other hand, from a geometric perspective, man-made features always have sparse vertical structures, the terrain points is relatively larger and the vertical structure of the terrain points is relatively denser.."), to get terrain points, the proposed MCD-based multiple attributes model also can be used to separate terrain points and off-terrain points, where terrain

points refer to “normal points” and off-terrain points refer to “outliers”. To establish the theoretical background of the proposed MCD-based multiple attribute model, details of the terms of “Mahalanobis distance”, “Minimum Covariance Determinant” and the threshold determinant will be introduced in Section4.1.1, Section4.1.2 and Section4.1.3.



(1)



(2)

Figure 4.1 A spatial data set with multiple attributes: (1) Objects located in the X-Y plane with attribute value (a); (2) Objects located in the X-Y plane with attribute value (b).The height of each vertical line segment represents the attribute value of the corresponding object

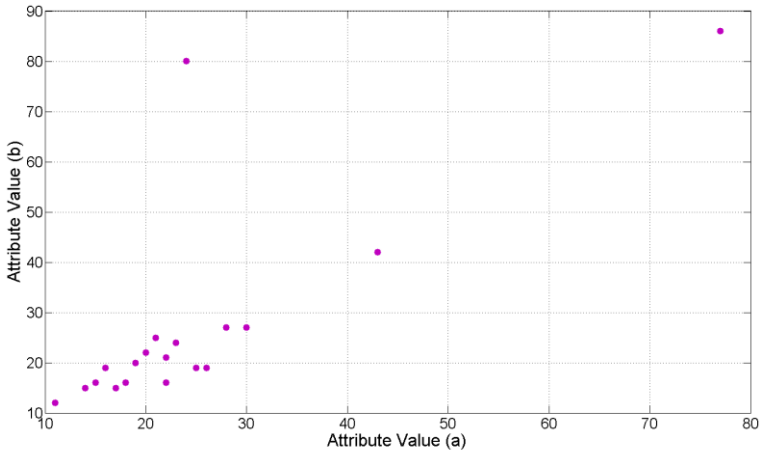


Figure 4.2 Space generation by using the extracted key attributes

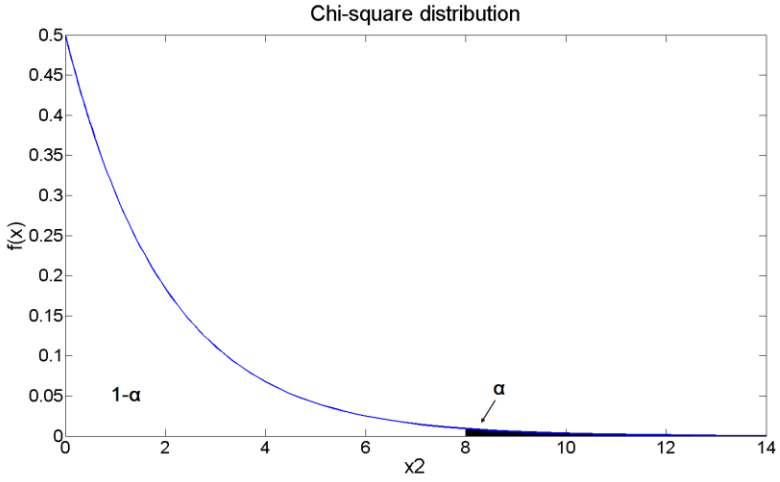


Figure 4.3 Curve of the Chi-square distribution with 2 degrees of freedom

4.1.1 Mahalanobis distance

Given a multivariate data set, it is very popular to determine whether an observation is outlier by calculating its Mahalanobis distance. As is one of the basic and standard outlier detection approaches, the Mahalanobis distance provides a suitable way to flag points which are far from all of the others ("center" of the data set) in a

multidimensional space as outliers. It has major advantages over traditional Euclidian distance when dealing with multivariate data. For instance, the Euclidian distance treats each variable as equally important in calculating the distance, while Mahalanobis distance automatically accounts for the scaling of the coordinate axes (Chen et al, 2008).The notion of the Mahalanobis distance is described as follows:

Given a multivariate data set:

$$X_n = \{x_1, \dots, x_n\} \tag{4.1}$$

with n observations

$$x_i = \left(x_{i1}, \dots, x_{ip}\right)^t \tag{4.2}$$

$i = 1, \dots, n$ in p dimensions.

Thus, to investigate whether such multivariate dataset appears as a homogeneous group or contains outlier points, the Mahalanobis distances of the observations are usually calculated, given by

$$MD_{x_i} = \sqrt{(x_i - \bar{x}_n)^t {S_n}^{-1} (x_i - \bar{x}_n)} \quad i = 1, \dots, n \tag{4.3}$$

Where $\bar{x}_n$ is the sample mean and S_n the sample covariance matrix of the data set. The sample mean and the sample covariance represent the "center" and the "shape" of the data set, obviously, a larger Mahalanobis distance which far from the "center" indicates a more possible of an outlier.

4.1.2 Minimum covariance determinant

Since data set always contains outliers, sample mean and the sample covariance matrix estimated by classical methods can be always highly affected by these outlying values that make the Mahalanobis distances tools can no longer detect the outliers. To get a reliable analysis of these data, robust estimators are required that can resist possible outliers. The MCD estimator is such a robust estimator (Hubert and Debruyne, 2009). It is a super robust statistic estimator of location and scatter (Rousseeuw, 1984&1985). Being resistant to outlying observations (points) which makes the MCD very sensitive in outlier detection.

Given a spatial dataset

$$X = \{x_1, x_2, \dots, x_n\}, \tag{4.4}$$

the MCD of those data is the mean and covariance matrix based on the sample of size h ($h \leq n$) that minimizes the determinant of the covariance matrix. That is,

$$MCD = (\hat{\mu}_J^* , \hat{\sum}_J^*) \tag{4.5}$$

$$J = \left\{set\ of\ h\ points: \left|\hat{\sum}_J^*\right| \leq \left|\hat{\sum}_M^*\right| \forall\ set\ M\ s.t. |M| = h\right\} \tag{4.6}$$

$$\hat{\mu}_J^* = \frac{1}{h}\sum_{i\in J} x_i \tag{4.7}$$

$$\hat{\sum}_J^* = \frac{1}{h}\sum_{i\in J}\left(x_i - \hat{\mu}_J^*\right)\left(x_i - \hat{\mu}_J^*\right)^T \tag{4.8}$$

The value h can be thought of as the minimum number of points which must not be outlying. While, the determination of "h" is quite challenge: if the 'h' is too large (beyond the number of normal samples), it may make the MCD lack of robustness; if the

'h' is too small (much less than number of normal samples), it may make the MCD lack of accuracy.

The MCD has its highest possible breakdown at

$$h = [(n + p + 1)/2] \tag{4.9}$$

Where [.] is the greatest integer function (Rousseeuw and Leroy 1987; Lopuha ä and Rousseeuw 1991). Obviously, the 'h' here is very small, and it is considered as the most conservative estimate (Lopuha ä and Rousseeuw 1991). There are many methods have been proposed by researchers to determine a precise 'h', in this study, we follow Wu and Yan's (2008) solution: a so called 'Modified MCD', that is:

The M-MCD starts from the most conservative estimate:

$$h_0 = [(n + p + 1)/2] \tag{4.10}$$

through an adaptive iterative process, the new 'h' will be continuously updated to find out the most precise 'h', named as 'h^*'.

For n normal distributed samples in p-dimensional space, the square of their Mahalanobis distances are distributed as the chi-square distribution and the degrees of freedom is 'p', that is

$$MD_{x_i}{}^2 \sim \chi_p^2(\alpha) \tag{4.11}$$

For the degrees of freedom are 'p' of the chi-square distribution, the unbiased estimation of standard deviation is:

$$D(\chi_p^2(\alpha)) = \sqrt{2p} \tag{4.12}$$

where $D(.)$ represents for the unbiased estimation of standard deviation.

The adaptive iterative process is based on the deviation between the standard deviation of the square of the robust Mahalanobis distances based on MCD and the theoretic standard deviation ($\sqrt{2p}$) to determine the most precise h^*.

Suppose, for n samples $x_1, x_2, \dots, x_n$ in the p-dimensional space under the parameter 'h_k', the $\hat{\mu}_{\mathrm{n-MCD}}$ and the $\widehat{\sum}_{n-MCD}$ are $\bar{X}_{\mathrm{MCD}_{h_k}}$ and $S_{\mathrm{MCD}_{h_k}}$ separately. Now, the adaptive iterative process to determine the parameter h_{k+1} is described as follows:

(1) Calculation of the robust Mahalanobis distances for the n samples:

$$RMD_{x_i} = \sqrt{\left(x_i - \bar{X}_{\mathrm{MCD}_{h_k}}\right)^t {S_{\mathrm{MCD}_{h_k}}}^{-1} \left(x_i - \bar{X}_{\mathrm{MCD}_{h_k}}\right)} \quad i = 1, \dots, n \tag{4.13}$$

(2) Order the square of the robust Mahalanobis distances for the n samples from small to large sequence, recorded as ${rmd_1}^2, \dots, {rmd_1}^2$, and then, start from constant $C = [(n + p + 1)/2]$, calculate the standard deviation of the square of the robust Mahalanobis distances $\sigma(j)$ as follows:

$$\begin{cases} \sigma(j) = \sqrt{\sum_{i=1}^{j+c-1}(({rmd_{\mathrm{i}}}^2 - \mu(j)^2)/(j + c - 2)} \\ \mu(j) = (\sum_{i=1}^{j+c-1}({rmd_{\mathrm{i}}}^2)/(j + c - 1) \end{cases} \tag{4.14}$$

(3) Calculate the rate of deviation between the standard deviation of the square of the robust Mahalanobis distances $\sigma(j)$ and the theoretic standard deviation ($\sqrt{2p}$):

$$\Delta\sigma(j) = \left|\sigma(j) - \sqrt{2p}\right| \quad j = 1, \dots, n - c + 1 \tag{4.15}$$

(4) Find out the minimum $\Delta\sigma(j)$, and mark it as $\Delta\sigma(j^*)$, order h_{k+1} as follow:

$$h_{k+1} = c + j^* - 1 \tag{4.16}$$

According to the above adaptive iterative process, h_k will be updated to h_{k+1}, then h_{k+1}will be updated to h_{k+2}, and so forth, until h no change, and the h here is the final definite optimal parameter 'h^*'.

Then we calculate the robust distance based on MCD:

$$RMD_{x_i} = \sqrt{\left(x_i - \hat{\mu}_{\mathrm{MCD}_{h^*}}\right)^t \hat{\sum}_{\mathrm{MCD}_{h^*}}{}^{-1} \left(x_i - \hat{\mu}_{\mathrm{MCD}_{h^*}}\right)} \quad i = 1, \ldots, n \tag{4.17}$$

Where $\hat{\mu}_{\mathrm{MCD}_{h^*}}$ is the MCD estimate of location which use h^* relative clean data, and $\hat{\sum}_{\mathrm{MCD}_{h^*}}$ the MCD covariance estimate which use also h^* relative clean data. For n observations there are n robust Mahalanobis distances.

4.1.3 Threshold determination

As mentioned before, a larger Mahalanobis distance indicates a higher possibility of an outlier. While, how large such a distance is large enough, a predetermined cutoff value is needed, this is based on the following observation:

(1) Based on the *central limit theorem*, Duda et al. (2001), naturally measured samples will be distributed as a normal distribution, and then, we could get that RMD_{x_i} follows a multivariate normal distribution.

(2) If RMD_{x_i} is distributed as $N_q(\mu, \sum)$, then ${RMD_{x_i}}^2$ is distributed as χ_q^2, where χ_q^2 is the chi-square distribution with q degrees of freedom. Therefore the probability of outliers that ${RMD_{x_i}}^2$ satisfies

$${RMD_{x_i}}^2 > \chi_q^2(\alpha) \tag{4.18}$$

Here $\chi_q^2(\alpha)$ is the upper (100α)th percentile of a chi-square distribution with q degrees of freedom (see **Figure 4.3**, q=2 as an example).

4.1.4 The multiple attributes model

As introduced above, the MCD estimator appears as a super robust statistic estimator of location and scatter (Rousseeuw, 1984&1985), and being resistant to outlying observations (points) which makes the MCD very sensitive in outlier detection. In this study, we apply such estimator into LiDAR data both for outlier detection and data filtering by calculating the Robust Mahalanobis distance. Since the Mahalanobis approach considers both the average value and its variance and covariance of the attributes measured, it accounts for ranges of variance between attributes and compensates for interactions (covariance) between attributes. For outlier detection, the multiple attributes refer to the height attribute and the COF attribute. While, for data filtering, the multiple attributes refer to the height attribute and the intensity attribute. The data processing model provides a platform to process multiple attributes data, and even for further applications, the attributes may be even more. To illustrate the process of the proposed method, a bivariate simulated data set is considered. A scatter plot of the bivariate simulated data is shown in **Figure.4.4**, together with the classical and the robust tolerance ellipse generated by Mahalanobis distances and robust Mahalanobis

distances based on MCD separately, where the black cycles represent as normal points, while, the red points represent as outlier points. From the figure, we can see, only three outliers can be detected by using the classical Mahalanobis distances method, while the total number is eight. By contrasting, most outliers can be flagged out by using the robust Mahalanobis distances method based on MCD.

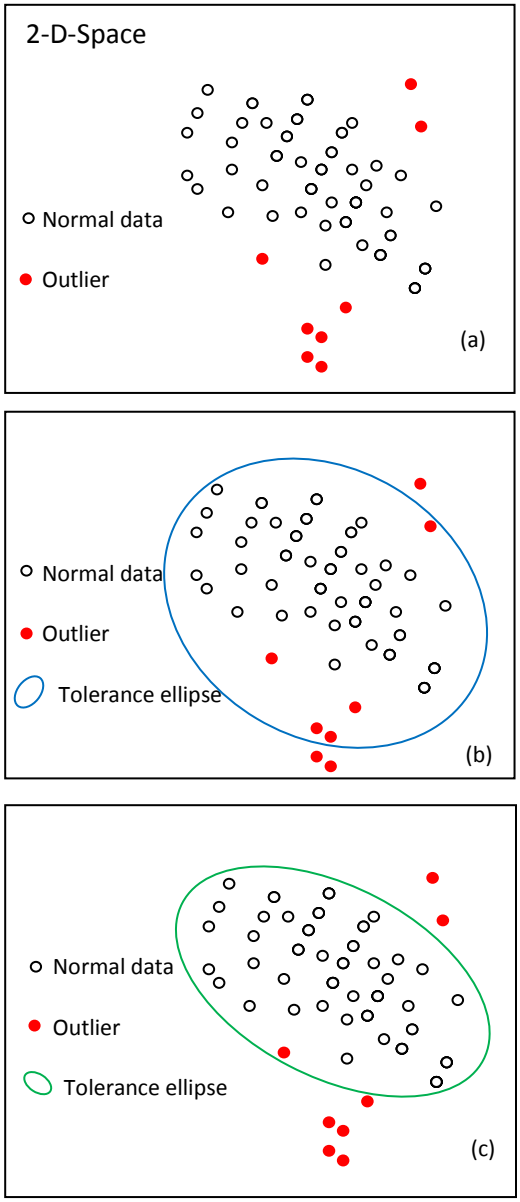


Figure 4.4. Illustration of the process of the MCD-based multiple attribute model for a bivariate simulated data: (a) a 2-D space with normal data and outliers (b) Use classical estimator calculate$[\mu, \sum]$, and then calculate the MD_{x_i}. Since $MD_{x_i}{}^2$ is distributed as χ^2_q, outlier points $= MD_{x_i}{}^2 > \chi^2_q(\alpha)$. Only three of the total eight outlier points are detected, tolerance ellipse is then generated by using the 'clean' data; (c), Use MCD estimator calculate $[\widehat{\mu}_{n-MCD}, \widehat{\sum}_{n-MCD}]$, and then calculate the$RMD_{x_i}$. Since $RMD_{x_i}{}^2$ is also distributed as χ^2_q, outlier points $=RMD_{x_i}{}^2 > \chi^2_q(\alpha)$. Most of the total eight outlier points are detected; tolerance ellipse is then generated by using the 'clean' data.

CHAPTER 5 Outlier Detection in LiDAR Data with Multiple Attributes

As discussed in the previous section (see Section1.1), LiDAR has emerged as a robust technique for high accuracy in the survey of terrestrial landscapes (Bretar et al., 2003), and even taken the place of traditional photogrammetric approaches. However, due to the existence of outliers in LiDAR data, automated processing of the raw data is not always successful. Since in the data filtering step, many of the filtering algorithms work on the assumption that a lowest point must belong to the terrain points, however, in cases where the lowest point is an outlier, the assumption is totally wrong, same cases may happen where the highest point is an outlier. These cases may introduce errors to DEM, therefore, outlier detection issue is frequently discussed in the LiDAR-drived DEM quality control and accuracy assessment (Höhle, 2009; Aguilar and Mills, 2008; Peng and Shih, 2006; Akca et al., 2009). In addition, outlier detection also attracted a lot of attention in the process of automatic classification, building extraction (3-D reconstruction) and city modeling of raw LiDAR data (Forlani et al., 2006; Chehata et al., 2008). Therefore, outlier detection becomes an essential preprocessing step for overall LiDAR data filtering and modeling, and has been addressed by many researchers (Amiri and Sargent, 2007; Sotoodeh, 2006 & 2007; Eisenbeiss, 2009; Chen et al., 2007; Meng et al., 2009; Silván-Cárdenas and Wang, 2006; Wang et al., 2005; Kobler et al., 2007; Arefi et al., 2007; Sithole and Vosselman, 2004; Höhle, 2009).

There are many kinds of outlier detection approaches, commonly, based on the classification of Papadimitriou et al., (2003), these approaches were divided into five major categories, and they are: distribution-based, depth-based, clustering-based, distance-based and density-based (see Section2.1). Specially, according to the outlier characteristics in LiDAR data sets: outliers appearing in LiDAR point clouds can be both single points and also small clusters with elevations, either much higher or lower than the surrounding points, the frequency distribution of elevation values method (Meng et al., 2009; Silván-Cárdenas and Wang, 2006; Wang et al., 2005) which belongs to the distribution-based approach, the mathematical morphology method(Chen et al., 2007; Kobler et al., 2007), and the density-based method (Sotoodeh, 2006 & 2007) have widely proposed by researchers. However, as summarized before (see Section1.2), most of the existing outlier detection schemes could only identify single outliers, and potentially misclassify normal objects as outliers by using single attribute: elevation or spatial neighborhood relationship (such as "LOF"). To accurately detect both single and cluster outliers in LiDAR data, in this section, we will use the proposed MCD-based multiple attributes model to achieve the outlier detection by using multiple attributes: elevation and spatial neighborhood relationship. Firstly, we define the connectivity based outlier factor (COF) which also indicates the spatial neighborhood relationship of an point as an attribute; then the COF attribute and the height attribute are extracted from LiDAR data to organize a 2-D space; lastly, in the formed 2-D space, the MCD-based multiple attributes model is conducted to identify outliers.

5.1 Attributes extraction in LiDAR data for outlier detection

Based on the characteristics of outliers in LiDAR data set, which appearing whether 'too high or too low', in the form of single (isolated points in most cases) or clusters, two significant attributes are extracted to illustrate the issue of outlier detection in LiDAR data set: They are COF information and height information respectively. The former one is a spatial neighborhood relationship function which is used to indicate the spatial connectivity of a point to its neighborhoods. The latter one is a location attribute which is used to show the vertical location information of a point. Both of the two attributes are introduced in detail as bellows:

5.1.1 COF attribute

In this study, the connectivity-based outlier factor (COF) scheme which is a density-based outlier factor will be applied to LiDAR data. COF improves the effectiveness of LOF and is considered as a robust outlier detection scheme for large data sets (Tang et al., 2002). By determine the "isolativity", which refers to the degree that an object is connected to other objects, an object with higher isolativity can be picked out as an outlier. By following notations, the proposed connectivity-based outlier scheme will be formulated as follows (Tang et al., 2002):

DEFINATION 1 (Nearest neighbor):

Let $P, Q \subseteq D$, $P \cap Q = \emptyset$ and $P, Q \neq \emptyset$.We define $dist(P,Q) = min\{dist(x,y) : x \in P \,\&\, y \in Q\}$, and where we call $dist(P,Q)$ the distance between P and Q. For any given $q \in Q$, we say that q is the nearest neighbor of P in Q if there is a $p \in P$ such that $dist(p,q) = dist(P,Q)$.

DEFINATION 2 (SBN-path):

Let $G = \{p_1, p_2, \dots, p_r\}$ be a subset of D. A set based nearest path (SBN-path), from p_1 on G is a sequence $\langle p_1, p_2, \dots, p_r \rangle$, such that for all $1 \leq i \leq r-1$, p_{i+1} is the nearest neighbor of set $\{p_1, p_2, \dots, p_i\}$ in $\{p_{i+1}, p_{i+2}, \dots, p_r\}$.

Finding SBN-path is an iterative process which applies to every object in the data set. And the SBN-path of an object shows the order of its nearest neighborhood objects.

DEFINATION 3 (SBN-trail):

Let $s = \langle p_1, p_2, \dots, p_r \rangle$ be an SBN-path. A set based nearest trail (SBN-trail) with respect to s is a sequence $\langle e_1, e_2, \dots, e_{r-1} \rangle$, such that for all $1 \leq i \leq r-1$, $e_i = (o_i, p_{i+1})$ where $o_i \in \{p_1, p_2, \dots, p_i\}$, and $dist(e_i) = dist(\{p_1, p_2, \dots, p_i\}, \{p_{i+1}, p_{i+2}, \dots, p_r\})$. We call each e_i an edge and the sequence $\langle dist(e_1), \dots, dist(e_{r-1}) \rangle$ the cost description of $\langle e_1, e_2, \dots, e_{r-1} \rangle$.

By recording distances of an object to its nearest neighbors, distances or edges are then sequenced as a trail. This is also an iterative process which will apply to every object in the data set.

DEFINATION 4 (Average chaining distance):

Let $s = \langle p_1, p_2, \dots, p_r \rangle$ be an SBN-path from p_1 and $e = \langle e_1, e_2, \dots, e_{r-1} \rangle$ be the SBN-trail with respect to s.The average chaining distance from p_1 to $G - \{p_1\}$, denoted by $ac\_dist_G(p_1)$ is defined as

$$ac\_dist_G(p_1) = \sum_{i=1}^{r-1} \frac{2(r-i)}{r(r-1)} \, dist(\mathrm{e_i}) \tag{5.1}$$

The average chaining distance from p_1 to $G\_\{p_1\}$ is the weighted sum of the cost description of the SBN-trail for some SBN-path fromp_1. Since this cost description is unique for p_1, the definition is well defined. Rewriting

$$ac\_dist_G(p_1) = \frac{1}{r-1}\sum_{i=1}^{r-1}\frac{2(r-i)}{r}\ dist(e_i) \tag{5.2}$$

DEFINATION 5 (COF):

Let $p \in D$ and *k* be a positive integer. The connectivity-based outlier factor (COF) at p with respect to its k-neighborhood is defined as

$$COF_k(p) = \frac{|N_k(p)| * ac\_dist_{N_k(p)}(p)}{\sum_{o \in N_k(p)} ac\_dist_{N_k(o)}(o)} \tag{5.3}$$

Where，

$N_k(p)$: *k nearest neighbors of element p;*

$|N_k(p)|$: *Numbers of* $N_k(p)$

$ac\_dist_{N_k(p)}(p)$: *The average chaining distance from p to its k nearest neighbors;*

$ac\_dist_{N_k(o)}(o)$: *The average chaining distance from o to its k nearest neighbors*

The connectivity-based outlier factor at p is the ratio of the average chaining distance from p to $N_k(p)$ and the average of the average chaining distance fromp' k-distance neighbors to their own k-distance neighbors. And it indicates how large the isolativity an object is respected to its neighbors.

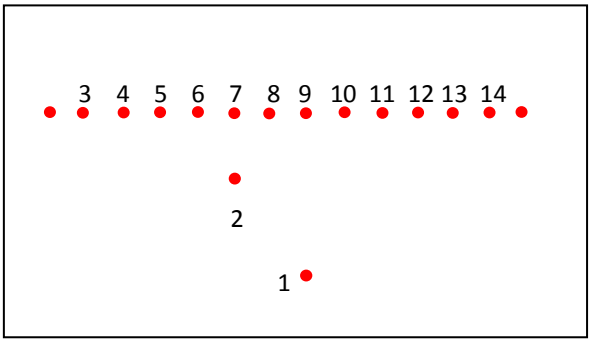


Figure 5.1 Calculating COF (reproduced from Tang et al., 2002)

To illustrate the process of calculating COF, we take the data set in **Figure 5.1** for example. The figure shows a individual line with two points shift away from it (Tang et al., 2002). Suppose $dist(1,2) = 5$, $dist(2,7) = 3$, the distance between two adjacent points in the single line is 1. Let $k = 5$, we now calculate the COF values of three representative points, point 1, point 2 and point7 respectively:

For point 1, $N_k(p) = N_5(1) = \{2, 9, 10, 8, 7\}$. The SBN-path from 1 on $N_5(1) \cup \{1\}$ is:

$$s1 =< 1,2,7,8,9,10 >$$

The SBN-trail for $s1$ is

$$tr1 =< (1,2), (2,7), (7,8), (8,9), (9,10) >$$

The cost description of $tr1$ is

$$c1 =< 5, 3, 1, 1, 1) >$$

And:

$$ac\_dist_{N_k(1)}(1) = 2.87$$

Then we can also get

$$ac\_dist_{N_k(2)}(2) = 1.67$$

$$ac\_dist_{N_k(7)}(7) = 1.00$$

Based on formulation 5.3, finally we can obtain that

$$COF_5(1) = 1.46$$

$$COF_5(2) = 1.18$$

$$COF_5(7) = 0.84$$

The average chaining distance for other points as well as their COF values can be calculated similarly. The above results indicate that the more a point shift from the "pattern", the more items in front of the their cost description lists tend to be lager values, and the lager average chaining distance as well as the COF value. And the COF values of points in the "pattern" should be close to 1.

5.1.2 Height attribute:

With emergence of airborne LiDAR technology, accurate elevation data can be acquired and be better than any other spectral images because of their dependence of color and texture information (Forlani et al., 2006). Thus, as a major characteristic of LiDAR data, the height data is normally used to generate digital terrain models (DSMs) or digital elevation models (DEMs), realizes 3-D profiles (like buildings) visualization, classify roof structures (Alexander et al., 2009). Furthermore, height texture information (variation in height) is also widely used in various LiDAR data-based applications, such

as building extraction, tree identification. The texture information includes several possibilities such as standard deviation, absolute deviation from the mean, and the difference between the maximum and minimum height values (Charaniya, 2004; Parian and Sargent, 2007). Usually, in these applications, the local height frequency distribution histograms are conducted. Two data sets in Wang and Glenn's (2009) study illustrate this issue (see **Figure 5.2**). From **Figure 5.2 (a)**, we can see that height data in bare-earth distributed as a Gaussian distribution if the sample data is large enough; while, in the vegetated area, and the height data is then distributed as a bimodal Gaussian curve (see **Figure 5.2 (b)**). Furthermore, in most cases, the mean height and standard deviation of the height of the forest canopy are used to illustrate and summarize forest structure. For outlier detection in this study, we only use the raw height information to do further analysis.

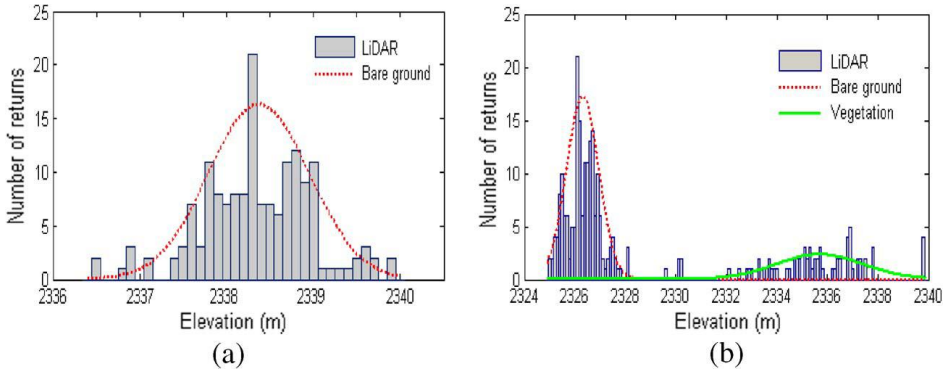


Figure 5.2 Frequency distributions of height (elevation) histograms: (a) for bare-earth area; (b) for vegetated area (Wang and Glenn, 2009)

5.1.3 2-D space generation

After attributes extraction, a 2-D space is formed based on the extracted COF and height attributes (see **Figure 5.3**). The horizontal ordinate represents the elevation values, while

the vertical ordinate represents the COF values. In the formed 2-D space, the proposed MCD-based multiple attributes model will be conducted.

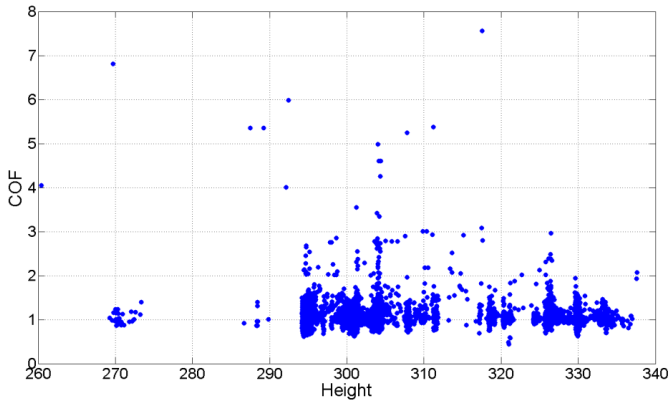


Figure 5.3 Formed 2-D space based on the height and COF attributes: the horizontal ordinate represents the height values, while, the vertical ordinate represents the COF values

5.2 Cut-off value for the chi-square distribution

An important work need to be solved is to determine the cut-off value for the chi-square distribution. As discussed in the previous section (see Section4.2.3), the ${RMD_{x_i}}^2$ is distributed as χ_q^2, where χ_q^2 is the chi-square distribution with q degrees of freedom. Therefore points in 'clean' data that ${RMD_{x_i}}^2$satisfies ${RMD_{x_i}}^2 < \chi_q^2(\alpha)$.Here $\chi_q^2(\alpha)$ is the upper (100α)th percentile of a chi-square distribution with q degrees of freedom. In this study, since the MCD is conducted in a 2-D space, $q = 2$.Then how to determine a proper α becomes as a key issue. To avoid misclassification due to the "α" (if it is a large α, more points will be removed as outliers, if it is a small α, outliers will not be removed completely), in this study, $\alpha = 5\%$ which is a mezzo value is used. By define

the cut-off value $\chi_q^2(\alpha) = \chi_2^2(0.5)$; Points whose ${RMD_{x_i}}^2 > \chi_2^2(0.5)$ are regarded as outliers.

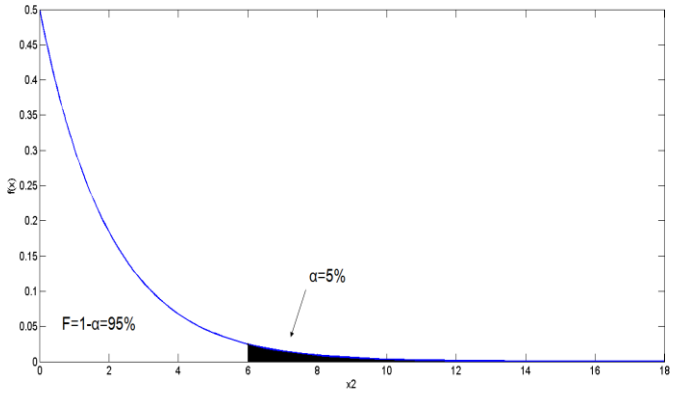


Figure 5.3 Illustration of the chi-square distribution curve with 2 degrees of freedom: the white area is the integral of the distribution from 0 to 5.99, and 95 percent of the area under the curve is to the left of 5.99, or the upper tail is 5 percent (the rest), points which fall into the black area are flagged to outliers.

CHAPTER 6 Filtering LiDAR Data with Multiple Attributes

Data filtering is an essential step for DEM generation in the overall LiDAR data processing duration. Literature review show that most existing filtering methods are mainly based on the analysis of geometrical information of LiDAR points, while, radiometric information such as intensity data is seldom used. Being as a robust technique for high accuracy in the survey of terrestrial landscapes (Bretar et al., 2003), it provides not only high accuracy geometric information which mainly refers to height data, but also the radiometric information which mainly refers to intensity data. Since the height data and the intensity are simultaneously generated on the same platform, both the two data describe the same features geometrically, although it has challenges to calibrate the raw intensity data which always has speckle noise, the comprehensive utilization of both the height and intensity data simultaneously provided by LiDAR may be advantageous over using either data individually (Wang and Glenn, 2009). Similar suggestions can be found in Clément Mallet (2009) and Vosserman's (2010) works, they pointed out that intensity even more radiometric information (full-wave form data) could be utilized as additional information to improve the filter or classifier performances. Researchers have investigated on this issue for years, however, most of their attentions are mainly focused on the forestry areas. Since in the forestry areas, features are relatively few (probably most features are vegetation and bare earth), and vegetated structures are relatively simple and likeness comparing with urban areas which has

various features and more complex structures, it then has great potential to use intensity data to separate different features in such area. While, in urban areas, it seems the potentialities of intensity are less obvious without data fusion such as remote sensing images which attract few researchers to investigate on this study.

In this study, the potential of using the intensity data to filter LiDAR data in urban area is analyzed, a data filtering scheme by using both the geometric information (intensity data) and radiometric information (height data) to separate terrain points and off-terrain points is introduced. The basic process steps of the proposed scheme are summarized as follows: (1) The geometric information (height data) and radiometric information (intensity data) are extracted from LiDAR points as the two attributes; (2) organize a 2-D space based on the generated two attributes; (3) in the formed 2-D space, the MCD-based multiple attributes model is conducted to separate terrain points and off-terrain points. However, in real world, urban areas always have various features and more complex structures, in order to achieve the filtering process successfully, several preprocessing works need to be done which will be detailed introduced in this chapter.

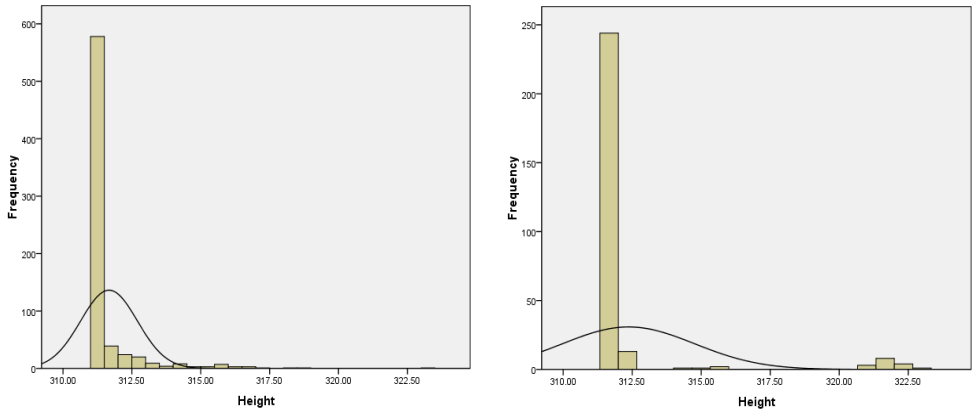
6.1 Attributes extraction in LiDAR data for data filtering

6.1.1 Height attribute

As analyzed in Section5.1.2, with emergence of airborne LiDAR technology, accurate elevation data can be acquired and be better than any other spectral images because of their dependence of color and texture information (Forlani et al., 2006). Thus, as a major characteristic of LiDAR data, the height data is normally used to generate digital terrain models (DSMs) or digital elevation models (DEMs), realizes 3-D profiles (like buildings) visualization, classify roof structures (Alexander et al., 2009). Furthermore, height

texture information (variation in height) is also widely used in various LiDAR data-based applications, such as building extraction, tree identification. The texture information includes several possibilities such as standard deviation, absolute deviation from the mean, and the difference between the maximum and minimum height values (Charaniya, 2004; Parian and Sargent, 2007). Usually, in these applications, the local height histograms are conducted.

Under the consideration of height texture information, it is easy to find that in a small local area, man-made features always have sparse vertical structures, while the ground has relatively dense vertical structure because of the planimetric resolution of the LiDAR points. We use the height frequency distribution of two patches areas in data Samp31 as examples to explain this issue. From **Figure 6.1** we can see that, in a local area, LiDAR data have an approximate Gaussian distribution if there are enough samples in a patch, points which belong to the ground are crowded with similar height values, while, points which belong to man-made features are sparse with different height values. The discriminating performance of ground and man-made features makes it as a criterion for filtering by using the height data.



(a) (b)

Figure 6.1 Frequency distributions of the height value in a small local area (a and b are two patches in data Samp31)

6.1.2 Intensity attribute

In addition to the height information, intensity information which refers to the backscattered energy reflected back from the terrain to the laser scanner is also acquired. Although suffering from the disturbance of noise, there are still augments to support the intensity data as an attribute information to filter LiDAR data. They are: (1) the intensity data has less influence on shadowing effect and relief displacement which are two of the major issues faced in high resolution optical remote sensing image (2) it has high separability of surface reflectance in the spectrum range of the near infrared and short-wavelength near infrared spectrum (1064 nm or 1550 nm) (see **Figure 6.2**) under where commercial airborne LiDAR is operated, (3) in a small local area, the intensity of the vegetation is much less than that of the ground, all the three arguments make it believable that the intensity data could be used to improve the data filtering in LiDAR data.

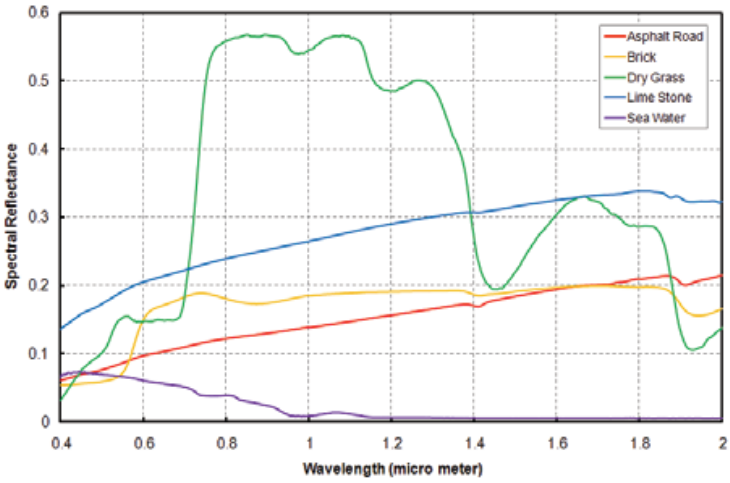


Figure 6.2 Spectral reflectance of different land cover features (Yan and Shaker, 2010)

6.1.3 2-D space generation

After attributes extraction, a 2-D space is formed based on the extracted height and intensity information (see **Figure 6.3**). In the formed 2-D space, the MCD-based multiple attributes model will be conducted to separate terrain points and off-terrain points. As explained in Section6.2.1 and Section6.2.2, for a small local area, on the one hand, from a radiometric perspective, the intensity of the vegetation is much less than that of the ground; on the other hand, from a geometric perspective, man-made features always have sparse vertical structures, the terrain points is relatively larger and the vertical structure of the terrain points is relatively denser. The two arguments mentioned above are considered as the criteria of proposed method in this study.

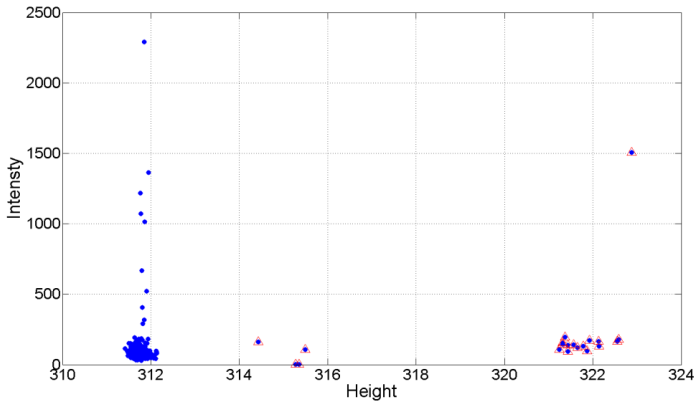


Figure 6.3 2-D-space generation by using height and intensity data. Blue points are Ground features (terrain points); man-made features (off-terrain points) are enveloped by red triangles

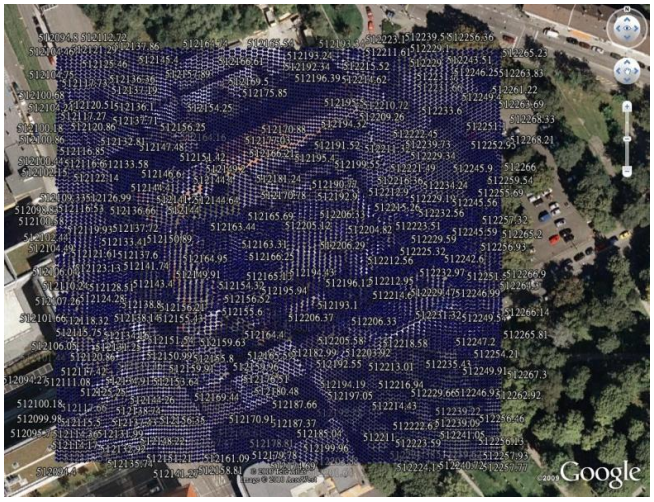
6.2 Preprocessing works

In real world, urban areas always have various features and more complex structures, in order to achieve the filtering process successfully, several preprocessing works need to be done:

6.2.1 Local area determination

Since the MCD estimator works on the assumption that in a small local area, the number of the terrain points is relatively larger and the vertical structure of the terrain points is relatively denser. The experimental target regions will be divided into regular patches (for example, 20 m squares), and the patch size should be large enough to make sure the patch contains terrain points. Besides, because MCD is a statistic estimator, it also should have enough samples to satisfy the statistical analysis. In Wang and Glenn's (2009) experiments, more than 180 points were used in a window (patch). A training dataset which is used to estimate the parameters of the proposed filter is conducted. The filter is then tested on the other datasets referred to as the test datasets. Such process

follows a very important assumption is that the distribution of features is similar between the test dataset and the train dataset to have a good performance of the filter. In this study, data Samp31 is used as the training dataset and the other eight datasets are used as test datasets. For obtaining ground truth as reference data, Google Earth which is a very popular virtual global product is used to visually display these features and offers a rough impression on the target region. After certain coordinates and formats transformation, the LiDAR points are plotted on the Google Earth in their reference positions (see **Figure 6.4**-(a)). As described in **Table3.1** and displayed in the Google Earth, terrain features like 'densely packed buildings with vegetation between them', building with eccentric roof, open space with mixture of low and high features, data gaps' could be found. Profile views of the target region (Samp31) as well as the 3-D buildings layer in Google earth are shown in **Figure 6.4**-(b) and **Figure 6.4**-(c). There are total 28862 points, and the area is about 28188 m2, therefore, we can get the planimetric resolution is about 1 point/m2. We use the measurement tools provided by the Google Earth to measure the largest building in the target region (see **Figure 6.4**-(d) marked in red polygon), and got an approximate area of such building is 400m2. Then the target region follows 8*8 grid divided into 64 pitches (see **Figure 6.5**), each pitch may have about 450 points, and each pitch may have an about 450m2 area to make sure each pitch have terrain point.



(a)

(b)

(c)

(d)

Figure 6.4 Features in Google Earth: (a) Plotted LiDAR points of the target region (Samp31) in Google Earth; (b) Profile view of the target region (Samp31); (c) Profile view of the target region (Samp31) with 3-D buildings; (d) largest building marked in red polygon

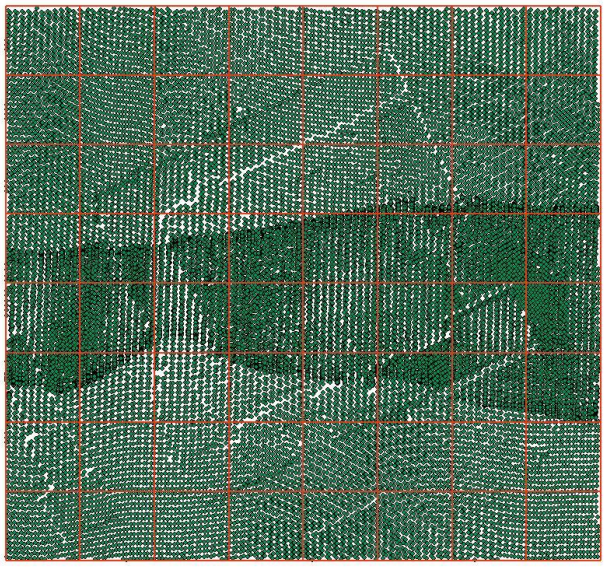


Figure 6.5 Partition of the experimental region: it follows a 8*8 grid and divided into 64 pitches

6.2.2 Threshold determination

In Section6.2.1, we have successfully determined the patch size to make sure the patch contains terrain points for further MCD application and have enough samples (points) to satisfy the statistical analysis. However, although the patch contains terrain points, it cannot make sure that the terrain points are larger than the off-terrain points which is the criteria for the MCD application.

In a small local area, we consider the characteristics of these points, in a general case, if the terrain points are larger than the off-terrain points, and the terrain points are crowded with similar height values, while, the off-terrain points are sparse with different height values, then we calculate the statistical information of the patch, we then may get Hmean-Hmedian≥0 (Hmean and Hmedian stand for the mean and median height values in a patch, respectively). Building upon this view, we use the Hmean-Hmedian to classify the patches into two major cases: (A) Hmean-Hmedian≥0 and (B) Hmean-Hmedian<0, and CaseB approximately indicates that the number of terrain points is smaller than the off-terrain points.

To illustrate the above cases, we take a simulated data set to address these issues. Suppose there is a selected small local area, in such area, there are total 30 points, in which there are 25 terrain points with height values are around 3 or 4, and there are 5 off-terrain points, and the terrain points are crowded with similar height values, while, the off-terrain points are sparse with different height values (see **Figure 6.6**), then we can get the Hmean=6.2, while, Hmean=3, then Hmean-Hmedian≥0, and it belongs to CaseA.

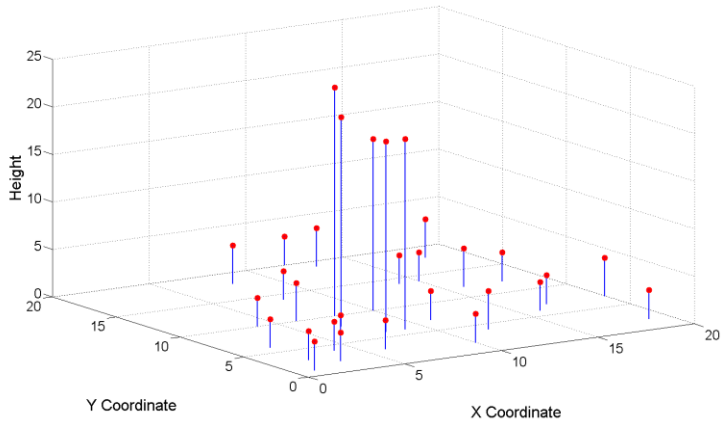


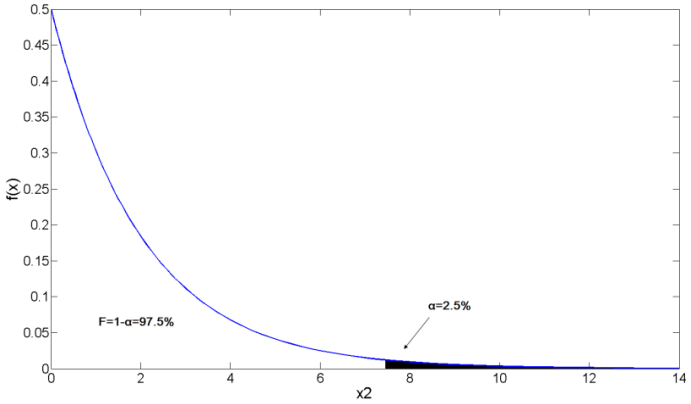
Figure 6.6 Simulated data for CaseA

This is a very rough estimation. For CaseB, we use "median" filter to remove certain points (if the points are larger than the median one, they are removed), MCD will be conducted when Hmean-Hmedian≥0, and then follow operations in CaseA which will detailed introduced as follows:

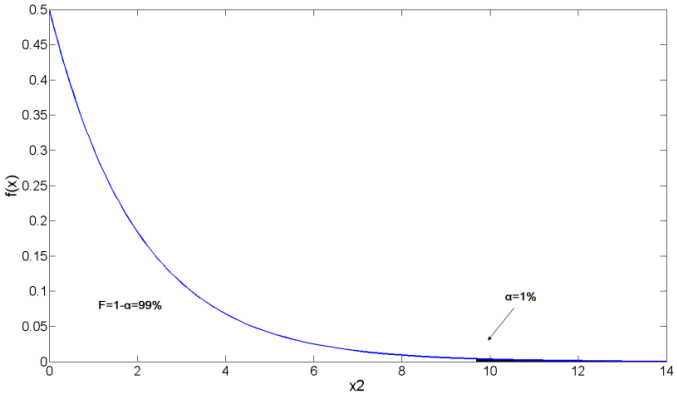
As discussed in the previous section (see Section4.1.3), the ${RMD_{x_i}}^2$ is distributed as χ_q^2, where χ_q^2 is the chi-square distribution with q degrees of freedom. Therefore points in 'clean' data that ${RMD_{x_i}}^2$satisfies ${RMD_{x_i}}^2 < \chi_q^2(\alpha)$.Here $\chi_q^2(\alpha)$ is the upper (100α)th percentile of a chi-square distribution with q degrees of freedom. In this study, since the MCD is conducted in a 2-D space, $q = 2$. Then how to determine a proper α becomes as a key issue. Since terrain points are referring to 'clean' data, and off-terrain points are referring to 'outliers', the standard deviation of height (Hstd) could describe the vertical structure of the data in a patch to a great extent. In this study, Hstd is used to determine the α.To avoid misclassification due to a big "α" (the larger α, more points will be removed), certain relatively very conservative α are proposed with iterations. In this

study $\alpha = 2.5\%$， $\alpha = 1\%$ and $\alpha = 0.5\%$ (see **Figure 6.7**) are used for different Hstd. To have a good matching between α and Hstd, as training in data Samp31, we got the following results: (1) when 0≤Hstd≤0.25, the patch probably is flat, and considered as ground without any operations temporarily; (2) when 0.25<Hstd≤0.5, the most conservative $\alpha = 0.5\%$ is used; (3) when 0.5<Hstd≤1, $\alpha = 1\%$ is used; (4) when 1 <Hstd≤7.5, $\alpha = 0.5\%$ is used; (5) when Hstd>7, the vertical structure probably is very complex, we also use "median" filter to remove certain points (if the points are larger than the median one, they are removed) to have a rough separation, MCD will be conducted when its Hstd≤7, and then follow (1)-(4) to have a fully separation.

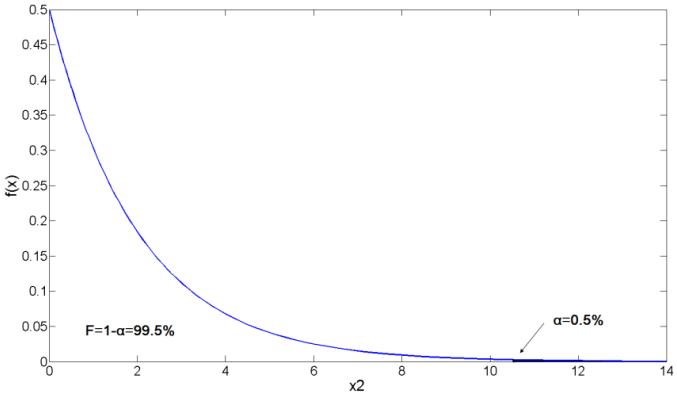
In addition to α for the MCD, a limitation was also applied to data processing to acquire reliable estimations: only when Hmax-Hmin<1m (Hmax and Hmin stand for the maximum and minimum height values in a patch, respectively), the data processing ends, otherwise, it needs iterations with different α by checking its Hstd of the remaining data. Particularly, when 0≤Hstd≤0.25, we check it if Hmax-Hmin<1m, the data processing ends, if Hmax-Hmin≥1m, the most conservative $\alpha = 0.5\%$ is conducted till if Hmax-Hmin<1m.The limitation assumes that, in a small local area, the ground is relatively flat, and the rise and fall in vertical is less than 1m.



(a)



(b)



(c)

Figure 6.7 Illustration of the chi-square distribution curve with 2 degrees of freedom. (a)The left area is the integral of the distribution from 0 to 7.378, and 97.5 percent of the area under the curve is to the left of 7.378, or the upper tail is 2.5 percent (the rest), points which fall into the black area are flagged to outliers; (b)The white area is the integral of the distribution from 0 to 9.21, and 99 percent of the area under the curve is to the left of 9.21, or the upper tail is 1 percent (the rest), points which fall into the black area are flagged to outliers. (c)The white area is the integral of the distribution from 0 to 10.597, and 99.5 percent of the area under the curve is to the left of 10.597, or the upper tail is 0.5 percent (the rest), points which fall into the black area are flagged to outliers.

At last, we consider the weights of both the intensity and height data. Since the intensity data is raw data without calibration, it contains an amount of noises caused by the atmospheric refraction, scattering and absorption and so on, although Yan and Shaker (2010) has pointed out that it is still believable despite the existence of the noise in surface classification, the data should treat less worthy than the height value in this study. Then we give the weight of intensity is 0.6, while, by contrasting, the weight of height is given for 1. Finally, we got a threshold value table as shown in **Table 6.1**.

Table 6.1 Threshold values

Thresholds / Parameters		T1 99.50%	T2 99%	T3 97.50%	Median+ (T1, T2…)
Hstd	0-0.25	0.25-0.5	0.5-1	1-7.5	>7.5
Weight of intensity	0.6				
Weight of height	1				

6.2.3 Complex senses

During the data process, in some patches, their Hmean-Hmedian≥0, and their Hstd is less than 7.5, we conducted the MCD to separate their terrain points and off-terrain points, and however the program were blocked. It suffered the complex senses. Since the proposed α for MCD are relatively conservative, and the MCD estimates the

distances of points to their center, however, when the points are disperse or points are clustered into several groups (see **Figure 6.8**) makes the MCD cannot remove points any longer. When it suffers complex senses as mentioned, the "median" filter will be used until the MCD works, and then follow operations in general senses, Case A or Case B (see Section6.2.2).

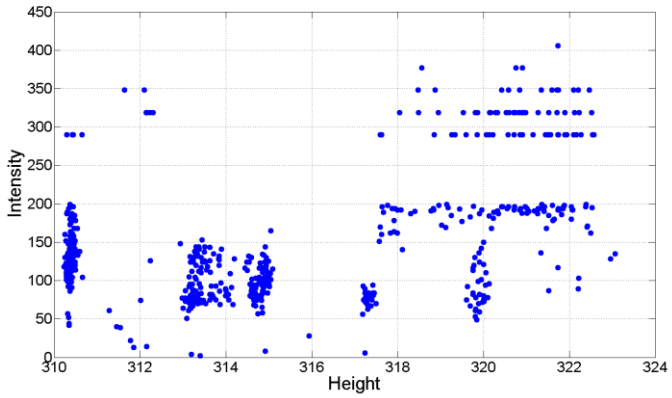


Figure 6.8 Illustration of complex senses

6.3 The post processing step

After apply the MCD to every patch with iterations (if needs), we will get two datasets finally: one for terrain points which are remained by iterations; and one for off-terrain points which are removed by iterations. While, there might be misclassification issues: terrain points which have very high or low intensity values compared with surroundings, they are probably caused by reflectivity, moisture content or roughness of the reflected objects. To avoid this issue, it needs reclassification. We calculate the Hmax and Hmin (Hmax and Hmin stand for the maximum and minimum height values in the remained terrain points, respectively) of the remained terrain points, and then we check the height

value of the removed point, if it satisfies that: Hmax ≤h≤ Hmin, it will be reclassified as terrain points; otherwise, they are off-terrain points (see **Figure 6.9**).

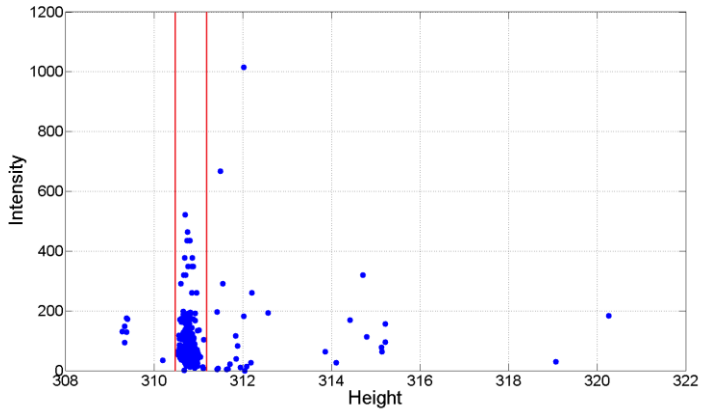


Figure 6.9 Illustration of complex senses the post processing step, points with very high intensity values fall into the red strips will be reclassified as terrain points

A designed framework for the proposed multiple attributes based filter is list as follow (see **Figure 6.10**). The proposed framework comprises three steps (1) extraction the height and intensity attributes from raw LiDAR data to generate a 2-D space; (2) divide the experimental regions into small patches, (3) preparative works before applying the data processing model into the raw LiDAR data (4) conduct the MCD-based multiple attributes model in the formed space to separate terrain and off-terrain points. The threshold values for Hstd is generated by training the data Samp 31, as experience values, such threshold values will be applied to the other data sets

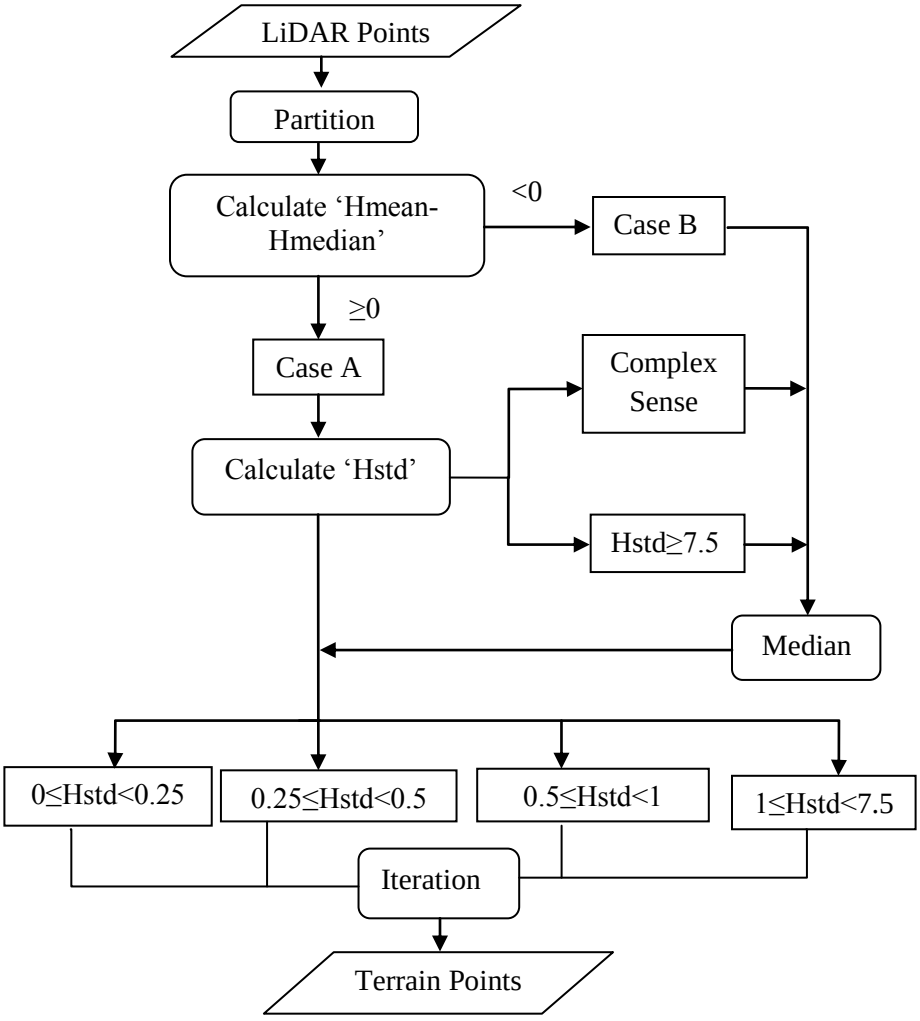


Figure 6.10 Flow chart of the process of data filtering

CHAPTER 7 Experimental Results and Discussion

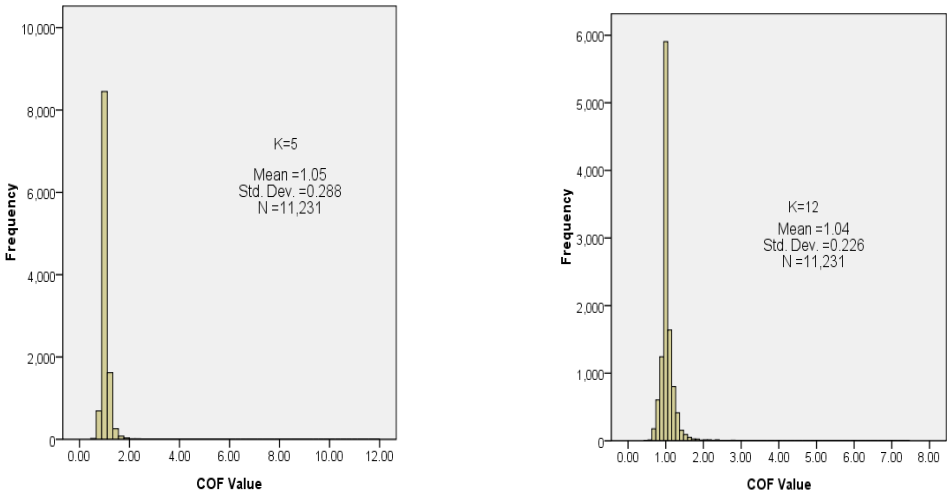
To verify the proposed MCD-based multiple attributes model both for outlier detection and data filtering in LiDAR data, in this chapter, we conduct a series of experimental studies by using the experimental data introduced in Chapter 3. For outlier detection, two typical experimental data "Samp41" and "Samp31" are applied, and the experimental results are presented and analyzed to illustrate the effectiveness the proposed multiple attributes outlier detection approach. For data filtering, nine reference urban sites are applied, both the qualitative and quantitative assessment are generated to evaluate the performance of the proposed filtering method.

7.1 Experimental Results and discussion for outlier detection issue

7.1.1 Determination of the parameter of k for COF

Since the COF at a point p is the ratio of the average chaining distance from p to $N_k(p)$ and the average of the average chaining distance from p' k-distance neighbors to their own k-distance neighbors. Before apply the data processing model into the raw LiDAR data for outlier detection, the work of determination of the parameter of k for COF needs to be done. And how to determine an appropriate k value to appropriately show the spatial neighborhood relationship of a point is an essential issue. There are two cases should be considered: If the k value is too large, it may less sensitive to noise, and lead to higher bias which is less precise, while if the k value is too small, it may cause higher variance which is less stable. Therefore, to determine an appropriate k, these two cases should be balanced. The calculation of COF is started from k=3 (obviously, k=1 or 2

cannot show the spatial neighborhood relationship). Frequency distributions of COF values for different k in Samp41 are illustrated as histograms in **Figure 7.1**, and the COF numbers in different intervals are recorded. Since from the view of the figure, most COF values are intensively appearing from 0.5 to 2, thus, the whole intervals are divided into three parts: 0-0.5, 0.5-2, >2 as shown in **Table 7.1**. In this study, three major issues are explained to balance the mentioned two cases, they are: (1) COF numbers in different intervals for different k; (2) the mean and standard deviation of the COF for different k; (3) tracked max and a common COF values for different k. Since as discussed in Section5.1.1, we can get the conclusion that the COF values of points in the "pattern" are close to 1, if the points are shifted away from the "pattern", they normally have a lager COF value, which are larger than 1, and they have a great possibilities to be outliers. Cases appear in **Figure 7.1** just match such conclusion. Based on such theory, if a data set contains outliers, there should be certain numbers of COF values much larger than 1. In this study, outliers are probably in the interval of ">2", and we can imagine that when we get a proper k value, COF numbers in different intervals, the mean and standard deviation of the COF values and the tracked max and a common COF values all should be stable.



(a) (b)

Figure 7.1 Illustration of frequency distribution histograms of the COF for different k: Take (a) k=5; (b) k=12 for instance, the mean together with the standard deviation of the COF for different k are illustrated as well.

Table 7.1 COF numbers in different intervals for different k

COF / K	0-0.5	0.5-2	>2
3	0	11089	142
4	0	11130	101
5	0	11153	78
6	0	11157	74
7	1	11149	81
8	2	11147	82
9	2	11139	90
10	4	11139	88
11	2	11145	84
12	2	11149	80
13	1	11153	77
14	1	11161	69

From **Table 7.1** we can see, despite the increasing of the k value, most COF values are intensively appearing from 0.5 to 2; seldom COF values are fall into the interval of 0-0.5, that is to say, COF in the interval of 0-0.5 and 0.5-2 are not sensitive to the change of k. Numbers in the interval of >2 are also relatively small, while, it tends to stable from k=5.

Table 7.2 Tracked max and a common COF values as well as the mean together with the standard deviation of the COF for different k

COF / K	Max	Common	Mean	Std.Dev
3	17.9	1.06	1.07	0.455
4	14.3	0.96	1.05	0.335
5	11.9	0.89	1.05	0.288
6	9.6	0.87	1.05	0.262
7	8.3	0.94	1.05	0.249
8	7.6	1.23	1.05	0.242
9	7.6	0.92	1.05	0.236
10	7.4	0.95	1.04	0.23
11	7.3	1.03	1.04	0.226
12	7.4	0.95	1.04	0.226
13	7.4	0.93	1.04	0.224
14	7.3	0.95	1.04	0.223

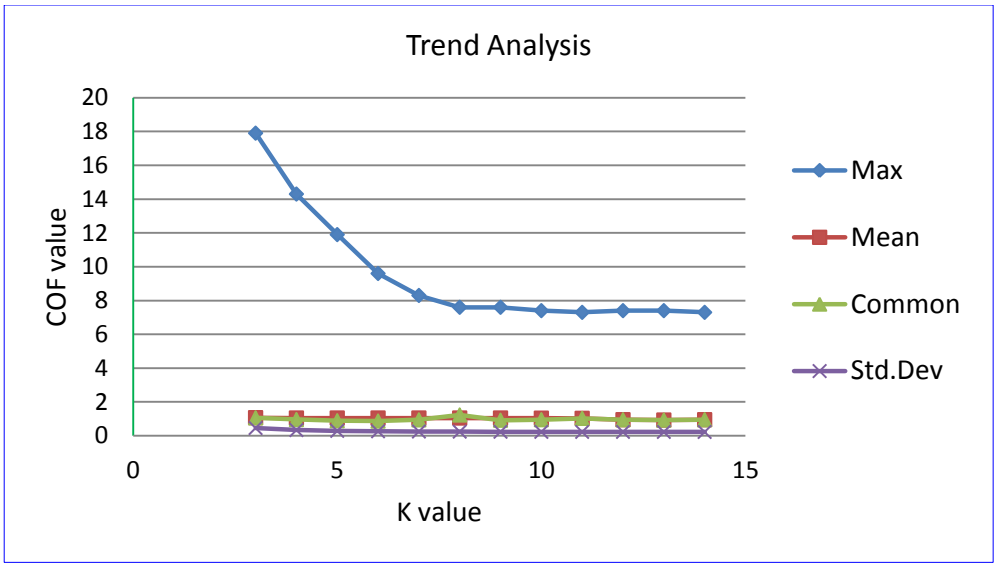


Figure 7.2 Trend of the tracked standard deviation Max common COF value for different k

Then we find out the maximum COF value (=17.9) when k=3, it is in the interval of larger than 2. Within the increasing of k, its performances are tracked and recorded as shown in **Table 7.2**. Correspondingly, **Figure 7.2** illustrates the COF trend within the increasing of k. From **Figure 7.2** we can see that, with the increasing of k value from 3 to 8, the COF value is decreasing of a high rate with a sharp curve, while, in contrast, when the k value comes to 8 and 9, it is appearing a series of stable rates, and the COF value trends to about 7.5 stably. It indicates that the increasing of the k value has considerable influence on points whose COF values are in the interval of >2, however, when the k value is large enough, for example, k=9, it tends to stable.

A very common COF value (=1.06) is picked out which is in the interval of 0.5-2 when k=3. Within the increasing of k, its performances are also tracked and recorded as shown in Table **7.2**. Correspondingly, **Figure 7.2** illustrates the COF trend within the increasing of k. The trend of the tracked common COF value for different k shows a

series of stable rates. Despite the increasing of the k value, the COF value is always wandering about 1. It indicates that the increasing of the k value has less influence on points whose COF values are in the interval of 0.5-2.

From **Table 7.2**, we can see, most of the COF means for different k are very close, either 1.04 or 1.05, only when k=3, it is 1.07. From another point of view, **Figure 7.2** also shows the trend of the COF standard deviation for different k, with the increasing of k value from 3 to 8, the standard deviation value is decreasing of a high rate with a sharp curve, and it indicates that the standard deviation values are spread out over a large range of values. Whereas, when the k value comes to 8 and 9, it is decreasing of a very low rate and the standard deviation value trends to about 0.23 stably, and it indicates that the standard deviation values tend to be very close to the mean.

In summary, we have explained the above three major issues to balance an appropriate k. From the view of COF numbers in different intervals for different k, we can get that the COF numbers in either intervals tends to stable from k=5. From the view of the mean and standard deviation of the COF for different k, we can get that the mean and standard deviation of the COF tend to stable when the k value comes to 8 and 9. From the view of the tracked max and a common COF values for different k, we can get that the tracked max and a common COF values tend to stable when the k value comes 9. To conclude, k=9 is selected finally. Since data “Samp41” and “Samp31” are acquired from the same platform, then the data density and spatial resolution are approximately same, then we also use k=9 in data Samp31.

7.1.2 Outlier detection results in“Samp41”

The COF attribute (k=9) and the height attribute are extracted from the LiDAR point cloud data to organize a 2-D space as shown in **Figure 7.3**. In the formed 2-D space, the horizontal ordinate represents the elevation values, while the vertical ordinate represents the COF values. The proposed robust statistical methods are conducted into the data in the space. Outliers are flagged to red triangles, as shown in **Figure 7.4**. **Figure 7.5** shows the 3-D view of the result, from the figure we can see, outliers both in the single and cluster form are detected.

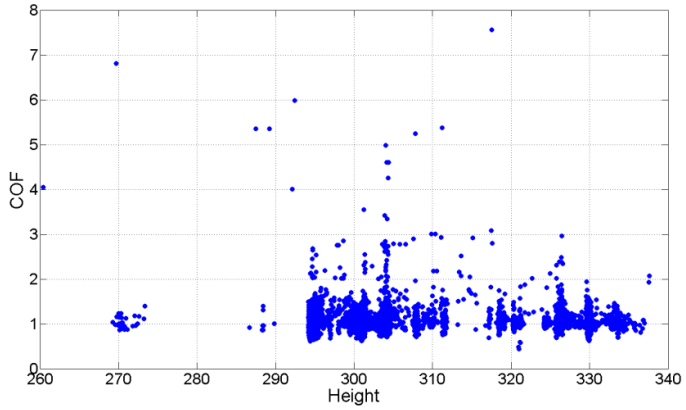


Figure7.3 Formed 2-D space based on the height and COF attributes. The horizontal ordinate represents the elevation values, while, the vertical ordinate represents the COF values

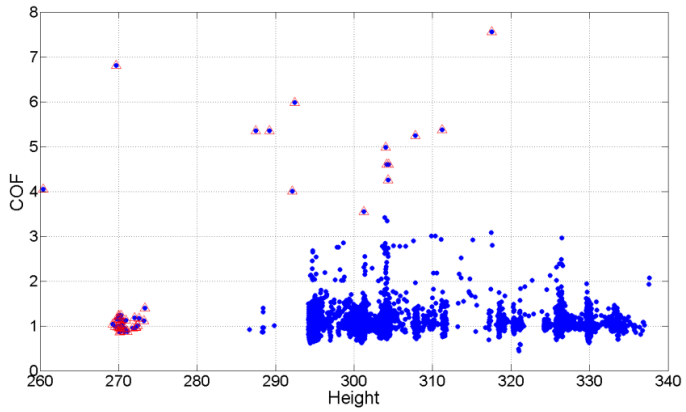


Figure 7.4 Illustration of the outlier detection result in the formed 2-D space, outliers are marked in red triangles.

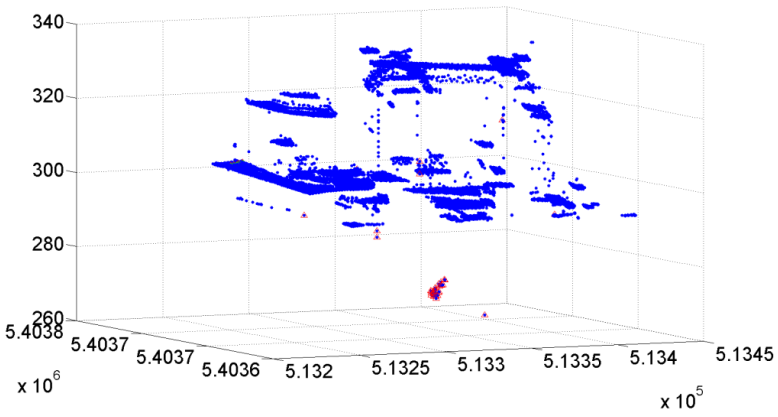


Figure 7.5 Illustration of 3-D view of the outlier detection result, outliers are marked in red triangles.

Since the Mahalanobis approach considers both the average value and its variance and covariance of the attributes measured, it accounts for ranges of variance between attributes and compensates for interactions (covariance) between attributes. Here, we use two figures to address this issue: **Figure 7.6** shows the illustration of identifying outliers with the height attributes individually by block the COF attributes, from the figure we can see that it is not surprising that points with very low height value are identified, however, points with not very low elevation value are also identified, they are must with high COF value, that is the interactions of the height attributes and COF attributes by using the Mahalanobis approach. Similar situation occurs in **Figure 7.7** which shows the illustration of identifying outliers with the COF attributes individually by block the height attributes. From the figure we can see that it is not surprising that points with very high COF value are identified, however, points with not very high COF value are also identified, they are must with low elevation value, that is also the

interactions of the height attributes and COF attributes by using the Mahalanobis approach.

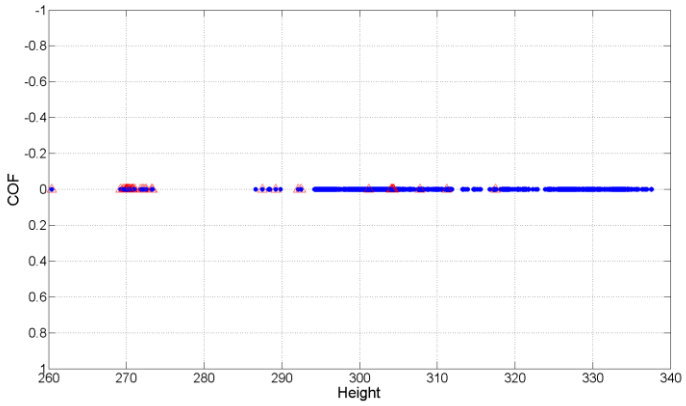


Figure 7.6 Illustration of identifying outliers with the height attributes individually by block the COF attributes

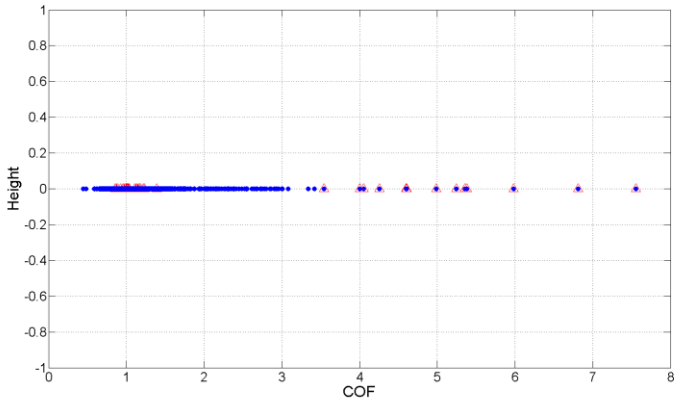
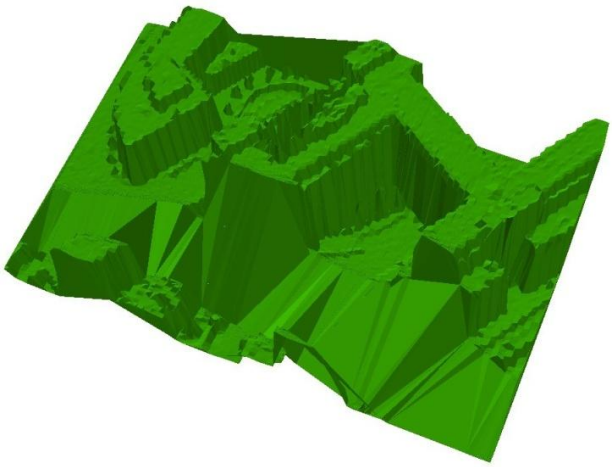
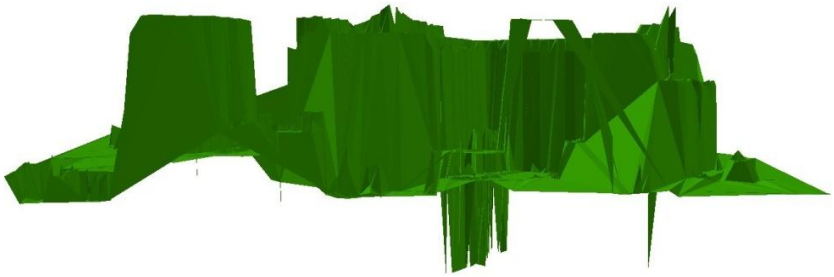


Figure 7.7 Illustration of identifying outliers with the COF attributes individually by block the height attributes

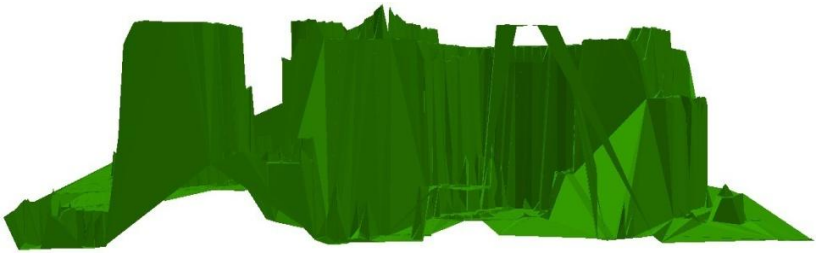
Then, based on the detection results, we use the triangulated irregular network (TIN) model to create DSMs, in which, **Figure 7.8 (a)** illustrates a 3-D TIN view of original Samp41 data, and **Figure 7.8 (b)** shows a 3-D TIN view of original Samp41 data with outliers, **Figure 7.8 (c)** indicates the TIN view after outlier removal.



(a)



(b)



(c)

Figure 7.8 Outlier detection: (a) a 3-D TIN view of original Samp41 data, (b) a 3-D TIN view of original Samp41 data with outliers, (c) the TIN view after outlier removal

7.1.3 Outlier detection results in "Samp31"

For data "Samp31", The COF attribute (k=9) and the height attribute are also extracted from the LiDAR point cloud data to organize a 2-D space as shown in **Figure 7.9**. In the formed 2-D space, the horizontal ordinate represents the elevation values, while the vertical ordinate represents the COF values. The proposed robust statistical methods are conducted into the data in the space. Outliers are flagged to red triangles, as shown in **Figure 7.10**. **Figure 7.11** shows the 3-D view of the result, from the figure we can see, the single outliers are detected.

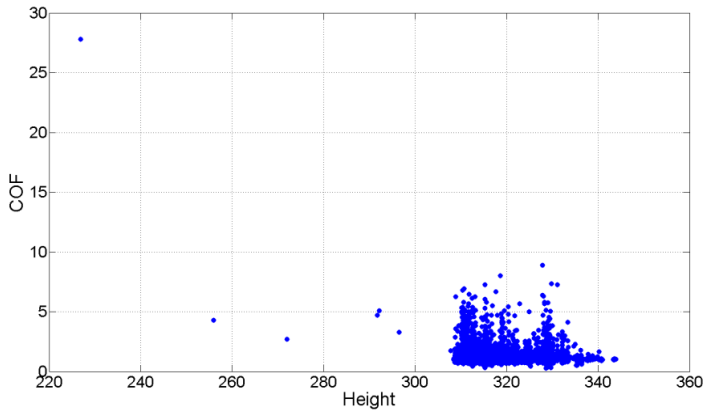


Figure 7.9 Formed 2-D space based on the height and COF attributes. The horizontal ordinate represents the elevation values, while, the vertical ordinate represents the COF values

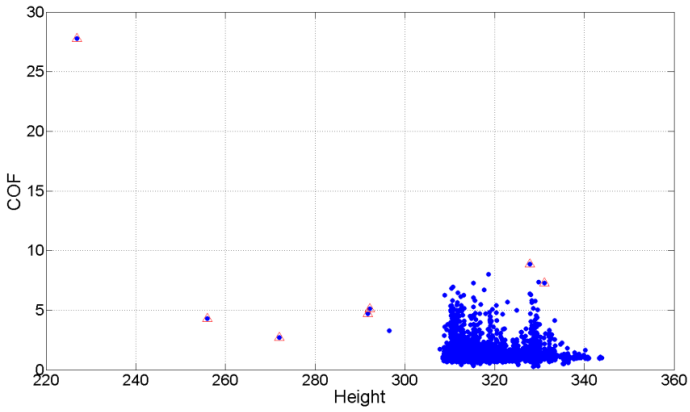


Figure 7.10 Illustration of the outlier detection result in the formed 2-D space, outliers are marked in red triangles.

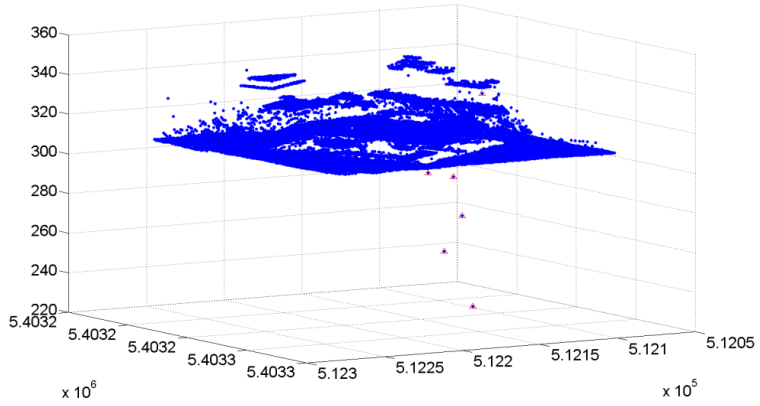
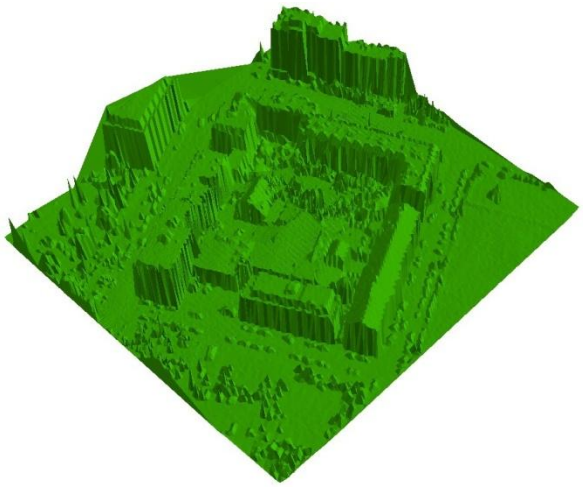
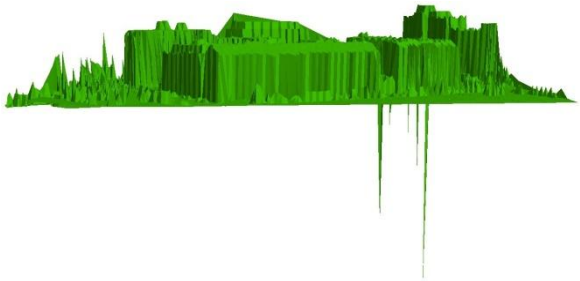


Figure 7.11 Illustration of 3-D view of the outlier detection result, outliers are marked in red triangles

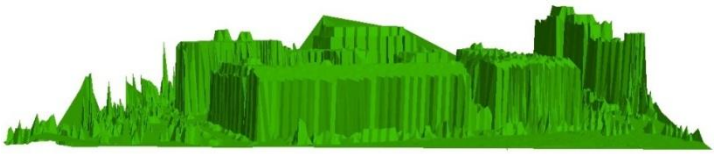
Then, based on the detection results, we use the triangulated irregular network (TIN) model to create DSMs, in which, **Figure 7.12 (a)** illustrates a 3-D TIN view of original Samp31 data, and **Figure 7.12 (b)** shows a 3-D TIN view of original Samp31 data with outliers, **Figure 7.12 (c)** indicates the TIN view after outlier removal.



(a)



(b)



(c)

Figure 7.12 Outlier detection: (a) a 3-D TIN view of original Samp31 data, (b) a 3-D TIN view of original Samp31 data with outliers, (c) the TIN view after outlier removal

7.1.4 Discussions

For outlier detection, as described by Sithole and Vosselman (2003) in the report of "ISPRS test on extracting DEMs from point clouds: A comparison of existing automatic filters", although the number of outliers (both high and low) are relatively small, even for a single outlier, the influence on filtering in its neighborhoods can be considerable, and experiments on certain filters such as (Hubert and Debruyne, 2009; Axelsson, 1999&2000) and Sithole and Vosselman (2004) have proven this issue. In their experiments, it shows that most filters can detect single outlier easily, because they are so far elevated above neighboring points. However, for many low outliers (outliers in the form of both single and cluster), it may cause problems for many filters such as (Brovelli et al., 2002; Hubert and Debruyne, 2009; Axelsson, 1999&2000; Sithole and

Vosselman, 2004). **Table 7.3** illustrates the outlier detection result comparison by using the COF, Height, the proposed COF+Height and other eight representative algorithms in data Samp41 and Samp31 respectively. And the result shows that the proposed method can highly detect most outliers in both data sets.

Table 7.3 Outlier detection result comparison

Methods / % of outlier detected	COF	Height	COF+ Height	Others(eight representative algorithms)
Samp41 (Mainly for low cluster outliers)	28.9%	61.54%	88.46%	Most are Failed Sithole and Vosselman (2003)
Samp31 (Mainly for individual outliers)	50%	62.5%	87.5%	Fair-Good (>50%) Sithole and Vosselman (2003)

7.2 Experimental Results and discussion for data filtering issue

The proposed multiple attributes based MCD filter has been applied to the nine reference urban sites (includes data Samp31) offered by ISPRS. Both the Qualitative and quantitative assessment are applied to evaluate the performance of the proposed filtering method.

7.2.1 Experimental Results

We follow the designed framework (see **Figure 6.10**) to do filtering by conducting the nine experimental data sets, then we get the overall final results which are listed in

Table 7.4. In the table, we record the different cases "**Case A**" or "**Case B**" as well as the accuracy assessment for each data set.

Table7.4 Overall final results of the data filtering

Patch			Case A		Case B		Accuracy Assessment		
Region	Count	Grid	General Sense	Complex Sense	General Sense	Complex Sense	Type I Error	Type II Error	Total Error
Samp11	38010	20*5	30	29	25	16	33.28%	3.34%	20.48%
Samp12	52119	20*6	61	18	23	8	20.02%	0.86%	10.84%
Samp21	12960	6*5	22	7	1	0	8.30%	0.56%	6.58%
Samp22	32706	9*9	40	19	17	5	27.98%	1.45%	19.76%
Samp23	25095	12*5	26	9	18	7	39.97%	1.81%	21.92%
Samp24	7492	4*4	9	3	3	1	25.24%	2.77%	19.07%
Samp31	28862	8*8	34	13	12	5	5.15%	1.03%	3.31%
Samp41	11231	5*5	16	4	3	2	23.68%	0.84%	12.78%
Samp42	42470	10*10	30	11	49	10	11.75%	0.23%	3.58%

To illustrate the detailed process steps both for Case A and Case B, we select two typical patches for the two cases to record their process stages. For Case A, the selected patch contains 600 original points which need to be separated into terrain points and off-terrain points, we just follow the filtering process described in Section6.2.2, based on the statistical information, we can obtain that Hmean-Hmedian>0, and Hstd=2.2115, then we use the T3 filter (see **Table 6.1**) to pick out the off-terrain points, and remain terrain points, after such operation, we can get the filtering result, in which, the remained points is total 90, and Hmax-Hmin<1, then based on the limitations introduced

in Section6.2.2, we ends the process. Accuracy assessment is conducted finally, corresponding figures for each step are also provided. The overall process steps is illustrated in **Figure 7.13.**

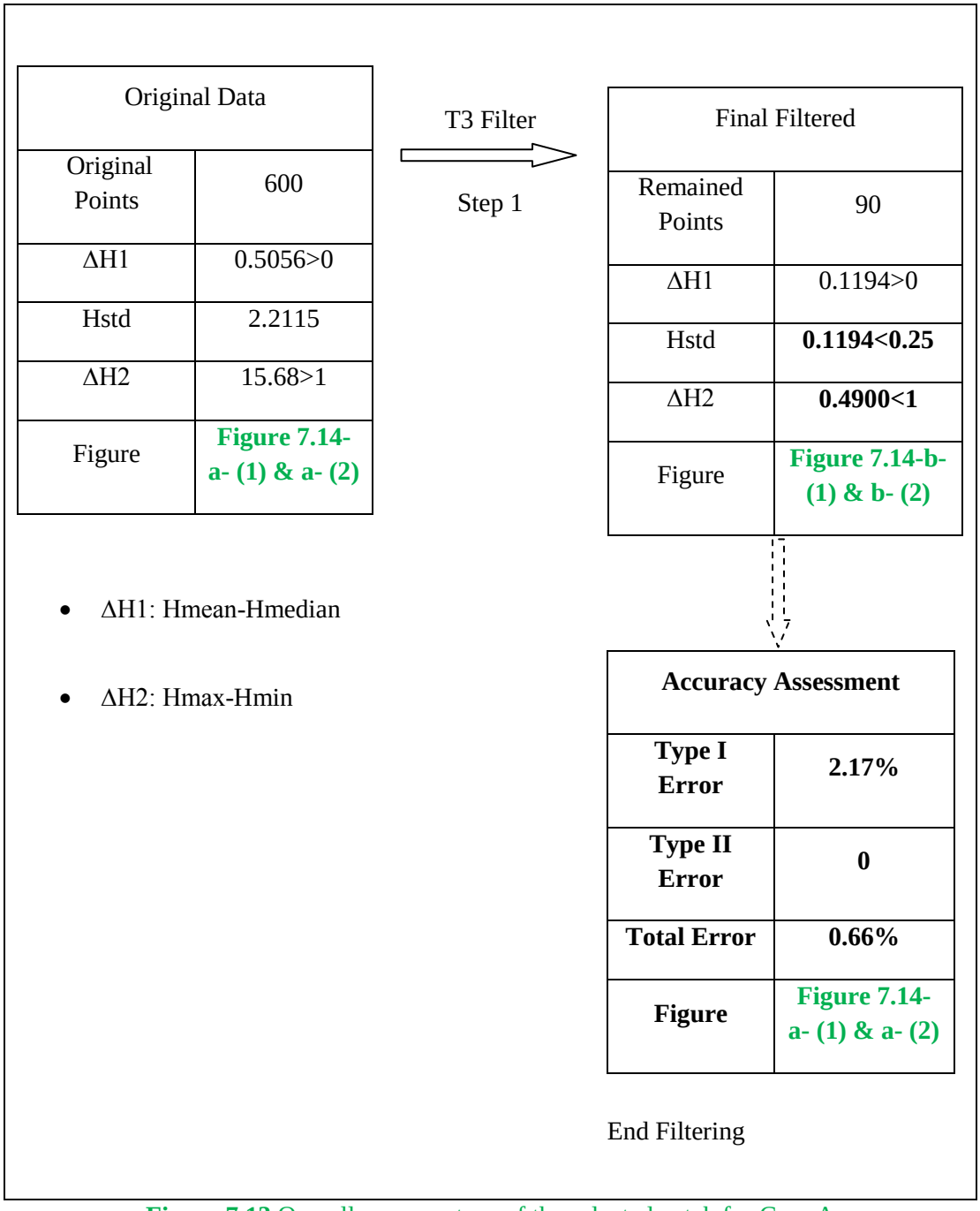
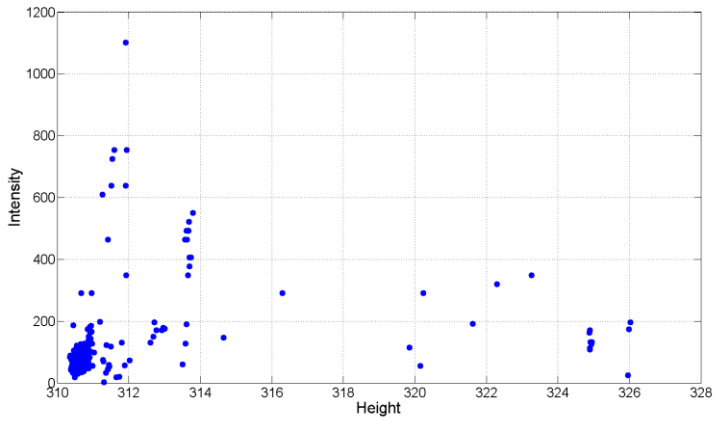
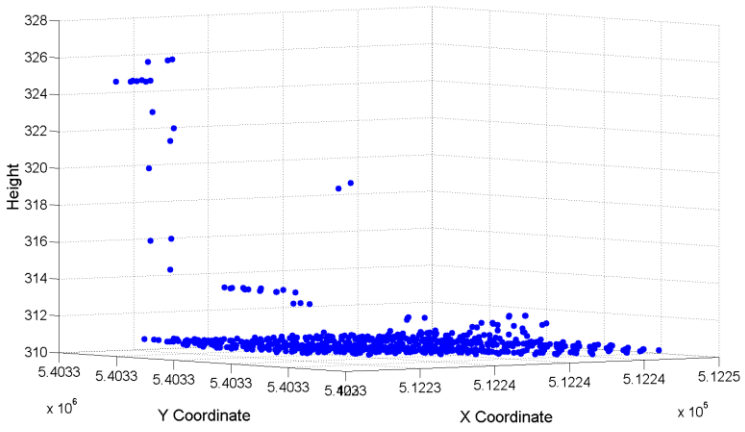


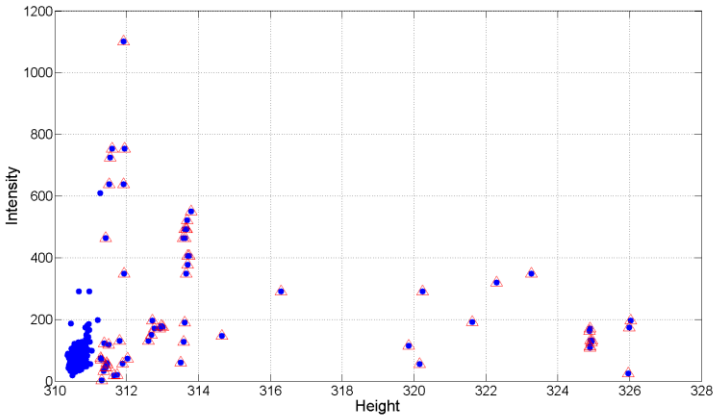
Figure 7.13 Overall process steps of the selected patch for Case A



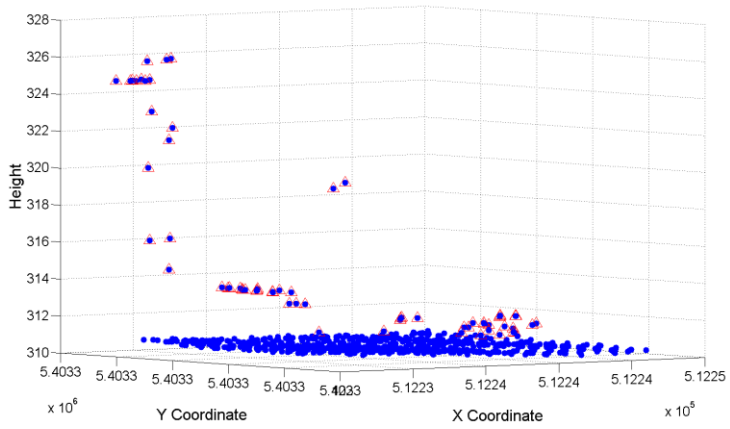
a-(1) Original data in 2-D view



a-(2) Original data in 3-D view



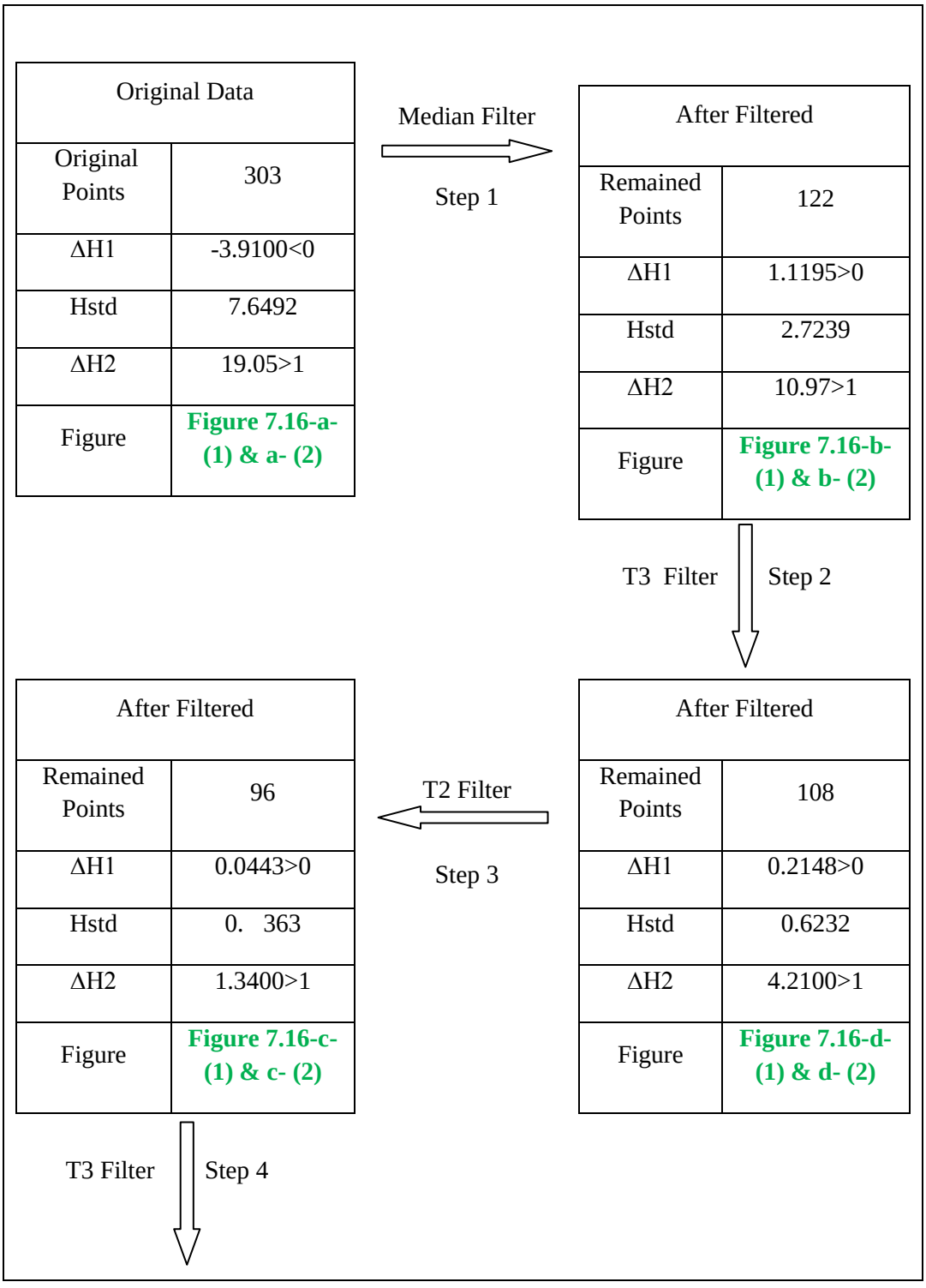
b-(1) Final filtered data in 2-D view



b-(2) Final filtered data in 3-D view

Figure 7.14 Corresponding figures for each step in the process of selected patch for Case A: (1) for 2-D views, and (2) for 3-D views, filtered off-terrain points are marked in red triangles.

For Case B, the selected patch contains 303 original points which need to be separated into terrain points and off-terrain points, we just follow the filtering process described in Section6.2.2, based on the statistical information, we can obtain that Hmean-Hmedian<0, then we use the Median filter (see **Table 6.1**) to pick out parts of the off-terrain points, and remain terrain points, after such operation, we can get the filtering result, in which, the remained points is total 122, and Hmean-Hmedian>0, then certain iterations are conducted by using the related filters introduced in Section6.2.2, we ends the process when its statistical information fits the proposed limitations: Hmax-Hmin<1 and Hstd<0.25. Accuracy assessment is conducted finally, corresponding figures for each step are also provided. The over all process steps is illustrated in **Figure 7.15**.


Original Data
Original Points
303
ΔH1
-3.9100<0
Hstd
7.6492
ΔH2
19.05>1
Figure
Figure 7.16-a- (1) & a- (2)
Median Filter
Step 1
After Filtered
Remained Points
122
ΔH1
1.1195>0
Hstd
2.7239
ΔH2
10.97>1
Figure
Figure 7.16-b- (1) & b- (2)
T3 Filter
Step 2
After Filtered
Remained Points
108
ΔH1
0.2148>0
Hstd
0.6232
ΔH2
4.2100>1
Figure
Figure 7.16-d- (1) & d- (2)
T2 Filter
Step 3
After Filtered
Remained Points
96
ΔH1
0.0443>0
Hstd
0. 363
ΔH2
1.3400>1
Figure
Figure 7.16-c- (1) & c- (2)
T3 Filter
Step 4

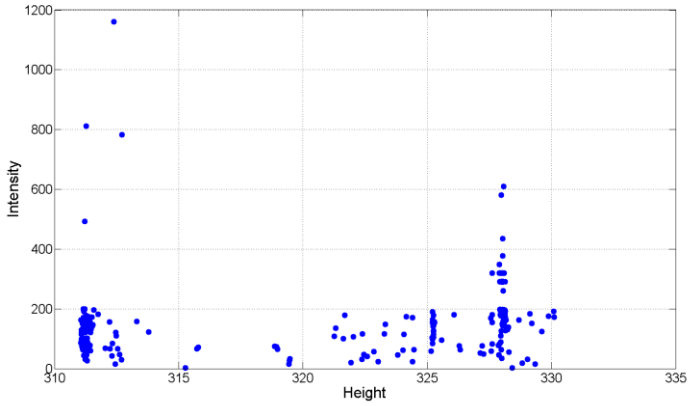
Final Filtered	
Remained Points	90
ΔH1	0.1194>0
Hstd	**0.1194<0.25**
ΔH2	**0.4900<1**
Figure	**Figure 7.16-e-(1) & e- (2)**

End Filtering

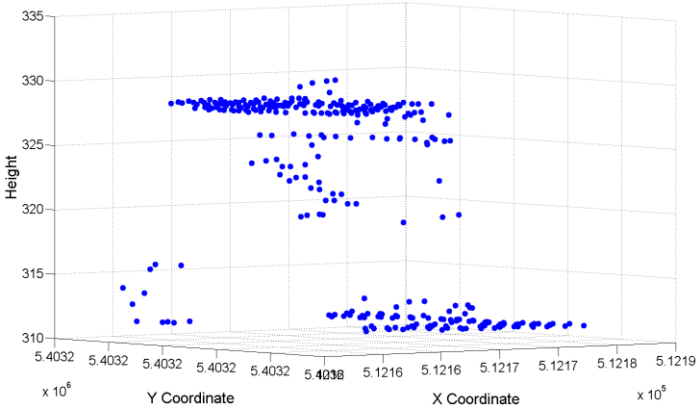
Accuracy Assessment	
Type I Error	**2.17%**
Type II Error	**0**
Total Error	**0.66%**
Figure	**Figure 7.16-f- (1) & f- (2)**

- ΔH1: Hmean-Hmedian
- ΔH2: Hmax-Hmin

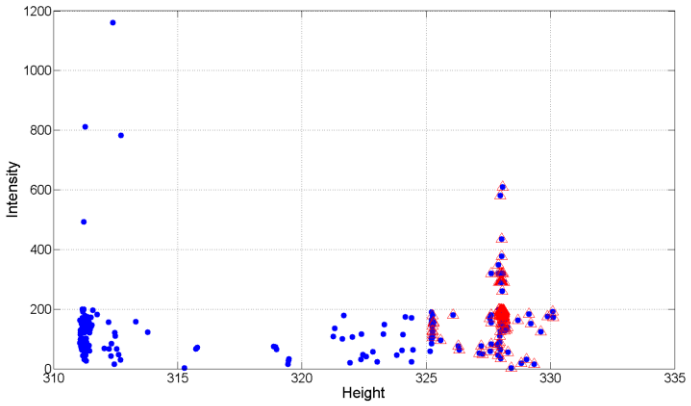
Figure 7.15 Overall process steps of the selected patch for Case B



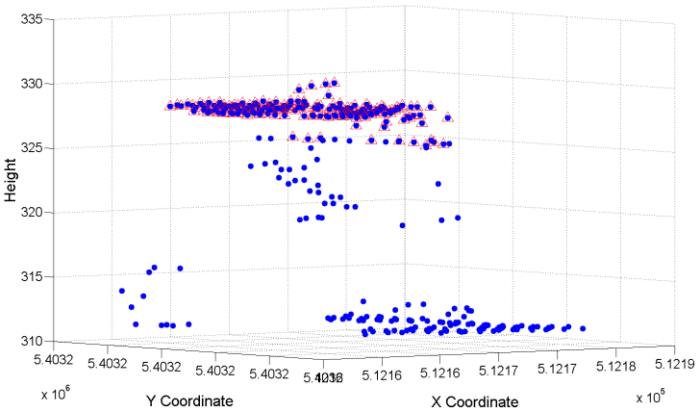
a-(1) Original data in 2-D view



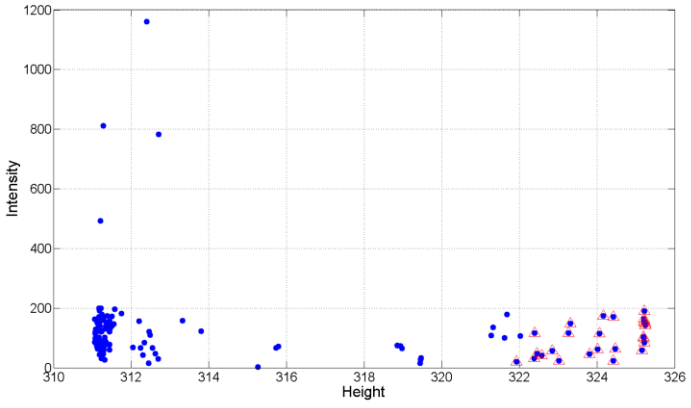
a-(2) Original data in 3-D view



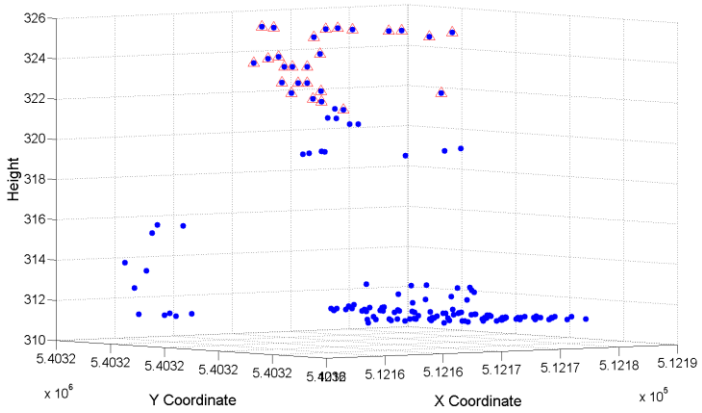
b-(1) Filtered data in 2-D view after step 1



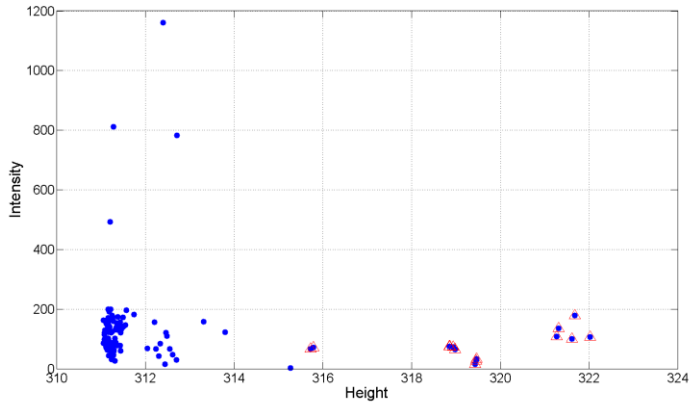
b-(2) Filtered data in 3-D view after step 2



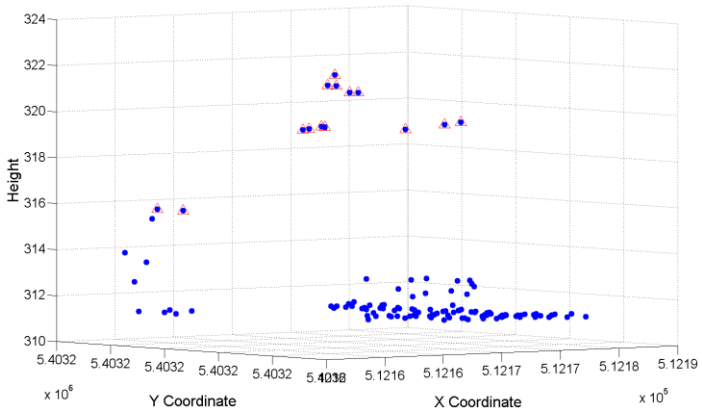
c-(1) Filtered data in 2-D view after step 2



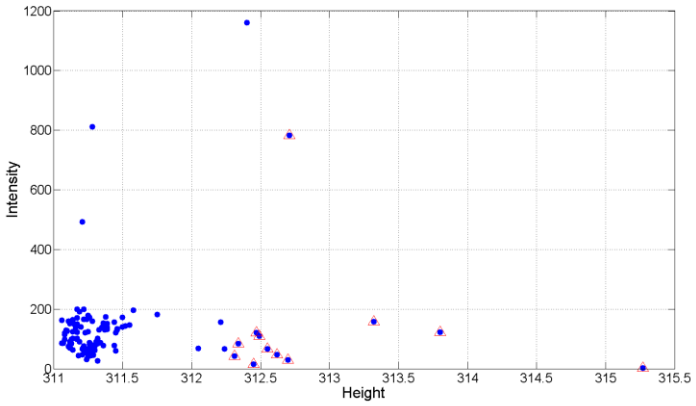
c-(2) Filtered data in 3-D view after step 2



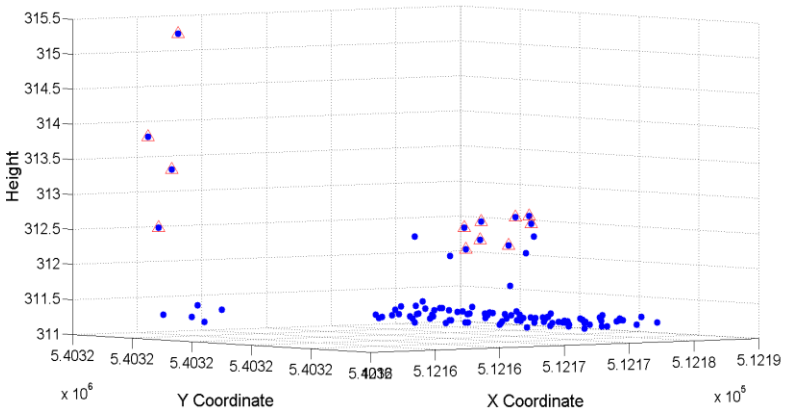
d-(1) Filtered data in 2-D view after step 3



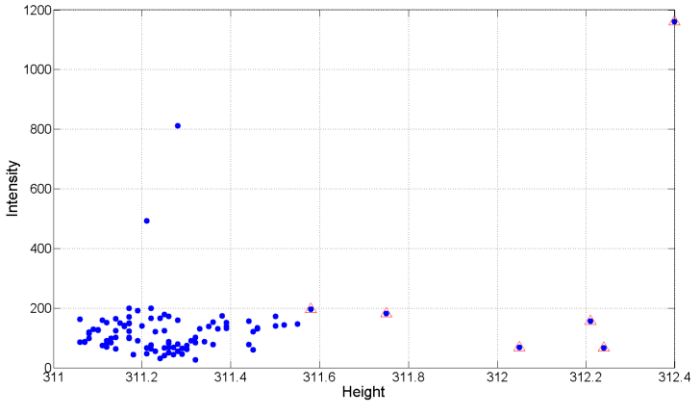
d-(2) Filtered data in 3-D view after step 3



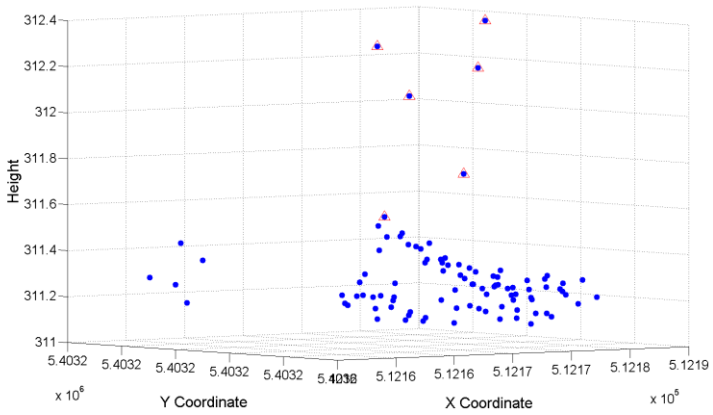
e-(1) Filtered data in 2-D view after step 4



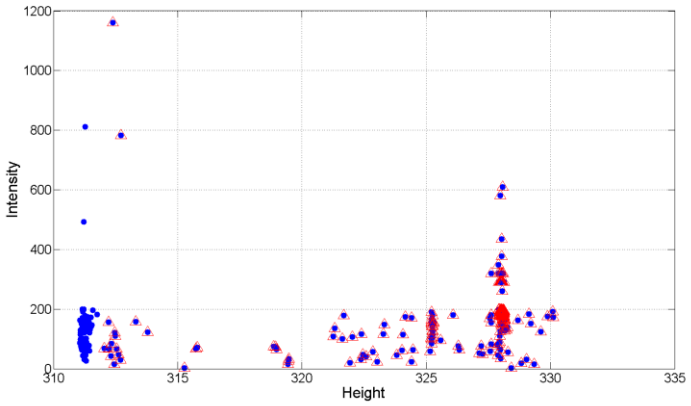
e-(2) Filtered data in 3-D view after step 4



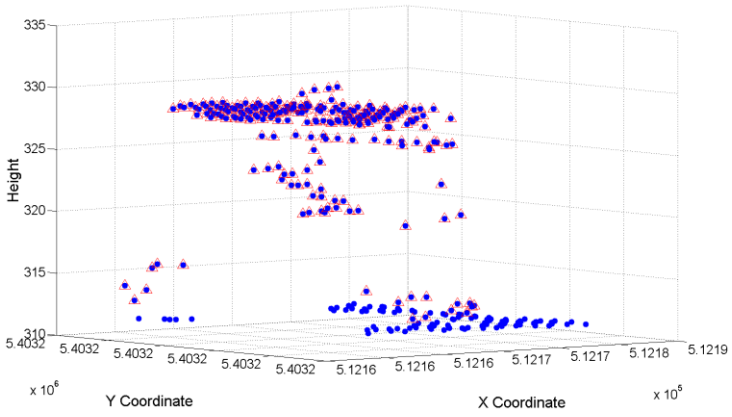
f-(1) Filtered data in 2-D view after ending step



f-(2) Filtered data in 3-D view ending step



g-(1) Final filtered data in 2-D view



g-(2) Final filtered data in 3-D view

Figure 7.16 Corresponding figures for each step in the process of selected patch for Case B: (1) for 2-D views, and (2) for 3-D views, filtered off-terrain points are marked in red triangles.

7.2.2 Qualitative assessment

Since many of the real world objects are very complex, it has proven that extreme difficult to filter such objects. For very large objects, whether they could be detected due to the local area size if the algorithm is localized. In this study, to avoid the size of such large objects exceeds the patch size (localized experimental study area) which may lead the filtering failure, we take a representative data: data Samp31 as training dataset to determine a proper patch size by using the Google Earth tools. And then the training parameters are conducted to other datasets, and have fair results. For very small objects, such as vehicles, due to the spatial resolution of the data, points belong to such small objects are relatively small, and they have sparse vertical structure. Since the proposed method is sensitive to such sense, by conducting the limitation (Hmax-Hmin<1m) it is not difficult to detect such objects. For very low objects, they are normally very close to the ground, and it is difficult to pick them out from the ground. Since the proposed method considering both the intensity information and the height information, such objects are also removed easily. For complex configuration objects, since we use the Hstd to determine proper threshold values, even when the Hstd is very large, the 'median' filter is used to have a rough removal till it becomes to general sense. Filtering such objects is tough; however, to get clean terrain points, the result is fair though it lost a number of terrain points and has a large number of the Type I error. For vegetations, based on the differences on the vertical structure and intensity value to terrain points,

they are normally easy to be removed by using the proposed method. However, when objects are on slopes such as building on slopes or discontinuity objects such as steep slopes, it may face challenges. Based on the characteristics of the filter, the result is also fair though it lost a number of terrain points and has a large number of the Type I error. In this study, it is very easy to remove high outliers, for low outliers, in the post processing, step, they are also removed.

7.2.3 Quantitative assessment and performance comparison

The proposed multiple attributes based MCD filter has been applied to the nine reference urban sites (includes data Samp31) offered by ISPRS. Quantitative assessment was also done by evaluate the Type I, Type II and Total errors for each sites. The Type I error which refers to the omission error is the rate of terrain points misclassified as off-terrain points. The type II error which refers to the commission error is the rate of off-terrain points misclassified as terrain points. The total error which refers to the balanced Type I and Type II error is the ratio of all misclassified points in total number of points. **Table7.5** shows the calculation of the three kinds of errors.

Table 7.5 Calculation of the three kinds of errors

Filtered / Reference	Terrain	Off-terrain
Terrain	a	b
Off-terrain	c	d

Then we have: Type I error: b/a+b; Type II error: c/c+d; Total error: (b+c)/(a+b+c+d). Comparative analysis with the eight other representative methods is provided. The eight

other methods are respectively proposed by Elmqvist, Sohn, Axelsson, Brovelli, Pfeifer, Brovelli, Roggero, Wack and Sithole which are detailed explained and compared by Sithole and Vosselman(2004) in their experiments.

Table 7.6 Type I error comparison with ISPRS tested filters

	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	Pang&Shi
Samp11	33.63%	26.56%	15.96%	28.26%	62.00%	33.16%	39.12%	37.69%	33.28%
Samp12	12.36%	8.87%	4.89%	7.29%	29.63%	11.92%	11.94%	19.19%	20.02%
Samp21	25.91%	8.38%	0.46%	2.81%	11.35%	12.46%	5.15%	9.64%	8.30%
Samp22	20.55%	5.68%	2.68%	8.25%	31.19%	33.43%	9.73%	29.29%	27.98%
Samp23	18.74%	7.25%	3.69%	12.08%	50.25%	41.88%	18.40%	40.92%	39.97%
Samp24	31.80%	13.17%	3.38%	8.54%	47.63%	30.43%	14.41%	32.79%	25.24%
Samp31	8.47%	4.81%	7.91%	1.60%	21.75%	3.03%	3.15%	4.85%	5.15%
Samp41	14.42%	19.25%	25.81%	19.85%	32.41%	21.55%	17.63%	47.13%	23.68%
Samp42	4.30%	1.01%	4.68%	8.02%	20.40%	13.37%	10.65%	12.18%	11.75%

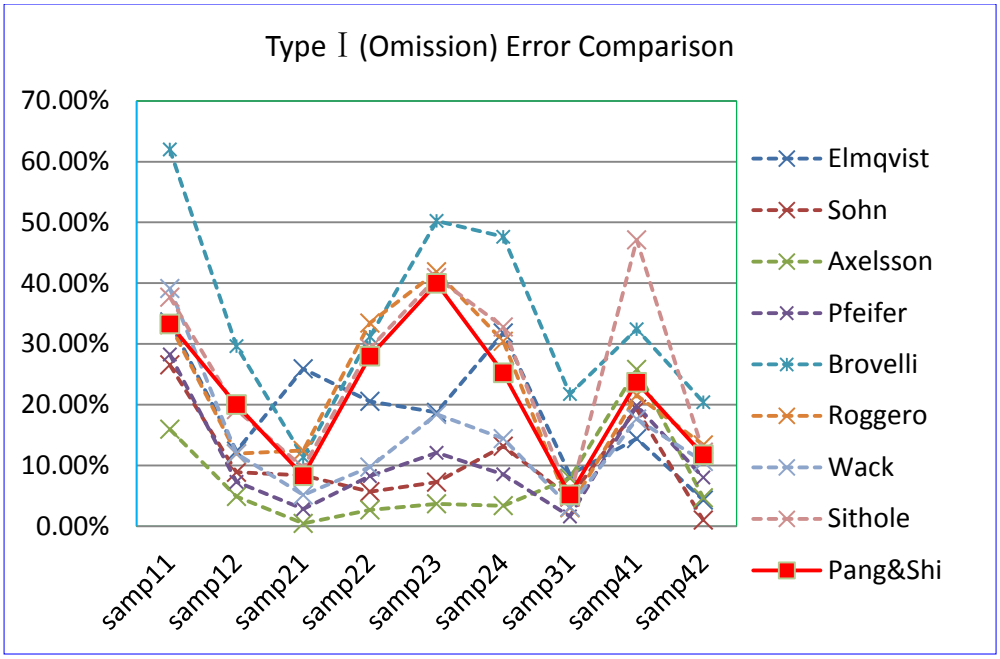


Figure 7.17 Type I error comparison with ISPRS tested filters

Table 7.7 Type II error comparison with ISPRS tested filters

	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	Pang&Shi
Samp11	4.38%	12.17%	3.65%	2.41%	2.53%	3.88%	3.38%	3.49%	3.34%
Samp12	3.30%	7.87%	1.48%	1.52%	2.04%	0.91%	0.89%	0.57%	0.86%
Samp21	1.75%	10.40%	18.53%	1.64%	1.56%	0.00%	2.31%	0.67%	0.56%
Samp22	1.91%	11.91%	5.87%	3.08%	1.79%	1.01%	2.37%	0.98%	1.45%
Samp23	3.99%	12.79%	4.34%	3.81%	2.38%	1.94%	2.58%	2.09%	1.81%
Samp24	2.98%	13.81%	7.45%	8.95%	2.87%	1.70%	3.26%	3.48%	2.77%
Samp31	2.33%	8.28%	1.03%	2.04%	2.39%	1.08%	1.09%	1.12%	1.03%
Samp41	0.85%	3.20%	1.89%	1.57%	0.46%	2.74%	0.49%	1.65%	0.84%
Samp42	1.98%	2.12%	0.26%	0.24%	0.25%	0.26%	0.39%	0.15%	0.23%

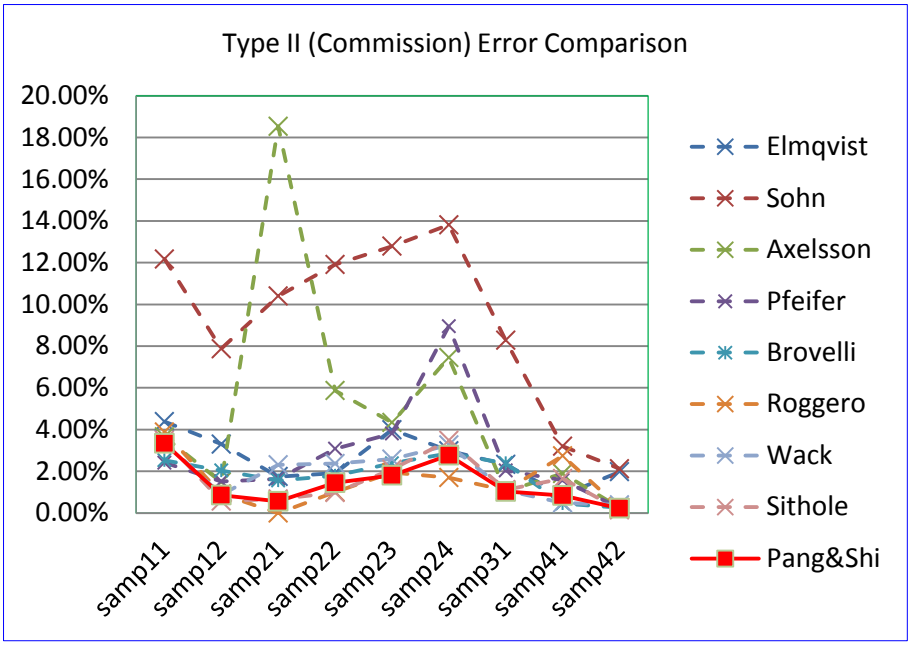


Figure7.18 Type II error comparison with ISPRS tested filters

Table 7.8 Total error comparison with ISPRS tested filters

	Elmqvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	Pang&Shi
Samp11	22.40%	20.49%	10.76%	17.35%	36.96%	20.80%	24.02%	23.25%	20.48%
Samp12	8.18%	8.39%	3.25%	4.50%	16.28%	6.61%	6.61%	10.21%	10.84%
Samp21	8.53%	8.80%	4.25%	2.57%	9.30%	9.84%	4.55%	7.76%	6.58%
Samp22	8.93%	7.54%	3.63%	6.71%	22.28%	23.78%	7.51%	20.86%	19.76%
Samp23	12.28%	9.84%	4.00%	8.22%	27.80%	23.20%	10.97%	22.71%	21.92%
Samp24	13.83%	13.33%	4.42%	8.64%	36.06%	23.25%	11.53%	25.28%	19.07%
Samp31	5.34%	6.39%	4.78%	1.80%	12.92%	2.14%	2.21%	3.15%	3.31%
Samp41	8.76%	11.27%	13.91%	10.75%	17.03%	12.21%	9.01%	23.67%	12.78%
Samp42	3.68%	1.78%	1.62%	2.64%	6.38%	4.30%	3.54%	3.85%	3.58%

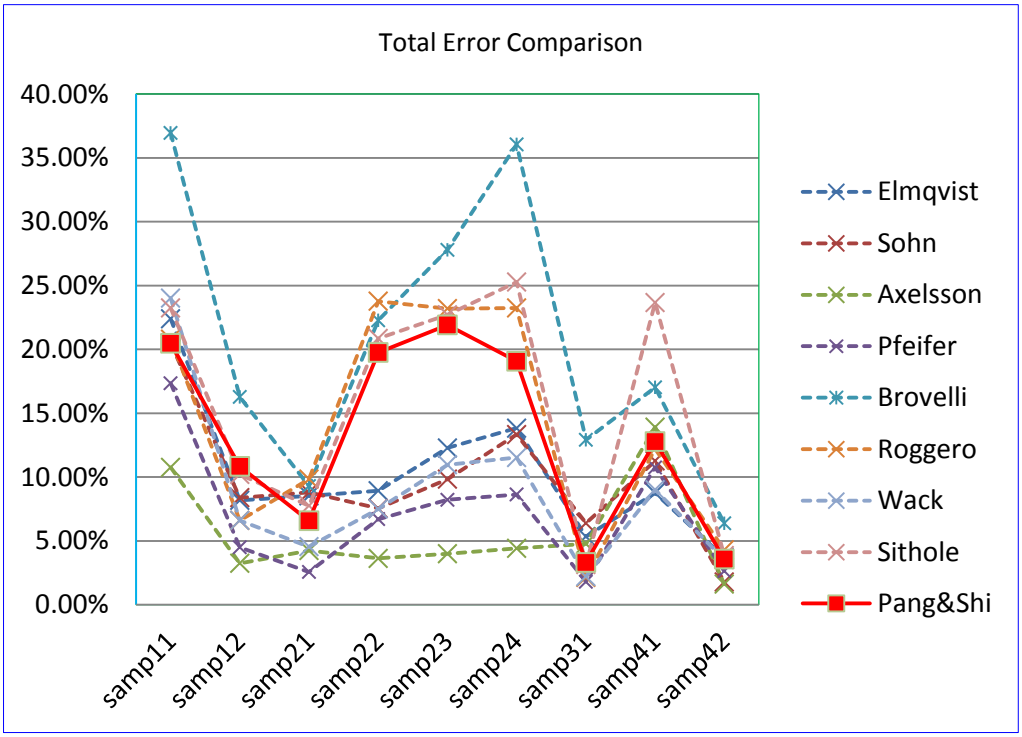


Figure7.19 Total error comparison with ISPRS tested filters

Table7.6, **Table7.7** and **Table7.8** and their related charts **Figure7.17, Figure7.18,** and **Figure7.19** illustrate the performances of the proposed method by comparing with the Type I error, Type II error and Total error with other eight representative methods. Since the major objective by using the filtered terrain points in this study is to generate DEMs,

we are we are focusing on minimize the Type II errors, the more clean the filtered terrain points, the more accuracy the DEM. Considering this situation, we are focusing on minimize the Type II errors and also balance the Total error at the meanwhile. In the data processing steps, the "median" filter are used to roughly remove points when the MCD filter meets the CaseB or Hstd>7 in CaseA (see Section6.3.2), or when it suffers complex senses (see Section6.3.3). This rough operation probably may remove certain terrain points when roughly remove off-terrain points, which may lead a higher Type I error. Besides, the limitation for ending the data processing: only when $0 \leq Hstd \leq 0.25$ and Hmax-Hmin<1m (see Section6.3.2), this limitation assumes the ground is relatively flat, and the rise and fall in vertical is less than 1m which is fair for Type II error but seems crude to some extents for Type I error. To balance both the Type I error and the Type II error, in the post processing step, the filtered off-terrain points are reclassified which may reduce the Type I error. Results show that Type II error in our method ranks at about top 3 of every sample region with others, and simultaneity, Type I error and Total error ranks at a middle level. While, which error need we reduce more? Type I or Type II? To reduce which error, it all depends on the cost of the error for the application that will use the filtered data. Most of the tested filtering algorithms focus on minimizing Type II errors, filter parameters are chosen to remove as many object points which may cause large Type I error. In this study, we are focusing on reducing the Type II errors for two major reasons: on the one hand, DEM generation is the application by using the final filtered data, therefore, more "clean" terrain points is needed, on the other hand, by reducing Type II errors, we could have fair comparison with others. Our

method ranks as top 3 regarding type II error, and simultaneity, Type I error and Total error ranks at a middle level.

CHAPTER 8 Conclusions and Future Works

8.1 Conclusions

In this study, we have comprehensively illustrated both the outlier detection and data filtering issues in LiDAR point clouds data for urban areas. Challenges and limitations of current methods in both two issues are presented. Specifically, for outlier detection, since such application is considered as an essential preprocessing step for overall LiDAR data filtering and modeling, and has been frequently discussed in the LiDAR-drived DEM quality control and accuracy assessment, process of automatic classification, building extraction (3-D reconstruction) and city modeling of raw LiDAR data, many researchers have developed various methods to remove outliers. These methods can be summarized into two major categories: (1) Analysis of the elevation deviation; (2) Analysis of the spatial neighborhood relationship. Literature review indicates that most of the current schemes in both the two categories could only identify individual outliers, while, potentially misclassify normal objects as outliers by analyzing single attribute: elevation or spatial neighborhood relationship (such as "LOF"). Thus, to accurate detect all outliers, the mentioned multiple attributes need to be considered. While, for data filtering, since it is regarded as an essential step for DEM generation, various methods also have been developed. Literature review demonstrates that most of the existing methods are mainly based on the analysis of geometrical information of LiDAR points, while, radiometric information such as intensity data is seldom used. Since the geometrical information and the radiometric information are simultaneously

generated on the same platform, both the two data describe the same features geometrically, although it has challenges to calibrate the raw intensity data which always has speckle noise, the comprehensive utilization of both the height and intensity data simultaneously provided by LiDAR may be advantageous over using either data individually (Wang and Glenn, 2009). Similar suggestions can be found in Mallet and Bretar, (2009) and Vosserman's (2010) works.

Considering the situations explained above, to fit the requirements of multiples attributes data processing both for outlier detection and data filtering, in this study, the MCD-based multiple attributes model is introduced, which extends traditional data processing methods from single attribute to multiple attributes, from one dimension to two dimensions. Firstly, we apply the proposed model into the LiDAR data for outlier detection purpose, judging from the characteristics of outliers in LiDAR point clouds data, which can be "both single points and also clusters with elevations, either much higher or lower than the surrounding points", a spatial neighborhood relationship indicator "COF" is adopted as an attribute of LiDAR points which demonstrates the isolativity of a point to its neighbors. Then the COF attribute together with the height attribute are extracted from LiDAR points to organize a 2-D space, in the formed 2-D space, we conduct the proposed MCD-based multiple attributes model to identify outliers. To get a stable COF value, we have explained three major issues which are "COF numbers in different intervals for different k", "the mean and standard deviation of the COF for different k" and "tracked max and a common COF values for different k" to balance an appropriate k to calculate COF. Finally, k=9 is selected. Two typical experimental data are implemented into the proposed method to evaluate its

performance. Comparative results by using the COF, height, the proposed COF+Height and other eight representative algorithms in data Samp41 and Samp31 are generated and analyzed. And the result shows that the proposed method can detect most of the outliers effectively in both forms: individual and cluster. Secondly, we apply the proposed model into the LiDAR data for data filtering purpose. The intensity data and height data are extracted from LiDAR point as two significant attributes to organize a 2-D space, in the formed 2-D space, we conduct the proposed MCD-based multiple attributes model to do data filtering. Before applying the model, some preprocessing works such as "Local area determination" and "Threshold determination" are needed. Nine typical experimental data sets are implemented into the proposed method to evaluate its performance. Both quantitative and qualitative assessments of the results are carried out. By comparing with eight representative methods at the ISPRS filter test, it shows that our method is fair by minimizing the Type II error, in which, Type II error in our method ranks at about top 3 of every sample region with others, and simultaneity, Type I error and Total error ranks at a middle level.

8.2 Future works

During recent years, with the latest developments in new laser scanners, the full-waveform (FWF) airborne laser scanning system which is recognized as the new generation of airborne laser scanning system has emerged. Such system records the complete waveform of the backscattered pulse which makes it have the ability to provide not only rich information about range estimation but also advanced pulse detection. Thus, from the geometric views, the former product can lead to denser point clouds and a better range determination by storing the full waveforms, while, from the

radiometric views, the latter product can bring certain other information such as intensity and pulse width by modeling the return waveforms (Mallet and Bretar, 2009). The FWF data has been widely used in woodlands for forest analysis (Mallet and Bretar, 2009; Gross et al., 2007) and also used for archaeological reconnaissance (Doneus and Briese, 2006; Doneus et al., 2008). Furthermore, some researchers have also conducted the data which includes the intensity and pulse information to analysis its potential for the improvement of DTM generation, ground classification and characteristic line extraction in urban areas (Mallet et al., 2008; M ̈ucke, 2008; Jutzi and Stilla, 2005). Although the current studies indicate that it has less potentialities of using the FWF data in urban areas than in forest areas because of the penetration issues for buildings or man-made features, it is still believed that it could be further used by analyzing multiple additional features and spatial neighborhood relationships to improve certain applications mentioned before such as DTM generation and ground classification (Mallet and Bretar, 2009).

In this study, the MCD-based multiple attributes model both for outlier detection and data filtering in raw airborne LiDAR data is introduced, such model provides a platform to process multiple attributes data, and even for further applications, the attributes may be even more: (1) Firstly, we have discussed the outlier detection issue with two attributes: COF attribute and height attribute, however, both of the two attributes are acquired from the geometric information, in future, radiometric information such intensity and width of backscattered echo provided by FWF also could be used as other attributes to improve the outlier detection issue by using the proposed multiple attributes model; (2) Secondly, we have also discussed the data filtering issue with two attributes:

intensity attribute and height attribute, in future, even more radiometric information such as width of backscattered echo provided by FWF also could be used as other attribute to improve the filtering issue by using the proposed multiple attributes model; (3) Lastly, the proposed model is also considerable for other applications such as ground classification and building extraction when dealing with multiple attributes.

Reference

Aguilar, F.J. and Mills, J., 2008, Accuracy assessment of LIDAR-derived digital elevation models. *The Photogrammetric Record*, 23(122): 148-169.

Agyemang, M. and Ezeife, C.I., 2004, LSCMine: Algorithm for Mining Local Outliers. *15th Information Resources Management Association (IRMA) International Conference*, May 23-26, New Orleans, Louisiana, USA.

Akca, D., Gruen, A., Freeman, M. and Sargent I., 2009, Fast quality control of 3-D city models. *The Int.LIDAR Mapping Forum (ILMF'09)*, January 26-28, New Orleans, Louisiana, US, (only on CD-ROM).

Alexander, C., Smith-Voysey, S., Jarvis, C. and Tansey, K., 2009, Integrating building footprints and LiDAR elevation data to classify roof structures and visualize buildings. *Computers, Environment and Urban Systems*, 33(4): 285-292.

Almeida, J.A.S., Barbosa, L.M.S., Pais, A., and Formosinho,S.J., 2007, Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering. *Chemometrics and Intelligent Laboratory Systems*, 87(2): 208-217.

Antonarakis, A.S., Richards, K.S., and Brasington, J., 2008, Object-based land cover classification using airborne LiDAR. *Remote Sensing of Environment*, 112(6): 2988-2998.

Arefi, H., Engels, J., Hahn, M. and Mayer, H., 2007, Automatic DTM generation from laser-scanning data in residential hilly area. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(4/W45).

Axelsson P., 1999, Processing of laser scanner data-algorithms and applications. *ISPRS Journal of Photogrammetry & Remote Sensing* 54: 138 -147.

Axelsson P., 2000, DEM Generation from Laser Scanner Data Using Adaptive TIN Models. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 33(B4/1): 110-117.

Bamett, V. and Lewis, T., 1994, Outliers in Statistical Data. New York, NY, John Wiley and Sons.

Bao, Y.F., Li, G.P., Cao, C.X., Li, X.W., Zhang,H., He, Q.S., Bai, L.Y. and Chang, C.Y., 2008, Classification of LIDAR point cloud and generation of DTM from LIDAR height and intensity data in forested area. In *ISPRS Congress*, Beijing.

Barnett, V. and Lewis, T., 1994, Outliers in Statistical Data. *John Wiley and Sons, Inc*., Hoboken, New Jersey.

Beasy, C., Hopkinson, C. and Webster, T., 2005, Classification of nearshore materials on the Bey of Fundy coast using LiDAR intensity data. *Proceedings of the Canadian Symposium for Remote Sensing*, June, Wolfville.

Brennan, R., and Webster, T.L., 2006, Object-oriented land cover classification of lidar-derived surfaces. *Canadian Journal of Remote Sensing*, 32(2): 162-172.

Bretar, F., Pierrot D. M. and Roux, M., 2003, Estimating intrinsic accuracy of airborne laser data with local 3D-offsets. *International Archives of Photogrammetry and Remote Sensing*, 34: 20-26.

Bretar, F., Pierrot-Deseilligny, M. and Vosselman, G. (Eds), 2009, Laser scanning 2009, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Paris, France, 35(3/W8).

Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J., 2000, Lof: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, May 14–19, 2000, Dallas, Texas, United States, pp. 93-104.

Brovelli, M.A., Cannata, M. and Longoni U.M., 2002, "Managing and processing LIDAR data within GRASS". In *Proceedings of the GRASS Users Conference*, Trento, September11-13 Trento, Italy.

Charaniya, A. P., 2004, 3D Urban Reconstruction from Aerial LiDAR data. Computer Science University of California, Santa Cruz. http://www.soe.ucsc.edu/~amin/research/proposal.pdf (accessed 25 MAR. 2010).

Chehata, N., David, N. and Bretar, Frédéric., 2008, LIDAR Data Classification using Hierarchical K-means clustering. *ISPRS Congress Beijing*, 37:325-330.

Chen, D.C., Lu, C.T., Kou, Y.F. and Chen, F., 2008, On Detecting Spatial Outliers, *Geoinformatica*, 12(4): 455-475

Chen, Q., 2009, Improvement of the Edge-based Morphological (EM) method for lidar data filtering. *International Journal of Remote Sensing* 30 (4): 1069-1074.

Chen, Q., Gong, P., Baldocchi, D.D. and Xie, G., 2007, Filtering airborne laser scanning data with morphological methods. *Photogrammetric Engineering & Remote Sensing* 73(2): 175-185.

Chen, Y., Su, W., Li, J. and Sun, Z., 2009, Hierarchical object oriented classification using very high resolution imagery and LIDAR data over urban areas. *Advances in Space Research*, 43(7): 1101-1110.

Chen, Z., Fu, A. and Tang, J. 2003, On complementarity of cluster and outlier detection schemes. *Proceedings of the 5th international conference on data warehousing and knowledge discovery*, Prague, Czech Republic, pp 234–243.

Doneus, M., and Briese, C., 2006, Full-waveform airborne laer scanning as a tool for archeological reconnaissance. *Proceedings of the* 2nd *international conference on remote sensing in archeology*, *BAR international series*, pp.99-105

Doneus, M., Briese, C., Fera, M. and Janner, M., 2008, Archaelogical prospection of forested areas using full-waveform airborne laser scanning. *Journal of Archaeological Science*, 35(4): 882-893.

Duda, R.O., Hart, P. E. and Stork, D.G., 2001, Pattern classification. New York: Wiley.

Dudoit, S. and Fridlyand, J., 2002, A Prediction-Based Resampling method for Estimating the number of clusters in a dataset. *Genome Biology*, 3(7): 1-21.

Eisenbeiss, H., 2009, UAV photogrammetry. DISS. ETH NO. 18515, Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland, Mitteilungen Nr.105, pp. 235.

Elmqvist, M., 2001, Ground estimation of laser radar data using active shape models, *Proceedings of the OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Elevation Models*, March 1-3, Stockholm, Sweden.

Elmqvist, M., Jungert, E., Lantz, F., Persson A. and Söderman, U., 2001, Terrain modeling and analysis using laser scanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34 (3/W4): 211-218.

Ester, M., Kriegel, H.P., Sander, J. and X, Xu., 1996, A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the second international conference on knowledge discovery and data mining*, pp. 226-231.

Forlani, G., Nardinocchi, C., Scaioni, M. and Zingaretti P., 2006, Complete classification of raw lidar data and 3d reconstruction of buildings. *Pattern Analysis & Applications*, 8(4): 357-374.

Friedman J.H. and Tukey, J.W., 1974, A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computers*, 23(9): 881-890.

Goodale, R., Hopkinson, C., Colville, D., and Amirault- Langlais, D., 2007, Mapping piping plover (Charadrius melodus melodus) habitat in coastal areas using airborne lidar data. *Canadian Journal of Remote Sensing*, 33(6): 519-533.

Gross, H., Jutzi, B. and Thoenessen, U., 2007. Segmentation of Tree Regions using Data of a Full-Waveform Laser. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W49A): 57-62.

Guha, S., Rastogi, R. and Shim, K., 1998, An efficient clustering algorithm for large databases. *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 73-84.

Haugerud, R.A., and Harding, D.J., 2001, Some algorithms for virtual deforestation (VDF) of LIDAR topographic survey data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 34 (3/W4): 211-218.

Hawkins, D.M., 1980, Identification of Outliers. Chapman and Hall, London.

Hodge, V. and Austin, J., 2004, A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22(1): 85-126.

Höhle, J.and Höhle, M., 2009, Accuracy assessment of Digital Elevation Models by means of robust statistical methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(4): 398-406.

Hubert, M. and Debruyne, M., 2009, Minimum Covariance Determinant. *Wiley Interdisciplinary Reviews: Computational Statistics* 2(1): 36-61.

Jain, A.K. and Dubes, R.C., 1988, Algorithms for clustering data. Prentice-Hall, Englewood Cliffs.

Jain, A.K., Murty, M.N. and Flynn, P.J., 1999, Data clustering: A review. *ACM Computing Surveys*, 31(3): 264-323.

Jin, W., Tung, A.K.H. and Han. J., 2001, Mining top-n local outliers in large databases. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. August 26–29, 2001, San Francisco, California, United States, 293-298.

Johnson, T., Kwok, I. and Ng, R., 1998, Fast computation of 2-dimensional depth contours. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 224-228.

Jutzi, B. and Stilla, U., 2005, Waveform processing of laser pulses for reconstruction of surfaces in urban areas. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 (8/W27) (on CD-ROM).

Kaufman, L. and Rousseeuw, P., 1990, Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons.

Kilian, J., Haala, N. and Englich, M., 1996, Capture and evaluation of airborne laser scanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* 31(B3): 383-388.

Knorr, E.M. and Ng, R.T., 1998, Algorithms for Mining Distance-Based Outliers in Large Dataset. *Proceedings of the 24th VLDB Conference,* New York, USA.

Knorr, E.M., Ng, R.T. and Tucakov. V., 2000, Distance-based outliers: Algorithms and applications. *VLDB Journal*, 8(3-4): 237-253.

Knorr, E.M., Ng, R.T. and Zamar, R.H., 2001, Robust space transformations for distance-based operations. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. San Francisco, California, ACM Press, 126–135.

Kobler, A., Pfeifer, N., Ogrinc, P., Todorovski, L., Oštir, K. and Džeroski, S., 2007, Repetitive interpolation: a robust algorithm for DTM generation from aerial laser scanner data in forested terrain. *Remote Sensing of Environment,* 108(1): 9-23.

Kraus, K. and Pfeifer, N., 1998, Determination of terrain models in wooded areas with airborne laser scanner data, *ISPRS Journal of Photogrammetry and Remote Sensing* 53:193-203.

Krzystek, P., 2003, Filtering of laser scanning data in forest areas using finite elements. *Proceedings of 3-D Reconstruction from Airborne Laser Scanner and InSAR Data*, unpaginated CD–ROM.

Laan, M., Pollard, K. and Bryan, J., 2003, A New Partitioning Around Medoids Algorithms. *Journal of Statistical Computation and Simulation*, 73(8): 575-584.

Lane, T. and Brodley, C. E., 1999, Temporal Sequence Learning and Data Reduction for Anomaly Detection. *ACM Transactions on Information and System Security*, 2(3): 295-331.

Lillesand T.M. and Kiefer R.W., 2000, Remote Sensing and Image Interpretation. John Wiley & Sons Inc., New York.

Liu, X., 2008, Airborne LiDAR for DEM generation: some critical issues. *Progress in Physical Geograph*, 32(1): 31-49.

Lohmann, P., Koch, A. and Schaeffer, M. 2000, Approaches to the filtering of laser scanner data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33(B3): 540-547.

Lopuhaä, H.P. and Rousseeuw, P.J., 1991, Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *Annals of* Statistics1991, 19: 229-248.

Loureiro, A., Torgo, L. and Soares, C., 2004, Outlier Detection using Clustering Methods: a Data Cleaning Application. *Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector*, Bonn, Germany.

Mallet, C. and Bretar. F., (2009). Full-waveform topographic LiDAR: State-of-the-art, *ISPRS Journal of Photogrammetry and Remote Sensing*, 64:1-16.

Mansur, M.O. and Sap, M.N.M., 2005, Outlier Detection Technique in Data Mining: A Research Perspective. In *Postgraduate Annual Research Seminar*, Brazil.

Meng, X., Wang, L., Silvan-Cardenas, J. L. and Currit, N., 2009, A multidirectional ground filtering algorithm for airborne LiDAR. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(1): 117-124.

Mohd Belal, M. and Zoubi, A., (2009). An Effective Clustering-Based Approach for Outlier Detection. *European Journal of Scientific Research*, ISSN 1450-216X, 28(2):

310-316.

Mucke, W., 2008. Analysis of full-waveform airborne laser scanning data for the improvement of DTM generation. Thesis, Institute of Photogrammetry and Remote Sensing, Technical University Vienna, Vienna.

Ng, R.T. and Han, J., 1994, Efficient and effective clustering methods for spatial data mining. *Proceedings of the 20th international conference on very large data bases*, September 12-15, Santiago de Chile, Chile, 144–155.

Papadimitriou, S., Hiroyuki, K., Gibbons, P. B. and Faloutsos, C., 2003, LOCI: Fast Outlier Detection Using the Local Correlation Integral. *Proceedings of the 19th International Conference on Data Engineering*, 315-326.

Parian, J.A. and Sargent, I., 2007, Automatic height attribute assignment for building polygons: City modeling with level of detail zero. *Proceedings of the 8th Conference on Optical 3-D Measurement Techniques*, Zurich, Switzerland, 1: 371-378.

Peng, M.H. and Shih, T.Y., 2006, Error assessment in two lidar-derived TIN datasets. *Photogrammetric Engineering and Remote Sensing*, 72(8): 933-947.

Pfeifer, N., Stadler, P. and Briese, C., 2001, Derivation of digital terrain models in the SCOP++ environment, *Proceedings of the OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Elevation Models*, March 1-3, Stockholm, Sweden.

Preparata F. P. and Shamos M. I., 1985, Computational Geometry: An Introduction. Springer.

Raber, G. T., Jensen, J. R., Hodgson, M. E., Tullis, J. A., Davis, B. A. and Berglend, J., 2007, Imact of LiDAR nominal post-spacing on DEM accuracy and flood zone delineation. *Photogrammetric Engineering and Remote Sensing*, 73(7): 793-804.

Ramaswamy, S., Rastogi, R. and Shim, K., (2000). Efficient algorithms for mining outliers from large data sets. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, May 16–18, Dallas, Texas, United States, 29, 427-438.

Roggero, M., 2001, Airborne laser scanning: Clustering in raw data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(3/W4): 227-232.

Rottensteiner F. and Briese C., 2003, Automatic generation of building models from LIDAR data and the integration of aerial images. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Dresden, Germany, 34(3/W13):174-180.

Rousseeuw, P. and Leroy, A., 1987, Robust Regression and Outlier Detection. John Wiley and Sons, Inc., Hoboken, New Jersey.

Rousseeuw, P.J., (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388): 871-880.

Rousseeuw P.J., 1985, Multivariate estimation with high breakdown point. *Mathematical Statistics and Applications*, 283-297.

Ruts I. and Rousseeuw P.J., 1996, Computing depth contours of bivariate point clouds. *Computational Statistics and Data Analysis*, 23(1): 153-168.

Schickler, W. and Thorpe, A., 2001, Surface estimation based on LIDAR. *Proceedings of the ASPRS Annual Conference*, April 23-27, St. Louis, Missouri, unpaginated CD-ROM.

Shan, J. and Sampath, A., 2005, Urban DEM generation from raw LIDAR data: A labeling algorithm and its performance. *Photogrammetric Engineering & Remote Sensing*, 71(2): 217-226.

Shao, Y.C. and Chen, L.C., 2008, Automated Searching of Ground Points from Airborne Lidar Data Using a Climbing and Sliding Method. *Photogrammetric Engineering & Remote Sensing*, 74(5): 625-635.

Silván-Cárdenas, J.L. and Wang, L., 2006, A multi-resolution approach for filtering LiDAR altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(1): 11-22.

Sithole, G., 2001, Filtering of laser altimetry data using a slope adaptive filter. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 34(3/W4): 203-210.

Sithole, G. and Vosselman, G., 2003, Report: ISPRS Comparison of Filters. *ISPRS Commission III, Working Group 3*.

Sithole, G. and Vosselman, G., 2004, Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59: 85-101.

Sohn, G. and Dowman, I., 2002, Terrain surface reconstruction by the use of tetrahedron model with the MDL Criterion, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 34(3A): 336-344.

Song, J.H., Han, S.H., Yu, K., and Kim, Y.I., 2002, Assessing the possibility of land cover classification using LiDAR intensity data. *Proceedings of the ISPRS Technical Commission III Symposium*, Graz, Austria, September 9-13, 2002, 34(3B): 259-262.

Sotoodeh, S., 2006, Outlier detection in laserscanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(5): 297-302.

Sotoodeh, S., 2007, Hierarchical clustered outlier detection in laser scanner point clouds. *International Archive of Photogrammetry and Remote Sensing*, 35(3/W52): 383–388.

Tang, J., Chen, Z., Fu, A. and Cheung, D., 2002, A Robust Outlier Detection Scheme in Large Data Sets. *6th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Taipei, Taiwan.

Vanicek, P., Krakiwsky, E., 1982, Geodesy: The Concepts. *University of New Brunswick,* Canada, ISBN: 0-444-86149-1, 214-231.

Verma, V., Kumar, R. and Hsu, S., 2006, 3D building detection and modeling from aerial LIDAR data. *IEEE Conference on Computer Vision and Pattern Recognition,* 2: 2213–2220

Vosselman, G., 2000, Slope based filtering of Laser altimetry data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* 33(B3-2): 935-942.

Vosselmann, G. and Maas H.G., (2010). Airborne and Terrestrial Laser Scanning. 1st edn, Whittles Publishing, Dunbeath, Caithness, Scotland, UK.

Wang, C. and Glenn, N.F., 2009, Integrating LiDAR intensity and elevation data for terrain characterization in forested area. *IEEE Geoscience and Remote Sensing Letters,* 9(3): 463- 466.

Wang, C., Menenti, M., Stoll, M.P., Alessandra, F., Enrica, B. and Marco, M., 2009, Separation of Ground and Low Vegetation Signatures in LiDAR Measurements of Salt-Marsh Environments. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7): 2014-2023.

Watkins, D., 2005, LiDAR Types and Uses: with a Case Study in Forestry. State College, PA, USA: Department of Geography, Pennsylvania State University.

Weitkamp, C., 2005, LiDAR: Introduction. In Fujii, T. and Fukuchi, T., editors, *Laser Remote Sensing*, Boca Raton, London, New York and Singapore: Taylor & Francis,1-36.

Wu, G.Q. and Yan, X.F., 2008, Outlier detection based on modified MCD and its performance. *Journal of East China University of Science and Technology*, 34(2): 267-272.

Yamanishi, K., Takeuchi, J., Williams, G. and Milne, P., 2004, On-line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms. *Data Mining and Knowledge Discovery*, 8(3): 275-300.

Yan, W.Y. and Shaker, A., 2009, Radiometric calibration of airborne LiDAR intensity data for land cover classification. *Proceedings of the ISPRS Symposium of Commission I, WG I/2*, June 15-18, TELUS Convention Centre Calgary, Alberta, Canada

Yoon, J.S., Shin, J.I. and Lee, K.S., 2008, Land cover characteristics of airborne LiDAR intensity data: a case study. *IEEE Geoscience and Remote Sensing Letters*, 5(4): 801-805.

Zakšek, K. and Pfeifer, N. 2006, An improved morphological filter for selecting relief points from a LiDAR point cloud in steep areas with dense vegetation. *Technical Report at Delft Institute of EarthObservation and Space systems of the Netherlands*.

Zhang, K. and Whitman, D. 2005, Comparison of three algorithms for filtering airborne lidar data. *Photogrammetric Engineering and Remote Sensing* 71(3): 313-324.

Zhang, K. Q., Chen, S.C., Whitman, D., Shyu, M.L., Yan, J.H. and Zhang, C.C., 2003, A progressive morphological filter for removing nonground measurements from airborne LiDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4): 872-882.

Zhang, T., Ramakrishnan, R. and Livny, M., 1996, An efficient data clustering method for very large databases. *Proceedings of the international conference on management of data*, Montreal, Canada, pp. 103-114.

Appendix A

Main program for COF calculation in Java environment

```
/*******************************************************/
/*                                                     */
/*         Main program for COF calculation            */
/*                                                     */
/*******************************************************/

package processor;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map.Entry;


public class COF implements Serializable{

    private static final long serialVersionUID = 229L;

    private int k; // K value in KNN

    private HashMap<String, String> data;
    private HashMap<String, String> extend_data;

    private HashMap<String, Double> ac_dist_map;
    private HashMap<String, Double> COF; //
    HashMap<String, String[]> knn_points; //

    public COF(HashMap<String, String> data,
            HashMap<String, String> extend_data, int k) throws
IOException {
        this.data = data;
        this.extend_data = extend_data;
        this.k = k;
        init();
    }

    void init() {
        this.COF = new HashMap<String, Double>();
        this.ac_dist_map = new HashMap<String, Double>();
        this.knn_points = new HashMap<String, String[]>();
    }


    public HashMap<String, Double> getCOF() {
        try {
            calculateCOF();
        } catch (IOException e) {
            e.printStackTrace();
```

```
        }
        return COF;
    }

    // Caculate COF
    private void calculateCOF() throws IOException {



        HashMap<String, String> points = new HashMap<String,
String>();


        for (Entry<String, String> e : data.entrySet()) {
            points.clear();


            String knn[] = getKNN(e.getKey(), k, extend_data);
            knn_points.put(e.getKey(), knn);
            int index = 1;
            points.put(index + "", e.getValue());

            for (int i = 0; i < knn.length; i++) {
                index++;
                points.put(index + "", extend_data.get(knn[i]));
            }



            double ac_dist = new EMSTTool().calculate_ac_dist(points,
"1", k);
            ac_dist_map.put(e.getKey(), ac_dist);
        }

        knn_points.clear();
        for (Entry<String, String> e : data.entrySet()) {
            String knn[] = getKNN(e.getKey(), k, data);
            knn_points.put(e.getKey(), knn);
        }
        double sum;
        int index = 1;
        for (Entry<String, String[]> e : knn_points.entrySet()) {
            String[] s = e.getValue();
            sum = 0.0;
            for (int i = 0; i < s.length; i++) {
                sum += ac_dist_map.get(s[i]);
            }
            COF.put(e.getKey(), k * ac_dist_map.get(e.getKey()) /
sum);
            index++;
        }
    }

    private String[] getKNN(String p, int k_value,
            HashMap<String, String> region) {
        String[] point1, point2;
        double distance;

        String points[] = new String[k_value];
        ArrayList<Distance> list = new ArrayList<Distance>();
        String point = data.get(p);
        point1 = DataOperatingUtil.getArray(point);

        for (Entry<String, String> e2 : region.entrySet()) {
```

```
            String p2 = e2.getKey();
            point2 = DataOperatingUtil.getArray(e2.getValue());
            distance = DataOperatingUtil.getDistance(point1, point2);
            if (distance < 0.00000001)
                continue;
            list.add(new Distance(Integer.parseInt(p),
Integer.parseInt(p2),
                    distance, 0));
        }

        Collections.sort(list);
        for (int i = 0; i < k_value; i++) {
            points[i] = list.get(i).p2 + "";
        }
        return points;
    }

    public String[] getKNN(String p, int k_value) {

        String[] point1, point2;
        double distance;

        String points[] = new String[k_value];
        ArrayList<Distance> list = new ArrayList<Distance>();
        String point = data.get(p);
        point1 = DataOperatingUtil.getArray(point);

        for (Entry<String, String> e2 : extend_data.entrySet()) {

            String p2 = e2.getKey();
            point2 = DataOperatingUtil.getArray(e2.getValue());
            distance = DataOperatingUtil.getDistance(point1, point2);
            if (distance < 0.00000001)
                continue;
            list.add(new Distance(Integer.parseInt(p),
Integer.parseInt(p2),
                    distance, 0));
        }

        Collections.sort(list);
        for (int i = 0; i < k_value; i++) {
            points[i] = list.get(i).p2 + "";
        }
        return points;
    }
    public static void main(String args[])throws Exception
    {
        long t1=System.currentTimeMillis();
        File file1=new File("c:/data/partition_0_0.txt");
        File file2=new File("c:/data/extend_partition_0_0.txt");
        int index = 1;
        FileReader fr = new FileReader(file1);
        BufferedReader reader = new BufferedReader(fr);
        String line = "";
        HashMap<String, String> data = new HashMap<String, String>();
        while ((line = reader.readLine()) != null) {
            data.put(index + "", line);
            index++;
        }
        int index2 = 1;
        fr = new FileReader(file2);
        reader = new BufferedReader(fr);
        line = "";
```

```
        HashMap<String, String> extend_data = new HashMap<String,
String>();
        while ((line = reader.readLine()) != null) {
            extend_data.put(index2 + "", line);
            index2++;
        }
        System.out.println("Processing partition_0_0"+" ......");
        COF task = new COF(data, extend_data, 7);

        //save results
        HashMap<String, Double> map=task.getCOF();
        File result=new File("c:/cof_result_0_0"+".txt");
        PrintWriter writer=new PrintWriter(result);
        for (Entry<String, Double> e : map.entrySet())
        {
            writer.println(data.get(e.getKey())+" "+e.getValue());
        }
        writer.close();
        System.out.println(System.currentTimeMillis()-t1);
    }


}



/*******************************************/
/*             Minimum Spanning Tree       */
/*******************************************/
package processor;


import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.Iterator;
import java.util.StringTokenizer;
import java.util.Vector;
import java.util.Map.Entry;


public class EMSTTool implements Serializable{

    private HashMap<String, String> rootPoints;
    private ArrayList<Vector<String>> global;
    private ArrayList<Distance> sorted_list;
    private ArrayList<Distance> list;
    private double ac_dist;
    private int k;

    public EMSTTool() {

    }

    public double calculate_ac_dist(HashMap<String, String> data,
String point,
            int k_value) {
        double ac_dist;
        this.rootPoints = data;
```

```
        this.k = k_value;
        sorted_list = new ArrayList<Distance>();
        list = new ArrayList<Distance>();
        global = new ArrayList<Vector<String>>();
        rootPoints = data;

        for (Entry<String, String> e : rootPoints.entrySet()) {
            Vector<String> v = new Vector<String>();
            v.add(e.getKey());
            global.add(v);
        }
        calculateDistance();
        generateEMST(list);
        ArrayList<String> sbn_path = new ArrayList<String>();
        ArrayList<String> sbn_trails = new ArrayList<String>();
        track(point, list, sbn_path, sbn_trails);
        ac_dist = avgChainDist(sbn_trails);
        return ac_dist;

    }

    private void calculateDistance() {

        String[] point1 = null;
        String[] point2 = null;
        double distance;

        for (Entry<String, String> e1 : rootPoints.entrySet()) {
            point1 = getArray(e1.getValue());

            for (Entry<String, String> e2 : rootPoints.entrySet()) {
                if (e2.getKey().equals(e1.getKey())) {
                    continue;
                }

                point2 = getArray(e2.getValue());

                distance = getDistance(point1, point2);

                sorted_list.add(new
Distance(Integer.parseInt(e1.getKey()),
                        Integer.parseInt(e2.getKey()), distance, 0));
            }
        }

        Collections.sort(sorted_list);

    }

    public double get_ac_dist() {
        return ac_dist;
    }

    private double avgChainDist(ArrayList<String> trails) {
        String points[];
        String[] point1 = null;
        String[] point2 = null;
        double distance;
        double sum = 0.0;
        int r = k + 1;
        for (int i = 1; i <= r - 1; i++) {
            String trail = trails.get(i - 1);
            points = trail.split(" ");
```

```
            point1 =
DataOperatingUtil.getArray(rootPoints.get(points[0]));
            point2 =
DataOperatingUtil.getArray(rootPoints.get(points[1]));
            distance = getDistance(point1, point2);
            sum += 2 * distance * (r - i) / r;

        }
        sum = sum / (r - 1);
        return sum;
    }

    private void generateEMST(ArrayList<Distance> list) {
        Distance dis;
        int p1, p2, c1, c2;
        int index = 0;
        while (global.size() > 1) {
            dis = sorted_list.get(index);
            index++;
            p1 = dis.p1;
            p2 = dis.p2;
            c1 = check_global(p1);
            c2 = check_global(p2);

            if (c1 == c2) {
                continue;
            } else {
                global.get(c1).addAll(global.get(c2));
                global.remove(c2);
            }

            list.add(dis);
        }
    }

    private int check_global(int p) {
        Iterator<Vector<String>> iterator = global.iterator();
        int i = 0;
        while (iterator.hasNext()) {
            if (iterator.next().contains(p + "")) {
                return i;
            }
            i++;
        }
        return -1;
    }

    private void track(String start, ArrayList<Distance> list,
            ArrayList<String> sbn_path, ArrayList<String> sbn_trails)
{

        ArrayList<String> next = new ArrayList<String>();
        ArrayList<Distance> sorted_set = new ArrayList<Distance>();
        if (sbn_path.size() == list.size() + 1)
            return;
        sbn_path.add(start);

        int p1, p2;
        for (Distance s : list) {
            p1 = s.p1;
            p2 = s.p2;
            if (start.equals(p1 + "") && !sbn_path.contains(p2 + ""))
{
                next.add(p2 + "");
```

```
                sorted_set.add(new Distance(Integer.parseInt(start),
p2,
                        s.distance, 0));
            } else if (start.equals(p2 + "") && !sbn_path.contains(p1
+ "")) {
                next.add(p1 + "");
                sorted_set.add(new Distance(Integer.parseInt(start),
p1,
                        s.distance, 0));
            }
        }
        Collections.sort(sorted_set);
        Iterator<Distance> iterator = sorted_set.iterator();
        while (iterator.hasNext()) {
            Distance dist = (Distance) iterator.next();
            sbn_trails.add(dist.p1 + " " + dist.p2);
            track(dist.p2 + "", list, sbn_path, sbn_trails);
        }

    }

    private synchronized String[] getArray(String s) {
        String temp = "";
        StringTokenizer token = new StringTokenizer(s, " ");
        while (token.hasMoreTokens()) {
            temp = temp + token.nextToken() + " ";
        }

        return temp.split(" ");

    }

    public static void main(String[] args) throws IOException {
        File file = new File("data/mst.txt");
        FileReader fr = new FileReader(file);
        BufferedReader reader = new BufferedReader(fr);
        String line = "";
        HashMap<String, String> rootPoints = new HashMap<String,
String>();
        int index = 1;
        while ((line = reader.readLine()) != null) {
            rootPoints.put(index + "", line);
            index++;
        }

        EMSTTool e = new EMSTTool();
        e.calculate_ac_dist(rootPoints, "3", 9);

    }

    public static double getDistance(String p1[], String p2[]) {
        double distance;

        distance = Math.sqrt((str2double(p1[0]) - str2double(p2[0]))
                * (str2double(p1[0]) - str2double(p2[0]))
                + (str2double(p1[1]) - str2double(p2[1]))
                * (str2double(p1[1]) - str2double(p2[1]))
                + (str2double(p1[2]) - str2double(p2[2]))
                * (str2double(p1[2]) - str2double(p2[2])));
        return distance;
    }

    public static double str2double(String s) {
        return Double.parseDouble(s);
```

```
    }
}
```

Appendix B

Main program for MCD-based multiple attributes model for outlier detection in LiDAR data in MATLAB environment

```
%=MCD-based multiple attributes model for outlier detection in LiDAR
data=%

%======================Configuration====================%
input_filename='COF/COF_9.txt';% input data,k=9 for COF
c=5617;% iteration starting point
threshold=2.4474;% threshold
%==================Programming==================%
cof=importdata(input_filename);
max_height=max(cof(:,7));
min_height=min(cof(:,7));
ptsymb = {'r^','c+','b.','r+','b^','.','g^','r^','r.','c+','ro','b^'};
index=1;
B=cof(:,10);
A=cof(:,7);
B=nthroot(B,4);
points=[cof(:,5),cof(:,6),cof(:,7),cof(:,10)];
mcd=[A,B];
len=length(mcd);
mcd_mean=mean(mcd)
mcd_cov=cov(mcd)
md1=zeros(len,1);
for i=1:length(mcd)
   md1(i,1)=(mcd(i,:)-mcd_mean)*inv(mcd_cov)*(mcd(i,:)-mcd_mean)';
end
points=[points(:,1),points(:,2),points(:,3),points(:,4),md1(:,1)];
result=sortrows(points,5);
```

```
x=3;

while(x>1)

sum_me=0;

for j=1:(len-c+1)
    mu(j)=sum(result(1:(j+c-1),5))/(j+c-1);
    for k=1:(j+c-1)
        sum_me=sum_me+(result(k,5)-mu(j))^2;
    end
    sigma(j)=sqrt(sum_me/(j+c-2));
end

delt_sigma=abs(sigma'-sqrt(2*2));
[x,y]=find(delt_sigma==min(delt_sigma(:)));
h=c+x-1;
c=h;
mu=0;
sigm=0;

end

result2=result(1:c,:);
mcd2=[result2(:,3),result2(:,4)];
mcd_mean=mean(mcd2)
mcd_cov=cov(mcd2)
md2=zeros(len,1);

for i=1:length(mcd)
   md2(i,1)=(mcd(i,:)-mcd_mean)*inv(mcd_cov)*(mcd(i,:)-mcd_mean)';
   md2(i,1)=sqrt(md2(i,1));
end

points=[points(:,1),points(:,2),points(:,3),points(:,4),md2(:,1),cof(:
,9)];
outlier=points(find(points(:,5)>threshold),:);


%=========================Figures=======================%

%3D figure
figure;
plot3(points(:,1),points(:,2),points(:,3),'.');
hold on;
plot3(outlier(:,1),outlier(:,2),outlier(:,3),'r^')
hold on;
grid on;

%2D figure
figure;
plot(points(:,3),points(:,4),'.');
hold on;
outlier=points(find(points(:,5)> threshold),:);
plot(outlier(:,3),outlier(:,4),'r^');
grid on;
```

Appendix C

Main program for MCD-based multiple attributes model for LiDAR data filtering in MATLAB environment

```
%==MCD-based multiple attributes model for LiDAR data filtering==%
Samp31_net_rawdata=dlmread('d:\samp31_net_rawdata88.txt');%8*8 patches
Net_rawdata=Samp31_net_rawdata(find(Samp31_net_rawdata(:,10)==14),:);%
No.14 patch


%================Statistical information==============%

Size=size(Net_rawdata,1)
H_Mean=mean(Net_rawdata(:,7))
H_Median=median(Net_rawdata(:,7))
H_Maximum=max(Net_rawdata(:,7))
H_Minimum=min(Net_rawdata(:,7))
H_STD=std(Net_rawdata(:,7))
I_Mean=mean(Net_rawdata(:,8))
I_Median=median(Net_rawdata(:,8))
I_Maximum=max(Net_rawdata(:,8))
I_Minimum=min(Net_rawdata(:,8))
I_STD=std(Net_rawdata(:,8))


%================Configuration====================%


Net=Net_rawdata;
std(Net(:,7))
Paul=Net_rawdata(find(Net_rawdata(:,9)==1),:);
size(Paul,1)
net=[Net(:,1),Net(:,2),Net(:,3),Net(:,4),Net(:,5),Net(:,6),Net(:,7),Ne
t(:,8),Net(:,9)];
Threshold=2.1459;%threshold
output_filename='terrain_net1.txt';% output terrain points
```

```
output_filename2='off-terrain_net1.txt';% output off-terrain points

output_filename3='AccuracyAssessment_net1.txt';% output Accuracy
Assessment result



%====================Programming====================%


ptsymb = { 'r^','g^','c+','b.','r+','.','g^','r^','r.','b+','ro',};
index=1;

A=net(:,7);
B=net(:,8);
B=nthroot(B,2);
B=0.6*B;%weight for intensity


points=[net(:,5),net(:,6),net(:,7),net(:,8)];

mcd=[A,B];
len=length(mcd);

c= fix(len*0.6);

mcd_mean=mean(mcd);
mcd_cov=cov(mcd);

md1=zeros(len,1);

Invmcdcov=inv(mcd_cov);

for i=1:length(mcd)
   md1(i,1)=(mcd(i,:)-mcd_mean)*Invmcdcov*(mcd(i,:)-mcd_mean)';
end

points=[points(:,1),points(:,2),points(:,3),points(:,4),md1(:,1)];
result=sortrows(points,5);


x=3;
while(x>1)

sum_me=0;

for j=1:(len-c+1)
    mu(j)=sum(result(1:(j+c-1),5))/(j+c-1);
    for k=1:(j+c-1)
        sum_me=sum_me+(result(k,5)-mu(j))^2;
    end
    sigma(j)=sqrt(sum_me/(j+c-2));
end

delt_sigma=abs(sigma'-sqrt(2*2));

[x,y]=find(delt_sigma==min(delt_sigma(:)));
h=c+x-1;
c=h;
mu=0;
sigm=0;
end
```

```
result2=result(1:c,:);

mcd2=[result2(:,3),result2(:,4)];
mcd_mean2=mean(mcd2);
mcd_cov2=cov(mcd2);
Rmd2=zeros(len,1);

Invmcdcov=inv(mcd_cov2);

for i=1:length(mcd)
   Rmd2(i,1)=(mcd(i,:)-mcd_mean2)*Invmcdcov*(mcd(i,:)-mcd_mean2)';
   Rmd(i,1)=sqrt(Rmd2(i,1));
end

points=[net(:,1),net(:,2),net(:,3),net(:,4),points(:,1),points(:,2),po
ints(:,3),points(:,4),Rmd(:,1),net(:,9)];
outlier=points(find(points(:,9)>Threshold),:);



%=====================Figures=========================%


%2-D figure

figure;

plot(points(:,7),points(:,8),'.');

hold on;

outlier=points(find(points(:,9)>Threshold),:);

plot(outlier(:,7),outlier(:,8),'r^');
grid on;

%3D figure z for height

figure;

plot3(points(:,5),points(:,6),points(:,7),'.');

hold on;

plot3(outlier(:,5),outlier(:,6),outlier(:,7),'r^');

grid on;

%3D figure z for intensity

figure;

plot3(points(:,5),points(:,6),points(:,8),'.');

hold on;

plot3(outlier(:,5),outlier(:,6),outlier(:,8),'r^');

grid on;


%================Accuracy Assessment=============%
```

```
clean=points(points(:,9)<Threshold,:);
fidout=fopen(output_filename,'w');
for i=1 : length(clean)
     fprintf(fidout,'%s\n',num2str(clean(i,:)));
end
fclose(fidout);


clean=points(points(:,9)>=Threshold,:);
fidout=fopen(output_filename2,'w');
for i=1 : length(clean)
     fprintf(fidout,'%s\n',num2str(clean(i,:)));
end
fclose(fidout);



check2=find(readt(:,10)==0);

size(check2,1)% a

readofft=dlmread(output_filename2);

check3=find(readofft(:,10)==0);

size(check3,1)% b

readt=dlmread(output_filename);

check1=find(readt(:,10)==1);

size(check1,1)% c


check4=find(readofft(:,10)==1);

size(check4,1)% d

Type1error=size(check3,1)/(size(check2,1)+size(check3,1));

Type2error=size(check1,1)/(size(check1,1)+size(check4,1));

Totalerror=(size(check3,1)+size(check1,1))/len;
```